****

PFIM 3.2

Caroline Bazzoli, Thu Thuy Nguyen, Anne Dubois, Sylvie Retout, Emanuelle Comets, France Mentré

INSERM, UMR738, Paris, France ; Université Paris 7, Paris, France

Janvier 2010

www.pfim.biostat.fr

User guide*

This document is an adds-on to PFIM 3.0 documentation. Thus, it only outlines the new features implemented in PFIM 3.2 and explains how to carry out to use them. This documentation version does not detail the features that were previously released in version 3.0 of PFIM.

****

PFIM 3.2 is free library of functions.
The Université Paris Diderot and INSERM are the co-owners of this library of functions.

Disclaimer

We inform users that the PFIM 3.2 is a tool developed by the Laboratory « Models and methods of the therapeutic evaluation of the chronic diseases »- UMR-S 738, under R and GCC.

PFIM 3.2 is a library of functions. The functions are published after a scientific validation.

However, it may be that only extracts are published.

By using this library of functions, the user accepts all the conditions of use set forth hereinafter.

Licence

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

* The document PFIM3.2_Examples.pdf with fully described examples is also available.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE UNIVERSITE PARIS DIDEROT OR INSERM OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Redistribution and use in source and binary forms, with or without modification, are permitted under the terms of the GNU General Public Licence and provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by Université Paris Diderot and INSERM (http://www.biostat.fr)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names "PFIM" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact france.mentre@inserm.fr.

5. Products derived from this software may not be called "PFIM", nor may "PFIM" appear in their name, without prior written permission of the Université Paris Diderot and INSERM.

www.pfim.biostat.fr

CONTENT

1. Description of the new features in PFIM 3.2

PFIM 3.2 is a new release of the R script function PFIM 3.0 **[1]** dedicated to design evaluation and optimisation for multiple response models.

This version incorporates new features in terms of model specification and development of the expression of the Fisher information matrix (M_F).

Regarding model specification, the library of standard pharmacokinetic (PK) models has been completed with three compartment models with linear elimination and models with Mickaelis–Menten elimination (one, two and three compartment models). Furthermore, a library of pharmacodynamic (PD) models is now available.
Concerning the expression of the Fisher information matrix, PFIM 3.2 can handle either a block diagonal Fisher information matrix or the complete one.
The computation of the Fisher information matrix to perform evaluation and optimisation of individual designs (i.e. models with no random effects) can be carried out.
It is now also possible in PFIM 3.2 to use models including inter-occasion variability (IOV) with replicated designs at each occasion [2].
Last, a new feature of PFIM 3.2 is the computation of the Fisher information matrix for models including fixed effects for the influence of discrete covariates on the parameters. Specification of covariates can depend or not of the occasion. The computation of the predicted power of the Wald test for comparison or equivalence test for a given distribution of a discrete covariate as well as the number of subjects needed to achieve a given power can be computed [2, 3, 4].

The same input file, named by default *stdin.r*, used in PFIM 3.0 can be used in PFIM 3.2 but the new features would then not be active.

PFIM 3.2 is also developed for R 2.4.1 and higher versions.

1.1. Model specification

Models can be specified either with their analytical form or with systems of differential equations, using the libraries of models or the user defined model option. In the later case, users can define their own model analytically or use a system of differential equations. This option has not been modified in PFIM 3.2, only the libraries of models have been completed.
Compared to PFIM 3.0, three compartment models with linear elimination and models with Mickaelis–Menten elimination (one, two and three compartment models) have been added to the library of PK models. Moreover, a library of PD models is now available, supporting immediate response models (alone or linked to a pharmacokinetic model) and the turnover response models (linked to pharmacokinetic model). These libraries have been derived from the PKPD library developed by Bertrand and Mentré [5] for the MONOLIX software, and all analytical expressions are in that document. Presently, there is no model with lag time in both libraries.

As in PFIM 3.0, to use the library of models, the user has to specify the path of the file in the modelfile named by default *model.r*.

1.1.1. Library of pharmacokinetic models

Two types of PK models can now be used in PFIM 3.2, PK models with a first order linear elimination or PK models with a Mickaelis–Menten elimination.

The PK models with a linear elimination are written using an analytical form whereas the PK models with a Mickaelis-Menten are written using a differential equation system.
These both types of PK model are written in the file *LibraryPK.r* available in the folder Program. Thus, the user has to specify the path of this file in the model file to use this library of models:

source (paste(directory.program,dirsep,"LibraryPK.r", sep=""))

For each type of PK models, the list of models are presented in separated tables in the following sections.

These tables return all the information in order to use the model function chosen. The model is described by:

- a **name**
- the type of **input**
- the type of **elimination**
- the **number of compartments**
- the parameters used (**parameterisation**)
- the type of **administration** (**sd** : single dose, **md**: multiple dose, **ss**: steady state)
- for each administration type, some variables are required (or not). They are specified in the column named: **Needed variables** (**N**: number of doses, **tau**: interval between two doses, **TInf**: duration of the infusion, **doseMM**: dose)

For models with infusion, the user has to specify the duration of infusion (**TInf**) in the needed variable. The rate of infusion is computed automatically in the function model by the expression: dose/TInf. As in PFIM 3.0, for PK models with linear elimination, the variable **dose** has to be specified in the input file.

For example, if one uses after a multiple dose administration, the first order oral absorption with one compartment model (**oral1_1cpt_kaVCl with option md**) from the library, the function of the model uses three parameters (**ka**, **Cl** and **V**) and two needed variables (**N**, **tau**): the number of doses (**N**) and the interval between two doses (**tau**).

Examples of the use of the library of pharmacokinetic models are presented in section 2.2 of the present document as in section 2.1 of the PFIM 3.0 documentation.

1.1.1.1. Pharmacokinetic models with a linear elimination

Compared to PFIM 3.0, the library of PK models with linear elimination has been completed by the three compartment models for the three types of input (bolus, infusion and first order oral absorption) and the three types of administration (single dose, multiple dose, steady state).
The list of these PK models is given in Table 1. It is an update of the Table 1 presented in the documentation of PFIM 3.0.

Table 1. Pharmacokinetic models with first order linear elimination included in the library of models

Name	Input	Cpt	Elimination	Parameterisation	Administration	Needed Variable(s)
bolus_1cpt_Vk	IV-bolus	1	1st order	V, k	sd md ss	– N, tau tau
bolus_1cpt_VCl	IV-bolus	1	1st order	V, Cl	sd md ss	– N, tau tau
infusion_1cpt_Vk	IV-infusion	1	1st order	V, k	sd md ss	TInf TInf, N, tau TInf, tau
infusion_1cpt_VCl	IV-infusion	1	1st order	V, Cl	sd md ss	TInf TInf, N, tau TInf, tau
oral1_1cpt_kaVk	1st order	1	1st order	ka, V, k	sd md ss	– N, tau tau
oral1_1cpt_kaVCl	1st order	1	1st order	ka, V, Cl	sd md ss	– N, tau tau
bolus_2cpt_Vkk12k21	IV-bolus	2	1st order	V, k, k12, k21	sd md ss	– N, tau tau
bolus_2cpt_ClV1QV2	IV-bolus	2	1st order	Cl, V1, Q, V2	sd md ss	– N, tau tau
infusion_2cpt_Vkk12k21	IV-infusion	2	1st order	V, k, k12, k21	sd md ss	TInf TInf, N, tau TInf, tau

infusion_2cpt_ClV1QV2	IV-infusion	2	1st order	Cl, V1, Q, V2	sd md ss	TInf TInf, N, tau TInf, tau
oral1_2cpt_kaVkk12k21	1st order	2	1st order	ka, V, k, k12, k21	sd md ss	– N, tau tau
oral1_2cpt_kaClV1QV2	1st order	2	1st order	ka, Cl, V1, Q, V2	sd md ss	– N, tau tau
bolus_3cpt_Vkk12k21k13k31	IV-bolus	3	1st order	V, k, k12, k21, k13, k31	sd md ss	– N, tau tau
bolus_3cpt_ClV1Q1V2Q2V3	IV-bolus	3	1st order	Cl, V1, Q1, V2, Q2, V3	sd md ss	– N, tau tau
infusion_3cpt_Vkk12k21k13k31	IV-infusion	3	1st order	V, k, k12, k21, k13, k31	sd md ss	TInf TInf, N, tau TInf, tau
infusion_3cpt_ClV1Q1V2Q2V3	IV-infusion	3	1st order	Cl, V1, Q1, V2, Q2, V3	sd md ss	TInf TInf, N, tau TInf, tau
oral1_3cpt_kaVkk12k21k13k31	1st order	3	1st order	ka, V, k, k12, k21, k13, k31	sd md ss	– N, tau tau
oral1_3cpt_kaClV1Q1V2Q2V3	1st order	3	1st order	ka, Cl, V1, Q1, V2, Q2, V3	sd md ss	– N, tau tau

1.1.1.2. wPharmacokinetic models with a Mickaelis-Menten elimination

One, two and three compartment models are implemented for the three types of input. For bolus input and Mickaelis-Menten elimination, only single dose models are implemented. For infusion and first order absorption input, single dose and multiple dose are implemented. There is no steady-stae conditions for PK models with Mickaelis-Menten elimination. The list of these PK models is given in Table 2.

