Received –; | Revised –; | Accepted –

DOI: xxx/xxxx

ARTICLE TYPE

Biexponential fitting for noisy data with error propagation

Paola Lecca*[1] | Michela Lecca[2] | Cecilia Ada Maestri[3] | Marina Scarpa[3]

[1]Faculty of Computer Science, Free University of Bozen-Bolzano, Piazza Domenicani 3, 39100 Bolzano, Italy
[2]Center for Information and Communication Technology, Fondazione Bruno Kessler, Via Sommarive 18, 38123 Trento, Italy
[3]Dipartment of Physics, University of Trento, Via Somamrive 14, 38123 Trento, Italy

Correspondence
*Paola Lecca, Piazza Domenicani 3, 39100 Bozalno,. Email: Paola.Lecca@unibz.it

Summary

Biexponential time-series models commonly find use in biophysics, biochemistry and pharmacokinetics. Indeed, reactions that are described by biexponential functions are typical for many biological processes. The kinetics of these reactions is modelled by transcendental irrational equations interconnecting the reagent concentrations, time and rate constants. The biexponential is apparently a case of non-linear regression, and as such, very often the estimate of its parameters is obtained with methods and software tools performing non-linear fits. The first problem that the user encounters when using these techniques consists in having to provide the software with a not too inaccurate estimate of the intervals within which the parameters can vary. Providing arbitrary initial guesses on the parameter ranges to the non-linear fit procedure causes its non-convergence. The second problem is the need to obtain an estimate of the parameters with an error interval due to the propagation of the experimental error that affects the measurements of the dependent variable. In this study, we propose an extension of a well-established mathematical method based on linearisation techniques of integral equations for the efficient and unsupervised estimation of the parameters of a bi-exponential function. Our extension consists in the integration of a model of error propagation from the measurements of the dependent variable to the parameter estimates. There are three main innovative contributions of this work: 1. having made the unsupervised regression method available in experimental practical applications; 2. having provided methods for error propagation in complex operations, such as integration, matrix inversion and multiplication; 3. the calibration of the biexponential dynamics of (i) water desorption of a small ligand from a surface where two types of binding sites are present, and (ii) of the decrease of a determinant of viability of organs sustaining ischaemic injury before transplantation.

KEYWORDS:
biexponential chemical kinetics, fitting sum of exponentials, error propagation, integro-differential equations, linearisation

1 | INTRODUCTION

Regression studies the relationship between a variable of interest y and one or more explanatory or predictor variables $x = X_1, X_2, \dots, X_n$. The general model is

$$y = F(x_1, x_2, \dots, x_n; \theta_1, \theta_2, \dots, \theta_p) + \epsilon \tag{1}$$

where F is an appropriate function that depends on the predictor variables $x_1, x_2, \dots, x_n$, and parameters $\theta_1, \theta_2, \dots, \theta_p$. In non-linear regression, we use functions F that are not linear in the parameters. In principle, there are unlimited possibilities for F. This flexibility often means a greater effort to make statistical statements on the model.

Given a model, the estimation of its parameters is performed by fitting the model to experimental data. It is well-known that the estimation of model parameters from experimental data is a hard task. This difficulty can be encountered in linear regression, but is particularly insidious in the case of non-linear regression. As reported by Transtrum et al.[1] a non-linear model with tens of parameters, fit, for instance, by non-linear least squares minimization to experimental data, often demands a noticeably time expensive human guidance to find suitable initial guesses of the parameters. Furthermore, even then, accurate, physically plausible estimate of the parameters cannot usually be inferred from the data.

In presence of noise in the data and/or irregular spaced data, both general minimization algorithms[2] and algorithms like the Levenberg-Marquardt algorithm and its recent refinements[3,4], that are designed for least-squares fits routinely explore the parameter space without converge to a solution unless the user instructs them with suitable initial guesses of the parameter values. It is well-known that fitting scattered experimental data with a multi-parameter model by non-linear regression is often hampered by the difficulty in making sufficiently good guesses of the parameters' initial values. Furthermore, typically, algorithms get lost in regions of parameter space in which the model is unresponsive to changes in parameters, and the user is again left to make adjustments by hand. It is worth to say that this is a serious obstacle to progress when one is unsure of the validity of the model.

The biexponential regression, and more in general the fit of a sum of exponentials to numerical data, is a common problem in experimental sciences. The problem consists in approximating the behaviour of a given set of data consisting of n pairs of real numbers (t_i, y_i) by the following equation

$$y(t) = \alpha_0 + \sum_{i=1}^{n} \alpha_i e^{-\beta_i t_i}. \tag{2}$$

where α_i, β_i are unknown real numbers which have to be chosen so that the fit of the Eq. (2) to the data is optimal. If the β_i were known, the problem would be a well posed linear problem, but if the β_i are unknown, the problem turns out to be ill posed[5,6,7]. Fitting Eq. (2) to the data looks a difficult non-linear problem, whose solution is hampered by the difficulties discussed above. However, it has been shown by[8,9], that Eq. (2) can be expressed as a linear combination of powers of t and successive integrals of $y(t)$. In this way, the problem is reduced to a multi-linear regression procedure. More recently, Jacquelin[10] also showed that Eq. (2) can be expressed as a suitable integral equation that can turn a difficult non-linear regression problem into a simple linear regression. The method of linearisation of Jacquelin in[10] was applied to the biexponential case, is unsupervised, i.e. does not ask a priori guesses on parameters, and is particularly suitable for an efficient algorithmic implementation. However, in order to make it usable in real application contexts where the biexponential model is frequently encountered, such as in biology, biochemistry and applied physics, it needs to be extended in such a way as to take into account the experimental errors by which the data are affected.

In this work, we present an extension of the Jaquelin's biexponential regression method. Our extension implements the experimental error propagation in the Jacquelin's method, in order to provide the estimates of the parameters α_i and β_i with their error interval. Moreover we provide a software implementation in R language[11]. The method of error propagation constitutes the main innovative contribution of our study. Error propagation calculation make any procedure of unsupervised fit more usable in practical experimental contexts. Furthermore, the methods that we propose in this paper, unlike previous methods such as[7], provide general procedures for the propagation of the experimental errors in mathematical operations such as derivation, integration, calculation of the determinant, and matrix inversion, that are frequently performed in many parameter inference methods.

At the same time, as already mentioned, the solution of the problem of sum of exponential regression is not only of purely mathematical interest, but is of considerable importance when the independent variable t is the time, as currently occurs in biology. Looking at relatively short-time scale phenomena, a special case is found when the same biological event proceeds following a two-phase model, that is, it results from contemporary fast and slow exponential processes. This is indeed a very common situation, which is encountered looking at molecular transformations, but also in more complex physiological events. In general, a transformation process fits an exponential law when the rate of transformation of the variable is proportional to the present value of the variable itself. A biexponential law means that two independent processes contribute to the transformation, with different time scale. Representative examples of biexponential processes are: a) the kinetics of consumption of a substrate when two different enzymes contribute to the control of the substrate level and first-order kinetic conditions hold. This situation occurs in liver where sulfation and glucoronidation reactions compete for drug removal[12] b) the biexponential dissociation kinetics of a protein which interacts with two distinct receptors such as nucleoporins interacting with nuclear pore complexes[13];

c) the dark-time decays of green fluorescent proteins (GFP) trajectories of fluorescence emission, where blinking and switching contribute with different time scales[14]; d) the behaviour of biological water at the interface of macromolecules and solid surfaces which is a process with two general types of trajectories due to weak interactions[15] with the selected surface sites, and to a stronger interaction producing a rigid water structure[16]; e) the magnetic relaxation times ($T1$, $T2$ and $T1_\rho$) of protons in biological tissues, which depend on different water pools[17] and must be correctly evaluated for quantitative diagnostic NMR imaging; f) the bimodal response of tissues to ischemia which occurs according to a two-phase consumption of high energy metabolites[18,19].

In all the above reported examples the physical process underlying the time course is essentially known and rationally translated into biexponential models. However, there are situations where the behaviour of the system under investigation is not completely understood so that a mathematical model is a high-level abstraction of the reality. This could be the case of conformational changes associated to protein folding[20] or to the biomolecular recognition process[21,22], the dynamic of which fits biexponential models with reasonable, but not *a priori* proved, physical justification.

In this study we applied our implementation of the extended Jacquelin's method to two representative experimental processes whose dynamics is known to be biexponential: a) the desorption of a small ligand from a surface where two types of binding sites are present[15], and b) the decrease of a determinant of viability of organs sustaining ischemic injury before transplantation[19]. The case study b) is particularly challenging because the data are noticeably noisy and and irregularly sampled. In spite of this, we show that even in this limiting case the method is able to give parameters estimates that within their error interval are in agreement with the estimates obtained by well-established supervised method of non-linear fitting. This agreement proves that the method is directly applicable without the need of regularization procedures, that are generally used to mitigate the overfitting in case of data affected by large experimental errors, but that require a careful selection of the best regularization techniques, initial guesses on the parameters, and the multiple solutions of the regularized optimization problem[23].

The paper is organized as follows. In Section 2 we review the theoretical foundation of the Jacquelin's method and of its algorithmic procedure. In Section 3, we present how the experimental errors propagate from the experimental input data to the estimate of the parameters. In Section 4, we present numerical experiments aimed at analysing the response of the methods to an increasing of the experimental error and at determining the maximum value of the experimental error on the input data within which the method is able to give a good estimate of the parameters without crashing due to numerical software errors or instabilities. In Section 5, we report the performances of the method on two representative experimental processes, and finally in Section 6 we draw some conclusions.

