Comput Stat (2017) 32:409–428
DOI 10.1007/s00180-016-0707-x

ORIGINAL PAPER

Unsupervised learning of pharmacokinetic responses

Elson Tomás[1] · **Susana Vinga**[1] · **Alexandra M. Carvalho**[2]

Received: 17 December 2015 / Accepted: 15 December 2016 / Published online: 17 January 2017

Abstract Pharmacokinetics (PK) is a branch of pharmacology dedicated to the study of the time course of drug concentrations, from absorption to excretion from the body. PK dynamic models are often based on homogeneous, multi-compartment assumptions, which allow to identify the PK parameters and further predict the time evolution of drug concentration for a given subject. One key characteristic of these time series is their high variability among patients, which may hamper their correct stratification. In the present work, we address this variability by estimating the PK parameters and simultaneously clustering the corresponding subjects using the time series. We propose an expectation maximization algorithm that clusters subjects based on their PK drug responses, in an unsupervised way, collapsing clusters that are closer than a given threshold. Experimental results show that the proposed algorithm converges fast and leads to meaningful results in synthetic and real scenarios.

Keywords Clustering · Expectation-maximization · One-compartment model

✉ Alexandra M. Carvalho
alexandra.carvalho@tecnico.ulisboa.pt

Susana Vinga
susanavinga@tecnico.ulisboa.pt

[1] IDMEC, Instituto Superior Técnico, Universidade de Lisboa, Av. Rovisco Pais 1, 1049-001 Lisbon, Portugal

[2] Instituto de Telecomunicações, Instituto Superior Técnico, Universidade de Lisboa, Av. Rovisco Pais 1, 1049-001 Lisbon, Portugal

1 Introduction

A model commonly used in *pharmacokinetics* (PK) is the *compartment model* (Dayneka et al. 1993; Derendorf et al. 2000; Mager et al. 2003; Gueorguieva et al. 2007) which is used to describe the time course of drug concentrations in the body. Compartment models define a family of drug concentration functions depicting the variability of PK drug responses among subjects. They are categorized according to the number of compartments needed to describe the behavior of the drug. Each compartment can represent a group of similar tissues, an organ or a fluid; drug concentration is measured in the blood, plasma, urine, saliva, and other easily sampled fluids.

Literature in population PK is vast and typically aims at performing drug development and therapeutic drug monitoring. Compartment models are fitted to the data mostly using *nonlinear mixed effects models* (NLMEM) (Beal and Sheiner 1980; Sheiner et al. 1977; Davidian and Giltinan 2003). NLMEM include *fixed effects* associated with the entire population and *random effects* which are subject/group-specific. In these models at least one of the fixed or random effects appears nonlinearly in the model function.

The usual approach is to integrate out the random effects to allow maximum likelihood estimation. Unfortunately, this cannot be done analytically due to random effects not being linear. This problem cannot be circumvented by standard *Expectation-Maximization* (EM) because of the nonlinear structure of the model (Dempster et al. 1977; Lindstrom and Bates 1988. To overcome these shortcomings, *Monte Carlo EM* techniques have been proposed where the E-step can only be approximated with empirical averages (Wei and Tanners 1991; Walker 1996; Wu 2002, 2004. Additional approaches include *stochastic approximation EM* where the E-step is approximated by a weighted average (Delyon et al. 1999; Kuhn and Lavielle 2005). MONOLIX is a software package (Trout et al. 2004) that incorporates several of these techniques.

The aforementioned methods do not cluster subjects using PK responses. They only estimate the parameters for a single cluster describing the average population effects. Random effects that are subject/group-specific are not learned. Grouping subjects in clusters according to their PK profiles is desirable in order to design tailored therapies for each group. The advantage of personalized therapies is that the administered dose, as well as intervals of administration, can be tuned in order to keep the correct concentration of the drug in the body with the smallest side-effects. This is the key objective of this paper, to cluster subjects according to their PK drug response parameters.

To this end, we consider an one-compartment model where drug concentration in the body compartment is described by a function over time whose parameters are group-dependent. We provide a novel unsupervised learning method for clustering PK drug responses. The proposed algorithm is an EM method and a special case of that devised by Azzimonti et al. (2013). The latter can be applied to any nonlinear function, allowing to include both fixed and random effects with Gaussian noise to the measurements. In our case, we omit fixed effects, but we consider a cluster-specific error variance aiming to address PK drug responses. Indeed, based on empirical evidence, each PK profile seems to have its own variance.

We assess the merits of the proposed method against synthetic data, by considering several datasets where clusters of PK drug responses were simulated. For all datasets

considered, the algorithm was able to recover the original clusters, and moreover, it performed linearly on the number of subjects and the number of initial clusters. In addition, the algorithm showed to be insensitive to imbalanced data, specially when clusters are well defined. Real data from theophylline pharmacokinetics was also used to assess the algorithm; three meaningful clusters of PK drug responses were retrieved.

This paper is organized as follows. Section 2 describes the one-compartment model. In Sect. 3 the novel EM algorithm to cluster PK drug responses is presented. Experimental results are presented next, followed by some conclusions and future work. In the end, we provide two appendixes with details concerning the implementation of the proposed method.

2 One-compartment model

The simplest PK drug response model is the one-compartment model. Despite being over-simplistic, one-compartment models are the most frequently used in clinical practice.

In the one-compartment model the drug enters the compartment with an initial *Dose*. Drug concentration can then be monitored by continuously measuring the concentration of the drug in the compartment. With this repeated measurements a curve of drug concentration against time, $C(t)$, can be plotted. Note that

$$C(t) = \frac{Q(t)}{V},$$

where $Q(t)$ is the amount of drug in the compartment and V the volume of the compartment.

If the drug is administered orally, it has a gradual absorption, reaching maximum concentrations later when compared with intravenous administration. Therefore, k_a is defined as the *absorption rate constant*. A subcategory of absorption is the *bioavailability*, denoted by F, accounting for the fraction of unchanged drug that effectively reaches the compartment. The process of gradual absorption, called *sustained release* (Lee and Amidon 1996), is described by a function $I(t)$ which follows a first-order kinetics, described by the constant k_a, with initial condition defined by the Dose and F. Concretely, $I'(t) = -k_a I(t)$ and $I(0) = Dose \times F$. Similarly, the rate at which a drug is removed from the compartment is also given by k_e, called *the elimination rate constant*. A figure depicting this model is presented in Fig. 1.

By putting everything together, we obtain

$$Q'(t) = -k_e Q(t) + k_a I(t).$$

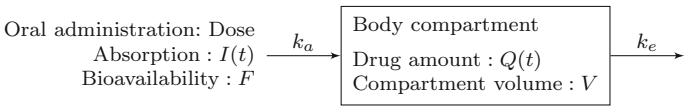

Fig. 1 Scheme of one-compartment model for sustained release with first-order kinetics

Solving the previous system of equations, the concentration of the drug in the compartment over time is given by:

$$C(t) = \alpha(e^{-\beta_1 t} - e^{-\beta_2 t}), \tag{1}$$

where

$$\alpha = \frac{k_a \text{Dose} \times F}{V(k_a - k_e)},$$
$$\beta_1 = k_e \text{ and } \beta_2 = k_a.$$

3 Unsupervised learning of pharmacokinetic responses

In this section, we propose an unsupervised learning algorithm to cluster subjects based on their PK drug responses. More specifically, we draw up an EM algorithm that estimates the parameters of the curves modelling the responses. Before presenting the EM algorithm we introduce notation and the stochastic assumptions needed to derive the algorithm.

