NeuroImage 264 (2022) 119729

Contents lists available at ScienceDirect

NeuroImage

journal homepage: www.elsevier.com/locate/neuroimage

BLMM: Parallelised computing for big linear mixed models

Thomas Maullin-Sapey[a,*], Thomas E. Nichols[a,b]

[a] Big Data Institute, Li Ka Shing Centre for Health Information and Discovery, Nuffield Department of Population Health, University of Oxford, Oxford, UK
[b] Wellcome Centre for Integrative Neuroimaging, FMRIB, Nuffield Department of Clinical Neurosciences, University of Oxford, Oxford, UK

ABSTRACT

Within neuroimaging large-scale, shared datasets are becoming increasingly commonplace, challenging existing tools both in terms of overall scale and complexity of the study designs. As sample sizes grow, researchers are presented with new opportunities to detect and account for grouping factors and covariance structure present in large experimental designs. In particular, standard linear model methods cannot account for the covariance and grouping structures present in large datasets, and the existing linear mixed models (LMM) tools are neither scalable nor exploit the computational speed-ups afforded by vectorisation of computations over voxels. Further, nearly all existing tools for imaging (fixed or mixed effect) do not account for variability in the patterns of missing data near cortical boundaries and the edge of the brain, and instead omit any voxels with any missing data. Yet in the large-*n* setting, such a voxel-wise deletion missing data strategy leads to severe shrinkage of the final analysis mask. To counter these issues, we describe the "Big" Linear Mixed Models (BLMM) toolbox, an efficient Python package for large-scale fMRI LMM analyses. BLMM is designed for use on high performance computing clusters and utilizes a Fisher Scoring procedure made possible by derivations for the LMM Fisher information matrix and score vectors derived in our previous work, Maullin-Sapey and Nichols (2021).

1. Introduction

1.1. Background

The field of functional Magnetic Resonance Imaging (fMRI) has recently seen a drastic improvement in terms of the volume of data collected and shared publicly. Many researchers now regularly face analyses involving "large-*n*" (large number of observations) datasets consisting of tens of thousands of images, typically endowed with some form of complex covariance structure (Smith and Nichols, 2018; Li et al., 2019; Haworth et al., 2019). For example, the Adolescent Brain Cognitive Development (ABCD) and UK Biobank (UKB) datasets, which contain imaging data from 10,000 and 30,000 subjects, respectively, both possess a multi-level covariance structure induced through a repeated measures experimental design (Casey et al., 2018; Allen et al., 2012). Similarly, the well-known Enhancing NeuroImaging Genetics through Meta-Analysis (ENIGMA) cohort, which contains imaging data from tens of thousands of subjects, also possesses a multi-level covariance structure due to its pooling of data from many different sources (Bearden and Thompson, 2017). A more complex, but equally popular example is the Human Connectome Project (HCP) dataset, which contains observations drawn from 1200 subjects, but exhibits a constrained covariance structure due to relatedness between individuals and the sampling of family units (Essen et al., 2013).

Often, covariance structures in an experimental design arise from grouping factors present in the data. Accounting for complex grouping factors during an fMRI analysis is an historically routine practice for small-sample studies. Commonly employed analysis designs in the small-sample setting include longitudinal multi-session analyses (observations grouped by subject), comparative group analyses (subjects grouped by study conditions) and mega-analyses (analysis results grouped according to study protocols). The widely-accepted, conventional approach to modelling such datasets is to employ the Linear Mixed Model (LMM) (c.f. Laird and Ware, 1982; Friston et al., 2002). The LMM accounts for complex grouping structures in datasets via the inclusion of both "fixed effects" and "random effects" during model specification. Fixed effects are unknown constant parameters that are associated with covariates in the experimental design. Random effects are random variables that model the systematic differences between instances of a categorical variable (e.g. between-subject differences, between-site differences).

There are many strong obstacles to the practical execution of LMM analyses for large-*n* fMRI datasets. Generally, an LMM analysis consists of two stages: parameter estimation and statistical inference, each of which presents unique computational and theoretical challenges in the large-*n* fMRI analysis setting. An overview of the challenges specific to each of these stages is provided in Sections 1.1.1 and 1.1.3, respectively. An additional issue, which merits separate consideration, concerning missing data found near cortical boundaries, is detailed in Section 1.1.2. In the univariate (non-imaging) setting, many tools exist for estimating the parameters of and performing inference upon the LMM (c.f. Section 2.1.3). However, many of these tools are not designed to (a) be scalable to arbitrarily large datasets and (b) exploit vectorisation speed-ups from processing multiple voxels simultaneously. To counter this, in recent work we derived novel closed form expressions for the Fisher information matrices and LMM gradient vectors (Maullin-Sapey and Nichols, 2021), making vectorised Fisher scoring practical for

[*] Corresponding author.
E-mail address: thomas.maullin-sapey@bdi.ox.ac.uk (T. Maullin-Sapey).

https://doi.org/10.1016/j.neuroimage.2022.119729.
Received 1 March 2022; Received in revised form 29 October 2022; Accepted 31 October 2022
Available online 4 November 2022.
1053-8119/© 2022 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

mass-univariate analysis. In this work, we shall demonstrate that these expressions may be employed to perform fast and scalable LMM parameter estimation and inference in the context of large-*n* fMRI analysis.

In this paper, we present “Big” Linear Mixed Models (BLMM), a Python-based tool for parameter estimation and statistical inference of mass-univariate LMMs. The BLMM tool partitions fMRI analyses to limit memory consumption, while still being able to exploit vectorization speed-ups from working with multiple voxels. Despite being built specifically for use on Sun Grid Engine (SGE) clusters, SGE-specific code in BLMM is isolated in one file (‘blmm_cluster.sh’) and may be adapted for use on any HPC scheduler. In the following sections, we first provide background on LMM parameter estimation and inference as well as the ǣvoxel-wise missingnessg ubiquitous in large-*n* fMRI analyses. Following this, we give preliminary statistical information describing the univariate LMM and its extension to the mass-univariate voxel-wise setting. In the methods section, we then outline the computational pipeline of BLMM, starting with the input specification, followed by the distributed stages of computation, parameter estimation and finally, inference. Next, the correctness and performance of BLMM are assessed via simulation. We conclude by providing a real data example based on the UK Biobank.

1.1.1. LMM parameter estimation

A vast amount of literature exists on the development of LMM parameter estimation tools and methodology. Primarily proposed in the late 1970s and early 1980s, many early approaches to LMM parameter estimation involved performing likelihood maximisation via numerical methods such as Fisher Scoring, Newton-Raphson and Expectation Maximisation (c.f. Dempster et al., 1977; Jennrich and Schluchter, 1986; Laird et al., 1987). More recently, several tools which build upon these fundamental ideas have become available for LMM parameter estimation in the univariate (single-model) setting. The most popular of these include the SAS and SPSS packages, PROC-MIXED and MIXED (SAS Institute, 2015; Corp, 2015), the R package lme4 (Bates et al., 2015) and the Windows package HLM (Raudenbush et al., 2019). These tools have been widely adopted within the statistical literature, largely due to their demonstrable computational efficiency when performing parameter estimation for a single univariate model.

However, in mass-univariate fMRI, parameter estimation is not performed for a single model, but rather for hundreds of thousands of models, each corresponding to a different voxel in the analysis mask. For a mass-univariate voxel-wise analysis to truly utilise the computational power available, it is a necessity that computation is vectorised across voxels. Many of the established LMM tools are reliant upon operations that are not naturally amenable to vectorisation. Examples of such operations include the sparse Cholesky decomposition employed by lme4 (Bates and DebRoy, 2004) and the sparse sweep operation employed by PROC-MIXED and MIXED (Wolfinger et al., 1994). The approaches employed by HLM circumvent this issue, but at the cost of generalizability since HLM only allows the estimation of LMMs which exhibit pre-specified structures (Raudenbush and Bryk, 2002). Operations that are not amenable to vectorisation create bottlenecks for mass-univariate computation as they must be executed separately for each voxel in the image. Serial execution of such operations can result in severe computational overheads, especially when modelling large sample sizes. As a result, many of the tools available for univariate LMM analysis cannot be employed in the large-*n* fMRI setting.

In the small-sample fMRI setting, several tools exist for mass-univariate LMM parameter estimation. These include SPM’s built-in mixed-effects module (Friston et al., 2005), FSLs FLAME package (Beckmann et al., 2003), FreeSurfers longitudinal analysis pipeline (Bernal-Rusiel et al., 2013a) and AFNIs 3dLME package (Chen et al., 2013). The tools provided by SPM, FSL and FreeSurfer perform parameter estimation for only LMMs in which observations are grouped by a single categorical factor. Specifically, SPM and FSL allow LMM parameter estimation of second-level designs (i.e. designs with subjects grouped by experimental features) whilst FreeSurfer allows for parameter estimation of longitudinal LMM designs (i.e. designs with timepoints grouped by subject) (Group, 2020; Woolrich et al., 2004; Bernal-Rusiel et al., 2013b; Madhyastha et al., 2018). In SPM and FreeSurfer, parameter estimates are obtained via Restricted Maximum Likelihood (REML) estimation, whilst in FSL a Bayesian approach is adopted. In contrast, AFNI’s 3dLME package allows parameter estimation of a much broader range of LMM designs and provides similar options to those offered by standard tools in the univariate setting. 3dLME provides this support by calling directly to the R package lme4 and parallelising computation across all available processors and cluster nodes via the use of the Dask Python package (Rocklin, 2015).

The tools provided by SPM, FSL, FreeSurfer and AFNI are computationally efficient for parameter estimation in the small-sample mass-univariate analysis setting, but were not originally designed for applications involving large sample sizes. In the context of large-*n* analyses, SPM, FSL and FreeSurfer quickly encounter memory errors as sample sizes increase into the hundreds whilst AFNI experiences overheads in terms of computation time. For all tools, reduced computational performance in the large-*n* setting primarily stems from two issues: (1) construction and storage of the analysis design (i.e. the design matrices and response vector) and (2) bottleneck computations which must be performed independently for each voxel.

1.1.2. Missing data

An important issue that must be addressed prior to or during the parameter estimation stage of an fMRI analysis is the missing data observed on and around cortical boundaries. Such missingness is ubiquitous in whole-brain fMRI analyses and can be attributed to several commonplace sources of between-image spatial variability. Such sources include magnetic susceptibility artefacts, imperfections in the image alignment process, differing image acquisition parameters and, indirectly, between-subject biological variation. Conventionally, standard fMRI analysis tools address this missing data problem by omitting voxels with incomplete observations from the analysis. As detailed by Vaden et al. (2012), and later by Gebregziabher et al. (2017), adopting this approach can negatively impact both the specificity and sensitivity of the analysis results, especially when spatial extent thresholding is employed for multiple comparisons correction. In terms of specificity, an inflated Type II error rate may be observed when the removal of voxels with incomplete data causes brain regions that are near cortical boundaries to be excluded from the analysis. In terms of sensitivity, an inflated Type I error rate may result from the smaller number of tests being performed, and consequently, the use of a less conservative multiple comparisons correction. Whilst the removal of missing-data voxels in the small-sample setting typically has a negligible effect, in the large-*n* setting omitting voxels can profoundly influence results of an analysis, often deleting large chunks of the final images produced. The reason that the severity of this issue becomes notably more pronounced in the large-*n* setting is that the probability of a given voxel being missing in at least one image increases with the number of images in the analysis. To address this issue, the patterns of missingness observed for each voxel must be carefully considered and accounted for in large-*n* analyses.

1.1.3. Inference

In the small-sample setting, fMRI LMM analyses conclude with significance-based hypothesis tests for the fixed-effects (predictors) in the model by using Wald test statistics. Commonly adopted Wald-based hypothesis testing procedures include the T-test and the F-test. The T-test is used to assess whether linear relationships exist between the BOLD response and model predictors (such as age, weight and experimental design), whilst the F-test is used to assess whether the inclusion of predictors in a model improves the model’s goodness of fit. In some circumstances, it may also be of practical interest to assess whether the inclusion of random effects in the model improves the model fit. Such questions are less commonly considered, but may be addressed using a likelihood ratio test (c.f. Stram and Lee, 1995; Verbeke and Molenberghs, 2001).

