Article

SMMR
STATISTICAL METHODS IN MEDICAL RESEARCH

Statistical Methods in Medical Research
0(0) 1–19

DOI: 10.1177/0962280220946080
journals.sagepub.com/home/smm
SAGE

Random forests for high-dimensional longitudinal data

Louis Capitaine, Robin Genuer and Rodolphe Thiébaut

Abstract

Random forests are one of the state-of-the-art supervised machine learning methods and achieve good performance in high-dimensional settings where p, the number of predictors, is much larger than n, the number of observations. Repeated measurements provide, in general, additional information, hence they are worth accounted especially when analyzing high-dimensional data. Tree-based methods have already been adapted to clustered and longitudinal data by using a semi-parametric mixed effects model, in which the non-parametric part is estimated using regression trees or random forests. We propose a general approach of random forests for high-dimensional longitudinal data. It includes a flexible stochastic model which allows the covariance structure to vary over time. Furthermore, we introduce a new method which takes intra-individual covariance into consideration to build random forests. Through simulation experiments, we then study the behavior of different estimation methods, especially in the context of high-dimensional data. Finally, the proposed method has been applied to an HIV vaccine trial including 17 HIV-infected patients with 10 repeated measurements of 20,000 gene transcripts and blood concentration of human immunodeficiency virus RNA. The approach selected 21 gene transcripts for which the association with HIV viral load was fully relevant and consistent with results observed during primary infection.

Keywords

Stochastic mixed effects model, tree-based methods, high-dimensional data, repeated measurements

1 Introduction

Random forests (RFs henceforth), introduced by Breiman,[1] are one of the state-of-the-art machine learning methods.[2] In several domains, RFs achieve good prediction performance for high-dimensional data, where the number of predictors p is much larger than the number of observations n (e.g. Cutler et al.[3] and Chen and Ishwaran[4]). On the other hand, theoretical results have also been recently obtained for RF. Scornet et al.[5] proved a consistency result for RF in the context of additive regression models. Mentch and Hooker[6] and Wager[7] studied asymptotic normality of RF predictions and proposed confidence intervals for those predictions. We refer to Biau and Scornet[8] for further reading on that matter.

When the number of predictors p is much larger than the number of observations n, i.e. $p >> n$, application of RF must be done with care since RF parameters, specifically the number of variables randomly picked at each node of a tree, must be carefully tuned to optimize their prediction performance.[9] Some recent improvements have been suggested to deal especially with high-dimensional data. Zhu et al.[10] introduced ideas from reinforcement learning within the tree-based model framework, in order to focus more efficiently on relevant variables during the tree building. Linero[11] developed Bayesian regression trees[12] with sparsity, by using appropriate priors.

INSERM U1219 Bordeaux Population Health Research Center, INRIA Bordeaux Sud-Ouest, SISTM Team, Bordeaux University, Bordeaux, France

Corresponding author:
Robin Genuer, INSERM U1219 Bordeaux Population Health Research Center, INRIA Bordeaux Sud-Ouest, SISTM Team, Bordeaux University, Bordeaux, France.
Email: robin.genuer@u-bordeaux.fr

The case, where repeated measurements are available, we focus on, is quite specific. Indeed, with longitudinal data, within unit variations bring information in addition to the between units variations. Even though an outcome does not change over time, getting repeated observations increases information about the link between predictors and the outcome. Compared to survival (i.e. censored) data, for which there is a large bulk of work (see Hothorn et al.,[13] Ishwaran et al.,[14,15] and Steingrimsson et al.,[16] among others), less has been done to adapt RF approaches to repeated measurements or clustered data. The analysis of longitudinal data requires to take into account the specific correlation structure, as in mixed effects models.[17,18] Concerning tree-based methods, some approaches proposed to adapt the splitting nodes criterion to longitudinal data: Segal[19] adapted a multivariate approach to split nodes, Eo and Cho[20] used polynomial mixed effects models inside each node, while more recently, in the framework of clinical trials, Wei et al.[21] combine mixed effects models with regression splines and use a likelihood ratio test at each node. On the other hand, Hajjem et al.[22] and Sela and Simonoff[23] independently introduced tree-based methods using a semi-parametric mixed effects model, in which the non-parametric part is estimated using regression trees. Fu and Simonoff[24] studied an alternative method which uses conditional inference trees[25] instead of Classification and Regression Trees (CART)[26] while Hajjem et al.[27] have extended their methodology with the use of RF instead of regression trees. Their common estimation procedure is based on an Expectation Maximization (EM) algorithm,[28] which iterates between estimation of the fixed part (with a regression tree or a RF) and estimation of the random part parameters. The work of Hajjem et al.[22,27] focused on clustered data only. The correlation structures considered were much simpler than what is requested for longitudinal data. Recently, Kundu and Harezlak[29] also proposed to use mixed effects model on different clusters corresponding to the leaves of a regression tree applied on baseline data only, and Calhoun et al.[30] developed RFs that handle repeated measurements for classification problems.

Finally, all those previous works considered standard data, where n was always larger (and often much larger) than p. Hence, the potential gain due to repeated measurements and the behavior of the approaches in a high-dimensional context were not studied, although applications in such context are skyrocketing.

In this work, we propose a general approach of RF method for high-dimensional longitudinal data. First, we develop a flexible stochastic semi-parametric mixed effects model and introduce a new RF method for longitudinal data. We compare all available tree-based methods for longitudinal data in an extensive simulation study especially in the context of high-dimensional data. Finally, the proposed method has been applied to real data from a therapeutic vaccine trial in HIV-infected patients.

All existing and proposed methods have been implemented together in an R[31] package called longituRF.[a]

2 The semi-parametric stochastic mixed effects model

Let us consider longitudinal data with n individuals, the ith individual having n_i observations over time. Suppose Y_{ij} (for all $i = 1, \ldots, n$ and $j = 1, \ldots, n_i$), the response of the ith individual at time t_{ij}, satisfies

$$Y_{ij} = f(X_{ij}) + Z_{ij}b_i + \omega_i(t_{ij}) + \varepsilon_{ij} \tag{1}$$

where X_{ij} is the $p \times 1$ vector of covariates, $f : \mathbb{R}^p \to \mathbb{R}$ is the unknown mean behavior function, b_i is a $q \times 1$ vector of random effects associated with a $1 \times q$ vector of covariates Z_{ij}, $\omega_i(t)$ is a stochastic process used to model serial correlation, and ε_{ij} denotes a measurement error.

For all $i = 1, \ldots, n$, the b_i are independent, as well as the $\omega_i(t)$. And the ε_{ij} are also independent for all $i = 1, \ldots, n; j = 1, \ldots, n_i$. We assume that $b_i, \omega_i(t)$ and ε_{ij} are mutually independent. We also suppose that the ε_{ij} are normally distributed as $\mathcal{N}(0, \sigma^2)$, the b_i are normally distributed as $\mathcal{N}(0, B)$ where B is a $q \times q$ positive definite matrix and $\omega_i(t)$ is a centered Gaussian process with covariance function $Cov(\omega_i(t), \omega_i(s)) = \gamma^2\Gamma(s, t)$ depending on a parameter γ^2. More precisely, we denote the covariance matrix of the stochastic process ω_i for the ith individual by $\left(\gamma^2\Gamma(t_{ij}, t_{ik})\right)_{1 \le j,k \le n_i} = \gamma^2 K_i$ where K_i is a positive definite matrix. It should be noted that function Γ, which depends only on the measurement times, fully determines the covariance structure of the stochastic process. For example, in the case of a standard Brownian motion, Γ is defined by $\Gamma(s, t) = \min(s, t)$. The parameter γ^2 tunes the variability of the stochastic process. We will also consider the case where Γ depends on an additional parameter α in the next section.

We consider in model (1) that the evolution of the response variable for the ith individual Y_i over time varies around a mean behavior function given by f. These variations specify the individual trajectories around f and are

driven by the random effects b_i and the stochastic process $\omega_i(t)$ for the ith individual. Note that if the function f is assumed linear then model (1) reduces to the linear stochastic model of Diggle and Hutchinson.[32]

Zhang et al.[33] already considered a semi-parametric stochastic mixed effects model but with f a function of the time only, hence model (1) can be seen as a generalization of their model. Hajjem et al.[22,27] considered non-stochastic model, i.e. model (1) without the stochastic process $\omega_i(t)$, because they only worked with clustered data. The closest approach to ours is the one of Sela and Simonoff,[23] which took into account the serial correlation by the use of autoregressive processes in semi-parametric mixed effects model but that was not extended to RFs.

In the following, we will consider high-dimensional cases where p, the number of variables of f, is much larger than $N = \sum_{i=1}^{n} n_i$ the total number of observations.

3 Estimation

We consider the vectorized form of model (1) as follows, for all $i = 1, \ldots, n$

$$Y_i = f_i + Z_i b_i + \omega_i + \varepsilon_i \tag{2}$$

where $f_i = \left(f(X_{i1}), \ldots, f(X_{in_i})\right)^T$, $Y_i = (Y_{i1}, \ldots, Y_{in_i})^T$, $Z_i = [Z_{i1}, \ldots, Z_{in_i}]^T$, $\omega_i = \left(\omega_i(t_{i1}), \ldots, \omega(t_{in_i})\right)^T$, and $\varepsilon_i = (\varepsilon_{i1}, \ldots, \varepsilon_{in_i})^T$.