 For models with a bolus input, the dose has to be specified in the input file (*stdin.r* by default) as the initial condition of the differential equation system. For models with infusion or first order absorption input, dose has to be specified as an argument and NOT IN THE INITIAL CONDITION OF THE MODEL IN THE INPUT FILE.

 As the dose is an argument, it is not possible to specify different doses per group for models with infusion or first order absorption input. All groups of the design considered have the same dose. Otherwise, the user should use the user defined model option.

Table 2. Pharmacokinetic models with Mickaelis-Menten elimination included in the library of models

Name	Input	Cpt	Elimination	Parameterisation	Administration	Needed Variable(s)
bolus_1cpt_VVmkm	IV-bolus	1	Mickaelis-Menten	V, Vm, km	sd	-
infusion_1cpt_VVmkm	IV-infusion	1	Mickaelis-Menten	V, Vm, km	sd md	doseMM,TInf doseMM,TInf, tau
oral1_1cpt_kaVVmkm	1st order	1	Mickaelis-Menten	ka, V,Vm, km	sd md	doseMM doseMM,tau
bolus_2cpt_Vk12k21Vmkm	IV-bolus	2	Mickaelis-Menten	V, k12, k21, Vm, km	sd	-
bolus_2cpt_V1QV2Vmkm	IV-bolus	2	Mickaelis-Menten	V1, Q, V2, Vm, km	sd	-
infusion_2cpt_Vk12k21Vmkm	IV-infusion	2	Mickaelis-Menten	V, k12, k21, Vm, km	sd md	doseMM,TInf doseMM,TInf, tau
infusion_2cpt_ V1QV2Vmkm	IV-infusion	2	Mickaelis-Menten	V1, Q, V2, Vm, km	sd md	doseMM,TInf doseMM,TInf, tau
oral1_2cpt_kaVk12k21Vmkm	1st order	2	Mickaelis-Menten	ka, V, k12, k21, Vm, km	sd md	doseMM doseMM, tau
oral1_2cpt_kaV1QV2Vmkm	1st order	2	Mickaelis-Menten	ka, V1, Q, V2, Vm, km	sd md	doseMM doseMM, tau
bolus_3cpt_Vk12k21k31k13Vmkm	IV-bolus	3	Mickaelis-Menten	V, k12, k21, k13, k31, Vm, km	sd	-
bolus_3cpt_ V1Q1V2Q2V3Vmkm	IV-bolus	3	Mickaelis-Menten	V1, Q1, V2, Q2, V3, Vm, km	sd	-
infusion_3cpt_Vk12k21k13k31Vmkm	IV-infusion	3	Mickaelis-Menten	V, k12, k21, k13, k31, Vm, km	sd md	doseMM,TInf doseMM,TInf, tau
infusion_3cpt_V1Q1V2Q2V3Vmkm	IV-infusion	3	Mickaelis-Menten	V1, Q1, V2, Q2, V3, Vm, km	sd md	doseMM,TInf doseMM,TInf, tau
oral1_3cpt_kak12k21k13k31Vmkm	1st order	3	Mickaelis-Menten	ka, k12, k21, k13, k31, Vm, km	sd md	doseMM doseMM,tau
oral1_3cpt_kaV1Q1V2Q2V3Vmkm	1st order	3	Mickaelis-Menten	ka, V1, Q1, V2, Q2, V3, Vm, km	sd md	doseMM doseMM, tau

1.1.2. Library of pharmacodynamic models

The library of PD models supports immediate response models (alone or linked to a pharmacokinetic model) and turnover response models (linked to pharmacokinetic models).
The tables presenting these models return all the information in order to use the model function chosen:

- a **name**
- the parameters used (**parameterisation**)

Examples of the use of the library of pharmacodynamic models are presented in section 2.2.

1.1.2.1. Immediate response pharmacodynamic models alone

Linear, quadratic, logarithmic, Emax, sigmoid Emax, Imax, sigmoid Imax models with null or constant baseline are available. The list of these models is given in Table 3.
These models are written with an analytical form and have to be used in the case of a model with one response (PD evaluation or optimisation). They are implemented in the file *LibraryPD_PDdesign.r*. Thus, the user has to specify the path of this file in the model file to use this library of models:

source (paste(directory.program,dirsep,"LibraryPD_PDdesign.r", sep=""))

For these models, the design variables are the concentrations or the doses instead of the sampling times.
For example, if one uses a linear drug action model with a constant baseline (**immed_lin_const**) from the library, the model uses two parameters (**Alin, S0**).

Table 3. Immediate response pharmacodynamic models included in the PD library for PD alone and for PK/PD model

Drug action models	Baseline			
	Null baseline		Constant baseline	
	Name	Parameterisation	Name	Parameterisation
Linear	**immed_lin_null**	Alin	**immed_lin_const**	Alin, S0
Quadratic	**immed_quad_null**	Alin, Aquad	**immed_quad_const**	Alin, Aquad, S0
Logarithmic	**immed_log_null**	Alog	**immed_log_const**	Alog, S0
Emax	**immed_Emax_null**	Emax, C50	**immed_Emax_const**	Emax, C50, S0
Sigmoid Emax	**immed_gammaEmax_null**	Emax, C50, gamma	**immed_gammaEmax_const**	Emax, C50, gamma, S0
Imax	**immed_Imax_null**	Imax, C50	**immed_Imax_const**	Imax, C50, S0
Sigmoid Imax	**immed_gammaImax_null**	Imax, C50, gamma	**immed_gammaImax_const**	Imax, C50, gamma, S0

1.1.2.2. Pharmacodynamic models linked to pharmacokinetic model

In this section, we deal with a two response model, with one response for the PK and the other one for the PD. We thus optimise sampling times for both responses using a PK/PD model. Using the libraries of models, we have four cases to compose the PK/PD model according to the writing of each response model: either with an analytical form (AF) or a differential equation system (ODE).
Therefore, there are four cases of PK/PD models:

1. PK model with linear elimination (AF) and immediate response PD model (AF)

2. PK model with linear elimination (AF) and turnover response PD model (ODE)

3. PK model with Mickaelis-Menten elimination (ODE) and immediate response PD model (AF)

4. PK model with Mickaelis-Menten elimination and turnover response PD model (ODE)

To use PFIM for design evaluation and optimisation for a PK/PD model, it is necessary to have a PK response and a PD response implemented with a similar form.
In the first case, immediate response pharmacodynamic models are written with an analytical form in the file *LibraryPD_PKPDdesign.r* and thus they can be associated to pharmacokinetic models with first order linear elimination (Table 1) implemented in the file *libraryPK.r*, which are also written with analytical forms. In these PD functions, the expression of the PK model is given as an argument.
In this case, the user has to fill in the *stdin.r* using analytical form options and to specify the paths of the library files in *model.r*:

```
source(paste(directory.program,dirsep,"LibraryPK.r",sep=""))
source(paste(directory.program,dirsep,"LibraryPD_PKPDdesign.r",sep=""))
```

However, for the three other cases, the PK response and the PD response are written either with different forms or both with a differential equation system (Case 4). That is why, the user has to call a specific function in order to create a system of differential equations describing the corresponding PK/PD model. This function named ***Create_formED()*** is implemented in the file ***CreateModel_PKPDdesign.r*** and has to be used in the model file as follows:

```
source (paste(directory.program,dirsep,"CreateModel_PKPDdesign.r", sep=""))
create_formED(fun_pk,fun_pd,dose=NA,tau=NA,TInf=NA)
```

where

- ***fun_pk*** and ***fun_pd***: the names of the PK and PD models, respectively
- ***dose:*** value of the dose only for a PK model with infusion or oral input (by default: NA)
- ***tau***: dosing interval to specify only for multiple dose conditions (by default: NA)
- ***TInf***: time of infusion to specify only for PK model with infusion input (by default: NA)

Using this function, a new file named ***model_created.r*** is created in the directory currently used. This new file contains the complete writing of the differential equation system describing the corresponding PK/PD model

created by the function ***Create_formED()***. This file can be deleted after running PFIM. It will be erased at each new use of the function ***Create_formED()***.