Hypothesis testing of fixed effects has long been a contentious topic in the broader LMM literature, due to unknown variability in the estimation of variance components and lack of exact distribution for the Wald test statistics (Verbeke and Molenberghs, 2001). As a result, whilst many of the popular tools for univariate LMM analysis provide support for calculating Wald T-statistics and F-statistics, there is debate concerning how the corresponding p-values are to be calculated and whether such practices should be widely adopted (c.f. Manor and Zucker, 2004; Baayen et al., 2008; Luke, 2017).

The lack of consensus in the LMM literature is reflected by the range of options available for performing LMM inference in the univariate setting. For example, HLM, MIXED and PROC-MIXED each adopt the assumption that the Wald test statistics for the LMM 'approximately' follow students t- and F-distributions, but employ different techniques for estimating the associated unknown degrees of freedom. HLM approximates the degrees of freedom via closed-form expressions resembling those employed for multi-level linear model analyses (c.f. Raudenbush and Bryk, 2002; West et al., 2014). On the other hand, MIXED and PROC-MIXED utilize the Welch-Satterthwaite equation (c.f. Section 2.1.4) and numerical gradient optimization (Satterthwaite, 1946; Welch, 1947; Fai and Cornelius, 1996) in order to obtain degrees of freedom estimates for a more general breadth of LMM applications. For brevity, in the remainder of this paper we shall refer to methods involving degrees of freedom estimation using the Welch-Satterthwaite equation as 'WSDF' (Welch-Satterthwaite Degrees of Freedom) based methods.

By employing this 'approximate distribution' assumption, HLM, MIXED and PROC-MIXED are able to provide support for significance-based hypothesis testing, outputting p-values alongside Wald statistics for a wide variety of LMMs. While lmer rejects any approximation and offers no p-values (Bates, 2006), the supporting lmerTest package (Kuznetsova et al., 2017) provides p-values by using the same methods as MIXED and PROC-MIXED. Numerous studies have found that the WSDF-based approach employed by MIXED, PROC-MIXED and lmerTest is notably more robust to small sample sizes, unbalanced designs, covariance heterogeneity (when fitting the correct covariance structure) and non-normality than the approach adopted by HLM (c.f. Keselman et al., 1999; Schaalje et al., 2002; Kuznetsova et al., 2017; Luke, 2017; Francq et al., 2019). However, the computational burden of the numerical gradient estimation required for the WSDF-based approach is substantial and constitutes a significant obstacle to large-n LMM analysis in the mass-univariate fMRI setting.

Several tools are available in the small-sample fMRI setting for significance-based hypothesis testing via Wald statistics. As with LMM parameter estimation, these include SPM's built-in mixed-effects module, FSL's FLAME package, FreeSurfer's longitudinal analysis pipeline and AFNI's 3dLME package. Due to the low computational costs required, SPM, FreeSurfer and FSL each employ a similar approach to that used by HLM in the univariate setting by using closed-form expressions, which can be found, for example, in Pinheiro and Bates (2009), to approximate the degrees of freedom. AFNI alternately employs the WSDF approach by acting as a wrapper for the lmerTest package. While the former approach is more efficient in terms of computation time and memory, it provides less accurate estimates for the degrees of freedom. Conversely, the latter approach provides more accurate estimation of the approximate sampling distributions of the Wald test statistics, but at the cost of increased computation time that scales with the number of observations. In this work, we make use of recent novel closed-form expressions we developed for evaluating the gradients required by the WSDF approach (Maullin-Sapey and Nichols, 2021). These expressions offer a viable and accurate alternative to gradient estimation and may be employed in the large-n fMRI setting using vectorised computation without sacrificing statistical accuracy.

1.2. Preliminaries

In this section, a brief overview and statement of the mass-univariate LMM is provided. To simplify notation, we begin by defining the univariate LMM in Section 1.2.1. Following this, in Section 1.2.2 we describe how the definition and notation of Section 1.2.1 are extended to the mass-univariate fMRI setting.

1.2.1. The linear mixed model

In the traditional univariate setting, an LMM containing n observations is assumed to take the following form:

$$\begin{aligned} &Y = X\beta + Zb + \epsilon \\ &\epsilon \sim N(0,\sigma^2 I_n),\, b \sim N(0,\sigma^2 D) \end{aligned} \tag{1}$$

where the observed quantities are the response vector Y, fixed effects design matrix X, and random effects design matrix Z, and the unknown model parameters are the fixed effects parameter vector β, the scalar fixed effects variance σ^2, and the random effects covariance matrix D.

The random effects in the model are specified using factors (categorical variables which group the random effects) and levels (the individual instances of the categorical factors). The total number of factors that group the random effects in the model is denoted as r. For the k^{th} factor in the model, l_k and q_k are used to denote the number of levels belonging to the factor and the number of random effects that the factor groups, respectively. The random effects design matrix, Z, can be partitioned horizontally as $Z = [Z_{(1,1)}, \dots, Z_{(1,l_1)}, Z_{(2,1)}, \dots, Z_{(r,l_r)}]$ where $Z_{(k,j)}$ consists of the random effects covariates which are grouped into the j^{th} level of the k^{th} factor in the model. The random effects covariance matrix, D, is block diagonal and can be specified as $D = \bigoplus_{k=1}^{r}(I_{l_k} \otimes D^k)$, where $\bigoplus$ is the direct sum and D^k is the $(q_k \times q_k)$ within-level covariance matrix for the k^{th} factor in the model. This notation is essential for describing the Fisher Scoring algorithm approach that will be employed by BLMM for parameter estimation (see Section 2.1.3). To aid understanding, several worked examples of this notation's usage in practice are provided in Supplementary Material Section S1, alongside extensive discussion of the model covariance structures that may be estimated using BLMM.

From Eq. (1), the restricted log-likelihood function for the LMM, ignoring constant terms, is given by:

$$l_R(\theta) = -\frac{1}{2}\left\{ (n-p)\log(\sigma^2) + \sigma^{-2}e'V^{-1}e + \log|V| + \log|X'V^{-1}X| \right\} \tag{2}$$

where θ is shorthand for the parameters (β, σ^2, D), p is the number of fixed effect parameters in the design, $V = I_n + ZDZ'$ is the marginal variance, and $e = Y - X\beta$ is the residual vector. Throughout this work, we assume that θ takes the form $\theta = [\beta', \sigma^2, \mathrm{vec}(D^1)', \dots, \mathrm{vec}(D^r)']'$, where vec represents the vectorization operator. It should be noted that whilst this may seem like a natural representation of θ, it is by no means the only possible representation. A full discussion of this, alongside a more detailed introduction to the LMM and the notation described in this section, is provided in our previous work, Maullin-Sapey, Nichols, 2021.

1.2.2. The mass univariate model

In the mass-univariate setting we fit and infer on hundreds of thousands of LMMs concurrently. In the setting of fMRI, each LMM corresponds to a voxel in the study's analysis mask. Adapting the notation of the previous section, this is represented as:

$$\begin{aligned} &Y_v = X\beta_v + Zb_v + \epsilon_v \\ &\epsilon_v \sim N(0,\sigma_v^2 I_n), \qquad b_v \sim N(0,\sigma_v^2 D_v) \end{aligned} \tag{3}$$

where the subscript v represents voxel number. In Eq. (3), the fixed effects and random effects design matrices (X and Z) are treated as constant across all voxels, whilst the response vector (Y), design parameters (β, σ^2 and D) and random terms (ϵ and b) vary from voxel to voxel. By extension, this also means that $n, r, \{l_k\}_{k\in\{1,\dots r\}}$ and $\{q_k\}_{k\in\{1,\dots r\}}$ are also treated as constant across voxels. Eq. (3) extends the conventional form of the univariate LMM to the mass-univariate setting. As hierarchical, multi-stage designs can be expressed using the above formulation (c.f. Pinheiro and Bates, 2009; West et al., 2014), the model specifications adopted by the each of the existing fMRI LMM software packages may be viewed as particular instances of Eq. (3). As noted in Section 1.1.2, this model does not account for mask-variability and, as a result, must be adapted to reflect the pattern of missingness observed at each voxel.

To account for such missingness in Y_v, we adopt an MCAR (Missing Completely At Random) assumption and define the $(n \times n)$-dimensional 'missingness matrix', M_v, as a diagonal indicator matrix, where the $(i,i)^{th}$ element is 1 if the i^{th} element of Y_v is not missing and 0 otherwise. We now define $X_v = M_v X$ and $Z_v = M_v Z$ and assume that missingness in Y_v and ϵ_v is encoded as 0. The model specification for a mass-univariate LMM that accounts for missingness induced by mask-variability is now given as follows:

$$\begin{aligned} &Y_v = X_v\beta_v + Z_v b_v + \epsilon_v \\ &\epsilon_v \sim N(0, \sigma^2_v M_v), \qquad b_v \sim N(0, \sigma^2_v D_v) \end{aligned} \tag{4}$$

The inclusion of the missingness matrix, M_v, ensures that rows of the design at which missingness occurred for voxel v are replaced with zeros, or "zero-ed out". This process ensures that the analysis proceeds as though such 'missing-data' rows had not been included in the design at all.

An important ramification of this model construction is that the fixed effects and random effects design, X_v and Z_v, as well as the number of observations, n_v, are now treated as spatially varying. If naively implemented, a model involving a spatially varying design (such as Eq. (4)) presents a much more formidable computational challenge than its non-spatially varying counterpart (i.e. Eq. (3)), as additional expenses in memory and computation time arise from accounting for voxel-specific design matrices. However, in this context, it must be noted that as X_v and Z_v are constructed by 'zero-ing out' rows of X and Z, many voxels will employ identical design matrices for analysis (i.e. it is often the case that $X_{v_1} = X_{v_2}$ and $Z_{v_1} = Z_{v_2}$ for two separate voxels v_1 and v_2). In particular, it is expected that inside the brain, far from cortical boundaries, no missingness will be observed, and therefore, for many voxels, $X_v = X$ and $Z_v = Z$. In other words, whilst it is true that the design matrices, X_v and Z_v, vary spatially across the whole brain, it is expected that large contiguous groups of voxels will exist over which X_v and Z_v do not vary at all. Accounting for this "between-voxel design commonality" can result in drastic improvements in computation speed and efficiency, and thus, constitutes a key motivation behind the approach adopted by BLMM.

In the following sections, unless stated otherwise the subscript v, representing voxel number, will be dropped from our notation. For the remainder of this work, it is assumed implicitly that any equations provided correspond to a model of the form (4) for some given voxel.

2. Methods

In this section, we describe the computational pipeline employed by BLMM to perform mass-univariate LMM analysis of large-n fMRI data, as well as the simulations and real-data examples for which results are later presented in Section 3. To begin, Section 2.1 gives an in-depth overview of the stages of the BLMM computational pipeline. Following this, Section 2.2 describes simulations that shall be used to assess the correctness and performance of BLMM. Finally, Section 2.3 details a real-world example based on repeated-measures data drawn from the UK Biobank, demonstrating BLMM's usage in practice.

2.1. The BLMM pipeline

A visual overview of the BLMM pipeline is provided by Fig. 1 in the form of an activity diagram. Highlighted in Fig. 1 are the four "stages" of the BLMM pipeline: input specification, product form computation, parameter estimation, and inference and output. Each of these stages is described in turn by Sections 2.1.1–2.1.4, respectively. Also labelled in Fig. 1 are the steps of the pipeline which employ distributed computation: "Image-wise batching" and "Voxel-wise batching", each of which is further discussed further in Sections 2.1.2 and 2.1.3–2.1.4, respectively.

