Piraña

and the Piraña cluster

Version 2.3.0 "Oahu"
Installation guide and Manual

May 12, 2010

Contents

Chapter 1

Introduction

Piraña is a convenient graphical user interface to NONMEM, PsN and Wings for NON-MEM. The major aim of the Pirana project is to supply both novice and expert pharmacometricians with a useful modeling environment, and making the modelling workflow more efficient. Development of Piraña and the PCluster was started at the end of 2006, and is ongoing.

Piraña can just be used for modeling on stand-alone computers, or to connect to cluster-systems like MOSIX or SGE. Moreover, Pira/ na offers an integrated cluster solution, PCluster, which is a no-cost solution for distributed NONMEM computing intended for small modeling groups, e.g. in hospital or academic settings.

As efforts were made to make the use of Piraña intuitive, this manual is kept to a minimum, but should contain all the information you need to set-up and use this software. Also note that all buttons in Piraña display information about it's action when hovered over with the mouse pointer. Furthermore, the Piraña website at pirana.sf.net contains a FAQ and a mailinglist. If you're still in the dark at some point, do not hesitate to contact us.

Ron Keizer
ronkeizer@gmail.com
Coen van Hasselt
coenvanhasselt@gmail.com

Chapter 2

Piraña

2.1 Setting up the Piraña environment

Requirements are summarized below, followed by some additional details on the installation procedure.

2.1.1 Required / recommended software

Although the only real requirement is an installation of NONMEM, some additional software is highly recommended for optimal use of Piraña, while other software might be useful as well.

NONMEM Piraña can use both standard 'from-CD'-installation of NONMEM and NMQual NONMEM installations. NONMEM doesn't have to be installed on your local PC, since Piraña can also connect to other PC's/clusters that have NONMEM installed.

R This open-source software can be obtained from http://www.r-project.org/, and is highly recommended for optimal use of Piraña.

Perl and PsN On Windows, an installation of Perl is not required as Piraña is distributed as executable. For the Linux and Mac versions, only perl-sourcefiles are supplied, so an installation of Perl is required in that case. It is recommended to obtain the latest version from ActiveState (http://www.activestate.com/). Several additional packages need to be installed as well, which is detailed in the section "Installation procedure on Linux"

PsN Strictly, Piraña does not require PsN installed, although the PsN-toolkit is highly recommended. The latest version of PsN can be obtained from http://psn.sourceforge.net/.

WFN Not required. Piraña offers basic support for Wings for NONMEM. http://wfn.sourceforge.net/.

XPose This R-package for model fit evaluation can be obtained from http://xpose.sourceforge.net. From the data-files overview in Piraña, XPose datasets can be selected and loaded in R/XPose. Xpose scripts to automate creation of plots from within Piraña are included.

NMQual Recommended for keeping an audit trail of NM installations and autamatically performing bug-fixes. NMQual also requires Perl and the module XML::XPath installed. More details can be found at the NMQual website (http://www.metruminstitute.org).

2.1.2 Installation procedure on Windows

Download the installer from the website (http://pirana.sf.net), and install to any location on your hard-drive, e.g. 'C:\Program files\Pirana'. Before exploring Piraña's functions, you should first check if your preferences ('File → Preferences...') and software integration ('File → Software...') are set correctly. The most important settings are detailed in the section 'Preferences and software'. Piraña is tested on Windows XP, Windows Vista, and Windows7.

2.1.3 Installation procedure on Linux

For Linux, a compiled binary (executable version) of Piraña is not made available, and the source-code is to be executed by Perl. Therefore, it is necessary to have a distribution of Perl installed (which is the case for most Linux distributions). For Piraña to be able to create the GUI, the X11 development libraries (libX11-dev) should be installed, as well as the Perl/Tk module. In Ubuntu / Debian, you can use the Synaptic package manager to install these, or using apt from the shell:

```
sudo apt-get install libX11-dev perl-tk
```

Moreover, as Piraña makes use of a number of publicly available Perl-modules, these should also be installed. Some of these modules are likely to be already installed with you current Perl distribution, while others have to be installed manually. Below is a short guidance on how to install these modules. Further guidance on installing Perl modules can be found here: http://www.cpan.org/modules/INSTALL.html). To make a connection to the Perl module archive (CPAN), type:

```
sudo perl -MCPAN -e shell
```

The following commands may be needed to set up the CPAN shell to be able to correctly 'make' the modules into your Perl distribution.

```
o conf make /usr/bin/make
o conf make_install_make_command 'sudo make'
o conf commit
```

Next, use the 'install' command to install the following modules into your Perl distribution (mind the case-sensitivity for the module names). Some of these may already be installed, which will be noticed by your system, and which is OK. If you cannot install these modules from CPAN, you have to installed these modules (and their dependencies) manually.

```
install Tk::PlotDataset
install HTTP::Date
install List::Util
install DBI
install DBD::SQLite
install Math::BigFloat
```

Additionally, several the required modules cannot be installed directly from CPAN. These modules are supplied with Pirana (in the folder '/packages') and should be installed manually. From within each of the two package folders, execute in a shell:

```
perl Makefile.PL
make
sudo make install
```

Note: Checking whether Perl modules are installed correctly can be done by executing the following in the terminal window, e.g.: perl -e 'use Tk' which should result in no reported error messages.

After installing these Perl modules, copy the entire pirana folder contained in the zip-file to e.g. your home folder (/home/username/) or to /opt/pirana/. Make sure that all perl files in that folder have execution rights. To grant these rights to yourself you can execute the following in the shell from within the Piraña folder:

```
 sudo chmod 711 -R *
```

Piraña can now be started from the command line with 'perl pirana.pl' from within the Piraña folder. Piraña was tested on Ubuntu (9.04, Jaunty Jackalope, and higher) and OpenSUSE (11.1) with the Activestate Perl distribution (5.10.0), but should work on any Linux distribution with X-windows and Perl/Tk installed.

When accessing a Linux installation of Piraña from another PC through SSH-tunneling, it is essential to use the PuTTy terminal directly, and not plink.exe. The latter program tends to make Piraña crash often for unknown reasons. OpenSSH may also be used.