2 | FIT OF SUM OF EXPONENTIAL

Here we first summarize the method proposed by J. Jacquelin[10] to perform unsupervised non-linear regression, and then we describe our adaptation to regression of biexponential model from noisy data.

2.1 | Theoretical foundations

Jacquelin[10] showed that a suitable integral equation can turn a difficult non-linear regression problem into a simple linear regression. The Jacquelin's method is based on the following principles of the linearization of the differential and / or integral equations. Given n experimental time points (t_i, y_i) located in proximity of a representative function $y(\vec{\theta}; t)$, where $\vec{\theta} = (a, b, c, \dots)$ is the vector of parameters $a, b, c, \dots$, and t, in our study, is the time variable, we can approximate the integral and the double integral of the $y(u)$ in the following way

$$\int_{t_1}^{t_i} y(u)du \approx S_i \tag{3}$$

where

$$S_i = \begin{cases} 0 & \text{if } i = 1 \\ S_{i-1} + \frac{1}{2}\left(y_i + y_{i-1}\right)\left(t_i - t_{i-1}\right) & i = 2, \dots, n. \end{cases} \tag{4}$$

where $S_i = S(t_i)$, and similarly $y_i \equiv y(t_i)$.

The double integral of $f(u)y(u)$ is thus

$$\int_{t_1}^{t_i}\left[\int_{t_1}^{v} y(u)du\right]dv \approx SS_i \tag{5}$$

$$SS_i = \begin{cases} 0 & \text{if } i = 1 \\ SS_{i-1} + \frac{1}{2}\left(S_i + S_{i-1}\right)\left(t_i - t_{i-1}\right) & i = 2, \dots, n \end{cases} \tag{6}$$

where $SS_i \equiv SS(t_i)$. Consider now a biexponential model for $y(\vec{\theta}; t)$, with $\vec{\theta} = (a, b, c, p, q,)$, i.e.

$$y(\vec{\theta}; t) = a + be^{pt} + ce^{qt}. \tag{7}$$

Applying two successive integrations to $y(\vec{\theta}; t)$, we obtain two formulas, which can be combined with the $y(u)$ formula in order to eliminate the terms e^{pt} and e^{qt}: the function y can be expressed as follows[10]:

$$y(\vec{\theta}; t) = pq \int_{t_1}^{t}\left(\int_{t_1}^{v} y(u)du\right)dv + (p+q)\int_{t_1}^{t} y(u)du + Ct + D \tag{8}$$

where C and D are constants depending on the lower bound of the integral. Jacquelin[10] notes that is possible to analytically express C and D., but the formulas are complicated. Instead of using C and D to compute the approximates of b and c, Jacquelin proposes to implement a complementary linear regression, according to which the Eq. (8) can be re-written as

$$y(\vec{\theta}; t) = C_1 \cdot SS(t) + C_2 \cdot S(t) + C_3 t + C_4 \tag{9}$$

that is a linear equation in the parameters C_1, C_2, C_3, C_4, and

$$C_1 = pq; \quad C_2 = (p+q) \tag{10}$$

$$S(t) = \int_{t_1}^{t} y(u)du \tag{11}$$

$$SS(t) = \int_{t_1}^{t}\left(\int_{t_1}^{v} y(u)du\right)dv \tag{12}$$

where $S(t)$ and $SS(t)$ are estimated via numerical integration with the trapezoidal rule (i.e. Eqs. (4) and (6)). Using Eq. (9), a non-linear regression is turned into a linear regression. Denoting with $\hat{C}_1$ and $\hat{C}_2$ the estimates of C_1 and C_2 obtained from the regression of Eq. (9), the estimates of p and q, denoted respectively with $\hat{p}$ and $\hat{q}$ are

$$\hat{p} = \frac{1}{2}\left(\hat{C}_2 + \sqrt{\hat{C}_2^2 + 4\hat{C}_1}\right) \tag{13}$$

$$\hat{q} = \frac{1}{2}\left(\hat{C}_2 - \sqrt{\hat{C}_2^2 + 4\hat{C}_1}\right) \tag{14}$$

2.2 | The algorithmic procedure

The algorithmic procedure based on the theoretical foundations described above is as follows. Suppose to have n measurements of y at n time point,

$$(y_1, t_1), (y_2, t_2), \dots (y_n, t_n)$$

in increasing order of t_i $(i = 1, \dots, n)$. We introduce the quantities S_i defined as

$$S_1 = 0, \tag{15}$$

$$S_i = S_{i-1} + \frac{1}{2}(y_i + y_{i-1})(t_i - t_{i-1})), \; i = 2, \dots, n \tag{16}$$

and the quantities SS_k defined as

$$SS_1 = 0, \tag{17}$$

$$SS_i = SS_{i-1} + \frac{1}{2}(S_i + S_{i-1})(t_i - t_{i-1})),\ i = 2, \dots, n. \tag{18}$$

In order to solve the regression systems for C_1, C_2 a vector V, and a full rank matrix M are introduced as follows

$$V = \begin{bmatrix} \sum_{i=1}^n SS_i y_i \\ \sum_{i=1}^n S_i y_i \\ \sum_{i=1}^n t_i^2 y_i \\ \sum_{i=1}^n x_i t_i \\ \sum_{i=1}^n y_i \end{bmatrix} \tag{19}$$

$$M = \begin{bmatrix} \sum_{i=1}^n SS_i^2 & \sum_{i=1}^n SS_i S_i & \sum_{i=1}^n SS_i t_i^2 & \sum_{i=1}^n SS_i t_i & \sum_{i=1}^n SS_i \\ \sum_{i=1}^n SS_i S_i & \sum_{i=1}^n S_i^2 & \sum_{i=1}^n S_i t_i^2 & \sum_{i=1}^n S_i t_i & \sum_{i=1}^n S_i \\ \sum_{i=1}^n SS_i t_i^2 & \sum_{i=1}^n S_i t_i^2 & \sum_{i=1}^n t_i^4 & \sum_{i=1}^n t_i^3 & \sum_{i=1}^n t_i^2 \\ \sum_{i=1}^n SS_i t_i & \sum_{i=1}^n S_i t_i & \sum_{i=1}^n t_i^3 & \sum_{i=1}^n t_i^2 & \sum_{i=1}^n t_i \\ \sum_{i=1}^n SS_i & \sum_{i=1}^n S_i & \sum_{i=1}^n t_i^2 & \sum_{i=1}^n t_i & n \end{bmatrix} \tag{20}$$

The vector $C = (C_1, C_2, C_3, C_4, C_5)$ is the product of the inverse of matrix M and the vector V

$$C = M^{-1} V. \tag{21}$$

The coefficient p and q in equation (7) are calculated as

$$p = \frac{1}{2}\left(C_2 + \sqrt{C_2^2 + 4C_1}\right) \tag{22}$$

$$q = \frac{1}{2}\left(C_2 - \sqrt{C_2^2 + 4C_1}\right). \tag{23}$$

Finally, the coefficients a, b, c in equation (7) are given by

$$\begin{bmatrix} a & b & c \end{bmatrix} = Q_1^{-1} Q_2 \tag{24}$$

where

$$Q_1 = \begin{bmatrix} n & \sum_{i=1}^n \beta_i & \sum_{i=1}^n \eta_i \\ \sum_{i=1}^n \beta_i & \sum_{i=1}^n \beta_i^2 & \sum_{i=1}^n \beta_i \eta_i \\ \sum_{i=1}^n \eta_i & \sum_{i=1}^n \beta_i \eta_i & \sum_{i=1}^n \eta_i^2 \end{bmatrix} \tag{25}$$

$$Q_2 = \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n \beta_i y_i \\ \sum_{i=1}^n \eta_i y_i \end{bmatrix} \tag{26}$$

and

$$\beta_i = e^{pt_i}, \quad \eta_i = e^{qt_i}. \tag{27}$$

We implemented the Jacquelin procedure as a R language[11] function `biexponential.fit` reported in Tables A1 -A2 of the Appendix. In the script we used the same nomenclature of the variable also used in the formulae in these sections.

3 | ERROR PROPAGATION

We estimated the errors on the coefficients p, q, a, b, and c that propagated from the experimental uncertainties affecting y_i.

3.1 Errors on coefficients p, q

In order to estimate the errors on p and q, we have to estimate the errors on C_1 and C_2 that appear in the equations defining p and q (Eqs. (22) and (23)). Since from the Eq. (21) $C = M^{-1}V$, we first find an explicit expression for M^{-1}.