2.1.1. Input specification

To specify an analysis design in BLMM, the user must provide the fixed effects design matrix, X, the response images, Y, and the random effects design matrix, Z. For specification of the random effects design matrix, Z, a similar approach to that of lmer is adopted (Bates et al., 2015). For each factor in the design, the user provides a factor vector, g_k, and a "raw" regressor matrix, z_k. The raw regressor matrix, z_k, is a matrix of dimension $(n \times q_k)$ and contains the covariates which correspond to the random effects that are to be grouped by the k^{th} factor in the model. The factor vector, g_k, is a numerical vector of dimension $(n \times 1)$ with entries indicating to which level of the k^{th} factor each observation belongs. From these inputs, the following construction is used to obtain the random effects design matrix Z:

$$Z = \left[J_1' * z_1', \quad \dots, \quad J_r' * z_r'\right], \quad \text{where} \quad (J_k)_{[i,j]} = \begin{cases} 1 & \text{if } (g_k)_{[i]} = j \\ 0 & \text{otherwise} \end{cases}$$

and $*$ is the Khatri-Rao product. For example, in a longitudinal design, g_1 would be a vector of subject identifiers and z_1 could contain a column of 1's for a random intercept and a column of study times for a random slope for the time effect. See Supplementary Material Section S1 for several worked examples.

During input specification, BLMM also provides a range of masking options. By default, the user is required to specify an analysis mask. In addition, the user may specify one mask per input image, as well as a "missingness threshold". The missingness threshold, which may be specified as either a percentage or an integer, indicates how many input images a voxel must have recorded data for in order for it to be retained in the final analysis. This threshold is essential, as while accommodating missingness is an essential feature of BLMM, allowing excessive missingness (down to a small faction of the data) is not advised and may result in rank-deficient designs. Implicit masking is also supported by BLMM, with any voxel set to 0 or NaN in an input image being treated as 'missing' from the analysis.

Whilst specifying the analysis model, the user may also opt to perform hypothesis testing using an approximate Wald $T-$test or $F-$test. To specify a hypothesis test of this form, the user must provide a contrast vector, L, representing the null hypothesis, and the type of statistic to be used to perform the test (i.e. T or F). Further detail on hypothesis testing via approximate Wald statistics is provided in Section 2.1.4.

2.1.2. Product form computation

Computation within the BLMM pipeline begins by, for each voxel v, computing the "product forms" defined as follows:

$$\begin{aligned} &P_v = X_v'X_v, \quad Q_v = X_v'Y_v, \quad R_v = X_v'Z_v, \\ &S_v = Y_v'Y_v, \quad T_v = Y_v'Z_v, \quad U_v = Z_v'Z_v. \end{aligned} \tag{5}$$

Following the computation of the product forms, the original matrices, X_v, Y_v and Z_v, can be discarded since only the product forms are required for future computation (c.f. Sections 2.1.3–2.1.4). This approach is adopted by BLMM as the dimensions of the product forms do not scale with n but rather with p and q, the second dimensions of the fixed effects and random effects design matrices, respectively. As p and q are assumed to be much smaller than n, working with the product forms instead of X_v, Y_v and Z_v can provide large reductions in both memory consumption and computation time (c.f. Maullin-Sapey and Nichols, 2021).

To compute the product forms efficiently, BLMM employs an image-wise batching approach. In this approach, the input images and model matrices are split into batches and computation is performed in parallel across several cluster nodes. More precisely, given B nodes, X and Z are vertically partitioned into evenly sized blocks $\{X^{(b)}\}_{b\in\{1,\dots,B\}}$ and

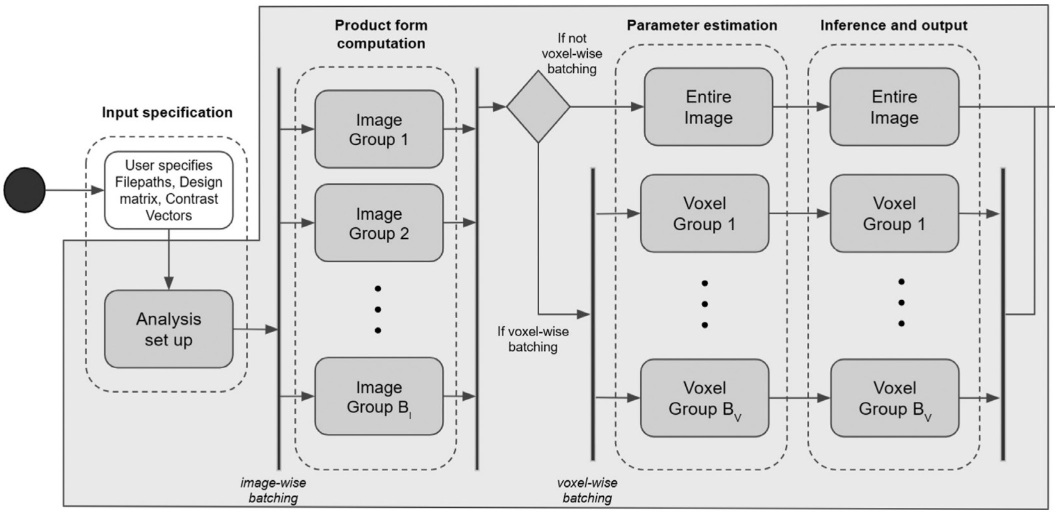

Fig. 1. Activity diagram detailing the BLMM pipeline. The boundary of the BLMM code is indicated by the gray outline. The start and end nodes of the pipeline are represented by the black circle and nested black and white circles, respectively. Decision nodes are represented by diamonds and parallel stages of computation are represented with vertical bars. Also included are dotted lines indicating the distinct "stages" of the BLMM pipeline. Of particular note are the image-wise and voxel-wise batching stages of the pipeline, in which computation is parallelised across B_I groups of images and B_v groups of voxels, respectively.

$\{Z^{(b)}\}_{b\in\{1,\dots,B\}}$, respectively. The list of input images, Y, is also partitioned into B lists of $\frac{n}{B}$ input images, $\{Y^{(b)}\}_{b\in\{1,\dots,B\}}$, each corresponding to a partition of X and Z. Each node is assigned a partition of X, the corresponding partition of Z and the corresponding list of input images. For every voxel in the analysis mask, this means that the b^{th} node now possesses $\frac{n}{B}$ observations (assuming missing values are encoded as zero). We denote the response vector of observations at voxel v, taken from the images $Y^{(b)}$, as $Y_v^{(b)}$.

The b^{th} node now applies voxel-specific masking to $X^{(b)}$ and $Z^{(b)}$ to obtain the spatially-varying designs $\{X_v^{(b)}\}$ and $\{Z_v^{(b)}\}$. For each voxel, v, this is achieved by considering whether v has missing data in the input images listed in $Y^{(b)}$ and "zero-ing" out the corresponding rows of $X^{(b)}$ and $Z^{(b)}$ accordingly (c.f. Section 1.2.2). Given the spatially varying designs, $\{X_v^{(b)}\}$ and $\{Z_v^{(b)}\}$, and response vector, $\{Y_v^{(b)}\}$, the product forms for this partition of the model are now computed as: $P_v^{(b)} = X_v^{(b)\prime} X_v^{(b)}$, $Q_v^{(b)} = X_v^{(b)\prime} Y_v^{(b)}$,... and so forth.

For each voxel, the product forms for each partition are then sent to a designated central node. For the v^{th} voxel, the central node now computes the product forms for the entire model by summing over those sent from each node (i.e. $P_v = \sum_b P_v^{(b)}$, $Q_v = \sum_b Q_v^{(b)}$,...). This approach can be seen to produce the product forms which were defined by Eq. (5) by noting, for arbitrary matrices of appropriate dimension, A and B, with corresponding vertical partitions $\{A^{(b)}\}$ and $\{B^{(b)}\}$, that $A'B = \sum_b A^{(b)\prime} B^{(b)}$. Pseudocode for the product form computation stage of the BLMM pipeline is provided by Algorithm 1.

To prevent convergence failure due to rank deficiency, following product form calculation, any voxels for which $\text{rank}(P_v) < p$ or $\text{rank}(U_v) < q$ are dropped from the analysis. Removal of such voxels is advised during analysis but can be prevented by setting the "safeMode" option to 0. The approach described in this section was initially motivated by a similar method employed for parameter estimation of the Linear Model. Details of this method are provided in Supplementary Material Section S2, alongside a corresponding implementation written in Python. Further notes on the computational efficiency of Algorithm 1 are also provided in Supplementary Material Section S3.

Algorithm 1: Product Form Computation Pseudocode.

```
On the central node
    Partition $X$ and $Z$ vertically into $B$ batches.
    Partition list of $Y$ images into $B$ batches.
end
On each node, node $b$
    Read in the $b^{th}$ batch of $Y$ images as $Y^{(b)}$.
    Read in the $b^{th}$ batches of $X$ and $Z$ as $X^{(b)}$ and $Z^{(b)}$, respectively.
    for all voxels do
        Compute mask $M_v^{(b)}$ by considering the pattern of zeros in $Y_v^{(b)}$.
        Apply masking to $X_v^{(b)}$ and $Z_v^{(b)}$ to obtain $X_v^{(b)} = M_v^{(b)} X^{(b)}$ and
        $Z_v^{(b)} = M_v^{(b)} Z^{(b)}$.
        Compute the product forms for this partition: $P_v^{(b)}, Q_v^{(b)}, R_v^{(b)}, S_v^{(b)}, T_v^{(b)}$ and
        $U_v^{(b)}$.
        Send the product forms to the central node.
    end
end
On the central node
    for all voxels do
        Compute the product forms; $P_v, Q_v, R_v, S_v, T_v$ and $U_v$ by summing over
        results from nodes (e.g. $P_v = \sum_b P_v^{(b)}$).
    end
end
```

2.1.3. Parameter estimation

The most computationally intensive stage of any LMM analysis is the estimation of the unknown model parameters (β, σ^2, D). A common approach to estimating the unknown model parameters of the LMM is to perform Restricted Maximum Likelihood (REML) estimation using Eq. (2). BLMM employs the Full Simplified Fisher Scoring (FSFS) algorithm to perform this task for each voxel in the analysis mask. Proposed for the multi-factor LMM in our previous work, the FSFS algorithm iteratively performs updates to the fixed effects parameter vector, β, and fixed effects variance estimate, σ^2, using the Generalized Least Squares (GLS) estimators, and to $\{\text{vec}(D^k)\}_{k\in\{1,\dots,r\}}$ separately using a Fisher Scoring update step based on the "full" representation of D^k, $\text{vec}(D^k)$. ("Full" refers to the parameterization of $\text{vec}(D^k)$; see Section 2.1.2 of Maullin-Sapey and Nichols (2021) for further detail). Formally, during each iteration, β and σ^2 are updated according to the following GLS update rules:

$$\beta_{s+1} = \left(X'V_s^{-1}X\right)^{-1}X'V_s^{-1}Y, \quad \sigma^2_{s+1} = \frac{e_{s+1}'V_s^{-1}e_{s+1}}{n}, \tag{6}$$

where $V_s = I + ZD_sZ'$, $e_s = Y - X\beta_s$ and the subscript s here, and throughout the remainder of this section only, denotes iteration number. For $k \in \{1, \dots, r\}$, the update rule employed for D^k takes the following form:

$$\text{vec}(D^k_{s+1}) = \text{vec}(D^k_s) + \alpha_s(F^k_s)^{-1}\partial^k_s. \tag{7}$$

Here, α_s is a scalar step size, which is initialized to $\alpha_0 = 1$ and halved each time a decrease in log-likelihood is observed between iterations, F^k_s acts as a Fisher Information matrix given by:

$$F^k_s = \sum_{i,j=1}^{l_k} \left(Z'_{(k,i)} V_s^{-1} Z_{(k,j)} \otimes Z'_{(k,i)} V_s^{-1} Z_{(k,j)} \right),$$

and ∂^k_s is the score vector given by:

$$\partial^k_s = \text{vec}\Bigg(\sum_{j=1}^{l_k} \left(Z'_{(k,j)} V_s^{-1} e_s \right) \left(Z'_{(k,j)} V_s^{-1} e_s \right)' - Z'_{(k,j)} V_s^{-1} Z_{(k,j)} + Z'_{(k,j)} V^{-1} X \left(X' V^{-1} X \right)^{-1} X' V^{-1} Z_{(k,j)} \Bigg).$$

In this approach, to ensure that the estimates of $\{D^k\}_{k\in\{1,\dots r\}}$ are non-negative definite following each evaluation of the above update rules, an eigendecomposition based approach is used to project D^k to the space of non-negative definite matrices. Further detail on the use of the eigendecomposition in this manner can be found, for example, in Demidenko (2013). We note here that F^k_s is technically not a Fisher Information matrix, but rather a simplified version of the true Fisher Information matrix for the "full" representation of θ that provides the exact same updates during optimisation. For further details on this distinction, see our previous work, Maullin-Sapey and Nichols (2021).