2.1.4 Installation procedure on Mac OS X

On Mac, you should have the X11 environment and the Xcode tools installed, which are included as optional installs on the (Snow) Leopard install DVDs. Pirana was tested using ActiveState Perl 5.10.0, but should work with any Perl distribution. The same Perl libraries as mentioned under 'Linux installation' should be installed from CPAN, and for which the same procedure can be followed.

2.1.5 Preferences and Software

Although most preferences will be correct by default, it is recommended to check at least the settings detailed below.

name_researcher Your name (no spaces allowed).

ext_ctl File-extension of NONMEM control streams/model files. Default is .mod

ext_csv File-extension of comma separated files. Default is .csv

ext_res File-extension of NONMEM output files. Default is .lst

Moreover, Piraña needs to know where other important software is installed, which is specified under 'Software' from the 'File' menu. References to software that you do not have installed, may be disregarded as they are ignored by Piraña. The most important references are:

editor The location of a code-editor, preferably with syntax-highlighting. We recommended the use of Emacs and ESNM (http://esnm.sourceforge.com/), PSPad (http://www.pspad.com/) or ConTEXT (http://www.contexteditor.org/). If none is entered, or a non-existing program is specified, Piraña will use its built-in NM-TRAN editor.

psn_dir The location of PsN-toolbox. Default is C:/perl/site/lib/PsN_x_x_x on Windows.

psn_toolkit This is the location of the PsN commands. (Default is C:/perl/bin on Windows, or /usr/bin/ on Linux). Also check perl_dir. If this folder is already in your PATH, you can leave it empty.

nmqual_dir Path to NMQual, e.g. C:/NMQual

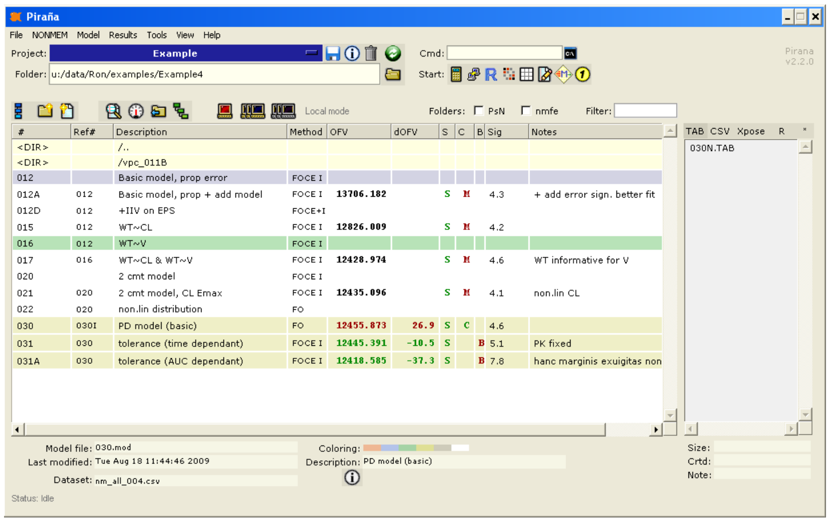
r_dir Path to R, e.g. C:/Program Files/R/R-2.11.0. On Windows, at first start-up of Piraña it will search for the latest version of R that is installed, and automatically updates this setting accordingly.

spreadsheet The location of your spreadsheet application, e.g. Excel or Gnumeric.

2.2 Using Piraña

Basic functions

The Piraña window is divided in a large table on the left and a smaller one on the right. The main one on the left shows an overview models and results, while the right side shows either data files, R-scripts, XPose datasets or all files in the active folder. All buttons in Piraña's main screen are accompanied by a short description which is displayed if hovered above with the mouse pointer. Some specific functionality is detailed below. By selecting a model or (data-)file and right-clicking the mouse, a list with actions on models or run results becomes available.



Project: Example
Folder: u:/data/Ron/examples/Example4
Cmd:
Start:
Folders: PsN nmfe Filter:

#	Ref#	Description	Method	OFV	dOFV	S	C	B	Sig	Notes
<DIR>		/..								
<DIR>		/vpc_011B								
012		Basic model, prop error	FOCE I							
012A	012	Basic model, prop + add model	FOCE I	13706.182		S	M		4.3	+ add error sign. better fit
012D	012	+IIV on EPS	FOCE+I							
015	012	WT~CL	FOCE I	12826.009		S	M		4.2	
016	012	WT~V	FOCE I							
017	016	WT~CL & WT~V	FOCE I	12428.974		S	M		4.6	WT informative for V
020		2 cmt model	FOCE I							
021	020	2 cmt model, CL Emax	FOCE I	12435.096		S	M		4.1	non.lin CL
022	020	non.lin distribution	FO							
030	030I	PD model (basic)	FO	12455.873	26.9	S	C		4.6	
031	030	tolerance (time dependant)	FOCE I	12445.391	-10.5	S		B	5.1	PK fixed
031A	030	tolerance (AUC dependant)	FOCE I	12418.585	-37.3	S		B	7.8	hanc marginis exuigitas non

030N.TAB

Model file: 030.mod
Last modified: Tue Aug 18 11:44:46 2009
Dataset: nm_all_004.csv
Coloring:
Description: PD model (basic)
Size:
Crtd:
Note:
Status: Idle

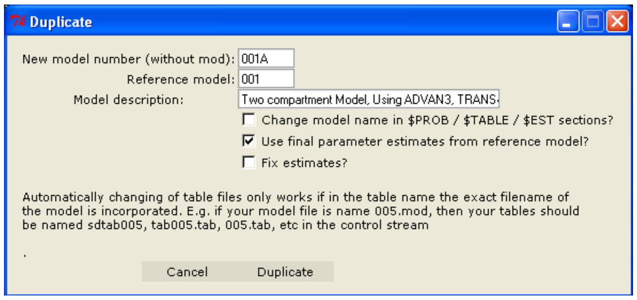
Figure 2.1: Piraña main screen

Model management

The main table is the place where all models and subdirectories in the current working directory are displayed. Only models are displayed that have a file-extension corresponding to the file extension for model files specified in the *preferences*. When double-clicked, models are opened in the code-editor (if specified), or else in Piraña's built-in NM-TRAN editor. Models can be filtered using the 'Filter' above the main overview table. This filters all models based on the columns: run number, description, and notes.

