

User Guide for PKgraph Package

Xiaoyong Sun^{†‡*}

June 16, 2010

[†]Bioinformatics and Computational Biology Program, [‡]Department of Statistics
Iowa State University, Ames, Iowa 50010, USA

Contents

1	Introduction	2
2	Installation	2
3	PKgraph infrastructure	3
3.1	Graphical user interfaces	3
3.1.1	Main interface	3
3.1.2	Graph interface	4
3.2	Functional module	4
4	Functions	5
4.1	Project	5
4.2	Configure	6
4.3	Data management	7
4.4	Exploratory data analysis	7
4.4.1	Univariate	8
4.4.2	Bivariate	8
4.4.3	Parallel coordinate plots	8
4.5	PK models	9
4.5.1	Configure model result	11
4.5.2	Parameters	11
4.6	Model validation	12
4.6.1	Influence analysis summary (PsN)	12
4.6.2	Visualization for influence analysis	13
4.6.3	Bootstrap summary (PsN)	13
4.6.4	Visualization for bootstrap	14
4.7	Model comparison	15
4.7.1	Select datasets	15
4.7.2	Configure mapping	16
4.7.3	Comparison	16
4.8	Interactive graphics	17

*johnsunx1@gmail.com

1 Introduction

Population pharmacokinetic (PopPK) modeling has become increasingly important in drug development because it allows unbalanced design, sparse data and the study of individual variation. However, this complexity of the model makes it a challenge to diagnose the fit. Graphics can play an important and unique role in PopPK model diagnostics. The software described in this paper, PKgraph, provides a graphical user interface for PopPK model diagnosis with interactive graphics. It also provides an integrated and comprehensive platform for analysis of pharmacokinetic data including exploratory data analysis, goodness of model fit, model validation and model comparison. It can be used with a variety of modeling fitting software, including NONMEM, Monolix, SAS and R. PKGraph is programmed in R, and uses the R packages `lattice`, `ggplot2` for static graphics, and `rggobi` for interactive graphics. This R package is supported with a user-friendly graphical user interface so that users can easily control diagnosing with simple clicks. The PKGraph software serves as a supplement to the existing packages: NONMEM, Xpose and PsN for diagnosing models.

PKgraph is a R package built on the following R packages: `RGtk2`, `gWidgets`, `gWidgetsRGtk2`, `lattice`, and `ggplot2`. It requires R (> 2.0) and GTK+, and runs under Windows, Linux and Mac.

2 Installation

PKgraph needs to install the following programs and R packages:

1. install GTK

For Windows, you can download the GTK Developer's Pack from <http://gladewin32.sourceforge.net/>

For Unix, you can fetch the source files for the different libraries from <ftp://ftp.gtk.org/pub/gtk/v2.8/>

2. Install RGtk2 (Please see RGtk2 Installation notes if you have problems)

- a. Install R package, `RGtk2`: `install.packages("RGtk2")`

3. install rggobi

- a. Download and install `ggobi` (www.ggobi.org)
- b. Install `rggobi`: `install.packages("rggobi")`

4. Install gWidgets

- a. Install R package, `gWidgets`: `install.packages("gWidgets")`

5. Install gWidgetsRGtk2

- a. Install R package, `gWidgetsRGtk2`: `install.packages("gWidgetsRGtk2")`

6. Install lattice
 - a. Install R package, lattice: `install.packages("lattice")`
7. Install ggplot2
 - a. Install R package, lattice: `install.packages("ggplot2")`

3 PKgraph infrastructure

The software incorporates a key concept: interactive graphics to link various datasets and diagnostics plots. The framework is programmed using RGtk2 and consists of main formats of interfaces, (1) main, containing links to all parts of the software, and handles the basic data management, and links to diagnostic modules, and (2) graph, which provides tools specifically for each diagnostic module. (2). Basic module: data input/output module, configuration module, and data management module.

3.1 Graphical user interfaces

3.1.1 Main interface

The main interface (Figure~1) of PKgraph provide the links to all components of the software. There are four areas: (1) tool area (tool bar and menu bar, top), (2) directory area (middle-left), (3) data area (middle-right) and (4) status bar (bottom).

- The tool area has menu items linking to the basic management modules (project, configuration, data management) and the diagnostic modules (exploratory data analysis, PK models, model validation, model comparison and interactive diagnostics). These are menu items containing numerous functions associated with each of the different types of diagnostics.
- The directory area shows current directory and all of its files. These files might be data files, or code, depending on the modeling software used.
- Clicking on any of the data files, will open them and display them in the data area (3). Choosing the file also brings up a panel allowing for different formats to be read, thus handling all possible modeling software formats. The data files might contain raw data, and model diagnostics such as parameter estimates, fitted values and residuals and these are displayed in the table view of the data area.
- The stats bar displays the progress of the different functions, for example here it says “Data is loaded successfully” to indicate that there were no problems with opening the data file.

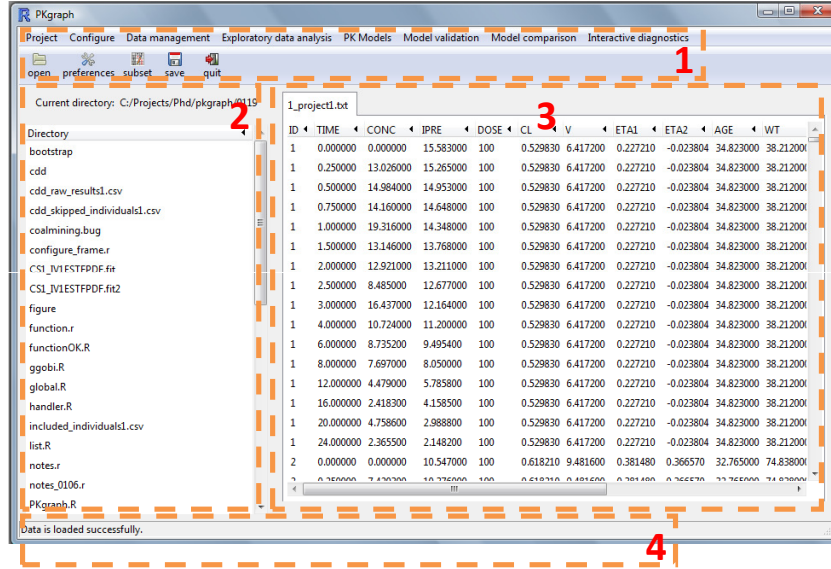


Figure 1: Main interface of PKgraph

3.1.2 Graph interface

Selecting an item from a diagnostic module menu brings up a graph interface (Figure~2). The style of the interface is the same for all diagnostic functionality. It contains three areas: 1) parameter setup area, 2) tool bar, 3) plot area.

- The parameter area setup allows choice of variable, plot labels, layout for trellis or faceted plots. A choice of lattice or ggplot2 graphics is provided.
- The tool bar allows the plots to be saved, opening the plot in ggobi for interaction on the plot (*ctrl + b* for brushing data), synchronize subset selection from ggobi to the data in R, and close ggobi.
- The plot area displays the figure, and multiple figures if more than one are created.