It is important to note that, in every equation given above, the right-hand side can be reformulated to be expressed solely in terms of the product forms and the parameter estimates (β, σ^2, D). This is crucial to the BLMM framework as, as is noted in Section 2.1.2, to prevent memory consumption and computation time from scaling with n, only the product forms are retained in memory following product form computation. Detail of how the above equations can be expressed in terms of the product forms is provided in Appendix A. Initial starting points for the FSFS algorithm are detailed in Appendix B.

Successful convergence of the FSFS algorithm is deemed to occur when the difference in log-likelihood observed between successive iterations becomes less than a predefined tolerance (10^{-6} by default) whilst convergence failure is deemed to have occured if the maximum number of iterations (10^4 by default) is exceeded. The predefined tolerance and maximum number of iterations may be changed by the user via the "tol" and "maxnit" options, respectively. We note here that the default value of 10^4 for maxnit is likely over-cautious as the simulations in our previous work (Maullin-Sapey and Nichols, 2021) found that, for a range of well-specified designs, the FSFS algorithm typically converged within $5-30$ iterations.

Pseudocode for the FSFS algorithm is provided by Algorithm 2. In certain instances, further improvements in terms of computational performance can be obtained by utilising structural features of the analysis design which simplify the above expressions. In particular, BLMM has been optimized to give faster performance for models that contain (i) one random effect grouped by one random factor and (ii) multiple random effects grouped by one random factor. Further detail and discussion of the improvements employed for such models can be found in Supplementary Material Section S4.

In the fMRI analysis setting, careful attention must be given to computational efficiency during parameter estimation, as parameters must be estimated for every voxel in the analysis mask. As noted in Section 1.1.1, computation that is performed independently for one voxel at a time can result in prohibitively slow computation speeds. It follows that, in order to execute LMM parameter estimation for fMRI data in a practical time frame, computation must be streamlined to allow for parameter estimation to be performed concurrently across multiple voxels at once.

Algorithm 2: Full Simplified Fisher Scoring Pseudocode.

```
Assign $\beta_0, \sigma^2_0$ and $\{D^k_0\}_{k\in\{1,\dots r\}}$ to aninitial estimate (c.f. Appendix B).
while current $l_R(\theta)$ and previous $l_R(\theta)$ differ by more than a predefined tolerance do
    Update $\beta$ and $\sigma^2$ using (6).
    for $k \in \{1, \dots r\}$ do
        Update vec($D^k$) using (7).
        Project $D^k$ to the space of non-negative definite matrices using an
        eigendecomposition.
    end
    Recompute $l_R(\theta)$ using (2).
    Assign $\alpha = \frac{\alpha}{2}$ if $l_R(\theta)$ has decreased in value.
end
```

On a single node, parameter estimation may be parallelised across voxels by using broadcasted computation, which exploits the repetitive nature of simplistic operations to streamline calculation. A considerable advantage in using the FSFS algorithm is that it relies upon only conceptually simplistic operations (such as matrix multiplication, matrix inversion and the eigendecomposition), for which a wealth of broadcasted support already exists in modern programming languages such as MATLAB and Python. By utilizing this support, the FSFS algorithm may be executed for multiple voxels concurrently in order to achieve quick and efficient computational performance (c.f. Section 3.1.2 for an assessment of BLMM computation time).

If multiple nodes are available, parameter estimation may also be parallelised further by partitioning the analysis mask into "batches" of voxels and having each node perform parameter estimation for an individual batch. This approach is also provided by BLMM and is referred to as "voxel-wise batching". The ability to distribute computation in this manner is advantageous in situations where the analysis design is large and the product forms cannot be read into memory for many voxels at once. However, this additional layer of parallelisation may not be necessary for smaller designs and is therefore offered optionally in the BLMM package (as shown in Fig. 1).

2.1.4. Inference and output

The final stage of the BLMM pipeline is to perform inference on the fixed effects parameters and output the analysis results in NIfTI format. To support null-hypothesis testing for the fixed effects parameter vector, β, BLMM adopts an approach that is similar to that taken by the popular univariate LMM packages lmerTest, MIXED and PROC-MIXED (c.f. Section 1.1.3). In this approach, the REML estimates of the parameter vector, $(\hat{\beta}, \hat{\sigma}^2, \hat{D})$, are used to construct Wald test statistics. To obtain corresponding p-values, a WSDF-based approach is then employed to model the sampling distribution of the Wald statistics.

Assuming the user has provided a contrast vector, L, to specify a null hypothesis $H_0 : L\beta = 0$, BLMM will compute the corresponding Wald T-statistic or F-statistic for each voxel as follows:

$$T = \frac{L\hat{\beta}}{\sqrt{\hat{\sigma}^2 L \left(X'\hat{V}^{-1}X \right)^{-1} L'}},$$

$$F = \frac{\hat{\beta}' L' \left[L \left(X'\hat{V}^{-1}X \right)^{-1} L' \right]^{-1} L\hat{\beta}}{\hat{\sigma}^2 \text{rank}(L)},$$

where $\hat{V} = I_n + Z\hat{D}Z'$. BLMM assumes that the distributions of T and F are reasonably approximated with a student's t- or F-distribution. As the distributional assumptions for T and F are approximate, and not exact, the degrees of freedom are unknown and must be estimated. The estimation is performed using a WSDF-based approach based on the

Welch-Satterthwaite equation;

$$v(\hat{\eta}) = \frac{2(S^2(\hat{\eta}))^2}{\text{Var}(S^2(\hat{\eta}))},$$

where $\hat{\eta}$ represents an estimate of the variance parameters $\eta = (\sigma^2, D^1, \ldots D^r)$ and $S^2(\hat{\eta}) = \hat{\sigma}^2 L(X'\hat{V}^{-1}X)^{-1}L'$. The numerator of the above expression may be evaluated directly. However, the below approximation, obtained using a second order Taylor expansion, must be employed to estimate the unknown variance term in the denominator:

$$\text{Var}(S^2(\hat{\eta})) \approx \left(\frac{dS^2(\hat{\eta})}{d\hat{\eta}}\right)'\text{Var}(\hat{\eta})\left(\frac{dS^2(\hat{\eta})}{d\hat{\eta}}\right).$$

While other tools (lmerTest, MIXED, PROC-MIXED) require numerical optimisation to obtain this denominator, we make use of our previous work, Maullin-Sapey and Nichols (2021), where we presented novel closed-form expressions which may be used to evaluate the variance and derivative terms in the above directly. These expressions not only provide a computationally efficient alternative to the use of numerical optimisation, but also can be expressed purely in terms of the product forms (see Supplementary Material Section S5 for further detail). As a consequence of this, using a similar approach to that of Section 2.1.3, BLMM is able to perform degrees of freedom estimation by utilising only the product forms and parameter estimates at each voxel, and by employing operations that can be broadcasted or further accelerated through batch-wise parallelisation.

Once an analysis has been executed using BLMM, parameter estimate images for β, σ^2 and D, contrast images of the form $L\hat{\beta}$ and Wald statistic images with corresponding $p-$value images are output, which must then be corrected for multiple testings. Correction for multiple testing is essential, though the non-linear estimation process precludes the use of standard random field theory and, while permutation or wild bootstrap methods are available for mixed models, such methods add substantial computational burden. Hence, either control of the Family-Wise Error rate (FWE) with the Bonferroni correction (Dunn, 1961) or the False Discovery Rate (FDR) with the Benjamini-Hochberg method (Benjamini and Hochberg, 1995) should be used to account for multiple testing.

Following the conclusion of a BLMM analysis, BLMM also provides Likelihood Ratio Testing (LRT) for comparison of LMM analyses which contain a single-random factor and differ only by the inclusion of random effects. More precisely, the output from several BLMM (or BLM, c.f. Supplementary Material Section S2) analyses may be used to perform hypothesis testing of the below form:

$$H_0 : D_{\{J\}} = 0 \qquad H_1 : D_{\{J\}} \neq 0$$

where J represents a predetermined set of elements of D which correspond to the removal of $\tilde{q}$ random effects from the study design. The above hypothesis is tested using the LRT statistic of the form $-2\ln(l(\hat{\theta}_0)/l(\hat{\theta}))$, where $\hat{\theta}_0$ and $\hat{\theta}$ are the parameter estimates obtained from BLMM analyses in which the random effects $D_{\{J\}}$ are and are not included in the model specification, respectively. Following the recommendations of Verbeke and Molenberghs (2001), BLMM assumes that the LRT test statistic follows the distribution $\chi^2_{q-\tilde{q},q}$, where $\chi^2_{a,b}$ represents an even mixture of the distributions χ^2_a and χ^2_b. Under this distributional assumption, BLMM may be used to generate uncorrected p-value significance images for the LRT statistic. Examples of this method in practice are provided by Section 3.2.

2.2. Simulation methods

In order to quantitatively assess and demonstrate the computational accuracy and efficiency of BLMM, extensive simulations were conducted. Simulated data was generated for nine simulation settings: three sample sizes ($n = 200, 500$ and 1000, respectively), each generated under three experimental designs. The experimental designs considered for simulation in this work reflect the two settings for which parameter estimation in BLMM has been explicitly optimized (c.f. Section 2.1.3 and Supplementary Material Section S4), as well as the most general model specification that BLMM caters to. These were the settings in which the experimental design contains; (i) one random factor which groups one random effect, (ii) one random factor which groups multiple random effects and (iii) multiple random factors, respectively. These models correspond to common use cases designs employing, for example, (i) a subject-level random intercept in a repeated measures setting, (ii) a subject-level random intercept and slope in a repeated measures setting, and (iii) a subject-level intercept and site-level intercept for repeated measures taken from multiple subjects across multiple sites. For each simulation setting, 1000 individual simulation instances were performed, and all reported results are given as averages taken across the 1000 instances.

In each simulation setting, the fixed effects parameter vector, β, and the fixed effects variance, σ^2, were fixed across simulation instances and given by $\beta = [4, 3, 2, 1, 0]'$ and $\sigma^2 = 1$, respectively. The fixed effects design matrix, X, contained an intercept and four regressors, each of which varied across simulation instances and consisted of values generated according to a uniform $[-0.5, 0.5]$ distribution. Each experimental design enforced a different structure on the random effects design and covariance matrices, Z and D. The first experimental design (Design 1) included a single factor which grouped one random effect into 100 levels (i.e. $r = 1$, $q_1 = 1$ and $l_1 = 100$). The second experimental design (Design 2) included a single factor which grouped two random effects into 50 levels ($r = 1$, $q_1 = 2$, $l_1 = 50$). The third experimental design (Design 3) included two crossed factors, the first of which grouped two random effects into 20 levels and the second of which grouped one random effect into 10 levels (i.e. $r = 2, q_1 = 2, q_2 = 1, l_1 = 20$ and $l_2 = 10$). In all simulation instances, the first random effect appearing in Z was an intercept. Any additional random effects regressors varied across simulation instances and were generated according to a uniform $[-0.5, 0.5]$ distribution. For each factor, observations were assigned to levels uniformly at random so that the probability of an observation belonging to any specific level was the same for all levels. The diagonal and off-diagonal elements of the random effects covariance matrix for each factor were held fixed across simulation instances and given as 1 and 0.5, respectively.

The spatially varying random terms, ϵ and b were generated as images of Gaussian noise, with the appropriate covariance between images induced for b. The response images, Y, were then calculated using X, Z, β, b and ϵ according to Eq. (1). An isotropic Gaussian filter with a Full Width Half Maximum (FWHM) of 5 was then applied to the response images, Y, to induce spatial correlation across voxels. Following this, for each response image, random perturbations were applied to the FSL standard 2mm MNI brain mask in order to generate a corresponding "random mask" image, thus simulating missingness near the edge of the brain (c.f. Supplementary Material Section S6). The "random" masks were applied to the response images to finally obtain a masked smoothed random response, roughly resembling null fMRI data. A visual overview of the data generation process is provided by Fig. 2. All NIfTI volumes generated for simulation were of dimension $(100 \times 100 \times 100)$ voxels. It must be stressed that this process was not designed to simulate realistic fMRI data rigorously, but rather data that had approximately the same shape, size, smoothness, and degree of missingness as might be observed in real fMRI data.