For these cases, the user has thus to fill in the *sdtin.r* using differential equation options.

The list of the immediate response PD models is thus given in Table 3 plus those of Table 4. The list of the turnover response PD models is given in Table 5.
For the second case where a PK model with linear elimination is associated to a turnover PD response model, the PK model is written with a differential equations system. Consequently, only some PK models from the Table 1 are implemented:

- for bolus input, only single dose models;
- for infusion input, single dose and multiple dose
- for first order absorption input, single dose and multiple dose

 For models with a bolus input, the dose has to be specified in the input file (*stdin.r* by default) as the initial condition of the differential equation system. For models with infusion or first order absorption input, dose has to be specified as an argument of the function ***Create_formED()*** and NOT IN THE INITIAL CONDITION OF THE MODEL IN THE INPUT FILE. Consequently, it is not possible to specify different doses per group for models with infusion or first order absorption input. All groups of the design considered have the same dose. Otherwise, the user should use the user defined model option.

Table 4. Immediate response pharmacodynamic models linked to a pharmacokinetic model included in the library*

Drug action models	Baseline/disease models					
	Linear progression		Exponential increase		Exponential decrease	
	Name	Param.	Name	Param.	Name	Param.
Linear	**immed_lin_lin**	Alin, S0, kprog	**immed_lin_exp**	Alin, S0, kprog	**immed_lin_dexp**	Alin, S0, kprog
Quadratic	**immed_quad_lin**	Alin, Aquad, S0, kprog	**immed_quad_exp**	Alin, Aquad, S0, kprog	**immed_quad_dexp**	Alin, Aquad, S0, kprog
Logarithmic	**immed_log_lin**	Alog, S0, kprog	**immed_log_exp**	Alog, S0, kprog	**immed_log_dexp**	Alog, S0, kprog
Emax	**immed_Emax_lin**	Emax, C50, S0, kprog	**immed_Emax_exp**	Emax, C50, S0, kprog	**immed_Emax_dexp**	Emax, C50, S0, kprog
Sigmoid Emax	**immed_gammaEmax_lin**	Emax, C50, gamma, S0, kprog	**immed_gammaEmax_exp**	Emax, C50, gamma, S0, kprog	**immed_gammaEmax_dexp**	Emax, C50, gamma, S0, kprog
Imax	**immed_Imax_lin**	Imax, C50, S0, kprog	**immed_Imax_exp**	Imax, C50, S0, kprog	**immed_Imax_dexp**	Imax, C50, S0, kprog
Sigmoid Imax	**immed_gammaImax_lin**	Imax, C50, gamma, S0, kprog	**immed_gammaImax_exp**	Imax, C50, gamma, S0, kprog	**immed_gammaImax_dexp**	Imax, C50, gamma, S0, kprog

* In addition to those in Table 3.

Table 5. Turnover response pharmacodynamic models linked to a pharmacokinetic model included in the library

Types of response	Models with impact on the			
	Input		Output	
	Name	Parameterisation	Name	Parameterisation
Emax	turn_input_Emax	Rin,kout,Emax,C50	turn_output_Emax	Rin,kout,Emax,C50
Sigmoid Emax	turn_input_gammaEmax	Rin,kout,Emax,C50,gamma	turn_output_gammaEmax	Rin,kout,Emax,C50,gamma
Imax	turn_input_Imax	Rin,kout,Imax,C50	turn_output_Imax	Rin,kout,Imax,C50
Sigmoid Imax	turn_input_gammaImax	Rin,kout,Imax,C50,gamma	turn_output_gammaImax	Rin,kout,Imax,C50,gamma
Full Imax[a]	turn_input_Imaxfull	Rin,kout,C50	turn_output_Imaxfull	Rin,kout,C50
Sigmoid full Imax[a]	turn_input_gammaImaxfull	Rin,kout,C50,gamma	turn_output_gammaImaxfull	Rin,kout,C50,gamma

[a] Full Imax means Imax is fixed equal to 1

1.2. Full expression of the Fisher information matrix

The population Fisher information matrix $M_F(\Psi,\xi)$ for multiple response models, for an individual with an elementary design ξ for the vector of population parameters Ψ, is given as:

$$M_F(\Psi,\xi) \cong \frac{1}{2}\begin{pmatrix} A(E,V) & C(E,V) \\ C^T(E,V) & B(E,V) \end{pmatrix}$$

with *E* and *V* the approximated marginal expectation and the variance of the observations of the individual. The vector of population parameter Ψ is defined by $\Psi' = (\mu', \lambda')$ with μ the p-vector of the fixed effects and λ the vector of the variance terms. M_F is given as a block matrix (more details are given in **[1]**) with:

$$(A(E,V))_{ml} = 2\frac{\partial E^T}{\partial \mu_m} V^{-1} \frac{\partial E}{\partial \mu_l} + tr\left(\frac{\partial V}{\partial \mu_l} V^{-1} \frac{\partial V}{\partial \mu_m} V^{-1}\right) \text{ with } m \text{ and } l = 1,\ldots,p$$

$$(B(E,V))_{ml} = tr\left(\frac{\partial V}{\partial \lambda_m} V^{-1} \frac{\partial V}{\partial \lambda_l} V^{-1}\right) \text{ with } m \text{ and } l = 1,\ldots,\dim(\lambda)$$

$$(C(E,V))_{ml} = tr\left(\frac{\partial V}{\partial \lambda_l} V^{-1} \frac{\partial V}{\partial \mu_m} V^{-1}\right) \text{ with } l = 1,\ldots,\dim(\lambda) \text{ and } m = 1,\ldots,p$$

In the previous versions of PFIM, the dependence of V in μ was neglected so that $\frac{\partial V}{\partial \mu} = 0$. Then, the population Fisher information matrix is approximated a block diagonal matrix that is to say the block C of the matrix was supposed to be 0 (see details in **[1]**). Also, the block A is simplified and expressed as:

$$(A(E,V))_{ml} = 2\frac{\partial E^T}{\partial \mu_m} V^{-1} \frac{\partial E}{\partial \mu_l} \text{ with } m \text{ and } l = 1,\ldots,p$$

In the present version, the user can now choose if a full or block diagonal information matrix is needed.


However, this implementation is not developed yet for models with covariates and / or inter-occasion variability.

1.3. Models with no random effects

PFIM 3.2 can also address the problem of evaluation and optimisation of individual designs by assuming no random effect in the model. This is done by specifying inter-subject variability equal to 0. The user specifies an individual design i.e. only one individual subject with a number of sampling times and their allocation in time.
However, it is also possible to specify a design composed of several groups with different sampling times involving different numbers of subjects. In that case, as there is no random effect model, design evaluation and optimisation with PFIM 3.2 is performed as a naïve pooled data approach. Indeed, the model is fitted to the pooled data from all individuals as thought it comes from one "giant subject".

Design optimisation for models with no random effect can be executed only with the Simplex algorithm and no optimisation of the number of subjects is possible.

1.4. Inter-occasion variability (IOV) specification

The expression of the population Fisher information matrix has been extended for model including additional random effects for inter-occasion variability (or within subject variability).
The individual parameters of an individual *i* at occasion *h* are thus expressed by the following relation, which can be additive as

$$\theta_{ih} = \mu + b_i + \kappa_{ih}$$

or exponential as

$$\theta_{ih} = \mu \exp\left(b_i + \kappa_{ih}\right)$$

where μ is the p-vector of fixed effects, b_i the vector of random effects associated to the individual *i* and κ_{ih} the vector of random effects associated to the individual *i* for occasion *h* (h=1,…,H with H the number of occasions). b_i and κ_{ih} are independent. It is assumed that $b_i \sim N(0,\Omega)$ and $\kappa_{ih} \sim N(0,\Gamma)$ with Ω and Γ defined as diagonal matrices of size *p* x *p*. Each element ω_j of Ω and γ_j of Γ represent the inter-individual variability of the j^{th} component of b_i and the inter-occasion variability of the j^{th} component of κ_{ih}, respectively.

