# User Guide for PKgraph Package

Xiaoyong Sun[†‡*]

February 23, 2011

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

## Contents



[*]johnsunx1@gmail.com

# 1 Introduction

Population pharmacokinetic (PopPK) modeling has become increasing 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 an R packaged 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)
*install.packages("RGtk2")*

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

4. Install gWidgets
*install.packages("gWidgets")*

5. Install cairoDevice
*install.packages("cairoDevice")*

6. Install gWidgetsRGtk2
*install.packages("gWidgetsRGtk2")*

7. Install lattice
*install.packages("lattice")*

8. Install ggplot2
*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.

## 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.

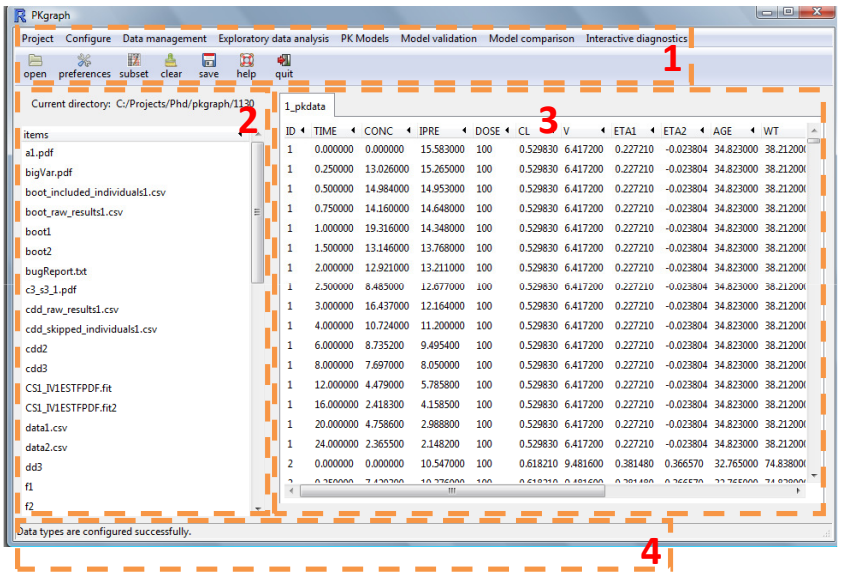


Figure 1: Main interface of PKgraph

### 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 facetted plots. A choice of lattice or ggplot2 graphics is provided. Note: At the bottom of this area, there is a module called "command area", which is for next release. It is not fully functional at this point.
- The tool bar allows users to 1) save plots, 2) open plots for interactive graphics (**ctrl + b for brushing data**), 3) display subset selection from ggobi, 4) save subset selection from ggobi and 5) close ggobi.
  - save plots: this button can save the current plot from **plot area**. The figure can be saved as pdf, jpg, tiff, png formats. This format is configured by *Set saving format* in the *Configure* menu item. For multiple plots generated with one parameter set, such as plots for observation concentration versus time conditioned on 50 patients, R will only keep the last few patients as one page in the **plot area**. This button will automatically save all pages for all patients with the specified figure format.
  - open plots for interactive graphics: this button opens two plots in ggobi for interactive graphics. The first plot is a time series plot for this data (observed concentration versus time), and the second plot is the current plot from **plot area**. These two plots are linked by patient ID. A specific feature of interactive graphics is to explore data by brushing. In the ggobi, users can use **ctrl + b for brushing data** to link two plots. For those who would like to use more advanced features of interactive graphics, ggobi manual is a good resource (http://www.ggobi.org/).
  - display subset selection from ggobi: this button helps user to visualize and analyze the brushed data from the previous step: *open plots for interactive graphics*. The brushed data is shown as a new dialog.
  - save subset selection from ggobi: this buttons save all brushed data from previous step: *open plots for interactive graphics*, *display subset selection from ggobi*.
  - close ggobi: close all related ggobi instances.
- 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
- Configure
- Data management
- Exploratory data analysis

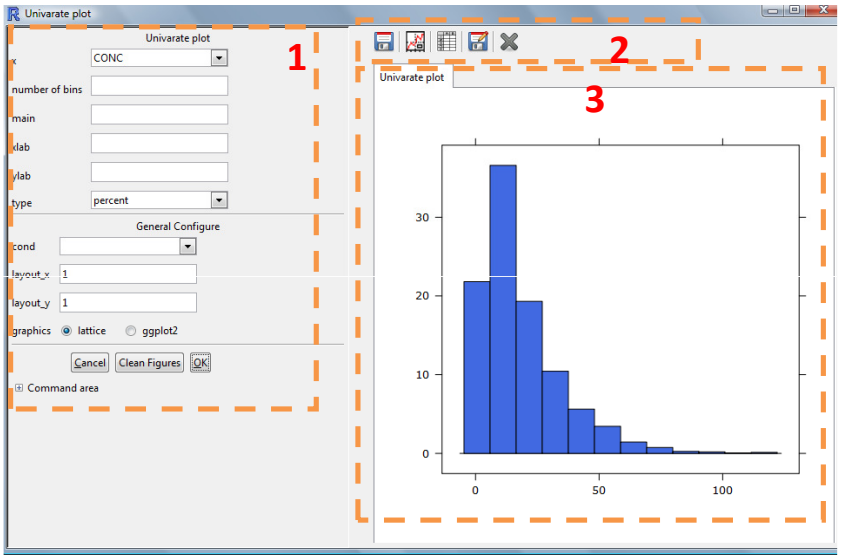


Figure 2: Graph interface of PKgraph

- PK models
- Model validation
- Model comparison
- Interactive graphics

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

# 4 Quick start

PKgraph targets audiences working in population pharmacokinetics models, and particularly those professionals who have only basic knowledge of R.

## 4.1 Input data

### 4.1.1 data.frame

PKgraph accepts one type of input data: data.frame. It can be model fit results from NONMEM, Monolix, SAS or R. This data.frame should include ID, time, observed concentration, individual predicated concentration, population predicted concentration, residuals, weighted residuals, parameters, random effects, etc. Details are as follows. Note: Abbreviated terms are explained in Table˜2.

- Exploratory data analysis: ID, Time, CONC
- PK models: See details in Table˜4.
- Model comparison: ID, Time, CONC, and interested variables from fit results, such as WRES, IPRE, etc. For this module, two data.frames come from two model fit results are required.

Dependent on the modeling software, users need to convert the model fit results to this single data.frame. After that, PKgraph can read in and diagnose the model fit results. For NONMEM, the *tab* file can be considered as this single data.frame, and read in R directly. For Monolix, there are a few output files, and users have to combine them as one single file, including all the interested variables.

