Bioinactivation SE
User Manual

April 2017

Version 0.1.1

Document Revisions

Date	Version Number	Document Changes
26/04/2016	0.1.0	Initial Draft

Table of Contents

1 Abbreviations and symbols

t Elapsed time since inoculation.

N Microbial count.

N_0 Microbial count at the moment of inoculation.

S Fraction of survivors (N/N_0).

T Temperature.

T_{ref} Reference temperature.

$\log_{10}$ Decimal logarithm.

D D-value of Bigelow model.

D_R D-value at the reference temperature.

z z-value of Bigelow, Mafart and Geeraerd model.

b Parameter of Peleg model.

n Parameter of Peleg model.

k_b Parameter of Peleg model.

T_c Critical temperature of Peleg model.

δ Scale factor of the Weibull distribution for Mafart model (δ-value).

δ_R δ-value at the reference temperature.

p Shape factor of the Weibull distribution for Mafart model.

k_{max} Maximum rate of inactivation for Geeraerd model.

k Instantaneous rate of inactivation for Geeraerd model.

α Parameter of Geeraerd model including the shoulder effect.

γ Parameter of Geeraerd model including the tail effect.

C_c Dummy substance limiting the microbial inactivation in Geeraerd model.

N_{res} Residual microbial count in Geeraerd model.

2 Introduction

Industrials usually apply dynamic heat treatments to food products in order to ensure their health safety. However, the lack of in-deep knowledge of the response of the pathogens to the sterilization treatments applied usually results in unnecessary overprocessing which reduces the organoleptic properties of the final product.

Current consumer demands for minimally processed products (fresh-like) require the optimization of the heat processes applied to the products, while guaranteeing health safety. This can only be accomplished with a detailed knowledge of the response of the microbial populations to the treatments applied.

Predictive microbiology serves a key role in this aspect. This branch of the microbiology tries to model the evolution of the microbial cells under known environmental conditions using mathematical models. However, the parameters of these models cannot be determined analytically. Thus, experiments in laboratory conditions must be performed in order to estimate their values. Consequently, inactivation curves cannot be predicted with complete certainty. Instead, predictions intervals at the desired confidence level must be generated.

Published research have shown that non-isothermal heat treatments like those commonly applied in industry can only be described using inactivation models whose model parameters have been estimated using dynamic experiments. However, the estimation of model parameters using from the dynamic experimental data requires their determination directly from the system of differential and algebraic equations. This requires advanced mathematical procedures.

Currently, there is no software package available for the estimation of the parameters of inactivation models from dynamic experimental data. Hence, research teams must develop in-house software or restrict their studies to isothermal experiments with an analytical solution.

The bioinactivation package for the R programming language (Garre, Fernández, & Egea, 2015) implements several features for the modeling of microbial inactivation either isothermal or non-isothermal. It includes functions for the estimation of model parameters from isothermal or non-isothermal data, as well as for the prediction of the inactivation process (inactivation curves or prediction intervals). Nevertheless, knowledge of programming in R is required to access these functions.

Bioinactivation SE (Simplified Environment) provides a simplified user interface to selected functions of the bioinactivation package, namely the estimation of model parameters from non-isothermal experiments and the generation of prediction intervals of the inactivation

process. Therefore, it provides tools for modeling dynamic inactivation in a user-friendly way which can be used by researchers or industrials without programming or mathematics expertise.

The layout of Bioinactivation SE has been developed using the *shiny* package of R (Chang, Cheng, Allaire, Xie, & McPherson, 2015). This package provides a framework for the creation of web applications which can run R code. Hence, the user does not need to have R installed in his computer to use Bioinactivation SE. The only requirements are an internet connection and a web browser. Hence, it can even be used using a smartphone.

Bioinactivation SE is hosted in the following link:

https://opada-upct.shinyapps.io/bioinactivation_SE/

A permanent link to this page is kept on the Department of Food Microbiology of the Universidad Politécnica de Cartagena, under the "Recursos" ("Resources") section.

Bioinactivation SE can also be used on computers without internet connection. In this case, however, the software must be installed locally. Please contact the maintainers of the software as indicated in the section Contact.

This document is structured as follows:

- Chapter 3 lists the features of the bioinactivation package that have been included in Bioinactivation SE.
- Chapter 4 describes the requirements of the software regarding the unit system of the input data.
- Chapter 5 presents the structure of the tool and its different components, describing the function of each of the widgets and the output generated.
- Chapter 6 describes how write citations to Bioinactivation SE when used in scientific papers.
- Chapter 7 provides the contact data of the maintainers of the application.
- Chapter 8 includes the articles referenced in the text.
- Appendix A: Inactivation models included in Bioinactivation SE presents the constitutive equations of the inactivation models included in the application.
- Appendix B: Common error messages and known bugs includes a list of common error messages encountered when working with Bioinactivation SE and how to solve them.
- Appendix C: Example input file provides a valid input file for the tool to be used as training material.

3 Features included in Bioinactivation SE

Bioinactivation SE implements selected functions of the bioinactivation package for R, which is publicly in the Comprehensive R Archive Network (CRAN) under the following link:

https://cran.rstudio.com/web/packages/bioinactivation/index.html

The version of the package used by the tool is indicated in the *About* page of Bioinactivation SE.

Bioinactivation SE includes the following features:

- Model fitting of dynamic inactivation experiments using non-linear regression.
- Model fitting of dynamic inactivation experiments using an MCMC algorithm.
- Output of the adjusted curve as a text file.
- Estimation of values, standard deviations and confidence intervals of model parameters.
- Estimation of correlations between model parameters.
- Calculation of errors measurements (SSE, MSE and RMSE) of the curve fitted to a set of dynamic inactivation experiments.
- Generation of prediction intervals of the dynamic inactivation process based on the adjustment performed.

4 Units

Bioinactivation SE does not require the user to input the values in any specific unit system. The output is generated in the same unit system defined by the user.

5 Layout of Bioinactivation SE

The layout of Bioinactivation SE has been design trying to guide the natural workflow followed during the modelling of dynamic inactivation:

1. Data input.
2. Visualization of the data imported.
3. Setting of the adjustment parameters.
4. Model fitting.
5. Visualization of the adjusted curve.
6. Visualization of the statistical information of the adjustment.
7. Generation of prediction intervals.
8. Output of the adjusted curve for further post-processing.

Figure 1 illustrates the overall organization of the user interface of Bioinactivation SE. The navigation bar on top defines the mayor separation between the different tasks: the *Data input and the actual Model fitting*. Moreover, an *About* page with miscellaneous information regarding the tool is provided. The space below is separated in a lateral panel and a main panel. The lateral panel contains the widgets which can be manipulated by the user, while the main panel is used to show the results.