New models can be created from scratch, from a template, or by duplicating an existing model. Some basic template models are included, but it is possible to build your own library with base models that you often use. Templates can be added by copying a model file to `/templates` in the Piraña directory. The template models should have the same file extension as your model files to be recognized as a template.

Duplication of models can be performed easily by selecting the parent model and clicking 'duplicate' from the context-menu (right click on the model). Final parameter estimates from the reference model can be updated in the new model, and also model-file numbers in $TABLE and $EST records. Some basic syntax rules should be adhered to, see Conventions and Methods at the end of this chapter.



New model number (without mod): 001A
Reference model: 001
Model description: Two compartment Model, Using ADVAN3, TRANS

☐ Change model name in $PROB / $TABLE / $EST sections?
☑ Use final parameter estimates from reference model?
☐ Fix estimates?

Automatically changing of table files only works if in the table name the exact filename of the model is incorporated. E.g. if your model file is name 005.mod, then your tables should be named sdtab005, tab005.tab, 005.tab, etc in the control stream

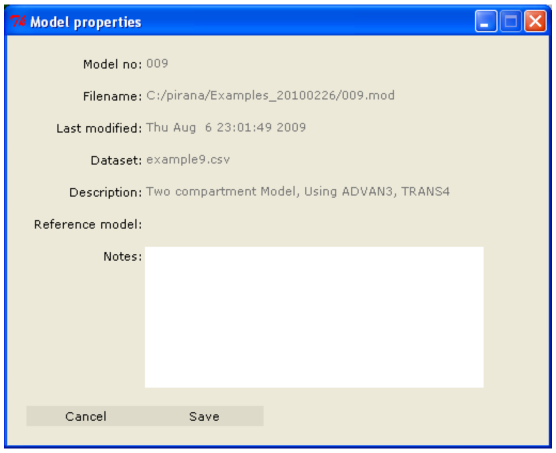
Figure 2.2: Duplicate model

Notes and colors

Piraña offers the option to store notes about models in a database. For this, in each active folder that is visited with Piraña, a small SQLite database is created ('pirana.dir') which is able to record model information. To add notes to a model, select the model and click the model properties from the menu, or the '(i)' icon from the dashboard on the bottom. Models and results can also be given a color, which is done by selecting the model and clicking on the desired color from the dashboard. The meaning of the color-coding is of course all up to you!

Projects

Piraña can handle multiple projects, which are shown in the blue bar above the active folder entry, and enables instant switching between them. To add a project to the list, browse to a folder by clicking on the folder-icon next to the location bar, or by clicking through the directory-listing in the model overview. Next, click the 'save'-icon next to the 'Project'-listbox, give your project a unique name and press the save button. Your project is now available from the listbox. To delete a project from the list, click the trash icon next to the project list. The green refresh-icon refreshes the view of the current directory.


Model properties

Model no: 009

Filename: C:/pirana/Examples_20100226/009.mod

Last modified: Thu Aug 6 23:01:49 2009

Dataset: example9.csv

Description: Two compartment Model, Using ADVAN3, TRANS4

Reference model:

Notes:

Figure 2.3: Model properties

Data files

The right table shows the data-files, R scripts, Xpose files, and other files, which are selected by clicking the desired tab button above the box. The file extension of 'tab' and 'csv' file-types can be set in the preferences while the XPose filenames should adhere to the conventions specified in the XPose manual (sdtab, patab etc). Right-clicking on a selected model shows optional actions on the file.

Note: When the XPose option is chosen, only unique run numbers are shown, instead of all tabualar data files. When, after an XPose dataset is chosen, the 'X'-button is clicked, R will open with the selected dataset in XPose.

Buttons: Models, runs, folders

Several buttons are available from the Piraña main screen for changing the model overview, or perform actions on models (left set of buttons). In the middle set of buttons, functionality is included for e.g. viewing a log of run executions, intermediate run results. On the right, two checkboxes are implemented controlling the visibility of folders created by Piraña and PsN. The 'nmfe_' folders are filtered out from the Piraña model list by default, but these can be made visible by checking the 'nmfe' checkbox, located above the model list. The same is done for 'modelfit_' and 'npcdir_' directories created by PsN.

Buttons: Quick links

Next to the active directory entry, a bar with quick links to useful programs is shown. The location of these links point to can be changed in 'File → Software'.

Managing NONMEM installations

Existing NONMEM installations can be added to Piraña using 'Manage installations' from the 'NONMEM' menu. Here, both local and cluster (connected to through SSH) installations can be added and made available in Piraña.

It may sometimes be necessary to increase (or decrease) internal variable sizes, after which NONMEM has to be re-compiled. Piraña can take care of this for you, but only for local regular (non-NMQual) installations of NONMEM. In the 'Manage NONMEM installations' dialog, all settings found in the `SIZES` file of the active NONMEM installation can be edited and saved here, and the installation can be re-compiled. The variables for NMQual NONMEM installations can be viewed but not edited.

2.2.1 Running models

General

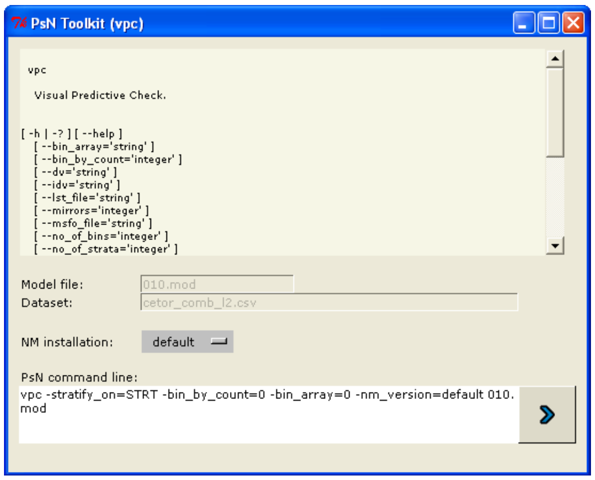
When one or more NONMEM installations have been added to Piraña, starting a model execution is easy: select a model from the list, right-click to show the context-menu, and click 'Run (nmfe)', or press Control-R. This will open up a dialog showing two additional options for running the model, e.g. which NONMEM installation to use, and if models should be executed in separate folders or on a cluster system.

