# Model Selection for Clustering of Pharmacokinetic Responses with the Minimum Description Length

Rui Pedro Pimentel de Almeida Guerra
ruippguerra@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

November 2017

**Abstract**

When a patient has some kind of illness, a medical doctor is often able to recommend some dosage of a certain drug. The reaction to the drug, however, greatly depends on the patient and is described by the temporal evolution of the concentration of the drug in the patient's blood, called the pharmacokinetic curve, if a simple model of the human body is assumed. Personalized medical treatment according to the needs of different patients would be the ideal procedure. However, the large amount of patients can make this become unfeasible. The solution is to perform clustering on the pharmacokinetic curves of each patient's response to a drug to group patients with similar needs. Doing so must be reliable and efficient. The aim of this work is to adapt an existing solution to this problem that performed clustering using the Expectation-Maximization algorithm and apply different techniques, such as the Minimum Description Length principle and the Normalized Maximum Likelihood codelength. The result will be the possibility of finding the best clustering output without the need of arbitrary parameters to merge clusters and optimizing the number of clusters which is unknown at the start. In addition, the resulting implementation will be cost efficient.
**Keywords:** Clustering, Model Selection, Pharmacokinetics, Minimum Description Length, Normalized Maximum Likelihood.

## 1. Introduction

When a person has some kind of illness, the best solution is often to consume some dosage of a certain drug recommended by a medical doctor. This drug usage can, however, if not applied with a specific dosage that depends on the subject, cause undesired side-effects that can hamper the success of the therapy that the person is subject to.

Clinical pharmacokinetics [1] aims to study the evolution of the drug in a subject's body in terms of the variable concentration of the drug in his blood vessels, while taking in account the body's natural absorption and elimination of the drug. It then applies this knowledge to personalize the treatment of different subjects.

The concept is simple, if it is possible to use pharmacokinetics to build a relatively simple model of the human body, the treatment can be easily adjusted to that patient's needs. For instance, considering that the entire human circulatory system is a single compartment with constant drug concentration in a single instant, which is actually not a bad simplification, a model can be used to see how the drug behaves in that subject's body.

The problem, however, is the fact that when there is a large number of subjects in need of treatment, attempting to personalize the treatment to each individual subject's needs is unfeasible. The solution to this problem is to divide the subjects in groups, where subjects in the same group have similar drug responses, and then adjust the treatment according to the average pharmacokinetic response of each group. This can be achieved by performing clustering on the pharmacokinetic curves of the subjects.

Being able to reliably and efficiently find methods to perform the clustering of these pharmacokinetic (PK) curves is the motivation of this work.

The objective of this thesis is to build on a previous work done on the subject of clustering of PK curves [2] in order to improve its reliability, by reducing the risks of data overfitting, and its efficiency, by minimizing the amount of time needed in order to compute the results.

The Expectation-Maximization (EM) algorithm used is based on a different work [3]. Therein, the algorithm is used to estimate nonlinear parameters, with the small change that the cluster variances are fixed instead of being specific to each cluster.

The clustering program can be more reliable if different criteria is used in order to identify the best

clustering outputs, such as the Minimum Description Length (MDL) principle [4] or the Normalized Maximum Likelihood (NML) method [5], both of which will be used in this work. Doing so will allow the possibility of finding the best outputs without the need of arbitrary parameters to merge clusters by being able to optimize the number of clusters in the data, which is unknown at the start. Although applying this kind of criteria to obtain clustering results can become more computationally heavy than by using a simple EM algorithm, by allowing the program to run using a parallel implementation, it is still possible to make it become efficient.

In this work, two novel model selection criteria were proposed to perform clustering of pharmacokinetic responses using both MDL and NML coding, which are free of cluster merging parameters by optimizing the number of clusters. The resulting cost-efficient implementation as well as a user guide and some synthetic datasets are available at a GitHub repository in [6].

## 2. Background

In clinical pharmacokinetics the main objective is to monitor along time a specific drug concentration in the blood or plasma of individual patients in order to attempt to personalize the patient's treatment.

This section will firstly explain basic pharmacokinetic concepts, then mention existing solutions regarding data clustering and their problems as well as proposed techniques to solve them, such as the Minimum Description Length principle.

### 2.1. Pharmacokinetic Models

In order to describe and predict the effect that a drug has on a patient, it is necessary to create a simplified model of the human body. These are called Pharmacokinetic (PK) Models [7].

The most commonly used model in practice is the one-compartment model [2]. In this very simple model, the entire human circulatory system is considered as a single compartment with a constant volume $V$, commonly measured in litres, and a time-variant quantity of drug $Q(t)$ within that volume, measured in milligrams. This time variance is caused either by absorption or elimination of the drug from the body. These processes are ruled by two constants that depend on the patient, namely the absorption rate constant ($k_a$) and the elimination rate constant ($k_e$). Considering the absorption of the drug by the body as a function of time $I(t)$, with initial condition $I(0)$ given by $I(0) = Dose \times F$, where $Dose$ is the initial dosage and $F$ is a constant related to the bioavailability of the patient, it is possible to write a differential equation representing the quantity of drug in the body given by

$$Q'(t) = -k_e Q(t) + k_a I(t)\,. \tag{1}$$

A basic schematic of the one-compartment model is shown in Fig. 1.