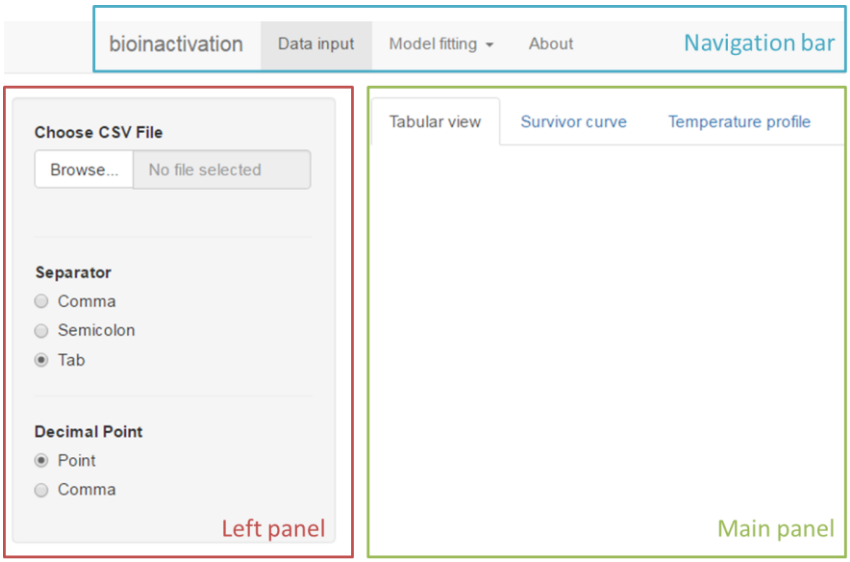


Figure 1: Overview of the tool layout

5.1 *Data input* page

The *Data input* page is the one shown when the user opens Bioinactivation SE. In this section, the user can upload the data of inactivation experiments to the tool for further analysis. Figure 2 depicts the layout of this page as it is first loaded by the browser. Clicking the button ① allows importing a text file with the experimental results. It has to fulfill the following requirements:

- It must be a plain text file.
- Every data point must be included in one single file.
- It has to be in tabular form.
- The first line is reserved for column headers.
- It must contain one column named *time* (with the elapsed time since inoculation), one named *N* (with the microbial count) and one named *temp* (with the experimental temperature).
- Every row has the same number of elements.
- Missing values must be labeled as *NA*.

Appendix C: Example input file provides an example of a valid input file to be used as training material.

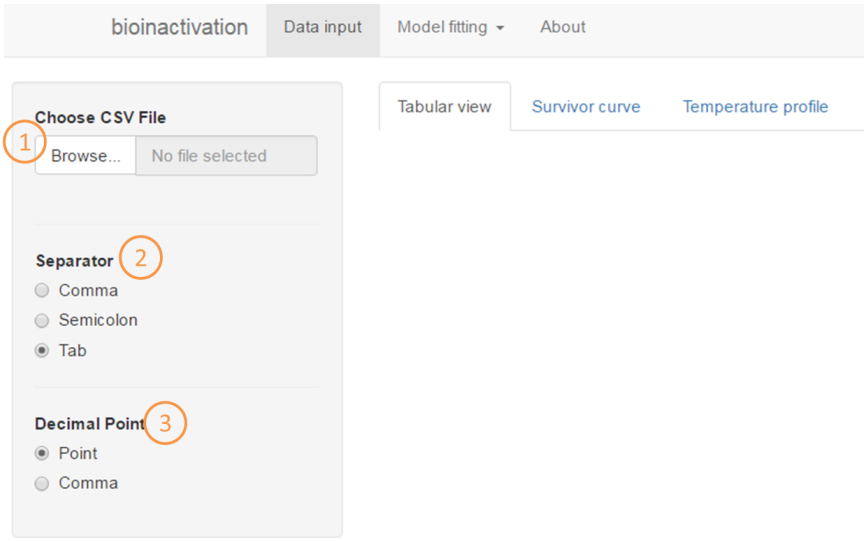


Figure 2: Layout of the *Data input* page as it is loaded by the web browser

The columns can be separated either by commas, semicolons or tabs. The separator chosen must be selected through the radio buttons in ②. The character used for separating the decimal points can be selected in ③.

After a file has been successfully imported, a message is shown below he button, as depicted in Figure 3. The results imported can be visualized in the main panel, in tabs ①, ② and ③. Tab ① (default state) shows the results imported in tabular form. Note that an additional column, named *log.UFC,* is added with the logarithmic microbial count.

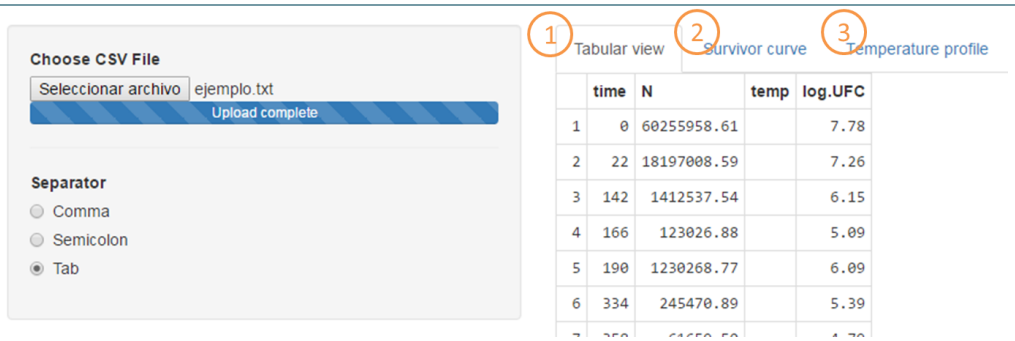


Figure 3: State of the *Data input* page after data has been imported in the tool

Tab ② provides the survivor curve of the data imported, as shown in Figure 4**.** Lastly, tab ③ plots the temperature profile defined by the data imported (Figure 5).