In order to assess the accuracy and performance of the parameter estimation in each simulation instance, the R function lmer was also used to obtain parameter estimates for each voxel in the analysis mask. To measure computation speed, we define the 'Serial Computation Time' (SCT) for BLMM parameter estimation as the time in seconds that would have been spent executing the FSFS algorithm if the computation performed by each node had been run back-to-back in serial. The serial computation time for lmer parameter estimation is similarly defined as the total amount of time that would have been spent executing the function 'optimizeLmer' had all computation been performed in serial. The performance of lmer and BLMM was contrasted in terms of SCT averaged

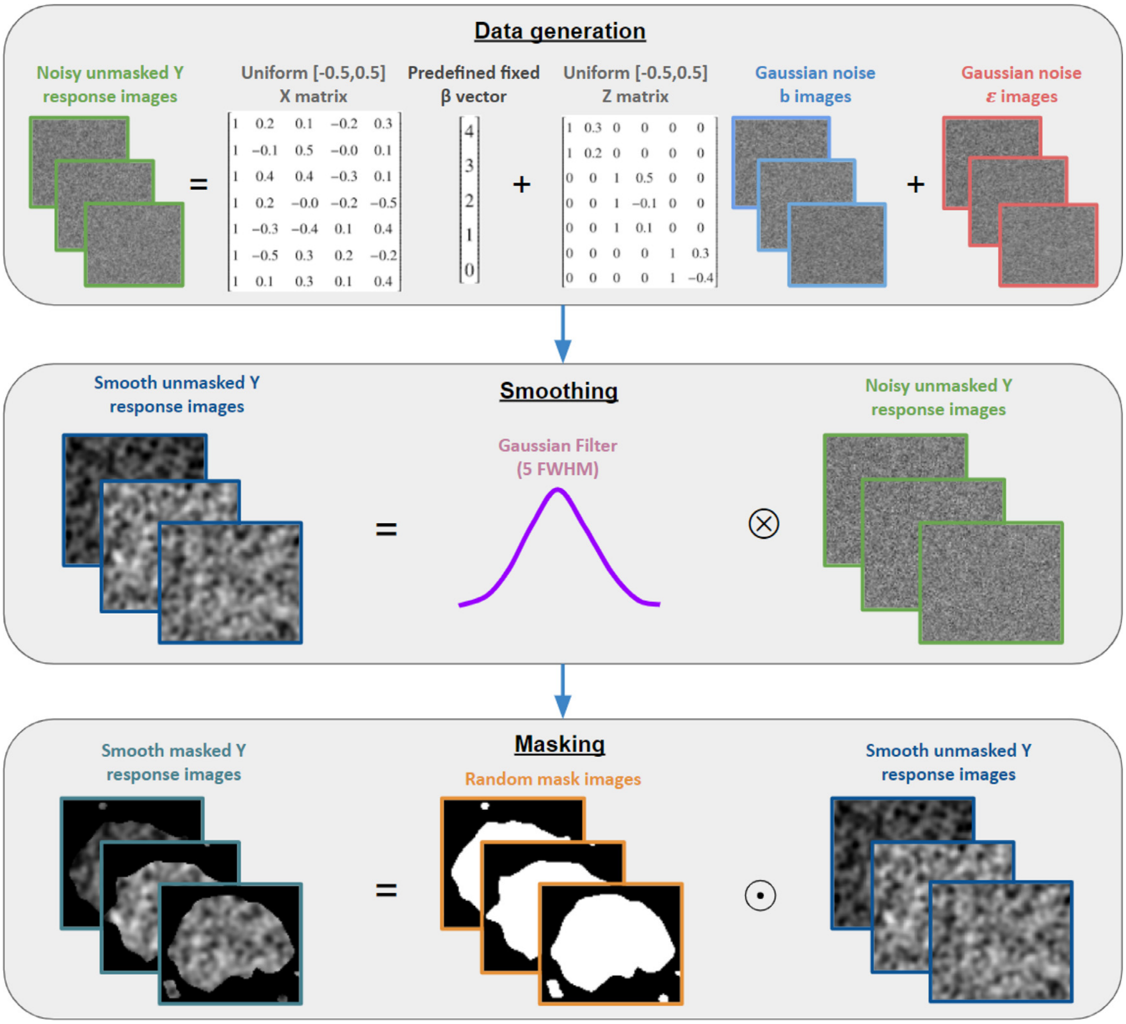

Fig. 2. A visual representation of the pipeline employed to generate the simulated data of Section 3.1. The first box depicts the model which was used for data generation, notably highlighting that ϵ and b varied across space. The second box details the smoothing process, with the ⊗ symbol representing convolution in this instance. The third box details the masking stage, with ⊙ representing the Hadamard (element-wise) product.

across simulation instances, whilst the parameter estimates produced by lmer and BLMM were compared in terms of image-wide mean absolute difference to assess the correctness of the FSFS algorithm employed by BLMM. It is noted here that the computation of product forms is not included in our evaluation of SCT for BLMM. However, we do not believe this has impacted the results of our analysis as preliminary testing demonstrated that the time spent performing product form computation was many orders of magnitude (approximately 10^5) smaller than the time spent performing REML estimation using the FSFS algorithm.

The primary purpose in choosing lmer as a baseline for comparison was to demonstrate the substantial computational benefits of using the BLMM pipeline for mass-univariate analysis in the place of naive computation via ‘for loops’ and lmer. We stress that it is not the author’s intention for the results of this analysis to be interpreted as a reflection on the ability of the lme4 package, which is not designed for use in the mass-univariate setting, but rather the efficiency of the FSFS algorithm when combined with vectorised computation. To ensure the missingness capabilities of BLMM were exhaustively tested, each simulated analysis employed a lenient missingness threshold of 50%.

In sum, the simulations we have described assess BLMM’s (i) correctness in terms of parameter estimation, (ii) ability to handle missing data, and (iii) computation speed for parameter estimation. All reported results were obtained using an HPC cluster with Intel(R) Xeon(R) Gold 6126 2.60GHz processors each with 16GB RAM.

2.3. Real data methods

As a demonstration of the large-scale capabilities of BLMM, here we provide an example involving a much larger model than those considered during the simulations of Section 2.2. In this example, we utilise data from the UK Biobank, in which repeated measures were recorded for 2461 subjects, each of whom completed a “faces vs shapes” task twice across separate visits. Each stimuli image of a face had either an angry or fearful expression, designed to elicit a strong emotional response, whilst the stimuli images of shapes were abstract and neutral. In FSL, a first-level analysis was conducted independently for each visit for each subject. In each first-level analysis, the task design was regressed onto Blood Oxygenation Level Dependent (BOLD) response. During first-level analysis, for each subject and visit, a Contrast Parameter Estimate (COPE) map was generated. Each COPE map represented, for a given visit and subject, the difference in BOLD response between the subject viewing images of faces and images of abstract shapes.

At the group level, BLM and BLMM were used to perform parameter estimation for three models, each designed to estimate the average group level ‘faces>shapes’ response. In each model, the response images, Y, were the COPE maps that were output by FSL during the first-level analysis, registered to MNI space. The fixed effects design matrix, X, in each model included an intercept, the cross-sectional effect of age (age of subject in years, averaged across visits), longitudinal time (age

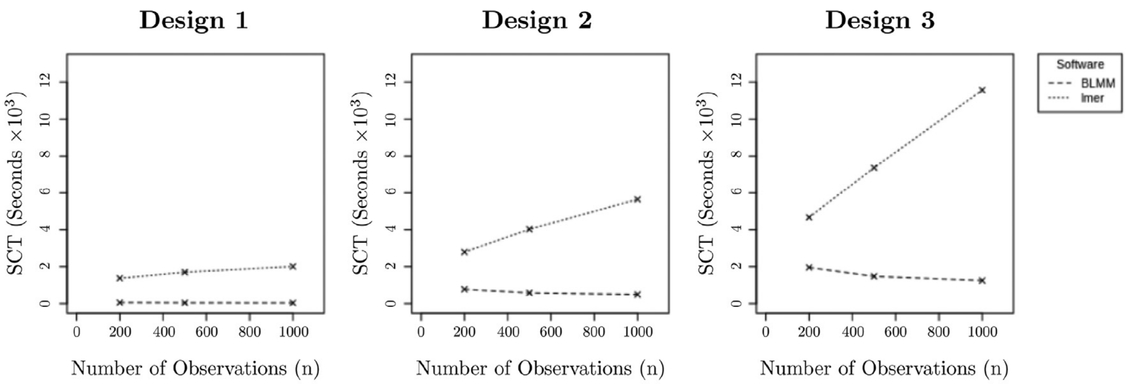

Fig. 3. Observed serial computation times for each experimental design, displayed as a function of the number of observations, n. Displayed are the SCT in kiloseconds for BLMM (dashed) and lmer (dotted).

at each visit, demeaned for each subject individually), sex (encoded as a two-valued factor with -0.5 = Male and 0.5 = Female), a cross-sectional age-sex interaction effect and the Townsend deprivation index (a measure of socio-economic status).

The primary effect of interest in this model was the group-level average BOLD response to the faces vs shapes task (i.e. the intercept). However, we were also interested in the effect of age. In this repeated measures context, simply adding an age covariate imposes a strong assumption that cross-sectional and longitudinal effects of ageing are equal, which is often not the case (Neuhaus and Kalbfleisch, 1998; Guillaume et al., 2014). Instead, two regressors are added, one a pure cross-sectional effect of age (subject age, averaged over visits, for each subject), and one a pure within-subject effect of ageing (age, centred by subject). Such an approach is commonly adopted in the univariate longitudinal modelling literature (c.f. Brant and Verbeke, 1997; Morrell et al., 2009).

The first of the three group-level analyses (Model 1) was a linear regression model estimated using BLM. As it was a standard linear regression, this model included no random-effects design matrix, Z. The second group-level model (Model 2) was analysed using BLMM and included a subject-level random intercept in the random-effects design matrix. The third of the group-level models (Model 3) was run using BLMM and included both a subject-level random intercept and a subject-level random slope for longitudinal time in the random-effects design matrix. For each model, approximate Wald $T-$statistics were computed, using the methods of Section 2.1.4, for the model intercept (the group-level average response to the 'faces>shapes' task), the cross-sectional age and longitudinal time. Once the parameters of each model had been estimated, to assess goodness of fit, model comparison was performed using the LRT method detailed in Section 2.1.4.

It must be noted that, as each model contained two observations per subject and model 3 contained two random effects per subject, model 3 contained the same number of observations as random effects. As a result of this, model 3 may be expected to be unidentifiable for many voxels, as parameter estimation for any voxel with missing data will attempt to model at least two random effects using less than two visits. This choice of model is deliberate as model 3 is an extreme use-case that both stress-tests the BLMM code and serves as a clear example of a model that is expected to be rejected by the LRT procedure. It must be stressed that, by including model 3 in this example, it is not our intention to endorse estimation of models that have been ill-specified in this manner. Model 3 is provided purely for the purposes of demonstration and can only be estimated in BLMM by turning off a 'safeMode' setting during input specification.

The primary purpose of the analyses described above is to demonstrate BLMM's usage in practice and highlight BLMM's efficiency and scalability via a worked example. In order to assess computational efficiency, model 2 was also estimated voxel-wise using lmer and, as in Section 2.2, serial computation times were recorded for parameter estimation for both lmer and BLMM. However, the same comparison could not be performed for model 3, as the default settings in lmer will not allow for models with equal numbers of random effects and observations to be estimated. In Section 3.2, results are reported for both Likelihood Ratio Tests (model 1 vs model 2 and model 2 vs model 3), with p-value significance maps for the approximate Wald T-tests then being provided for the selected model. All reported significance regions were obtained using a 5% Bonferonni corrected threshold. All analysis results were obtained using a SGE cluster with Intel(R) Xeon(R) Gold 61262.60GHz processors, each with 16GB RAM.