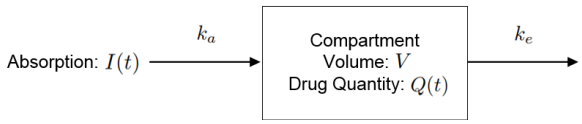



Figure 1: One-compartment model.

The concentration $C(t)$ of the drug along time in the single compartment is used to more consistently compare the reaction of the drug in different patients. This can simply be computed using the expression

$$C(t) = \frac{Q(t)}{V}\,. \tag{2}$$

### 2.2. Clustering

In order to specialize the treatment to different patients, it is important to group these among others that have similar responses to the same drug. Doing this would allow the same treatment to be used with patients with similar responses, facilitating a sense of personalized medicine, without needing to specialize for each individual patient.

To accomplish this, clustering algorithms can be used. There are several different types of clustering methods that can be used but for the purpose of this work the focus will be on the ones based on finding probability distributions, such is the case of ones using the Expectation-Maximization (EM) algorithm [8]. Its name comes from the two steps of its iterative process, the Expectation (E) step and the Maximization (M) step [9].

Given a vector of unknown parameters $\boldsymbol{\theta}$, consisting of the model and weight parameters, as well as the current estimates of such parameters $\boldsymbol{\theta}^{(k)}$, the E step consists of computing an objective function $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)})$ which maximization corresponds to the maximization of the likelihood of the data. This function is defined as

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)}) = \sum_{l=1}^{M} \sum_{i=1}^{N} X_{il}^{(k)} \log(\omega_l p_l(\boldsymbol{y}_i))\,, \tag{3}$$

where $\omega_l$ is the estimated probability of an observation to belong to cluster $l$, called the weight of the cluster, $\boldsymbol{y}$ is an $n$-dimensional observation from an array of $N$ observations, $p_l(\boldsymbol{y}_i)$ is the probability of observation $\boldsymbol{y}_i$ belonging to cluster $l$, $M$ is the number of clusters and $X_{il}^{(k)}$ is the degree of belonging of observation $\boldsymbol{y}_i$ to cluster $l$ given by

$$X_{il}^{(k)} = \frac{\omega_l^{(k)} p_l^{(k)}(\boldsymbol{y}_i)}{\sum_{r=1}^{M} \omega_r^{(k)} p_r^{(k)}(\boldsymbol{y}_i)}\,. \tag{4}$$

The M step consists in finding a new set of parameters $\boldsymbol{\theta}^{(k+1)}$ that maximizes the objective function given by $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)})$.

The EM algorithm requires the number of clusters to be known from the start. This, however, might not be a trivial parameter to deduce, especially with large quantities and high complexity of the data. Finding a method to determine the optimal number of clusters to use has been a subject of research, but the best attempts at solving this problem have been through the use of heuristic methods.

2.2.1. Application to the Pharmacokinetic Problem To apply a clustering method to the problem at hand, it is necessary to firstly create a model for the pharmacokinetic curves representing the drug concentration along time on the patients. Prior work has been done on the application of the EM algorithm to this problem [2], which will be summarized in this section.

By solving the system of equations given by (1) and (2), one can obtain an expression for the drug concentration given by

$$C(t) = \alpha(e^{-\beta_1 t} - e^{-\beta_2 t}) \,, \tag{5}$$

where

$$\alpha = \frac{k_a Dose \times F}{V(k_a - k_e)}, \; \beta_1 = k_e \text{ and } \beta_2 = k_a \,. \tag{6}$$

The variables $\alpha$, $\beta_1$ and $\beta_2$ will be the parameters of the pharmacokinetic curves modeling the drug responses of the patients. It is around this model that it is possible to apply clustering algorithms in order to learn these responses. A given cluster $l$ can have its drug concentration over time described by

$$C_l(t) = \alpha_l(e^{-\beta_{1l} t} - e^{-\beta_{2l} t}) \,, \tag{7}$$

where each subject $i$ of cluster $l$ has a concentration $y_{il}$ at time $j$ given by

$$y_{il} = C_l(t_j) + \epsilon_{ijl} \,, \tag{8}$$

where $\epsilon_{ijl}$ is gaussian error with zero mean and variance $\upsilon_l$. If we assume that the errors at different time instants $j$ are independent, and that the instants of time of the measurements are the same for every subject given by $\boldsymbol{t} = (t_1, ..., t_n)$, we can write the probability density function of $\boldsymbol{y}_i = (y_{i1}, ..., y_{in})$, as in the measurements in the first $n$ instants of time for subject $i$ in cluster $l$ as

$$p_l(\boldsymbol{y}_i) = \frac{1}{(2\pi\upsilon_l)^{\frac{n}{2}}} e^{\frac{-1}{2\upsilon_l}\sum_{j=1}^{n}(y_{ij} - C_l(t_j))^2} \,. \tag{9}$$

Since each subject is assumed to have to belong to a cluster, we can give weights to each cluster to represent the probability of a given subject to belong to that cluster. For cluster $l$ this weight is $\omega_l$. Therefore, considering a random vector $\boldsymbol{W} = (W_1, ..., W_N)$ where $W_i$ is a random variable that describes the cluster to which observation $i$ belongs, we have

$$P(W_i = l) = \omega_l \text{ for } 1 \leq l \leq M \,. \tag{10}$$

The EM algorithm will be applied to the model defined. The observed data can be stored in a matrix $Y$ of size $N \times n$ corresponding to the responses of $N$ patients during $n$ sampling instants, assumed the same for every patient for the sake of simplicity. The number of clusters is also assumed to be known with value $M$.