The elements of matrix M in (20) are affected by errors that propagated from the experimental errors Δy_i on the measurements y_i. Consequently, each element of M^{-1} is affected by an uncertainty too. Since inverting a $m \times m$ matrix means to solve m linear systems

$$M\mathbf{X}^{(j)} = \mathbf{e}^{(j)} \tag{28}$$

where $\mathbf{X}^{(j)}$ is an array of unknown variables and $\mathbf{e}^{(j)}$ is a vector of the standard basis of $\mathbb{R}$. Since in our case $m = 5$, $j = 1, 2, \dots, 5$, and

$$\begin{aligned}
&\mathbf{e}^{(1)} = [1 \quad 0 \quad 0 \quad 0 \quad 0], \quad \mathbf{e}^{(2)} = [0 \quad 1 \quad 0 \quad 0 \quad 0]\\
&\mathbf{e}^{(3)} = [0 \quad 0 \quad 1 \quad 0 \quad 0], \quad \mathbf{e}^{(4)} = [0 \quad 0 \quad 0 \quad 1 \quad 0]\\
&\mathbf{e}^{(5)} = [0 \quad 0 \quad 0 \quad 0 \quad 1].
\end{aligned}$$

Using the Cramer rule, the i-*th* component of the vector $\mathbf{X}^{(j)}$, say $X_i^{(j)}$, is calculated as

$$X_i^{(j)} = \frac{|M_{(\text{col } i)\leftarrow \mathbf{e}^{(j)}}|}{|M|} \tag{29}$$

where $|\cdot|$ denotes the determinant, $M_{(\text{col } i)\leftarrow \mathbf{e}^{(1)}}$ is the matrix obtained from M, by replacing the i-*th* column of M with the vector $\mathbf{e}^{(j)}$. Therefore, M^{-1} is

$$M^{-1} = \frac{1}{|M|}\begin{bmatrix}
|M_{(\text{col } 1)\leftarrow \mathbf{e}^{(1)}}| & |M_{(\text{col } 1)\leftarrow \mathbf{e}^{(2)}}| & |M_{(\text{col } 1)\leftarrow \mathbf{e}^{(3)}}| & |M_{(\text{col } 1)\leftarrow \mathbf{e}^{(4)}}| & |M_{(\text{col } 1)\leftarrow \mathbf{e}^{(5)}}|\\
|M_{(\text{col } 2)\leftarrow \mathbf{e}^{(1)}}| & |M_{(\text{col } 2)\leftarrow \mathbf{e}^{(2)}}| & |M_{(\text{col } 2)\leftarrow \mathbf{e}^{(3)}}| & |M_{(\text{col } 2)\leftarrow \mathbf{e}^{(4)}}| & |M_{(\text{col } 2)\leftarrow \mathbf{e}^{(5)}}|\\
|M_{(\text{col } 3)\leftarrow \mathbf{e}^{(1)}}| & |M_{(\text{col } 3)\leftarrow \mathbf{e}^{(2)}}| & |M_{(\text{col } 3)\leftarrow \mathbf{e}^{(3)}}| & |M_{(\text{col } 3)\leftarrow \mathbf{e}^{(4)}}| & |M_{(\text{col } 3)\leftarrow \mathbf{e}^{(5)}}|\\
|M_{(\text{col } 4)\leftarrow \mathbf{e}^{(1)}}| & |M_{(\text{col } 4)\leftarrow \mathbf{e}^{(2)}}| & |M_{(\text{col } 4)\leftarrow \mathbf{e}^{(3)}}| & |M_{(\text{col } 4)\leftarrow \mathbf{e}^{(4)}}| & |M_{(\text{col } 4)\leftarrow \mathbf{e}^{(5)}}|\\
|M_{(\text{col } 5)\leftarrow \mathbf{e}^{(1)}}| & |M_{(\text{col } 5)\leftarrow \mathbf{e}^{(2)}}| & |M_{(\text{col } 5)\leftarrow \mathbf{e}^{(3)}}| & |M_{(\text{col } 5)\leftarrow \mathbf{e}^{(4)}}| & |M_{(\text{col } 5)\leftarrow \mathbf{e}^{(5)}}|
\end{bmatrix}. \tag{30}$$

Consequently, according to Eq. (21), the r-*th* component of the vector C is the inner product of the r-*th* row of matrix M^{-1} and the vector V, i.e.

$$\begin{aligned}
C_r &= M^{-1}_{\text{row } r} \cdot V\\
&= |M_{(\text{col } r)\leftarrow \mathbf{e}^{(1)}}| \sum_{i=1}^{n} SS_i y_i + |M_{(\text{col } r)\leftarrow \mathbf{e}^{(2)}}| \sum_{i=1}^{n} S_i y_i\\
&\quad + |M_{(\text{col } i)\leftarrow \mathbf{e}^{(3)}}| \sum_{i=1}^{n} t_i^2 y_i + |M_{(\text{col } r)\leftarrow \mathbf{e}^{(4)}}| \sum_{i=1}^{n} x_i t_i\\
&\quad + |M_{(\text{col } r)\leftarrow \mathbf{e}^{(5)}}| \sum_{i=1}^{n} y_i
\end{aligned} \tag{31}$$

where $r = 1, 2, \dots, 5$. For the sake of convenience, let us introduce $C_r^{(1)}, C_r^{(2)}, C_r^{(3)}, C_r^{(4)}, C_r^{(5)}$ denoting the terms of the sum in Eq. (31), i.e.

$$
\begin{aligned}
C_r^{(1)} &= \frac{|M_{(\text{col } r)\leftarrow \mathbf{e}^{(1)}}|}{|M|}\sum_{i=1}^{n} SS_i y_i \\
C_r^{(2)} &= \frac{|M_{(\text{col } r)\leftarrow \mathbf{e}^{(2)}}|}{|M|}\sum_{i=1}^{n} S_i y_i \\
C_r^{(3)} &= \frac{|M_{(\text{col } i)\leftarrow \mathbf{e}^{(3)}}|}{|M|}\sum_{i=1}^{n} t_i^2 y_i \\
C_r^{(4)} &= \frac{|M_{(\text{col } r)\leftarrow \mathbf{e}^{(4)}}|}{|M|}\sum_{i=1}^{n} x_i t_i \\
C_r^{(5)} &= \frac{|M_{(\text{col } r)\leftarrow \mathbf{e}^{(5)}}|}{|M|}\sum_{i=1}^{n} y_i .
\end{aligned}
$$

Using the error propagation rules, we found that the error on C_r is given by

$$
\Delta C_r = \sqrt{(\Delta C_r^{(1)})^2 + (\Delta C_r^{(2)})^2 + (\Delta C_r^{(3)})^2 + (\Delta C_r^{(4)})^2 + (\Delta C_r^{(5)})^2} \tag{32}
$$

where

$$
\Delta C_r^{(1)} = C_r^{(1)}\sqrt{\frac{\left(\Delta\left(\sum_{i=1}^{n} SS_i y_i\right)\right)^2}{\left(\sum_{i=1}^{n} SS_i y_i\right)^2} + \frac{\left(\Delta|M|\right)^2}{|M|^2} + \frac{\left(\Delta\left(|M_{(\text{col } r)\leftarrow \mathbf{e}^{(1)}}|\right)\right)^2}{|M_{(\text{col } r)\leftarrow \mathbf{e}^{(1)}}|^2}} \tag{33}
$$

$$
\Delta C_r^{(2)} = C_r^{(2)}\sqrt{\frac{\left(\Delta\left(\sum_{i=1}^{n} S_i y_i\right)\right)^2}{\left(\sum_{i=1}^{n} S_i y_i\right)^2} + \frac{\left(\Delta|M|\right)^2}{|M|^2} + \frac{\Delta\left(\left(|M_{(\text{col } r)\leftarrow \mathbf{e}^{(2)}}|\right)\right)^2}{|M_{(\text{col } r)\leftarrow \mathbf{e}^{(2)}}|^2}} \tag{34}
$$

$$
\Delta C_r^{(3)} = C_r^{(3)}\sqrt{\frac{\left(\Delta\left(\sum_{i=1}^{n} t_i^2 y_i\right)\right)^2}{\left(\sum_{i=1}^{n} t_i^2 y_i\right)^2} + \frac{\left(\Delta|M|\right)^2}{|M|^2} + \frac{\Delta\left(\left(|M_{(\text{col } r)\leftarrow \mathbf{e}^{(3)}}|\right)\right)^2}{|M_{(\text{col } r)\leftarrow \mathbf{e}^{(3)}}|^2}} \tag{35}
$$

$$
\Delta C_r^{(4)} = C_r^{(4)}\sqrt{\frac{\left(\Delta\left(\sum_{i=1}^{n} x_i t_i\right)\right)^2}{\left(\sum_{i=1}^{n} x_i t_i\right)^2} + \frac{\left(\Delta|M|\right)^2}{|M|^2} + \frac{\Delta\left(\left(|M_{(\text{col } r)\leftarrow \mathbf{e}^{(4)}}|\right)\right)^2}{|M_{(\text{col } r)\leftarrow \mathbf{e}^{(4)}}|^2}} \tag{36}
$$

$$
\Delta C_r^{(5)} = C_r^{(5)}\sqrt{\frac{\left(\Delta\left(\sum_{i=1}^{n} y_i\right)\right)^2}{\left(\sum_{i=1}^{n} y_i\right)^2} + \frac{\left(\Delta|M|\right)^2}{|M|^2} + \frac{\Delta\left(\left(|M_{(\text{col } r)\leftarrow \mathbf{e}^{(5)}}|\right)\right)^2}{|M_{(\text{col } r)\leftarrow \mathbf{e}^{(5)}}|^2}} \tag{37}
$$

and

$$
\Delta\left(\sum_{i=1}^{n} SS_i y_i\right) = \sqrt{\sum_{i=1}^{n}\left(\Delta(SS_i y_i)\right)^2} = \sqrt{\sum_{i=1}^{n}\left[\left(\frac{\Delta SS_i}{SS_i}\right)^2 + \left(\frac{\Delta y_i}{y_i}\right)^2\right] SS_i^2 y_i^2} \tag{38}
$$

$$
\Delta\left(\sum_{i=1}^{n} S_i y_i\right) = \sqrt{\sum_{i=1}^{n}\left(\Delta(S_i y_i)\right)^2} = \sqrt{\sum_{i=1}^{n}\left[\left(\frac{\Delta S_i}{S_i}\right)^2 + \left(\frac{\Delta y_i}{y_i}\right)^2\right] S_i^2 y_i^2} \tag{39}
$$

$$
\Delta\left(\sum_{i=1}^{n} t_i^2 y_i\right) = \sqrt{\sum_{i=1}^{n}\left(\Delta(t_i^2 y_i)\right)^2} = \sqrt{\sum_{i=1}^{n}\left[\left(\frac{\Delta t_i^2}{t_i^2}\right)^2 + \left(\frac{\Delta y_i}{y_i}\right)^2\right] t_i^4 y_i^2} \tag{40}
$$

$$
\Delta\left(\sum_{i=1}^{n} x_i t_i\right) = \sqrt{\sum_{i=1}^{n}\left(\Delta(x_i t_i)\right)^2} = \sqrt{\sum_{i=1}^{n}\left[\left(\frac{\Delta x_i}{x_i}\right)^2 + \left(\frac{\Delta t_i}{t_i}\right)^2\right] x_i^2 t_i^2} \tag{41}
$$

$$
\Delta\left(\sum_{i=1}^{n} y_i\right) = \sqrt{\sum_{i=1}^{n}\left(\Delta(yi)\right)^2} = \sqrt{\sum_{i=1}^{n}\left[\left(\frac{\Delta y_i}{y_i}\right)^2 y_i^2\right]}. \tag{42}
$$