The size of the block C and the block B of the expression of the Fisher information matrix are thus modified to incorporate the elements of Γ.

This new development was performed for any number of occasions H. It is implemented in PFIM 3.2 for the case where the same elementary designs are used at each occasion.
The user can include inter-occasion variability in the model as well as covariates.

1.5. Discrete covariate specification

The present expression of the Fisher information matrix accommodates models with parameters quantifying the influence of discrete covariates. Two or more categories can be included. In PFIM 3.2, it can be assumed either that covariates are additive on parameter if the random effect model is additive, or that covariates are additive on log parameters if the random effect model is exponential.
For instance, the individual parameter θ_i is described as the function of a discrete covariate C_i, which takes K values defining K categories, with additive effect model:

$$\theta_i = \mu + \sum_{k=2}^{K} \beta_k \mathbb{I}_{C_i = k} + b_i$$

where here k=1 is defined as the reference group and $\beta_1 = 0$

For each covariate, the user has to specify β, the vector of covariate effect coefficients and the proportions of subjects associated to the K categories.
However, it can be specified if covariates change or not through the different occasions. In the latter case, additional objects are needed: the vector of sequences of values of each covariate at each occasion and the vector of proportions of the elementary designs corresponding to each sequence of covariate values.
The expected Fisher information matrix is computed for each covariate.
The number of covariates, the number of parameters associated to each covariate as well as the number of categories for each covariate, are not limited. But in this version of PFIM, the distribution of the covariates are supposed independent.

1.6. Computation of power and number of subjects needed to treat

1.6.1. Comparison test

Computation of the expected power

The Wald test can be used to asses the difference of a covariate effect *β*. In PFIM, the Wald test is performed on the *β* of each category for each covariate, a global Wald test on the vector *β* (all effect coefficients) is not implemented.
For one covariate and an effect of one category *β* (K=2), the null hypothesis is H_0: {β=0} while the alternative hypothesis is H_1: {β≠0}. The statistic of the Wald test is defined as, $S_W = \frac{\hat{\beta}}{SE(\hat{\beta})}$ with $\hat{\beta}$ the covariate effect estimates and $SE(\hat{\beta})$ its associated standard error. Under H_1, when $\beta=\beta_1$, we then compute the power of the Wald test defined as:

$$P_{diff} = 1-\phi\left(z_{1-\alpha/2} - \frac{\beta_1}{SE(\beta_1)}\right) + \phi\left(-z_{1-\alpha/2} - \frac{\beta_1}{SE(\beta_1)}\right) \quad (1)$$

where Φ is the cumulative distribution function of the standard normal distribution and $z_{\alpha/2}$ is such that $\phi(z_{\alpha/2}) = 1-\alpha/2$.

Using the covariate effect β_1 fixed by the user, the corresponding standard error $SE(\beta_1)$ is predicted by PFIM 3.2 for a given design and the values of population parameters.

Computation of the number of subjects needed

The number of subjects needed to achieve a power *P* to detect a covariate effect using the Wald test is also computed. First, from the equation (1), we compute the SE needed on *β* to obtain a power of *P*, called NSE(P), using the following relation:

$$NSE(P) = \frac{\beta_1}{z_{\alpha/2} - \phi^{-1}(1-P)} \quad (2)$$

Last, we compute the number of subjects needed to be included to obtained a power of *P*, called NNI(P) using

$$NNI(P) = N \times \frac{SE(\beta_1)}{NSE(P)} \quad (3)$$

where N is the initial number of subjects in the given design and $SE(\beta_1)$ the corresponding predicted SE of *β* for the given design.

1.6.2. Equivalence test

Computation of the expected power

The Wald test can be used to asses the equivalence of a covariate effect β. In PFIM, the Wald test is performed on the β of each category for each covariate, a global Wald test on the vector β (all effect coefficients) is not implemented.
For one covariate and an effect of one category β (K=2), the null hypothesis is H_0: $\{\beta \leq -\Delta_L \text{ or } \beta \geq +\Delta_L\}$ while the alternative hypothesis is H_1: $\{-\Delta_L \leq \beta \leq +\Delta_L\}$. H_0 is composed of two unilateral hypothesis $H_{0,-\Delta_L}$:$\{\beta \leq -\Delta_L\}$ and $H_{0,+\Delta_L}$:$\{\beta \geq +\Delta_L\}$. Equivalence between two covariate effects can be concluded if and only if the two hypotheses $H_{0,-\Delta_L}$ and $H_{0,+\Delta_L}$ are rejected.
The two statistics of the unilateral Wald test under the null hypothesis are defined as, $S_{W_{-\Delta_L}} = \frac{\hat{\beta} + \Delta_L}{SE(\hat{\beta})}$ and $S_{W_{+\Delta_L}} = \frac{\hat{\beta} - \Delta_L}{SE(\hat{\beta})}$ with $\hat{\beta}$ the covariate effect estimates and, its associated standard error. Under H_1, when $\beta=\beta_1$ with $\beta_1 \in [-\Delta_L, \Delta_L]$, we then compute the power of the equivalence Wald test defined as:

$$P_{equi} = 1 - \phi\left(z_{1-\alpha} - \frac{\beta_1 + \Delta_L}{SE(\beta_1)} \right) \quad if\ \beta_1 \in [-\Delta_L, 0] \tag{4}$$

$$P_{equi} = \phi\left(-z_{1-\alpha} - \frac{\beta_1 - \Delta_L}{SE(\beta_1)} \right) \quad if\ \beta_1 \in [0, +\Delta_L] \tag{5}$$

where Φ is the cumulative distribution function of the standard normal distribution and z_α is such that $\phi(z_\alpha) = 1 - \alpha$.

In equivalence test β_1 is usually chosen to be zero.

Using the covariate effect β_1 fixed by the user, the corresponding standard error $SE(\beta_1)$ is predicted by PFIM 3.2 for a given design and the values of population parameters.

Computation of the number of subjects needed

The number of subjects needed to achieve a power P to show equivalence between two covariate effects using the Wald test is also computed. First, from equations (4) and (5), we compute the SE needed on β to obtain a power of P, called NSE(P), using the following relation:

$$NSE(P) = \frac{(-\beta_1 - \Delta_L)}{-z_\alpha + \phi^{-1}(1-P)} \quad if\ \beta_1 \in [-\Delta_L, 0] \tag{6}$$

$$NSE(P) = \frac{(-\beta_1 + \Delta_L)}{z_\alpha + \phi^{-1}(P)} \quad if\ \beta_1 \in [0, +\Delta_L] \tag{7}$$

where Φ is the cumulative distribution function of the standard normal distribution and z_α is such that $\phi(z_\alpha)=1-\alpha$.

Last, we compute the number of subjects needed to be included to obtained a power of *P*, called NNI(P) using the equation (3) like for comparison test.

1.7. References

[1]. Bazzoli C, Retout S, Mentré F. Fisher information matrix for nonlinear mixed effects multiple response models: evaluation of the first order linearization using a pharmacokinetic/pharmacodynamics model. *Statistics in Medicine*. 2009 28(14): 1940–56.

[2]. Retout S, Mentré F. Further developments of the Fisher information matrix in nonlinear mixed effects models with evaluation in population pharmacokinetics. *Journal of Biopharmaceutical Statistics*. 2003 13(2) : 209–27.

[3]. Retout S, Comets E, Samson A, Mentré F. Design in nonlinear mixed effects models: Optimization using the Federov–Wynn algorithm and power of the Wald test for binary covariates. *Statistics in Medicine*. 2007 26(28) : 5162–79.

[4]. Nguyen TT, Bazzoli C, Mentré F. Design evaluation and optimization in crossover pharmacokinetic studies analysed by nonlinear mixed effect models. Application to bioequivalence or interaction trials. *American Conference on Pharmacometrics*. 2009, Mystic, United-States.