The cluster parameters to estimate are

$$\boldsymbol{\theta} = \{\alpha_l, \beta_{1l}, \beta_{2l}, \upsilon_l, \omega_l\}_{l \in 1,...,M} \,. \tag{11}$$

Let $\boldsymbol{w}$ be a realization of $\boldsymbol{W}$ such as $\boldsymbol{w} = (l_1, ..., l_N)$ where $l_i$ is the cluster subject $i$ belongs to. It is possible to write the probability of observing data $Y$ and clustering result $\boldsymbol{w}$ as

$$p_\theta(Y, \boldsymbol{w}) = \prod_{i=1}^{N} \omega_{li} p_l(\boldsymbol{y}_i) \,. \tag{12}$$

The E step of the algorithm is simply to compute the objective function $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)})$. This was defined in Equation (3).

For the M step, it is necessary to derive equations to update each of the parameters in Equation (11).

The update of the model parameters $\alpha_l, \beta_{1l}$ and $\beta_{2l}$ is more complex since it would require solving transcendental equations. Numerical methods are, therefore, used in order to update these parameters, namely the coordinate descent method and Newton's method. The weight and variance parameters $\omega_l$ and $\upsilon_l$ have an update equation based on the canonical expressions for Gaussian mixtures. The full explanation of how the expressions to update these parameters were obtained is explained in [2].

As was explained previously, to use the EM algorithm it is necessary to assume a fixed number $M$ of clusters. In an attempt to improve the result obtained, previous works used certain techniques such as disregarding negligible clusters and merging similar clusters.

Several tests of the EM algorithm with cluster collapsing techniques were performed on both synthetic and real data [2]. While the results were acceptable, there were situations containing noisy data that more clusters were detected than expected, containing less subjects, without merges being performed successfully. This occurred because the algorithm tends to overfit to the data, and the parameters to avoid it were difficult to tune.

2.3. Model Selection

A possibility proposed to solve the problem of selecting the correct model for the data is the usage of the Minimum Description Length (MDL) principle or the Normalized Maximum Likelihood codelength, described in the following sections.

2.3.1. Minimum Description Length Principle

The MDL principle [4] is a method proposed by J. Rissanen which states that the best description of data is the one that manages to compress it the most according to its regularity [10]. This is because the amount of information that can be learned from the data corresponds to how much it is possible to compress it. For example, a sequence of perfectly random numbers can only be described by its entire length. On the other hand, if this sequence follows a repeating pattern, one can describe it using only one instance of said pattern. Identifying the pattern means something has been *learned* from the data. This is what MDL strives for, finding an hypothesis to explain the data that at the same time is simple as possible and compresses the data as much as possible.

A data model can be defined as a set of probability distributions, or functions in general, that have the same format and contain generic, unrealized parameters. A data hypothesis is a single realization of the model, by giving values to its parameters.

The first and simplest implementation of the MDL principle is one that divides an objective function in two parts [10], crudely given by $L(H) + L(Y|H)$, where $L(H)$ corresponds to the length, in bits, needed to describe an hypothesis $H$ and $L(Y|H)$ is the length, in bits, of the description of data $Y$ according to hypothesis $H$.

The minimization of this function shows the main objective of MDL: to find a balanced solution $H$ that at the same time is simplistic enough in itself and describes the data $Y$ as simply as possible.

To use this form of MDL, it is necessary to define both parts of this code in a way that allows the expression to be computable.

In order to define the second part of the expression, $L(Y|H)$, it is possible to use the fact that the hypothesis $H$ defines a probability distribution for the data. Considering that the length is measured in bits, the codelength of data $Y$ while using hypothesis $H$ can be given by the *Shannon-Fano* code [11], and so it is possible to write

$$L(Y|H) = -\log P(Y|H)\,, \tag{13}$$

where $P(Y|H)$ is the probability density of data $Y$ according to hypothesis $H$. This expression shows the parallelism between finding the shortest length code and finding the distribution with the highest log-likelihood.

It is visible that the other part of the code, $L(H)$, depends only on the hypothesis at hand and not on the observed data. It is difficult to specify an expression to define the description length to compare different hypotheses because different hypotheses can have different relative lengths depending on the code used. This is intuitive because if these codes are seen as different programming languages, the codelength will depend on it, thus making the choice arbitrary and defeating the purpose of having different description lengths for hypotheses. This is the issue of trying to find the hypothesis with the best Kolmogorov complexity [12]. This is the reason why a different, refined version of MDL is commonly used instead, which uses a single one-part code in its application.

The main idea to create a refined version of MDL is by encoding the data using not the different hypothesis $H$, but the full model $\mathcal{H}$ instead. Doing this prevents the problem encountered with crude MDL since the objective function will be simply a one-part code given by $\overline{L}(Y|\mathcal{H})$, which can be reffered to as the stochastic complexity of the model $\mathcal{H}$ [10, 13]. This change can be done because when a certain parameter in the model is a good fit to the data, this term becomes smaller as the $L(Y|H)$ term did in crude MDL.