We assume that subjects split among M clusters and that, for each cluster $\ell \in \{1, \ldots, M\}$, the drug concentration evolves over time as described by Eq. (1), that is,

$$C_\ell(t) = f(\alpha_\ell, \beta_{1\ell}, \beta_{2\ell}, t) = \alpha_\ell(e^{-\beta_{1\ell} t} - e^{-\beta_{2\ell} t}). \tag{2}$$

We denote by N the number of subjects and n the number of measurements for each subject. Motivated by empirical evidence from PK drug response, we assume that each cluster has a specific error variance, which can be due, for instance, to genetics (Roden and George 2002) or interaction with other drugs (Lee et al. 2014). We consider a Gaussian distribution to model this cluster-specific error. Concretely, we consider

$$y_{ij} = C_\ell(t_j) + \epsilon_{ij\ell}, \quad i = 1, ..., N \text{ and } j = 1, \ldots, n,$$

to be the observed drug concentration at instant t_j for the i-th subject in the ℓ-th cluster, where $\epsilon_{ij\ell} \sim N(0, v_\ell)$ is the cluster-specific error. Given that $\epsilon_{ij\ell} = y_{ij} - C_\ell(t_j)$, the probability density function for the observed drug concentration is given by:

$$p_\ell(y_{ij}) = \frac{1}{(2\pi v_\ell)^{\frac{1}{2}}} e^{\frac{-1}{2v_\ell}(y_{ij} - C_\ell(t_j))^2}.$$

In addition, we assume that errors are independent, and so the probability density function of measuring $\mathbf{y}_i = (y_{i1}, \ldots, y_{in})$ for the i-th subject in the ℓ-th cluster is given by:

$$p_\ell(\mathbf{y_i}) = \frac{1}{(2\pi v_\ell)^{\frac{n}{2}}} e^{\frac{-1}{2v_\ell}\sum_{j=1}^{n}(y_{ij} - C_\ell(t_j))^2}. \tag{3}$$

Finally, we assume that each subject belongs to some cluster ℓ. The cluster a subject belongs to is unknown a priori, and so, we consider the random vector $\mathbf{W} = (W_1, \ldots, W_N)$ where each random variable W_i describes the cluster to which subject i belongs. In this case, we assume that

$$P(W_i = \ell) = \omega_\ell \text{ for } 1 \leq \ell \leq M,$$

where each ω_ℓ is called the ℓ-*weight* and amounts to the probability of the subjects belonging to the ℓ-th cluster.

3.1 EM algorithm

Under the above assumptions, we now set out to derive the EM algorithm. The algorithm estimates, for each cluster $\ell \in \{1, \ldots, M\}$, the parameters α_ℓ, $\beta_{1\ell}$ and $\beta_{2\ell}$, that best fit the data. It also elicits the cluster each subject belongs to, as well as the variance of the error for each cluster. As input, the algorithm receives a matrix

$$Y = \begin{pmatrix} y_{11} & \cdots & y_{1n} \\ \vdots & \ddots & \vdots \\ y_{N1} & \cdots & y_{Nn} \end{pmatrix}$$

and a vector $\mathbf{t} = (t_1, \ldots t_n)$, where y_{ij} is the observed drug concentration at instant t_j for the i-th subject, since drug ingestion. For the sake of simplicity we assume that $\mathbf{t} = (t_1, \ldots t_n)$ is the same for all subjects, that is, sampling times and number of samples are the same for all subjects. However, it is straightforward to adapt the proposed algorithm to different sampling times and different number of samples for each subject.

The algorithm aims at estimating the parameters by minimizing the error $\epsilon_{ij\ell}$ according to the inputs, which amounts to maximize the likelihood of the data. Given the parameters $\boldsymbol{\theta} = \{\alpha_\ell, \beta_{1\ell}, \beta_{2\ell}, v_\ell, \omega_\ell\}_{\ell \in 1,\ldots,M}$ and $\mathbf{W} = \mathbf{w}$, where $\mathbf{w} = (\ell_1, \ldots, \ell_N)$ and ℓ_i is the cluster to which the i-th subject belongs, the probability of observing data Y and $\mathbf{w}$ is given by

$$\begin{aligned} p_{\boldsymbol{\theta}}(Y, \mathbf{w}) &= \prod_{i=1}^{N} P(W_i = \ell_i)\, p_{\boldsymbol{\theta}}(\mathbf{y}_i \mid W_i = \ell_i) \\ &= \prod_{i=1}^{N} \omega_{\ell_i}\, p_\ell(\mathbf{y}_i), \end{aligned}$$

where, for the sake of notation, we drop the parameters $\boldsymbol{\theta}$ in p_ℓ. Recall that $p_\ell(\mathbf{y}_i)$ is as given in Eq. (3).

The EM algorithm is an iterative method, where the parameters $\boldsymbol{\theta}$ are iteratively refined until convergence. Thus, we denote by

$$\boldsymbol{\theta}^{(k)} = \{\alpha_\ell^{(k)}, \beta_{1\ell}^{(k)}, \beta_{2\ell}^{(k)}, v_\ell^{(k)}, \omega_\ell^{(k)}\}_{\ell \in 1,\ldots,M}$$

the parameters at iteration k of the EM algorithm. As usual, the EM algorithm is going to have multiple random restarts for the initial parameters $\boldsymbol{\theta}^{(0)}$, from which it is elicited the clustering with highest likelihood.

The EM algorithm consists of two steps, an *expectation step* (E-step) and a *maximization step* (M-step). We devote the remainder of this section to describe these steps.

3.1.1 E-step

In this step we compute the objective function Q defined as

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)}) &= E_{p_{\boldsymbol{\theta}^{(k)}}(\mathbf{W}|Y)}[\log(p_{\boldsymbol{\theta}}(Y, \mathbf{W})] \\ &= \sum_{\ell=1}^{M} \sum_{i=1}^{N} X_{i\ell}^{(k)} \log(\omega_\ell p_\ell(\mathbf{y}_i)), \end{aligned}$$

where

$$X_{i\ell}^{(k)} = \frac{\omega_\ell^{(k)} p_\ell^{(k)}(\mathbf{y}_i)}{\sum_{m=1}^{M} \omega_m^{(k)} p_m^{(k)}(\mathbf{y}_i)}.$$

Note that the expected value is taken over $p_{\boldsymbol{\theta}^{(k)}}(\mathbf{W} \mid Y)$ and that Q is a function of $\boldsymbol{\theta}$, given the present estimates of the parameters $\boldsymbol{\theta}^{(k)}$. It is well known that maximizing Q corresponds to maximizing the likelihood of Y.

3.1.2 M-step

In this step we find the parameters $\boldsymbol{\theta}$ that maximize $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)})$. Next, we present the expressions to update the parameters α_ℓ, $\beta_{1\ell}$, $\beta_{2\ell}$, v_ℓ and ω_ℓ for iteration $k+1$, given their current values at iteration k; recall that the parameters α_ℓ, $\beta_{1\ell}$ and $\beta_{2\ell}$ define the mean concentration for the ℓ-th cluster, given by $C_\ell(t)$, and so, their update is jointly presented.