3.2 Functional module

Functional module matches the menu items in PKgraph toolbar. It includes the following menu itmes:

- Project

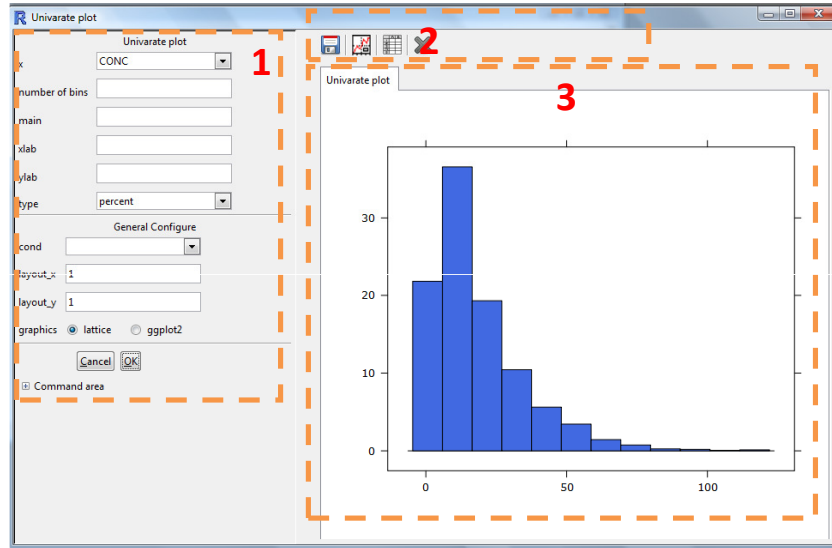


Figure 2: Graph interface of PKgraph

- Configure
- Data management
- Exploratory data analysis
- PK models
- Model validation
- Model comparison
- Interactive graphics

In the next sections, I will go through each menu item in detail.

4 Functions

In this section, I will go through each function in the menu item of toolbar.

4.1 Project

This menu item is in charge of input, output and save data. It has the following functions (Figure~3),

- *Open data*: open modeling fit result from NONMEM, Monolix, SAS, R or other software. It has options to setup the data format, start line and separation symbol.
- *Save a file*: save a file.
- *Save a workspace*: save a workspace for later usage. It generally saves a group of lists for configuration and related data.
- *Restore old workspace*: restore the workspace from the data and list you saved from previous step.
- *Exit*: exit from PKgraph.

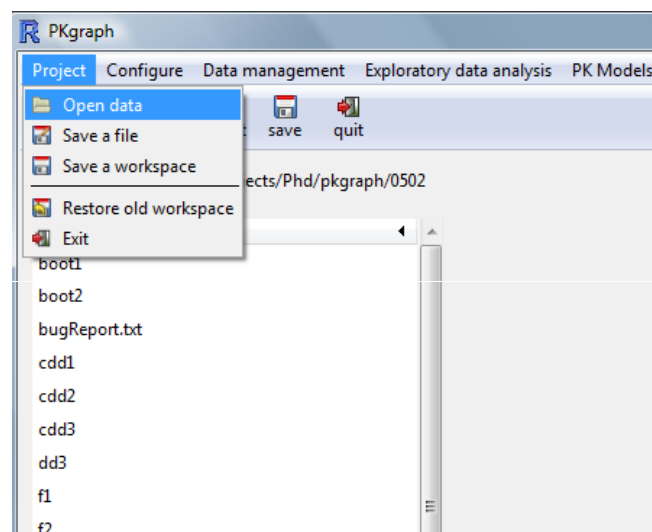


Figure 3: Menu items in *Project*

4.2 Configure

This menu item is utilized to configure PKgraph. It has the following functions (Figure~4),

- *Set working directory*: change current directory.
- *Set saving format*: set up saving format for figures, including pdf, jpg, tiff, png bmp, win.metafile, and figure width and height. If figure width and height is not setup, a default one will be used.

- *Save a workspace*: save a workspace for later usage. It generally saves a group of lists for configuration and related data.
- *Set figure configuration*: color and loess can be setup here for figures.

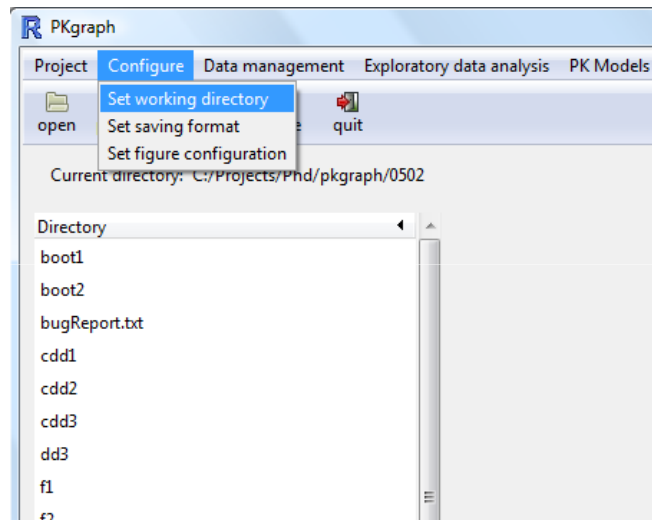


Figure 4: Menu items in *Configure*

4.3 Data management

This menu item is utilized to manage data. It has the following functions (Figure~5),

- *Subset*: subset current data.
- *Factor*: factor categorical variables. Graphical packages require variables to be factor type in order to display related symbol in figures.

4.4 Exploratory data analysis

This menu item is utilized to explore data and screen patterns. It has the following functions (Figure~6),

- *Univariates*: plot univariate variables.
- *Bivariates*: plot bivariate variables.
- *Parallel coordinate plot*: Parallel coordinate plot for multivariate variables.

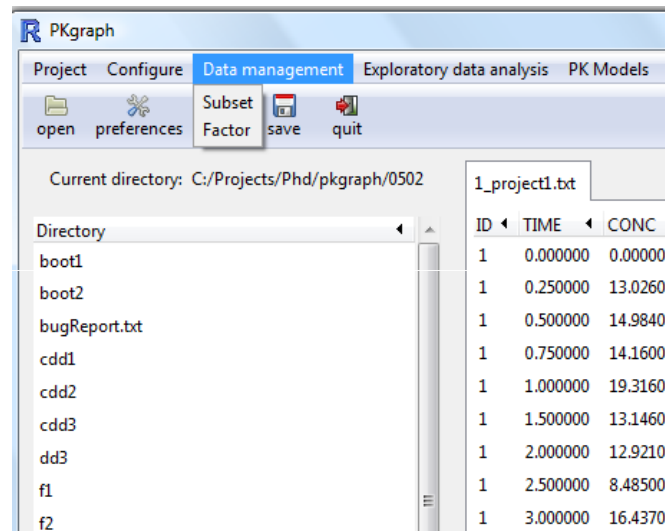


Figure 5: Menu items in *Data management*

4.4.1 Univariate

When clicking this menu item, users will generate a “graph interface”(Figure~2). In this interface, users can specify all parameters in the left area of window. In the right area of window, it has the following four buttons on the top:

- *save*: save figures.
- *ggobi*: open the plot in ggobi for interaction on the plot.
- *synchronize*: synchronize subset selection from ggobi to the data in R.
- *close*: close ggobi.

4.4.2 Bivariate

This menu item also generates a “graph interface”. It is similar to the Univariate interface, except that users will have two variables instead of one.

4.4.3 Parallel coordinate plots

This menu item provides access to lattice function: *parallel* function from lattice package.