An important term to define the objective function to minimize is called the parametric complexity of the model, denoted as $\mathbf{COMP}(\mathcal{H})$. This term is related with the geometry and the number of degrees of freedom existing in model $\mathcal{H}$, thus being a good indicator of how well the model fits random data. This value is commonly computed as a function of the logarithm of the number of free variables in the model. The parametric complexity can be related with the stochastic complexity by the expression

$$\overline{L}(Y|\mathcal{H}) = L(Y|\hat{H}) + \mathbf{COMP}(\mathcal{H})\,, \tag{14}$$

where $\hat{H}$ is the model distribution that maximizes the probability of $\mathcal{H}$ being correct. Therefore, the $L(Y|\hat{H})$ term can be seen as the term that describes the goodness of fit of the data.

The parametric complexity term is the one that allows the comparison of results obtained using different numbers of clusters. It is based on a model selection criterion proposed by Schwarz [14] known as the *Bayesian Information Criterion*, identical to one derived [15] by Rissanen [16] which represents the extra amount of bits needed to describe the data given a number of parameters or degrees of freedom. This term depends on the number of observations $N$ and the total number of degrees of freedom in the model $K$ as well as the number of observations that belongs to each cluster, given by $h_k$ for cluster

$k$. It is given by

$$\mathbf{COMP}(\mathcal{H}) = \frac{1}{2}\sum_{k=1}^{M}\log h_k + \frac{1}{2}K\log N\,. \quad (15)$$

Since the first term of this expression commonly has a small weight, it is often disregarded for the sake of simplicity, which is what will be done in this work.

With the new and refined expression for the objective function of MDL, a trade-off is achieved that guarantees that the model fits the data as well as possible, while simultaneously guaranteeing that the model itself is as simple as possible.

2.3.2. Normalized Maximum Likelihood Coding

An alternative method to define the stochastic complexity of a model is by using the NML codelength [17]. In order to understand the need for this method, it is important to know the concept of the regret [18] of a model.

In terms of codelengths, the regret can be seen as the additional length in bits needed to encode a certain observation using a given probability distribution $f$ of a model with parameters $\boldsymbol{\theta}$ in comparison to the bits needed if an "optimal" distribution (with parameters $\hat{\boldsymbol{\theta}}$) is used. A good measure to see how good a model is can be obtained by checking what the regret is in the worst case, by finding the maximum possible regret. The best model distribution is, therefore, one that minimizes the maximum regret $\hat{\boldsymbol{\theta}}$ as in

$$\min_f \mathcal{R}_{\max}(f) = \min_f \max_{\boldsymbol{y}_i}(-\log f(\boldsymbol{y}_i) + \log f(\boldsymbol{y}_i;\hat{\boldsymbol{\theta}}(\boldsymbol{y}_i)))\,. \quad (16)$$

The solution to this minimax problem is achieved by the NML distribution, also known as the Shtarkov distribution [19], given by

$$f_{\mathrm{NML}}(\boldsymbol{y}_i,\mathcal{H}) = \frac{f(\boldsymbol{y}_i;\hat{\boldsymbol{\theta}}(\boldsymbol{y}_i))}{\mathcal{C}(\mathcal{H})}\,, \quad (17)$$

which for the continuous case has

$$\mathcal{C}(\mathcal{H}) = \int_{\boldsymbol{y}_i} f(\boldsymbol{y}_i;\hat{\boldsymbol{\theta}}(\boldsymbol{y}_i))\,. \quad (18)$$

By applying a logarithm to the NML distribution, it is possible to obtain the NML code length, or stochastic complexity given by

$$-\log f_{\mathrm{NML}}(\boldsymbol{y}_i,\mathcal{H}) = -\log f(\boldsymbol{y}_i;\hat{\boldsymbol{\theta}}(\boldsymbol{y}_i)) + \log\mathcal{C}(\mathcal{H})\,. \quad (19)$$

From this expression it is easy to understand that the first tirm corresponds, again, to the goodness-of-fit term, which is identical to the normal MDL case, and the second term corresponds to the parametric complexity $\mathbf{COMP}(\mathcal{H})$.

The value of the parametric complexity can be difficult to compute for certain probability distributions of the models. However, for Gaussian mixture models, which is the case in study in this work, an expression has been derived in [5] dependant on the number of clusters $M$ and the number of subjects $N$ given by

$$\mathcal{C}(\mathcal{H}(M),N) = \sum_{h_1+\cdots+h_M=N}\frac{N!}{h_1!\cdots h_M!} \times\prod_{k=1}^{M}\left(\frac{h_k}{N}\right)^{h_k}\times I(h_k)\,, \quad (20)$$

where

$$I(h_k) = B(n,\lambda_{min},R)\left(\frac{h_k}{2e}\right)^{\frac{nh_k}{2}}\frac{1}{\Gamma_m\left(\frac{h_k-1}{2}\right)}\,, \quad (21)$$

$$B(n,\lambda_{min},R) = \frac{2^{n+1}R^{\frac{n}{2}}\prod_{j=1}^{n}{\lambda_{min}^{(j)}}^{-\frac{n}{2}}}{n^{n+1}\Gamma\left(\frac{n}{2}\right)}\,, \quad (22)$$

$\lambda_{min}$ is a lower bound of the variance of the Gaussian distributions, which is a constant in this case because the variance in each time instant is independent and assumed to be the same, $R$ is an upper bound of the square of the mean value of the Gaussian mixture output, $n$ is the number of time instants, $\Gamma$ is the Gamma function and $\Gamma_m$ is the multivariate Gamma function.