Update of ω_ℓ. The update of ω_ℓ is similar to the canonical EM algorithm for mixtures of Gaussians and for that reason we omit its derivation; the update is given by:

$$\omega_\ell^{(k+1)} = \frac{1}{N} \sum_{i=1}^{N} X_{i\ell}^{(k)} \quad \text{for all } \ell = 1, \ldots, M.$$

Update of $C_\ell(t)$. The update of $C_\ell(t)$ consists in updating α_ℓ, $\beta_{1\ell}$ and $\beta_{2\ell}$. Unfortunately, it is not straightforward to analytically find these parameters, as this corresponds to solve a system of transcendental equations.

For this reason, we need to rely on numerical methods, where a possibility is to use *Newton's method* for the multidimensional case. This method allows to find successively better approximations to the roots (or zeros) of a real-valued function. In our specific case, we want to find the parameters $\boldsymbol{\theta}$ that maximize the objective function $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)})$ and those correspond to finding the roots on the partial derivatives of $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)})$. However, for large-scale data, Netwon's method becomes unfeasible as it requires to compute the inverse of the Hessian matrix at each iteration of the EM algorithm. On one hand, the determinant of the matrix might be close to zero, which will make the method unstable; on the other hand, the method is quite space and time demanding.

To overcome this difficulty, we followed the canon of the literature and used a coordinate descent method (Wright 2015; Nesterov 2012). In this method, we greedily search the maximum of each parameter by fixing the others on their previous values. For this simplified approach we are able to find $\alpha_\ell^{(k+1)}$ analytically while $\beta_{1\ell}^{(k+1)}$ and $\beta_{2\ell}^{(k+1)}$ are obtained using the Newton's method for the univariate case. The experimental results show that this simplification does not prevent our method from reaching good performance.

To update α_ℓ we have to solve $\frac{\partial Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)})}{\partial \alpha_\ell} = 0$. The solution for this equation is

$$\alpha_\ell = \frac{\sum_{i=1}^{N} \sum_{j=1}^{n} X_{i\ell}^{(k)} y_{ij} (e^{-\beta_{1\ell} t_j} - e^{-\beta_{2\ell} t_j})}{\sum_{i=1}^{N} \sum_{j=1}^{n} X_{i\ell}^{(k)} (e^{-\beta_{1\ell} t_j} - e^{-\beta_{2\ell} t_j})^2}$$

and, as mentioned before, we shall use $\beta_{1\ell}^{(k)}$ and $\beta_{2\ell}^{(k)}$, instead of the maxima for $\beta_{1\ell}$ and $\beta_{2\ell}$, to compute the update of α_ℓ, which leads to

$$\alpha_\ell^{(k+1)} = \frac{\sum_{i=1}^{N} \sum_{j=1}^{n} X_{i\ell}^{(k)} y_{ij} (e^{-\beta_{1\ell}^{(k)} t_j} - e^{-\beta_{2\ell}^{(k)} t_j})}{\sum_{i=1}^{N} \sum_{j=1}^{n} X_{i\ell}^{(k)} (e^{-\beta_{1\ell}^{(k)} t_j} - e^{-\beta_{2\ell}^{(k)} t_j})^2}.$$

It is straightforward to see that the above value is indeed a maximum, by computing the second derivative, and checking that it is negative in $\alpha_\ell^{(k+1)}$.

To update $\beta_{1\ell}$, we have to find when the partial derivative of $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)})$ in $\beta_{1\ell}$ takes the value zero, that is, $\frac{\partial Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)})}{\partial \beta_{1\ell}} = 0$, which corresponds to solving

$$\sum_{i=1}^{N} \sum_{j=1}^{n} \Big(- \frac{\alpha_\ell}{v_\ell} X_{i\ell}^{(k)} t_j e^{-\beta_{1\ell} t_j} \left(y_{ij} - \alpha_\ell (e^{-\beta_{1\ell} t_j} - e^{-\beta_{2\ell} t_j}) \right) \Big) = 0. \tag{4}$$

For this purpose, let $h_{1\ell}^{(k)}(\beta_{1\ell})$ be defined as

$$\sum_{i=1}^{N}\sum_{j=1}^{n}\Big(-\frac{\alpha_\ell^{(k+1)}}{v_\ell^{(k)}}X_{i\ell}^{(k)}t_j e^{-\beta_{1\ell}t_j}\left(y_{ij}-\alpha_\ell^{(k+1)}(e^{-\beta_{1\ell}t_j}-e^{-\beta_{2\ell}^{(k)}t_j})\right)\Big).$$

This function is obtained by replacing in the lefthand side of Eq. (4) the variable α_ℓ by $\alpha_\ell^{(k+1)}$ and $\beta_{2\ell}$ by $\beta_{2\ell}^{(k)}$. Since $h_{1\ell}^{(k)}$ is continuous and differentiable for $\beta_{1\ell} > 0$, we can apply Newton's method, and moreover, it is easy to see that it converges to a maximum, as the second derivative, analytically computed, is negative in the convergence point. Therefore,

$$\beta_{1\ell}^{(k+1)} = \text{Newton}(\beta_{1\ell}^{(k)}, h_{1\ell}^{(k)}),$$

where $\text{Newton}(x, h)$ is the output of the Newton's method for function h with starting point x.

Finally, to update $\beta_{2\ell}$, we also have to find when the partial derivative of $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)})$ in $\beta_{2\ell}$ takes the value zero, that is, $\frac{\partial Q(\boldsymbol{\theta},\boldsymbol{\theta}^{(k)})}{\partial\beta_{2\ell}} = 0$, which corresponds to solving

$$\sum_{i=1}^{N}\sum_{j=1}^{n}\Big(\frac{\alpha_\ell}{v_\ell}X_{i\ell}^{(k)}t_j e^{-\beta_{2\ell}t_j}\left(y_{ij}-\alpha_\ell(e^{-\beta_{1\ell}t_j}-e^{-\beta_{2\ell}t_j})\right)\Big) = 0. \tag{5}$$

Similarly to the previous update, let $h_{2\ell}^{(k)}(\beta_{2\ell})$ be defined as

$$\sum_{i=1}^{N}\sum_{j=1}^{n}\Big(\frac{\alpha_\ell^{(k+1)}}{v_\ell^{(k)}}X_{i\ell}^{(k)}t_j e^{-\beta_{2\ell}t_j}\left(y_{ij}-\alpha_\ell^{(k+1)}(e^{-\beta_{1\ell}^{(k+1)}t_j}-e^{-\beta_{2\ell}t_j})\right)\Big).$$

Observe that this function is obtained by replacing in the lefthand side of Eq. (5) the variable α_ℓ by $\alpha_\ell^{(k+1)}$ and $\beta_{1\ell}$ by $\beta_{1\ell}^{(k+1)}$. Once again, since $h_{2\ell}^{(k)}$ is continuous and differentiable for $\beta_{2\ell} > 0$, we can apply Newton's method that will converge to a maximum, as the second derivative is negative in the convergence point. Thus,

$$\beta_{2\ell}^{(k+1)} = \text{Newton}(\beta_{2\ell}^{(k)}, h_{2\ell}^{(k)}).$$

To sum up, $C_\ell(t)$ is updated as:

$$C_\ell^{(k+1)}(t) = \alpha_\ell^{(k+1)}(e^{-\beta_{1\ell}^{(k+1)}t} - e^{\beta_{2\ell}^{(k+1)}t}).$$