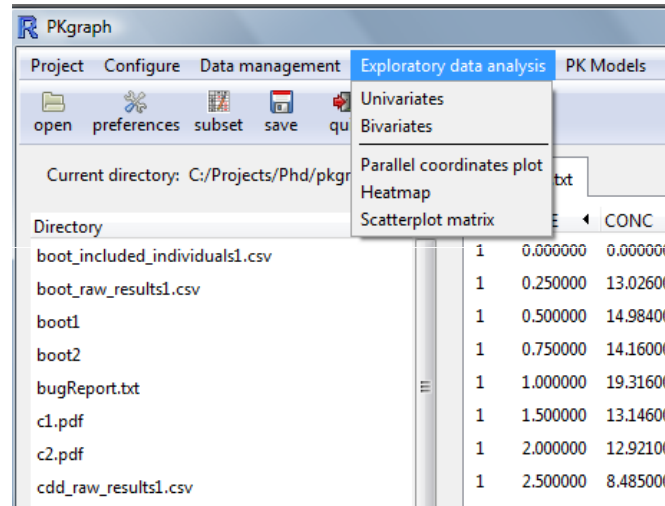


Figure 6: Menu items in *Exploratory data analysis*

4.5 PK models

This menu item is utilized to check model assumptions and goodness of fit. The guideline follows Census menu (<http://census.sourceforge.net/>). It has the following functions (Figure 7),

- *Configure model result*: This is the key step to match data variables to default metric system. By this step, data from any platform (NONMEM, Monolix, SAS, R) can be interpreted graphically in figures.
- *Individual plots*: Bivariate plot for each individual.
- *Goodness of fit plots*: Goodness of fit plot is one of key tools to check model fitting. These kinds of plots will give an overall perspective of model performance, including scatter plot for concentration versus PRED, concentration versus IPRED, PRED versus time and IPRED versus time.
- *Parameters*: Generally, there are assumptions for distribution of parameters during modeling process. The histogram is utilized to check this distribution. In addition, the correlation of parameters has significant effect on modeling performance, and it can be checked by scatter plots or a scatterplot matrix.
- *Random effects*: The assumptions for random effects also need to be tested for distribution and correlation by histogram, scatter plots or a scatterplot matrix.

- *Structural model*: Structural model can be diagnosed by PRED versus concentration conditioned on time, IPRED versus concentration conditioned on time, WRES versus time, WRES versus PRED, PRED versus concentration conditioned on covariates, IPRED versus concentration conditioned on covariates.
- *Residual error model*: Two assumptions are related to this submodel: 1) homoscedastic variability; 2) symmetrically distributed residuals. To test these assumptions, we applied the following techniques: 1) histogram for distributions of WRES; 2) histogram for individual distribution of WRES; 3) scatterplot of |WRES| versus PRED to check the shape of residual; 4) scatterplot of |WRES| versus PRED conditioned on covariates to screen the covariate effects; 5) autocorrelation of WRES.
- *Covariate model*: Parameters, ETA and WRES are of great use to help screen proper covariates. We can utilize the following methods to check covariate models: 1) scatter plot for parameters versus covariates, ETAs versus covariates, WRES versus covariates; 2) scatterplot matrix of covariates.

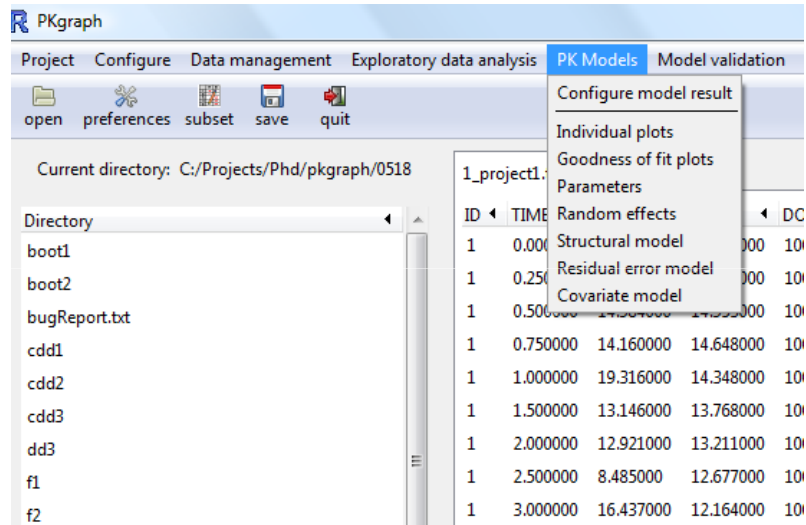


Figure 7: Menu items in *PK models*

Users have to configure data variable first before going to specific model diagnostics. To illustrate the usage of this menu item, I will use *Configure model result* and *Parameters* as examples.

4.5.1 Configure model result

The interface for this function is shown in Figure~8. The fixed variables are from data, and the flexible variables are from default metric system (Table~1).

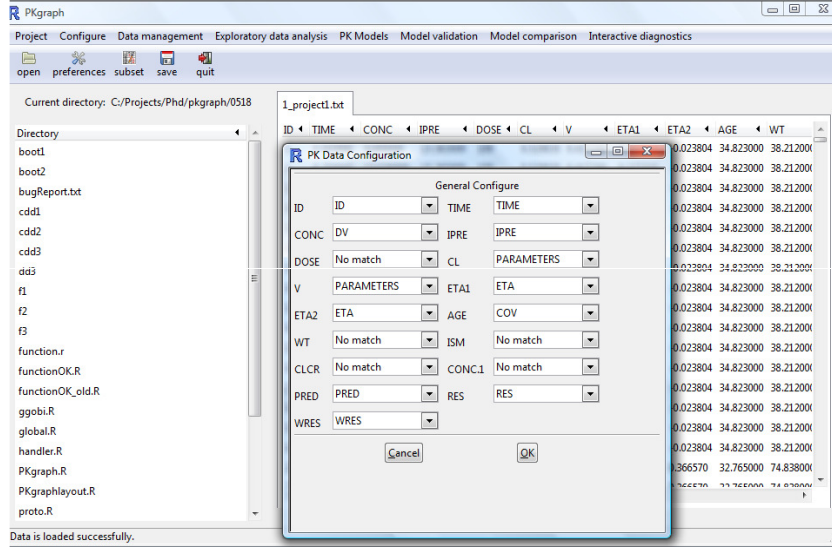


Figure 8: *Configure model result in PK models*

Package variable	Description
ID	Patient ID
TIME	Time after dose
CONC	The concentration of drug in the body
PRED	Prediction generated from model fitting
RES	Residual
WRES	Weighted residual
IPRED	Individual prediction
IWRES	Individual weighted residual
COV	Covariates

Table 1: Package metric system

4.5.2 Parameters

The interface for this function is shown in Figure~9. After users choose proper figures in the left window, the system will produce all figures automatically. Users can pick specific figures for diagnosing with functions in the toolbar.

