Syneos Health
Shortening the distance from lab to life®.

Assembling pharmacometrics datasets in R: the puzzle package

Mario González-Sales[1,2], Olivier Barriere[3], Pierre-Olivier Tremblay[1], Guillaume Bonnefois[1], Julie Desrochers[1], Fahima Nekka[4].

[1]Syneos Health, Quebec, Canada, [2]Modeling Great Solutions, Escaldes-Engordany, Andorra, [3]Certara, Quebec, Canada, [4]Université de Montreal, Quebec, Canada.

Introduction

The pharmacometrics (PMX) workflow has routine steps (see Figure 1). The automation of these steps saves time and money, reduces the risk of errors, and increases reproducibility. Currently, several excellent tools are available and used to enhance the PMX workflow.[1-12] However, to the best of our knowledge, there is no open-source tool to support assembling PMX datasets.

Objective

To develop an R package to simplify the time consuming and error prone task of assembling PMX datasets in order to speed up and increase the reproducibility of the PMX workflow.

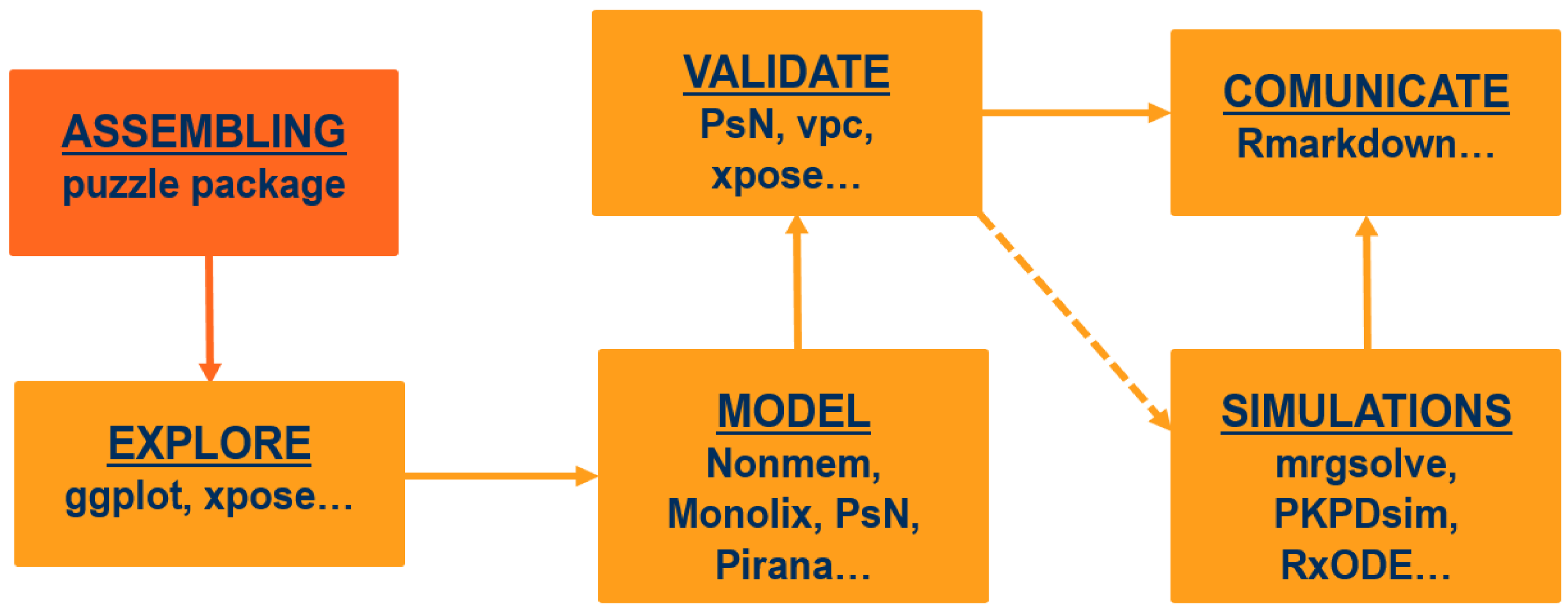

Figure 1. Typical pharmacometrics workflow.

Methods

Puzzle consists of a series of functions written in R that are able to create from tabulated files, PMX datasets compatible with the NONMEM® formatting.

Results

The **puzzle()** function easily assembles complex PMX datasets. Users are able to select from different absorption processes such as zero- and first-order, or a combination of both. Furthermore, datasets containing data from one or more analytes, and/or one or more responses, and/or time in- and/or dependent, and/or event varying covariates, and/or urine data can be simultaneously assembled.
The default output of **puzzle()** is a ".csv" file. If no `name` is specified in `nm`, the output is a data frame that can be further manipulated in R. The **puzzle** package can be installed in R using: **devtools::install_github("syneoshealth/puzzle")**.

pk.csv

ID	TIME	DV
1	0	0
1	2	10.8
1	8	7.6
1	30	3.2
2	0	0
2	1.9	9.97
2	7.5	12.1
2	30.1	2.5

pd.csv

ID	TIME	DV
1	0	25
1	12.1	100
1	48.4	45
2	0	32
2	12.2	77
2	48.5	53

dose.csv

ID	TIME	AMT
1	0	200
1	24	100
2	0	200
2	24	100

covariate.csv

ID	TIME	VARIABLE	VALUE
1	0	WEIGHT	72
1	12	WEIGHT	71
1	24	WEIGHT	70
2	0	WEIGHT	95
2	12	WEIGHT	96
2	24	WEIGHT	94
1	0	SEX	F
2	0	SEX	M

Figure 2. Example of the structure of the tabulated files required by puzzle().