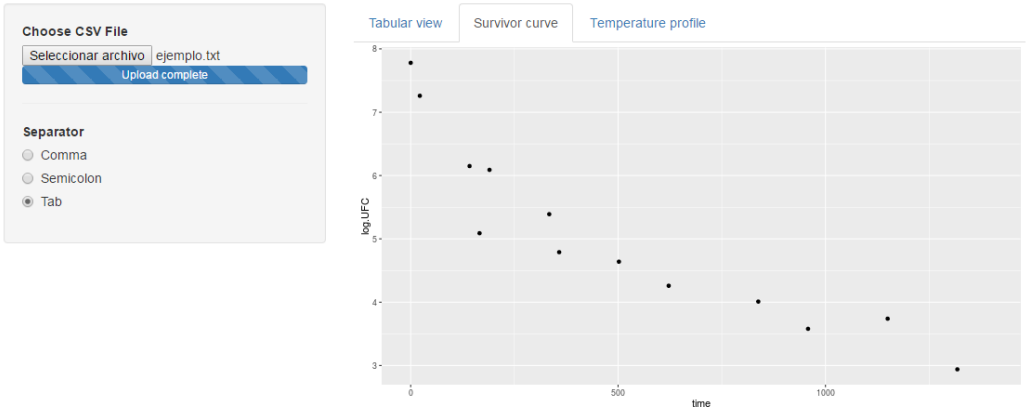


Figure 4: Survivor curve illustrating the results imported in the tool

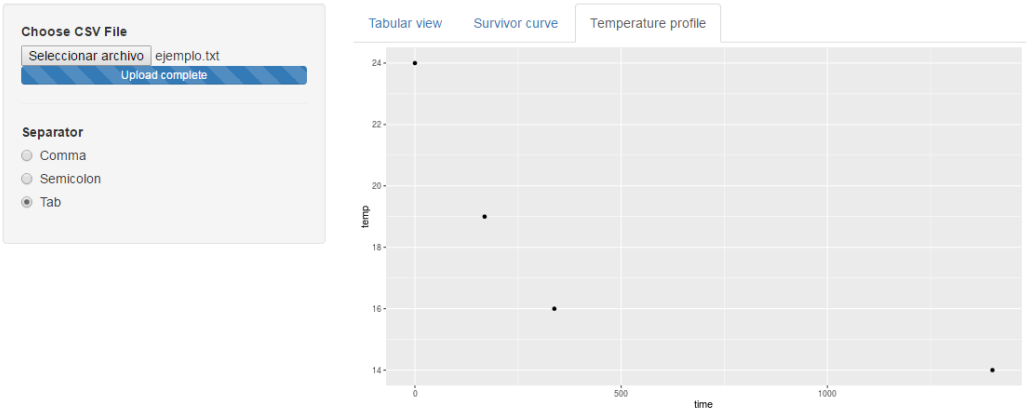


Figure 5: Plot illustrating the temperature profile imported

5.2 Model fitting page

Clicking the button *Model fitting* pops-up a drop down list showing the inactivation models included in Bioinactivation SE. Bigelow, Peleg, Mafart and Geeraerd model are currently

included. Appendix A: Inactivation models included in Bioinactivation SE describes the constitutive equations of each model.

Clicking on any inactivation model opens a page similar to the one shown in Figure 6Figure 5. Above the lateral panel, a label ① shows the inactivation model which has been selected. The lateral panel of the *Model fitting* section is divided in three tabs: *Model parameters* ②, *Fitting parameters* ③ and *Make adjustment* ④.

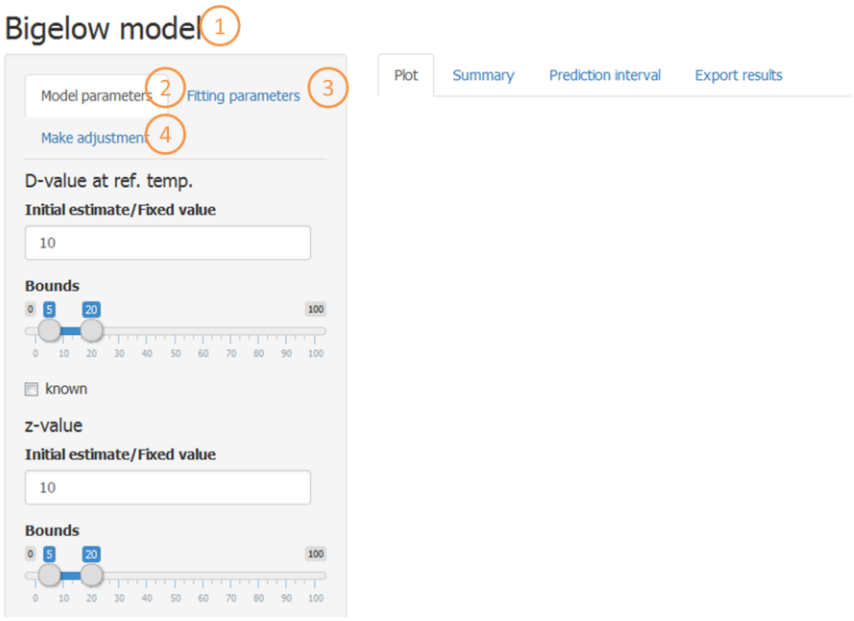


Figure 6: Layout of the model fitting page

The model fitting algorithms implemented in Bioinactivation SE require the definition of initial guesses for the model parameters. The *Model parameters* page serves this purpose. This page is made out of the repetition of the four widgets illustrated in Figure 7. These widgets are repeated once for each parameter of the inactivation selected. The label ① indicates the model parameter affected by the widgets below. The initial estimate of the model parameter is set using the numeric input ②, while the slider ③ is used for defining upper and lower bounds for the adjustment. Lastly, Bioinactivation SE allows the user to consider any model parameter of the model as known. If the the checkbox ④ is ticked, the model parameter is fixed to its initial estimate (i.e. not adjusted to the experimental data). Definition of unrealistic initial estimates and bounds for the model parameters will, most likely, result in the fitting algorithm failing to converge. Care is advised when setting them.

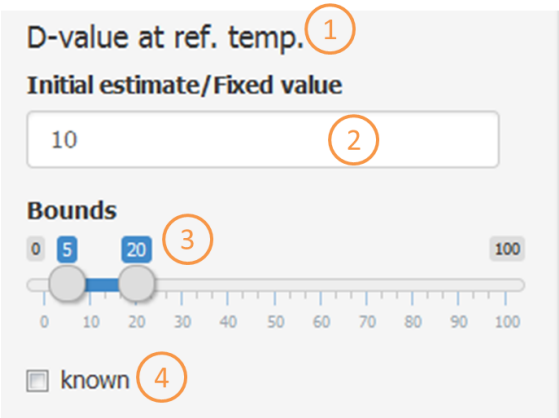


Figure 7: Group of four widgets forming the *Model parameters* page

Several parameters of the adjustment algorithm can be set in the tab *Fitting parameters*. Figure 8 illustrates the widgets included in this page. Bioinactivation SE includes two different fitting algorithms: non-linear regression (nlr) and a Markov Chain Monte Carlo algorithm (MCMC). The algorithm to use can be selected in the drop-drown box ①. The number of iterations required for convergence of the nlr algorithm is set by the software and does not require any input by the user. However, the MCMC requires the user to define the number of iterations of the algorithm to perform using slider ②. Due to the nature of the algorithm, omitting the first iterations of the MCMC algorithm can accelerate its convergence to a solution. The number of iterations to omit is defined through slider ③. Bioinactivation SE is able to generate a prediction interval for the inactivation process based on the results of the adjustment. The probability levels at which the prediction interval will be calculated is defined using slider ④.

