A flexible and transparent MATLAB framework for empirical and mechanistic pharmacometric modelling

Niklas Hartung[1] · Charlotte Kloft[2] · Wilhelm Huisinga[1]

(1) Institute of Mathematics, University of Potsdam, Germany (2) Institute of Pharmacy, Freie Universität Berlin, Germany

Background

For many research questions, model complexity needs to be adapted during the modelling process, e.g. by refining empirical models or coarsening mechanistic models.

Established software tools mostly support either an empirical PK/PD approach (e.g. NONMEM, Monolix) or a mechanistic PBPK / Systems Pharmacology approach (e.g., GastroPlus, PK-Sim, SimCYP) and **lack** the **flexibility** to switch between these two paradigms.

Building on a MATLAB toolbox developed over the years in research and teaching as part of the PharMetrX PhD program [1-5], we developed a **new MATLAB-based pharmacometric modelling framework** specifically designed to

- adapt model complexity to the research question under investigation
- support integration of databases
- promote modelling transparency

Scope of toolboxes

	Databases (physiology / drug properties)	Virtual populations and scaling methods	Fully flexibile model design (empirical / mechanistic)	Check for unit consistency / tracking of assumptions
PBPK toolboxes (PK-Sim, SimCyp, GastroPlus)	✓	✓		
Plain Matlab / R			✓	
This toolbox	✓	✓	✓	✓

Computation with units

- Extension of contributed MATLAB toolbox [6]
- Unit consistency check during any computation

Example: units
How many drug molecules per E. coli cell?

```
>> V = 2e-15*u.L;  % E. coli volume
>> C = 1*u.nM;     % drug concentration
>> n = V*C*u.NA
n = 1.2
```

Solving ODEs with units

class-based design, operator overloading

⇓

same ODE syntax with/without units

⇓

unit check frequency is flexible

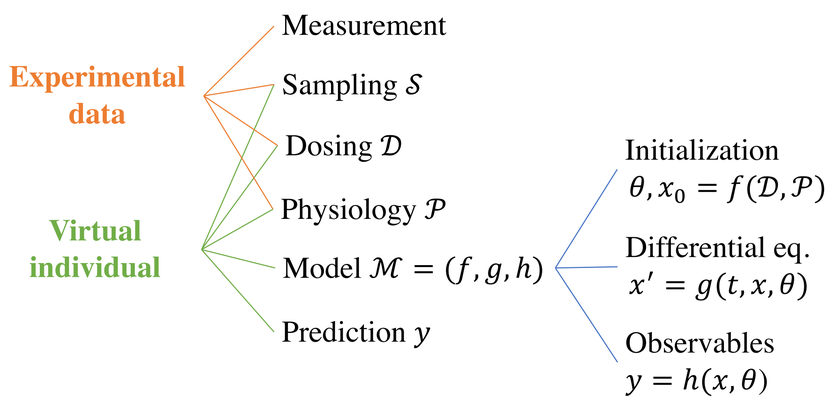
Runtime comparison

Frequency of unit check

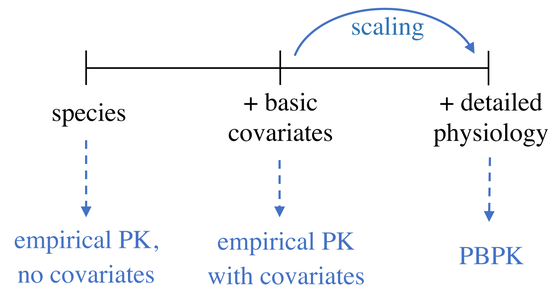
Model	Never	Once	Always
1 cmt PK	1.0 ms	1.3 ms	18 ms
13 cmt PBPK	35 ms	41 ms	1.2 s

Toolbox overview

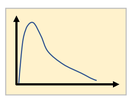
Infrastructure



Physiology: levels of detailedness

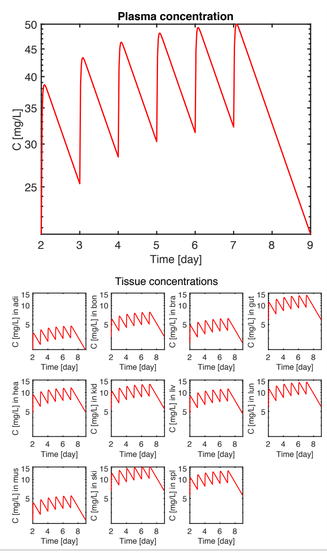


Simple workflow

1. define $\mathcal{S}, \mathcal{D}, \mathcal{P}, \mathcal{M}$
2. initialize: $\mathcal{D}, \mathcal{P} \xrightarrow{f} x_0, \theta$
3. simulate: $x_0, \theta, \mathcal{S} \xrightarrow{g} x$
4. plot: $x, \theta \xrightarrow{h}$ 