[5]. Bertrand J, Mentré F. Mathematical Expressions of the Pharmacokinetic and Pharmacodynamic Models implemented in the MONOLIX software (http://www.monolix.org). 2008.

2. Installation

2.1. Pre-requirement

Like for PFIM 3.0, the software R is required. For an optimal use of PFIM 3.2, several packages might be needed in the R library directory:

- for differential equation system to describe the model: "odesolve" and "nlme" packages
- for the Federov-Wynn algorithm: "combinat" package


Compared to PFIM 3.0, an additional package "numDeriv" is needed for the computation of the full Fisher information matrix.

The easiest way to install packages is directly from the web. To install the packages odesolve, nlme, combinat and numDeriv, start R and choose the Packages item from the menu. Choose Install package(s) from CRAN to install from the web (you will see a list of all available packages pop up -- choose odesolve, nlme combinat and numDeriv).
To install PFIM 3.2, the user has to download the package named PFIM 3.2 available on the webpage www.pfim.biostat.fr.

2.2. Components

The PFIM 3.2 package includes two main folders called:

- PFIM 3.2
- Examples

The folder PFIM 3.2 is composed of 3 principal files and one folder:

- The 3 principal files are:
  - The main function (program) file (**PFIM3.2.r**)
  - The input file (**Stdin.r**)
  - The model file (**model.r**).

- The folder is called **Program** and contains 10 files of functions:
  - ***Pfim3.2op1.r***: To compute the block diagonal Fisher Information matrix (option 1) to evaluate a population design using an analytical form to describe the model.
  - ***PfimOPT3.2op1.r***: To compute the block diagonal Fisher Information matrix (option 1) to optimise a population design using an analytical form to describe the model
  - ***EQPfim3.2op1.r***: To compute the block diagonal Fisher Information matrix (option 1) to evaluate a population design using a differential equation system to describe the model
  - ***EQPfimOPT3.2op1.r***: To compute the block diagonal Fisher Information matrix (option 1) to optimise a population design using a differential equation system to describe the model
  - ***Pfim3.2op2.r***: To compute the full Fisher Information matrix (option 2) to evaluate a population design using an analytical form to describe the model.
  - ***PfimOPT3.2op2.r***: To compute the full Fisher Information matrix (option 2) to optimise a population design using an analytical form to describe the model
  - ***EQPfim3.2op2.r***: To compute the full Fisher Information matrix (option 2) to evaluate a population design using a differential equation system to describe the model
  - ***EQPfimOPT3.2op2.r***: To compute the full Fisher Information matrix (option 2) to optimise a population design using a differential equation system to describe the model
  - ***Algosimplex3.2.r***: To use the Simplex algorithm
  - ***initfedoR.c*** and ***classfed.h***: To compile the dll
  - ***libFED.dll:*** The dynamic library of the Federov-Wynn algorithm
  - ***algofedorov3.2.r:*** To use the dynamic library libFED.dll

- ***librayPK.r*:** To use the library of pharmacokinetic models
- ***librayPD.PDdesign.r*:** To use the library of immediate response pharmacodynamic models alone
- ***librayPD.PKPDdesign.r*:** To use the library of pharmacodynamic models linked to pharmacokinetic models both written using analytical form
- ***CreateModel_PKPDdesign.r:*** To use the libraries of pharmacokinetic and pharmacodynamic models when they are writing either with different forms or both with differential equation systems.

The files in the folder Program should not be changed.

The folder called Examples contains the example files. The documentation which gives their description is included in the package PFIM3.2 with this documentation.

To install PFIM 3.2, create a directory (for example directory "U:\\My Documents\\PFIM 3.2") and download the package PFIM 3.2.

3. Use

3.1. Working directory

- Create a working directory, for example:

  "U:\\My Documents\\PFIM 3.2_examples\\Example1"

- Copy the files PFIM3.2.r, Stdin.r and model.r in this directory

- In the file "PFIM3.2.r", specify your working directory:

  directory<-"U:\\My Documents\\PFIM 3.2_examples\\Example1"

- Then, specify your program directory i.e. where is the folder called **Program**

  directory.program<-"U:\\My Documents\\PFIM 3.2\\Program"

- Save the file PFIM3.2.r

3.2. Model writting

Compared to PFIM 3.0, there is no change in the way to write a model using an analytical form or a differential equation system for single or multiple response models using the user defined option.
The main change is to write a PK/PD model using the libraries of models. Thus, several examples are presented below with the different ways of writing models.

3.2.1. Example 1: Single response with a PK or a PD model

1. PK model using an analytical form with the library of models

*In this illustration, the user creates a one response model using the model function implemented in the pharmacokinetic library (**Oral1_1cpt_kaVCl**) describing a one compartment oral absorption after a multiple dose administration (**md**). N and tau are the needed variables and thus, they have to be specified by the user in the function model. Here, we have five oral administration doses with an interval between two doses equal to twelve hours.*

```
source(paste(directory.program,dirsep,"LibraryPK.r",sep=""))

formA<-Oral1_1cpt_kaVCl_md(N=5,tau=12)[[1]]
form<-c(formA)
```

2. PK model using a differential equation form with the library of models

In this illustration, the user creates a one response model using the model function implemented in the pharmacokinetic library (**bolus_1cpt_VVmkm**) describing a one compartment bolus input with Mickaelis-Menten elimination after a single dose administration (**sd**). The dose is specified in a part of the R-script file stdin.r:

```
time.condinit<-0
condinit<-expression(c(100)) # dose=100
source(paste(directory.program,dirsep,"LibraryPK.r",sep=""))

formED<-bolus_1cpt_VVmkm()
```

3. PK model using a differential equation form with the library of models

In this illustration, the user creates a one response model using the model function implemented in the pharmacokinetic library (**infusion_1cpt_VVmkm**) describing a one compartment infusion input with Mickaelis-Menten elimination after a single dose administration (**sd**). The dose is specified as an argument of the PK function in the file model.r, not in the initial condition described in a part of the R-script file stdin.r:

```
time.condinit<-0
condinit<-expression(c(0))
source(paste(directory.program,dirsep,"LibraryPK.r",sep=""))

formED<-infusion_1cpt_VVmkm(doseMM=100, Tinf=1)
```

4. PD model using an analytical form with the library of models

*In this illustration, the user creates a one response model using the model function implemented in the library (**immed_lin_null**) describing an immediate response model with a linear drug action and without baseline.*

```
source(paste(directory.program,dirsep,"LibraryPD_PDdesign.r",sep=""))

formA<-immed_lin_null()[[1]]
form<-c(formA)
```

3.2.2. Example 2: Two responses defined by a PK/PD model

5. PK model with a linear elimination + immediate response PD model

In this illustration the user creates for the PK model, a one compartment model with bolus input and first order elimination for a single dose, and for the PD model, an immediate response model with a linear drug action and no baseline is used. As shown in the example, the PK model is given as an argument of the PD model. Thus, in the PD model the drug concentration corresponds to the expression of the PK model.

```
source(paste(directory.program,dirsep,"LibraryPK.r",sep=""))
source(paste(directory.program,dirsep,"LibraryPD_PKPDdesign.r",sep=""))
```

```
formA<-bolus_1cpt_Vk()[[1]]
formB<-immedPD_lin_null(formA)[[1]]
form<-c(formA, formB)
```

6. PK model with a linear elimination + turnover response PD model

In this example, the user creates a PK/PD model with a one compartment bolus input for the PK and a turnover response model with an inhibition on the input for the PD, using the function create_formED. The dose is specified in a part of the R-script file stdin.r.

```
source(paste(directory.program,dirsep,"CreateModel_PKPDdesign.r",sep=""))

create_formED(bolus_1cpt_Vk,turn_input_Imax)
```

7. PK model with a Mickaelis-Menten elimination + immediate response PD model

In this example, the user creates a PK/PD model with a one compartment infusion input with Mickaelis-Menten elimination for the PK and an immediate response model with a linear drug action and no baseline for the PD, using the function create_formED. In this case, the user needs to specify the dose (here equal to 100) and the duration of infusion (here equal to 1 hour)as arguments, not in the initial condition described in a part of the R-script file stdin.r.

```
source(paste(directory.program,dirsep,"CreateModel_PKPDdesign.r",sep=""))

create_formED(infusion_1cpt_VVmkm,immedPD_lin_null,doseMM=100,TInf=1)
```

3.3. General objects required for Evaluation and Optimisation

According to the new features in PFIM 3.2, only some objects have been added or modified in the input file named by default: stdin.r.