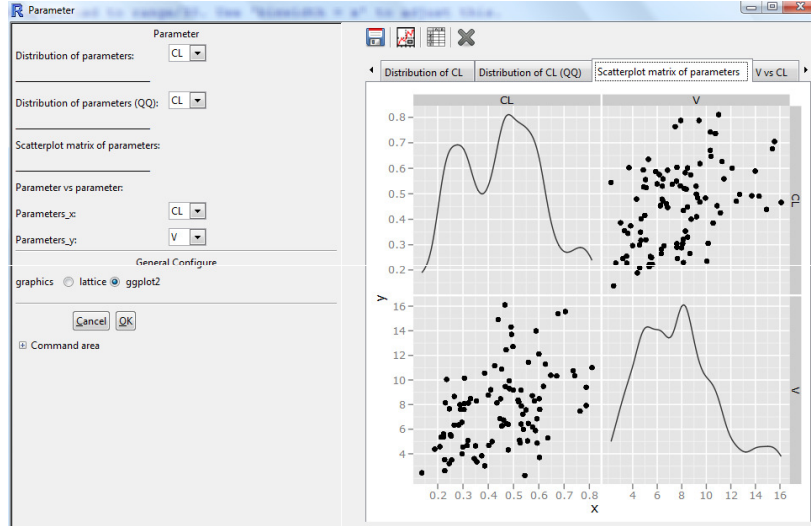


Figure 9: *Parameters in PK models*

4.6 Model validation

Resampling methods has been extensively employed in the model validation. Currently, bootstrap targets for confidence interval, case deletion diagnostics identify influential cases, and stochastic simulation is utilized to compare models (PsN). PKgraph mainly focuses on case deletion diagnostics and bootstrap. It provides the following functions (Figure~10),

- *Influence analysis summary (PsN)*: analyze PsN *cdd* results.
- *Visualization for influence analysis*: apply parallel coordinate plots and multidimensional scaling to visualize data from case deletion diagnostics (multiple NONMEM runs).
- *Bootstrap summary (PsN)*: analyze PsN *boot* results.
- *Visualization for bootstrap*: visualize data from bootstrap (multiple NONMEM runs).

4.6.1 Influence analysis summary (PsN)

This function is specifically for PsN *cdd* results (Figure~11). It takes two result files from PsN: *raw_results1.csv* and *skipped_individuals1.csv*, and generates a scatter plot for *cov.raito* versus *cov.score*.

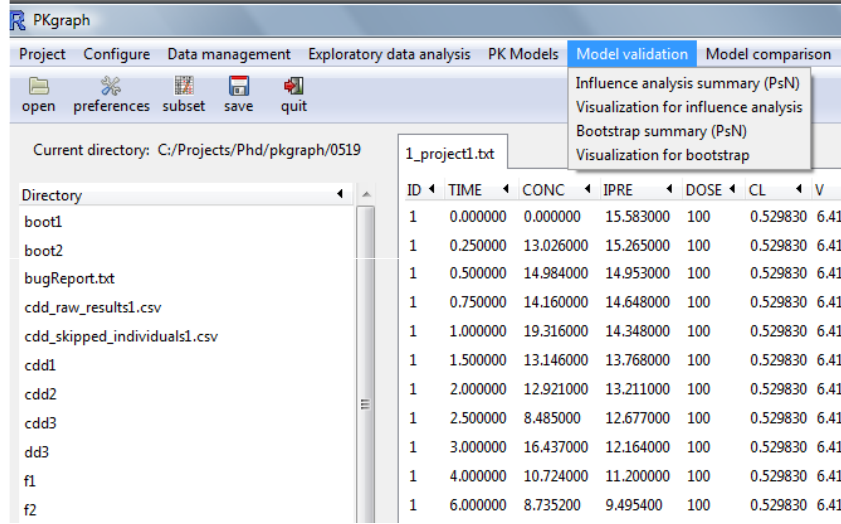


Figure 10: Menu items in *Model validation*

4.6.2 Visualization for influence analysis

This function is to visualize data from case deletion diagnostics (multiple NON-MEM runs). Let's use multiple NONMEM run from PsN (Figure~12), and find file directory for these runs. Then we can select parameters as shown in Figure~13. These parameters include:

- *Target directory path*: the path for multiple NONMEM runs. It is a required parameter.
- *Simulation folder pattern*: the common name style for multiple NONMEM runs. For this example, it is *NM_run*. It is a required parameter.
- *Patient ID*: the ID for each subject. It is a required parameter.
- *Plot variable*: the variable you use to detect difference among patients. For this example, we choose *CL*. It is a required parameter.
- *xlabel*: the name label for each NONMEM run. It is optional.

4.6.3 Bootstrap summary (PsN)

This function is specifically for PsN *boot* results (Figure~14). It takes two result files from PsN: *raw_results1.csv* and *included_individuals1.csv*, and generates related plots.

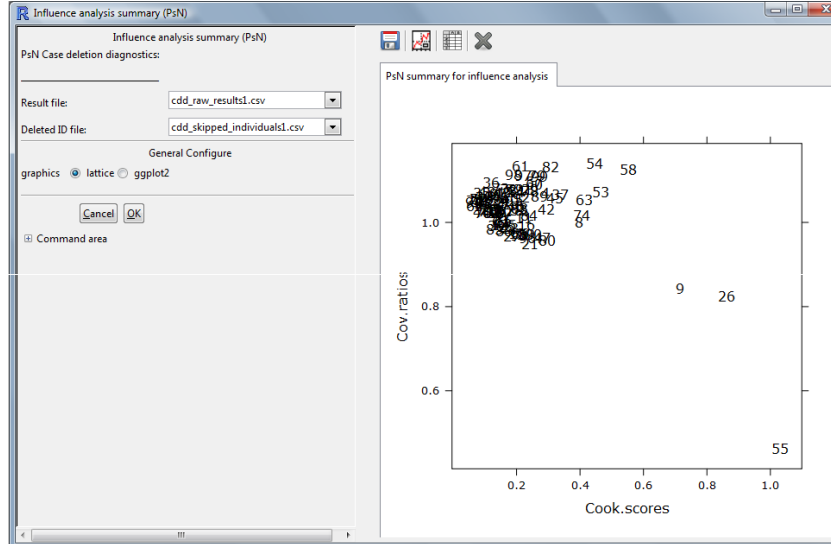


Figure 11: Influence analysis summary (PsN)

4.6.4 Visualization for bootstrap

This function is to visualize data from bootstrap (multiple NONMEM runs). Let's use multiple NONMEM run form PsN (Figure~15), and find file directory for these runs. Then we can select parameters as shown in Figure~16. These parameters include:

- *Target directory path*: the path for multiple NONMEM runs. It is a required parameter.
- *Bootstrap folder pattern*: the common name style for multiple NONMEM runs. For this example, it is *NM_run*. It is a required parameter.
- *NONMEM result file name*: the fit result for each NONMEM run. In this example, it is *CS1_IV1ESTFPDF-1.fit* . It is a required parameter.
- *Bootstrap key table path*: the path for bootstrap key file, which is file describing the sampling schema for patient IDs. It is a required parameter.
- *Bootstrap key table name*: The file describes the sampling schema for patient IDs In this example, it is *included_individuals1.csv*. It is a required parameter.
- *Patient ID*: the ID for each subject. It is a required parameter.
- *Plot variable*: the variable you use to detect difference among patients. For this example, we choose *CL*. It is a required parameter.

Name	Date modified	Type
NM_run1	2/4/2010 10:44 AM	File Folder
NM_run2	2/4/2010 10:44 AM	File Folder
NM_run3	2/4/2010 10:44 AM	File Folder
NM_run4	2/4/2010 10:45 AM	File Folder
NM_run5	2/4/2010 10:45 AM	File Folder
NM_run6	2/4/2010 10:45 AM	File Folder
NM_run7	2/4/2010 10:45 AM	File Folder
NM_run8	2/4/2010 10:45 AM	File Folder
NM_run9	2/4/2010 10:45 AM	File Folder
NM_run10	2/4/2010 10:44 AM	File Folder

Figure 12: Multiple NONMEM runs for case deletion diagnostics

- *xlabel*: the name label for each NONMEM run. It is optional.