Finally, in order to propagate the errors on the entries m_{ij} of the matrix M $(i,j = 1,2,\dots,5)$ defined in Eq. (20) to the determinant, we calculated the determinant using the Dodgson condensation method[24,25,26]. According to this method the 5×5 matrix M can be reduced first to a 4×4 matrix whose elements D_{ij} are calculated as the following determinant

$$D_{ij} = \begin{vmatrix} m_{ij} & m_{i\ (j+1)} \\ m_{(i+1)\ j} & m_{(i+1)\ (j+1)} \end{vmatrix}, \quad i,j = 1,2,3,4 \tag{43}$$

then to a 3×3 matrix whose elements D'_{ij} are

$$D'_{ij} = \frac{1}{m_{(i+1)\ (j+1)}} \begin{vmatrix} D_{ij} & D_{i\ (j+1)} \\ D_{(i+1)\ j} & D_{(i+1)\ (j+1)} \end{vmatrix}, \quad i,j = 1,2,3 \tag{44}$$

and finally to a 2×2 matrix whose elements D''_{ij} are

$$D''_{ij} = \frac{1}{D'_{(i+1)\ (j+1)}} \begin{vmatrix} D'_{ij} & D'_{i\ (j+1)} \\ D'_{(i+1)\ j} & D'_{(i+1)\ (j+1)} \end{vmatrix}, \quad i,j = 1,2. \tag{45}$$

The determinant of M calculated with the Dodgson method is

$$|M| = D''_{11}D''_{22} - D''_{12}D''_{21}. \tag{46}$$

The errors on the determinants defined by Eqs (43), (44), and (45) are then

$$\begin{aligned} \Delta D_{ij} = \Bigg\{ \Bigg[m_{ij} m_{(i+1)\ (j+1)} \Bigg(\frac{\Delta m_{ij}}{m_i j} + \frac{\Delta m_{(i+1)\ (j+1)}}{m_{(i+1)\ (j+1)}} \Bigg) \Bigg]^2 \\ + \Bigg[m_{i\ (j+1)} m_{(i+1)\ j} \Bigg(\frac{\Delta m_{i\ (j+1)}}{m_{i\ (j+1)}} + \frac{\Delta m_{(i+1)\ j}}{m_{(i+1)\ j}} \Bigg) \Bigg]^2 \Bigg\}^{\frac{1}{2}}, \quad i,j = 1,2,3,4. \end{aligned} \tag{47}$$

$$\begin{aligned} \Delta D'_{ij} = \Bigg\{ \Bigg[\frac{\Delta m_{(i+1)\ (j+1)}}{m_{(i+1)\ (j+1)}} \Bigg]^2 + \Bigg[D_{ij} D_{(i+1)\ (j+1)} \Bigg(\frac{\Delta D_{ij}}{D_i j} + \frac{\Delta D_{(i+1)\ (j+1)}}{D_{(i+1)\ (j+1)}} \Bigg) \Bigg]^2 \\ + \Bigg[D_{i\ (j+1)} D_{(i+1)\ j} \Bigg(\frac{\Delta D_{i\ (j+1)}}{D_{i\ (j+1)}} + \frac{\Delta D_{(i+1)\ j}}{D_{(i+1)\ j}} \Bigg) \Bigg]^2 \Bigg\}^{\frac{1}{2}}, \quad i,j = 1,2,3. \end{aligned} \tag{48}$$

$$\begin{aligned} \Delta D''_{ij} = \Bigg\{ \Bigg[\frac{\Delta D'_{(i+1)\ (j+1)}}{D'_{(i+1)\ (j+1)}} \Bigg]^2 + \Bigg[D'_{ij} D'_{(i+1)\ (j+1)} \Bigg(\frac{\Delta D'_{ij}}{D'_i j} + \frac{\Delta D'_{(i+1)\ (j+1)}}{D'_{(i+1)\ (j+1)}} \Bigg) \Bigg]^2 \\ + \Bigg[D'_{i\ (j+1)} D'_{(i+1)\ j} \Bigg(\frac{\Delta D'_{i\ (j+1)}}{D'_{i\ (j+1)}} + \frac{\Delta D'_{(i+1)\ j}}{D'_{(i+1)\ j}} \Bigg) \Bigg]^2 \Bigg\}^{\frac{1}{2}}, \quad i,j = 1,2. \end{aligned} \tag{49}$$

Using the above expressions, we can calculate $\Delta D_{11}, \Delta D_{22}, \Delta D_{12}$, and ΔD_{21}, and finally the error on $|M|$, that is

$$\Delta|M| = |M| \left[\sqrt{\left(\left(\frac{\Delta D_{11}}{D_{11}}\right)^2 + \frac{\Delta D_{22}}{D_{22}} \right)^2} + \sqrt{\left(\left(\frac{\Delta D_{12}}{D_{12}}\right)^2 + \frac{\Delta D_{21}}{D_{21}} \right)^2} \right]^{\frac{1}{2}} \tag{50}$$

Finally, using the rules of errors' propagation we find that the errors on p and q are

$$\Delta p = \Delta q = \frac{1}{2} \sqrt{(\Delta C_2)^2 + \frac{C_2^2(\Delta C_2)^2 + 4(\Delta C_1)^2}{C_2^2 + 4C_1}}. \tag{51}$$

3.2 | Errors on coefficients a, b, c

In order to calculate the errors on the coefficients a, b, and c we implemented the same methodology adopted to calculate the errors on p and q. First, we calculate the inverse of Q_1 (given in Eq. (25) using the Cramer rule as shown in the previous subsection. Q_1^{-1} results

$$Q_1^{-1} = \frac{1}{|Q_1|}\begin{bmatrix} |Q_{1\ (\mathrm{col}\ 1)\leftarrow \mathbf{e}^{(1)}}| & |Q_{1\ (\mathrm{col}\ 1)\leftarrow \mathbf{e}^{(2)}}| & |Q_{1\ (\mathrm{col}\ 1)\leftarrow \mathbf{e}^{(3)}}| \\ |Q_{1\ (\mathrm{col}\ 2)\leftarrow \mathbf{e}^{(1)}}| & |Q_{1\ (\mathrm{col}\ 2)\leftarrow \mathbf{e}^{(2)}}| & |Q_{1\ (\mathrm{col}\ 2)\leftarrow \mathbf{e}^{(3)}}| \\ |Q_{1\ (\mathrm{col}\ 3)\leftarrow \mathbf{e}^{(1)}}| & |Q_{1\ (\mathrm{col}\ 3)\leftarrow \mathbf{e}^{(2)}}| & |Q_{1\ (\mathrm{col}\ 3)\leftarrow \mathbf{e}^{(3)}}| \end{bmatrix}, \tag{52}$$

where here $\mathbf{e}^{(1)} = [1 \quad 0 \quad 0]$, $\mathbf{e}^{(2)} = [0 \quad 1 \quad 0]$, and $\mathbf{e}^{(3)} = [0 \quad 0 \quad 1]$. The coefficients a, b, and c then result

$$\begin{aligned} a = \frac{1}{|Q_1|}\Bigg(& |Q_{1\ (\mathrm{col}\ 1)\leftarrow \mathbf{e}^{(1)}}| \sum_{i=1}^{n} y_i + |Q_{1\ (\mathrm{col}\ 1)\leftarrow \mathbf{e}^{(2)}}| \sum_{i=1}^{n} \beta_i y_i \\ & + |Q_{1\ (\mathrm{col}\ 1)\leftarrow \mathbf{e}^{(3)}}| \sum_{i=1}^{n} \eta_i y_i \Bigg) \end{aligned} \tag{53}$$

$$\begin{aligned} b = \frac{1}{|Q_1|}\Bigg(& |Q_{1\ (\mathrm{col}\ 2)\leftarrow \mathbf{e}^{(1)}}| \sum_{i=1}^{n} y_i + |Q_{1\ (\mathrm{col}\ 2)\leftarrow \mathbf{e}^{(2)}}| \sum_{i=1}^{n} \beta_i y_i \\ & + |Q_{1\ (\mathrm{col}\ 2)\leftarrow \mathbf{e}^{(3)}}| \sum_{i=1}^{n} \eta_i y_i \Bigg) \end{aligned} \tag{54}$$

$$\begin{aligned} c = \frac{1}{|Q_1|}\Bigg(& |Q_{1\ (\mathrm{col}\ 3)\leftarrow \mathbf{e}^{(1)}}| \sum_{i=1}^{n} y_i + |Q_{1\ (\mathrm{col}\ 3)\leftarrow \mathbf{e}^{(2)}}| \sum_{i=1}^{n} \beta_i y_i \\ & + |Q_{1\ (\mathrm{col}\ 3)\leftarrow \mathbf{e}^{(3)}}| \sum_{i=1}^{n} \eta_i y_i \Bigg). \end{aligned} \tag{55}$$