Update of v_ℓ. To update v_ℓ we need to have the mean value of the concentration for cluster ℓ at time t_j, which is given by $C_\ell^{(k+1)}(t_j)$. Given this, the update of v_ℓ is similar to the canonical EM algorithm for mixtures of Gaussians, given by:

$$v_\ell^{(k+1)} = \frac{\sum_{i=1}^{N} \sum_{j=1}^{n} X_{i\ell}^{(k)} (y_{ij} - C_\ell^{(k+1)}(t_j))^2}{\sum_{i=1}^{N} n X_{i\ell}^{(k)}}.$$

3.2 Collapsing clusters

The EM algorithm starts with an initial number of clusters M and provides a weight for each of these clusters. After EM convergence, clusters are analyzed to check whether they are negligible or overlap to some extent. If a cluster ℓ is *negligible*, that is, ω_ℓ is below some user-defined threshold $\overline{W}$, then the ℓ-th cluster is *disregarded*, being M decremented accordingly. The subjects that were grouped in the ℓ-th cluster are randomly distributed by the remaining clusters. The weights of the clusters are recomputed correspondingly.

In addition, if two clusters ℓ_1 and ℓ_2 *overlap*, that is,

$$\sum_{j=1}^{n} \frac{(C_{\ell_1}(t_j) - C_{\ell_2}(t_j))^2}{n} < \overline{L},$$

where $\overline{L}$ is an user-defined threshold, then ℓ_1 and ℓ_2 are *merged*. Without loss of generality, assume that cluster ℓ_2 is disregarded and all subjects now belong to cluster ℓ_1 whose weight becomes $\omega_{\ell_1} + \omega_{\ell_2}$. The parameters of cluster ℓ_1 are updated with the weighted average of the parameters of both clusters, that is,

$$\begin{aligned}
\alpha_{\ell_1} &= \frac{\omega_{\ell_1}\alpha_{\ell_1} + \omega_{\ell_2}\alpha_{\ell_2}}{\omega_{\ell_1} + \omega_{\ell_2}}, \\
\beta_{1\ell_1} &= \frac{\omega_{\ell_1}\beta_{1\ell_1} + \omega_{\ell_2}\beta_{1\ell_1}}{\omega_{\ell_1} + \omega_{\ell_2}}, \\
\beta_{2\ell_1} &= \frac{\omega_{\ell_1}\beta_{2\ell_1} + \omega_{\ell_2}\beta_{2\ell_2}}{\omega_{\ell_1} + \omega_{\ell_2}}, \text{ and} \\
v_{\ell_1} &= \frac{\omega_{\ell_1} v_{\ell_1} + \omega_{\ell_2} v_{\ell_2}}{\omega_{\ell_1} + \omega_{\ell_2}}.
\end{aligned}$$

If at least one cluster is disregarded, the EM algorithm restarts with the new parameters as initial values for the method. The overall algorithm stops when EM converges and there is no cluster to disregard according to the previous criteria.

4 Experimental results

We implemented the proposed algorithm in Java; the implementation is available at https://asmcarvalho.github.io/EMPK/, together with data used in the experiments. The details about the coordinate descent method used to update the parameters $\beta_{1\ell}$ and $\beta_{2\ell}$ in each EM iteration are provided in Appendix A. Moreover, in Appendix B we provide further implementation details about the proposed EM algorithm.

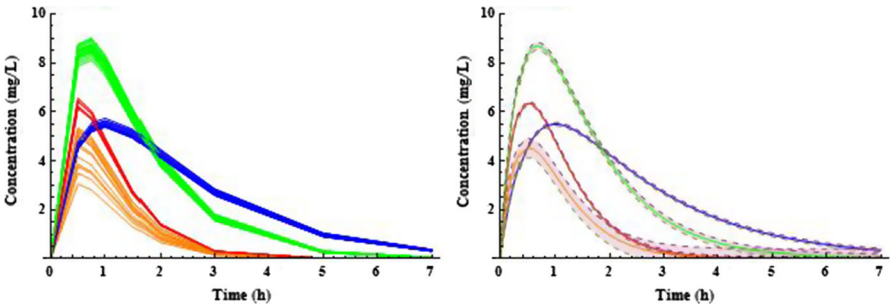

Fig. 2 On the *left* it is presented the dataset of experiment (i). The four clusters elicited by the proposed EM algorithm are depicted in *different colors*; each *line* represents a subject. The algorithm was initialized with ten clusters and was able to recover the original four, where two are very similar (*orange* and *red*). On the right it is depicted the clusters found together with their standard deviation, that is, $C_\ell(t) \pm \sigma_\ell$ (color figure online)

First, we present experimental results in synthetic data. Afterwards, we address real data with the analysis of theophylline pharmacokinetics.

4.1 Synthetic data

In this section we evaluated the merits of the proposed algorithm with synthetic data. Firstly, we considered cluster recovery and homogeneity. Then, we tested the clustering procedure with imbalanced data and empirically studied the running time of the algorithm.

4.1.1 Cluster recovering

We performed two tests with the following (balanced) data: (i) a dataset with 60 subjects ($N = 60$), nine measurements over time for each subject ($n = 9$), low within-group variance, and four clusters ($M = 4$); (ii) a dataset with 100 subjects ($N = 100$), nine measurements over time for each subject ($n = 9$), high within-group variance, and four non-overlapping clusters ($M = 4$). In this setup low within-group variance means that the standard deviation of each cluster, although different in all of them, is lower than 0.6; on the other hand, high within-group variance means a standard deviation bigger than 1.2.

For each dataset, we ran the algorithm with 100 random initializations (to avoid local maxima in the EM procedure) and ten initial clusters; the result with the highest likelihood was chosen. The best result for experiments (i) and (ii) is presented in Figs. 2 and 3, respectively.

We conclude that the algorithm was able to completely recover the original four clusters, even with two clusters having low between-group variance (orange and red curves in Fig. 2, and blue and green curves in Fig. 3).