A common ground among previous works is to use an adaptation of the maximum likelihood (ML)-based EM algorithm (as described in Wu and Zhang[34]) to estimate all quantities (all unknown parameters and the unknown mean behavior function) of model (1). The main principle of the estimation procedure is given in Algorithm 1, while further details can be found in the two following sections. Remark that all existing methods apply this estimation procedure, the difference being (i) in the methodology used to estimate the mean behavior function f at step 1 of Algorithm 1 and (ii) in the fact that they include or not a stochastic process in the model.

Algorithm 1: General estimation procedure for model (1)

Initialization: Let $r = 0$, $\hat{b}_{i,(0)} = 0_q$, $\hat{\omega}_{i,(0)} = 0_{n_i}$, $\hat{B}_{(0)} = I_q$, $\hat{\gamma}^2_{(0)} = 1$ and $\hat{\sigma}^2_{(0)} = 1$.

Repeat

1. Set $r = r + 1$, compute $\tilde{Y}_{ij,(r-1)} = Y_{ij} - Z_{ij}\hat{b}_{i,(r-1)} - \hat{\omega}_{ij,(r-1)}$ estimate f in the standard regression framework (with all N observations):

$$\tilde{Y}_{ij,(r-1)} = f(X_{ij}) + \varepsilon_{ij}$$

   to get $\hat{f}_{i,(r)}$.
   Then predict $\hat{b}_{i,(r)}$ and $\hat{\omega}_{i,(r)}$ using $\hat{B}_{(r-1)}, \hat{\gamma}^2_{(r-1)}, \hat{\sigma}^2_{(r-1)}$ and $\hat{f}_{i,(r)}$.

2. Update $\hat{B}_{(r)}$, $\hat{\gamma}^2_{(r)}$ and $\hat{\sigma}^2_{(r)}$ using $\hat{f}_{i,(r)}$, $\hat{b}_{i,(r)}$ and $\hat{\omega}_{i,(r)}$,

until *convergence*;

3.1 Mean behavior function estimation

At step 1 of Algorithm 1, the mean behavior function f could actually be estimated with any regression method, but in this work we focus on tree-based methods. Hajjem et al.[22] introduced **MERT** (**M**ixed **E**ffects **R**andom **T**rees), which consists in using a regression tree to estimate f in a model that does not include any stochastic process (because they focus on clustered data). More precisely, they used CART,[26] which consists in a binary data-driven recursive partitioning of the explanatory variables space. At each step of the partitioning, a node is split into two child nodes. Hence, the resulting partition can naturally be associated with a binary tree which is called a CART. Furthermore, we stress that each node splitting is optimized among all explanatory variables and that the CART algorithm works with two steps: the maximal tree building followed by a pruning step, which ensure to get the tree-structured predictor with the best prediction performance.

Later, Hajjem et al.[27] introduced **MERF** (**M**ixed **E**ffects **R**andom **F**orest) in which f is estimated using RF, again without including a stochastic process in the model. RFs[1] are obtained by aggregating a collection of

randomized CART, where the aggregation consists in averaging individual trees predictions. Each tree is a maximal tree, built using random perturbations: first, it is built on a bootstrap sample of the learning set, and second, at each step of the partitioning, the best split is optimized among a randomly drawn subset of explanatory variables. The size of the subset of variables, often called mtry, has usually a strong impact on RF performance: if mtry is too small, individual trees would give too poor predictions, and if mtry is too high, the collection of trees could be not diverse enough.[9,35] RFs naturally estimate the prediction error with the Out-Of-Bag (OOB) error as follows: to predict the response of one particular observation of the learning set, only trees built on bootstrap samples not containing this observation are aggregated. Furthermore, OOB samples (made of observations not selected in bootstrap samples) are also used to compute a variable importance (VI) score. For a fixed variable, the VI score of this variable is defined as the mean increase of the error of a tree on its associated OOB sample after a random permutation of this variable values.

Independently, Sela and Simonoff[23] introduced **REEMtree** (**R**andom **E**ffects **E**xpectation **M**aximization Tree) in a model that includes serial dependencies between observations with the use of an autoregressive process. **REEMtree** uses a CART T as a first step in the estimation of f. Once T is built, the associated partition (of the explanatory variables space) is used to fit a Linear Mixed Effects Model (**LMEM**). More precisely, let Φ^i be the indicator matrix defined by $\Phi^i_{j\ell} = \mathbb{1}_{\{X_{ij} \in g_\ell\}}$ where g_ℓ is the ℓth leaf of tree T and consider the following **LMEM** (which we write directly in the framework of model (2))

$$Y_i = \Phi^i \mu_T + Z_i b_i + \omega_i + \varepsilon_i$$

The vector of the leaves values μ_T is estimated by

$$\hat{\mu}_T = \left(\sum_{1 \le i \le N} (\Phi^i)^T V_i^{-1} \Phi^i \right)^{-1} \left(\sum_{1 \le i \le N} (\Phi^i)^T V_i^{-1} Y_i \right)$$

with $V_i = Var(Y_i) = Z_i B Z_i^T + \gamma^2 K_i + \sigma^2 I_{n_i}$ for all $i = 1, \ldots, N$. The advantage of this method is that the leaves values are updated by taking into account intra-individual covariance matrix V_i (instead of taking the simple mean of values as in **MERT**). Finally, f_i is estimated by $\hat{f}_i = \Phi^i \hat{\mu}_T$.

In this article, we propose a novel method, called **REEMforest**, which consists in aggregating a collection of randomized **REEMtrees**. More precisely, consider L randomized trees (as in standard RF) $T_1, \ldots, T_L$. Let $\Phi^{i,\ell}$ be the indicator matrix associated with the ℓth random tree T_ℓ and $\hat{\mu}_{T_\ell}$ the vector of leaves values of T_ℓ estimated within the stochastic **LMEM**

$$Y_i = \Phi^{i,\ell} \mu_{T_\ell} + Z_i b_i + \omega_i + \varepsilon_i$$

f_i is thus estimated by

$$\hat{f}_i = \frac{1}{L} \sum_{\ell=1}^{L} \Phi^{i,\ell} \hat{\mu}_{T_\ell}$$

All details about **REEMforest** can be found in Algorithm 2.

Algorithm 2: REEMforest algorithm

Initialization: Let $r = 0$, $\hat{b}_{i,(0)} = 0_q$, $\hat{\omega}_{i,(0)} = 0_{n_i}$, $\hat{B}_{(0)} = I_q$, $\hat{\gamma}^2_{(0)} = 1$ and $\hat{\sigma}^2_{(0)} = 1$.

Repeat

1. Set $r = r + 1$, compute $\tilde{Y}_{ij,(r-1)} = Y_{ij} - Z_{ij} \hat{b}_{i,(r-1)} - \hat{\omega}_{ij,(r-1)}$. Estimate f in the standard regression framework (with all N observations) with a RF, build the $\Phi^{j,\ell}$ matrices for every tree T_ℓ in the forest and estimate the leaves values $\hat{\mu}_{T_\ell}$. Aggregate the updated trees to get

$$\hat{f}_{i,(r)} = \frac{1}{L} \sum_{\ell=1}^{L} \Phi^{i,\ell} \hat{\mu}_{T_\ell}$$

Then predict $\hat{b}_{i,(r)}$ and $\hat{\omega}_{i,(r)}$ for all $i = 1, \ldots, n$

$$\hat{b}_{i,(r)} = \hat{B}_{(r-1)} Z_i^T \hat{V}_{i,(r-1)}^{-1} \left(Y_i - \hat{f}_{i,(r)} \right)$$

$$\hat{\omega}_{i,(r)} = \hat{\gamma}_{(r-1)}^2 K_i \hat{V}_{i,(r-1)} \left(Y_i - \hat{f}_{i,(r)} \right)$$

2. Update $\hat{B}_{(r)}$, $\hat{\gamma}_{(r)}^2$ and $\hat{\sigma}_{(r)}^2$

$$\hat{B}_{(r)} = \frac{1}{n} \sum_{i=1}^{n} \left\{ \hat{b}_{i,(r)} \hat{b}_{i,(r)}^T + \hat{B}_{(r-1)} - \hat{B}_{(r)} Z_i^T \hat{V}_{i,(r-1)}^{-1} Z_i \hat{B}_{(r)} \right\}$$

$$\hat{\gamma}_{(r)}^2 = \frac{1}{N} \sum_{i=1}^{n} \left\{ \hat{\omega}_{i,(r)}^T K_i^{-1} \hat{\omega}_{i,(r)} + \hat{\gamma}_{(r-1)}^2 \left(n_i - \hat{\gamma}_{(r-1)}^2 tr\left(\hat{V}_{i,(r-1)}^{-1} K \right) \right) \right\}$$

$$\hat{\sigma}_{(r)}^2 = \frac{1}{N} \sum_{i=1}^{n} \hat{\varepsilon}_{i,(r)}^T \hat{\varepsilon}_{i,(r)} \hat{\sigma}_{(r-1)}^2 tr\left(\hat{V}_{i,(r-1)}^{-1} \right)$$

with $\hat{\varepsilon}_{i,(r)} = Y_i - \hat{f}_{i,(r)} - Z_i \hat{b}_{i,(r)} - \hat{\omega}_{i,(r)}$

$$\forall i = 1, \ldots, n$$

until *convergence*;

To sum up, **MERT**, **MERF**, and **REEMtree** are the already existing methods and we introduce **REEMforest** method that generalizes all previous methods. Our method is extended to RF which is an important component, especially in the context of high-dimensional data and the stochastic part of the model includes a general Gaussian process (Ornstein–Uhlenbeck process and fractional Brownian motion) in addition to the random effects. In the following, when a stochastic process is indeed included in the model, we add an **S** (for **S**tochastic) at the beginning of the method names, so in this case we denote the methods by **SMERT**, **SMERF**, **SREEMtree**, and **SREEMforest**, respectively.