Again, using the rules of error propagation, we found that the errors of a, b, and c are

$$\Delta a = \sqrt{\left(\Delta a^{(1)}\right)^2 + \left(\Delta a^{(2)}\right)^2 + \left(\Delta a^{(3)}\right)^2} \tag{56}$$

where

$$\begin{aligned} \Delta a^{(1)} = & \frac{|Q_{1\ (\mathrm{col}\ 1)\leftarrow \mathbf{e}^{(1)}}| \sum_{i=1}^{n} y_i}{|q_1|} \times \\ & \times \sqrt{\left(\frac{\Delta |Q_{1\ (\mathrm{col}\ 1)\leftarrow \mathbf{e}^{(1)}}|}{|Q_{1\ (\mathrm{col}\ 1)\leftarrow \mathbf{e}^{(1)}}|}\right)^2 + \left(\frac{\Delta\left(\sum_{i=1}^{n} y_i\right)}{\sum_{i=1}^{n} y_i}\right)^2 + \left(\frac{\Delta |Q_1|}{|Q_1|}\right)^2} \end{aligned} \tag{57}$$

$$\begin{aligned} \Delta a^{(2)} = & \frac{|Q_{1\ (\mathrm{col}\ 2)\leftarrow \mathbf{e}^{(1)}}| \sum_{i=1}^{n} y_i}{|q_1|} \times \\ & \times \sqrt{\left(\frac{\Delta |Q_{1\ (\mathrm{col}\ 2)\leftarrow \mathbf{e}^{(1)}}|}{|Q_{1\ (\mathrm{col}\ 2)\leftarrow \mathbf{e}^{(1)}}|}\right)^2 + \left(\frac{\Delta\left(\sum_{i=1}^{n} \beta_i y_i\right)}{\sum_{i=1}^{n} \beta_i y_i}\right)^2 + \left(\frac{\Delta |Q_1|}{|Q_1|}\right)^2} \end{aligned} \tag{58}$$

$$\begin{aligned} \Delta a^{(3)} = & \frac{|Q_{1\ (\mathrm{col}\ 3)\leftarrow \mathbf{e}^{(1)}}| \sum_{i=1}^{n} y_i}{|q_1|} \times \\ & \times \sqrt{\left(\frac{\Delta |Q_{1\ (\mathrm{col}\ 3)\leftarrow \mathbf{e}^{(1)}}|}{|Q_{1\ (\mathrm{col}\ 3)\leftarrow \mathbf{e}^{(1)}}|}\right)^2 + \left(\frac{\Delta\left(\sum_{i=1}^{n} \eta_i y_i\right)}{\sum_{i=1}^{n} \eta_i y_i}\right)^2 + \left(\frac{\Delta |Q_1|}{|Q_1|}\right)^2} \end{aligned} \tag{59}$$

where

$$\Delta\Big(\sum_{i=1}^{n}\beta_i y_i\Big) = \sqrt{\sum_{i=1}^{n}\beta_i^2 y_i^2\Big[\Big(\frac{\Delta\beta_i}{\beta_i}\Big)^2 + \Big(\frac{\Delta y_i}{y_i}\Big)^2\Big]} \tag{60}$$

$$\Delta\Big(\sum_{i=1}^{n}\eta_i y_i\Big) = \sqrt{\sum_{i=1}^{n}\eta_i^2 y_i^2\Big[\Big(\frac{\Delta\eta_i}{\eta_i}\Big)^2 + \Big(\frac{\Delta y_i}{y_i}\Big)^2\Big]} \tag{61}$$

$$\Delta\beta_i = \Big|t\beta_i\Big|\Delta p \tag{62}$$

$$\Delta\eta_i = \Big|t\eta_i\Big|\Delta q. \tag{63}$$

The error on the determinant of matrix Q_1 is calculated by exploiting the Dodgson condensation methods as reported in Section . So, here for the sake of brevity we omit the details of its calculation.

In the Supplementary Material (Tables A1 -A2) we provide scripts in R code implementing the biexponential regression method with error propagation.

4 | ANALYSIS ON SYNTHETIC DATA

We show the performance of the method on a synthetic case study. We considered a biexponential curve in Eq. (7) with $a = 0.3$, $b = c = 1$, $p = -1$, $q = -0.1$. In order to study the response of the algorithm to the increase of the experimental error, we choose the value of q much larger than the value of p. In this way, we place ourselves in the condition in which the difference between the exponents is very large and therefore the eventual poor perfomance of the procedure cannot be attributed to the impossibility of discriminating with great accuracy between two close values.

In order to mimic real experimental data we added to this curve normal noise with mean zero and variance ranging form 0 to 0.12 in steps of 0.001. 0.12 is the upper limit for the input data standard deviation beyond which the algorithm implementing the Jacquelin method (see Supplementary Material Tables A1 -A2) warns the user that a calculation produced NaN values:

```
Warning: In sqrt(C[2]^2 + 4 * C[1]) : NaNs produced.
```

For each value of the standard deviation of the noise we simulated in this way 500 noisy biexponential curves, and for each curve we estimated the values of the parameters. In Figures 1 and 2 , we report the behaviour of the mean parameter estimates and its standard error, respectively, as the standard deviation of noise (plotted on x-axis) changes.

The way in which parameter estimates deviate from the correct value is different from parameter to parameter. In Figure 1 we observe that the average of the estimate of p, q and of c is an increasing monotonic function of the standard deviation of the data noise. The averages of the estimates of a and b exhibit a slight downward trend close to the real value for standard deviations of the data noise less than 0.06, and once this value is exceeded they become unstable showing considerable fluctuations. In Figure 2 we observe an increasing monotonic behaviour of the standard error on the parameter estimates for p and c, an increasing trend of the standard error on q while the standard deviation of data noise is less than 0.08 followed by a slight downward trend. The standard error on a and b shows significant fluctuation for standard deviation of data noise greater than 0.06. Finally, in Figure 2 we observe an exponential increasing of the residual sum of squares as the standard deviation of the data noise increases. Further experiments (not reported here) done by progressively increasing the value of parameter q, (keeping all the others constant) till to reach the value of $p = -1$, gave very similar results. In summary, regardless of the value of ratio between p and q, the threshold of tolerance of this method to experimental noise on values of y - exponentially decreasing in the range $[0, 2]$ (as defined by the choice of $a = 0.3$, $b = c = 1$, $p = -1$, $q = -0.1$) - is around 0.06. For a standard deviation of data noise equal to 0.06, the means of the parameter estimates are still fairly close to value of the parameters initially chosen to generate the curve, i.e $\overline{p} \approx -0.09$, $\overline{q} \approx -0.65$, $\overline{a} \approx 0.3$, $\overline{b} \approx 0.87$, and $\overline{c} \approx 1.1$.

5 | CASE STUDIES

Herein we tested the code on two cases taken from the personal experience of the authors: a) the dynamic of desorption of a small ligand from a surface where two types of binding sites are present[15]; b) the decrease of a determinant of viability of

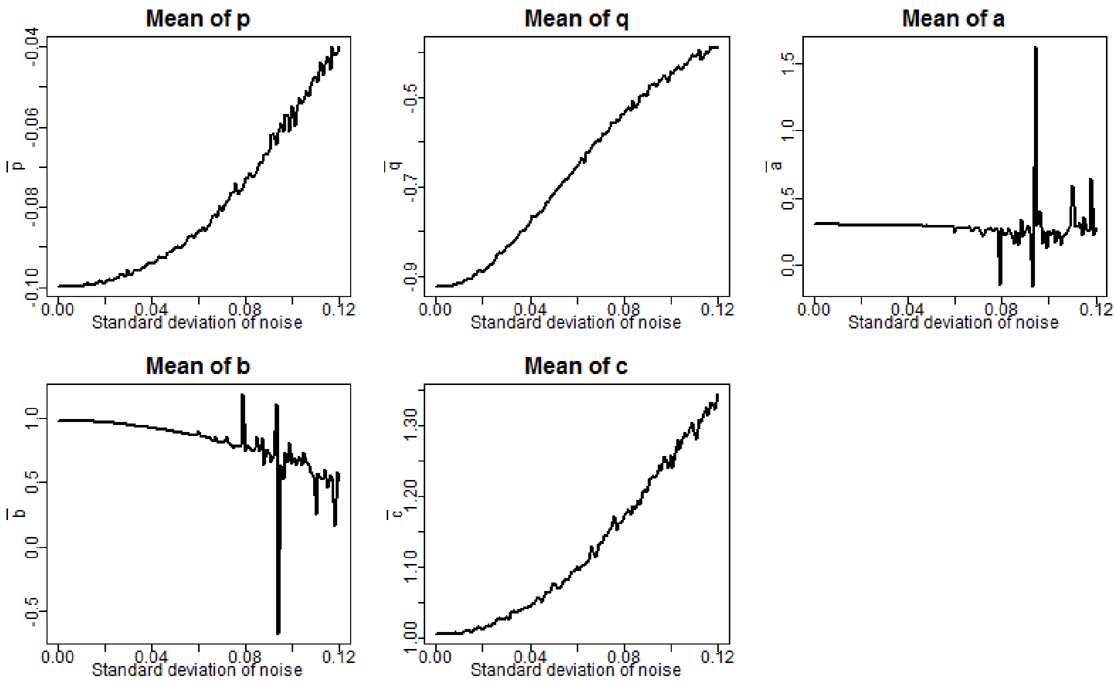

FIGURE 1 Mean of the parameters p, q, a, b, and c estimates of the model obtained in 500 simulated noisy biexponential $y = a + be^{-pt} + ce^{-qt} + \xi$, where $\xi \sim \mathcal{N}(0, \sigma^2)$, with $\sigma^2 \in [0, 0.12]$.