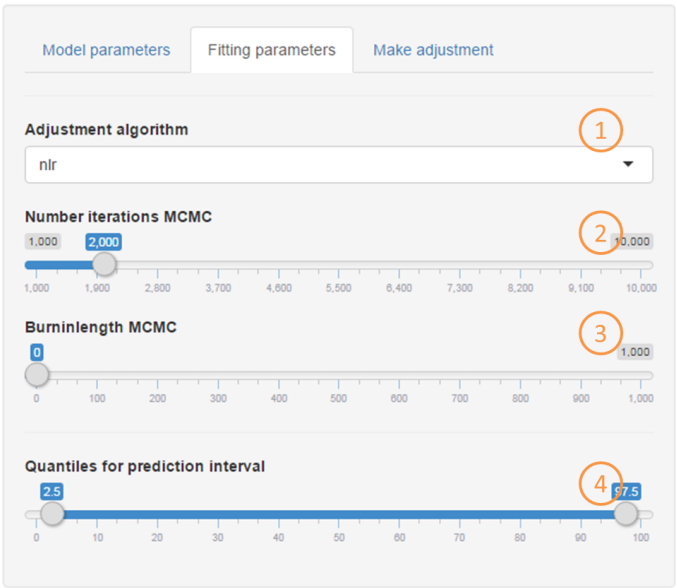


Figure 8: Widgets included in the *Fitting parameters* tab

Once the settings are defined, the model adjustment can be launched in the *Make adjustment* tab. This tab contains the button ①, shown in Figure 9, which launches the model fitting. The upper right part of the application will show a message while the adjustment process is taking place.

The MCMC algorithm is an heuristic procedure based on random numbers. As such, the results of different runs may produce different results. In order to be able to reproduce the results, Bioinactivation SE provides the button ② to reset state of the internal pseudo-random number generator in the *Reset state* page. Clicking this button just before performing the model adjustment will generate results that can be reproduced by any user.

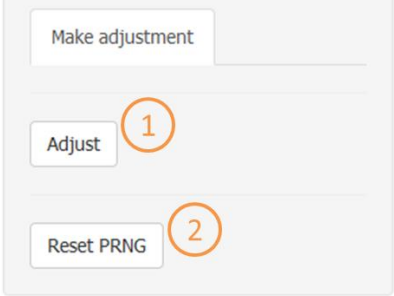


Figure 9: Contents of the *Make adjustment* tab

The results of the adjustment are provided in the main panel of the *Model fitting* page, which contains four tabs *Plot*, *Summary*, *Prediction interval* and *Export results*. The data provided here uses the same unit system introduced by the user. The *Plot* tab provides a visual comparison between the adjusted curve and the experimental results provided. An example of this plot is shown in Figure 10. This plot may show indications of a poor adjustment, raising the need to modify the initial estimates or the settings of the adjustment algorithm.

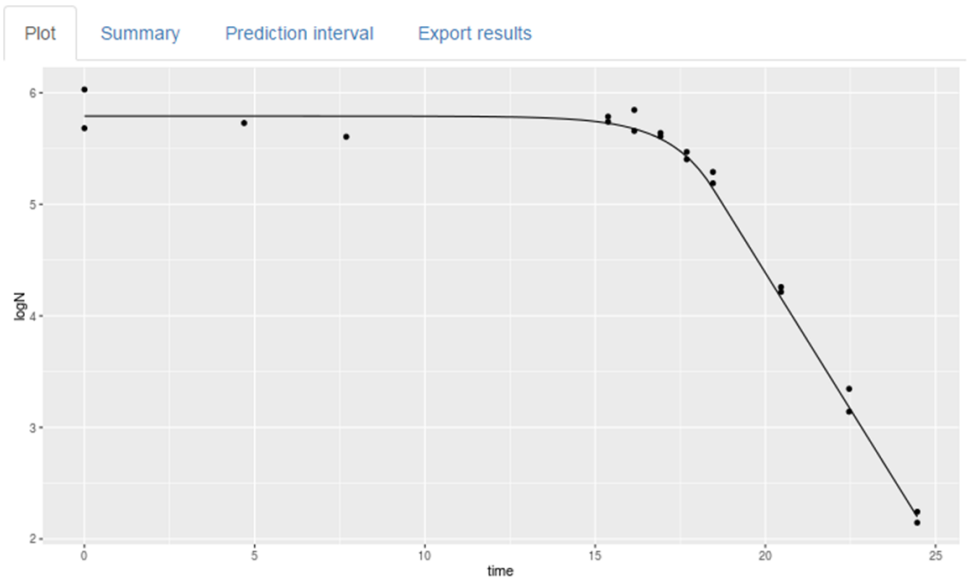


Figure 10: Plot comparing the adjusted model with the experimental results

Figure 11 illustrates the summary of the adjustment provided by Bioinactivation SE in tab *Summary*. This tab contains three different tables, labeled in this Figure as ①, ② and ③. Table ① provides the estimated values, standard deviations and confidence intervals at the 0.95 confidence level for each one of the model parameters adjusted. Model parameters which were fixed are not shown in this table. Note that the hypothesis for the generation of the confidence intervals differs depending on the adjustment algorithm selected. Intervals for the nlr algorithm are generated assuming that the model parameters follow a t-distribution with the mean and standard deviations estimated by the adjustment algorithm. This may result on negative lower bounds for the model parameters. On the other hand, intervals for the MCMC algorithm are calculated based on the distribution of the model parameters through the individual iterations of the adjustment algorithm.

Table ② provides basic information regarding the residuals of the adjusted curve. Namely the Sum Squared Error (SSE), Mean Squared Error (MSE) and Root Mean Squared Error (RMSE). Finally, table ③ contains the estimated parameter correlation.

The columns corresponding to the standard deviation and the confidence intervals in table ①, as well as the parameter correlation in table ③ may appear blank when the nlr algorithm is used. This is caused by an extremely high correlation between the model parameters, making the hessian of the system non-invertible. There are two solutions for this issue: using the MCMC algorithm or fixing some of the model parameters adjusted.