3. Results

3.1. Simulation results

3.1.1. Parameter estimation

Across the nine simulation settings outlined in Section 2.2 (three designs across three sample sizes), all parameter estimates and maximised restricted likelihood criteria produced by BLMM were near identical to those produced by lmer. In particular, extremely strong agreement was observed between the parameter estimates produced by BLMM and lmer for both voxels with missing observations and voxels with all observations present, illustrating BLMM's capacity for handling missing data. For each experimental design, the largest mean absolute difference for parameter estimation was observed in the estimation of the random effects covariance parameters (the unique non-zero elements of D) in the $n = 200$ setting. The observed mean absolute difference in the random effects covariance parameter estimates produced by BLMM and lmer for this setting were 6.86×10^{-9}, 4.39×10^{-5} and 6.02×10^{-3} for designs 1, 2 and 3, respectively. For double-precision floats, fractional rounding errors can occur at a magnitude of 2^{-52} by chance. Such errors can further propagate to be of size $2^{-26} \approx 1.49 \times 10^{-8}$ when square roots are involved in computation. If we take $2^{-26} \approx 1.49 \times 10^{-8}$ as the tolerance level for which double floating point representations of the parameter estimates are treated as no longer distinguishable from one another, it can be seen that this means the estimates produced by BLMM and lmer were indistinguishable at the machine precision level for design 1 and very similar for designs 2 and 3. Similar results may also be observed across simulations for the maximised REML criterions produced by BLMM and lmer (see Supplementary Material Sections S7–S10).

3.1.2. Computation time

The observed SCTs for each simulated setting are presented in Fig. 3. As can be seen from Fig. 3, BLMM significantly outperformed computation via 'for loops' and lmer and, notably, appeared to maintain an

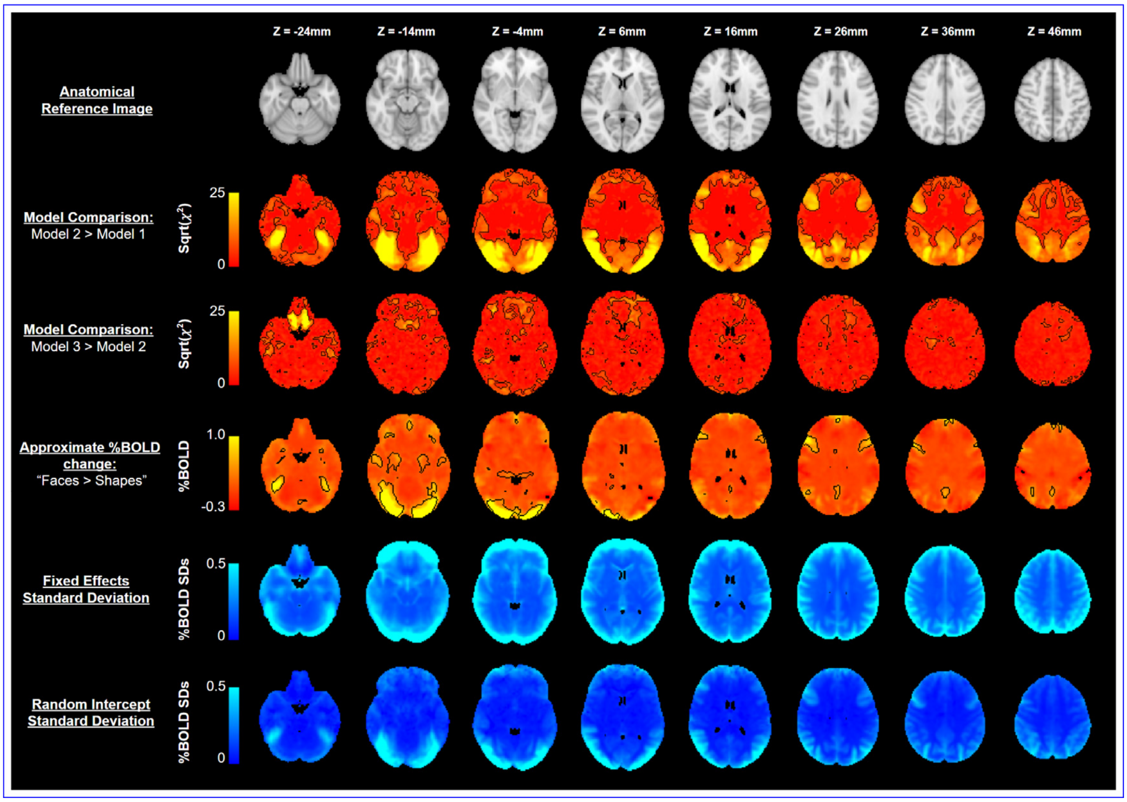
Z = -24mm
Z = -14mm
Z = -4mm
Z = 6mm
Z = 16mm
Z = 26mm
Z = 36mm
Z = 46mm
Anatomical Reference Image
Model Comparison: Model 2 > Model 1
Sqrt(χ^2)
25
0
Model Comparison: Model 3 > Model 2
Sqrt(χ^2)
25
0
Approximate %BOLD change: "Faces > Shapes"
%BOLD
1.0
-0.3
Fixed Effects Standard Deviation
%BOLD SDs
0.5
0
Random Intercept Standard Deviation
%BOLD SDs
0.5
0

Fig. 4. First row: The MNI152 2mm anatomical template, for reference. Second row: χ^2 statistics for comparison of model 1 and model 2, displayed on the square root scale; outlined in black are regions where evidence was found that inclusion of a random subject intercept significantly affected (at the 5% Bonferroni-significance level) the results of the analysis. Third row: χ^2 statistics for comparison of model 2 and model 3, displayed on the square root scale; outlined regions indicate where the inclusion of a random subject slope significantly affected the results of the analysis. Fourth row: Effect estimates for the "Faces > Shapes" contrast, derived from model 2; for this row, voxels demarcated are those Bonferroni-significant at the 5% level. Fifth row: The fixed effects standard deviation (σ), derived from model 2. Sixth row: The standard deviation of the subject-level random intercept ($\sigma\sqrt{d}$ where d is the only non-zero element of D in model 2).

approximately constant computation time as the number of observations, n, increased. In contrast, computation time for lmer appeared to increase as the number of observations increased. These results match expectation as, as noted in Section 2.1.2, the BLMM pipeline begins by computing the product forms and discarding any model matrices which had dimensions scaling with n. This means that during parameter estimation (the most intensive stage of the BLMM pipeline, which dominates the computation time), the computation required for a single iteration of the FSFS algorithm does not scale as a function of n. As the computation time required for each iteration of the FSFS algorithm does scale as a function of q, however, it should be noted that, as shown in Fig. 3, when q and n are close in magnitude the benefits of the product form approach to computation are less pronounced.

We note that, although computation for each individual iteration of the FSFS algorithm is dependent only upon q, and not n, the number of iterations required for the FSFS algorithm to converge may bear a more complex relationship to the number of observations and the structure of the random effects in the analysis. This is as misspecified random effects, complex covariance structures, and insufficient sample sizes can substantially impact the convergence of iterative likelihood estimation procedures. For a further comprehensive discussion of the convergence properties of the FSFS algorithm, we refer the reader to our previous work, Maullin-Sapey and Nichols (2021).

The results of Fig. 3 demonstrate BLMM's strong computational efficiency for analysing the designs which were simulated. However, it should be noted that the difference in SCT between lmer and BLMM may be smaller than those observed in these simulations for analyses involving designs in which the second dimension of the random effects design matrix, q, is very large. The reason for this is that, as q increases, the storage requirements associated with each voxel increase and, as a result, parameter estimation can be performed for fewer voxels concurrently via vectorisation. In the simulations presented in this section, q was set to 100 in designs 1 and 2 and set to 50 in design 3. We note that it would be of benefit to conduct further detailed analysis of BLMM's performance compared to that of lmer for models containing larger values of q and more complex random effects structures. However, at present, such comparative simulations are practically infeasible due to the inordinate computation time they would require (the reported simulations required several months to run, generated over five million simulated brain images and executed approximately 10^7 univariate LMM analyses in lmer). Whilst such extensive simulations may not be possible, we highlight here that the analyses presented in Section 3.2 provide further evidence of BLMM's strong performance for two models in which q is much larger than considered in the simulations presented in this section ($q = 2461$ and $q = 4922$ in models 1 and 2, respectively).

3.2. Real data results

3.2.1. Model comparison

The results for the real data analysis are presented in Fig. 4. The second and third rows of Fig. 4 display the χ^2-statistics (on the square root scale) for model comparison between models 1 and 2, and models 2 and 3, respectively, with 5% Bonferroni-significant voxels demarcated in black. The results of the first LRT, comparing model 2 (the random intercept model) and model 1 (the linear regression), highlight most of occipital, parietal and frontal lobes. This observation suggests that, in these regions, evidence was observed that the inclusion of subject-level random intercepts substantially influenced the outcome of the analysis.

This conclusion is reasonable and reflects regions where there is non-negligible BOLD response to this task (see Section 3.2.2).

The second LRT, comparing model 3 (the random intercept and slope model) to model 2, only identifies two different regions: orbitofrontal areas, which are often subject to signal loss, and white matter areas in the centre of brain, which have the poorest SNR with this multicoil acquisition. The sporadic and noisy appearance of the regions identified here could indicate that these regions were influenced by idiosyncratic temporal changes due to variation in head placement or simply by large random temporal changes. In either case, it can be seen that the inclusion of a random slope in the analysis had little impact on anatomical regions of practical relevance to the 'faces vs shapes' task. The conclusion of the LRTs is, therefore, that model 2 should be chosen as the suitable model upon which inference can be performed using approximate Wald T-tests.

3.2.2. *Analysis results*

Of the three contrasts that were estimated for model 2 using the approximate Wald T-test, only the first contrast (the main effect of the "Faces vs Shapes" task) reported significant regions following a 5% Bonferroni-corrected threshold. The effect estimates for this contrast ($L\beta$ values) are displayed on the fourth row of Fig. 4 with Bonferroni significant voxels demarcated in black. In this instance, the occipital lobe, known for its role in processing perceptual information and the amygdala, known to be involved in emotional response (c.f. Rehman and Khalili, 2019; Zald, 2003), have been identified as significant. This pattern of activation is similar to that found in Hariri et al. (2002), Manuck et al. (2007) and Barch et al. (2013) and is to be expected given the visual nature of the task and the emotional response the facial expressions were designed to elicit (c.f. Ekman and Friesen, 1976). The second and third contrasts, which assessed the impact of cross-sectional and longitudinal age on BOLD response, respectively, both displayed no significant regions of activation. These results may be interpreted as stating that no evidence was found to suggest that either age covariate had an impact on the BOLD response to the "faces vs shapes" task. For reference, the BOLD response images for the second and third contrasts have been included in Supplementary Material Section S11.

Also provided in the fifth and sixth rows of Fig. 4 are the estimated residual standard deviation (i.e. σ) and the subject-level random intercept standard deviation ($\sigma\sqrt{d}$, where d is the unique non-zero element in D) for model 2. The residual standard deviation demonstrates substantial measurement error at the edge of the brain, which is likely due to imperfections in registration and preprocessing. By contrast, the subject-level deviation bears a strong resemblance to the BOLD activation seen in the fourth row. This latter image serves as a useful diagnostic, as it highlights regions at which inclusion of the random intercept notably contributed to the analysis results. This illustrates the vital role that the inclusion of a random intercept played in obtaining the results of this analysis and reinforces the conclusion of the LRT testing procedure of Section 3.2.1.

3.2.3. *Computation time*

In total, computation time in BLMM for this analysis took approximately 55 min for model 2 and 4 h for model 3, using 500 nodes. The extreme computation time observed for model 3 can be attributed to the identifiability problems noted in Section 2.3, with parameter estimation for approximately 150 voxels exceeding the default maximum iteration limit. By way of comparison, for model 2, the SCT observed for BLMM was approximately 77.6% of that observed for lmer. As noted previously, this difference is less pronounced than those observed in the simulations of Section 3.1, as the second dimension of the random effects design matrix is extremely large ($q = 2461$) and close in magnitude to the value of n ($n = 4922$). The linear regression model, model 1, was run using BLM and took approximately 5 min using 60 computational nodes. The full analysis results for all three models are publicly available and may be accessed on NeuroVault (see the data and code availability declaration).