3.3.1. Full or block diagonal fisher information matrix

The following object has been added:

- **option**: integer indicating expression of the information matrix:
  - **1** for block diagonal
  - **2** for full

3.3.2. Graph option

This list of objects has been modified in the version PFIM 3.2. It allows to draw a graph with the evaluated design (evaluation step) or the optimised design (optimisation step).
Compared to the version 3.0 of PFIM, the object **names.data** has to be replaced by the two followings objects:

- **names.datax**: character vector for the names of X axis for each graph that corresponds to each type of measurement (the length of this vector must be equal to the number of responses).

- **names.datay**: character vector for the names of Y axis for each graph that corresponds to each type of measurement (the length of this vector must be equal to the number of responses).

3.4. Objects required only for IOV option

The following list of objects is associated to the specification of the inter-occasion variability in the model.

- **n_occ:**	integer indicating the number of occasions. Example: **n_occ=2**
- **gamma:**	vector of the *p* variances of the random effects for inter-occasion variability.

3.5. Objects required only for covariate option

This list of objects allows to specify the inclusion of covariate effects on some parameters of the model. In the stdin.r, it appears just before the object required for the optimisation. The user can now include in the model covariates that do not change with occasion and/or covariates that change with occasion.

3.5.1. Covariates not changing with occasion

If the user wants to deal with covariates which do not change with occasion, he has to specify the following object.

- **covariate.model:**	logical value; if T, covariates are added to the model

If the user has filled in by T the previous object, he has to specify the following objects:

- **covariate.name:**	list of character indicating the name of the covariate(s) Example: **covariate.name<-list(c("Gender"))**
- **covariate.category:**	list of vectors of categories. Each vector is associated to one covariate and defines its corresponding categories. They can be written as character or integer. Example: **covariate.category<-list(Gender=c("F","M"))**
- **covariate.proportions:**	list of vectors of proportions. Each vector is associated to one covariate and defines the corresponding proportions of subjects involved in each corresponding categories. Example: **covariate.proportions<-list(Gender=c(0.5,0.5))**
- **parameter.associated:**	list of vectors of parameter(s) associated with each covariate. Each vector is associated to one covariate and is defined by the corresponding parameters on which is added the covariate. Example: **parameter.associated<-list(Gender=c(Cl, V))**  **Name of the parameter(s) has to be identical to those entered in the object parameters.**

- **beta.covariate:** list of the values of parameters for all other categories than the reference category (for which beta=0. Example:
  beta.covariate<-list(Gender=list(c(0.5,0.6)))

3.5.2. Covariates changing with occasion

If the user wants to deal with covariates which change with occasion, he has to specify the following object.

- **covariate_occ.model:** logical value; if T, covariates changing with occasion are added to the model

If the user has filled in by T the previous object, he has to specify the following objects:

- **covariate_occ.name:** list of character indicating the name of the covariate(s) Example:
  covariate_occ.name<-list(c("Treat"))

- **covariate_occ.category:** list of vectors of categories. Each vector is associated to one covariate and defines its corresponding categories. They can be written as character or integer. Example:
  covariate_occ.category<-list(Treat=c("A","B"))

- **covariate_occ.sequence:** list of vectors of sequences. Each vector is associated to one sequence of values of covariates at each occasion. The size of each sequence has to be equal to the number of occasions (**n_occ**) for each covariate. Example:
  covariate_occ.sequence<-
  list(Treat=list(c("A","B"),c("B","A"))

- **covariate_occ.proportions:** list of vectors of proportions. Each vector is associated to one covariate and defines the proportions of elementary designs corresponding to each sequence of covariate values. The size of each vector has to be equal to the number of sequences. Example:
  covariate_occ.proportions<-
  list(Treat=list(0.5,0.5))

- **parameter_occ.associated:** list of vectors of parameter(s) associated with each covariate. Each vector is associated to one covariate and is defined by the corresponding parameters on which is added the covariate. Example:
  parameter_occ.associated<-list(Treat=c(Cl))

  Name of the parameter(s) has to be identical to those entered in the object parameters.

- **beta.covariate_occ:**	list of the values of parameters for all other categories than the reference category for which beta=0. Example: **beta.covariate_occ<- list(Treat=list(c(log(1.1)))**

3.6. Objects required only for computation of power and number of subjects needed for comparison test or equivalence test

To compute the expected power to detect covariate effects as to compute the number of subjects needed to achieve a given power, the previous object **covariate.model** has to be filling in by T.

Additional R objects are required to be created.

The following object is needed for both options

- **alpha:**	the value of the type one error for the Wald test. Example: **alpha<-0.05**

It is possible to compute either the expected power only or the number of subjects needed for a given power or both of them together.

- **compute.power**	logical value, if T the expected power **for comparison test** is computed for each covariate. Example: **compute.power<-T**
- **compute.nni**	logical value, if T the number of subjects needed for a given power **for comparison test** is computed for each covariate. Example: **compute.nni<-T**
- **interval_eq**	vector of equivalence interval. Example: **interval_eq<-c(log(0.8),log(1.25))**
- **compute.power_eq**	logical value, if T the expected power **for equivalence test** is computed for each covariate. Example: **compute.power_eq<-T**
- **compute.nni_eq**	logical value, if T the number of subjects needed for a given power **for equivalence test** is computed for each covariate. Example: **compute.nni_eq<-T**
- **given.power**	the value of the given power for comparison and/or equivalence test. Example: **given.power<-0.9**

4. Results

The results are written in the output file called by default stdout.r. This file is different when only evaluation or optimisation is performed. Compared to the one computed in PFIM 3.0, the file is only modified when inter-occasion variability and/or covariate options are added. It is only detailed below for evaluation but it is similar for optimisation.

Figure 1 represents the output file from the design evaluation of a model with covariates effect and inter-occasion variability.

The user can read on the Figure 1:

1 The name of the function used: PFIM 3.2.

2 The name of the project and the date.

3 A summary of the input: model(s), sampling times in the elementary designs for each model(s), doses or initial conditions and subjects corresponding to those designs, residual variance error model for each model(s), random effect model, error tolerances for the solver of differential equations system if used, a summary for inter-occasion variability, a summary of the covariate model

- 3a The number of occasions
- 3b Each covariate not changing with occasion, its (their) parameters associated, categories with their name and their corresponding proportions of subjects.
- 3c Each covariate changing with occasion, its (their) parameters associated, categories with their name and their corresponding proportions of subjects. The list of the sequence of values of categories for each occasion and each covariate.
- 3d If the evaluation has been performed using the full or the block diagonal expression of the Fisher information matrix

4 The population Fisher information matrix, a dim*dim symmetric matrix where dim is the total number of population parameters to be estimated.

5 The value of each population parameter with the expected standard error on each parameter and the corresponding coefficient of variation.

- 5a The value of the variance of the random effects for inter-occasion variability with the expected standard error on each parameter and the corresponding coefficient of variation.