An sample data from NONMEM,

```
> library(PKgraph)
> data(pkdata)
```

#### 4.1.2 NONMEM folders

For model validation, PKgraph accepts two kinds of model validation data: 1) results from PsN; 2) results from multiple NONMEM runs. For the first type of data, PsN has the following functions: bootstrap, case deletion and stochastic simulation. PKgraph provides the graphic ability to visualize the final results from PsN. For the second type of data, PKgraph can handle multiple NONMEM run folders and extract useful information to visualize. Please see details in **Model validation**.

### 4.2 Diagnose model

There are eight function menu items in the **main interface**. Each matches a functional module. They can be considered as two categories: **basic** and **diagnostic** modules. The basic module includes "Project", "Configure", "Data management" menu items, and the diagnostic module includes "Exploratory data analysis", "PK model", "Model validation", "Model comparison", "Interactive graphics" menu items. The five menu items in the diagnostic module can be utilized separately.

The basic module is utilized to read in data, configure data and manage data. The diagnostic module aims to test assumptions of population pharmacokinetic models. Please see the following section for details.

The workflow of diagnosing models is as follows,

### 4.3 Basic graphical parameters

In PKgraph, we use a lot of popular arguments from R graphics. Here is the explanation. For those who want to know more about these parameters, please check R manual.

| Abbreviated terms | Description |
|---|---|
| main | main title of the plot. It is the argument in R functions. |
| xlab/ylab | label of the x/y axis. It is the argument in R functions. |
| type | what type of plot should be drawn |
| layout_x | the number of columns in a multi panel display |
| layout_y | the number of rows in a multi panel display |
| cond | conditional variable |
| loess/lowess | locally weighted scatterplot smoothing |

Table 1: Basic graphical parameters

### 4.4 Abbreviations in the software

| Abbreviated terms | Description |
|---|---|
| ID | Patient ID |
| TIME | Time after dose |
| CONC | Observed concentration of drug in the body |
| PRED | Population predicted concentration |
| RES | Residual |
| WRES | Weighted residual |
| IPRED | individual predicted concentration |
| IWRES | Individual weighted residual |
| COV | Covariates |
| DV | Dependent variables (Usually observed concentration) |
| IDV | Independent variables (Usually time) |

Table 2: Abbreviated terms

The abbreviated variables used in the software are listed as Table˜2.

## 5 Menu items in main interface

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

### 5.1 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.
- *Clean data*: clean all loaded data.

- *Restore old workspace*: restore the workspace from the data and list you saved from previous step.
- *Exit*: exit from PKgraph.

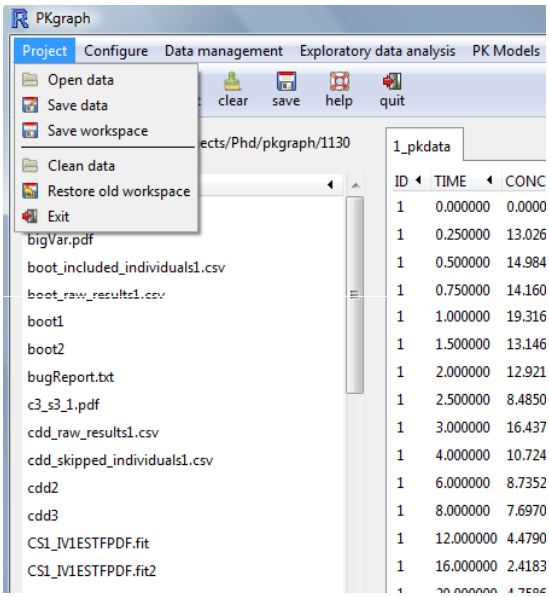


Figure 3: Menu items in *Project*

## 5.2 Configure

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

- *Set data type*: set the ID, TIME, CONC variables for current PK data. This configuration is used for integrative graphics to draw a time series plot automatically.
- *Set working directory*: change current working directory in R.
- *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 configured, the default one will be used. Note: to save figure in **graph interface**, users need to configure this menu first.
- *Set figure configuration*: color and loess can be selected for figures.

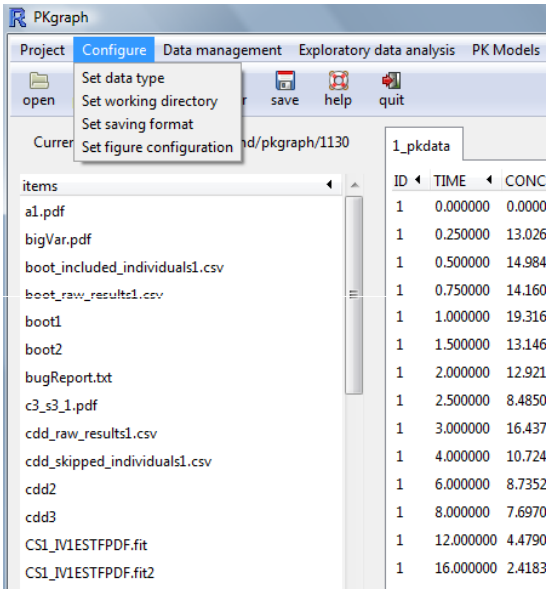


Figure 4: Menu items in *Configure*

## 5.3 Data management

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

- *Subset*: subset current data. After this, a new subset data will generate in the **data area** of **main interface**. And it will be the current working data for the following diagnosis. If users do not want to work on this data, users can click the tab of **data area** and select the proper one as the current working data.
- *Factor*: factor categorical variables. Graphical packages require the variable to be factor type in order to display the categorical symbol in figures. For example, in Figure˜21, after we make the "ISM" as a factor, the symbol "0/1" is show as the subtitle on the figure; otherwise, the name of variable "ISM" will be shown instead.

## 5.4 Exploratory data analysis

This menu item is utilized to explore data and screen patterns. The explanation for the basic parameter set is available at section: **Basic graphical parameters**. It has the following functions (Figure˜6),

- *Univariates*: plot univariate varaiables.
- *Bivariates*: plot bivariate variables.

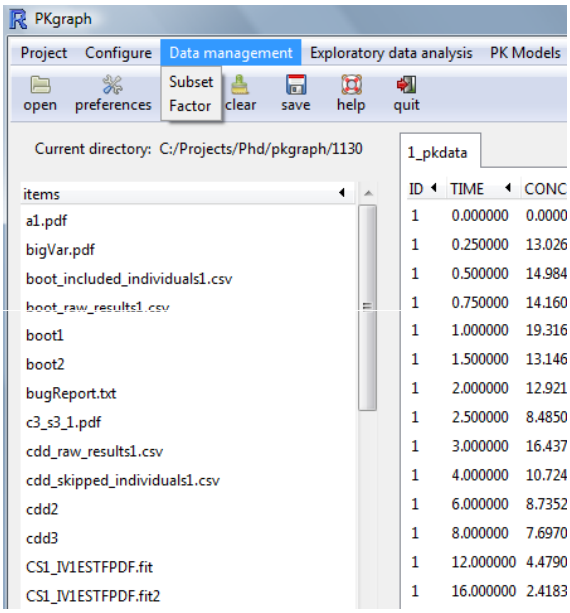


Figure 5: Menu items in *Data management*

- *Parallel coordinate plot*: Parallel coordinate plot for multivariate variables.
- *Scatterplot matrix*: Scatterplot matrix for multivariate variables.