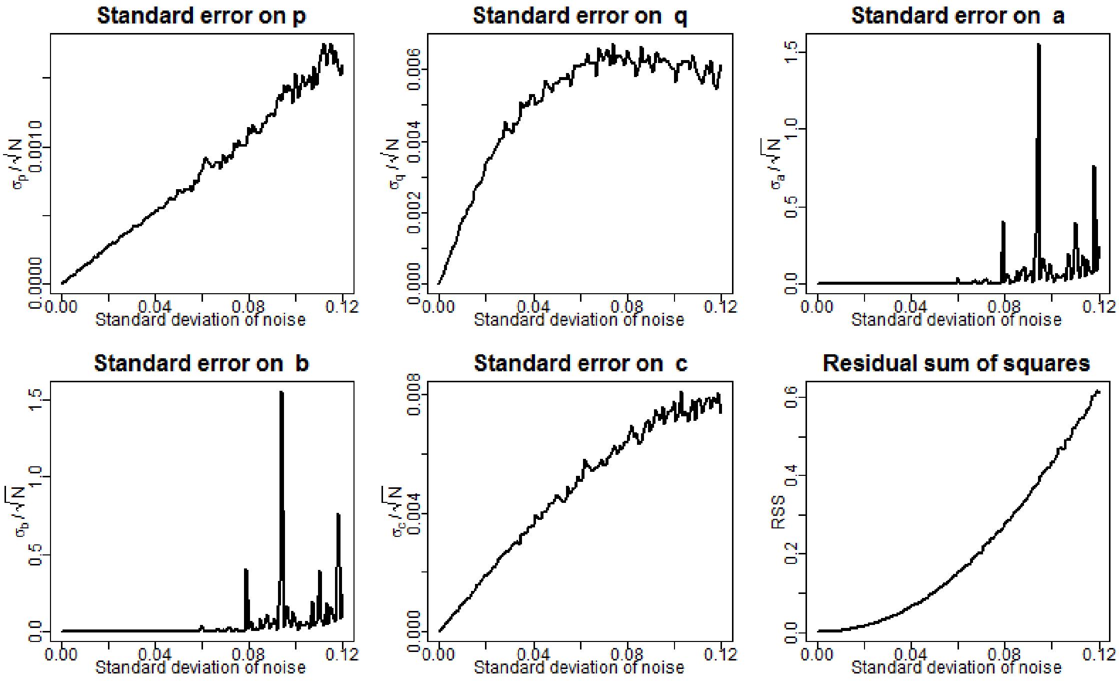

FIGURE 2 Standard errors on the estimate of the parameters p, q, a, b, and c and deviance (RSS) of the model obtained in 500 simulated noisy biexponential $y = a + be^{-pt} + ce^{-qt} + \xi$, where $\xi \sim \mathcal{N}(0, \sigma^2)$, with $\sigma^2 \in [0, 012]$.

organs sustaining ischemic injury before transplantation[19]. In both these case studies the rate at which the monitored variable is transformed is proportional to the amount which is left, according to a first-order law, and two mechanisms at different time scales are responsible for the transformation. Accordingly, the experimental data fit a double exponential law. These two case studies differ in the amount of experimental error on the measurements.

5.1 | Water desorption

Case study a) is representative example of the behaviour frequently observed when ligands bind to macromolecules or molecules and nanoparticles undergo adsorption on a solid surface. Information about the kinetics of these mechanisms contributes to the understanding of fundamental biological or physical processes such as: gene expression patterns, which are regulated by the binding of transcription factors to high and low affinity sites[27]; metabolic control mechanisms[28], or the forces determining absorption or desorption at the interface. The experimental data used in case study a) concern the dynamic of heavy water (D_2O) desorption from a film made of nanosized cellulose fibrils (NC) which fits a double exponential time course. The experiment was performed in two steps: a dry NC film was saturated with vapors of D_2O, which was then left to desorb. Infrared spectroscopy (FTIR) was used to monitor the time course of decrease of the characteristic spectral features of D2O during desorption. The time steps between the data points was determined by the time necessary to acquire the FTIR spectrum with a good signal to noise ratio (> 5) and was of the order of 1 min. The experimental details and data are reported in[15]. From these data, it appears that D_2O desorbs from NC film according to a double-stage process, the first of which ends in about 15 min, supporting the presence of two types of D_2O binding sites with different values of kinetic time constants of release. The desorption kinetic data shown in Figure 3 were satisfactorily fitted to a double exponential decay, and the exponential time constants calculated by a set of four replicated experiments using different NC films were found $\tau_1 = (2.2 \pm 0.9)$ min and $\tau_2 = (50 \pm 24)$ min, respectively .

Figure 3 shows the biexponential fit of the experimental data obtained with the linearization method. The parameters of the fit are reported in Table 1 (first row). These values within their error intervals are in agreement with those obtained with the software Origin® by OriginLab[29].

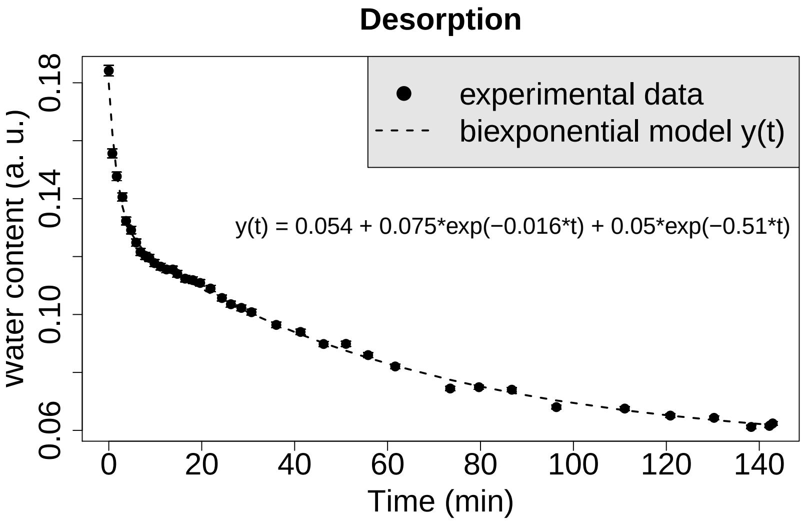

FIGURE 3 Dotted lines is the curve fitting the experimental data. The parameter estimates of the biexponential fit for the desorption dynamics are p =-0.016, q = -0.511, a = 0.0542, b = 0.075, c = 0.050. RSS = 0.00012. The experimental relative error amounts to 1% of the measurement, so the errors bars are scarcely visible on this plot.

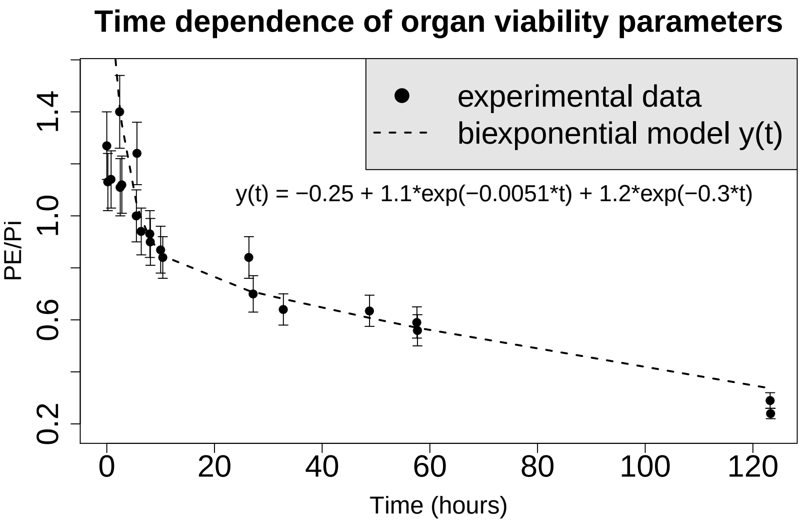

FIGURE 4 Dotted lines is the curve fitting the experimental data. The parameter estimates of the biexponential fit for the high energy metabolite decay in kidney are $a = -0.25$; $b = 1.1$; $c = 1.2$; $p = -0.0051$; $q = -0.3$. The residual sum of squares is 0.38.