4.7 Model comparison

In this process, there are three main steps: 1) select datasets; 2) configure mapping; 3) comparison (Figure~17). The first step is to select datasets for comparison. Currently the program only supports comparison of two models. Then users proceed to configure mapping by matching column names or variable names from two data sets. These matching variables are generally the variables from original data sets and they are not related to model fitting. When all parameters are set, the program offers three choices for comparison: “histogram comparison” (distribution comparison), “scatter plot comparison” and “transform comparison”.

4.7.1 Select datasets

This function is to select datasets available in the PKgraph data area. Figure~18 shows there are three data sets available, including fit result 2: 2_CS1_IV1ESTFPDF.fit (fit with additive error model) and fit result 3: 3_CS1_IV1ESTFPDF.fit2 (proportional error model). In this example, we will compare these two models.

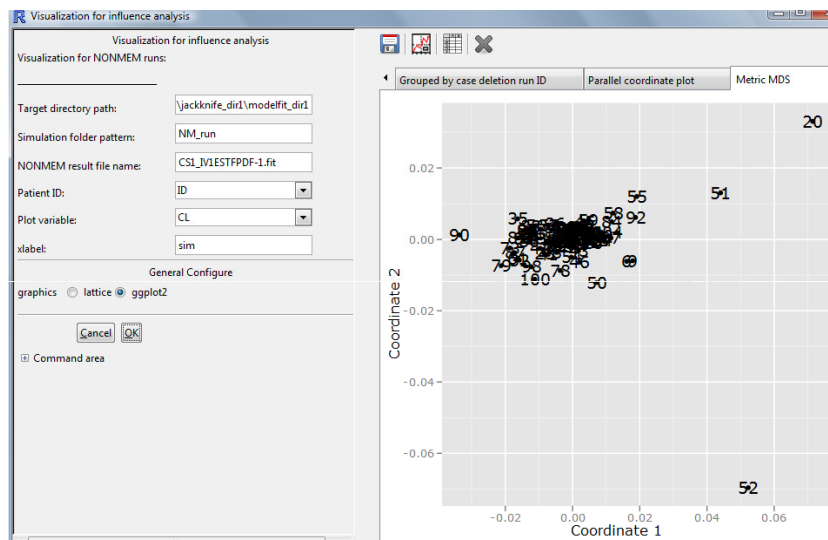


Figure 13: Parameters and results for case deletion diagnostics

4.7.2 Configure mapping

This step will join two fit results. As a result, users have to match the original data variables between two fit results. For example (Figure~19),

- *Matching variables:* *ID*, *Time*, *Concentration*, *WT*, *AGE*, etc must be matched in this step. These variables do not change with different models.
- *Non-matching variables:* *RES*, *PRED*, *WRES*, etc are fit results, and should NOT be matched. These variables change with different models.

After mapping, a new dataset joining two fit results will show in data area of main interface.

4.7.3 Comparison

“histogram comparison” enables to compare distributions of matching parameters from two models. “scatter plot comparison” provides a environment to compare matching parameters by scatter plot. “transform comparison” transforms data by ratio or log ratio in order to visualize the difference between variables from two models. All these models can be linked directly to ggobi for interactive diagnostics by clicking second button in the tool bar area on the top right panel.

All variable names for model 1 will have additional “.x” label, and all variable names for model 2 will have additional “.y” label.

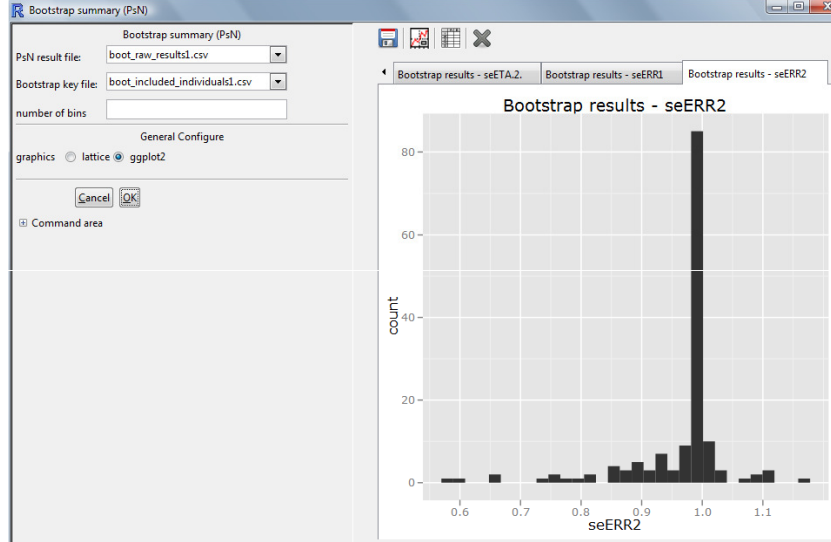


Figure 14: Bootstrap summary (PsN)

Let us look at “histogram comparison” as one example. First, we need to make sure that current data set is “4_ModelComparison” (Figure~20); second, we click “histogram comparison”. To compare CL , we select ISM (*gender*) as the conditional variable, and the result is shown in (Figure~21).

4.8 Interactive graphics

This functional module incorporates a unique feature: interactive graphics into every step of model diagnostics. It targets to link diverse data sets in one integrative platform. Users can have access to this feature through *ggobi* button in the graph interface. In addition, users have flexibility to apply this feature to achieve their specific goals. In the toolbar, there is option: *interactive graphics*, designed for this purpose. It includes three steps: select datasets; configure mapping; and diagnostics. By linking diverse data sets with a key variable, users can seek patterns by brushing, linking and diagnosing patterns conveniently.

In *ggobi*, the main operation for brushing data is $ctrl + b$. By moving the brushing rectangle, the users can select interesting subsets in *ggobi*. More information is available in [http: www.ggobi.org](http://www.ggobi.org) if needed.

Name	Date modified	Type	Size
jackknife_dir1	2/4/2010 10:50 AM	File Folder	
modelfit_dir1	2/4/2010 10:45 AM	File Folder	
boot_included_individuals1.csv	12/8/2009 2:16 PM	Microsoft Office E...	15 KB
boot_raw_results1.csv	12/8/2009 2:19 PM	Microsoft Office E...	24 KB
bootstrap.R	12/8/2009 3:25 PM	Tinn-R	8 KB
included_individuals1.csv	12/8/2009 2:16 PM	Microsoft Office E...	15 KB
raw_results1.csv	12/8/2009 2:19 PM	Microsoft Office E...	24 KB

Name	Date modified	Type
NM_run1	2/4/2010 10:44 AM	File Folder
NM_run2	2/4/2010 10:44 AM	File Folder
NM_run3	2/4/2010 10:44 AM	File Folder
NM_run4	2/4/2010 10:45 AM	File Folder
NM_run5	2/4/2010 10:45 AM	File Folder
NM_run6	2/4/2010 10:45 AM	File Folder
NM_run7	2/4/2010 10:45 AM	File Folder
NM_run8	2/4/2010 10:45 AM	File Folder
NM_run9	2/4/2010 10:45 AM	File Folder
NM_run10	2/4/2010 10:44 AM	File Folder

Figure 15: Multiple NONMEM runs for bootstrap

5 Example

One dataset from NONMEM is utilized to demonstrate PKgraph. This data set has 100 patients with covariates: ISM (gender), AGE, and WT. The data is fitted with one compartment model with zero order absorption and first order elimination.