### 5.4.1 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 five buttons on the top explained in section: **Graph interface**.

### 5.4.2 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.

### 5.4.3 Parallel coordinate plots

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

### 5.4.4 Scatterplot matrix

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

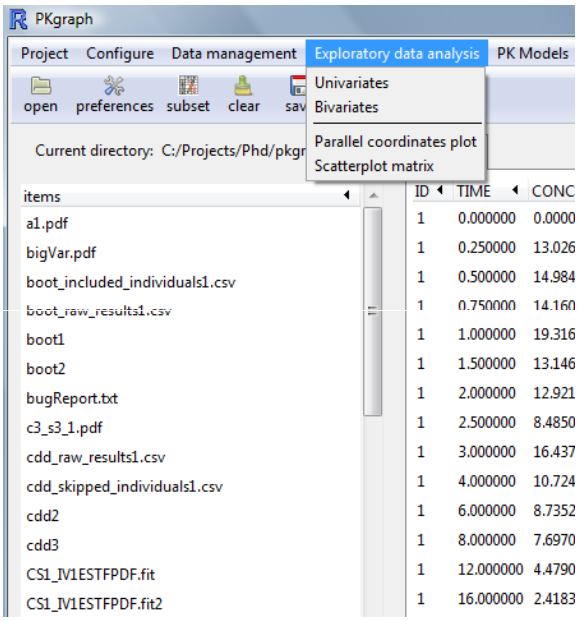


Figure 6: Menu items in *Exploratory data analysis*

## 5.5 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 eight functions (Figure˜7). **Configure model result** is required for the other seven functions. Users have to configure data variable first before going to specific model diagnostics.

### 5.5.1 Configure model result

This is the key step to match data variables to default metric system. By this step, fit results from any platform (NONMEM, Monolix, SAS, R) can be interpreted graphically in figures.

The interface for this function is shown in Figure˜8. The fixed column (left) is column name from data, and the selectable column (right) is variable name from the default metric system (Table˜3). By this matching, the other seven functions can be performed. However, these functions work independently, and some variables in the default metric system must be matched to those in real data (Table˜4).

### 5.5.2 Individual plots

Bivariate plot for each individual.

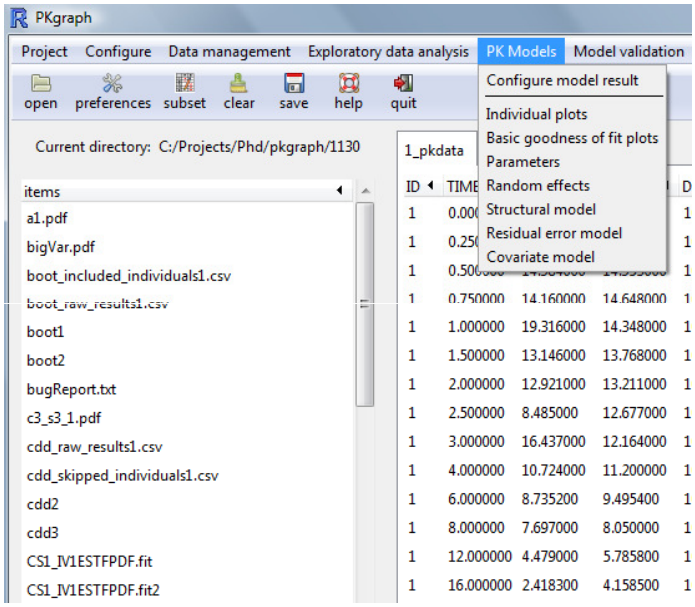


Figure 7: Menu items in *PK models*

| Package variable | Description |
|---|---|
| ID | Patient ID |
| TIME | Time after dose |
| CONC | Observed concentration of drug in the body |
| PRED | Population predicted concentration |
| RES | Residual |
| WRES | Weighted residual |
| IPRED | individual predicted concentration |
| IWRES | Individual weighted residual |
| COV | Covariates |
| DV | Dependent variables |
| IDV | Independent variables |

Table 3: Package metric system

### 5.5.3 Basic 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 IDV (time) and IPRED versus IDV (time).

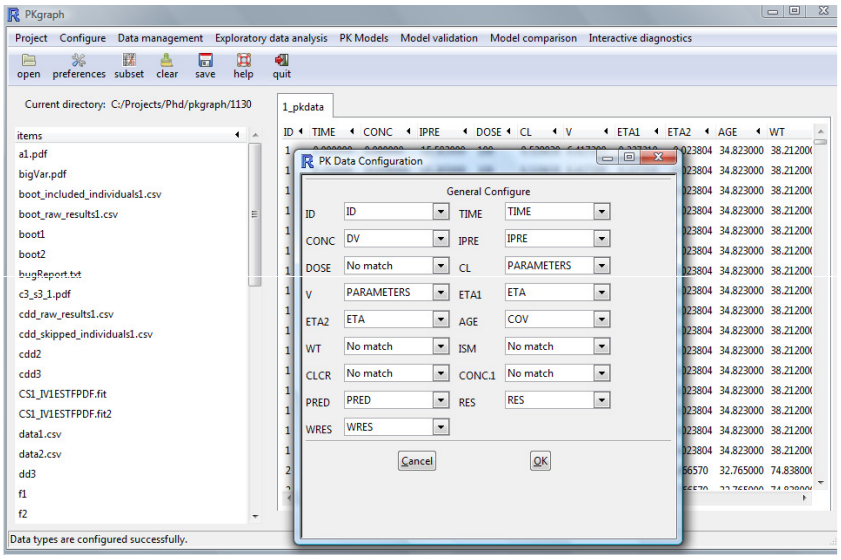


Figure 8: *Configure model result* in *PK models*

| **Functions** | **Required items to be selected in right column** |
|---|---|
| **Individual plots** | ID |
| **Basic goodness of fit plots** | PRED, IPRE, DV, IDV, WRES |
| **Parameters** | PARAMETERS |
| **Structural model** | PRED, IPRE, DV, IDV, WRES, COV |
| **Residual error model** | WRES, PRED, COV, IPRE |
| **Covariate model** | PARAMETERS, ETA, WRES, COV |
| **Random effects** | ETA |