3.2 Prediction of random effects and stochastic process

Once $\hat{f}_i$ has been computed (by either of the previously described methods), the predictions for the random effects b_i and the stochastic processes ω_i for fixed parameters $\left(B, \gamma^2, \sigma^2\right)$ are obtained by taking their conditional expectations given the data Y_i. The best linear unbiased predictors **BLUP** are thus

$$\hat{b}_i = B Z_i^T V_i^{-1} \left(Y_i - \hat{f}_i \right)$$
$$\hat{\omega}_i = \gamma^2 K_i V_i^{-1} \left(Y_i - \hat{f}_i \right)$$

This ends step 1 of Algorithm 1.

3.3 Variance components estimation

At step 2 of Algorithm 1, the estimation of the variance parameters is obtained by taking the conditional expectation of their ML estimators given the data Y_i. Thanks to the conditional independence between the individuals we can write, for fixed f_i, $i = 1, \ldots, n$, the likelihood function associated with model (2) as follows

$$\mathcal{L}(B, \gamma^2, \sigma^2; Y) = \prod_{i=1}^{n} \mathcal{L}_i\left(Y_i; B, \gamma^2, \sigma^2 \right)$$

with $\mathcal{L}_i(Y_i; B, \gamma^2, \sigma^2)$ the density function on the vector Y_i. Moreover, since $Y_i|b_i, \omega_i \sim \mathcal{N}(f_i + Z_i b_i + \omega_i, \sigma^2 I_{n_i})$, by using the independence of b_i, ω_i, and ε_i, we can easily write the likelihood function $\mathcal{L}$ as

$$\mathcal{L}(B, \gamma^2, \sigma^2; Y) = \prod_{i=1}^{n} \frac{1}{(2\pi\sigma^2)^{\frac{n_i}{2}}} \exp\left\{ -\frac{1}{2\sigma^2} (Y_i - f_i - Z_i b_i - \omega_i)(Y_i - f_i - Z_i b_i - \omega_i)^T \right\} \times \frac{1}{(2\pi)^{\frac{q}{2}} \sqrt{\det(B)}} \times \exp\left\{ \frac{1}{2} b_i^T B^{-1} b_i \right\} \times \frac{1}{(2\pi)^{\frac{n_i}{2}} \sqrt{\det(\gamma^2 K_i)}} \times \exp\left\{ \frac{1}{2} \omega_i^T (\gamma^2 K_i)^{-1} \omega_i \right\}$$

Using that $Y_i - f_i - Z_i b_i - \omega_i = \varepsilon_i$, the ML estimators of B, γ^2, and σ^2 are

$$\tilde{B} = \frac{1}{n} \sum_{i=1}^{n} b_i b_i^T, \quad \tilde{\gamma}^2 = \frac{1}{N} \sum_{i=1}^{n} \omega_i^T K_i^{-1} \omega_i, \quad \tilde{\sigma}^2 = \frac{1}{N} \sum_{i=1}^{n} \varepsilon_i^T \varepsilon_i$$

Because b_i, ω_i, and ε_i are unknown these estimators are not computable, this is why we take the expectation given the data Y_i. The conditional expectations of the estimators $\tilde{B}$ and $\tilde{\sigma}^2$ are given in Wu and Zhang[34]

$$\hat{B} = \mathbb{E}(\tilde{B}|Y) = \frac{1}{n} \sum_{i=1}^{n} \left\{ \hat{b}_i \hat{b}_i^T + B - B Z_i^T V_i^{-1} Z_i B \right\}$$
$$\hat{\sigma}^2 = \mathbb{E}(\tilde{\sigma}^2|Y) = \frac{1}{N} \sum_{i=1}^{n} \left\{ \hat{\varepsilon}_i^T \hat{\varepsilon}_i + \sigma^2 tr(V_i^{-1}) \right\}$$

The conditional expectation of the ML estimator $\tilde{\gamma}^2$ given the data Y_i is

$$\begin{aligned} \hat{\gamma}^2 = \mathbb{E}(\tilde{\gamma}^2|Y_i) &= \frac{1}{N} \sum_{i=1}^{n} tr\left(\mathbb{E}(K_i^{-1} \omega_i \omega_i^T | Y_i) \right) \\ &= \frac{1}{N} \sum_{i=1}^{n} tr\left(K_i^{-1} \hat{\omega}_i \hat{\omega}_i^T + Cov(K_i^{-1} \omega_i, \omega_i | Y_i) \right) \\ &= \frac{1}{N} \sum_{i=1}^{n} \left(\hat{\omega}_i^T K_i^{-1} \hat{\omega}_i + \gamma^2 \left(n_i - \gamma^2 tr\left(V_i^{-1} K_i \right) \right) \right) \end{aligned}$$

Estimators of variance parameters B, γ^2, and σ^2 at step 2 are thus given by $\hat{B}$, $\hat{\gamma}^2$, and $\hat{\sigma}^2$, respectively.

Gaussian processes such as Ornstein–Uhlenbeck process and fractional Brownian motion have a variance–covariance function $\mathrm{Cov}(\omega_i(s), \omega_i(t)) = \gamma^2 \Gamma(s, t; \alpha)$ which depends on two parameters γ^2 and α. In this case, we can write the covariance matrix of the stochastic process ω_i for the ith individual as $\left(\gamma^2 \Gamma(t_{ij}, t_{ik}; \alpha) \right)_{1 \le j,k \le n_i} = \gamma^2 K_i(\alpha)$ with $K_i(\alpha)$ depending on α. There is no analytic ML estimator of α. However, for a fixed value of α, the estimation procedure described in this section holds. Thus, for $\mathcal{H} = \{\alpha_1, \ldots, \alpha_d\}$, an ensemble of possible values of α parameter, the estimator of α is

$$\hat{\alpha} = \underset{\alpha \in \mathcal{H}}{\mathrm{argmax}}\, l(B, \gamma^2, \alpha, \sigma^2; y)$$

where l is the log-likelihood function.

4 Simulation study

4.1 Simulation model

4.1.1 Explanatory variables

In this section, we detail how the data matrix of the explanatory variables X is simulated. Our choices are motivated by the characteristics of the data coming from our application, which are transcriptomics data in a phase 1/2 vaccine trial (see section 5 for more details), called the DALIA trial.

As usual in high-dimensional contexts, we assume that a large majority of variables are not associated with the response variable Y (also known as a *sparsity* assumption). In our study, those variables are simulated as i.i.d. random draws from a multivariate Gaussian distribution $\mathcal{N}(0, 3I_N)$, where I_N denotes the identity matrix of size N (recall that $N = \sum_{i=1}^{n} n_i$ denotes the total number of observations).

Moreover, since we deal with longitudinal data in the context of gene expression, we assume that some explanatory variables vary over time and that some explanatory variables are clustered into groups (which correspond to genes involved in the same biological pathway).

Hejblum et al.[36] highlighted 10 examples of groups of genes with different temporal behaviors in the DALIA trial, and we mimic some of these trends by setting the following six behaviors over time in our simulations

$$\begin{cases} C_{g_1}(t) = 2.44 + 0.04\left(t - \dfrac{3(t-6)^2}{t}\right) \\ C_{g_2}(t) = 0.5t - 0.1(t-5)^2 \\ C_{g_3}(t) = 0.25t - 0.05(t-6)^2 \\ C_{g_4}(t) = \cos\left(\dfrac{t-1}{3}\right) \\ C_{g_5}(t) = 0.1t + \sin(0.6t + 1.3) \\ C_{g_6}(t) = -0.03t^2 \end{cases} \tag{3}$$

The explanatory variables with a temporal behavior are then simulated as follows

$$X^{(k)}(t) = C_{g_{(k)}}(t) + \zeta_k + \varepsilon_t$$

where $g_{(k)}$ is the group of the kth covariate $X^{(k)}$; $\zeta_k \sim \mathcal{N}(0, 0.1)$ corresponds to a random translation at the group level and $\varepsilon_t \sim \mathcal{N}(0, 0.2)$ is an additional time-dependent variability. Plots of Figure 1 give the explanatory variables trajectories associated with one simulated dataset under model (3).

In the following, we investigate two situations with different values of the total number of variables, p, as well as different sizes of each group of variables with temporal behavior.