The expression in Eq. (20) can still be quite difficult to compute. However, this computation can be simplified by performing a recursive algorithm described in Algorithm 1 as stated in [5], with a computational complexity of $\mathcal{O}(N^2\times M)$..

Using this algorithm it is possible to immediately obtain the parametric complexity term for every possible number of clusters $M$ by using the value stored in $\mathcal{C}(\mathcal{H}(M),N)$.

## 3. Methods and Implementation

In order to solve the problem related to the clustering of pharmacokinetic curves using the EM algorithm along with the MDL principle or the NML codelength, an implementation using the Java programming language was made. The resulting implementation is given in a GitHub repository in [6].

The resulting program was based on an adaptation of an existing program [2] that solved the problem using only the Expectation-Maximization algorithm along with user parameters to disregard and merge superfluous clusters. These parameters provided an heuristic solution to the problem without which there was a tendency to overfit the data, providing low-weight clusters with outliers.

**Algorithm 1** NML Parametric Complexity

1: Set $\mathcal{C}(\mathcal{H}(M), 0) = 1$;
2: Compute $\mathcal{C}(\mathcal{H}(1), j) = I(j)$ for $j = 1, \ldots, N$;
3: **for** $k = 2$ to $M$ **do**
4: **for** $j = 1$ to $N$ **do**
5: Compute $\mathcal{C}(\mathcal{H}(k), j) = \sum_{r_1+r_2=j} \binom{j}{r_1} \left(\frac{r_1}{j}\right)^{r_1} \left(\frac{r_2}{j}\right)^{r_2} \times \mathcal{C}(\mathcal{H}(k-1), r_1) I(r_2)$;
6: **end for**
7: **end for**

In an attempt to overcome the shortcomings of the existing implementation, two adaptations using the MDL principle and the NML codelength were made, which will be described next. Afterwards, the changes to the Expectation-Maximization algorithm and the solution for a cost efficient implementation will be discussed.

3.1. MDL for Clustering PK Responses

Since the algorithm that will be used will require computation of several different initializations, it is necessary to have a good measurement to compare the different outputs. In order to achieve that, the Minimum Description Length principle is used, in its refined form, as seen in Eq. (14).

According to this model, the goodness of fit term corresponds simply to the log-likelihood of the data, the original function that the EM algorithm intended to maximize, given by $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)})$ in Eq. (3).

The parametric complexity term can be given by Eq. (15), but only the second term will be used as a simplification, as was previously mentioned. In this case, the number of degrees of freedom of the model is five times $M$ minus one (due to the five cluster parameters $\alpha$, $\beta_1$, $\beta_2$, $\upsilon$ and $\omega$) since one of the cluster weight terms ($\omega$) is linearly dependant on the others.

The expression that compares the outputs of different initializations of the Expectation-Maximization algorithm is, therefore, given by

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)}) - \frac{1}{2} \log(N) \, (5M - 1) \; . \tag{23}$$

3.2. NML for Clustering PK Responses

An alternative to the regular MDL implemetation, by using the Normalized Maximum Likelihood codelength, was also implemented. The comparison measure in this case is NML's stochastic complexity as seen in Eq. (19).

Again, the goodness-of-fit term is the log-likelihood of the data as stated in Eq. (3). The parametric complexity term, however, was computed using the recursion stated in Algorithm 1 by using the obtained $\mathcal{C}(\mathcal{H}(M), N)$ values for each of the possible number of clusters $M$.

The values of parameters $\lambda_{min}$ and $R$ to be used in this computation should be chosen according to the input data. In the case in study, the covariance matrix of the Gaussian mixture is simply a diagonal matrix with a constant value because the variance in each time instant is independent and assumed to be the same. Therefore, $\lambda_{min}$ is a constant. The value for $R$ can be chosen by using the maximum value of the input data instead of the mean value of its distribution and using its square as the required upper bound. Although it might seem that this implementation has the same problem as the original algorithm that performed merging of clusters by requiring the use of parameters, unlike the original the $\lambda_{min}$ and $R$ parameters are physical and can be estimated from the data.

While computing the value of $\mathcal{C}(\mathcal{H}(k), j)$ for small values of $j$ as required by the recursive algorithm, it is needed to compute values of the multivariate Gamma function $\Gamma_m$ close to or equal to zero. However, the domain of this function is only valid for values greater than $\frac{n-1}{2}$, where $n$ is the number of time instants of the data.

To solve this issue while not greatly affecting the results of the parametric complexity, the output of function $I$ in Eq. (21) is forced to be equal to 1 for values outside its valid domain, thus having its logarithm equate to zero. Doing so means, however, that the output of the algorithm can be unreliable if the number of time instants $n$ is close to the number of subjects $N$.

The resulting expression that is used for comparison is, then, given by

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)}) - \log \mathcal{C}(\mathcal{H}(M), N) \, . \tag{24}$$

3.3. Expectation-Maximization Algorithm

In order to apply the Expectation-Maximization algorithm to the problem at hand and consistently obtain appropriate results, it is necessary to compute several random initializations of it. This is due to the fact that the algorithm can converge to local optima instead of the global optimum solution depending on the initialization. The number of random initializations for each of the possible number of clusters is defined by the user. In addition, the user defines the minimum and maximum number of clusters that the program considers for the data, which should contain the likely actual number of existing clusters.