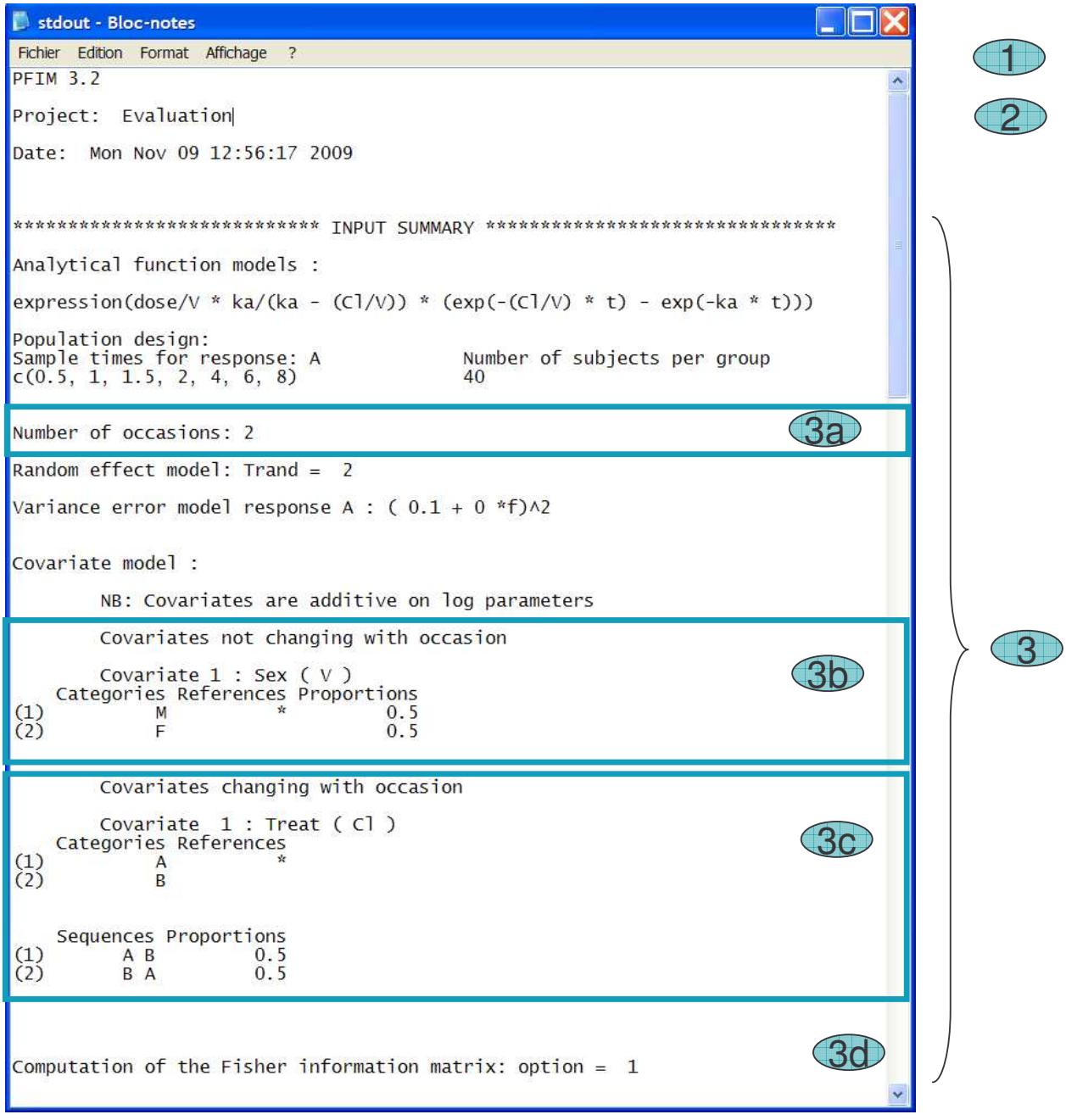
6

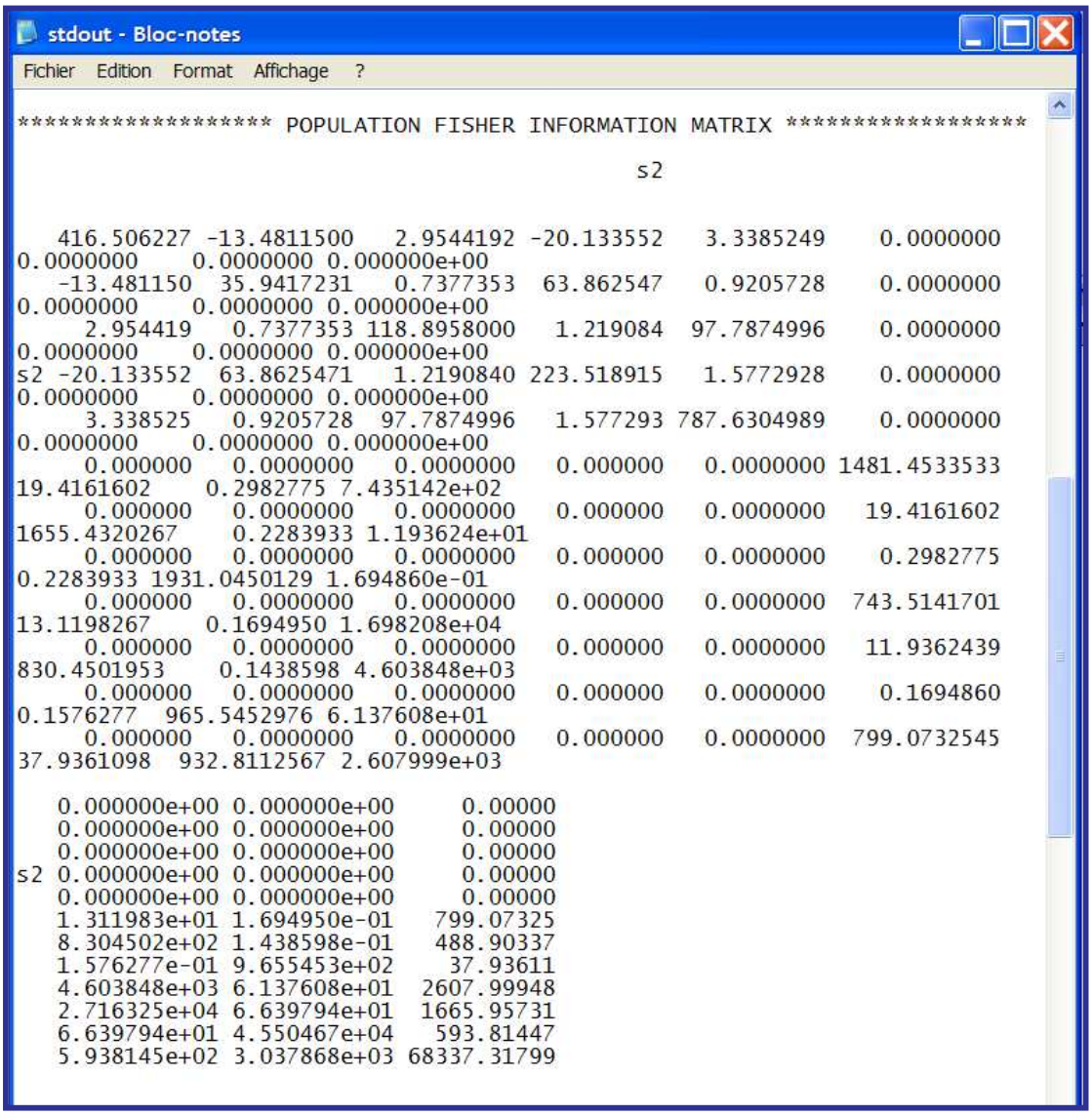
The value of the determinant of the Fisher information matrix and the value of the criterion (determinant^(1/dim), where dim is the total number of population parameters).

Results from the comparison test: the value and the exponential of the value of each covariate parameter with the corresponding 95% confidence interval of the parameter, the predicted value of the power of the Wald test and the number of subjects needed to detect this covariate effect with the given type one error and the given power.