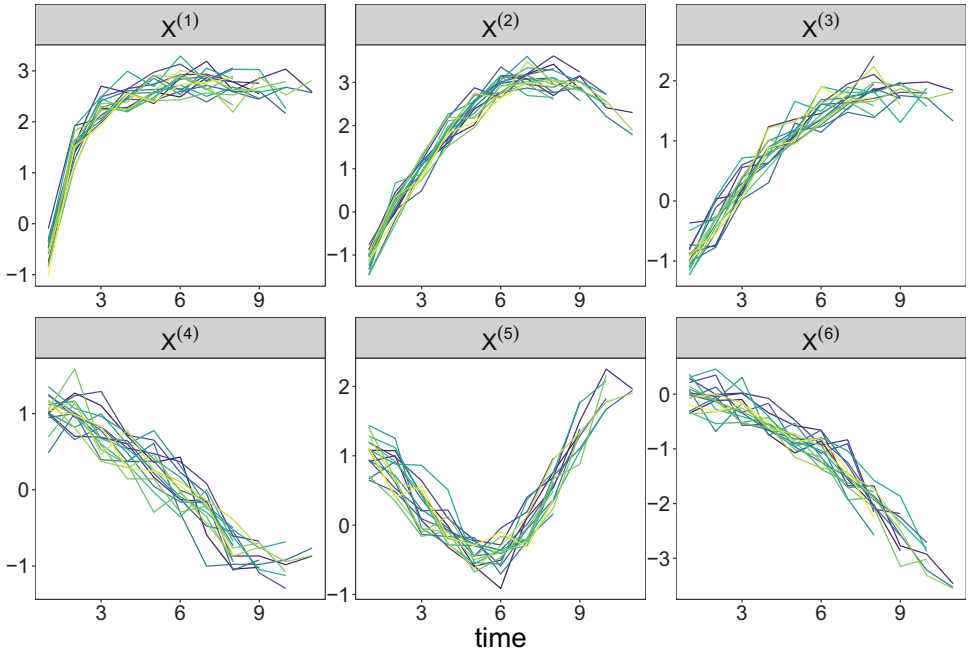

Figure 1. Dynamics of explanatory variables (one curve per individual, $n = 17$) simulated under model (3) in the low-dimensional case.

4.1.2 Outcome variable

The two following models, which are special cases of model 2, are used to simulate the outcome variable Y. For all $i = 1, \ldots, n$

$$Y_i = f_i + b_{0i} + z_i b_{1i} + \varepsilon_i \quad \text{(non-stochastic model)} \tag{4}$$

$$Y_i = f_i + b_{0i} + z_i b_{1i} + \omega_i + \varepsilon_i \quad \text{(stochastic model)} \tag{5}$$

where $(b_{0i}, b_{1i})^T \sim_{i.i.d} \mathcal{N}(0, B)$ with $B = \begin{pmatrix} 0.5 & 0.6 \\ 0.6 & 3 \end{pmatrix}$, ω_i is a Brownian motion with volatility $\gamma^2 = 0.8$, and $\varepsilon_i \sim_{i.i.d} \mathcal{N}(0, \sigma^2 I_{n_i})$ with $\sigma^2 = 0.5$. In these models, the random effect b_1 is associated with an exogenous variable $Z = (z_1, \ldots, z_n)^T$, where $z_i \sim_{i.i.d} \mathcal{U}([0, 3])$ for $i = 1, \ldots, n$.

The mean behavior function depends on the dimension of the simulated data:

- In the low-dimensional case (with $p = 6$)

$$f(x) = 1.3 \times \left(x^{(1)}\right)^2 + 2 \times \left|x^{(2)}\right|^{1/2} \tag{6}$$

- In the *high-dimensional case* (with $p = 8000$ and with at least 20 variables in the first two groups of explanatory variables)

$$f(x) = 1.3 \times \left(\frac{1}{20} \sum_{g \in g_1^{20}} X^{(g)}\right)^2 + 2 \times \left|\frac{1}{20} \sum_{g \in g_2^{20}} X^{(g)}\right|^{1/2} \tag{7}$$

where g_1^{20} and g_2^{20} represent two sets of 20 genes randomly picked from the group g_1 and g_2, respectively.

The mean behavior function is actually quite the same in the two situations. The difference lies in the fact that in the high-dimension case, 40 variables are related to the response variable, against two in the low-dimension case. It is indeed reasonable, in high-dimensional genomic data, to assume that several genes coming from the same group are linked to the mean behavior function f.

4.2 Squared bias and prediction error

The different methods are compared in terms of squared bias (associated with each estimated quantity) and prediction performance, computed among M repetitions of the simulation.

Squared biases are defined as follows

$$\text{bias}^2\left(\hat{f}^M\right) = \frac{1}{n \# \mathbb{T}} \sum_{t \in \mathbb{T}} \sum_{i=1}^{n} \left\{\hat{f}^M(X_i(t)) - f(X_i(t))\right\}^2$$

$$\text{bias}^2(\hat{B}^M) = \frac{1}{q^2} \sum_{1 \le k,l \le q} \left(\hat{B}^M_{kl} - B_{kl}\right)^2$$

$$\text{bias}^2\left(\hat{\gamma}^2_M\right) = \left(\hat{\gamma}^2_M - \gamma^2\right)^2 \quad \text{bias}^2\left(\hat{\sigma}^2_M\right) = \left(\hat{\sigma}^2_M - \sigma^2\right)^2$$

with

- $\mathbb{T}$ a fixed grid of times (different from the times of measurements),
- $\hat{f}^M(X_i(t)) = \frac{1}{M} \sum_{m=1}^{M} \hat{f}^m(X_i(t)) \quad \forall t \in \mathbb{T}, \forall i = 1, \ldots, n$,
- $\hat{f}^m$ the RF returned by the algorithm after convergence, associated with the mth repetition,
- $\hat{B}^M = \frac{1}{M} \sum_{m=1}^{M} \hat{B}^m$, $\hat{\gamma}^2_M = \frac{1}{M} \sum_{m=1}^{M} \hat{\gamma}^2_m$, $\hat{\sigma}^2_M = \frac{1}{M} \sum_{m=1}^{M} \hat{\sigma}^2_m$, and
- $\hat{B}^m$ the estimation of B on the mth repetition and similarly for $\hat{\gamma}^2_m$ and σ^2_m.

To evaluate prediction performance, each simulated dataset is split into a learning set and a test set, the test set being made of the two last measurements of each individual. With $\mathcal{T}_i^\ell$ denoting the index of the ith individual measurements in the ℓth test set, we define the prediction error as

$$\frac{1}{2nM}\sum_{\ell=1}^{M}\sum_{i=1}^{n}\sum_{j\in\mathcal{T}_i^\ell}\left(Y_{ij}-\hat{Y}_{ij}^{\ell}\right)^2$$

where $\hat{Y}_{ij}^{\ell}$ is the predicted response variable (see equation (8) below), for the jth measure of the ith individual, given by the RF returned by the algorithm after convergence.

The prediction of the response variable for the ith individual at time t is given by

$$\hat{Y}_i(t)=\hat{f}(X_i(t))+Z_i(t)\hat{b}_i+\tilde{\omega}_i(t) \tag{8}$$

with $X_i(t)$ and $Z_i(t)$ the fixed and random effects explanatory variables for the ith individual at time t and

$$\tilde{\omega}_i(t)=\begin{cases}\dfrac{1}{t_+-t_-}[(t-t_-)\hat{\omega}_i(t_-)+(t-t_+)\hat{\omega}_i(t_+)] & \text{if } t_{i,1}\le t\le t_{i,n_i}\\ \mathbb{E}\big(\omega_i(t)|\hat{\omega}_i(t_{i,1})\big)=\dfrac{\Gamma(t,t_{i,1})}{\Gamma(t_{i,1},t_{i,1})}\hat{\omega}_i(t_{i,1}) & \text{if } t<t_{i,1}\\ \mathbb{E}\big(\omega_i(t)|\hat{\omega}_i(t_{i,n_i})\big)=\dfrac{\Gamma(t,t_{i,n_i})}{\Gamma(t_{i,n_i},t_{i,n_i})}\hat{\omega}_i(t_{i,n_i}) & \text{if } t>t_{i,n_i}\end{cases}$$

with $t_-=\max\big(\{s\in\{t_{i,1},\ldots,t_{i,n_i}\},s\le t\}\big)$ and $t_+=\min\big(\{s\in\{t_{i,1},\ldots,t_{i,n_i}\},s\ge t\}\big)$.

4.3 Results

The number of individuals n, is fixed to 17 (the same as in the DALIA trial) all along the simulation study, and the number of measurements n_i for the ith individual is randomly chosen (with uniform distribution) between 8 and 11 for every $i=1,\ldots,n$, leading to an unbalanced design. We recall that the total number of observations is denoted by $N=\sum_{i=1}^{n}n_i$.

4.3.1 A low-dimensional case

We start by considering a low-dimensional example where $p=6$. We have six explanatory variables in the dataset and all of them have a temporal behavior (given by equation (3)). This framework allows to compare different tree-based methods as well as a linear mixed model for longitudinal data in a standard framework. First, we simulate one dataset under the non-stochastic model (4) using the mean behavior function f defined in equation (6) and study the behavior of the **MERF** method on that dataset.

Figure 2 shows that the convergence of the ML-EM algorithm for the **MERF** method is quite affected by the mtry parameter value (the number of variables randomly drawn before optimizing the node splitting in the trees composing the RF). Standard RFs are already sensitive to this parameter[9,35] but MERF is even more sensitive to it, because a sub-optimal value leads to a bad estimation of f which could also lead to bad predictions of random effects. In this example, mtry must be set at least equal to 3, in all our experiments we chose mtry $=4$.

We now simulate 100 datasets (again under model (4) with mean behavior function (6)) and study squared biases on estimations of quantities of interest (f, B, σ^2, and γ^2 when appropriate), given by the four tree-based methods (**MERT** and **REEMtree** for trees, **MERF** and **REEMforest** for forests) and also compare to the **LMEM** method. To study the robustness of the methods including stochastic process estimation, we also apply them by specifying a Brownian motion as stochastic process.