Table 4: Required variables for different functions

### 5.5.4 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.

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

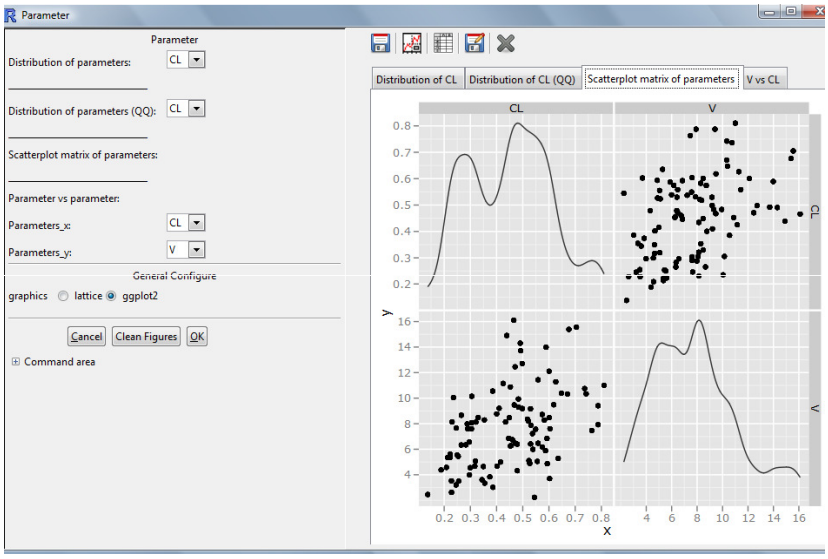


Figure 9: *Parameters* in *PK models*

### 5.5.5 Random effects

The assumptions for random effects also need to be tested for distribution and correlation by histogram, scatter plots or a scatterplot matrix.

### 5.5.6 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.

### 5.5.7 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.

### 5.5.8 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.

## 5.6 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. PKgraph mainly focuses on case deletion diagnostics and bootstrap.

It accepts two kinds of model validation data: 1) results from PsN; 2) results from multiple NONMEM runs. For the first type of data, PsN has the following functions: bootstrap, case deletion and stochastic simulation. PKgraph provides the graphic ability to visualize the final results from PsN. For the second type of data, PKgraph can handle multiple NONMEM run folders and extract useful information to visualize.

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).

### 5.6.1 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.

### 5.6.2 Visualization for influence analysis

This function is to visualize data from case deletion diagnostics (multiple NONMEM runs). Let's use multiple NONMEM run form PsN directly (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 folder name for multiple NONMEM runs. For this example, it is *NM_run*. It is a required parameter.
- *NONMEM result file name*: the file name for NONEM fitted result. In each NONMEM run, there should be a file with this name. It is required parameter.

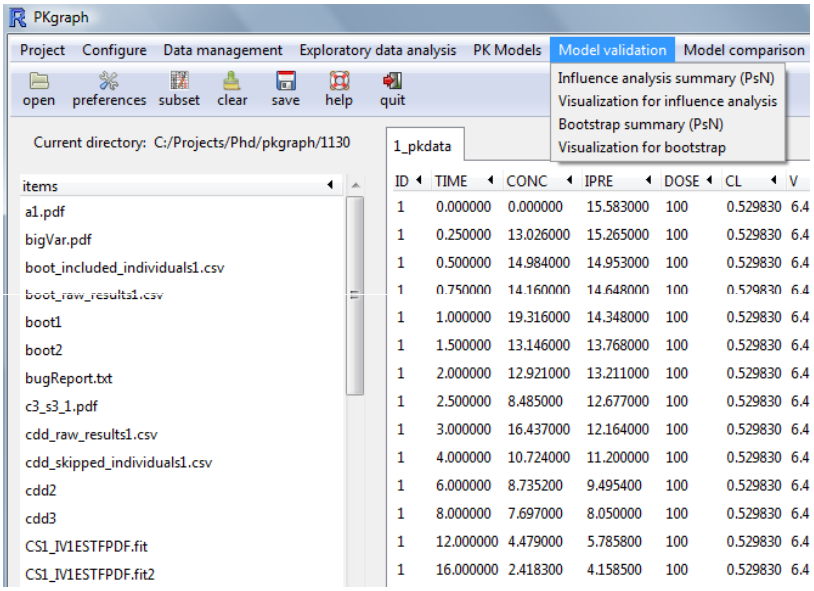


Figure 10: Menu items in *Model validation*

- *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.

### 5.6.3 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.

### 5.6.4 Visualization for bootstrap

This function is to visualize data from boostrap (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.

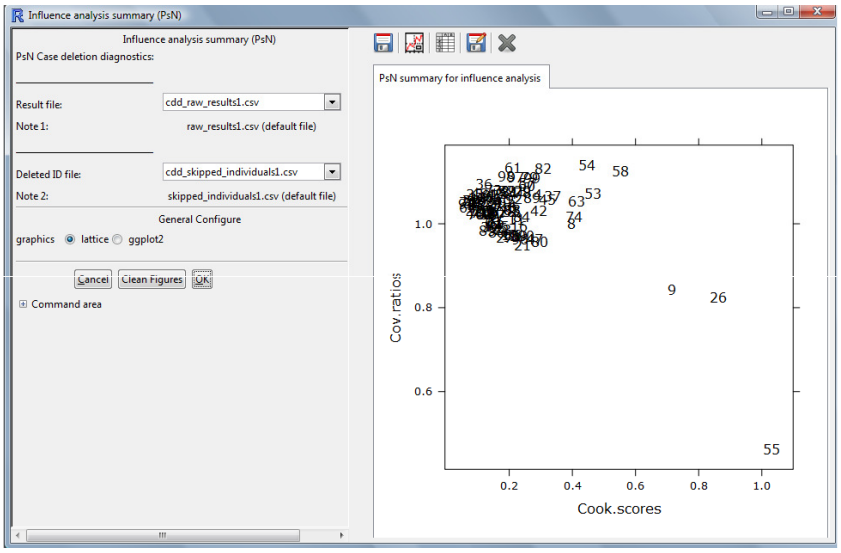


Figure 11: Influence analysis summary (PsN)

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

## 5.7 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. For example, we have to match TIME, ID, CONC, WT, etc from original data, but not match those variables from model fit, such as ETA, RES, WRES, etc. When

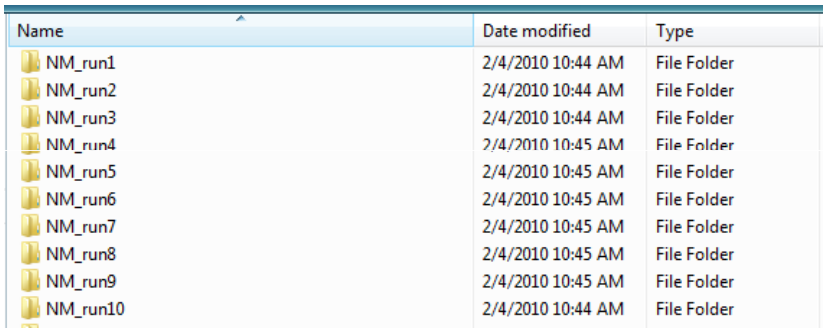


Figure 12: Multiple NONMEM runs for case deletion diagnostics

all parameters are set, the program offers three choices for comparison: "histogram comparison" (distribution comparison), "scatter plot comparison" and "transform comparison".