4. Discussion and conclusion

In this work, we have detailed and presented BLMM, a freely available software package for performing LMM parameter estimation and inference on large-n fMRI datasets. LMM computation for fMRI is an extremely computationally intensive task and, as a result, the work presented in this paper is both informed and limited by the currently available support in terms of software and technology. For this reason, BLMM is a continually evolving project, and it is expected that much of the methodology currently implemented in BLMM may gradually change over time.

A large driving force motivating the approaches taken in this work is the current lack of available support for broadcasted sparse matrix operations. Whilst the available support for sparse matrix methodology in programming languages such as MatLab and Python has substantially improved in recent years, currently, there is little support for performing sparse matrix operations concurrently many times. Unfortunately, this substantially limits the utility of sparsity-based approaches to LMM estimation in the context of fMRI analysis. This is due to the substantial overheads which would be accrued if LMM estimation were performed independently for each voxel in an image. An undesirable ramification of this is that the BLMM code, which has been explicitly optimised to account for the patterns of sparsity present in single-factor models (see Supplementary Material Section S4), may not provide comparable performance for multi-factor models in which q is very large. However, as programming languages evolve and new support becomes available, this is expected to change, and we predict that future development of BLMM is highly likely to incorporate sparse matrix methodology into its approach to analysing multi-factor models. For this reason, we suggest that the incorporation of sparse matrix methodology into BLMM may form a substantial basis for future development.

Another potential avenue for future development focuses on how BLMM currently handles file input and output on HPC clusters. In recent years, in Python especially, there has been a strong movement towards standardising and streamlining cluster-based computation. A project of particular note is the Dask Python package, which acts as a standardised specification to encode parallel algorithms and file I/O (Rocklin, 2015). As noted in Section 1.1.1, Dask is heavily employed by the AFNI package 3dLME and provides substantial benefits both in terms of computation time and ease of distribution. Although many of the broadcasted operations employed by BLMM are yet to be supported by Dask, we suggest that incorporating the approach of AFNI's 3dLME by integrating Dask with the existing BLMM code-base may provide further speed improvements and could form the foundation of future development within BLMM.

There is also much room for further theoretical development of BLMM. In our previous work, Maullin-Sapey and Nichols (2021), we presented a general approach for performing constrained optimisation to enforce structure in the random effects covariance matrix. This concept may be of utility in a wide range of commonplace applications, as structured covariance matrices are a feature of many popular statistical models. Such applications include, for example, modelling genetic components in twin studies and auto-correlation in time-series analyses. Whilst the approaches outlined in our previous work are sufficiently general to perform such analyses in the univariate setting, it is not immediately apparent whether they may be feasibly executed on a mass-univariate scale. For this reason, we suggest that the potential for enforcing covariance structure in BLMM merits further investigation.

Another potential direction for future work stems from the observation that the BLMM framework could be used to support remote distributed computation when raw data cannot be centrally stored. Such prohibitive situations are common when, for example, the raw data is extremely large, or subject to privacy constraints preventing it from being shared. The BLMM framework may be of utility in such situations as each remote site may compute their product forms using only their data and their portion of the design matrix, apply some form of differential privacy protection to the results (e.g. add calculated amounts of noise to the product forms to preserve privacy), and send the results to a central coordinate that never sees the raw data. Similar approaches to distributed computation have been notably adopted by the Collaborative Informatics and Neuroimaging Suite Toolkit for Anonymous Computation (COINSTAC, c.f. Plis et al., 2016) and suggest new alternative applications for the BLMM framework.

Declarations

Data/code availability statement

We have used data from The UK Biobank. The BLMM toolbox, as well as the code used for the simulations and timing comparisons described in Sections 2.2 and 2.3, are available at:

https://github.com/TomMaullin/BLMM.

The BLM toolbox, which is also referenced several times throughout this work, may be found at:

https://github.com/TomMaullin/BLM.

The images generated by BLMM for the three models discussed in Sections 2.3 and 3.2 may be found at:

Model 1: https://identifiers.org/neurovault.collection:13110.
Model 2: https://identifiers.org/neurovault.collection:13111.
Model 3: https://identifiers.org/neurovault.collection:13112.

The results of the LRT's for model comparison discussed in Sections 2.3 and 3.2 may also be found at:

https://identifiers.org/neurovault.collection:10451.

Ethics statement

The data we have used was provided by the UK Biobank. The UK Biobank has generic Research Tissue Bank (RTB) approval, which covers our research using the resource.

Disclosure of competing interests

Not applicable.

Role of the funding source

This work was supported by the Li Ka Shing Centre for Health Information and Discovery and NIH grant [R01EB026859] (TMS, TN) and the Wellcome Trust award [100309/Z/12/Z] (TN).

Credit authorship contribution statement

Thomas Maullin-Sapey: Conceptualization, Formal analysis, Investigation, Methodology, Software, Visualization, Writing – original draft, Writing – review & editing. **Thomas E. Nichols:** Conceptualization, Funding acquisition, Supervision, Writing – review & editing.

Data availability

The authors do not have permission to share data.

Appendix A. The Product Form Approach to Parameter Estimation

In this appendix, we provide further detail on how the product forms described in Section 2.1.2 are employed to perform the parameter estimation procedure of Section 2.1.3. The GLS update rules for β and σ^2, Eq. (6), can be written in terms of the product forms and parameter estimates as follows:

$$\beta_{s+1} = (P - RD_s(I+D_sU)^{-1}R')^{-1}(P - RD_s(I+D_sU)^{-1}R'),$$
$$\sigma^2_{s+1} = \frac{1}{n}(S - 2Q'\beta_s + \beta_s'P\beta_s),$$

and the expressions for F_s^k and ∂_s^k can be seen to be composed entirely of sub-matrices of $X'V_s^{-1}X$, $Z'V_s^{-1}Z$ and $Z'V_s^{-1}e_s$, which are given by:

$$\begin{aligned} X'V_s^{-1}X &= P - RD_s(I+D_sU)^{-1}R', \\ Z'V_s^{-1}Z &= U - UD_s(I_q+D_sU)^{-1}U, \\ Z'V_s^{-1}e_s &= T' - R'\beta_s - UD_s(I_q+D_sU)^{-1}(T'-R'\beta_s). \end{aligned} \tag{A.1}$$

The restricted log-likelihood function, given by Eq. (2), can similarly be rewritten in terms of product forms as follows:

$$\begin{aligned} l_R(\theta_s) = -\frac{1}{2}\Big\{ &(n-p)\log\left(\sigma_s^2\right) + \log|I + D_sU| + \log|P - RD_s\left(I+UD_s\right)^{-1}R'| \\ &+ \sigma_s^{-2}\Big(S - 2Q'\beta_s + \beta_s'P\beta_s + \left(T' - R'\beta_s\right)'D_s\left(I+UD_s\right)^{-1}\left(T - R'\beta_s\right)\Big)\Big\}. \end{aligned}$$

Appendix B. Initial Values for Parameter Estimation

In this appendix, we provide expressions for the initial values used by BLMM during the FSFS algorithm. To choose initial values for the optimization procedure, BLMM follows the recommendations of Demidenko (2013) and Maullin-Sapey and Nichols (2021), employing the OLS estimators as starting estimates for β and σ^2;

$$\beta_0 = \left(X'X\right)^{-1}X'Y, \qquad \sigma_0^2 = \frac{e_0'e_0}{n}, \tag{B.1}$$

and the FSFS update rule, (7), with I_n substituted in the place of V, for a starting estimate of vec(D^k):

$$\text{vec}\left(D_0^k\right) = \left(\sum_{j=1}^{l_k} Z'_{(k,j)}Z_{(k,j)} \otimes Z'_{(k,j)}Z_{(k,j)}\right)^{-1} \text{vec}\left(\sum_{j=1}^{l_k} Z'_{(k,j)}\left(\frac{e_0'e_0}{n} - I_n\right)Z_{(k,j)}\right). \tag{B.2}$$

Again, the above expressions may be evaluated using only the product forms and do not require use of the original matrices X, Y and Z.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.neuroimage.2022.119729.

References

Allen, N., Sudlow, C., Downey, P., Peakman, T., Danesh, J., Elliott, P., Gallacher, J., Green, J., Matthews, P., Pell, J., Sprosen, T., Collins, R., 2012. Uk biobank: current status and what it means for epidemiology. Health Policy Technol. 1 (3), 123–126. doi:10.1016/j.hlpt.2012.07.003. ISSN 2211-8837

Baayen, R.H., Davidson, D.J., Bates, D.M., 2008. Mixed-effects modeling with crossed random effects for subjects and items. J. Mem. Lang. 59 (4), 390–412. doi:10.1016/j.jml.2007.12.005. ISSN 0749-596X, URL http://www.sciencedirect.com/science/article/pii/S0749596X07001398. Special Issue: Emerging Data Analysis

Barch, D.M., Burgess, G.C., Harms, M.P., Petersen, S.E., Schlaggar, B.L., Corbetta, M., Glasser, M.F., Curtiss, S., Dixit, S., Feldt, C., Nolan, D., Bryant, E., Hartley, T., Footer, O., Bjork, J.M., Poldrack, R., Smith, S., Johansen-Berg, H., Snyder, A.Z., Essen, D.C.V., 2013. Function in the human connectome: task-fmri and individual differences in behavior. NeuroImage 2013-oct vol. 80 80. doi:10.1016/j.neuroimage.2013.05.033.

Bates, D., 2006. lmer, p-values and all that. https://stat.ethz.ch/pipermail/r-help/2006-May/094765.html, Accessed: 2020-12-07.

Bates, D., Mchler, M., Bolker, B., Walker, S., 2015. Fitting linear mixed-effects models using lme4. J. Stat. Softw. 67 (1), 1–48. doi:10.18637/jss.v067.i01. ISSN 1548-7660

Bates, D.M., DebRoy, S., 2004. Linear mixed models and penalized least squares. J. Multivar. Anal. 91. doi:10.1016/j.jmva.2004.04.013.

Bearden, C.E., Thompson, P.M., 2017. Emerging global initiatives in neurogenetics: the enhancing neuroimaging genetics through meta-analysis (enigma) consortium. Neuron 94 (2), 232–236. doi:10.1016/j.neuron.2017.03.033. ISSN 0896-6273. URL http://www.sciencedirect.com/science/article/pii/S0896627317302453

Beckmann, C., Jenkinson, M., Smith, S., 2003. General multilevel linear modeling for group analysis in FMRI. Neuroimage 20 (10), 1052–1063. doi:10.1016/S1053-8119(03)00435-X.

Benjamini, Y., Hochberg, Y., 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. J. R. Stat. Soc. Ser. B (Methodological) 57 (1), 289–300. ISSN 00359246. URL http://www.jstor.org/stable/2346101

Bernal-Rusiel, J.L., Greve, D.N., Reuter, M., Fischl, B., Sabuncu, M.R., 2013. Statistical analysis of longitudinal neuroimage data with linear mixed effects models. Neuroimage 66, 249–260. ISSN 1053-8119. URL http://www.sciencedirect.com/science/article/pii/S1053811912010683

Bernal-Rusiel, J.L., Reuter, M., Greve, D.N., Fischl, B., Sabuncu, M.R., 2013. Spatiotemporal linear mixed effects modeling for the mass-univariate analysis of longitudinal neuroimage data. Neuroimage 81, 358–370. ISSN 1053-8119. URL http://www.sciencedirect.com/science/article/pii/S1053811913005430

Brant, L., Verbeke, G., 1997. Describing the natural heterogeneity of aging using multilevel regression models. Int. J. Sports Med. 18. doi:10.1055/s-2007-972719.