As shown in Table 1, when the models are well specified, **LMEM** leads to much higher biases on all parameters compared to all other methods. Tree-based methods **MERT** and **REEMtree** are close to each other in terms of bias on f while **MERF** and **REEMforest** which use RFs, provide a much better mean behavior estimation. Moreover, the squared bias on f for **REEMforest** is about 20% lower than **MERF** whereas the squared bias on f of **REEMtree** is only 6% lower than the one obtained with **MERT**. Hence, in this framework, taking into account the intra-individual covariance structure to evaluate the tree leaves values generates a much more

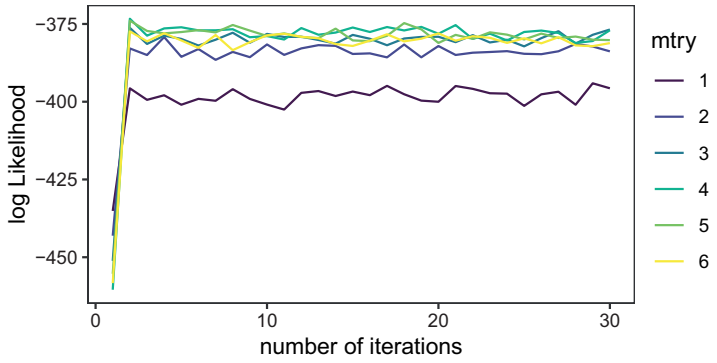

Figure 2. Evolution of the log-likelihood against the number of iterations in **MERF** method for different mtry values, data simulated under model (4) in the low-dimensional case.

important decrease of the squared bias on f when RFs are used instead of CART. Furthermore, the squared bias obtained on the random effects covariance matrix B and the residual variance parameter σ^2 are lower for all four tree-based methods compared to **LMEM**, with forests estimating σ^2 much better than trees. Finally, **REEMforest** gives slightly lower bias than **MERF**. In the case of misspecified models (that include a Brownian motion when there is no stochastic process), biases on f and σ^2 increase for almost all methods. It seems quite unavoidable since the covariance structure is changed, but in what follows we show that this increase does not harm much the results in terms of prediction performance. It is worth noting that, even in the misspecified case, the forest methods **SMERF** and **SREEMforest** still obtain a much lower bias than the ones obtained in the well-specified case by the other methods.

Next, we simulate 100 additional datasets under model (5) (still with f defined by equation (6)) and compare the five methods described above first when the stochastic process is well specified as a Brownian motion, second when it is misspecified as an Ornstein–Uhlenbeck process and then when no stochastic process is specified.

As shown in Table 2, when the models are well specified, **SLMEM** again leads to much higher biases on all parameters compared to the other methods. Concerning tree-based methods, **SMERT** performs better than **SREEMtree** on f while **SREEMtree** leads to much lower biases on γ^2 and σ^2. Forests methods (**SMERF** and **SREEMforest**) still perform better than trees with squared biases almost all much lower. In the first misspecified case, in which the models do not include a stochastic process, we notice a very high increase in biases on B and σ^2 for all methods. Nevertheless, the bias on f is quite stable for all methods except for **REEMtree** for which it decreases. Thus, not specifying a process when there is one does not prevent us from estimating the mean behavior function f very well. In the second misspecified case, i.e. when an Ornstein–Uhlenbeck process is used instead of the Brownian motion used to simulate the datasets, biases on B, σ^2, and especially on γ^2 increase. This was expected since those parameters help to model the residual variance and strongly depends on the specification of the stochastic process. However, the bias on f remains stable (except for **SREEMtree** for which it decreases), this illustrates the ability of the **(S)MERT**, **(S)REEMtree**, **(S)MERF**, and **(S)REEMforest** methods to correctly estimate the mean behavior function f even when the stochastic process is misspecified.

Finally, we compare the different methods on their prediction capacity by computing prediction errors on 100 simulated datasets, either under model (4) or (5). For each dataset, a test set if formed by picking, for each individual i, the two last observations. This gives a test set containing $2n$ observations and a learning set with $N - 2n$ observations. Breiman's RF is also included in this study in addition to the five methods already mentioned to illustrate the gain of taking into account the intra-individual correlation.

When the models are well specified, Figures 3 and 4 show that **(S)LMEM** reached a poor prediction ability, because f is not linear. **(S)MERT** and **(S)REEMtree** gave intermediate performance, whereas **(S)MERF** and **(S)REEMforest** reached the lowest test errors, with similar performance. Breiman's RFs were not included in the graphs because they are insensitive to changes in the model, but they reach quite high test errors (on average 6.74 and 13.15 for the datasets simulated under model (4) and (5), respectively).

In the misspecified cases, Figure 3 shows that using a stochastic process (Brownian motion), when there is none, increases the prediction error for all methods. This increase is more contained for the forest-based methods **SMERF** and **SREEMforest**. Figure 4 shows that specifying the wrong stochastic process increases prediction errors for **SMERT**, **SREEMtree**, **SMERF**, and **SREEMforest** methods. In addition, when no stochastic process is

Table 1. Squared bias of the estimated parameters, averaged on 100 datasets simulated under model (4) in the low-dimensional case, obtained with either well-specified models or misspecified models (which include a Brownian motion).

	f	B	γ^2	σ^2
Well-specified models				
LMEM	17.786	5.452	*	2.632
MERT	1.015	0.445	*	0.068
REEMtree	0.952	0.412	*	0.067
MERF	0.377	0.497	*	0.013
REEMforest	0.293	0.472	*	0.013
Misspecified models				
SLMEM	19.271	5.483	*	1.963
SMERT	1.287	0.476	*	0.112
SREEMtree	1.547	0.445	*	0.051
SMERF	0.495	0.504	*	0.033
SREEMforest	0.433	0.469	*	0.033

*means the bias for the parameter γ^2 is not computable, either because the data are simulated without a stochastic process or because no process is specified in the model.

Table 2. Squared bias of the estimated parameters, averaged on 100 datasets simulated under model (5) in the low-dimensional case, obtained with either well-specified models (that include a Brownian motion) or misspecified models (that include either no stochastic process or an Ornstein–Uhlenbeck process).

	f	B	γ^2	σ^2
Well-specified models				
SLMEM	16.647	3.791	0.129	2.555
SMERT	1.787	0.521	0.065	0.490
SREEMtree	2.200	0.493	0.019	0.144
SMERF	0.814	0.521	0.034	0.012
SREEMforest	0.779	0.510	0.036	0.013
Misspecified models				
LMEM	12.275	4.303	*	9.064
MERT	1.517	0.986	*	1.714
REEMtree	1.358	0.898	*	1.695
MERF	0.781	1.052	*	2.382
REEMforest	0.783	1.041	*	2.404
SLMEM	13.576	4.351	1.000	2.722
SMERT	1.764	0.783	0.138	0.601
SREEMtree	1.599	0.727	0.274	0.258
SMERF	0.832	0.902	0.513	0.029
SREEMforest	0.784	0.891	0.458	0.027

* means the bias for the parameter γ^2 is not computable, either because the data are simulated without a stochastic process or because no process is specified in the model.

included in the model, prediction errors increase for all methods (except for **LMEM**). More precisely, for the tree-based methods, prediction errors are comparable the two misspecified cases, whereas for forest-based methods, the performance is worse when no stochastic process is included. It therefore seems preferable to always use a stochastic process.

Finally, to highlight the stability of **(S)REEMforest** under model misspecification, VI scores, computed with the RF returned after convergence of the method, are plotted in Figure 5 for both non-stochastic and stochastic models with either well-specified or misspecified models (for the stochastic model, we only consider the misspecified case where an Ornstein–Uhlenbeck process is used). We can see that VI is very stable in every settings, even with misspecified models. Note that, since the variables $X^{(2)}$ and $X^{(3)}$ have very similar trajectories (see Figure 1), their importance is close. This illustrates that misspecification of the stochastic process has limited impact on the search of variables that are strongly related to the outcome.

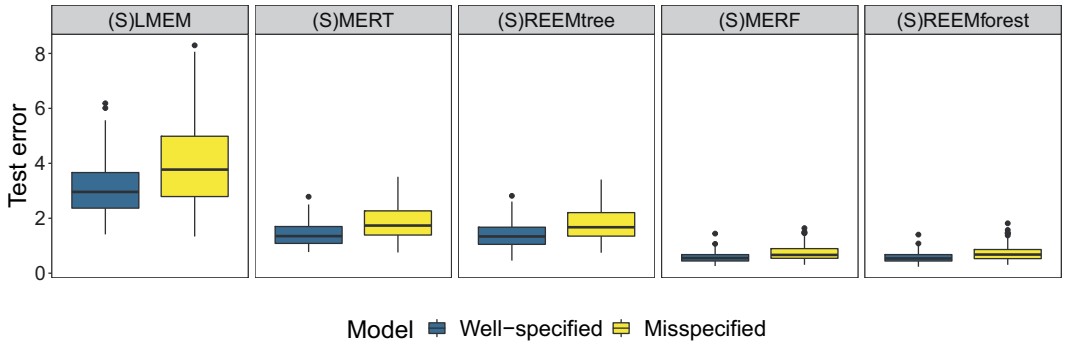

Figure 3. Boxplots of the test errors computed on 100 simulated datasets in the low-dimensional case under model (4). For each method (in column), the prediction errors were obtained either with well-specified models (with the same parameters as those used to simulate the data) or with misspecified models (which use a Brownian motion while none was used to generate the data).