4.1.2 Cluster homogeneity

We also performed nine additional experiments, with different number of clusters (M), summarized in Table 1. To assess the homogeneity of the clusters the *Sum of*

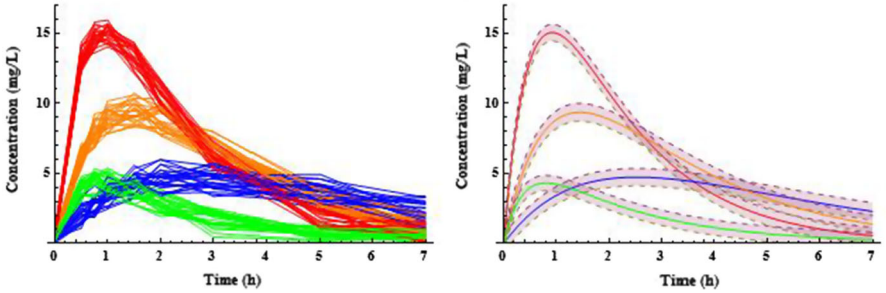

Fig. 3 On the left it is presented the dataset of experiment (ii). The four clusters elicited by the proposed EM algorithm are depicted in *different colors*; each *line* represents a subject. The algorithm was initialized with ten clusters and was able to recover the original four. On the right it is depicted the clusters found together with their standard deviation (color figure online)

Table 1 Description of results performed for nine additional experiments. The last column presents average time among the 100 runs for each experiment. The normalized SSw is higher when the dataset has higher within-group variance (see wgVar column). The algorithm was able to recover all original clusters

Test	wgVar	N	M	$\frac{\text{SSw}}{N \times n}$	Time
1	Low	60	4	0.1953	2.4125
2	High	100	4	1.4443	7.9685
3	Low	20	4	0.1482	0.4048
4	High	20	4	1.3421	0.7296
5	Low	60	3	0.2782	3.2308
6	High	200	4	1.4131	12.2858
7	Low	100	5	0.5467	2.1050
8	Low	100	5	0.1441	2.9747
9	Low	100	5	0.1724	4.2409

Squares within (SSw) was used. This performance metric calculates the sum of squared distances between all data points within the same cluster and the center of the cluster, for all clusters. To be able to compare among experiments, SSw was normalized with the number of computed distances, in our case, $N \times n$. Similarly to previous experiments, the algorithm was run over 100 random initializations and ten initial clusters. The result with the highest likelihood was chosen to compute the normalized SSw.

In conclusion, the algorithm was able to recover all original clusters. Moreover, the normalized SSw is higher, as expected, when the dataset has higher within-group variance (see wgVar column in Table 1).

4.1.3 Imbalanced data

In this section we studied how cluster recovering behaves with imbalanced data. Figure 4 depicts the curves used to generate the data.

In the first three datasets (Tests 1–3) the clusters are quite distinguishable due to the low standard deviations in each group. With these tests we wanted to assess the behaviour of the algorithm in a balanced scenario (20:20:20) and with between-

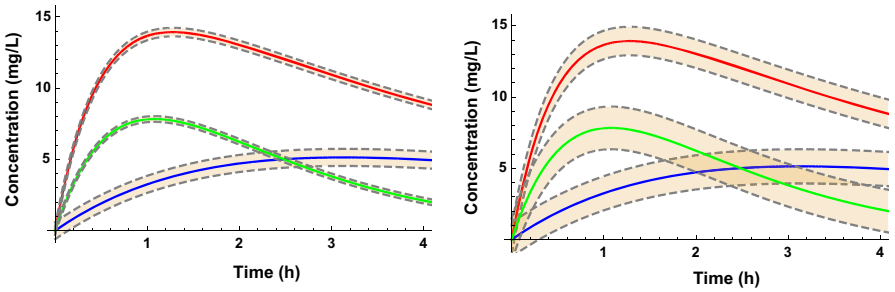

Fig. 4 On the *left* it is presented the curves used to generate synthetic tests 1–3; on the right the curves used to generate synthetic tests 4–6. Cluster 1 is depicted in *red*, 2 in *blue* and 3 in green (color figure online)

Table 2 Description of the imbalanced data

Test	$\sqrt{v_1} - \sqrt{v_2} - \sqrt{v_3}$	Imbalance
1		20:20:20
2	0.3–0.6–0.2	40:10:10
3		50:5:5
4		20:20:20
5	1.0–1.2–1.5	40:10:10
6		50:5:5

Each test has a total of 60 subjects, with eight time points, distributed among three different clusters. For instance, Test 3 has three clusters: (i) the first cluster with 50 subjects and a standard deviation $\sqrt{v_1} = 0.3$; (ii) the second cluster with 5 subjects and a standard deviation $\sqrt{v_2} = 0.6$; (iii) the third cluster with 5 subjects and a standard deviation $\sqrt{v_3} = 0.2$

cluster imbalances of 40:10:10 and 50:5:5. Check Table 2 for additional details in the parameters and Fig. 5 (top) for a plot of the data.

In contrast, in the last three datasets (Test 4–6) there is a significant overlap between the second (blue) and third (green) clusters, while the curves of the first (red) and third (green) cluster have a similar shape (and so, closer parameters); see Fig. 4 (right) and 5 (bottom). In this setup we wanted to model within-cluster imbalance by mixing the third (green) cluster with both the first (red) and the second (blue) cluster, and assess the behaviour of the algorithm.

We ran the algorithm in the above scenarios, with 100 random initializations, starting with ten clusters. The optimal solution of the algorithm was able to fully recover the original clusters for Tests 1–3. In Tests 4–6, the algorithm often split the clusters into new subconcepts but never merged the clusters or mixed subjects of distinct clusters. Results are presented in Table 3. Note that throughout the 100 random initializations sometimes two clusters were retrieved (Test 1–3 and 5–6); in these cases the second (blue) and the third (green) clusters were merged. In all the remaining cases, whenever three clusters were outputted, they were exactly the original clusters; if more than three clusters were retrieved, one or more of the original clusters were split in more than one cluster.

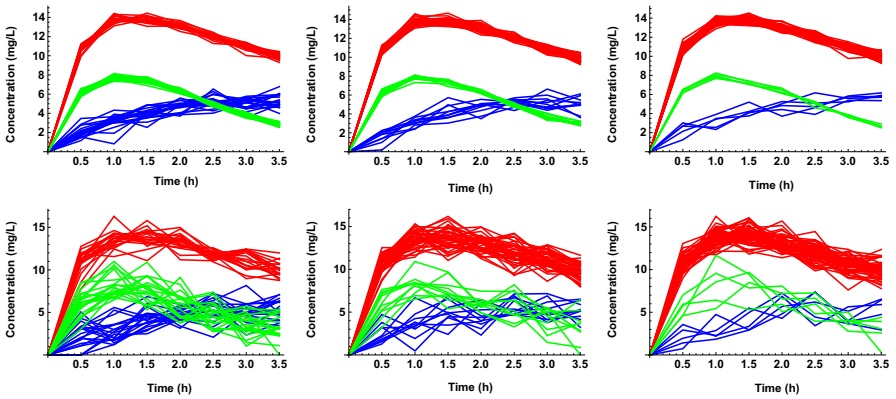

Fig. 5 The data of each test 1–6, from *top left* to *bottom right*, respectively (color figure online)

Table 3 Results of the clustering procedure for Test 1–6

Test	Optimal solution				Retrieved clusters			
	$\hat{\mathrm{M}}$	Cluster 1	Cluster 2	Cluster 3	2	3	4	5
1	3	20	20	20	8	92	0	0
2	3	40	10	10	2	98	0	0
3	3	50	5	5	5	95	0	0
4	5	20	$18+2$	$17+3$	0	17	49	34
5	4	40	10	$8+2$	11	56	32	1
6	4	40	5	$3+2$	20	62	18	0

The number of clusters found by the optimal solution is indicated by $\hat{\mathrm{M}}$. Column *Cluster i* indicates the number of subjects in the i-th cluster. For instance, in Test 1, the first cluster of the optimal solution has 20 subjects; in Test 4, the second original cluster was split in two in the optimal solution, one with 18 subjects and other with 2 subject. The column *Retrieved clusters* concerns the number of clusters found in the 100 random initializations, ranging from 2 to 5. For instance, in Test 1, the algorithm found 8 solutions with 2 clusters and 92 with 3. No solutions with more clusters were found in the 100 random initializations of Test 1

Although the results were close to the expected, we noticed that by choosing the likelihood to retrieve the optimal solution, the number of clusters tend to be higher with the extra clusters having low weight (in our case, 2 or 3 subjects). Indeed, it is well known that the likelihood overfits the data (Carvalho et al. 2011, 2014). Notwithstanding, having prior knowledge about the application domain, one can restrict the number of initial clusters and guide the algorithm to find the right ones. Alternatively, one can also merge clusters with low weight by setting $\overline{W}$ to a larger value (we used the default value of 0.025 in the experiments).

