



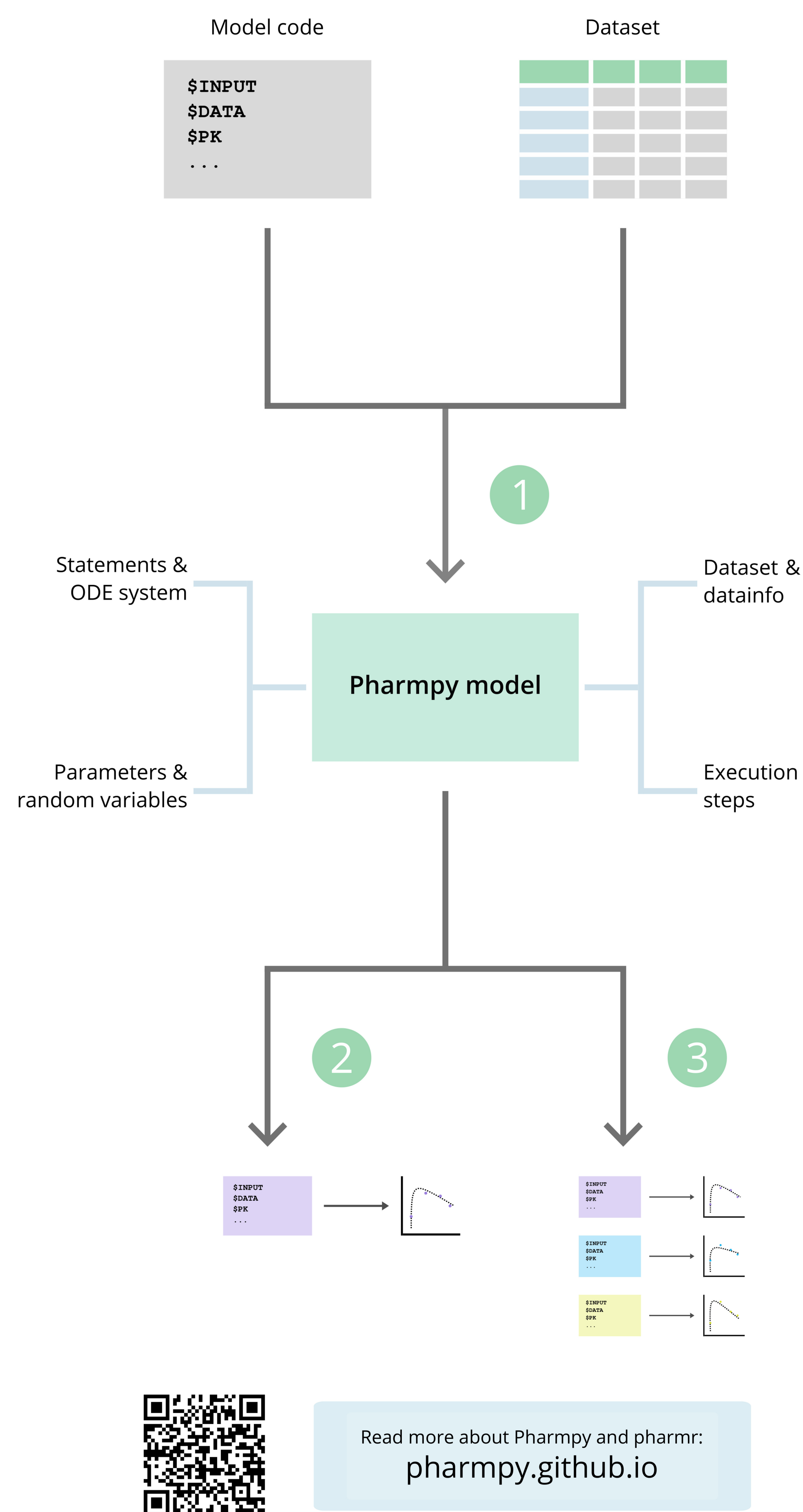
PharmPy: a versatile open-source library for pharmacometrics

Rikard Nordgren¹, Stella Belin¹, Xiaomei Chen¹, Johan Borg¹, Antonia Hufnagl¹, Keqi Shi¹, Andrew C. Hooker¹, Mats O. Karlsson¹
¹Department of Pharmacy, Uppsala University, Sweden

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 running bigger tools. Key features include:

- Ability to fully script modeling workflows
Script various aspects of model building, such as creating models from scratch, transforming the dataset, and estimating models.
- Independent of modeling language
Once parsed or created, all models are represented as a general PharmPy model object, and all transformations and tools are independent of the modeling language (e.g. NONMEM).
- Tools for automatic model building
Develop parts of or a full model automatically, either standalone or the fully automatic model development (AMD) tool where initial estimates, selection criteria, and search algorithms have been considered.
- Available in Python, R, and from command line
PharmPy can be run in Python and via its CLI, and pharmr can be used to run PharmPy from R.



- 1 Parsing and/or creating the PharmPy model
A PharmPy model can be created by reading in a NONMEM model or be created from scratch. A dataset can also be attached to the model, which can be modified (such as adding a time after dose column), used for informed initial estimates, etc. Once parsed, the general PharmPy object will be used for all transformations and tools, and only written to the software format once it's needed, such as when executing the model or writing the model to a file.

Modeling software support

- NONMEM
- nlmixr2
- rxode2

- 2 Transform and execute a model
Modeling functions can be used to modify the model object and the dataset. After modifying the model, you can also generate updated model code. It is also possible to execute the model in your Python or R script, where you can specify which external tool to use.

```
# Mu-reference a model and fit
# using nlmixr
m <- read_model('path/to/model') %>%
  mu_reference_model()
res <- fit(m, esttool='nlmixr')
```

```
# Create a model from scratch and modify it
m <- create_basic_pk_model('iv', 'path/to/data') %>%
  add_iov('VISI', 'CL') %>%
  add_allometry('WGT') %>%
  set_simulation(seed=1234) %>%
  convert_model('nonmem')
```

- 3 Automatically develop models
PharmPy has tools for automatic model building. These tools create model candidates, fit the candidates (using parallelization), then present the subsequent results, which includes a final model. Model types supported include PK, PKPD, TMDD, and drug-metabolite models.

Examples of tools

- AMD
Develops a model from a dataset
- IIVSearch
Chooses IIV structure
- CovSearch
Test covariate effects (features from SCM+)

```
# Run tool for automatic model development
mfl <- 'ABSORPTION([Z0,F0]);PERIPHERALS(0..2);
COVARIATE?(@PK, @CONTINUOUS, exp, *)'
res <- run_amd(input='path/to/dataset',
  modeltype='basic_pk',
  administration='oral',
  cl_init=2.0,
  vc_init=5.0,
  mat_init=3.0,
  strategy='default',
  search_space=search_space,
  allometric_variable='WGT',
  occasion='VISI')
```

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. Franziska Schaedeli Stark, Dr. Sreenath Krishnan, Dr. Orwa Albitar, and Dr. Valerie Cosson for conducting testing and providing feedback.

In collaboration with



UPPSALA
UNIVERSITET