Cluster Newton method GUI for Pharmacokinetics Model User Guide

Yasunori Aoki
yaoki@uwaterloo.ca

This software is a researcher-developed alpha version (i.e., not professionally developed). Software may crash unexpectedly so please SAVE FREQUENTLY.

Step1 Define the model using a system of ODEs

In this step we define a system of ODEs that defines the pharmacokinetics model.

Syntax

The system of ODEs should be defined using the following syntax.

d<variable_name1>dt = <right hand side1> ;<comment1>
d<variable_name2>dt = <right hand side2> ;<comment2>

Parser will automatically recognise <variable_name> and update the “Variables” list.
As <right hand side> is typed “Parameters” list will be updated. These lists can be seen in the pull down menu of “Replace” section of Step 1 or in Step 2 (variables) and Step 3 (parameters).

All spaces will be ignored.
All variable names and parameter names are case sensitive.
All special functions need to be written in capital letters.
<right hand side> can be continued on in multiple likes e.g.,

dPTV_12dt = ktransit * PTV_21 ; comment1
- ka / FaFg * PTV_12 ; comment2

Error message will appear at the bottom of the window in red if there is any potentially fatal error. It is possible to ignore the error and proceed as there can be a chance of false positive error message.

By pressing update Highlights button the variables will be updated to bold, parameter to be Italic and comments to be grey, i.e.,

d**PTV_12**dt = *ktransit* * **PTV_21** ; comment1
- *ka* / *FaFg* * **PTV_12** ; comment2

This highlighting function is implemented for the convenience purpose only and should not be used for the reference of the correct implementation of the ODE.

If the ODE is implemented correctly, all the ODE variable names should appear in the "Choose Compartment from ... " text in Step 2 and all the parameter names should be listed in Step 3. Similarly these variable names and parameter names can be seen in the pull down menu in "Replace" section of Step1.

Special characters:

; all characters following ";" symbol in the line will be treated as comments.
All the other punctuation characters will be treated as regular characters (note a variable name such as k.transit or k,transit is possible)

^ power
* multiplication
/ division
+ addition
- subtraction
(parenthesis left (Note: no other types of parentheses will be treated as parentheses)
) parenthesis right (Note: paring of the parentheses are not checked in the current implementation)

Special functions(all should be capitalised):

EXP()
exponential of the value of the expression within the parentheses
LOG()
natural log
LOG10()
log base 10
ABS()
absolute value
SIGN()
if the value of the expression inside of () is negative it returns -1 else +1
SQRT()
square root
SIN()
sine
COS()
cosine
TAN()
tangent
ASIN()
arcsine (inverse function of sine)
ACOS()
arccosine (inverse function of cosine)
ATAN()
arctangent (inverse function of tangent)
LN()
natural log (equivalent at LOG())

Step2 Define dose events

In this step we define the dose events, we do so by filling in the spread sheet like GUI. The rows of the spread sheet can be extended by pressing “increase dose rows” button. The rows of the spread sheet can be left empty. All the incomplete rows will be ignored. Multiple bolus and infusion can be coded using this GUI.

Time

Time of the dose event should be entered in the column indicated as “Time”. Any mathematical expression can be written in this column similarly to Step1. We can use any parameters and also can introduce new model parameter here; however cannot use any of the ODE variables here.

Compartment

Specify the ODE variable (“compartment” in terms of compartmental model) where the drug/compound/substance is released by typing in the name of ODE variable in the “Compartment” column. Any other expression other than a name of the ODE variable is not allowed in this column.

Dosage

Specify the amount of drug/compound/substance released. Any mathematical expression can be written in this column similarly to Step1. We can use any parameters and also can introduce new model parameter here; however cannot use any of the ODE variables here.

Duration

Specify the duration of drug/compound/substance released, for bolus administration specify duration as 0. Any mathematical expression can be written in this column similarly to Step1. We can use any parameters and also can introduce new model parameter here; however cannot use any of the ODE variables here.

Note

We can simply leave a memo on these columns. It will be saved with the model; however, have no influence to the computation.

Step3 Define Parameter Domain of Interest and Theoretical Upper and Lower Bound

Parameter Domain of Interest

The range of the model parameters that desired parameters are most likely to be found. This range should be chosen based on the a priori knowledge of the model parameter (e.g., from in vitro data or in animal data). This parameter domain is used to generate the initial cluster (initial distribution of the parameters).

Lower bound of this parameter domain should be entered in “Parameter Domain of Interest LB” column as a numerical value. (only numbers are allowed in this column)

Upper bound of this parameter domain should be entered in “Parameter Domain of Interest UB” column as a numerical value. (only numbers are allowed in this column)

Both “Parameter Domain of Interest LB” and “Parameter Domain of Interest UB” may NOT be left empty. If one wishes to fix a parameter value, enter the same value in both “Parameter Domain of Interest LB” and “Parameter Domain of Interest UB”. Then the parameter is treated as a fixed constant.

Similarly to setting the initial estimate of the parameter for any parameter estimation algorithm/method/software, the final parameter estimate does depend on how this domain is specified.