To avoid this shortcoming, we repeated Tests 4–6, with 100 random initializations, but starting with three random clusters instead of the previous ten. In this case, the algorithm was able to recover the original clusters in all the tests.

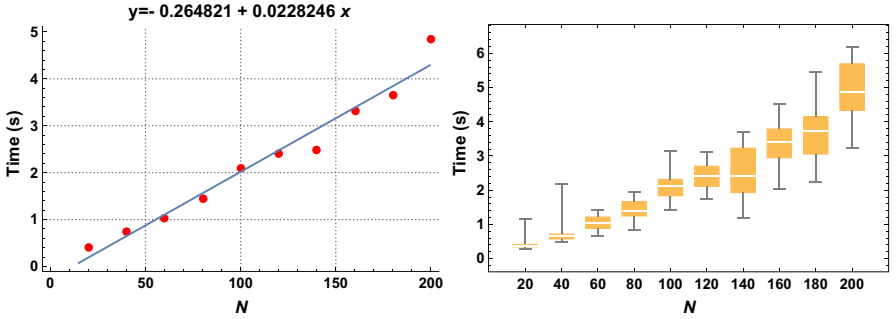

Fig. 6 On the *left* the average running time is plotted as a function of the number of subjects. The linear regression (in *blue*) is also plotted and it gives evidence that the algorithm runs linearly on the number of subjects. On the *right* the box-and-whisker chart for the 100 random initializations is presented (color figure online)

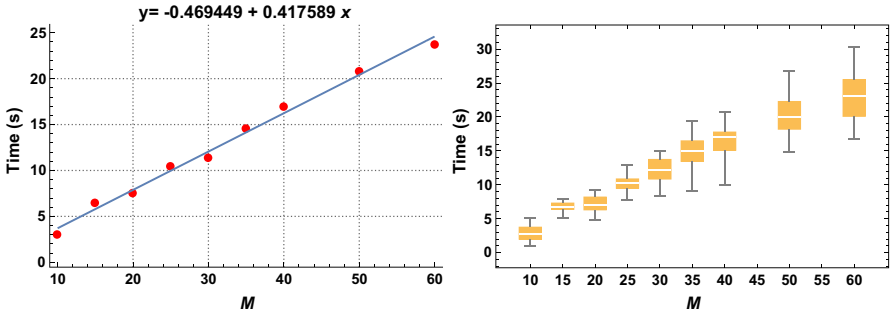

Fig. 7 On the *left* the average running time is plotted as a function of the number of initial clusters. The linear regression (in *blue*) is also plotted and it gives evidence that the algorithm runs linearly on the number of initial clusters. On the *right* the *box-and-whisker chart* for the 100 random initializations is presented (color figure online)

4.1.4 Running time

We assessed the average running time of the algorithm as a function of the number of subjects. For this purpose, we used datasets with the number of subjects ranging from 20 to 200, all with the same initial number of clusters $M = 10$. As usual, the algorithm was initialized randomly 100 times. The average running times are shown in Fig. 6.

A similar analysis was carried out to study the average running time of the algorithm in terms of the initial number of clusters. From a dataset with 100 subjects, we varied the initial number of clusters from 10 to 60. Again, the algorithm was initialized randomly 100 times. The average running times are exhibited in Fig. 7.

We conclude that the algorithm runs linearly both on the number of subjects and on the number of initial clusters.

4.2 Real data

The real dataset chosen was the analysis of theophylline pharmacokinetics. The clinical data (available e.g. in R), corresponds to a study on the kinetics of the anti-asthmatic

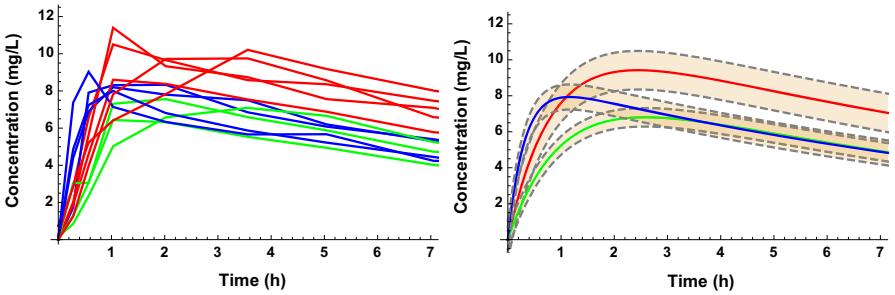

Fig. 8 On the *left* it is presented the dataset for the analysis of theophylline pharmacokinetics; each *line* represents a subject. On the *right* it is depicted the six clusters found together with their standard deviation. As clusters are unknown for this data, the *color* of a line on the *left* matches the corresponding cluster, found by the proposed algorithm, on the *right*. The algorithm was initialized with three clusters (as only 12 subjects were under study) and 100 random restarts (color figure online)

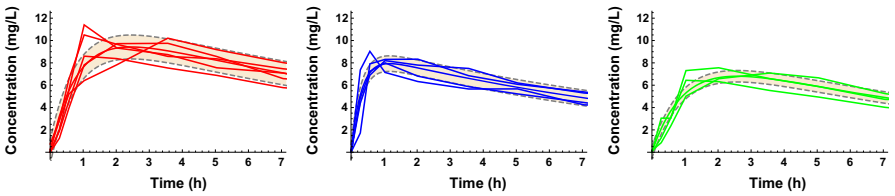

Fig. 9 The three clusters along with the original theophylline data grouped in the corresponding cluster. First cluster (*red*) parameters are: $\alpha_1 = 12.08$, $\beta_{11} = 0.08$ and $\beta_{21} = 1.21$; second cluster (*blue*) parameters are: $\alpha_2 = 9.03$, $\beta_{12} = 0.09$ and $\beta_{22} = 3.18$; third cluster (*green*) parameters are: $\alpha_3 = 9.60$, $\beta_{13} = 0.10$ and $\beta_{23} = 1.01$. The fitting of the data with the curves empirically supports the one-compartment model (color figure online)

drug theophylline where twelve patients were given oral doses of this drug and had their serum theophylline concentrations measured over the following 25 hours (Beal et al. 1993).

Figure 8 (left) shows the original time-series' concentrations. Figure 8 (right) shows the results of the clustering procedure with the proposed algorithm; it groups the theophylline time series in three clusters with different characteristics (corresponding to different absorption, distribution, metabolism, and excretion rates). Figure 9 shows the fit of each cluster with the original data.

From Fig. 9 we conclude that subjects in the first cluster (red) absorb more drug than those in the other clusters. Moreover, subjects in the second (blue) and third (green) cluster have more or less the same blood drug concentration after three hours, however, subjects in the second cluster tend to absorb the drug faster than those in the third cluster.