As a text file, the fitting result from NONMEM is imported into PKgraph for further investigation and analysis. In the “open” dialog, we set up file format for reading with default parameters, and as a result, the input data shows up on the right panel while a message, “Data is loaded successfully” appears in the status bar at bottom of panel.

To further explore data, first, we choose “Bivariates” from “Exploratory Data Analysis” located at menu bar to check the scatter plots of interested variables(Figure~22, Figure~23). The option “cond” from the functional model interface helps user to draw conditional plots to seek patterns for subgroups. Certainly, users can also select “ggplot2” graphic package with different taste of figure. Next, we can take advantage of interactive techniques to look at maximum concentration by clicking second image button on the right panel. This will start ggobi and load related data. GGobi includes two windows: console window and plot window. In order to link figures together, users need to open all interested figures by “Display” option in the menu bar. The following figure clearly shows that maximum concentration comes from male patients (value: 1). To look at these data

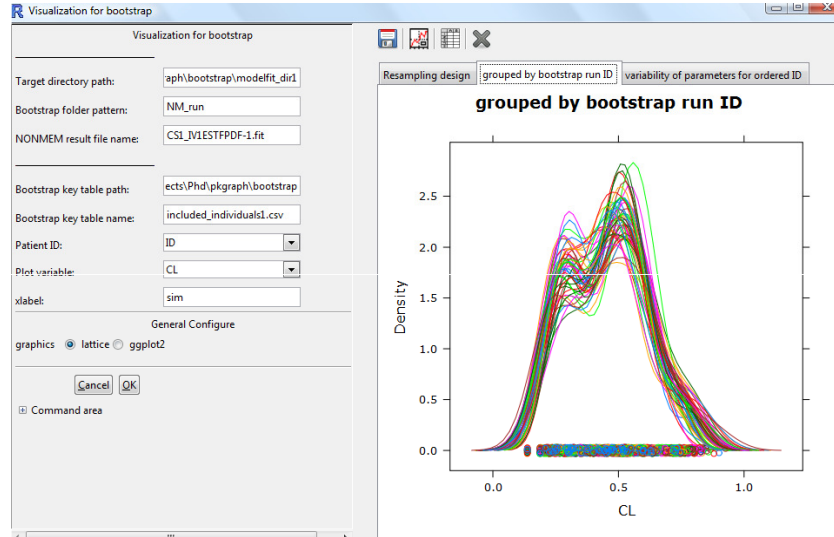


Figure 16: Parameters and results for bootstrap visualization

in detail, we go back to the figure graphical user interface and click third image button to check selected data set in ggobi. The selected data set pops up and links to patient with ID: 55. We repeat the same procedure for other variables to check patterns.

Next, we utilize “PK model” option to check model assumptions and diagnose model fitting. The program provides default names such as ID, TIME, COV, etc in order to automatically generate diagnosing results. After we match data variables to the default names, we can proceed to automatically generate routine goodness of fit plots for interested models. Figure ~24 is one of the results for structural model diagnostics.

To further look at the influential cases from same data set, we can link them together by “model validation” option in menu bar. In this process, we have 100 NONMEM runs available at directory: C:\ Projects\modelfit_dir1 using PsN function: cdd. Let’s input the path of these NONM runs, and select plot variable as “CL”. After clicking “OK”, we will have the parallel coordinates plot showing the CL variables for all NONMEM runs. From Figure ~25, we can see some patients have more influential effects on CL when records from these patients are deleted.

Let’s identify these influential cases with interactive graphics. Figure ~26 clearly demonstrates that these influential cases come from patient 52 and 20 based on multidimensional scaling and parallel coordinate plots.

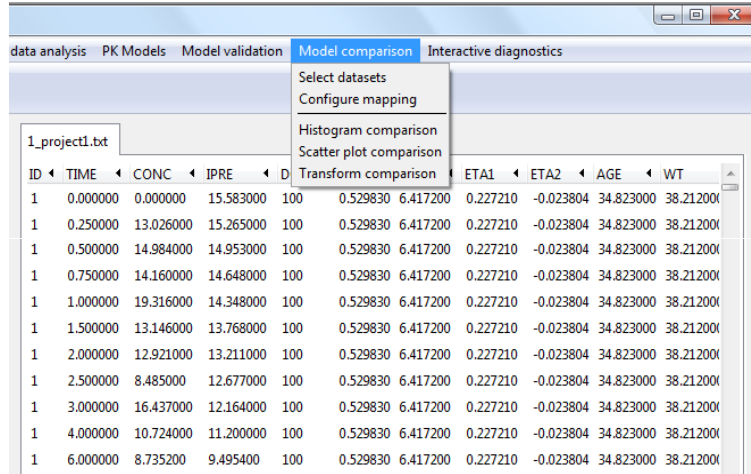


Figure 17: Menu items in *Model comparison*

In addition, we compare additive error model (2_CS1_IV1ESTFPDF.fit) with proportional error model (3_CS1_IV1ESTFPDF.fit2) by “model comparison” function in the menu bar. By comparing the distribution of two models, Figure ~27 does not find significant difference between two models for CL. In addition, using gender as a conditional variable, we found first model always gave a higher peak value for both male and female.