### 5.7.1 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.

### 5.7.2 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.

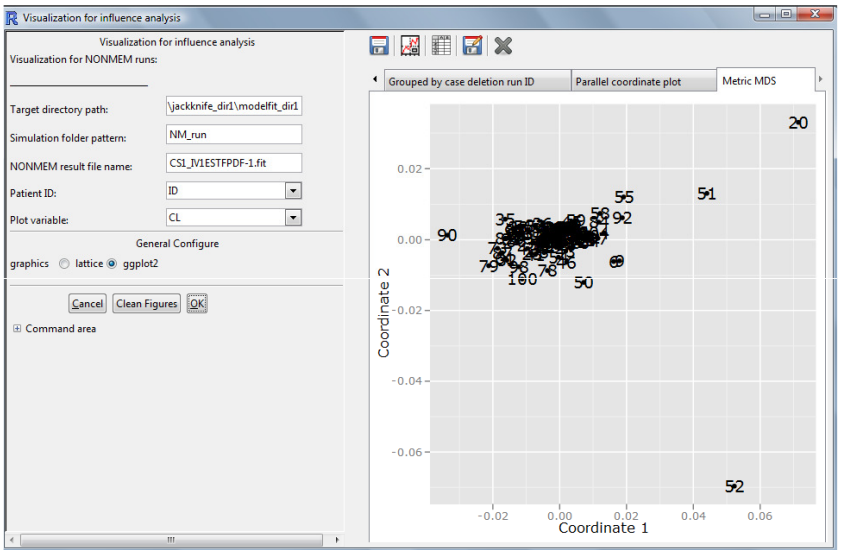


Figure 13: Parameters and results for case deletion diagnostics

### 5.7.3 Comparison

"histogram comparison" enables to compare distributions of matching parameters from two models. "scatter plot comparison" provides an 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.

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". The result is shown in (Figure˜21) for comparing *CL*.

## 5.8 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

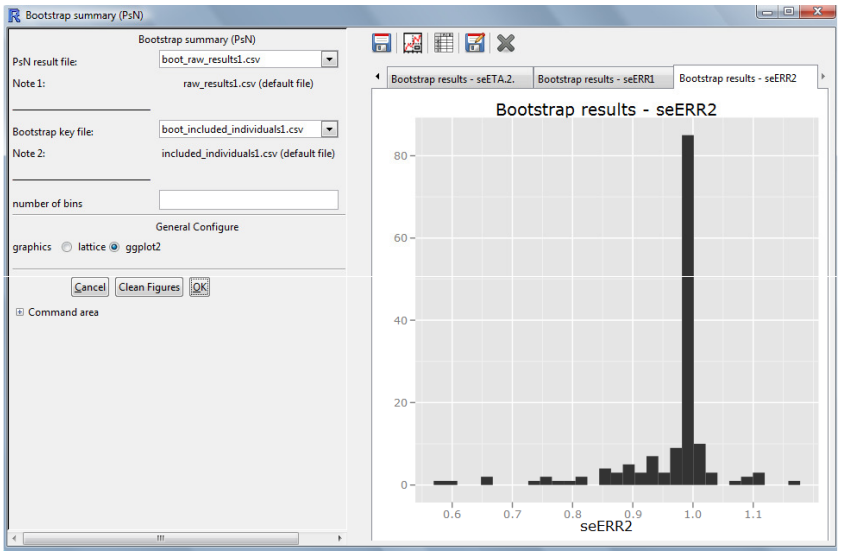


Figure 14: Bootstrap summary (PsN)

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 at `http:`
`www.ggobi.org` if needed.

# 6 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.

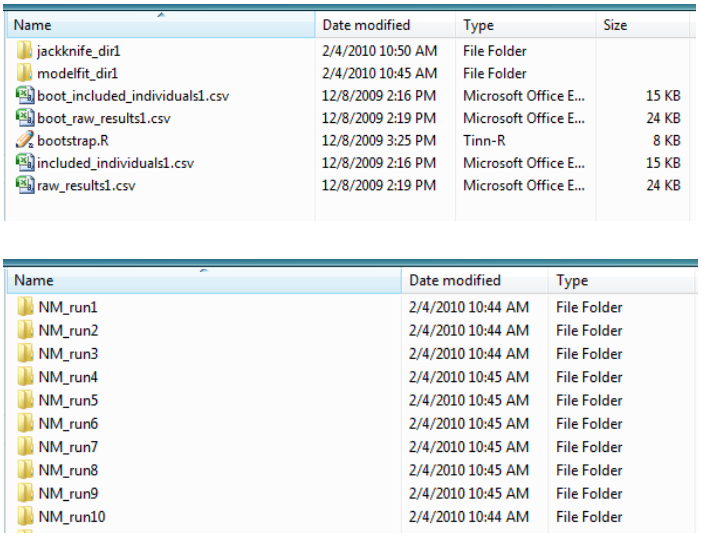


Figure 15: Multiple NONMEM runs for bootstrap

Alternatively, to make the input process flexible, users can input data into R first and then load data from "Data from R environment" in the "open" dialog. All the fitted results from a wide variety of software including NONMEM, SAS, etc can be loaded into this package.

```
> library(PKgraph)
> data(pkdata)
> PKgraph()
```

Figure˜22 demonstrates how to load default data in the software.