Finally, note that whenever clusters are elicited, the next step is to relate other patient features to the identified groups, allowing to predict *a priori* in which cluster a new patient belongs to. In this dataset such features were not present. This obviously requires domain knowledge to list which features are relevant to classify the patients into the obtained groups.

5 Conclusion and future work

The EM algorithm is a well established unsupervised learning method that allows to cluster data into similar groups. The main contribution of this work was to adapt EM to cluster time-series data for a relevant family of curves describing PK drug responses. As a special case of the method of Azzimonti et al. (2013), we explored the model features in order to skip the non-linear estimation step, hence favoring a single EM step. Experimental results showed that the proposed method was effective when learning in synthetic and real scenarios. In addition, the algorithm converged efficiently.

The ultimate goal of the proposed method is to study inter-patient variability in PK drug response. This will allow to stratify patients depending on their response and help physicians tailoring group-dependent therapies. This shall improve the therapy response while decreasing its side-effects. In particular, the method will be applied to study PK drug responses of HIV subjects co-infected with hepatitis B/C.

As future work we intend to identify and include fixed effects to describe population behavior. Research in PK drug response goes precisely towards finding such parameters, either using some empirical evidence or theoretical justification. From an algorithmic point of view, we also want to incorporate a penalty such like the *minimum description length* (Rissanen 1997) in the clustering procedure to avoid overfitting and low-weight clusters with outliers.

Acknowledgements The authors would like to express their appreciation to Paulo Mateus for many useful inputs and valuable comments. Special thanks go to the anonymous reviewers, who significant contributed to the quality of this manuscript with their valuable and well-aimed comments. This work was supported by national funds through FCT, Fundação para a Ciência e a Tecnologia, under contracts LAETA (UID/EMS/50022/2013) and IT (UID/EEA/50008/2013), and by projects InteleGen (PTDC/DTP-FTO/1747/2012), PERSEIDS (PTDC/EMS-SIS/0642/2014) and internal IT project QBig-Data. SV acknowledges support by Program Investigador FCT (IF/00653/2012) from FCT, co-funded by the European Social Fund (ESF) through the Operational Program Human Potential (POPH). This work was partially supported by the European Union's Horizon 2020 research and innovation program under grant agreement No. 633974 (SOUND project).

Appendix 1: Experimental setup

As described in Sect. 3.1, the M-step maximizes the likelihood over the parameters $\beta_{1\ell}$ and $\beta_{2\ell}$, for each cluster ℓ, resorting to a numerical method; indeed, it is overwhelming to obtain the analytical solution of the transcendental system of equations that provided the maxima. Next, we detail a simple optimization and stopping criteria for coordinate descent method used in the experiments presented in Sect. 4.

Optimization

To improve the convergence rate of coordinate descent method we made an improvement in the method by intercalating the (one-dimensional) Newtown iterations for variables $\beta_{1\ell}$ and $\beta_{2\ell}$, thus, solving Eqs. (4) and (5) simultaneously. With such modification, the method converged significantly faster, without deteriorating the results. Recall that Newton(b, h) is an iterative method to find a root of a function h given

an initial approximation b, such that, in the d-th iteration of Newton's method we have

$$b^{(0)} = b \text{ and } b^{(d+1)} = b^{(d)} - \frac{h(b^{(d)})}{h'(b^{(d)})}.$$

We can recast the M-step for maximizing $\beta_{1\ell}$ and $\beta_{2\ell}$ as the problem of finding b_1 that maximizes $h_1(b_1, b_2)$, given b_2 fixed, and b_2 that maximizes $h_2(b_1, b_2)$, given b_1 fixed, simultaneously. A simple approach to address this problem is to interleave the iterations of Newton's method as $b_1^{(0)}, b_2^{(0)}, b_1^{(1)}, b_2^{(1)}, \ldots$ so that

$$b_1^{(d+1)} = b_1^{(d)} - \frac{h_1(b_1^{(d)}, b_2^{(d)})}{\frac{\partial h_1}{\partial b_1}(b_1^{(d)}, b_2^{(d)})} \text{ and } b_2^{(d+1)} = b_2^{(d)} - \frac{h_2(b_1^{(d)}, b_2^{(d)})}{\frac{\partial h_2}{\partial b_2}(b_1^{(d)}, b_2^{(d)})}.$$

We took this approach where

$$b_1^{(0)} = \beta_{1\ell}^{(k)} \text{ and } b_2^{(0)} = \beta_{2\ell}^{(k)}.$$

Note that k is used for the k-th iteration of the EM algorithm, whereas d (in the above case $d = 0$) is used for the d-th iteration of the Newton's method. So the previous estimates of $\beta_{1\ell}$ and $\beta_{2\ell}$ by the EM algorithm, are the initial approximation for the roots in Newton's method. In this case, we let

$$h_1(b_1, b_2) = h_{1\ell}^{(k)}(b_1, b_2) \text{ and } h_2(b_1, b_2) = h_{2\ell}^{(k)}(b_1, b_2),$$

with $h_{1\ell}^{(k)}(b_1, b_2)$ defined as

$$\sum_{i=1}^{N}\sum_{j=1}^{n}\Big(-\frac{\alpha_\ell^{(k+1)}}{v_\ell^{(k)}} X_{i\ell}^{(k)} t_j e^{-b_1 t_j}\Big(y_{ij} - \alpha_\ell^{(k+1)}(e^{-b_1 t_j} - e^{-b_2 t_j})\Big)\Big),$$

and $h_{2\ell}^{(k)}(b_1, b_2)$ defined as

$$\sum_{i=1}^{N}\sum_{j=1}^{n}\Big(\frac{\alpha_\ell^{(k+1)}}{v_\ell^{(k)}} X_{i\ell}^{(k)} t_j e^{-b_2 t_j}\Big(y_{ij} - \alpha_\ell^{(k+1)}(e^{-b_1 t_j} - e^{-b_2 t_j})\Big)\Big).$$

Observe that functions $h_{1\ell}^{(k)}$ and $h_{2\ell}^{(k)}$ as defined in Sect. 3.1 have only one argument, while here they have two for the purpose of applying the envisage optimization. With this optimization we start with $b_1^{(0)} = \beta_{1\ell}^{(k)}$ and $b_2^{(0)} = \beta_{2\ell}^{(k)}$, then we iterate the Newton's method until convergence in, say, d iterations, and finally, we update $\beta_{1\ell}$ and $\beta_{2\ell}$ as $\beta_{1\ell}^{(k+1)} = b_1^{(d)}$ and $\beta_{2\ell}^{(k+1)} = b_2^{(d)}$.

Stopping criterion

In our implementation, the Newton(b, h) method is stopped at iteration d if

$$| b^{(d)} - b^{(d-1)} | < 10^{-10}.$$

Moreover, if the number of iterations exceeds 10^4 the method is aborted (with a warning to the user). However, in practice, this iteration limit was never achieved in the experiments performed in Sect. 4.

In our particular setting, as we are computing at each step both $\beta_{1\ell}$ and $\beta_{2\ell}$, we stop when both approximations fulfill the previous criteria.