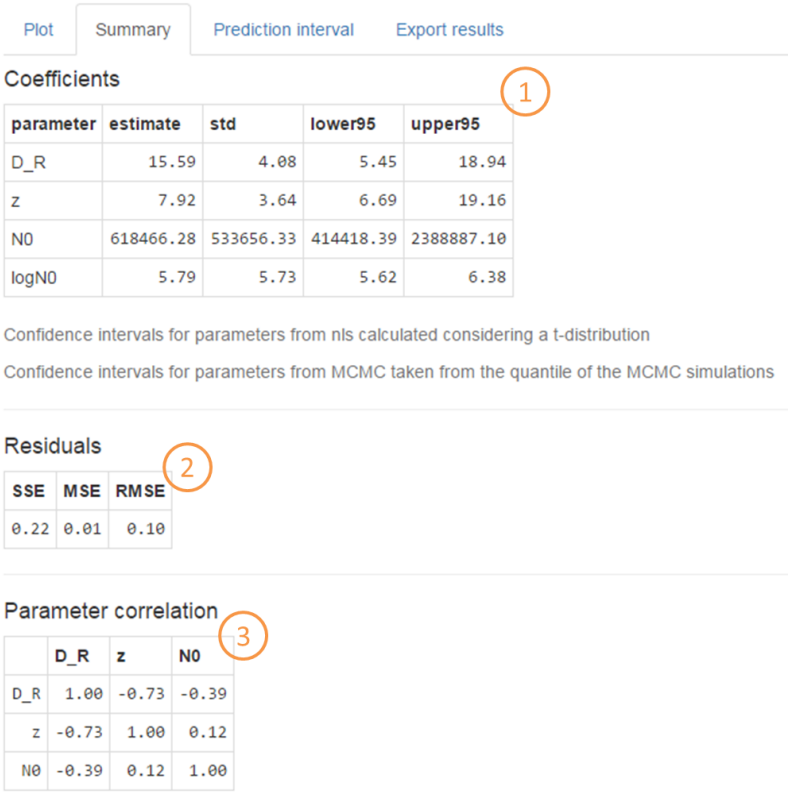


Figure 11: Summary of the adjustment provided by Bioinactivation SE

Figure 12 illustrates the prediction interval calculated by Bioinactivation SE based on the model adjustment performed. The prediction interval calculated at the selected confidence level is indicated by the blue shaded ribbon. Moreover, the dotted and dashed lines represent, respectively, the mean and median of the Monte Carlo simulations used for the calculation of the prediction interval.

The prediction interval is calculated based on a series of MCMC simulations of the inactivation process considering that the model parameter follow the probability distribution estimated by the adjustment algorithm. The prediction interval calculated based on the nlr model fitting may be discontinuous. This is due to the fact that the probability distribution of the model parameters estimated using this algorithm may contain negative values, which result on failed simulations.

Note that the prediction interval is calculated for the temperature profile input by the user.

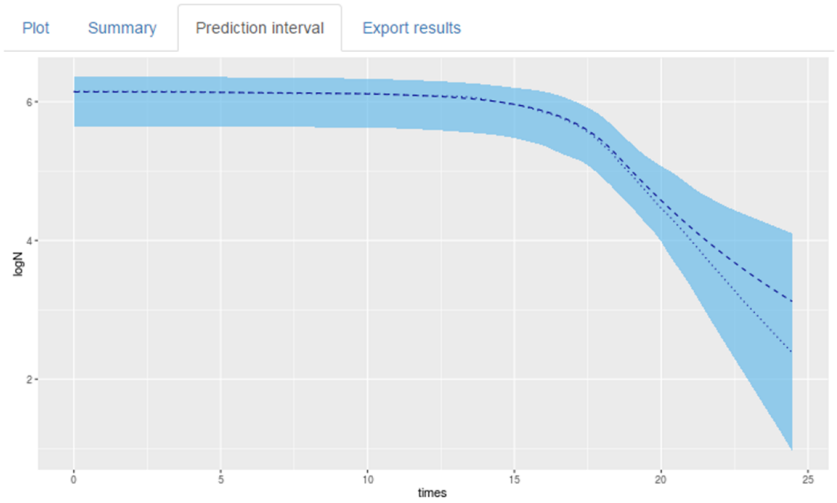


Figure 12: Example of the prediction interval calculated by Bioinactivation SE

Finally, the *Export results* tab, illustrated in Figure 13, allows to download the results of the adjustment as text files for further post-processing. This page contains four buttons:

- ① downloads the survivor curve as a tabular text file. As well as the model variables, this file contains the estimated survivor fraction (*S*) and its decimal logarithm (*logS*).
- ② downloads a table containing the estimated values of the model parameters, their standard deviations and confidence intervals.
- ③ downloads a table including the SSE, MSE and RMSE of the adjusted curve with respect to the experimental data.
- ④ downloads a table with the estimated parameter correlation.

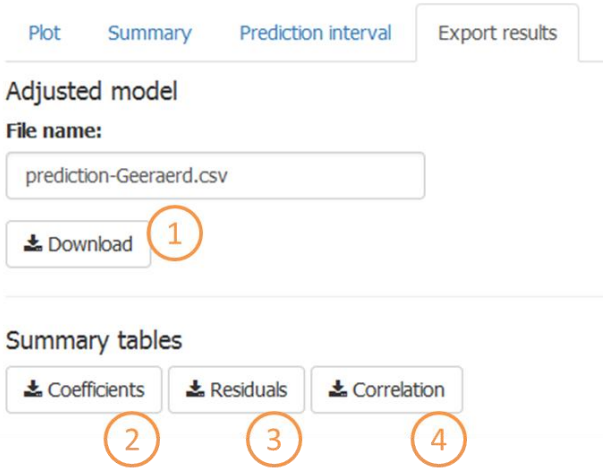


Figure 13: Contents of the *Export results* page

5.3 *About* page

The about page contains miscellaneous information regarding Bioinactivation SE. This includes:

- The version of Bioinactivation SE.
- The version of the bioinactivation package upon which this version of Bioinactivation SE has been built.
- The developers and maintainers of the application and how to contact them.
- Instructions on how to citate Bioinactivation SE on scientific papers.

6 Citation

When using Bioinactivation SE in scientific papers, please refer it as:

XXXX

7 Contact

For issues not covered in this manual, please contact:

Pablo S. Fernández Escámez
Departamento de Ingeniería de los Alimentos y del Equipamiento Agrícola
Instituto de Biotecnología Vegetal
Universidad Politécnica de Cartagena (ETSIA)
pablo.fernandez@upct.es

8 References

Bigelow, M. (1921). The logarithmic nature of thermal death time curves. *Journal of Infectious Disease, 29*, 528-536.