Casey, B.J., Cannonier, T., Conley, M.I., Cohen, A.O., Barch, D.M., Heitzeg, M.M., Soules, M.E., Teslovich, T., Dellarco, D.V., Garavan, H., Orr, C.A., Wager, T.D., Banich, M.T., Speer, N.K., Sutherland, M.T., Riedel, M.C., Dick, A.S., Bjork, J.M., Thomas, K.M., Chaarani, B., Mejia, M.H., Hagler, D.J., Daniela Cornejo, M., Sicat, C.S., Harms, M.P., Dosenbach, N.U.F., Rosenberg, M., Earl, E., Bartsch, H., Watts, R., Polimeni, J.R., Kuperman, J.M., Fair, D.A., Dale, A.M., 2018. The adolescent brain cognitive development (abcd) study: imaging acquisition across 21 sites. Dev. Cogn. Neurosci. 32, 43–54. doi:10.1016/j.dcn.2018.03.001. ISSN 1878-9293. URL http://www.sciencedirect.com/science/article/pii/S1878929317301214. The Adolescent Brain Cognitive Development (ABCD) Consortium: Rationale, Aims, and Assessment Strategy

Chen, G., Saad, Z., Britton, J., Pine, D., Cox, R., 2013. Linear mixed-effects modeling approach to fmri group analysis. Neuroimage 73. doi:10.1016/j.neuroimage.2013.01.047.

Corp, I., 2015. IBM SPSS Advanced statistics 23. Armonk, NY: IBM Corp..

Demidenko, E., 2013. Mixed models: theory and applications with R. Wiley Series in Probability and Statistics. Wiley. ISBN 9781118592991. URL https://books.google.co.uk/books?id=uSmRAAAAQBAJ

Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum likelihood from incomplete data via the em algorithm. J. R. Stat. Soc. Ser. B (Methodological) 39 (1), 1–38. ISSN 00359246. URL http://www.jstor.org/stable/2984875

Dunn, O.J., 1961. Multiple comparisons among means. J. Am. Stat. Assoc. 56 (293), 52–64. URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1961.10482090

Ekman, P., Friesen, W.V., 1976. Pictures of Facial Affect. Consulting Psychologists Press. URL https://books.google.co.uk/books?id=gbfMSgAACAAJ

Essen, D.V., Smith, S., Barch, D., Behrens, T., Yacoub, E., Ugurbil, K., 2013. The wu-minn human connectome project: an overview. Neuroimage 80. doi:10.1016/j.neuroimage.2013.05.041.

Fai, A.H.-T., Cornelius, P.L., 1996. Approximate f-tests of multiple degree of freedom hypotheses in generalized least squares analyses of unbalanced split-plot experiments. J. Stat. Comput. Simul. 54 (4), 363–378. doi:10.1080/00949659608811740.

Francq, B.G., Lin, D., Hoyer, W., 2019. Confidence, prediction, and tolerance in linear mixed models. Stat. Med. 38 (30), 5603–5622. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.8386

Friston, K., Stephan, K.E., Lund, T., Morcom, A., Kiebel, S., 2005. Mixed-effects and fmri studies. Neuroimage 24, 244–252. doi:10.1016/j.neuroimage.2004.08.055.

Friston, K.J., Glaser, D.E., Henson, R.N.A., Kiebel, S., Phillips, C., Ashburner, J., 2002. Classical and bayesian inference in neuroimaging: applications. Neuroimage 16 (2), 484–512. doi:10.1006/nimg.2002.1091. ISSN 1053-8119. URL http://www.sciencedirect.com/science/article/pii/S1053811902910918

Gebregziabher, M., Eckert, M., Vaden, K., Johnson, T., Lawson, A., 2017. Methods for the analysis of missing data in fmri studies. J. Biom. Biostat. 08. doi:10.4172/2155-6180.1000335.

Group, F.M., 2020. SPM12 manual. UCL Queen Square Institute of Neurology, 12 Queen Square, London.

Guillaume, B., Hua, X., Thompson, P.M., Waldorp, L., Nichols, T.E., 2014. Fast and accurate modelling of longitudinal and repeated measures neuroimaging data. NeuroImage 2014-jul vol. 94 94. doi:10.1016/j.neuroimage.2014.03.029.

Hariri, A.R., Tessitore, A., Mattay, V.S., Fera, F., Weinberger, D.R., 2002. The amygdala response to emotional stimuli: acomparison of faces and scenes. NeuroImage 2002-sep vol. 17 iss. 1 17. doi:10.1006/nimg.2002.1179.

Haworth, S., Mitchell, R., Corbin, L., Wade, K., Dudding, T., Budu-Aggrey, A., Carslake, D., Hemani, G., Paternoster, L., Smith, G., Davies, N., Lawson, D., Timpson, N., 2019. Apparent latent structure within the uk biobank sample has implications for epidemiological analysis. Nat. Commun. 10, 333. doi:10.1038/s41467-018-08219-1.

Jennrich, R.I., Schluchter, M.D., 1986. Unbalanced repeated-measures models with structured covariance matrices. Biometrics 42 (4), 805–820. ISSN 0006341X, 15410420. URL http://www.jstor.org/stable/2530695

Keselman, H.J., Algina, J., Kowalchuk, R.K., Wolfinger, R.D., 1999. The analysis of repeated measurements: a comparison of mixed-model satterthwaite f tests and a nonpooled adjusted degrees of freedom multivariate test. Commun. Stat. Theory Methods 28 (12), 2967–2999. doi:10.1080/03610929908832460.

Kuznetsova, A., Brockhoff, P., Christensen, R., 2017. Lmertest package: tests in linear mixed effects models. J. Stat. Softw. 82 (13), 1–26. doi:10.18637/jss.v082.i13. ISSN 1548-7660

Laird, N., Lange, N., Stram, D., 1987. Maximum likelihood computations with repeated measures: application of the em algorithm. J. Am. Stat. Assoc. 82 (397), 97–105. doi:10.1080/01621459.1987.10478395.

Laird, N.M., Ware, J.H., 1982. Random-effects models for longitudinal data. Biometrics 38 (4), 963–974. ISSN 0006341X, 15410420. URL http://www.jstor.org/stable/2529876

Li, X., Guo, N., Li, Q., 2019. Functional neuroimaging in the new era of big data. Genom. Proteom. Bioinform. 17 (4), 393–401. doi:10.1016/j.gpb.2018.11.005. ISSN 1672-0229. URL http://www.sciencedirect.com/science/article/pii/S1672022919301603. Big Data in Brain Science

Luke, S., 2017. Evaluating significance in linear mixed-effects models in R. Behav. Res. Methods 49, 1494–1502.

Madhyastha, T., Peverill, M., Koh, N., McCabe, C., Flournoy, J., Mills, K., King, K., Pfeifer, J., McLaughlin, K.A., 2018. Current methods and limitations for longitudinal fmri analysis across development. Dev. Cogn. Neurosci. 33, 118–128. doi:10.1016/j.dcn.2017.11.006. ISSN 1878-9293. URL http://www.sciencedirect.com/science/article/pii/S1878929317300713. Methodological Challenges in Developmental Neuroimaging: Contemporary Approaches and Solutions

Manor, O., Zucker, D.M., 2004. Small sample inference for the fixed effects in the mixed linear model. Comput. Stat. Data Anal. 46 (4), 801–817. ISSN 0167-9473. URL http://www.sciencedirect.com/science/article/pii/S016794730300238X

Manuck, S.B., Brown, S.M., Forbes, E.E., Hariri, A.R., 2007. Temporal stability of individual differences in amygdala reactivity. Am. J. Psychiatry 164 (10), 1613–1614. doi:10.1176/appi.ajp.2007.07040609. PMID: 17898358

Maullin-Sapey, T., Nichols, T.E., 2021. Fisher scoring for crossed factor linear mixed models. Stat. Comput. 31 (5), 53. ISSN 1573-1375. URL https://doi.org/10.1007/s11222-021-10026-6

Morrell, C.H., Brant, L.J., Ferrucci, L., 2009. Model choice can obscure results in longitudinal studies. J. Gerontol. Ser. A Biolog. Sci. Med. Sci. 64A. doi:10.1093/gerona/gln024.

Neuhaus, J.M., Kalbfleisch, J.D., 1998. Between- and within-cluster covariate effects in the analysis of clustered data. Biometrics 54 (2). doi:10.2307/3109770.

Pinheiro, J.C., Bates, D., 2009. Mixed-effects Models in S and S-PLUS. Statistics and Computing. Springer. ISBN 9781441903174

Plis, S.M., Sarwate, A.D., Wood, D., Dieringer, C., Landis, D., Reed, C., Panta, S.R., Turner, J.A., Shoemaker, J.M., Carter, K.W., Thompson, P., Hutchison, K., Calhoun, V.D., 2016. Coinstac: a privacy enabled model and prototype for leveraging and processing decentralized brain imaging data. Front. Neurosci. 10. doi:10.3389/fnins.2016.00365. ISSN 1662-453X

Raudenbush, A.S., Bryk, S.W., 2002. Hierarchical linear models: applications and data analysis methods. Advanced Quantitative Techniques in the Social Sciences 1, 2nd Sage Publications. ISBN 9780761919049,076191904X

Raudenbush, S.W., Bryk, A.S., Cheong, Y.F., Du Toit, M., 2019. Hlm 8: Hierarchical linear and nonlinear modeling. Scientific Software International, Inc.

Rehman, A., Khalili, Y., 2019. Neuroanatomy, occipital lobe.

Rocklin, M., 2015. Dask: parallel computation with blocked algorithms and Task Scheduling. In: Huff, K., Bergstra, J. (Eds.), Proceedings of the 14th Python in Science Conference, pp. 130–136.

SAS Institute, I., 2015. SAS/STAT®14.1 User's Guide The MIXED Procedure. Springer Berlin Heidelberg, Cary, NC: SAS Institute Inc..

Satterthwaite, F.E., 1946. An approximate distribution of estimates of variance components. Biom. Bull. 2 (6), 110–114. ISSN 00994987. URL http://www.jstor.org/stable/3002019

Schaalje, G.B., McBride, J.B., Fellingham, G.W., 2002. Adequacy of approximations to distributions of test statistics in complex mixed linear models. J. Agric. Biol. Environ. Stat. 7 (4), 512–524. ISSN 10857117. URL http://www.jstor.org/stable/1400374

Smith, S.M., Nichols, T.E., 2018. Statistical challenges in "big data" human neuroimaging. Neuron 97 (2), 263–268. doi:10.1016/j.neuron.2017.12.018. ISSN 0896-6273

Stram, D. O., Lee, J. W., 1995. Variance component testing in the longitudinal mixed effects model (vol 50, pg 1171, 1994).

Vaden, K.I., Gebregziabher, M., Kuchinsky, S., Eckert, M., 2012. Multiple imputation of missing fmri data in whole brain analysis. Neuroimage 60, 1843–1855.

Verbeke, G., Molenberghs, G., 2001. Linear Mixed Models for Longitudinal Data. Springer Series in Statistics. Springer New York. ISBN 9780387950273

Welch, B.L., 1947. The generalization of 'student's' problem when several different population variances are involved. Biometrika 34 (1/2), 28–35. ISSN 00063444. URL http://www.jstor.org/stable/2332510

West, B.T., Welch, K.B., Galecki, A.T., 2014. Linear Mixed Models: A Practical Guide Using Statistical Software. CRC Press. ISBN 9781420010435

Wolfinger, R., Tobias, R., Sall, J., 1994. Computing gaussian likelihoods and their derivatives for general linear mixed models. Siam J. Sci. Comput. 15. doi:10.1137/0915079.

Woolrich, M.W., Behrens, T.E.J., Beckmann, C.F., Jenkinson, M., Smith, S.M., 2004. Multilevel linear modelling for fmri group analysis using bayesian inference. Neuroimage 21 (4), 1732–1747. doi:10.1016/j.neuroimage.2003.12.023. ISSN 1053-8119. URL http://www.sciencedirect.com/science/article/pii/S1053811903007894

Zald, D.H., 2003. The human amygdala and the emotional evaluation of sensory stimuli. Brain Res. Rev. 41 (1), 88–123. doi:10.1016/S0165-0173(02)00248-5. ISSN 0165-0173. URL https://www.sciencedirect.com/science/article/pii/S0165017302002485