Theoretical Upper and Lower bounds

If the parameter has a theoretical lower or upper bounds, it should be specified in “Theoretical LB (if applicable)” and “Theoretical UB (if applicable)” columns. Only numerical values should be entered in these columns. Either or both of these columns can be left empty; however, in order to reliably sample the parameters it is necessary to include as many theoretical bounds as possible. For example, if model parameter cannot be negative in theory, put 0 in the Theoretical LB column.

Based on these theoretical bounds, the parameters are nonlinearly transformed so that CN method will search in the transformed parameter space whose values can be any real number (all the transformations can be checked and modified if necessary in Step3-2).

Step 3-2 (optional): Inspect and Edit parameter transformation

In this step we can see how the parameter are transformed internally and if necessary can be changed. Also we can inspect the initial cluster and make changes by “importing initial X” or by changing the number of initial cluster.

g(x) is a function that transforms the internal variable x1, x2,..., to the original parameter defined by the user. g inverse is an inverse of g(x). If g(x) is modified by a user, g inverse must be also modified by the user. There is a simple testing mechanism to check the inverse; however, user should note that it is the user responsibility to provide correct inverse of g in order for the software to function correctly.

In this step, only the necessary action is to inspect “Initial Cluster” to make sure all the values are not “NaN”, “Inf” nor “-Inf”. If any of these values are found in the Initial Cluster go back to Step 3 and make necessary correction.

Step4 Define observation events and input data

In this step we define the observation and measurement data by entering values and expressions in spread-sheet like interface. The rows of the spread sheet can be extended by pressing “increase observation rows” button. The rows of the spread sheet can be left empty. All the incomplete rows will be ignored.

Time

Enter the numerical value of the observation was taken in “Time” column. Only numerical values are allowed in this column.

Assume that the observation is always before bolus administration or start of the infusion when observation and dosage coincides. If not desirable, slightly move the dosage of observation so that it will not be exactly at the same time.

Observation Variable

Define the expression of the variable the observation was made. The mathematical expression should include ODE variable and can use any of the parameters that are already defined in Step3. New parameter cannot be introduced at this step.

Value

Enter the observed value of the variable defined in the “Compartment” column. We gan give his value as a mathematical expression; however, non of the parameter nor ODE variable can be used. (e.g., LOG10(1.234) is allowed).

Step5 Run Cluster Newton method

Once all Steps 1-4 are done, press “Start Cluster Newton method”. If the model is not saved prior to pressing this button, you will need to specify the run directory. Once this button is pressed then all the update from running of the algorithm will be displayed in the textfield. After first iteration is complete, the user can inspect the model fit etc. in Step6 and any of the figures in Step6 is updated dynamically at the iterations in the algorithm progresses.

User may fine tune the algorithm by going to the “Algorithm setting” tab.

Step6 Inspect Modelfit and Found Parameters

Various plots of the model fit and parameters can be found in this Step.
The parameter and model fit displayed in this step can be chosen and filtered using “Percentile” and “Iteration” spin-box interface. By adjusting “Percentile” the plots only displays the graphs obtained from the parameter sets of top 25% best fit model.

All plots can be exported as a .csv file by pressing “save plot as CSV” so that the plot can be redrawn using a graphing software of user’s choice.

Planning to implement so that the pdf file of the plot can be exported; however, currently please simply use the screenshot to export the plot.

Output Files

This software will generate three directories when the model is saved: “compResult”, “history”, “model”. Both “compResult” and “model” are used when “open” the model using this software and assumes strict format, so it is best NOT to alter any of the files in these directories.

Model

In this folder all the data related to the model definition is saved.

dose_events.txt: stores the dose events defined in Step2 as a csv format.

initial_cluster.txt: stores the initial cluster in the reparameterised scale (x-scale) in csv format.

observations.txt: stores the observation events defined in Step 4 in csv format.

ODE_expression.txt: stores the system of ODEs as defined in Step1 in text format.

parameter_specifications.csv: stores the parameter information defined in Step3.

parameterNames.txt: List of parameter names in text format.

reparaDefinition.txt: stores the internal reparameterisation that can be seen as g(x) in Step 3-2.

reparaInverse.txt: stores the inverse of the internal reparameterisation that can be seen as g inverse in Step 3-2.

variableNames.txt: List of variable names in text format.

x_ul.csv: stores the upper and lower bound of the initial cluster in reparameterised scale (x-scale).

rscript_ODE.R: the model file translated in R script language (the dose events need to be coded manually).

matlab_ODE.m : the model file translated in Matlab language (the dose events need to be coded manually).

compResult

In this folder all the data related to the model definition is saved.

computation_log.txt: stores computation log as displayed in Step 5 in text format.

ODE_sol_i_j.csv: stores solution of the j-th variable of the system of ODEs from i-th iteration.

ode_time.csv: stores the time ODE is evaluated in csv format.

Parameter_i.csv: stores the parameter values (in the original parameter scale) from the i-th iteration (note this list is NOT sorted).

R_i.csv: stores the list of sum of square residuals from the i-th iteration (note this list is NOT sorted).

sortIndex.csv: stores the index of the parameter from lowest sum of square residuals. (i-th row corresponds to i-th iteration).

X_i.csv: stores the list of parameter values in the reparameterised scale(in x-scale) (note this list is NOT sorted).

Y_i.csv: stores the list of observation variables at i-th iteration (note this list is NOT sorted).