In each random initialization of the algorithm, every possible number of clusters, from the minimum to the maximum number of clusters, needs to

be tested in order to search for the best clustering solution. The pseudocode described in Algorithm 2 summarizes the execution of one random initialization for each of the possible numbers of clusters. The *score* refers either to MDL or NML, which can be chosen in a user interface.

### 3.4. Cost Efficiency Analysis

The proposed solution to implement the clustering of pharmacokinetic curves requires comparing the outputs of a large set of random initializations for each possible number of clusters within a set range. This process can lead to the resulting program having a very high computational complexity. In order to improve the performance of the program, a parallel implementation was made.

The parallelization method used in the current implementation of the program is the one that divides the load such that each processor takes care of approximately the same number of random initializations for each of the possible numbers of clusters. With this process, it is possible to guarantee the best possible cost efficiency [20], with the load balance being perfect if the user defined number of random initializations is divisible by the number of available processors. In this case, a computer with $p$ processors is able to run the program approximately $p$ times faster than if it was run sequentially.

## 4. Results

In order to verify that the program performs correctly and produces consistent results, it is necessary to conduct a series of tests with varied input data. This section aims to describe the tests that were made by showing the input data of the pharmacokinetic curves of a certain number of subjects during a certain number of time instants and comparing it to the clustering output obtained by the program, for both synthetic and real data.

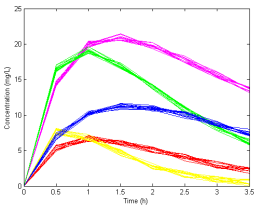
### 4.1. Synthetic Data

The synthetic input data was created by randomly generating the cluster parameters of a given number of clusters and using them to generate the input data of the subjects according to a set cluster distribution and error variance to the true value of the cluster's time function given by its parameters.

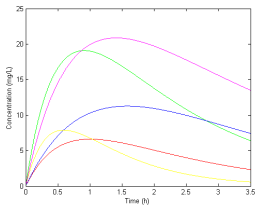
In this section, a side-by-side comparison is made between the input data, on the left, with different colors representing subjects that belonged to different clusters from the program that generated the data, and, on the right, the resulting output cluster functions obtained by the program developed to cluster pharmacokinetic curves using the MDL principle or the NML codelength, which was the subject of this work.

The input data was generated using 60 total subjects with each being sampled at the same eight time instants. To verify the well-functioning of the program, the subjects were generated according to different cluster distributions and numbers of clusters, as well as having low or high error variances. Since the same clusters were obtained for each of the synthetic datasets tested for both the MDL and the NML implementations, only one instance of each test will be displayed. Additionally, in this paper only the results for two datasets will be displayed in Fig. 2, one of which including five imbalanced clusters with high variance. Even under these conditions, the program still obtained the correct cluster functions and subject assignments, which shows the good reliability of the program.

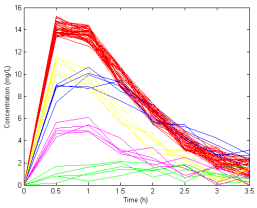
The output cluster functions were the ones obtained by the program using 1 and 10 as the minimu and maximum possible number of clusters and 200 as the number of random initializations used. The maximum number of clusters was chosen to be higher than the probable number of clusters in the data to be able to find the correct result. The number of random initializations was chosen as a lower bound that was consistently able to find the correct solution.



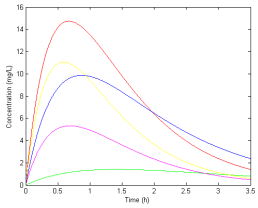
(a) Synthetic input data of 5 balanced clusters with low variance.



(b) Clustering output of data from (a).



(c) Synthetic input data of 5 imbalanced clusters with high variance.



(d) Clustering output of data from (c).

Figure 2: Algorithm output for synthetic data.

#### 4.1.1. Verification of Output Values

To verify that the program makes the correct decisions while comparing different outputs, Table 1 shows as an example the results of a single iteration of the algorithm applied to the input data given in Figure 2(a), using the MDL implementation, where a single random initialization was made for each of the possible numbers of clusters, from one to six. This dataset was chosen only as an example to

**Algorithm 2** Expectation-Maximization Algorithm

```
1: function RUNEM
2:     for each possible number of clusters M from M_min to M_max do
3:         randomly generate initial cluster parameters for M clusters;
4:         compute concentration of each cluster;
5:         compute log-likelihood of each patient data to belong to each cluster;
6:         compute degree of belonging of each patient to each cluster;
7:         update cluster weights ω and variances υ for each cluster;
8:         while maximum iterations is not reached and algorithm has not converged do
9:             update cluster parameters α, β_1 and β_2 for each cluster;
10:            compute concentration of each cluster;
11:            compute log-likelihood of each patient data to belong to each cluster;
12:            compute degree of belonging of each patient to each cluster;
13:            update cluster weights ω and variances υ for each cluster;
14:        end while
15:        assign each patient to the cluster with highest degree of belonging;
16:        compute score value;
17:        if score is higher than current best score value then
18:            save clustering output;
19:        end if
20:    end for
21:    return best clustering output;
22: end function
```

show the relation between the goodness-of-fit and the parametric complexity terms.