Using PsN

Models can also be run by one of the PsN-toolkit functions (execute: Control-E, vpc: Control-V, bootstrap: Control-B, etc.). The procedure is similar to described above, but now, a different dialog window is opened (shown below), which also shows the PsN-help info for the selected command. The command line editor can be used to specify additional specifications to PsN. Piraña stores the last command for each of the toolkit functions, so you don't need to retype the bulk of your command-line each time you use it.

Using Wings for NONMEM

On Windows, Piraña is capable of invoking the WFN-commands *nmgo* and *nmbs*, for run execution and *nmbs* for bootstrapping respectively. Since WFN does not support



```
 vpc

  Visual Predictive Check.


[ -h | -? ] [ --help ]
   [ --bin_array='string' ]
   [ --bin_by_count='integer' ]
   [ --dv='string' ]
   [ --idv='string' ]
   [ --lst_file='string' ]
   [ --mirrors='integer' ]
   [ --msfo_file='string' ]
   [ --no_of_bins='integer' ]
   [ --no_of_strata='integer' ]
```

Model file: 010.mod
Dataset: cetor_comb_l2.csv

NM installation: default

PsN command line:

```
vpc -stratify_on=STRT -bin_by_count=0 -bin_array=0 -nm_version=default 010.
mod
```

Figure 2.4: PsN dialog window

multiple model files to be processed by its commands, when multiple models are selected, only the first model file is executed. When the WFN method is selected, two parameter specification bars will become visible. In the upper entry, run parameters can be specified, e.g. for the bootstrap: '1 100' to specify a bootstrap with 100 replicates. The lower parameter bar specifies command-line parameters used when starting WFN.bat, e.g. 'g77 std' for specifying the compiler and the NONMEM version to be used. More information about WFN: http://wfn.sourceforge.net/wfninst.htm, http://wfn.sourceforge.net/wfnnmver.htm.

Note: What Piraña actually does when executing runs through WFN, is create a temporary batch-file in the current directory that starts *WFN.bat* to load the necessary environment variables, after which *nmgo* is started with the model-file and parameters specified.

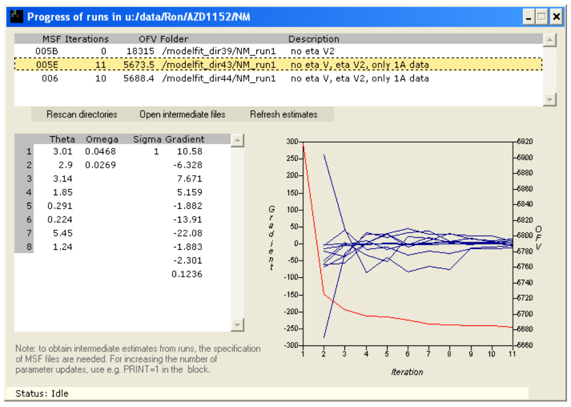
2.2.2 Analyzing output

HTML and LaTeX run reports

Run reports can automatically be generated from NONMEM output of completed runs in HTML or LaTeX format. The HTML output optionally displays basic run information, run statistics, description and notes, and parameter estimations for all estimation methods performed. The information to be included in the report can be specified in the menu under 'Results → Include in run reports'. LaTeXoutput is opened in the specified code editor, but also can be converted automatically to PDF, by setting the 'pdflatex' setting to 1 in the preferences.

Intermediate results

When running models through any of the available methods, the intermediate results (parameter estimates, objective function value (OFV), gradients) can be viewed by clicking on the "gauge"-icon (lower left button from the results buttons). This will open the window shown in the figure below, which shows the models that are currently running. By Clicking on a run intermediate data from INTER and OUTPUT files will be gather, and returned in the table and the plot. Note that some Fortran compilers (e.g. g77) do not support the real-time updating of these output files, and that for such NONMEM installations, this function will not work properly.



MSF	Iterations	OFV	Folder	Description
005B	0	18315	/modelfit_dir39/NM_run1	no eta V2
005E	11	5673.5	/modelfit_dir43/NM_run1	no eta V, eta V2, only 1A data
006	10	5688.4	/modelfit_dir44/NM_run1	no eta V, eta V2, only 1A data

	Theta	Omega	Sigma	Gradient
1	3.01	0.0468	1	10.58
2	2.9	0.0269		-6.328
3	3.14			7.671
4	1.85			5.159
5	0.291			-1.882
6	0.224			-13.91
7	5.45			-22.08
8	1.24			-1.883
				-2.301
				0.1236

Note: to obtain intermediate estimates from runs, the specification of MSF files are needed. For increasing the number of parameter updates, use e.g. PRINT=1 in the block.

Status: Idle

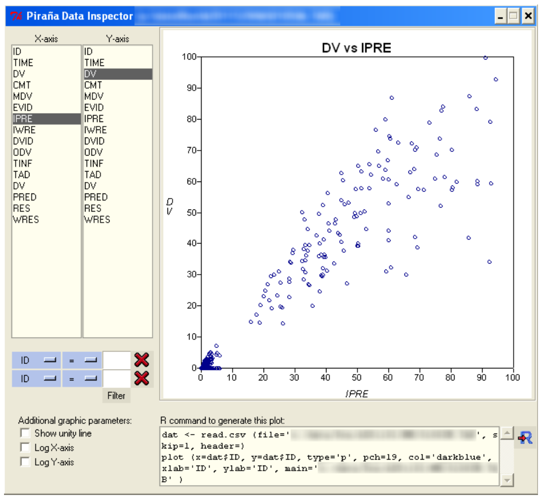
Figure 2.5: Intermediate results

Plot functionality

Piraña is able to construct scatter plots using the built-in 'Data-Inspector', e.g. for quickly inspecting goodness-of-fit, covariate relationships, distribution of etas, performing data checkout etc. The Data Inspector shows all the columns present in the dataset, which can be plotted on the X- or Y-axis (shown in figure 3.2). Piraña can handle NONMEM-generated tables (either created with the ONEHEADER and NOHEADER options), and CSV-files. Multiple Y values can be plotted by holding the Control- or shift-key and selecting multiple (up to three) data columns.