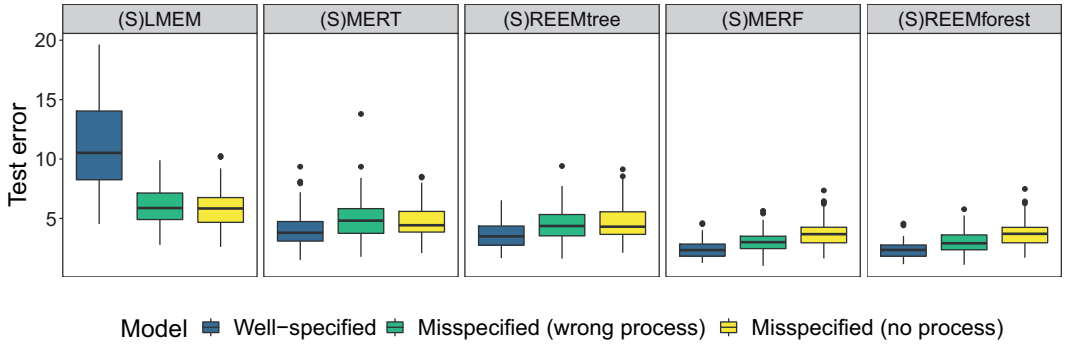

Figure 4. Boxplots of the test errors computed on 100 simulated datasets in the low-dimensional case under model (5). For each method (in column), the prediction errors were obtained either with well-specified models (with the same parameters as those used to simulate the data) or with misspecified models, i.e. with an Ornstein–Uhlenbeck process instead of the Brownian motion used to generate the data (wrong process) or models without stochastic process (no process).

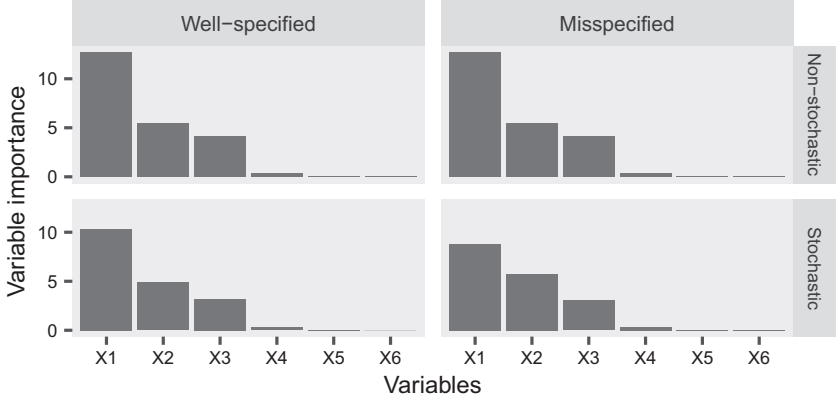

Figure 5. Barplots of the RF VI scores, computed after convergence of the **REEMforest** method, obtained on one dataset in the low-dimensional case, simulated either under model (4) at the top or under model (5) at the bottom. Results obtained with well-specified models are on the left, while those with misspecified models are on the right.

As a conclusion, we demonstrate the benefits of RF approaches for longitudinal data analysis in a low-dimensional case, especially in terms of prediction error. Moreover, those methods appear rather robust to misspecification of the stochastic process. **REEMforest** exhibited a slight advantage compared to **MERF** in terms of validity of the estimation of the mean behavior function f and of the other parameters B, σ^2, and γ^2.

4.4 A high-dimensional case

For the high-dimensional context, we kept $n = 17$ but set $p = 8000$. We also set the size of each of the six groups containing explanatory variables with temporal behaviors (given by equation (3)) to 266, leading to a total of 1596 variables that changed over time among the 8000 variables in the dataset.

First of all, according to Figure 2 for the low-dimensional case and some preliminary experiments, we fix the mtry parameter of RF to 5000 in all RF runs. This ensures convergence of ML-EM algorithms and avoids a too heavy computational load compared to an optimization of mtry for each RF.

We simulated 100 datasets under model (4) (and 100 other datasets under model (5)) with the mean behavior function given by equation (7) and computed squared biases on estimations given by the four tree-based methods: **(S)MERT**, **(S)REEMtree**, **(S)MERF**, and **(S)REEMforest**. We did not compare anymore with **LMEM** which is not adapted to the high-dimensional setting (Table 3).

For the non-stochastic scheme, the squared bias on f and all parameters obtained with **REEMforest** was slightly lower than the one obtained with the existing **MERF** method. For the stochastic model (5), the two forest-based methods gave similar bias on all parameters. As in the low-dimensional context, forests led systematically to lower biases on all estimations compared to trees, especially for the estimation of f. Concerning trees, we note that **REEMtree** gave much more precise estimation of f compared to **MERT**, especially in the stochastic case. However, **(S)MERF** and **(S)REEMforest** performed quite similarly.

We estimated the prediction errors of the different methods on test samples consisting of the last two measurements of each individual trajectory in each of the simulated datasets (as in the low-dimensional case). As illustrated in Figure 6, forests reached very low prediction error estimations compared to trees. This last result was expected because RFs perform better than trees most of the time[1] and especially for high-dimensional data.[37] In addition, **(S)REEMtree** performed better than **(S)MERT**, whereas **(S)MERF** and **(S)REEMforest** gave similar results. This suggests that the gain brought by the update of leaves values is less visible after aggregating an ensemble of trees. It can also be seen that Breiman's RFs (which assume independence between all observations in

Table 3. Squared bias of the estimated parameters, averaged on 100 datasets respectively simulated under model (4) and (5) in the high-dimensional case.

	f	B	γ^2	σ^2
Non-stochastic model				
MERT	1.902	0.603	*	0.112
REEMtree	1.543	0.499	*	0.070
MERF	0.750	0.504	*	0.005
REEMforest	0.729	0.493	*	0.005
Stochastic model				
SMERT	5.229	0.926	0.113	0.590
SREEMtree	3.519	0.738	0.071	0.065
SMERF	1.378	0.511	0.024	0.010
SREEMforest	1.367	0.496	0.023	0.011

*means the bias for the parameter γ^2 is not computable, either because the data are simulated without a stochastic process or because no process is specified in the model.

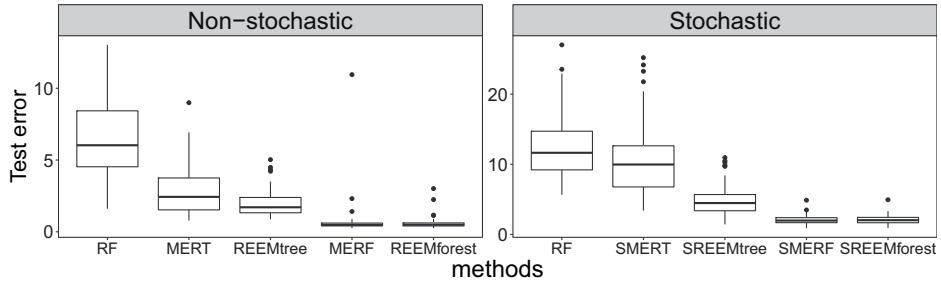

Figure 6. Boxplots of test errors computed on 100 simulated datasets, under either model (4) on the left or model (5) on the right, in the high-dimensional case.

the data) are competitive compared to trees, especially compared to **(S)MERT**. Hence, in that case, the gain of using RF instead of trees roughly compensates the fact that Breiman's RFs do not take into account the longitudinal feature of the data. We also studied biases and prediction errors obtained for the different methods under misspecification (not shown here) and we obtained results similar to those previously commented in the low-dimensional case. **(S)MERT**, **(S)REEMtree**, **(S)MERF**, and **(S)REEMforest** methods remain quite robust under misspecification in the high-dimensional case.

Finally, VI scores computed with the RF returned after convergence of the **REEMforest** method are plotted in decreasing order of VI in Figure 7 (only the 65 most important variables are plotted for the sake of clarity). This graph shows that the most important variables belong to one of the first three groups of explanatory variables. This result is satisfactory because the mean behavior function (defined by equation (7)) depends on variables that belongs to the first two groups only and the third group is very close to the second one in terms of dynamics (see equation (3)).

5 Application to the DALIA vaccine trial

DALIA is a therapeutic phase 1/2 vaccine trial including 19 HIV-infected patients who received an HIV vaccine before stopping their antiretroviral treatment (HAART). For a full description of the DALIA vaccine trial, we refer to Lévy et al.[38]

At each harvest time, 32,979 gene transcripts were measured as well as the plasma HIV RNA concentration (which was log-transformed) for every patient. In this application, we were interested in finding the gene transcripts associated with the HIV viral load dynamics after antiretroviral treatment interruption. The analysis was performed on the 17 patients with available data at the time of treatment interruption.

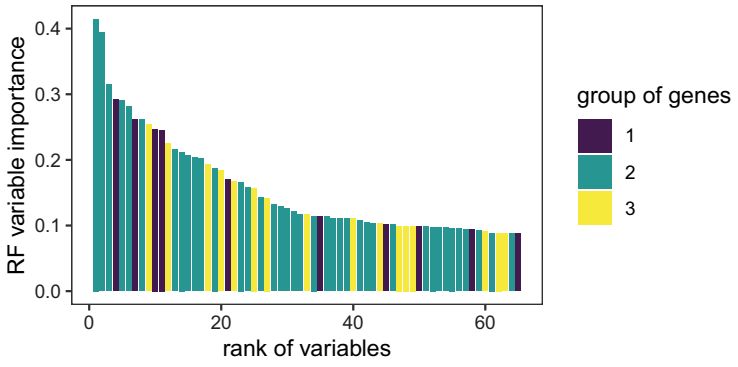

Figure 7. Barplot of the first 65 sorted (in decreasing order) VI scores, computed after convergence of the **REEMforest** method applied on one dataset simulated under model (4) in the high-dimensional case.