In each line, the table shows the $Q$ value and the MDL parametric complexity value corresponding, respectively, to the first and second terms of the comparison expression given in Eq. (23), as well as the resulting sum.

| $M$ | $Q$ value | MDL Par. Comp. | Sum |
|---|---|---|---|
| 6 | -572.166 | -49.132 | -621.298 |
| **5** | **-170.139** | **-40.943** | **-211.082** |
| 4 | -795.751 | -32.801 | -828.552 |
| 3 | -859.248 | -24.566 | -883.814 |
| 2 | -1076.491 | -16.377 | -1092.868 |
| 1 | -1464.166 | -8.189 | -1472.355 |

Table 1: Output values of a single iteration using the input data of five balanced clusters with low variance and MDL.

As expected, the algorithm rewards the models with fewer number of clusters with a lower parametric complexity penalty. However, the best fit (best $Q$ value) was found for five clusters as expected from the generated input data. The balance between these two values was the reason why the algorithm chose five clusters for the data, highlighted in bold, since it had the highest sum.

The same experiment with the same dataset was made with the NML implementation, with the results displayed on Table 2.

Note that the initializations are always random, so it is not possible to find the exact same results for the goodness-of-fit term of each of the possible numbers of clusters.

| $M$ | $Q$ value | NML Par. Comp. | Sum |
|---|---|---|---|
| 6 | -591.554 | -177.783 | -769.337 |
| **5** | **-158.722** | **-173.916** | **-332.638** |
| 4 | -602.393 | -163.462 | -765.855 |
| 3 | -852.163 | -143.993 | -996.156 |
| 2 | -1075.084 | -114.465 | -1189.549 |
| 1 | -1464.602 | -71.479 | -1536.081 |

Table 2: Output values of a single iteration using the input data of five balanced clusters with low variance and NML.

Again, simple models were preferred by NML, with the biggest difference being the fact that the parametric complexity penalty becomes smaller more quickly towards the lowest numbers of clusters while in MDL this progression was linear. Nevertheless, the correct output of five clusters was still found thanks to the goodness-of-fit value.

To show the cost efficiency of the algorithm developed, the execution times using different numbers of processors $p$ for both MDL and NML criteria are shown in Table 3 for the dataset of Figure 2(a). It is visible that the execution time approximately doubles as the number of processors are cut in half, which shows that the parallelization was correctly implemented. These results were obtained using the Ubuntu 14.04 operating system and a processor with 4 cores.

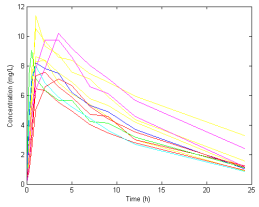
| $p$ | MDL Time [s] | NML Time [s] |
|---|---|---|
| 4 | 80.219 | 90.233 |
| 2 | 161.358 | 170.751 |
| 1 | 272.366 | 277.542 |

Table 3: Execution times for different numbers of processors using the input data from Figure 2(a) with both MDL and NML criteria.

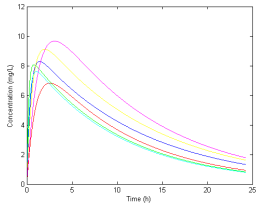
### 4.2. Real Data

It is important to test the algorithm with real data to verify that it performs adequately in real situations. As such, a real dataset was chosen of pharmacokinetic curves from theophylline analysis. The data is shown on Figure 3(a) and it is composed by 12 subjects with samples taken at the same 11 time instants, although not evenly distributed in time.

The algorithm was applied with the same conditions of 200 random initializations and with 10 as the maximum possible number of clusters for both the MDL and NML criteria. The algorithm found six different clusters, as color-coded in Figure 3.




(a) Real input data.



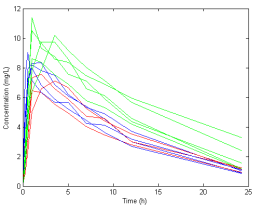

(b) Clustering output of real data.

Figure 3: Algorithm output for real input data with 10 as the maximum number of clusters for both MDL and NML criteria.

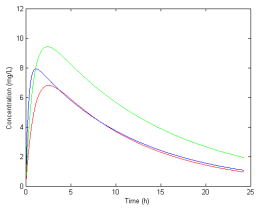
Using the original implementation that did not use MDL it is stated in [2] that the algorithm found three clusters for this dataset. This was due to the fact that three was chosen as the maximum number of clusters to be output by the program, which is an adequate value due to the low amount of observations. If a higher maximum number of clusters is chosen, six is the resulting output. This result was only possible after tuning the cluster merging parameters using arbitrary values, which is not necessary with the MDL and NML criteria.

If three is selected as the maximum number of clusters for the MDL and NML criteria, the result is, again, identical, and is shown in Figure 4.

Six clusters from a dataset of 12 subjects might seem excessive, but these results were caused precisely due to the small sample size of subjects for real data as the influence of the parametric complexity terms is directly related to the number of




(a) Real input data.




(b) Clustering output of real data.

Figure 4: Algorithm output for real input data with 3 as the maximum number of clusters for both MDL and NML criteria.

subjects $N$. This small sample size is also the reason with it is a good idea to choose a lower maximum number of clusters for this particular case, as it not a good idea to allow clusters containing a single observation to exist. Note, however, that these results should now be interpreted by experts with domain knowledge about the drug under study. After that, it should be possible to relate subject features with the elicited groups and predict which cluster a new subject belongs to.