Inside the plot, regions of interest may be selected, which are then zoomed. Double-clicking inside the plot region changes back to the previous view. Plots can be filtered, which can be useful, e.g. to show only data for one patient, or groups of patients or covariates.

In the text-box below the plot, a command is generated that creates the same graph in R, which can be used as a starting point for the generation of plots for manuscripts



```
dat <- read.csv (file='[redacted]', s
kip=1, header=)
plot (x=dat$ID, y=dat$ID, type='p', pch=19, col='darkblue',
xlab='ID', ylab='ID', main='[redacted]
B' )
```

Figure 2.6: Piraña plot functionality

or reports. On Windows, clicking the 'R'-button will invoke the RGUI and create the plot. On Linux, clicking the icon will open the code in the specified code editor.

2.2.3 Scripts

Piraña includes functionality to run custom R-scripts on models, or output from models-executions such as tables. This feature makes Piraña a very flexible tool to manage and evaluate modeling results, as custom scripts are easily created and adapted to suit specific needs. Scripts can be written by the user, but a considerable collection of scripts is also bundled with Piraña. Of course, these scripts may be used as starting point for your own implementations. The scripts are located in two locations, one for group-wide scripts (in the scripts-folder in the location where Piraña is installed), and one for user-scripts ('home/user/.pirana/scripts' on Linux, 'C:\Documents and Settings\user\Application data\Pirana\Scripts' on Windows). The folder structure underlying the scripts folder is maintained within the scripts menu, and scripts can be edited either by editing them from outside Pirana, or by choosing them from the 'Edit' menu option.

Scripts can be started by selecting a model, and selecting the desired script from the menu. What happens now, is that Piraña invokes R and runs the script in the directory 'pirana_temp' underlying the active folder. However, before execution, Piraña inserts an R object which specifies model and results information, e.g. as:

```
models <- list (
   "003" = list (
     "modelfile"       = "003.mod",
     "description"     = "PK model digoxin",
     "reference_model" = "002",
     "data_file"       = "nm_pk_001.csv",
     "output_file"     = "003.lst",
     "tables"          = c("003.TAB", "sdtab003")
   )
 )
```

To automatically load PDFs or images that are created by R after execution of the script, include e.g. the following line in the script:

```
#PIRANA model_output.pdf
```

where model_output.pdf is the file that you want Piraña to load. This may either be a PDF, or a common image format such as PNG, JPG, or GIF. Please have a look at the scripts included with Piraña for examples of how to get the most out of this functionality.

2.2.4 Other tools

Covariance calculator This opens the built-in 'Covariance Calculator', which can be used to re-calculate a covariance to a correlation on the SD-scale. The formula for correlation that is used is:

$$\rho_{i,j} = \frac{{\omega_{i,j}}^2}{\omega_{i,i} \cdot \omega_{j,j}}$$

with ρ specyfing the correlation between two elements (i,j) in a matrix, and ω specifying elements of Ω or Σ.

Checkout dataset This will create (and open) an HTML-file which displays a selected dataset using separate colors for different event-types. This will thus show the dataset in a slightly more convenient format for manual inspection than in a spreadsheet. The function needs an EVID column in the dataset to work properly.

Batch operations

Replace block This function enables you to replace a whole block of code in selected model files, e.g. the $DATA block if you want all model files to use a different data file, or the $THETA block if you want to use other initial estimates.

Add code With this function, lines of code can be added to the beginning or the end of selected models.

Batch duplicate Creates multiple duplicates of one model file, with (optionally) updated run/table numbers and final parameter estimates.

Random simulation seeds In all selected models, the $SIMULATION block will be updated with new seeds.

2.3 Conventions and Methods

Control stream conventions

- Piraña reads the description of the model from the first line. The words '`$PROBLEM`' or '`Model desc:`' are recognized.
- If you want to use the hierarchy functionality for models, you should specify the reference model in the first few lines of the model file. Piraña is flexible and compatible with Census, and understands e.g. the following syntaxes:

```
; Ref=001.mod
; Ref. model:001.mod
; Ref:001.mod
; Parent=001.mod
; Parent model:001.mod
; etc...
```

- Piraña reads the decriptions for model parameters from the model file (and not from the output file). They need to be specified after a semi-colon, e.g.

```
$THETA
(3, 5, 11) ; CL/F
(10, 50, 100) ; V/F
```

  To correctly read a covariance block, the covariance term needs to be specified as 'COV', e.g.

```
$OMEGA BLOCK(2)
0.1 ; IIV CL/F
0.05 ; COV CL~V
0.1 ; IIV V/F
```

- When models are to be executed in a separate directory, files needed for compilation (e.g. additional Fortran routines in .FOR files), are copied automatically by Piraña. These files should be specified in the `OTHER` and `CONTR` entries on the `$DATA` record. If more additional files are needed, you can instruct Piraña to copy this file by adding this line to your control stream:

  ```
  ; INCLUDE=file1_to_be_copied.ext,file2_to_be_copied.ext,etc
  ```

  Note that PsN has it's own functionality for doing this.

- For duplicating runs with updated initial estimates from the previous run, it is required to declare $OMEGA and $SIGMA blocks only once, and not on every new line in the block, except when specifying BLOCK structures. E.g.:

  ```
  $OMEGA 0.1 ; KA
  $OMEGA BLOCK(2)
  0.1  ; IIV CL/F
  0.05 ; COV CL~V
  0.1  ; IIV V/F
  $OMEGA
  0.1 ; EMAX
  0.1 ; V2
  ```

2.4 Troubleshooting

Reported errors encountered with previous releases have been fixed. Please report new bugs on the website (http://code.google.com/p/pirana/issues/list). A FAQ is also available on the website.

