

Quick Guide for PKgraph Package

Xiaoyong Sun^{†‡*}

May 31, 2010

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

Contents

1	Introduction	1
2	Installation	2
3	PKgraph infrastructure	2
3.1	Graphical user interfaces	3
3.1.1	Main interface	3
3.1.2	Graph interface	3
3.2	Functional module	4
4	Functions	4
4.1	Project	5
4.2	Configure	5
4.3	Data management	6

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

*sunx1@iastate.edu

packages: NONMEM, Xpose and PsN for diagnosing models.

PKgraph is a 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)

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.

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, 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.

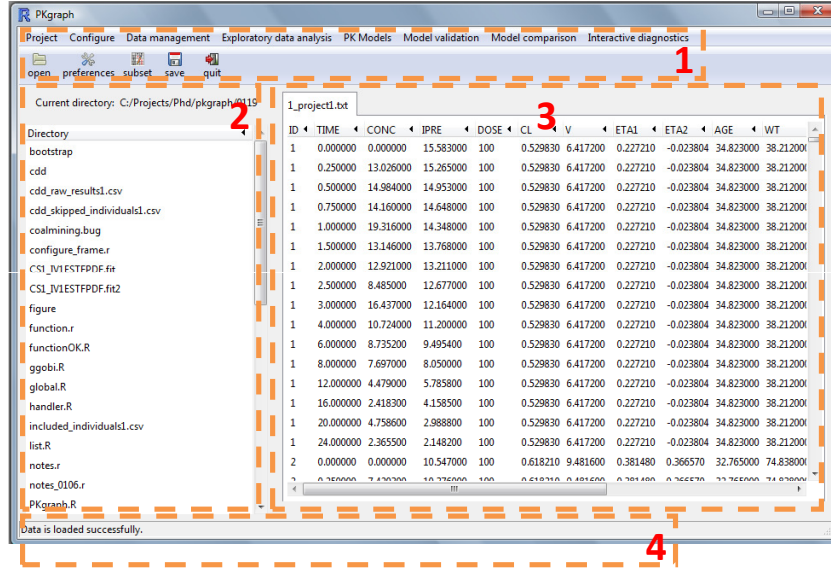


Figure 1: Main interface of PKgraph

3.2 Functional module

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

- Project
- 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.

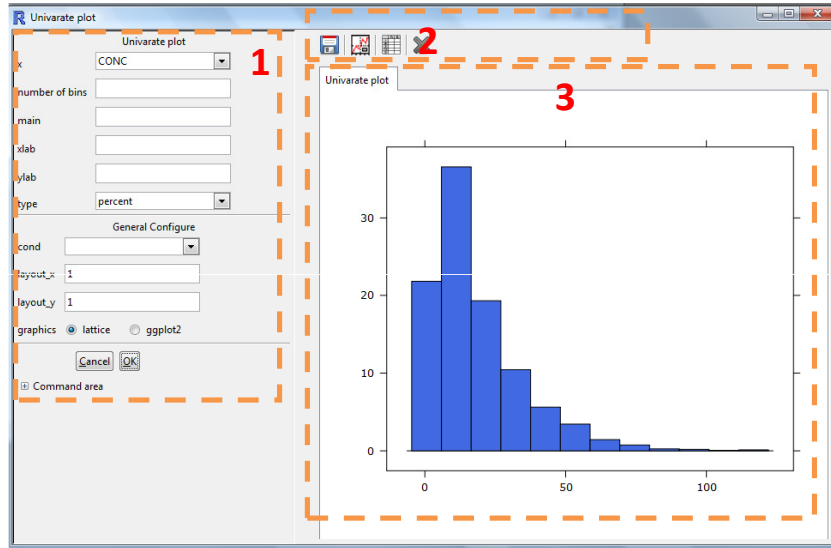


Figure 2: Functional model interface of PKgraph

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.

4.2 Configure

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

- *Set working directory*: change current directory.

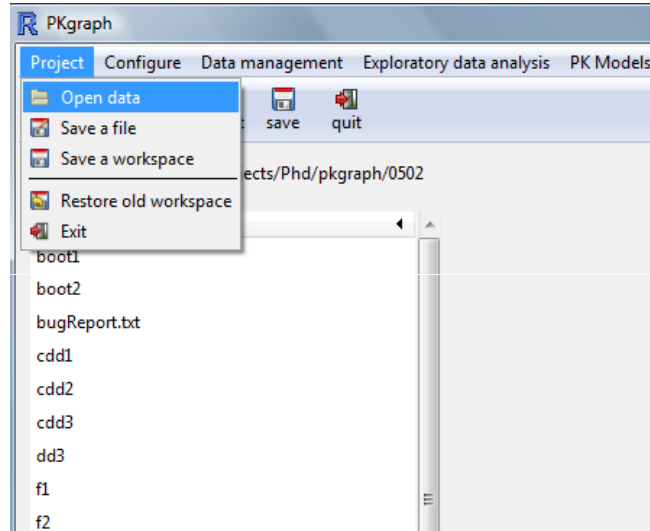


Figure 3: Menu items in *Project*

- *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.

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.

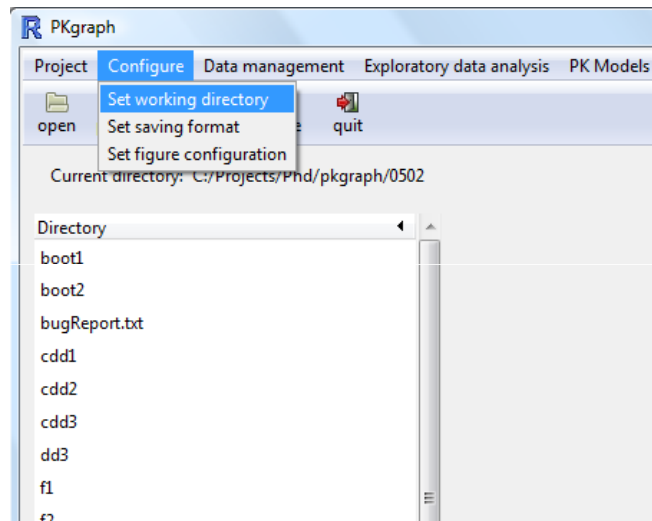


Figure 4: Menu items in *Configure*

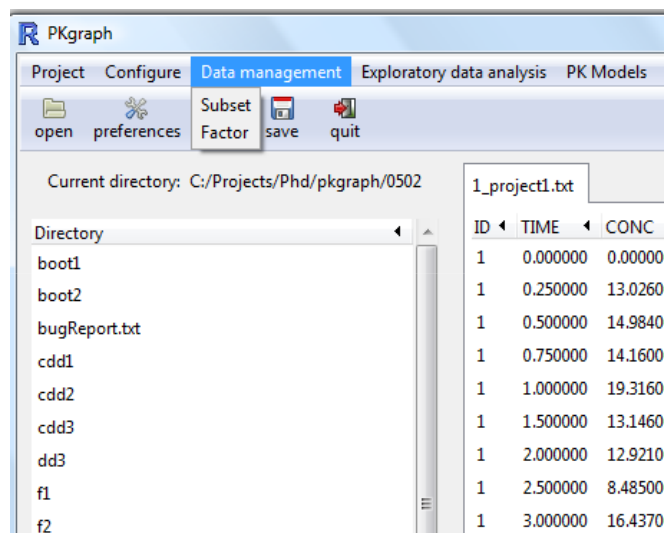


Figure 5: Menu items in *Data management*