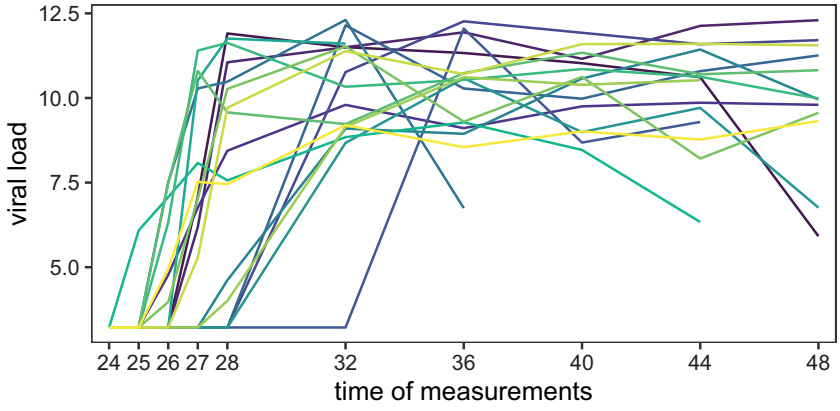

Figure 8. Dynamics of plasma HIV viral load (one curve per patient) after antiretroviral treatment interruption, DALIA vaccine trial.

Figure 8 illustrates the dynamics of the viral replication after antiretroviral treatment interruption with a large between-individuals variability.

A random intercept and a Gaussian process were included in the model

$$Y_{ij} = f(X_{ij}) + b_{0i} + \omega_i(t_{ij}) + \varepsilon_{ij} \quad i = 1, \ldots, n; \quad j = 1, \ldots, n_i \tag{9}$$

and we will refer to methods only using a random intercept as non-stochastic methods (**MERF** and **REEMforest** in the following). Moreover, as suggested by the simulation experiments, we did not include **MERT** and **REEMtree** methods because of the really high dimension of the problem.

Prediction errors were evaluated with 25 training/test sets random splits. As in the simulation study, a test set was obtained by randomly drawing two observations for each individual. We chose the stochastic process (between an Ornstein–Uhlenbeck's process and a fractional Brownian motion) that minimized the estimated prediction error. Hence, the fractional Brownian motion with Hurst exponent $h = \frac{1}{2}$ which is the standard Brownian motion was selected. Finally, the mtry parameter was fixed to $9p/10 = 29{,}681$ in all experiments of this section, according to the likelihood profile (Figure 9).

As illustrated in Figure 10, Breiman's RFs were comparable in terms of prediction error with **MERF** and **REEMforest** which only included a random intercept. However, **SMERF** and **SREEMforest** outperformed RF, with a slight advantage to **SREEMforest**. This confirms, in this real dataset, that taking precisely into account the longitudinal aspect of the data in RF leads to a significant drop of the prediction error. Furthermore, this illustrates that the methods introduced in this article (**SMERF** which generalizes **MERF** in the stochastic model and **SREEMforest** which generalizes **SREEMtree**) are the most suited to analyze high-dimensional longitudinal data.

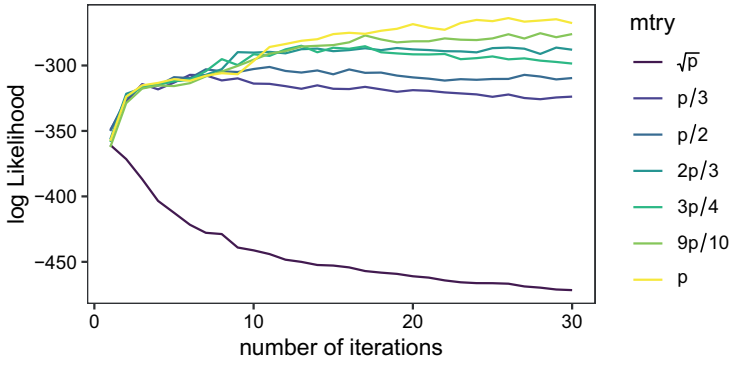

Figure 9. Log-likelihood according to the number of iterations in **SREEMforest** from the model (9) with standard Brownian motion, DALIA trial.

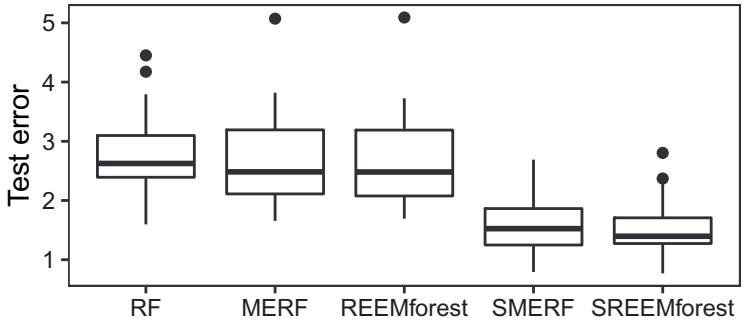

Figure 10. Boxplots of test errors computed using 25 training/test sets random splits, for Breiman's RF, **MERF**, **REEMforest**, **SMERF**, and **SREEMforest**, DALIA trial.

5.1 Variable selection using RFs

Once the algorithm (e.g. **SREEMforest**) has converged, a variable selection process is applied to select the genes the most associated with the viral load dynamics. More precisely, the last estimations of b_{0i} and ω_i (which are outputs of the algorithm) are subtracted from the output variable Y_i, for all i (as in step 1 of Algorithm 1) to come back to a classical regression framework (i.e. with independent observations). Hence, the Variable Selection Using Random Forests method from Genuer et al.[39] can be apply by using the VSURF package.[40]

This method is a fully automatic variable selection procedure based on RF and designed to deal with high-dimensional data in a regression framework as well as in supervised classification. It works in three steps: (i) first, the variables are sorted in decreasing order of RF VI, then a data-driven threshold is computed to eliminate variables with low VI; (ii) variables left are then introduced (one by one according to the previous order) in nested RF models and the one minimizing the OOB error is selected; and (iii) a refined ascending sequence of RF models (obtained in a stepwise way) is then built and finally the last model of this sequence is returned.

5.2 Stability of the selected variables set

We illustrate the stability of the selected variables set by introducing a stability score and studying the behavior of this score against the RF parameter mtry.

Let $\mathcal{V} = \{V_{(1)}, \ldots, V_{(p)}\}$ and $\mathcal{V}' = \{V_{(1)}', \ldots, V_{(p)}'\}$ be the decreasing ordered variables, respectively, to the VI obtained with two runs of the **SREEMforest** method. Due to the randomness aspect of the RF, **SREEMforest** is random and the sequences $\mathcal{V}$ and $\mathcal{V}'$ may be different. Hence, we introduce a stability score $\mathcal{SS}$ which measures the difference between two ordered sequence $\mathcal{V}$ and $\mathcal{V}'$

$$\mathcal{SS}^{\eta}(\mathcal{V}, \mathcal{V}') = \frac{1}{p}\sum_{i=1}^{p} \mathbb{1}_{\left\{V_{(i)} \in \mathcal{B}\left(V_{(i)}';\eta\right)\right\}}$$

with $\mathcal{B}\left(V_{(i)}';\eta\right) = \left\{V'_{(i-\eta)_+}, \ldots, V'_{(i+\eta)_-}\right\}$ where

$$V'_{(i-\eta)_+} = \begin{cases} V'_{(1)} & \text{if } i-\eta \leq 0 \\ V'_{(i-\eta)} & \text{else} \end{cases} \quad \text{and} \quad V'_{(i+\eta)_-} = \begin{cases} V_{(p)} & \text{if } i+\eta \geq p \\ V'_{(i+\eta)} & \text{else} \end{cases}$$

This score measures the proportion of variables ranked in a same neighborhood (η handles the size of the neighborhood). To stabilize the results, the score was computed with 30 pairs of sequences $\mathcal{V}$ and $\mathcal{V}'$ and the mean of the obtained stability scores was provided.

The computation of these stability scores was restricted to the 50 most important variables given by different runs of **SREEMforest** applied to the DALIA vaccine trial dataset. In Figure 11, we note that, except for mtry set

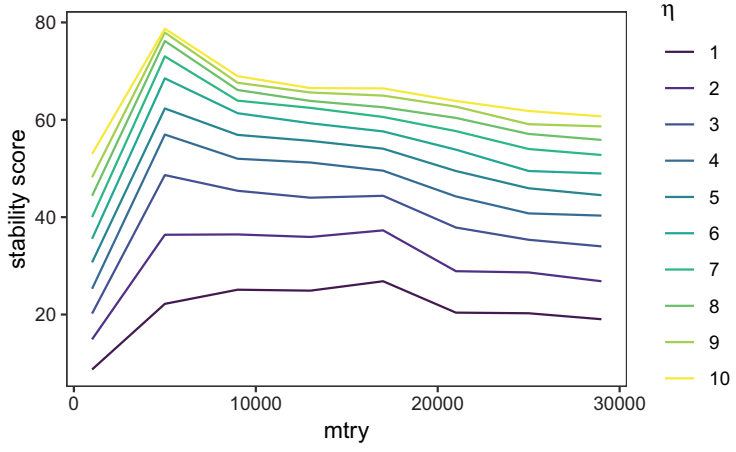

Figure 11. Evolution of the mean stability score against the mtry parameter and the neighborhood size (η), restricted to the 50 most important variables, for the **SREEMforest** method, DALIA trial.

to 1000, we obtained a stability score around 0.5 for a neighborhood size of 4. This means that for two lists of the 50 most important variables obtained with **SREEMforest**, approximately 50% of them were at the same rank (± 4 ranks). For a neighborhood size larger than 8, the score can exceed 75%. In conclusion, for a wide range of mtry values, variable ranking results were quite consistent.