Chapter 3

Piraña and clusters

3.1 Clusters

Piraña facilitates interaction with Linux-based clusters on which NONMEM and/or PsN are installed, and both MOSIX clusters and the Sun Grid Engine are supported. Basically, two options are available for working with Piraña on such systems. The first, and recommended option is to install Pirana only on the cluster-server, and start Pirana from the modeler's PC using SSH X-window tunneling. This has the advantage of requiring only one central installation of Pirana for the entire modeling group, and Piraña and other modeling software is installed in a controllable environment. This option is also available for users that want to work on Windows, by using XMing or Cygwin/X on the local Windows PC. The other method is to install Piraña on each modeler's PC and to connect using Pirana's 'SSH-connect' mode. This may require a few additional client- and server-side installations. For using this, you need to mount the cluster drive with your data on your local PC (e.g. using sshfs on Linux or ExpanDrive on Windows). Also you need to have ssh installed (OpenSSH on Windows), of which some details are specified below.

Additionally, Piraña supports the construction of a simple Windows-based clustering system (PCluster), which will be described in a subsequent section. PCluster makes it easy to set up a cluster using e.g. PCs of colleagues, and is easy to install on Windows systems. It must be noted however, that this kind of setup is slightly inferior to the clusters mentioned above in terms of stability and performance. Only if no funds or no IT know-how is available to a modelling group do we recommend using the PCluster.

3.1.1 MOSIX and SGE

While MOSIX systems distribute all active programs automatically, jobs for the SGE must be submitted to the queue controller (using 'qsub'), which then distributes the job to a node. If you are using an SGE cluster, you should therefore switch on the 'Run using SGE' option in the 'NONMEM' menu section, or set it in the 'Preferences section'. Basically, what Piraña does is submit all models that are run using the 'nmfe'-method to the queue controller, instead of executing the job directly. If you want to use SGE with the PsN-toolkit, use '-run_on_sge'. Please consult the PsN-documentation or its developers for more information.

3.1.2 Method 1: X-window tunneling

Xming

If Piraña is installed on a Linux-cluster, and accessed from a Windows client, you need to have an X-window system installed, which will enable the display of the Pirana window locally on your PC. Such a system is Xming, which can be obtained for free from http://sourceforge.net/projects/xming/. After installation of Xming, start the Xming X-window server.

Using the cluster

If everything is set up correctly, and the X-window server is started, Piraña on the cluster can be accessed through SSH, e.g. by using the SSH client. Now, you should be able to see the Piraña main window. If you get an error saying that the display cannot be started on localhost, you may have to enable X-window forwarding in OpenSSH or in PuTTY.

3.1.3 Method 2: SSH-mode

With this approach, Piraña is installed on the local PC. OpenSSH or PuTTY needs to be installed as well, and a drive-letter must be available on your local computer that enables access to your home directory on the cluster. This can be setup e.g. using ExpanDrive to connect to the cluster through SFTP. Alternatively, if, on the remote cluster a Samba server is installed, a connection can be established by giving the following command:

```
NET USE Z: \ \server_name\user_name /user:user_name /persistent:yes
```

Both on Windows and Linux, you need to specify in the preferences the mounted remote diskspace and the local location (ssh_mount_cluster, ssh_mount_local), which is used by Piraña to translate local paths to paths on the remote cluster. If everything is set up correctly, the 'SSH-mode' should be enabled when running a model. Models can then be run in a similar fashion as explained in the section for running models locally.

OpenSSH

In this mode, Piraña needs passwordless SSH-access to the cluster. In contrast to Linux, SSH is not available on Windows by default, but this can be achieved by installing a Windows version of OpenSSH (download from http://sshwindows.sourceforge.net/) or PuTTY (http://www.chiark.greenend.org.uk/s̃gtatham/putty/). After installation, ensure that you have an encryption key pair installed so that the cluster won't ask you for a password everytime you perform an action on the cluster, which is explained below.

Installing Public and private authentication keys

Either on Windows or Linux, type in a shell/console window:

```
ssh-keygen -t rsa
```

When asked for a passphrase, just press <Enter>. Now a public and a private key have been created in c:\Documents and Settings\User Name\.ssh (Windows) or /home/username/.ssh (Linux). In you home directory on the cluster, if it doesn't exist already, create the folder '.ssh'. In this folder, create the file 'authorized_keys' (no extension) and add the contents of id_rsa.pub to that file and save it. Now you should be able to login without being asked for a password (type 'ssh user_namecluster_name' in a console window, and you shouldn't be asked for a password anymore. If SSH asks you if you want to accept the cluster as valid host, accept). Keep your private key secret. In the Piraña preferences, speficy the username to connect to the cluster (ssh_login).

Tip: Tip: if you experience delays (about 5 secs) when logging in to the server by SSH, this may be caused by a reverse DNS lookup. You can circumvent this by adding 'useDNS no' to the file /etc/ssh/sshd_config on the server. Restart the ssh server for the changes to take effect: sudo /etc/init.d/ssh restart

3.2 Sun Grid Engine

Sun provides an open-source job scheduling agent for Linux, which enables the distribution of NONMEM runs, or PsN-toolkit jobs over computer clusters. Piraña supports the use of SGE, and also provides some functionality for managing the cluster.

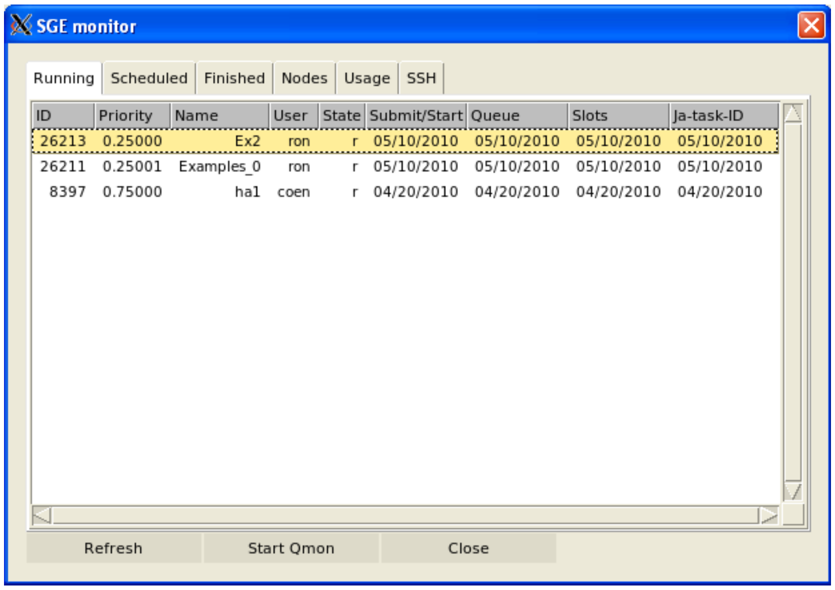
3.2.1 Installation

The reader is referred to the SGE manual for further information on setting up the SGE. If the SGE is up and running, no additional steps have to be taken to be able to use it. Some SGE settings can be specified in the Piraña preferences, but these are likely to be correct by default.