Knee analysis

For the clustered curves to have biological meaning (e.g., concentrations need to be positive, rates cannot become exponentially faster, etc.), we imposed that

$$\beta_{1\ell}, \beta_{2\ell} \in (0, 5) \text{ and } \beta_{1\ell} \leq \beta_{2\ell}.$$

If Newton's method ends with $\beta_{1\ell} > \beta_{2\ell}$ then we swap $\beta_{1\ell}$ with $\beta_{2\ell}$; in practice, this never happened in the experiments performed in Sect. 4. Nonetheless, if $\beta_{1\ell} \notin (0, 5)$ or $\beta_{2\ell} \notin (0, 5)$, we perform a knee analysis to elicit a value within the allowed range.

In a general setting of Newton(b, h), assume that the method converges in iteration d and $b^{(d)}$ is out of some allowed range. It might be very well the case $b^{(d)}$ is out of range but the images of h within the allowed range are very closed to $h(b^{(d)})$. In this case, the method does not converge inside the range because $b^{(d)} > b^{(d-1)}$ but $h(b^{(d)}) \approx h(b^{(d-1)})$. Therefore, the method should be stopped when

$$| h(b^{(d)}) - h(b^{(d-1)}) | < 10^{-10},$$

which is called a *knee analysis*.

In our particular case, we perform this knee analysis to avoid $\beta_{1\ell} \notin (0, 5)$ and $\beta_{2\ell} \notin (0, 5)$. If, even performing this analysis, we have $\beta_{1\ell} \notin (0, 5)$ then we keep the previous parameters approximation, that is, $\beta_{1\ell}^{(k+1)} = \beta_{1\ell}^{(k)}$, and the EM algorithm proceeds. The same analysis is carried out for $\beta_{2\ell}$.

Appendix 2: Implementation details

In this appendix, we present implementation details related to the proposed EM algorithm.

Initial number of clusters

The initial number of clusters is an user-defined parameter. However, if this value is not given, we set $M = \frac{N}{3}$.

Stopping criterion

Usually, an EM algorithm stops when the difference between consecutive values of the parameters reaches some threshold. In our case, the impact of the parameters $\beta_{1\ell}$ and $\beta_{2\ell}$ overwhelms the impact of all remaining parameters in the definition of the clusters; recall that $\beta_{1\ell}$ and $\beta_{2\ell}$ are exponents in Eq. (2). For this reason, only $\beta_{1\ell}$ and $\beta_{2\ell}$ are considered in the stopping criterion of the proposed EM algorithm. Specifically, the EM algorithm stops when

$$\frac{\mid \beta_{1\ell}^{(k+1)} - \beta_{1\ell}^{(k)} \mid}{\mid \beta_{1\ell}^{(k+1)} \mid} \leq 10^{-6} \text{ and } \frac{\mid \beta_{2\ell}^{(k+1)} - \beta_{2\ell}^{(k)} \mid}{\mid \beta_{2\ell}^{(k+1)} \mid} \leq 10^{-6},$$

for all clusters $\ell \in \{1, \ldots, M\}$.

Cluster thresholds

The thresholds for disregarding and merging clusters were defined (empirically) as $\overline{W} = 0.025$ and $\overline{L} = 1$.

References

Azzimonti L, Ieva F, Paganoni AM (2013) Nonlinear nonparametric mixed-effects models for unsupervised classification. Comput Stat 28(4):1549–1570

Beal SL, Sheiner LB (1980) The NONMEM system. Am Stat 34:118–119

Beal SL, Sheiner LB, Boeckmann AJ (1993) NONMEM users guide. Technical report, University of California, San Francisco

Carvalho AM, Adão P, Mateus P (2014) Hybrid learning of Bayesian multinets for binary classification. Pattern Recognit 47:3438–3450

Carvalho AM, Roos T, Oliveira AL, Myllymki P (2011) Discriminative learning of Bayesian networks via factorized conditional log-likelihood. J Mach Learn Res 12:2181–2210

Davidian M, Giltinan DM (2003) Nonlinear models for repeated measurement data: an overview and update. J Agric Biol Environ Stat 8:387–419

Dayneka NL, Garg V, Jusko WJ (1993) Comparison of four basic models of indirect pharmacodynamic responses. J Pharmacokinet Biopharm 21(4):457–478

Delyon B, Lavielle M, Moulines E (1999) Convergence os a stochastic approximation version of the EM procedure. Ann Stat 27:94–128

Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc Ser B 1:1–38

Derendorf H, Lesko LJ, Chaikin P, Colburn WA, Lee P, Miller R, Powell R, Rhodes G, Stanski D, Venitz J (2000) Pharmacokinetic/pharmacodynamic modeling in drug research and development. J Clin Pharmacol 40(12 Pt 2):1399–1418

Gueorguieva I, Ogungbenro K, Graham G, Glatt S, Aarons L (2007) A program for individual and population optimal design for univariate and multivariate response pharmacokinetic-pharmacodynamic models. Comput Methods Programs Biomed 86(1):51–61

Kuhn E, Lavielle M (2005) Maximum likelihood estimation in nonlinear mixed effects models. Comput Stat Data Anal 49:1020–1038

Lee J, Lee H, Jang K, Lim KS, Shin D, Yu KS (2014) Evaluation of the pharmacokinetic and pharmacodynamic drug interactions between cilnidipine and valsartan, in healthy volunteers. Drug Des Dev Ther 8:1781–1788

Lee PID, Amidon GL (1996) Pharmacokinetic analysis: a practical approach. CRC Press, Boca Raton

Lindstrom MJ, Bates DM (1988) Newton–Raphson and EM algorithms for linear mixed-effects models for repeated-measures data. J Am Stat Assoc 84:1014–1022
Mager DE, Wyska E, Jusko WJ (2003) Diversity of mechanism-based pharmacodynamic models. Drug Metab Dispos 31(5):510–518
Nesterov Y (2012) Efficiency of coordinate descent methods on huge-scale optimization problems. SIAM J Optim 22:341–362
Rissanen J (1997) Stochastic complexity in learning. J Comput Syst Sci 55:89–95
Roden DM, George AL Jr (2002) The genetic basis of variability in drug responses. Nat Rev Drug Discov 1:37–44
Sheiner LB, Rosenberg B, Marathe VV (1977) Estimation of population characteristics of pharmacokinetic parameters from routine clinical data. J Pharmacokinet Biopharm 5:445–479
Trout H, Mentré F, Panhard X, Kodjo A, Escaut L, Pernet P, Gobert JG, Vittecoq D, Knellwolf AL, Caulin C, Bergmann JF (2004) Enhanced saquinavir exposure in HIV1-infected patients with diarrhea and/or wasting syndrome. Antimicrob Agents Chemother 48:538–545
Walker G (1996) An em algorithm for non-linear random effects models. Biometrics 52:934–944
Wei GC, Tanners MZ (1991) Applications of multiple imputation to the analysis of censored regression data. Biometrics 47:1297–1309
Wright SJ (2015) Coordinate descent algorithms. Math Program 151:3–34
Wu L (2002) A joint model for nonlinear mixed-effects models with censoring and covariates measured with error, with application to AIDS studies. J Am Stat Assoc 97:955–964
Wu L (2004) Exact and approximate inferences for nonlinear mixed-effects models with missing covariates. J Am Stat Assoc 99:700–709