A *Silhouette* measure [21] was calculated as for the synthetic data. The result was approximately $-0.0593$ using the six-cluster solutions and 0.3628 using the three-cluster solutions in a scale from $-1$ to 1 that represents how well the data fits to the clusters. For the datasets in Figure 2 the results were 0.8932 and 0.7012. These results for real data were not very good in comparison, which is to be expected due to the very small number of subjects.

## 5. Conclusions

Starting from an existing solution from [2] that performed the clustering of subjects using only the Expectation-Maximization algorithm, this work was able to provide model selection to determine the output of the clustering algorithm. Doing so means that it is no longer necessary to choose arbitrary values for cluster merging parameters.

The first solution was to use the Minimum Description Length principle, as developed by Rissanen [4], as the measure for comparison between different outputs. By using this measure it became possible to balance the description of a model with its coplexity to obtain the best clustering outputs in a parameter-free manner.

The second solution was one based on the Normalized Maximum Likelihood coding, applied to the case of a Gaussian mixture model as derived in [5], to provide the comparison measure for different clustering outputs. However, this implementation was only reliable when the number of subjects was much larger than the number of time instants recorded for each subject due to domain issues found in its computations. Using this measure

required the usage of two parameters that could at least be estimated from the data, unlike the original solution. The results obtained with NML were not very different from the ones obtained with MDL and did have a sightly longer computation time, so the lack of parameters of MDL can still make it preferable to NML.

For both these solutions, a parallel implementation was made by assigning to each processor identical amounts of work in the form of the number of random initializations of the algorithm. By doing so, it was possible to guarantee the cost efficiency of the program. The final implementation was made with the Java programming language and its code, along with the datasets used for testing, are available at the GitHub repository in [6].

Possible future improvements to the work developed are to test the performance of the program using larger and better real input data, as well as extending the current implementation for different pharmacokinetic models other than the One Compartment Model used for this work. As for model selection, different criteria other than MDL and NML could be tested to try and obtain better performance or results.

**Acknowledgements**

The author would like to thank Prof. Alexandra Carvalho and Prof. Paulo Mateus for the supervision and motivation provided for this work and Elson Tomás for the prior work on the subject of clustering of pharmacokinetic curves that provided the basis for this work.

**References**

[1] J. T. DiPiro, W. J. Spruill, W. E. Wade, R. A. Blouin, and J. M. Pruemer. *Concepts In Clinical Pharmacokinetics*. American Society of Health-System Pharmacists, 4th edition, 2005.

[2] E. Tomás, S. Vinga, and A. M. Carvalho. Unsupervised learning of pharmacokinetic responses. *Computational Statistics*, 32:409–428, 2017.

[3] L. Azzimonti, F. Ieva, and A. M. Paganoni. Nonlinear nonparametric mixed-effects models for unsupervised classification. *Computational Statistics*, 28:1549–1570, 2013.

[4] P. D. Grünwald and J. Rissanen. *The Minimum Description Length Principle (Adaptive Computation and Machine Learning)*. Adaptive Computation and Machine Learning. The MIT Press, 2007.

[5] S. Hirai and K. Yamanishi. Efficient Computation of Normalized Maximum Likelihood Codes for Gaussian Mixture Models with its Applications to Clustering. *IEEE Transactions on Information Theory*, 59:7718–7727, 2013.

[6] R. Guerra. Clustering of pharmacokinetic responses. `https://rjri.github.io/pkclusteringmdl/`, 2017.

[7] A. Rescigno. *Foundations Of Pharmacokinetics*. Springer, 1st edition, 2003.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society, Series B*, 39:1–38, 1977.

[9] J. Bilmes. A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. Technical report, International Computer Science Institute, 1998.

[10] P. D. Grünwald. A tutorial introduction to the Minimum Description Length principle. *CoRR*, 2004. URL `http://arxiv.org/abs/math.ST/0406077`.

[11] C. E. Shannon. A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

[12] P. V. Ming Li. *An Introduction to Kolmogorov Complexity and Its Applications*. Texts in Computer Science. Springer, 3rd edition, 2008.

[13] J. Rissanen. Fisher Information and Stochastic Complexity. *IEEE Transactions on Information Theory*, 42:40–47, 1996.

[14] G. Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, 6:461–464, 1978.

[15] M. H. Hansen and B. Yu. Model Selection and the Principle of Minimum Description Length. *Journal of the American Statistical Association*, 96(454):746–774, 2001.

[16] J. Rissanen. Stochastic Complexity and Modeling. *The Annals of Statistics*, 14(3):1080–1100, 1986.

[17] J. I. Myung, D. J. Navarro, and M. A. Pitt. Model selection by normalized maximum likelihood. *Journal of Mathematical Psychology*, 50(2):167–179, 2006.

[18] D. E. Bell. Regret in Decision Making under Uncertainty. *Operations Research*, 30, 1982.

[19] Y. Shtarkov. Universal sequential coding of individual messages. *(translated from) Problems of Information Transmission*, 23(3):3–17, 1987.

[20] I. Foster. *Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering*. Addison Wesley, 1995.

[21] P. J. Rousseeuw. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.