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˜23, Figure˜24). 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 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

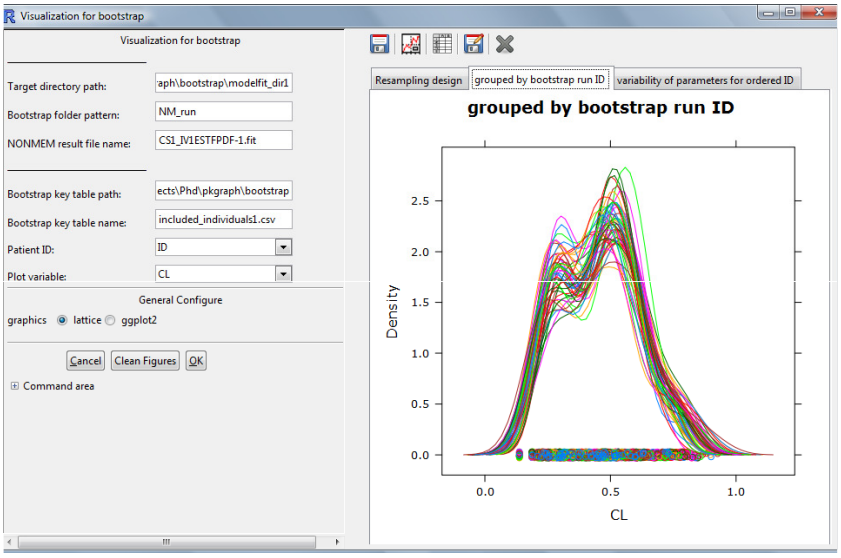


Figure 16: Parameters and results for bootstrap visualization

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 ˜25 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 ˜26, 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 ˜27 clearly demonstrates that these influential cases come from patient 52 and 20 based on multidimensional scaling and parallel coordinate plots.

In addition, we compare additive error model (2_CS1_IV1ESTFPDF.fit) with proportional error model (3_ CS1_IV1ESTFPDF.fit2) by "model comparison"

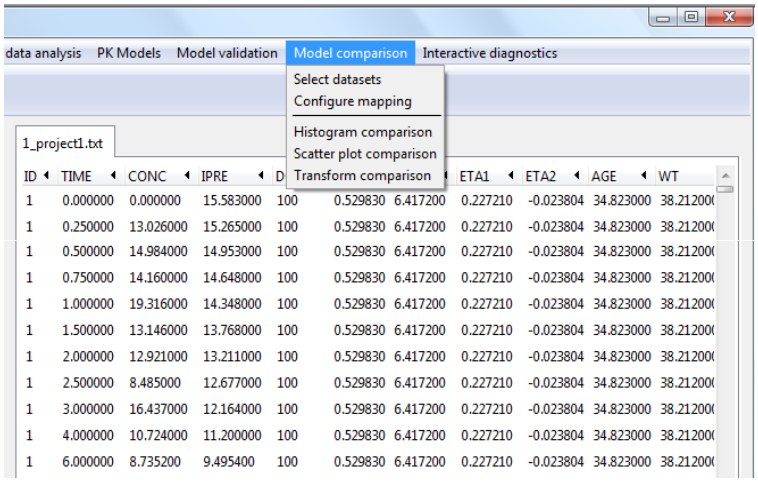


Figure 17: Menu items in *Model comparison*

function in the menu bar. By comparing the distribution of two models, Figure ˜28 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.