3.2.2 Working with the SGE

Submitting the execution of a model using nmfe to the SGE, can be done by selecting the 'Run on SGE' from the 'Run model' dialog window. This submits the model instead of starting it locally. If you want to use a PsN-toolkit command on the SGE, specify -run_on_sge on the command line.

Clicking the 'cluster'-icon (right-most run-icon in the pirana main screen), the SGE-monitor window is opened and will display an overview of currently running jobs. From the tabs, also scheduled jobs, and recently finished jobs can be viewed, and the nodes can be inspected.



ID	Priority	Name	User	State	Submit/Start	Queue	Slots	Ja-task-ID
26213	0.25000	Ex2	ron	r	05/10/2010	05/10/2010	05/10/2010	05/10/2010
26211	0.25001	Examples_0	ron	r	05/10/2010	05/10/2010	05/10/2010	05/10/2010
8397	0.75000	ha1	coen	r	04/20/2010	04/20/2010	04/20/2010	04/20/2010

Figure 3.1: SGE monitor

3.3 PCluster

The PCluster system is aimed to be a system for distributed NONMEM computing for smaller modeling groups, e.g. in hospital or academic settings. If budget poses no problems, other cluster solutions are probably more adequate. Our major development goal was to develop a system that could be installed on an existing (Windows-based) network, and that would use spare CPU cycles of non-dedicated PCs in the network.

3.3.1 Description and requirements

The infrastructure uses PC's in a standard Windows network environment, which can e.g. be PC's dedicated to running NONMEM, or PC's of co-workers, or a combination of both. The PCluster can be set up without the need for vast hardware/software knowledge. Distribution to multi-core CPU's is possible, and tested up to a total of 40 CPUs.

The infrastructure requires a shared network-drive accessible by all clients (standard desktop PC's in a network environment). On this shared network drive, a version of Perl and Fortran must be present. Furthermore, the PCluster *daemon* Perl-script needs to run in the background on each node. On the client computer, i.e. the modeler's computer, Piraña needs to be installed. Below, more information is provided on how to set this up.

3.3.2 Installation

First, it is necessary to reserve up some harddisk space on your network, to be accessible (read and write) by all clients, mounted as a drive on all nodes (let's assume this to be U:). Copy the contents of the folder `install_on_each_node` in the `/cluster` folder in the Piraña home folder to this drive. Next, the daemon script (or the compiled version of it) should be installed and running on each client in the network. For this, copy the folder \pcluster\install_on_each_node located in the Pirana home folder in the harddisk root (C:\). The daemon can be started manually by executing the Perl script daemon.pl or the daemon.exe. Before starting, check the settings in C:\pcluster\pcluster.ini. This file contains the info for the client, e.g. the number of CPUs to use, and how to connect to the shared drive. Instead of starting the daemon manually, it is recommend to run the daemon as Windows service, which has the advantage that users are able to log in and out without compromising the integrity of the PCluster, and that the service is automatically started when the PC is started. To install this service, run the Perl script inst_daemon.pl:

```
perl inst_daemon.pl
```

This will create the information in the Windows registry for the PDaemon service. If no error is reported, the service can now be started from the command line with:

```
NET START PDaemon
```

It has been reported that the daemon sometimes isn't working properly (i.e. not requesting PsN/nfme runs), while the script ís working properly when it is run manually. Running the daemon manually is however a problem, as a console window will appear each time a run is processing, which is annoying to the principal user of the machine. A workaround for this is to download a tool called CMDOW (available from http://www.commandline.co.uk/cmdow/). If you start the following command (e.g. in a batch script at startup), no consoles will appear:

```
u:\cmdow.exe /run /hid u:\perl\bin\perl.exe c:\pcluster\pdaemon.pl
```

Note: Thanks to Andreas Lindauer for suggesting this workaround.

Perl and PsN

Make sure Perl (with PsN installed) is available on the cluster-drive (e.g. in U:\Perl). Also check that you have the following Perl modules installed: LockFile, Sys::Hostname and File. Since this option hasn't been fully developed yet in PsN, it is necessary to make a few changes in some PsN source files (using PsN 3.1.0 as a template).

- In nonmem.pm in the PsN directory, add the following lines at the top of the source-file:

  ```
  my $fortran_dir = 'U:\MinGW\bin';
  unless ($ENV{'PATH'} =~ m/$fortran_dir/) {
    $ENV{'PATH'} = $fortran_dir.";".$ENV{'PATH'};
  }
  ```

  This ensures the Fortran compiler is in the path (change U:\MinGW\bin' to the fortran path that you use). Of course this is not necessary if this directory is already permanently in the PATH environment variable.

- In tool\modelfit.pm change line 3014:

  ```
    require File::Temp; # qw/tempfile tempdir/;
    require Sys::Hostname;
    require LockFile::Simple; # qw/lock unlock trylock/; #Non-standard module
  ```

  into

  ```
    use File::Temp qw/tempfile tempdir/;
    use Sys::Hostname;
    use LockFile::Simple qw/lock unlock trylock/;  # Non-standard module
    use File::Basename;
  ```

  In the same file, move the contents of line 3063:

```
  return $jobId;
```

to line 3057, just after the line with 'unlock $jobId;'.

In the same file, around line 3105, change:

```
  if( -e "$ZinkDoneDir$jobId" ){
```

into

```
  use File::Basename;
  my $job = basename($jobId);
  if( -e "$ZinkDoneDir/$job" ){
```

- In psn.conf file in the PsN folder on the cluster, edit the following lines with your settings (also remove leading semi-colons):

  ```
  40 : perl = U:\Perl\bin\perl.exe
  82 : zink_dir = U:
  105: default=U:\nmvi,6
  ```

  Of course, the other PsN settings should be set correctly as well.