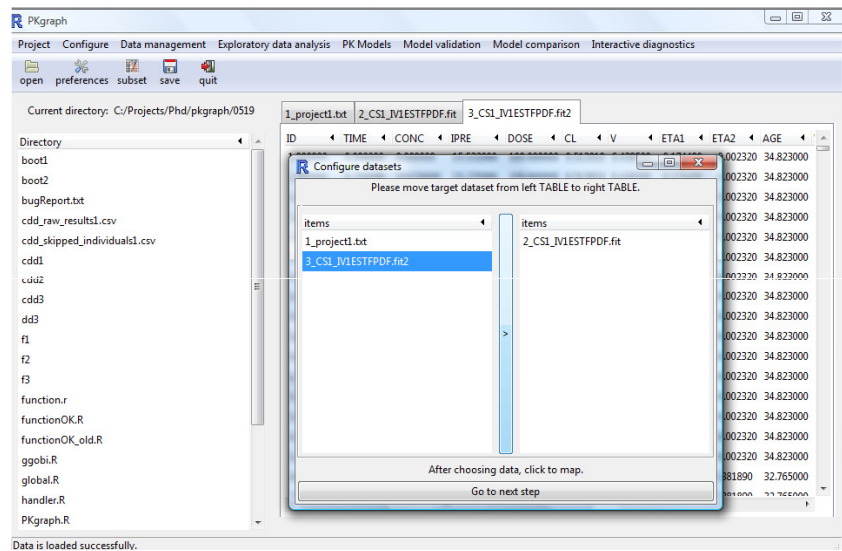


Figure 18: *Select datasets in Model comparison*

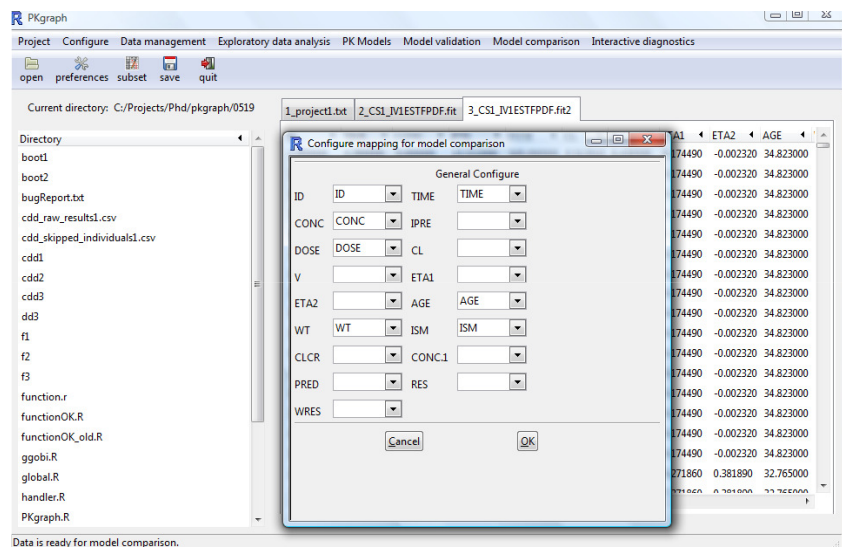


Figure 19: *Configure mapping in Model comparison*

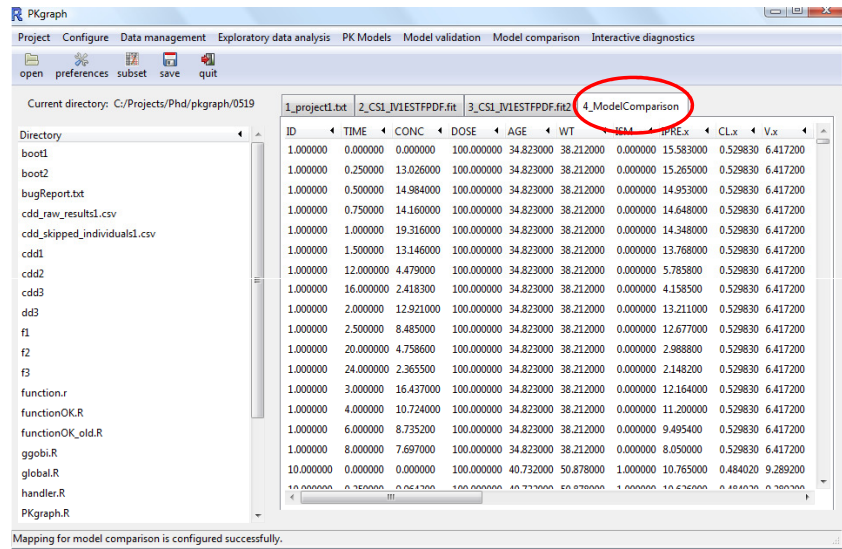


Figure 20: Current data set for *Model comparison*

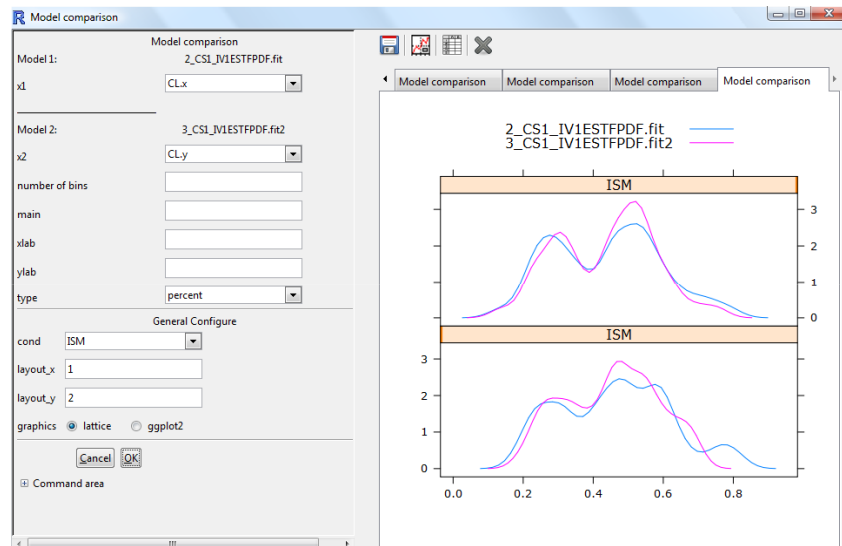


Figure 21: *histogram comparison* for *Model comparison*

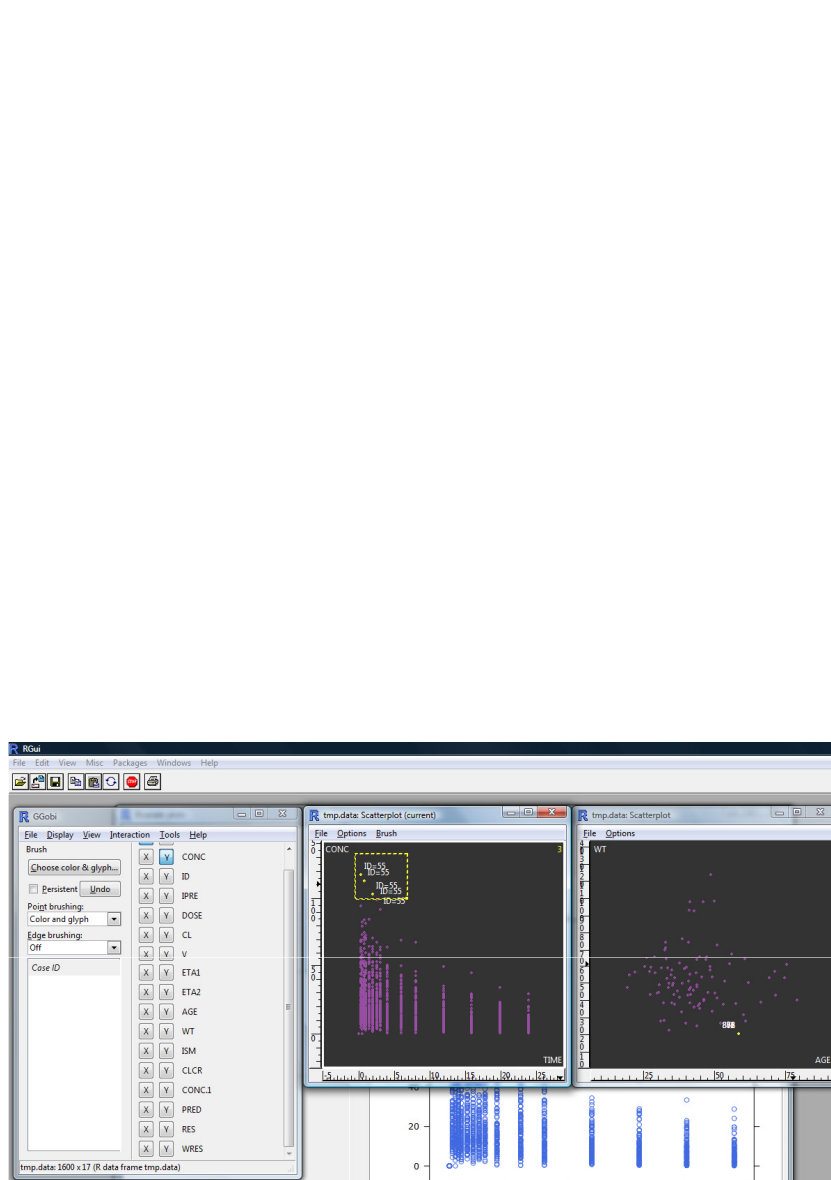


Figure 22: Exploratory data analysis. Peak is identified with brushing. This patient is from light weight and middle age group.