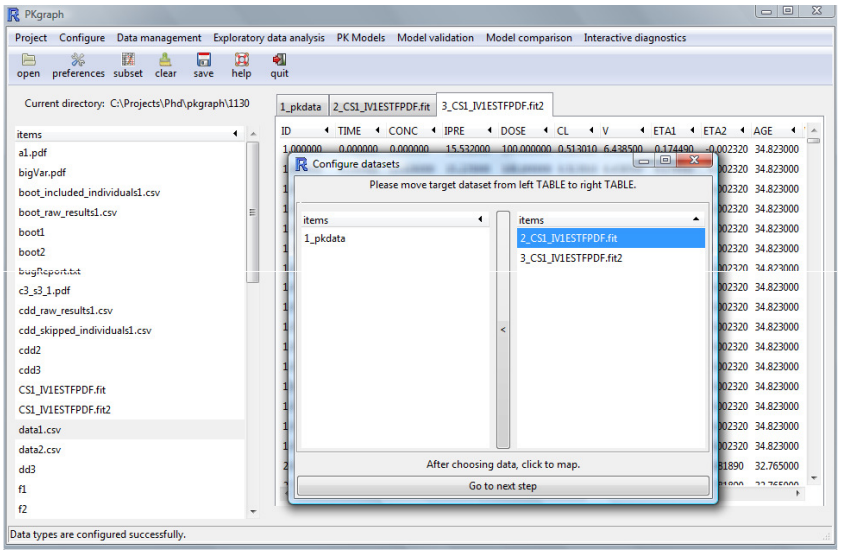


Figure 18: *Select datasets* in *Model comparison*

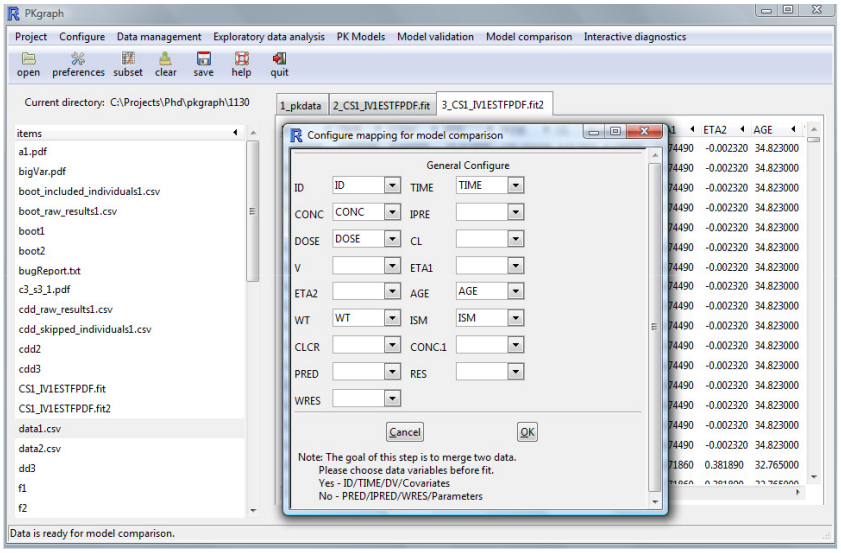


Figure 19: *Configure mapping* in *Model comparison*

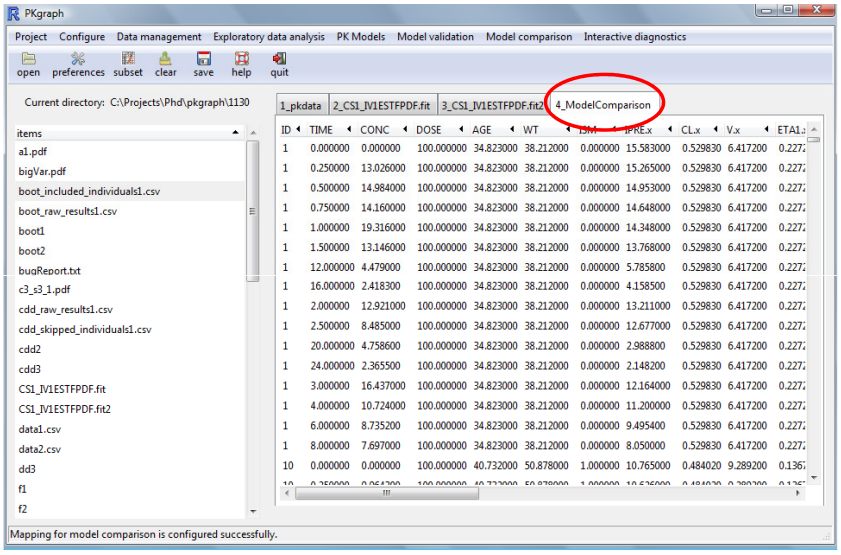


Figure 20: Current data set for *Model comparison*

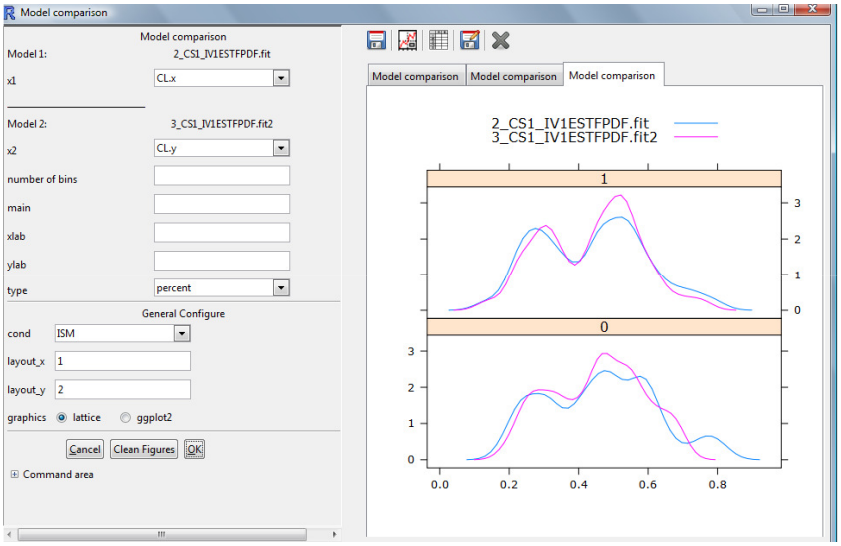


Figure 21: *histogram comparison* for *Model comparison*

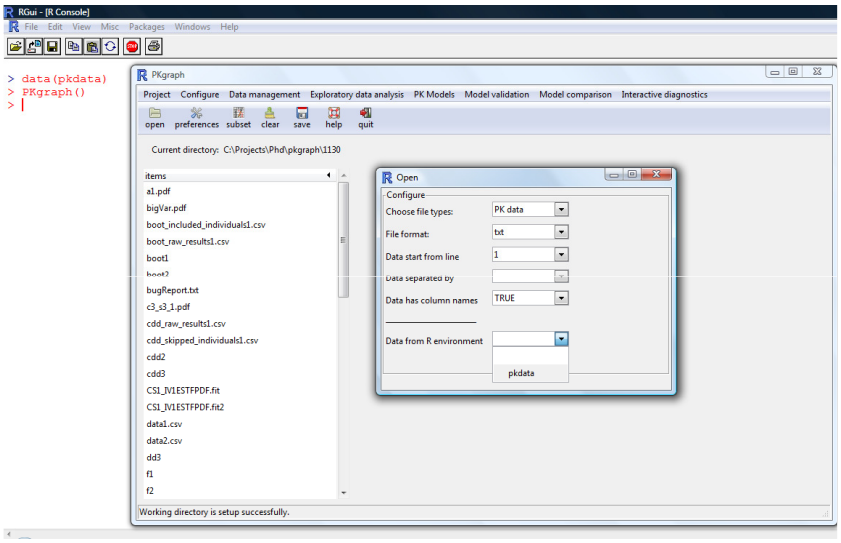


Figure 22: Load default data from "Open" dialog. After loading data with "data(pkdata)", users can select "pkdata" from Data from R environment in the "Open" dialog.

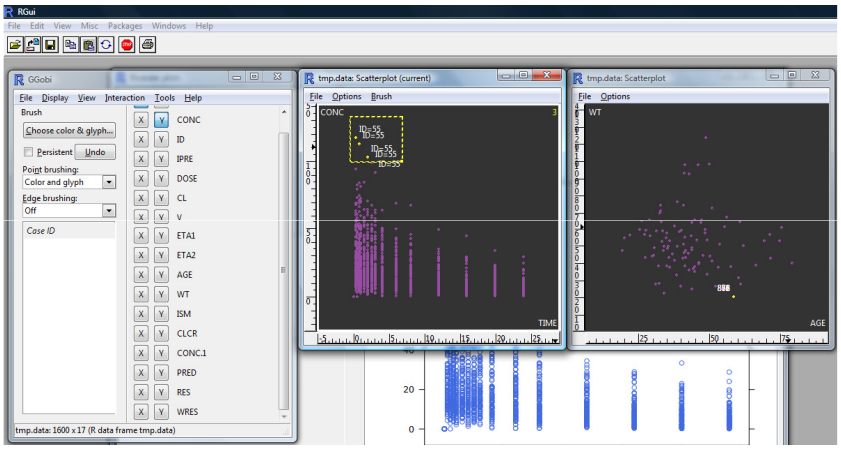


Figure 23: Exploratory data analysis. Peak is identified with brushing. This patient is from light weight and middle age group.

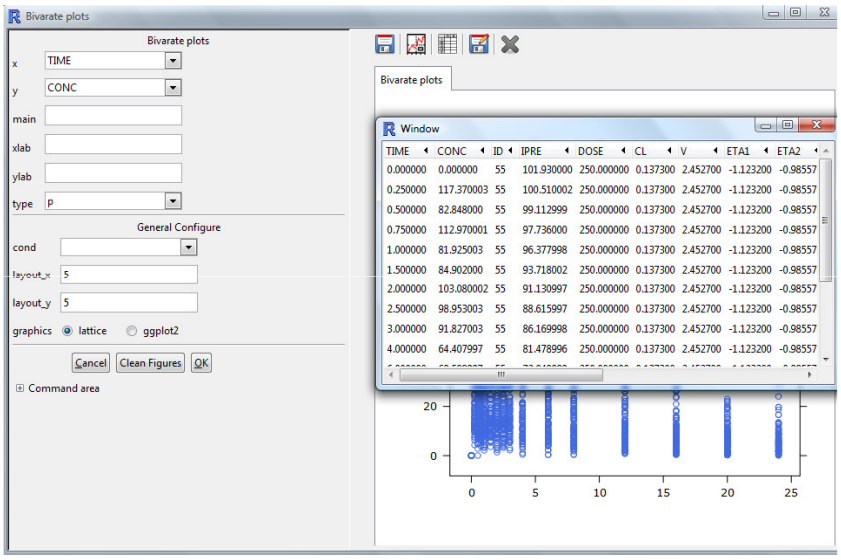


Figure 24: Exploratory data analysis. The detailed information for this patient is selected for investigation.