Example of puzzle() in action:

A pharmacokinetic (PK) and pharmacodynamics (PD) dataset assuming bolus administration, and including time dependent covariates (*i.e.* body weight) will be assembled using the tabulated files depicted in Figure 2. The R syntax is presented below and the output from **puzzle()** is represented in Figure 3.

```
puzzle::puzzle(
 directory=file.path(your_path),
 order=0,
 pk=list(name="pk.csv"),
 pd=list(name="pd.csv"),
 dose=list(name="dose.csv"),
 cov=list(name="covariate.csv"),
 coercion=list(name="coercion.txt"),
 nm=list(name="PKPD.csv"))
```

The **puzzle()** function simultaneously assembles the PK, PD, dose and covariate information to return a NONMEM® ready dataset (*i.e.* "PKPD.csv"). The variable type of observation (*i.e.* TYPE) has been created and appended to the "PKPD.csv" file. It always has a value of 0 for dosing event and values of 1 and 2 for PK, and PD observations, respectively. A bolus administration has been specified with the argument order (*i.e.* `order=0`),. Additionally, **puzzle()** appends useful items to the output: a placeholder to ignore records (*i.e.* C), the time after dose (*i.e.* TAD), the log of the DV (*i.e.* LDV), the event identification (*i.e.* EVID), the missing dependent variable item (*i.e.* MDV), the compartment item (*i.e.* CMT), and others (not showed). After running **puzzle()**, if there are characters to be coerced, or categories to be defined in character variables, an automatic coercion message will be printed in the R console.

	pk.csv and pd.csv			dose.csv	covariate.csv		Variables added by puzzle()						
	C	ID	TIME	DV	AMT	SEX	WEIGHT	TAD	LDV	EVID	MDV	TYPE	CMT
dose		1	0	.	200	0	72	0	.	1	1	0	1
pk		1	0	0	.	0	72	0	.	0	0	1	1
pd		1	0	25	.	0	72	0	3.218	0	0	2	2
pk		1	2	10.8	.	0	72	2	2.379	0	0	1	1
pk		1	8	7.6	.	0	72	8	2.028	0	0	1	1
pd		1	12.1	100	.	0	71	12.1	4.605	0	0	2	2
dose		1	24	.	100	0	70	0	.	1	1	0	1
pk		1	30	3.2	.	0	70	6	1.163	0	0	1	1
pd		1	48.4	45	.	0	70	24.4	3.806	0	0	2	2
dose		2	0	.	200	1	95	0	.	1	1	0	1
pk		2	0	0	.	1	95	0	.	0	0	1	1
pd		2	0	32	.	1	95	0	3.465	0	0	2	2
pk		2	1.9	9.97	.	1	95	1.9	2.299	0	0	1	1
pk		2	7.5	12.1	.	1	95	7.5	2.493	0	0	1	1
pd		2	12.2	77	.	1	96	12.2	4.343	0	0	2	2
dose		2	24	.	100	1	94	0	.	1	1	0	1
pk		2	30.1	2.5	.	1	94	6.1	0.916	0	0	1	1
pd		2	48.5	53	.	1	94	24.5	3.970	0	0	2	2

Figure 3. Example of the PMX dataset produced by puzzle().

Furthermore, if specified when calling **puzzle()**, (*i.e.* `coercion=list(name="coercion.txt")`) a ".txt" file will be created (*i.e.* "coercion.txt"). This file will contain the information of the coerced variables. In this example: `CMT: 1=pk, 2=pd` and `SEX: 0=F, 1=M`.
puzzle() has additional arguments to increase its flexibility: `extratimes`, adds extra times to obtain smooth predictions; `optionalcolumns`, allows adding extra columns defined in the dose.csv file; `fillcolumns`, uses the last observation carried forward to fill columns added with `optionalcolumns`; `parallel`, lets the user to define a sequential zero and first order absorption process; `datetimeformats`, defines the format for date and time; `timeunits`, defines the units for time; `timezone`, allows the user to change the location to compute the time according to time zones; `initialindex`, defines the initial value for coerced variables; `arrange`, may be used to arrange variables; `na.strings`, defines the value for a not available value; `missingvalues`, defines the value for missings, and `nocoercioncolums`, allows dropping variables to be coerced.

Conclusions

The **puzzle** package is a powerful and efficient tool that helps modelers, programmers and pharmacometricians through the difficult and complex process of assembling PMX datasets and it is the first open-source tool supporting this step of the PMX workflow.

References

[1] Keizer RJ, et al. Biomed 2011: 101; 72-9.
[2] Lindbom L, et al. Comput Methods Programs Biomed. 2005: 79; 241-57.
[3] Jonsson EN, et al. Comput Methods Programs Biomed. 1999: 58; 51-64.
[4] Guiastrennec B, et al. The xpose R package.
[5] Wang W, et al. CPT Pharmacometrics. 2006: 5; 3-10.
[6] Metrum Research Group. Mrgsolve package.
[7] Beal S, et al. NONMEM User's guide. (1989-2009).
[8] Lavielle M. Chapman & Hall Biostatistics 2014.
[9] Keizer RJ. The vpc package.
[10] RStudio team. The rmarkdown package.
[11] Keizer RJ. The PKPDsim package.
[12] Wickham H. The ggplot2 package.

mario.gonzalezsales@syneoshealth.com