Fortran (g77)

If you want to use the PsN-toolkit on the PCluster, you must provide access to a Fortran compiler for each client in the cluster. This is because PsN runs are compiled after distribution to a client. If you only make use of Pirana's own distribution capabilities, only a Fortran compiler on the modeler's own computer is required. The recommended way is to use a central installation of Fortran on the cluster drive, e.g. g77 supplied with MinGW. For this, e.g. install MinGW with g77 on your local computer. After installation, copy the folder MinGW to the cluster drive.

Pirana preferences

In Piraña the following preferences should be updated:

zink_host: e.g. ZinkHost, A host name for the Zink / PCluster, should correspond with the hostname specified in psn.conf.

And in 'software settings', change:

f77_dir: Path to Fortran on the cluster-drive e.g. U:/MinGW/bin

perl_dir: Path to Perl on the cluster-drive e.g. U:/Perl

psn_dir: Path to PsN on the cluster-drive e.g. U:/Perl/site/lib/PsN_2_3_1

Troubleshooting

The PCluster, although it functions quite adequately in the current setup, is actually still in beta phase. If you have questions regarding the set-up or functionality of the PCluster, please send a mail to the mailinglist (pirana-users@lists.sourceforge.net) or contact the Piraña development team directly.

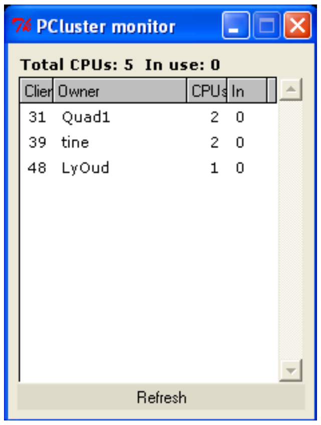
3.3.3 Working with the PCluster

Using the PsN toolkit on PCluster

This functionality is enabled by specifying '-run_on_zink' on the PsN command line from the Piraña window, after selecting a PsN tool from the toolkit. You should leave the PsN console window open until the run is finished, since the instance of PsN that is running locally will take care of the run distribution and calculation and formatting of data files after finishing the cluster run(s).

Cluster monitor

Using the cluster monitor ('View → Cluster monitor') the activity and availability of clients can be monitored. Note that status of the client might take a half a minute to refresh with the up-to-date information, due to the interval used by the daemon-script.



Total CPUs: 5 In use: 0

Clier	Owner	CPUs	In
31	Quad1	2	0
39	tine	2	0
48	LyOud	1	0

Figure 3.2: PCluster monitor

Chapter 4

Endnotes

4.1 Acknowledgements

For testing and feedback, thanks to (ex-)colleagues in the modeling & simulation group of the Slotervaart Hospital / the Netherlands Cancer Institute. Also various modelers from around the world are recognized for their valuable suggestions and bug-reports. The Uppsala PM group for good ideas and providing their Mosix-cluster for testing.

4.2 Disclaimer

This software is provided as free software under an open source license (GNU General Public License) and on an 'as is' basis, without warranty of any kind. Your use of the software is at your sole risk, and the author cannot be held liable for any harm, inflicted upon your system by this program.

Questions, comments, tips or bug-reports are very welcome at the the bug report list (http://code.google.com/p/pirana/issues/list), or at the mailinglist (http://pirana.sourceforge.net/mailinglist. Please be as specific as you can about the issue. Also, if you would like to contribute to the development of Piraña, please contact us.

4.3 Aims for future versions

Currently we are working on a quality assurance system for Piraña, which is already partially in place. Documentation will be posted online when available.

Aims for version 2.4.0: January 2011?

- Support for parallel NONMEM
- Enhance reporting features

4.4 Version history

Updates for version 2.3.0 ("Oahu"): April 2010

- NONMEM 7 output support
- Added scripts functionality
- Added SGE support
- Improved Linux support
- Included NM help, improved PsN help
- Updated manual and website

Updates in version 2.2.0 ("Pipeline"): November 2009

- Linux / Mac OS X compatible
- Multi-user functionality
- Basic NONMEM 7 support
- Much improvements to GUI
- bug-fixes

Updates in version 2.1.0 ("Waimea Bay"): August 2009

- Full screen mode
- Plots of gradients and OFV in intermediate results window
- Keep log of model executions
- More functionality for documenting models, tables and projects.
- More structure and documentation in source code
- Changes to GUI and Bugfixes

Updates in version 2.0.1 ("Hookipa"): May 2009

- Added δ OVF in model overview
- Resizable column headers in model overview
- Added minimization text to HTML output
- Overlay multiple Y-columns in Data Inspector plot (hold ctrl-key)
- Improved NONMEM run options in GUI
- Improved adding of NONMEM installations, NMQual/regular installations detected automatically
- Automatically detect PsN NONMEM installations
- Added covariance calculator
- Added dataset checkout function
- Bug-fixes

Updates in version 2.0.0 ("Waikiki"): March 2009

- Changed GUI: models and results in one window. More notebook-like, add notes and highlighting.
- Access linux-based (Mosix) clusters through OpenSSH. Use nmfe and PsN-toolkit on these clusters.
- Added support for running PsN toolkit on PCluster
- Overview of active runs in (sub)folders, show intermediate results and gradients.
- Removed reliance on PsN for creation of summary of run result.
- Quick view of parameter estimates window
- Create csv summary of all models and results
- Added filters to Data Inspector
- Added Installer
- Lots of bug-fixes

Updates in version 1.3: September 2008 (not released)

- Replaced R-plots with general plotting functionality (Data Inspector)

Updates in version 1.2.1: August 2008

- Added more WFN support: custom compiler / NM version

Updates in version 1.2: June 2008

- Fixed the script for bootstrapping on the PCluster
- Added feature: Run models using NMQual NONMEM installations
- Added feature: Install NONMEM from NMQual XML files
- Added feature: Support for Wings for NONMEM (nmgo/nmbs)
- Added new functions for monitoring cluster runs

Release version 1.0: May 2008

- First release version