Chang, W., Cheng, J., Allaire, J., Xie, Y., & McPherson, J. (2015). shiny: Web Application Framework for R.

Garre, A., Fernández, P. S., & Egea, J. A. (2015). bioinactivation: Simulation of Dynamic Microbial Inactivation.

Geeraerd, A., Herremans, C., & Van Impe, J. (2009). Structural model requirements to describe microbial inactivation during a mild heat treatment. *International Journal of Food Microbiology, 59*, 185-209.

Mafart, P. a. (2002). On calculating sterility in thermal preservation methods: Application of the Weibull frequency distribution model. *International Journal of Food Microbiology*(72), 107–113.

Peleg, M., & Cole, M. (1988). Reinterpretation of microbial survival curves. *Critical Reviews in Food Science, 380*, 353-380.

Appendix A: Inactivation models included in Bioinactivation SE

This section presents the constitutive equations of each inactivation model implemented in Bioinactivation SE. Each one of them has been implemented in differential form, to be able to describe dynamic inactivation. For further detail on each model, please consult the original papers referenced.

8.1 Bigelow model

The model defined by Bigelow (1921) is described by the following system of differential and algebraic equations:

$$\frac{d\log_{10}N(t)}{dt} = -D(T)$$

$$\log_{10}D(T) = \log_{10}D\left(T_{ref}\right) - \frac{T - T_{ref}}{z} \quad (1)$$

8.2 Peleg model

The model defined by Peleg (1988) is described by the following system of differential and algebraic equations:

$$\frac{d\log_{10}S(t)}{dt} = -b(T)n\left(\frac{log_{10}S}{b(T)}\right)^{\frac{n-1}{n}}$$

$$b(T) = \ln\left(1 + e^{k_b(T-T_c)}\right) \quad (2)$$

8.3 Mafart model

The model defined by Mafart et al. (2002) is described by the following system of differential and algebraic equations:

$$\frac{d\log_{10}S(t)}{dt} = -p\left(\frac{1}{\delta(T)}\right)^{p} t^{p-1}$$

$$\log_{10}\delta(T) = \log_{10}\delta\left(T_{ref}\right) - \frac{T - T_{ref}}{z} \quad (3)$$

8.4 Geeraerd model

The model defined by Geeraerd et al. (2009) is described by the following system of differential and algebraic equations.

$$\frac{dN}{dt} = -\alpha \cdot k_{max} \cdot \gamma \cdot N(t)$$

$$\frac{dC_c}{dt} = -k_{max}(T) \cdot C_c$$

$$\alpha = \frac{1}{1 + C_c} \quad (4)$$

$$\gamma = 1 - \frac{N_{res}}{N}$$

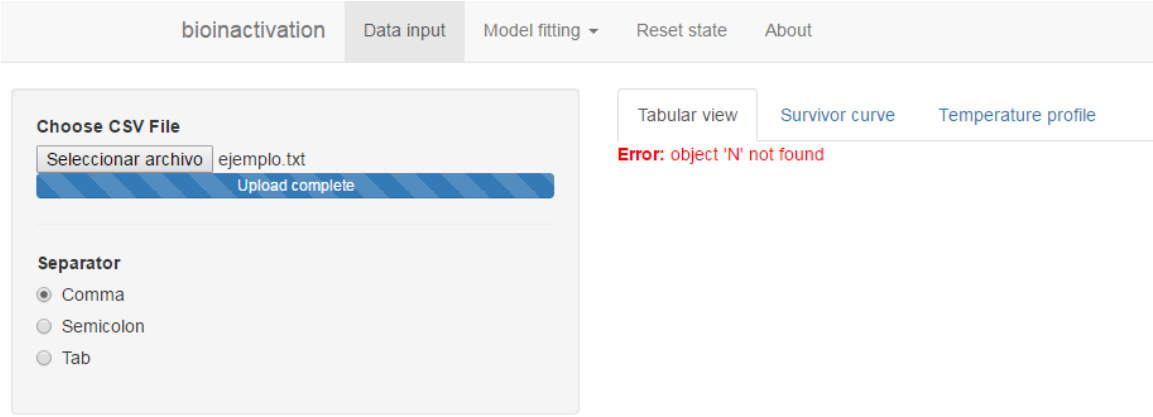
$$k_{max}(T) = k_{max} 10^{-(T - T_{ref})/z}$$

Appendix B: Common error messages and known bugs

1 Error in *Data input* after data input

ERROR DESCRIPTION

The main panel of the *Data input* section shows the message **Error:** object 'N' not found after file upload.



REASON

The file selected does not contain a column named *N* or the wrong separator has been selected.

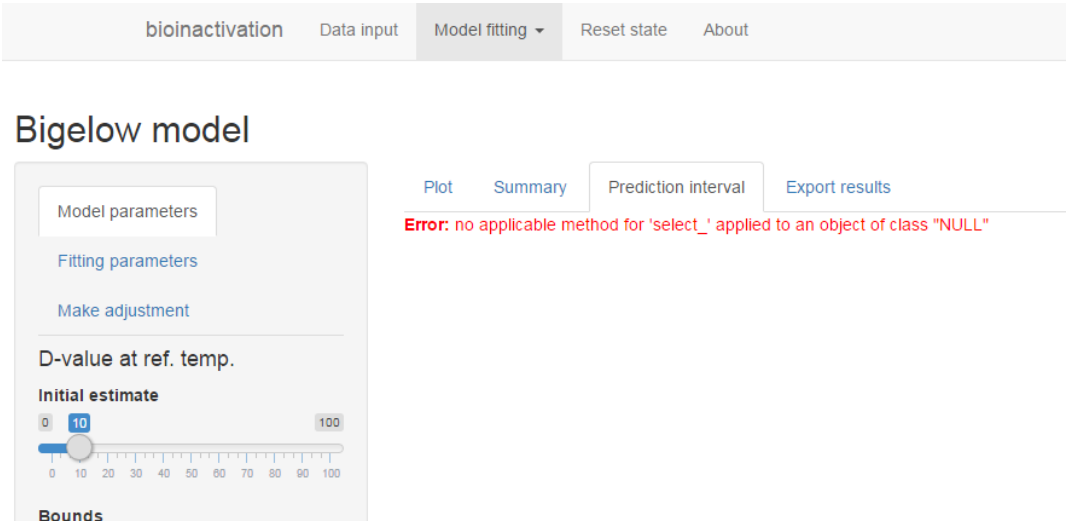
SOLUTION

Review the input file and the separator chosen.

