Pharmpy: a versatile open-source library for pharmacometrics

Rikard Nordgren[1], Stella Belin[1], Zhe Huang[1], Aurelien Ooms[1], Simon J. Carter[1], Xiaomei Chen[1], Osama Qutishat[1], Alzahra Hamdan[1], Shijun Wang[1], Tianwu Yang[1], Piyanan Assawasuwannakit[1], Simon Buatois[2], João A. Abrantes[2], Andrew C. Hooker[1], Mats O. Karlsson[1]

[1]Department of Pharmacy, Uppsala University, Sweden

[2]Roche Pharma Research and Early Development, Roche Innovation Center Basel, Basel, Switzerland

What is Pharmpy?

Pharmpy is an open-source software package for pharmacometric modeling (available in Python and in R via the pharmr package). It has functionality ranging from reading and manipulating model files and datasets, to executing workflows as full tools and collecting and presenting subsequent results. It can parse many different NONMEM models into components such as parameters, statements, and differential equations, however the model objects themselves are not bound to their original language, they are general. Due to the model abstraction in Pharmpy, many manipulations can be applied to the model and then either update the original code or, as a future extension, translate to other languages. The transformations can be combined into more complex tools which can generate candidate models, fit them with e.g. NONMEM and then select the best model based on a pre-specified criteria, such as OFV or BIC.

Key features

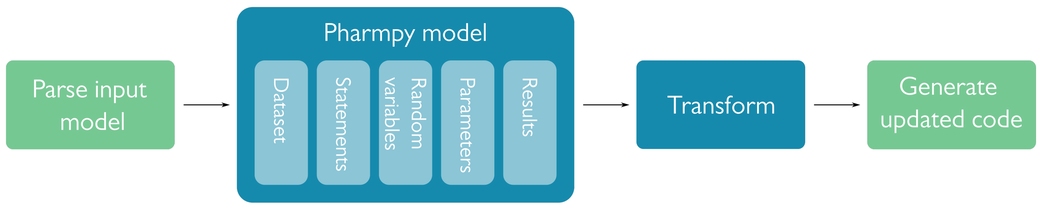
- **Independent of modeling language** – model abstraction is separate from model language which allows for easier integration of additional modeling languages
- **Tools are modular** – parts of tools can easily be split and recombined into new tools
- **Aid in reproducibility and traceability** – possible to fully script a workflow directly in R (including preprocessing of data, manipulations to the model, fitting, and summarizing results)
- **Flexible** – scope can range from low-level manipulations of models to full tools with automatic model building

Pharmpy is available on PyPI and pharmr is available on CRAN.

https://pharmpy.github.io/

Parse and represent models and its components

All parsed models are broken into different components such as parameters, random variables, and statements. The dataset is also parsed, and together with a datainfo-file you can specify the content of each column (IDV, covariate etc.). By abstracting an input model to a Pharmpy model, it is possible to read in a NONMEM model and translate the model into e.g. nlmixr (which is implemented as a proof of concept).



Script and explore model building

Many types of transformations can be applied to the Pharmpy object via the modeling API. The transformations can be piped and the subsequent model changes can be seen by printing different attributes of the Pharmpy models. After modifying the model, you can also generate updated model code. The modeling module also has functions for summarizing results, parameter sampling, and dataset manipulations.

Example functions

- Automatic conversion to mu-referenced model
- Adding time after dose to dataset
- Change structural model (automatic ADVAN conversion for NONMEM)
- Summarize modelfit results (e.g. OFV, runtime, parameter estimates)

An example: mu-referencing

```
m <- read_model('path/to/model') %>%
     mu_reference_model(m)
```