5.3 Biological results

The 21 variables selected by VSURF (applied after convergence of **SREEMforest**) were mainly transcripts (OAS, LY6E, HERC5, IFI/IFIT, EPSTI1, MX1, RSAD2, EIF2AK2, and XAF1) associated with the interferon-α pathway. For instance, they all belongs to the Chaussabel's modules M1.2 and M3.4 annotated "Interferon".[41] Interferon pathway is highly correlated to the viral replication as demonstrated previously.[42] Only, two transcripts were not associated with the interferon pathway (EPSTI1 and SAMD9L). The commitment of the interferon pathway reflects the immune response to viral infection. The relevance of these results is another argument for the validation of the proposed approach.

6 Discussion

In this article, we introduced a new RF approach suited for the analysis of high-dimensional longitudinal data. We also generalized existing methods so they can be applied in the stochastic semi-parametric mixed effects model. The simulation study revealed the superiority of both our approach and these generalizations. The proposed method has also been applied to a complex dataset coming from an HIV vaccine trial, illustrating its effectiveness and interest in such high-dimensional longitudinal context.

An important aspect highlighted by our study is the choice of the mtry parameter. Our advice is to choose a large value for mtry—roughly between $2p/3$ and $3p/4$—, not smaller than $p/2$. Indeed, as we are in an (very) high-dimensional context, the number of variables selected at each node of trees must not be too small—preventing to choose only non-informative variables too often. Second, since the different approaches are based on an EM algorithm, a too small value for mtry could lead to the non-convergence of the method and hence to very sub-optimal results, as illustrated by Figure 9. In addition, even if an automatic choice of mtry would obviously be appealing for users, it seems rather difficult to include it, because of the already quite high execution times of the proposed method.

Another key point about these approaches is the choice of the model, and more particularly the choice of the random effects. Driven by our application, we only use a random intercept (in addition to the stochastic process) regarding the number of individuals and the number of time points we had in the vaccine trial. However, in a context with more individuals and/or less time points, it could be interesting to add random effects on the different time points. This should make the model more flexible and hence increase the method capacity to estimate the inter-individual variability.

Following the work of Fu and Simonoff,[24] one could also study the effect of the use of conditional inference trees[25] instead of CART in (S)REEMforest. This did not appear mandatory in our particular framework where all explanatory variables are continuous, but this could be addressed in the more general case where both continuous and categorical (with different numbers of categories) variables are available.

Finally, the theoretical analysis of such complex methods (non-parametric estimates plugged into an EM algorithm) seems rather difficult and remains, to the extent of our knowledge, an open issue.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iD

Louis Capitaine https://orcid.org/0000-0001-6800-2342

Note

a. Available at https://github.com/Lcapitaine/longituRF.

References

1. Breiman L. Random forests. *Mach Learn* 2001; **45**: 5–32.
2. Fernández-Delgado M, Cernadas E, Barro S, et al. Do we need hundreds of classifiers to solve real world classification problems? *J Mach Learn Res* 2014; **15**: 3133–3181.
3. Cutler DR, Edwards TC, Beard KH, et al. Random forests for classification in ecology. *Ecology* 2007; **88**: 2783–2792.
4. Chen X and Ishwaran H. Random forests for genomic data analysis. *Genomics* 2012; **99**: 323–329.
5. Scornet E, Biau G and Vert JP. Consistency of random forests. *Ann Stat* 2015; **43**: 1716–1741.
6. Mentch L and Hooker G. Quantifying uncertainty in random forests via confidence intervals and hypothesis tests. *J Mach Learn Res* 2016; **17**: 841–881.
7. Wager S. Asymptotic theory for random forests. *arXiv preprint arXiv:14050352* 2014.
8. Biau G and Scornet E. A random forest guided tour. *Test* 2016; **25**: 197–227.
9. Genuer R, Poggi JM and Tuleau C. Random forests: some methodological insights. *arXiv preprint arXiv:08113619* 2008.
10. Zhu R, Zeng D and Kosorok MR. Reinforcement learning trees. *J Am Stat Assoc* 2015; **110**: 1770–1784.
11. Linero AR. Bayesian regression trees for high-dimensional prediction and variable selection. *J Am Stat Assoc* 2018; **113**: 626–636.
12. Chipman HA, George EI and McCulloch RE. Bayesian cart model search. *J Am Stat Assoc* 1998; **93**: 935–948.
13. Hothorn T, Bühlmann P, Dudoit S, et al. Survival ensembles. *Biostatistics* 2005; **7**: 355–373.
14. Ishwaran H, Kogalur UB, Blackstone EH, et al. Random survival forests. *Ann Appl Stat* 2008; **2**: 841–860.
15. Ishwaran H, Kogalur UB, Gorodeski EZ, et al. High-dimensional variable selection for survival data. *J Am Stat Assoc* 2010; **105**: 205–217.
16. Steingrimsson JA, Diao L and Strawderman RL. Censoring unbiased regression trees and ensembles. *J Am Stat Assoc* 2019; **114**: 370–383.
17. Laird NM and Ware JH. Random-effects models for longitudinal data. *Biometrics* 1982; 963–974.
18. Verbeke G and Molenberghs G. *Linear mixed models for longitudinal data*. New York: Springer, 2009.
19. Segal MR. Tree-structured methods for longitudinal data. *J Am Stat Assoc* 1992; **87**: 407–418.
20. Eo SH and Cho H. Tree-structured mixed-effects regression modeling for longitudinal data. *J Comput Graph Stat* 2014; **23**: 740–760.
21. Wei Y, Liu L, Su X et al. Precision medicine: Subgroup identification in longitudinal trajectories. *Statistical Methods in Medical Research, 29*(9), 2603–2616.
22. Hajjem A, Bellavance F and Larocque D. Mixed effects regression trees for clustered data. *Stat Probab Lett* 2011; **81**: 451–459.
23. Sela RJ and Simonoff JS. RE-EM trees: a data mining approach for longitudinal and clustered data. *Mach Learn* 2012; **86**: 169–207.
24. Fu W and Simonoff JS. Unbiased regression trees for longitudinal and clustered data. *Comput Stat Data Anal* 2015; **88**: 53–74.
25. Hothorn T, Hornik K and Zeileis A. Unbiased recursive partitioning: a conditional inference framework. *J Comput Graph Stat* 2006; **15**: 651–674.
26. Breiman L, Friedman JH, Olshen RA, et al. *Classification and regression trees*. London: Chapman & Hall, 1984.
27. Hajjem A, Bellavance F and Larocque D. Mixed-effects random forest for clustered data. *J Stat Comput Simul* 2014; **84**: 1313–1328.
28. McLachlan GJ and Krishnan T. *The EM algorithm and extensions*. Hoboken, NJ: John Wiley & Sons, 1997.
29. Kundu MG and Harezlak J. Regression trees for longitudinal data with baseline covariates. *Biostat Epidemiol* 2019; **3**: 1–22.
30. Calhoun P, Levine RA and Fan J. Repeated measures random forests (RMRF): Identifying factors associated with nocturnal hypoglycemia. *Biometrics*. Epub ahead of print 20 April 2020. DOI: 10.1111/biom.13284
31. R Core Team. *R: a language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing, 2019.
32. Diggle PJ and Hutchinson MF. On spline smoothing with autocorrelated errors. *Aust N Z J Stat* 1989; **31**: 166–182.
33. Zhang D, Lin X, Raz J, et al. Semiparametric stochastic mixed models for longitudinal data. *J Am Stat Assoc* 1998; **93**: 710–719.
34. Wu H and Zhang JT. *Nonparametric regression methods for longitudinal data analysis: mixed-effects modeling approaches*. Hoboken, NJ: John Wiley & Sons, 2006.
35. Díaz-Uriarte R and Alvarez De Andres S. Gene selection and classification of microarray data using random forest. *BMC Bioinf* 2006; **7**: 3.

36. Hejblum BP, Skinner J and Thiébaut R. Time-course gene set analysis for longitudinal gene expression data. *PLoS Comput Biol* 2015; **11**: e1004310.
37. Verikas A, Gelzinis A and Bacauskiene M. Mining data with random forests: a survey and results of new tests. *Pattern Recognit* 2011; **44**: 330–349.
38. Lévy Y, Thiébaut R, Montes M, et al. Dendritic cell-based therapeutic vaccine elicits polyfunctional HIV-specific T-cell immunity associated with control of viral load. *Eur J Immunol* 2014; **44**: 2802–2810.
39. Genuer R, Poggi JM and Tuleau-Malot C. Variable selection using random forests. *Pattern Recognit Lett* 2010; **31**: 2225–2236.
40. Genuer R, Poggi JM and Tuleau-Malot C. VSURF: an R package for variable selection using random forests. *R J* 2015; **7**: 19–33.
41. Chaussabel D and Baldwin N. Democratizing systems immunology with modular transcriptional repertoire analyses. *Nat Rev Immunol* 2014; **14**: 271.
42. Bosinger SE, Li Q, Gordon SN, et al. Global genomic analysis reveals rapid control of a robust innate response in SIV-infected sooty mangabeys. *J Clin Invest* 2009; **119**: 3556–3572.