5.2 | High energy metabolite decay in organs stored for transplantation

Case study b) is the time course of physiological response to ischemic injury of explanted kidneys. The parameter monitored to assess the residual viability of the organs is the ratio between a class of high energy metabolites (i.e. the phosphate monoesters, PE) and the product of of consumption of these metabolites, that is inorganic phosphate (Pi). The PE/Pi ratio has been demonstrated to correlate with viability in human recipients and is considered a good determinant of the postoperative function[30,31]. The double exponential time course of the PE/Pi ratio is indicative of two types of classes of the energy consumption reactions: one faster, probably due to membrane protein pumps which require high energy amounts but are progressively denatured, and one slower due to all the other metabolic processes. The PE/Pi can be experimentally measured on whole explanted organs by ^{31}P Nuclear Magnetic Resonance Spectroscopy (^{31}P NMR). The data set here used is a representative example of what is often obtained performing experiments on whole organs or living beings, when the intrinsic biological variability and the complexity of the overall experimental set-up are responsible for the acquisition of scattered results. Case study b) was used for stress testing the software under conditions of noisy experimental data. In particular, in case study b), the source of noise is due to: 1) the intrinsic biological variability between different subjects and between the two kidney explanted from the same subject; 2) the experimental procedure which requires the organ transfer from the container for storage to the measurement site inside the magnet of the nuclear magnetic resonance instrument; 3) the time scale of the experiment which lasts several days. In fact, to monitor the time dependence of the PE/Pi ratio of the kidneys excised from a pig on cold storage, the two organs of each pig were kept in a cold solution and transferred inside the magnet, one by one, at selected time intervals. The time the kidney remained inside the instrument was about 20 min for each experimental point. This is source of uncertainty, since during this time the storage conditions are not controlled, and the kinetic of PE/Pi decrease could be consequently affected. This uncertainty is more remarkable at the kinetic start when the PE/Pi decrease rate is higher. Moreover, due to the long time scale of the experiment, it is not possible to record the kinetic end since biological specimens undergo degradation process due to bacterial contamination. The experimental data of case study 2) are shown in Figure 4 . These data refer to kidneys excised just after flushing the organ with a cold solution and stored immersed in a cold solution. Since the organs did not suffer for warm ischemia before explantation, the starting PE/Pi values are in the highest range. The values and the error bars reported in the plot are the average and the standard deviation of two experimental runs (one run for the left and one for the right kidney from the same pig). The time course of the PE/Pi decay is visibly biphasic and the standard deviation between each measurement point performed on the two kidneys is about 40% for the faster phase and decrease progressively up to 20%. The kinetic was stopped at 120 h, when visible signs of organ suffering were observed. The kinetic constants p and q calculated using a data set of 10 kinetic runs (two kidneys from 5 pigs) were 0.100 h^{-1} and 0.010^{-1}, respectively with an estimated error of 40% and 20% for the faster and slower phase, respectively.

Figure 4 shows the biexponential fit of the experimental data of this case study. The parameters of the fit are reported in Table 1 (second row). Also in this case the mean values we obtained are in agreement with those obtained by SigmaPLot non-linear fitting procedure[32] by Corazza et al.[19]. We note however, the considerable error that affects the estimates of a, b, and q. This second case study is in fact a limiting case for this procedure, as the experimental error on the input data exceeds the threshold of tolerance 0.06 of an order of magnitude.

TABLE 1 Parameters and residual sum of squares of the biexponential fit for the desorption and PE/Pi temporal behaviours. For the case study of high energy metabolite decay, in spite of the considerable errors affecting the input data, the mean values of the estimate are in agreement within their interval errors with those obtained with the software Origin® by OriginLab[29], and SigmaPlot fitting program[32]. Some notes follow. (1) The parameter estimates have been calculated with OriginLab[29] by[15]. The error on the estimates is the standard error (SE), as usually estimated in non-linear fitting procedures, i.e. $SE(\theta_i) = \sqrt{(RSS/DF)\texttt{Cov}(i,i)}$, where RSS is the sum of squared residual, DF denotes the degrees of freedom (the number of data points minus number of parameters fit by regression), and Cov(i, i) the diagonal element of the covariance matrix of best-fit parameters. (2) The parameter estimates have been calculated with SigmaPlot fitting program by[19]. The estimates of a, b, and c are indicated as "NA" (not available), because they are not present in the model of the biphasic kinetics presented in[19].

Case study	p (in hours^{-1})	q (in hours^{-1})	a	b	c	RSS
			Linearization method			
Desorption	0.016 ± 0.001	0.511 ± 0.001	0.0542 ± 0.001	0.075 ± 0.001	0.050± 0.001	0.00012
High energy metabolite decay	0.0051 ± 0.0015	0.3 ± 0.16	-0.2 ± 0.12	1.1 ± 0.1	1.2 ± 0.1	0.38
			Non-linear fit			
Desorption	0.016 ± 0.001$^{(1)}$	0.5753 ± 0.0006$^{(1)}$	0.053 ± 0.002$^{(1)}$	0.076 ± 0.002$^{(1)}$	0.051± 0.002$^{(1)}$	not reported$^{(1)}$
High energy metabolite decay$^{(2)}$	0.007 ± 0.0014	0.74 ± 0.3	NA	NA	NA	not reported

6 | CONCLUSIONS

We have presented an extension of the Jacquelin linearisation method for the parameters of a bi-exponential. Our extension consists of a procedure that allows us to calculate the propagation of experimental errors from the measurements of the variable dependent on the estimates of the parameters obtained with this method. The inclusion of the error propagation allows to alert the user about the adequacy of the experimental data for the purpose of estimating the kinetic parameters. The analysis on synthetic data allowed to establish a tolerance threshold on the experimental error, beyond which the linearisation procedure provides unreliable estimates of the parameters. The application of the method to real case studies made it possible to assess its efficiency and accuracy. The proposed method, of a part of which (e.g. the parameter estimation mode) we present an implementation in R language, becomes in this way usable in the laboratory practices concerning the analysis of the acquisitions, where it is not possible to establish suitable initial guesses on parameters and it is desirable not to use computational time with iterative regression methods, and, finally, where it is necessary to monitor the propagation of errors on the estimates of the parameters of a model. The proposed method in this way, usable in the laboratory practices concerning the analysis of the acquisitions in terms of odds where it is not possible to establish a suitable interval of variation of the parameters and we do not want to use computational time with iterative regression methods, and where it is necessary to monitoring the propagation of experimental errors.

ACKNOWLEDGMENTS

The authors thank Ana Maria Alonso Rodriguez of the Department of Mathematics of the University of Trento (Italy), for here advices and for the careful revision of the manuscript.

Author contributions

All authors in their field of expertise made a substantial, direct and intellectual contribution to the work, and approved it for publication. In particular, PL developed and implemented the mathematical methods for the error propagation from the data to the estimate of the coefficients of the bi-exponential fit. ML made the code efficient by suggesting the most appropriate implementation choices for this purpose, in particular as regards the propagation of the error in the inversion of a matrix. CAM and MS contributed to the conception of the problem, the writing of the manuscript, and provided the information for the two case studies.

Financial disclosure

None reported.

Conflict of interest

The authors declare no potential conflict of interests.

□

APPENDIX

A THE CODE

We implemented a R script that execute the biexponential regression by calling the function `biexponential-fit.R` (Tables A1 -A1). To run the script the user must install R software and the libraries `matrixcalc`, `stinepack`, open a command prompt window and launch the command

```
Rscript biexponential-fit.R
```

The script takes as an input a .TXT file with three columns named `x`, `y`, `error`, where `x` is the independent variable (on x-axis), `y` is the dependent variable (on y-axis), and `error` is the experimental error of `y`, usually its standard deviation.

TABLE A1 The R script implementing the biexponenital fit algorithm of J. Jacques[10]. The scripts call the functions `biexponential.fit` and `error.propagation` implemented in `biexponential_functions.R`.

```
#####################################################################
#                  Call to the biexponential function               #
#####################################################################

# Include libraries
library(MASS, matrixcalc, stinepack)

# Include the functions "biexponential.fit" and "error.propagation"
source("./biexponential_functions.R)

# Ask the user to type the input data file
cat("Enter input file name:  ");
input <- readLines("stdin",1);

input.df <- read.table(input, header=T)
x1 <- input.df$x; y1 <- input.df$y

sdev <- input.df$error
adds <- seq(tail(input.df[,1],1)+0.1, 2*tail(input.df[,1],1), 0.1)

# plot the data
plot(x1, y1, ylim=range(c(y1-sdev, y1+sdev)), pch=19,
xlab="Measurements",
ylab="Mean +/- SD", main="Scatter plot with std.dev error bars")
arrows(x1, y1-sdev, x1, y1+sdev, length=0.05, angle=90, code=3)

# Perform the bi-exponetial fit and error propagation
biexponential.fit(x1, y1))
error.propagation
#####################################################################
```

TABLE A2 The R script implementing the function `biexponential.fit` proposed by[10].

```
#####################################################################
# R function: biexponential.R
#####################################################################

#  function implementing the biexponential fit
biexponential.fit <- function(x, y)
{
print("Bi-exponential model: y(x) = a + b*exp(p*x) + c*exp(q*x)")

n <- length(y); S <- array(0, n); SS <- array(0, n)
S[1] <-  0; SS[1] <- 0

for (k in 2:n)
{S[k] <- S[k-1] + 0.5 * (y[k] + y[k-1]) *(x[k] - x[k-1])
SS[k]<- SS[k-1] + 0.5 * (S[k] + S[k-1]) *(x[k] - x[k-1])}

# build matrix M and vector V
M.row.1 <- c(sum((SS)^2), sum(SS*S), sum(SS*x^2), sum(SS*x), sum(SS))
M.row.2 <- c(sum(SS*S), sum((S)^2), sum(S*x^2),  sum(S*x), sum(S))
M.row.3 <- c(sum(SS*x^2), sum(S*x^2), sum(x^4), sum(x^3), sum(x^2))
M.row.4 <- c(sum(SS*x), sum(S*x), sum(x^3), sum(x^2), sum(x))
M.row.5 <- c(sum(SS), sum(S), sum(x^2), sum(x), n)

M <- rbind(A.row.1, A.row.2, A.row.3, A.row.4, A.row.5)
V <- (c(sum(SS*y), sum(S*y), sum((y*x^2)), sum(x*y), sum(y)))

# compute vector C
C <- solve(M) %*% V

# coefficients p and q
p <- 0.5*(C[2] + sqrt(C[2]^2 + 4*C[1]))
q <- 0.5*(C[2] - sqrt(C[2]^2 + 4*C[1]))

# build matrices Q1 and Q2
beta <- exp(p*x); eta <- exp(q*x)

Q1 <- rbind(c(n, sum(beta), sum(eta)),
c(sum(beta), sum(beta^2), sum(beta*eta)),
c(sum(eta), sum(beta*eta), sum(eta^2)))
Q2 <- c(sum(y), sum(beta*y), sum(eta*y))

abc <- solve(Q1) %*% Q2

# calculate coefficients a, b, and c
a <- abc[1]; b <- abc[2]; c <- abc[3]
res <- c(p, q, a, b, c); names(res) <- c("p", "q", "a", "b", "c")

y.model <- res["a"] + res["b"]*exp(res["p"]*x) + res["c"]*exp(res["q"]*x)
deviance <- sum((y - y.model)^2)

# return coefficients, p, q, a, b, and c
res.final <- c(p, q, a, b, c, deviance)
names(res.final) <- c("p", "q", "a", "b", "c", "deviance")
res.final
}

#######################################################################
```

References

1. Transtrum Mark K., Machta Benjamin B., Sethna James P.. Why are Nonlinear Fits to Data so Challenging?. *Phys. Rev. Lett.*. 2010;104:060201.