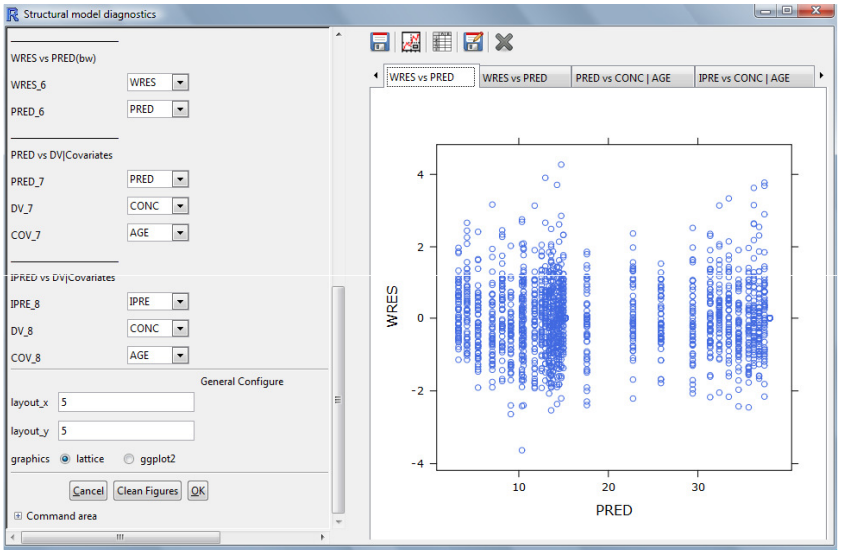


Figure 25: Structural model diagnostics.

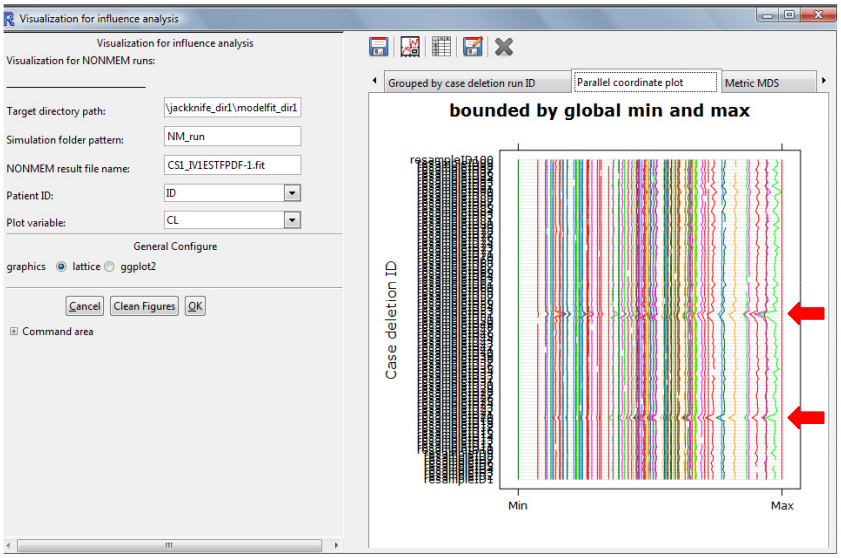


Figure 26: Influence analysis

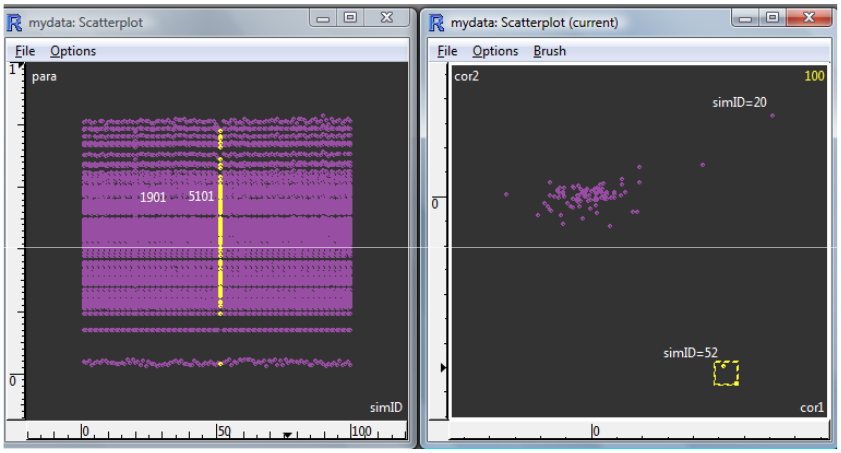


Figure 27: Influence analysis: linking results from multidimensional scaling and parallel coordinate plots.

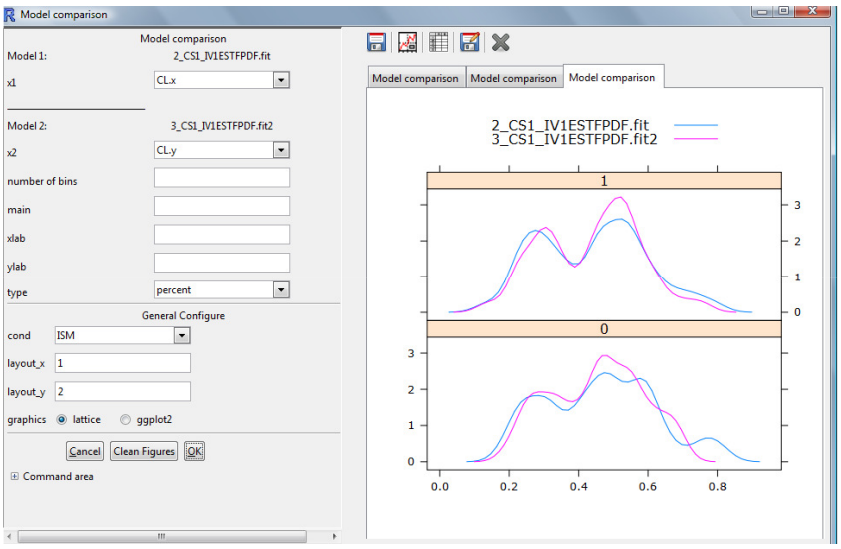


Figure 28: Histogram comparison for comparing distributions of CL from two models.