2 Error in *Prediction interval*

ERROR DESCRIPTION

The tab *Prediction interval* of the *Model input* page shows the message **Error:** no applicable method for 'select_' applied to an object of class 'NULL'.



REASON

The adjustment has not been performed yet.

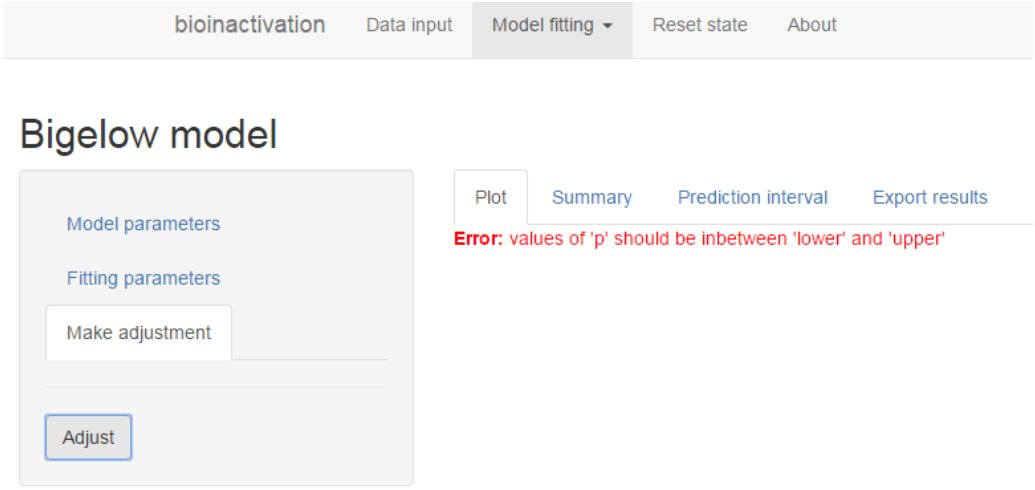
SOLUTION

Perform the model fitting.

3 Wrong bounds selected

ERROR DESCRIPTION

The tab *Plot* (or Summary) of the *Model input* page shows the message **Error:** values of 'p' should be inbetween 'lower' and 'upper' after launching the model adjustment.



REASON

The initial estimate of at least one model parameter (not considered known) is not within the admissible bounds defined by the user.

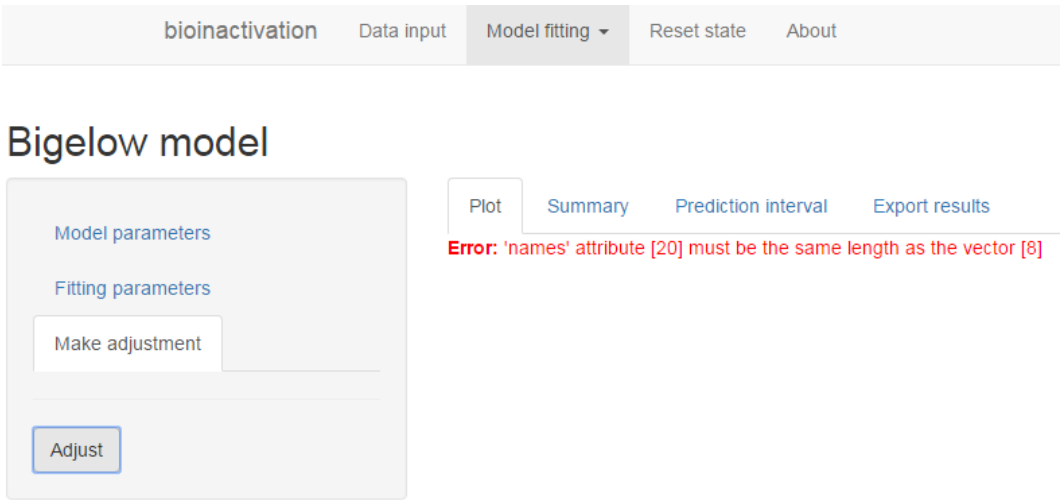
SOLUTION

Readjust the initial estimates and bounds of the model parameters.

4 Error after performing nlr adjustment

ERROR DESCRIPTION

The tab *Plot* (or *Summary* or *Prediction interval*) of the *Model input* page shows the message **Error:** 'names' attribute [X] must be the same length as the vector [Y] after launching a model adjustment using the nlr algorithm.



REASON

The nlr algorithm failed to converge.

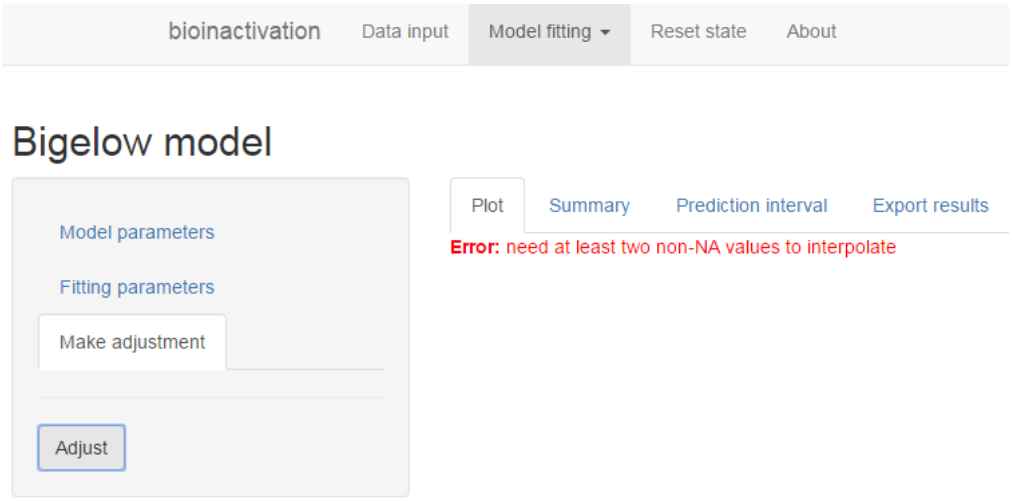
SOLUTION

Make sure that the reference temperature has been fixed to a value. If it has, repeat the adjustment with different initial estimates of the model parameters. If that fails, try fixing some of them to reasonable values. If the error persists, use the nlr algorithm.

5 Error after performing MCMC adjustment

ERROR DESCRIPTION

The tab *Plot* (or *Summary* or *Prediction interval*) of the *Model input* page shows the message **Error:** need at least two non-NA values to interpolate after launching a model adjustment using the MCMC algorithm.