2. Nocedal Jorge, Wright Stephen J.. *Numerical Optimization*. Springer; 2006.

3. Pujol J. The solution of nonlinear inverse problems and the Levenberg-Marquardt method. *Geophysics.* 2007;SEG. 72:W1–W16.

4. Kanzow Christian, Yamashita Nobuo, Fukushima Masao. Levenberg–Marquardt methods with strong local convergence properties for solving nonlinear equations with convex constraints. *Journal of Computational and Applied Mathematics.* 2004;172(2):375 - 397.

5. Kaufmann Bernhard. *Fitting a Sum of Exponentials to Numerical Data.* 2003.

6. Kundu Debasis, Mitra Amit. Fitting a Sum of Exponentials to Equispaced Data. *Sankhyā: The Indian Journal of Statistics, Series B (1960-2002).* 1998;60(3):448–463.

7. Groen P. De, Moor B. De. The fit of a sum of exponentials to noisy data. *Journal of Computational and Applied Mathematics.* 1987;20:175 - 187.

8. Squire William. A simple integral method for system identification. *Mathematical Biosciences.* 1971;10(1):145 - 148.

9. Diamessis J.E.. Least-square-exponential approximation. *Electronics Letters.* 1972;8(18):454.

10. Jacquelin Jean. *Regressions et equations integrales.* https://www.scribd.com/doc/14674814/Regressions-et-equations-integrales; accessed: 2018-09-01; 2009.

11. R-project . *The R Project for Statistical Computing.* Accessed: 2018-09-01; 2019.

12. Morris M. E., Pang K. S.. Competition between two enzymes for substrate removal in liver: modulating effects due to substrate recruitment of hepatocyte activity. *J Pharmacokinet Biopharm.* 1987;15(5):473–496.

13. Rabut Gwénaël, Doye Valérie, Ellenberg Jan. Mapping the dynamic organization of the nuclear pore complex inside single living cells. *Nature Cell Biology.* 2004;6(11):1114–1121.

14. Dickson Robert M., Cubitt Andrew B., Tsien Roger Y., Moerner W. E.. On/off blinking and switching behaviour of single molecules of green fluorescent protein. *Nature.* 1997;388(6640):355–358.

15. Bettotti Paolo, Maestri Cecilia Ada, Guider Romain, et al. Dynamics of Hydration of Nanocellulose Films. *Advanced Materials Interfaces.* 2015;3(5):1500415.

16. Menon Rajiv G., Sharafi Azadeh, Windschuh Johannes, Regatte Ravinder R.. Bi-exponential 3D-T1$_rho$ mapping of whole brain at 3 T. *Scientific Reports.* 2018;8(1).

17. Pal S. K., Peon J., Zewail A. H.. Biological water at the protein surface: Dynamical solvation probed directly with femtosecond resolution. *Proceedings of the National Academy of Sciences.* 2002;99(4):1763–1768.

18. Kalogeris Theodore, Baines Christopher P., Krenz Maike, Korthuis Ronald J.. Ischemia/Reperfusion. *Compr Physiol.*. 2017;7(1):113–170.

19. Corazza A., Vianello F., Rigo A., et al. Biphasic behavior of the kinetics of 31P-containing metabolites in ischemic porcine kidneys. *Transplantation Proceedings.* 2003;35(8):3111–3115.

20. Yang Wei Yuan, Gruebele Martin. Folding at the speed limit. *Nature.* 2003;423(6936):193–197.

21. Schmidt Hayden R., Betz Robin M., Dror Ron O., Kruse Andrew C.. Structural basis for σ_1 receptor ligand recognition. *Nature Structural & Molecular Biology.* 2018;25(10):981–987.

22. Payne Marvin A., Igo John D., Cao Zhenghua, Foster Samuel B., Newton Salete M. C., Klebba Phillip E.. Biphasic Binding Kinetics between FepA and Its Ligands. *Journal of Biological Chemistry.* 1997;272(35):21950–21955.

23. Gábor Attila, Banga Julio R.. Robust and efficient parameter estimation in dynamic models of biological systems. *BMC Systems Biology.* 2015;9(1).

24. Rice Adrian, Torrence Eve. Lewis Carrol's Condensation Method for eavluating determinant. *Math Horizons.* 2006;14(2):12–15.

25. Abeles Francine F.. Dodgson condensation: The historical and mathematical development of an experimental method. *Linear Algebra and its Applications.* 2008;429(2):429 - 438.

26. Main Mitch, Donor Micah, Harwood R. Corban. *An Elementary Proof of Dodgson's Condensation Method for Calculating Determinants.* 2016.

27. Wassarman Paul M. *Essays on Developmental Biology Part B (Current Topics in Developmental Biology Book 117).* Academic Press; 2016.

28. Frieden Carl. Protein oligomerization as a metabolic control mechanism: Application to apoE. *Protein Science.* 2019;28(4):837–842.

29. OriginLab . *Regression and Curve Fitting.* https://www.originlab.com/doc/Origin-Help/Regression-Curve-Fitting; 2019.

30. Peter Shawn D St, Imber Charles J, Friend Peter J. Liver and kidney preservation by perfusion. *The Lancet.* 2002;359(9306):604–613.

31. Hene R. J., Grond J., Boer W. H., Mali W. P., Koomans H. A.. Pre-transplantation assessment of renal viability with 31P magnetic resonance spectroscopy. *Kidney Int..* 1994;46(6):1694–1699.

32. SigmaPlot . *Curve fitting % regression analysis.* http://www.sigmaplot.co.uk/products/sigmaplot/curvefitting.php; 2019.

33. Ristanović D., Ristanović D., Milutinović B., Malešević J., Milin J.. A routine fitting of kinetic data to sums of exponentials with a programmable calculator. *Computers in Biology and Medicine.* 1984;14(3):363 - 376.

34. Secrier Maria, Schneider Reinhard. Visualizing time-related data in biology, a review. *Briefings in Bioinformatics.* 2013;15(5):771–782.

35. Phillips . *Physical Biology of the Cell.* Garland Science; 2012.

AUTHOR BIOGRAPHY

Paola Lecca. Paola Lecca received the master's degree in theoretical physics and the PhD degree in computer science and telecommunications. She has worked for several years since its inception as a researcher and principal investigator at the Microsoft Research-University of Trento, Centre for Computational and Systems Biology, where she led the Knowledge inference and Data Management research team. She was then a researcher at the Department of Cellular, Computational and Integrative Biology and then at the Mathematics Department of the University of Trento where she carried out studies of network theory and dynamical systems applied to biology. Currently she is Assistant Professor at the Faculty of Computer Science of the Free University of Bozen-Bolzano, Italy. Her main research activities focus on computational modelling and algorithmic procedures implementing efficient solutions for identiafiability, controllability, and simulation of complex dynamical networks. The main applicative domains of these studies are network biology, biochemistry, biological physics, microbiology, and synthetic biology.

Michela Lecca. Michela Lecca received the Ms. Degree in Mathematics from University of Trento in 2002. In the same year she joined the Research Unit Technologies of Vision, Fondazione Bruno Kessler, where she is currently working as a Permanent Researcher. Her research interests include mathematical models for image processing and computer vision applications with particular attention to unsupervised object recognition and image retrieval, machine colour constancy, and their implementation on smart embedded systems. Michela Lecca authored several research papers published in international journals and conference proceedings. She has taken part in the program committee of many editions of Applied Computing Machine Symposia, and carries on a reviewing activity for several international conferences and journals. She is a member of the International Association IAPR - CVPL.

Cecilia Ada Maestri. After a master degree in Experimental Physics, she gained a PhD in Biotechnology at University of Trento, Italy. Her main research interest is the study of bio-nano-materials, in particular nanocellulose, and their possible exploitation environmental or biomedical field.

Marina Scarpa. Marina Scarpa is currently a Full Professor of Biophysics at the University of Trento. She received a Master degree in Chemistry in 1982 and a Ph.D. in Molecular Biology and Pathology from the University of Padua. Her research activity has been focused on the understanding of the functional processes in biological systems that take place at the molecular level, particularly when these phenomena occur at the nanoscale, by using spectroscopy and multiscale simulations. In 2005 she joined the Nanoscience Laboratory of the University of Trento. The main, mission of this research laboratory is to generate new knowledge and develop spin-off applications from physical and biophysical phenomena associated with photons and their interactions with matter.