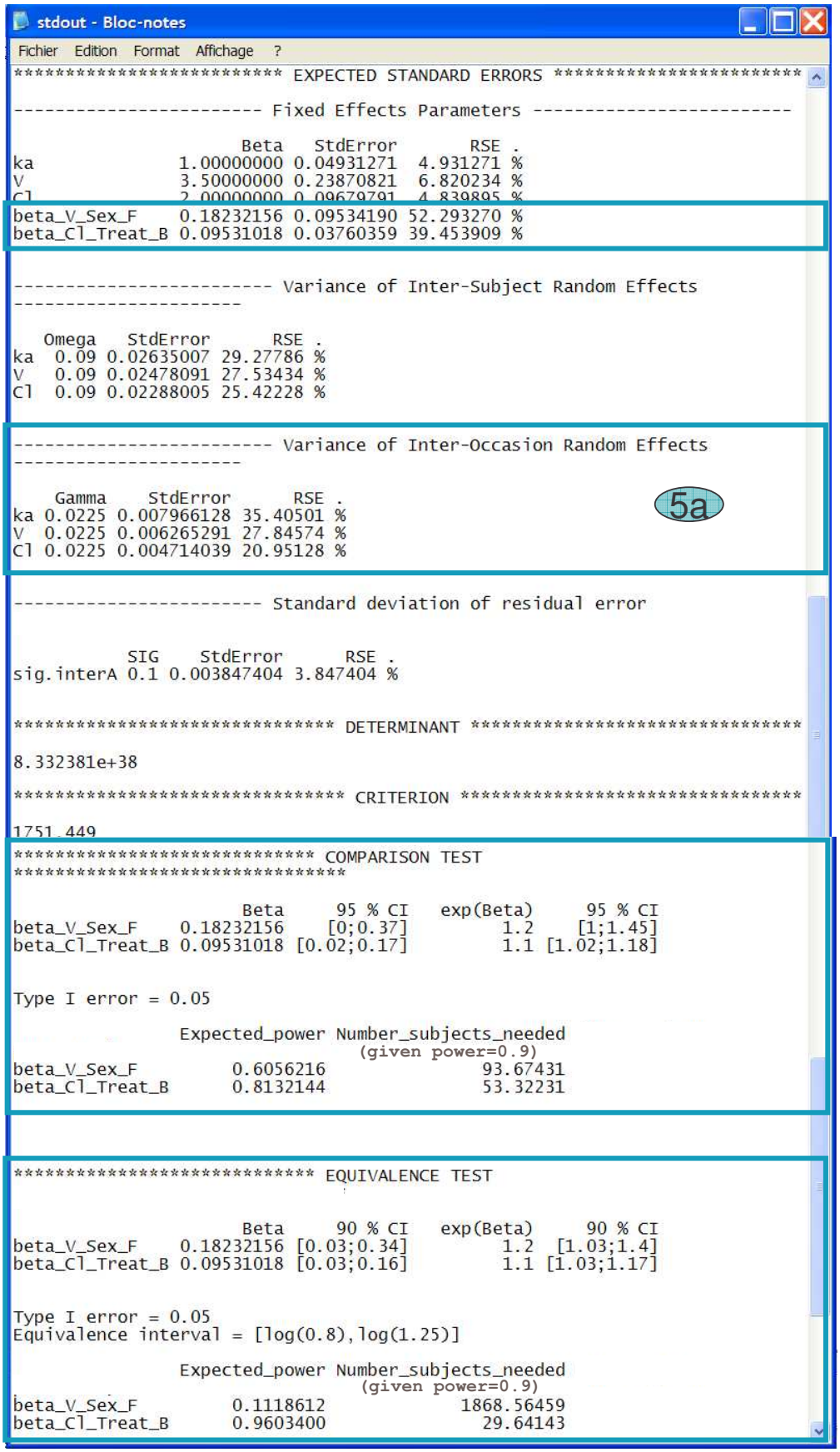
Results from the equivalence test: the value and the exponential of the value of each covariate parameter with the corresponding 90% confidence interval of the parameter, the power of the Wald test and the number of subjects needed to achieve the given power for this covariate effect with the given type one error and the given interval of equivalence.

Figure 1. Example of design evaluation output file with covariate effect and inter-occasion variability




stdout - Bloc-notes
Fichier Edition Format Affichage ?
********** POPULATION FISHER INFORMATION MATRIX **********
s2
416.506227 -13.4811500 2.9544192 -20.133552 3.3385249 0.0000000
0.0000000 0.0000000 0.000000e+00
-13.481150 35.9417231 0.7377353 63.862547 0.9205728 0.0000000
0.0000000 0.0000000 0.000000e+00
2.954419 0.7377353 118.8958000 1.219084 97.7874996 0.0000000
0.0000000 0.0000000 0.000000e+00
s2 -20.133552 63.8625471 1.2190840 223.518915 1.5772928 0.0000000
0.0000000 0.0000000 0.000000e+00
3.338525 0.9205728 97.7874996 1.577293 787.6304989 0.0000000
0.0000000 0.0000000 0.000000e+00
0.000000 0.0000000 0.0000000 0.000000 0.0000000 1481.4533533
19.4161602 0.2982775 7.435142e+02
0.000000 0.0000000 0.0000000 0.000000 0.0000000 19.4161602
1655.4320267 0.2283933 1.193624e+01
0.000000 0.0000000 0.0000000 0.000000 0.0000000 0.2982775
0.2283933 1931.0450129 1.694860e-01
0.000000 0.0000000 0.0000000 0.000000 0.0000000 743.5141701
13.1198267 0.1694950 1.698208e+04
0.000000 0.0000000 0.0000000 0.000000 0.0000000 11.9362439
830.4501953 0.1438598 4.603848e+03
0.000000 0.0000000 0.0000000 0.000000 0.0000000 0.1694860
0.1576277 965.5452976 6.137608e+01
0.000000 0.0000000 0.0000000 0.000000 0.0000000 799.0732545
37.9361098 932.8112567 2.607999e+03
0.000000e+00 0.000000e+00 0.00000
0.000000e+00 0.000000e+00 0.00000
0.000000e+00 0.000000e+00 0.00000
s2 0.000000e+00 0.000000e+00 0.00000
0.000000e+00 0.000000e+00 0.00000
1.311983e+01 1.694950e-01 799.07325
8.304502e+02 1.438598e-01 488.90337
1.576277e-01 9.655453e+02 37.93611
4.603848e+03 6.137608e+01 2607.99948
2.716325e+04 6.639794e+01 1665.95731
6.639794e+01 4.550467e+04 593.81447
5.938145e+02 3.037868e+03 68337.31799

4


stdout - Bloc-notes
Fichier Edition Format Affichage ?
********** EXPECTED STANDARD ERRORS **********
---------- Fixed Effects Parameters ----------
Beta StdError RSE .
ka 1.00000000 0.04931271 4.931271 %
V 3.50000000 0.23870821 6.820234 %
Cl 2.00000000 0.09679791 4.839895 %
beta_V_Sex_F 0.18232156 0.09534190 52.293270 %
beta_Cl_Treat_B 0.09531018 0.03760359 39.453909 %
---------- Variance of Inter-Subject Random Effects ----------
Omega StdError RSE .
ka 0.09 0.02635007 29.27786 %
V 0.09 0.02478091 27.53434 %
Cl 0.09 0.02288005 25.42228 %
---------- Variance of Inter-Occasion Random Effects ----------
Gamma StdError RSE .
ka 0.0225 0.007966128 35.40501 %
V 0.0225 0.006265291 27.84574 %
Cl 0.0225 0.004714039 20.95128 %
5a
---------- Standard deviation of residual error ----------
SIG StdError RSE .
sig.interA 0.1 0.003847404 3.847404 %
********** DETERMINANT **********
8.332381e+38
********** CRITERION **********
1751.449
********** COMPARISON TEST **********
Beta 95 % CI exp(Beta) 95 % CI
beta_V_Sex_F 0.18232156 [0;0.37] 1.2 [1;1.45]
beta_Cl_Treat_B 0.09531018 [0.02;0.17] 1.1 [1.02;1.18]
Type I error = 0.05
Expected_power Number_subjects_needed (given power=0.9)
beta_V_Sex_F 0.6056216 93.67431
beta_Cl_Treat_B 0.8132144 53.32231
********** EQUIVALENCE TEST **********
Beta 90 % CI exp(Beta) 90 % CI
beta_V_Sex_F 0.18232156 [0.03;0.34] 1.2 [1.03;1.4]
beta_Cl_Treat_B 0.09531018 [0.03;0.16] 1.1 [1.03;1.17]
Type I error = 0.05
Equivalence interval = [log(0.8),log(1.25)]
Expected_power Number_subjects_needed (given power=0.9)
beta_V_Sex_F 0.1118612 1868.56459
beta_Cl_Treat_B 0.9603400 29.64143
5
6
7
8

Moreover, the PFIM() function returns the following R objects:

- **Dose**
- **prot**: design evaluated for each response
- **subjects**: number of subjects for each group
- **mfisher**: the population Fisher information matrix
- **determinant**: the determinant of the population Fisher information matrix
- **crit**: the value of the criterion
- **p**: the vector of the fixed effect parameters
- **se**: the vector of the expected standard errors for each parameter
- **cv**: the corresponding coefficients of variation, expressed in percent.
- **summary.exp.power**: a matrix with each row corresponding to each covariate, the name of the covariates, the associated effect parameter, the 95% confidence interval, and the predicted power as columns
- **summary.nni**: a matrix with each row corresponding to each covariate, the name of the covariates, the associated effect parameter, the 95% confidence interval and the number of subjects needed as column