REASON

The MCMC algorithm failed to converge.

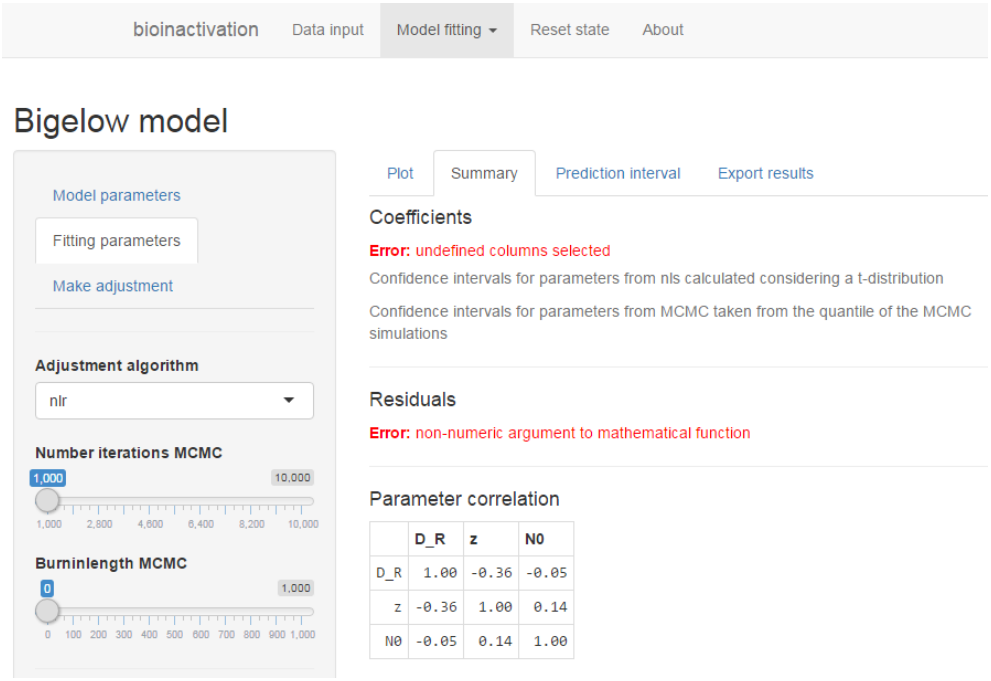
SOLUTION

Make sure that the reference temperature has been fixed to a value. If it has, repeat the adjustment with different initial estimates of the model parameters. If that fails, try fixing some of them to reasonable values. If the error persists, use the nlr algorithm.

6 Error in summary tables

ERROR DESCRIPTION

The places where the coefficient and residuals tables should be located show error messages but the survivor curve has been generated.



REASON

The adjustment algorithm has been modified.

SOLUTION

Modify the selection box where the fitting algorithm is selected or relaunch the model fitting.

7 Negative values are estimated for the confidence interval of the model parameters after nlr adjustment

ERROR DESCRIPTION

The application estimates negative values for the lower bounds of the confidence intervals of the model parameters after performing a nlr adjustment.

REASON

The calculation of confidence intervals for the model parameters based on a nlr model adjustment is based on the hypothesis that the model parameters follow a t-distribution with the mean and standard deviations estimated by the adjustment algorithm. If the standard deviation is large, or the mean is close to zero, the confidence interval estimated may contain negative values.

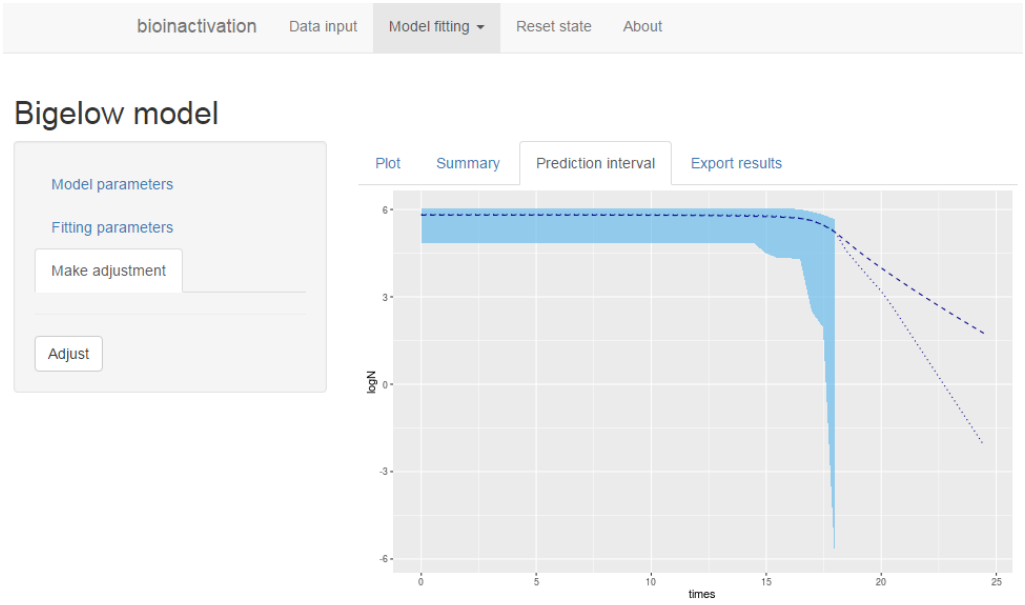
SOLUTION

Use the MCMC algorithm.

8 Strange prediction interval calculated after nlr adjustment

ERROR DESCRIPTION

The prediction interval calculated by the tool looks strange, with some blank areas after performing a nlr adjustment.



REASON

The hypothesis followed for the calculation of prediction intervals from the results of the nlr adjustment is similar to those use for the calculation of confidence intervals of the model parameters. This may produce failed iterations of the Monte Carlo algorithm used for the generation of the prediction interval.

SOLUTION

Use the MCMC algorithm.

Appendix C: Example input file

time	temp	N
0	75	1070000.0000118
7.69	95	403500.000001776
15.38	115	611499.999996755
16.15	117	702000.000000314
16.92	119	436499.999998404
17.69	121	253500.000000377
18.46	123	154499.999999696
20.46	123	18149.9999999109
22.46	123	2219.99999999673
24.46	123	140.000000000568
0	75	481499.999999387
4.69	95	535999.999996582
15.38	115	549500.000000621
16.15	117	455500.000001031
16.92	119	407000.000004479
17.69	121	295500.000001854
18.46	123	194999.99999887
20.46	123	16400.0000000869
22.46	123	1384.99999999851
24.46	123	175.000000001493