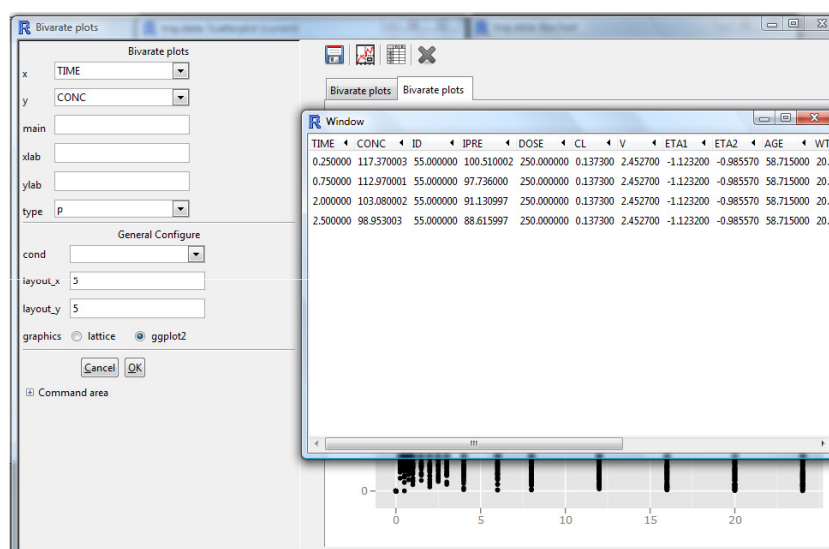


Figure 23: Exploratory data analysis. The detailed information for this patient is selected for investigation.

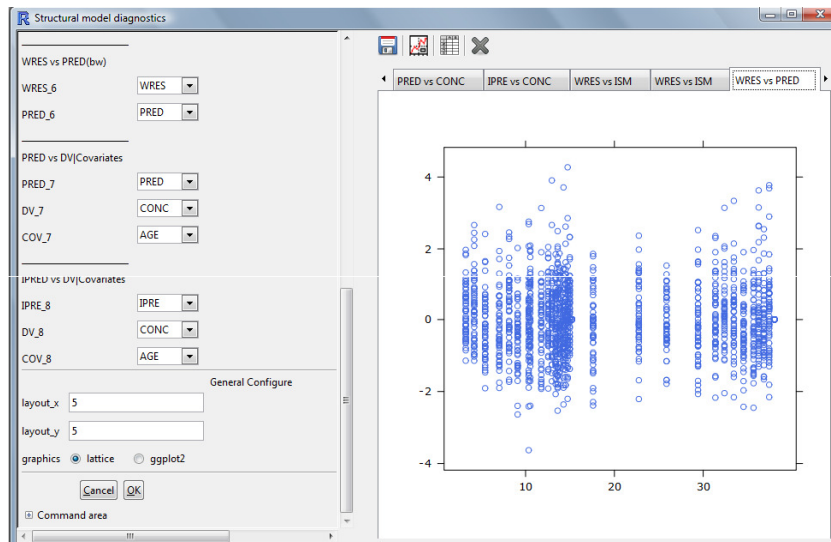


Figure 24: Structural model diagnostics.

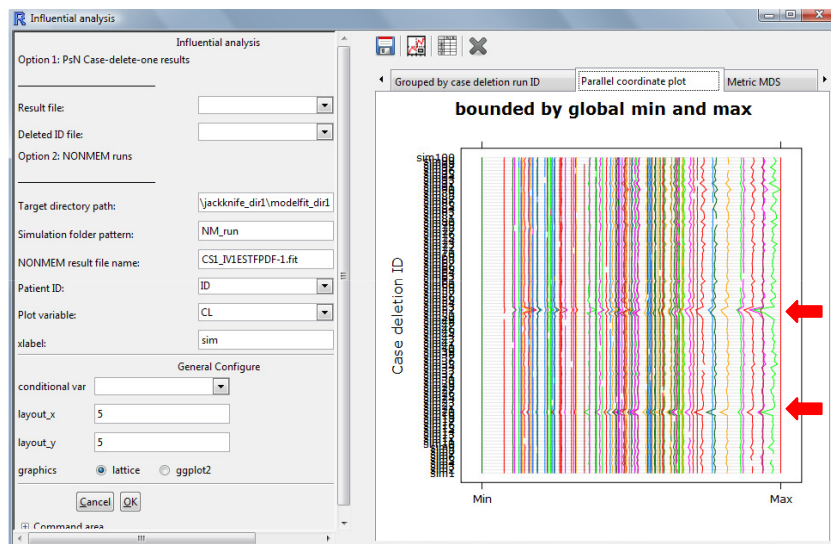


Figure 25: Influence analysis

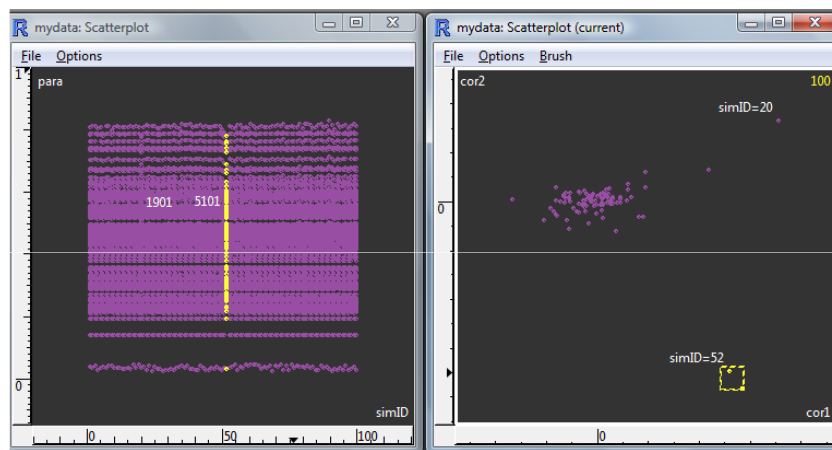


Figure 26: Influence analysis: linking results from multidimensional scaling and parallel coordinate plots.

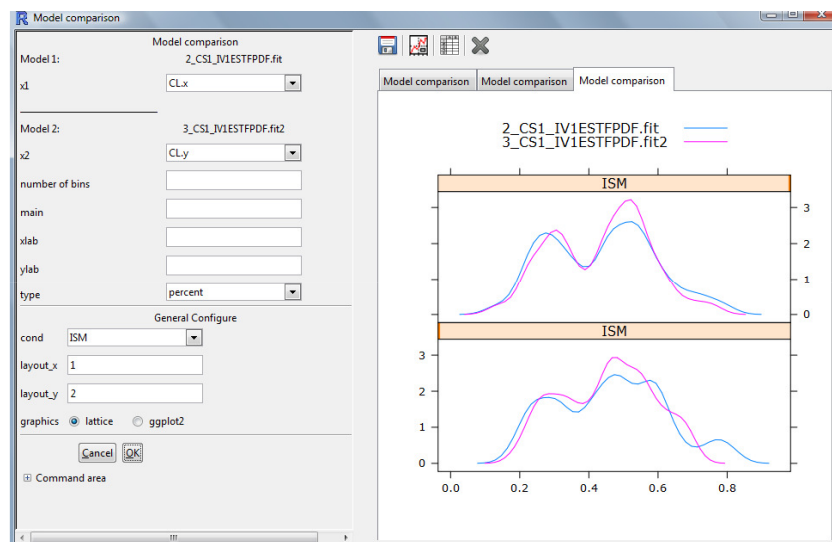


Figure 27: Histogram comparison for comparing distributions of CL from two models.