Before

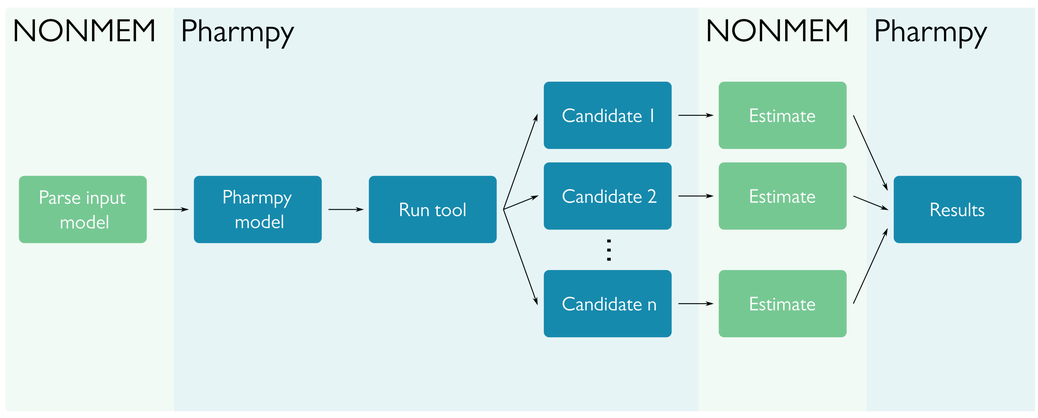
```
$PK
TVCL=THETA(1)*WGT
TVV=THETA(2)*WGT
CL=TVCL*EXP(ETA(1))
V=TVV*EXP(ETA(2))
S1=V
```

After

```
$PK
TVCL=THETA(1)*WGT
TVV=THETA(2)*WGT
MU_1 = LOG(TVCL)
CL = EXP(ETA(1) + MU_1)
MU_2 = LOG(TVV)
V = EXP(ETA(2) + MU_2)
S1=V
```

Automatic model building and complex tools

Finally, the transformations can be combined into more complex tools. The tools have functionality for creating model candidates and fitting multiple models (using parallelization where possible), then presenting the subsequent results. The tools are implemented independently from the estimation software, it is only when fitting that the tools use software specific code. Furthermore, multiple tools are parts of an overarching Automatic Model Development (AMD) tool [1]. In multiple tools, a machine learning method is used to predict influential individuals and outliers [2] for all candidate models created by the tool.



Example tools

- **modelsearch** – search for best structural model for a PK model [3]
- **iivsearch** – search for best IIV structure (both variance and covariance)
- **resmod** – search for best residual error model
- **estmethod** – compare estimation methods and ODE solvers

An example: iivsearch-tool

```
m <- read_model('path/to/model')
algorithm <- 'brute_force_no_of_etas'
res <- run_iivsearch(model=m,
                     algorithm=algorithm,
                     rank_type='bic')
```

Summary of tool

	description	dbic	rank
model			
candidate3	[CL,VC]	4.282024	1
base_model	[CL,VC,MAT]	0.000000	2
candidate2	[CL,MAT]	-217.691571	3
candidate6	[CL]	-385.641987	4
candidate1	[MAT,VC]	-631.871991	5
candidate5	[VC]	-633.419500	6
candidate4	[MAT]	-815.076729	7
candidate7	[]	-834.249063	8

Acknowledgements

This work was supported by F. Hoffmann-La Roche Ltd., Basel, Switzerland and Bayer AG. A special thanks to Dr. Emilie Schindler, Dr. Sylvie Retout, Dr. Valérie Cosson, and Dr. Franziska Schaedeli Stark from F. Hoffmann-La Roche Ltd., Basel for conducting testing and giving feedback.

In collaboration with

Roche

BAYER

UPPSALA UNIVERSITET

List of references

[1] Xiaomei Chen, Alzahra Hamdan, Shijun Wang, Tianwu Yang, Rikard Nordgren, Stella Belin, Zhe Huang, Simon J. Carter, Simon Buatois, João A. Abrantes, Andrew C. Hooker, Mats O. Karlsson, Development of a tool for fully automatic model development (AMD), PAGE 2022

[2] Osama Qutishat, Simon J. Carter, Rikard Nordgren, Alzahra Hamdan, Shijun Wang, Tianwu Yang, Xiaomei Chen, Simon Buatois, João A. Abrantes, Andrew C. Hooker and Mats O. Karlsson, The development of artificial neural networks for the prediction of influential individuals and outlying individuals and their application during the model building development process, PAGE 2022

[3] Alzahra Hamdan, Xiaomei Chen, Stella Belin, Rikard Nordgren, Simon Buatois, João A. Abrantes, Andrew C. Hooker and Mats O. Karlsson, Automatic Development of Pharmacokinetic Structural Models – Pharmpy Model Search Tool, PAGE 2022