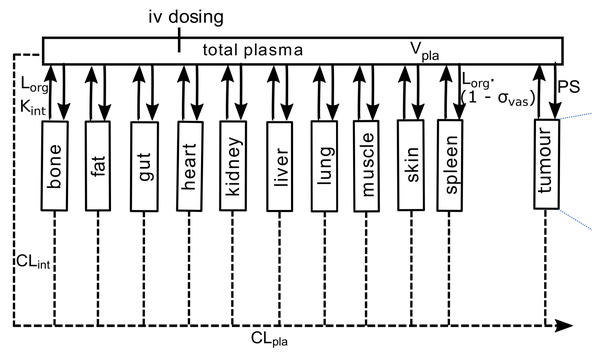
Example: main script

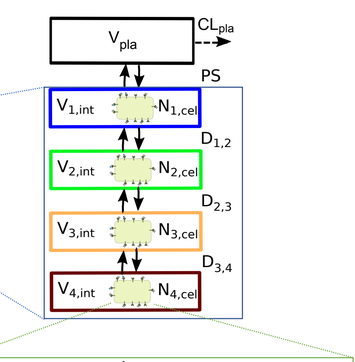
```
% Prepare scaling
refhuman  = Physiology('human_male_35y_73kg_176cm');
tallhuman = Covariates('species', 'human', ...
    'sex', 'male', 'age',35*u.year, 'BW', 90*u.kg, ...
    'BH', 1.90*u.m);

% define sampling (S), dosing (D), physiology (P), model (M)
indv = Individual;
indv.physiology = scaling_LBW(refhuman, tallhuman);
indv.dosing     = oral('Warfarin', (0:1:7)*u.day, 250*u.mg);
indv.sampling   = Sampling(2*u.day:30*u.min:8*u.day);
indv.modelfun   = @sMD_PBPK_13CMT_wellstirred;
indv.options.tissuePartitioning = @rodgersrowland;

% initialize, simulate, plot
indv.model  = initialize(indv);
indv.output = simulate(indv);
plot(indv)
```



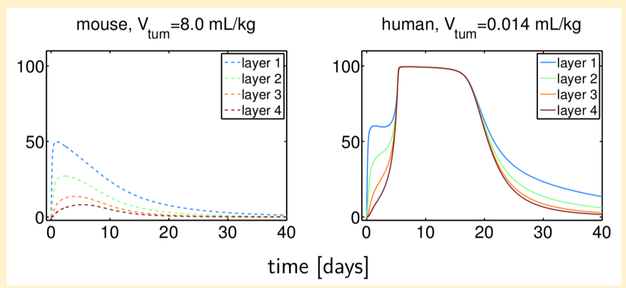
Examples shown in live demo

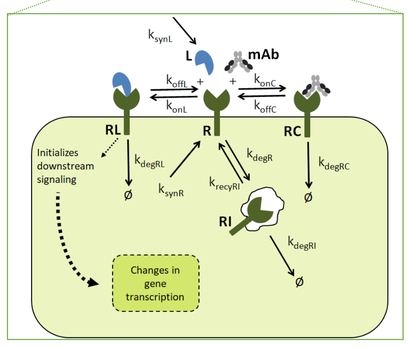
Kinetics of monoclonal antibodies in tumour [4]



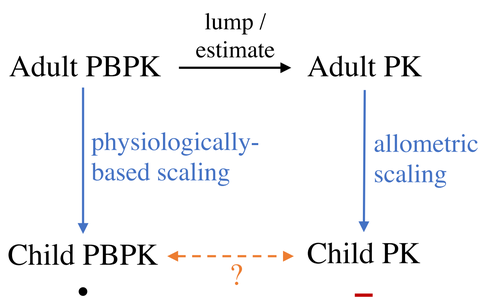


Simulation result: receptor saturation (in [%])

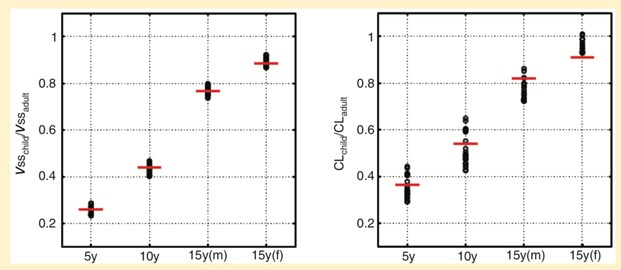




Comparison of scaling methods to children [1]



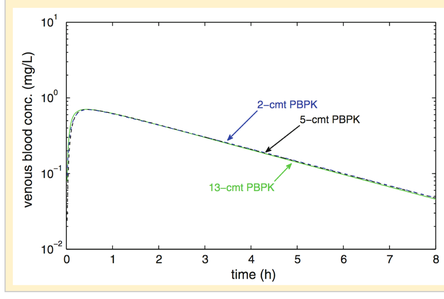
Simulation result: comparing PK parameter extrapolation



Interpretation of compartmental models via lumping [7]

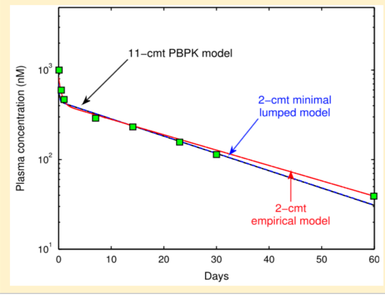
Small molecule PBPK model (well-stirred liver model)

Simulation result: Lidocaine (sMD)



Large molecule PBPK model (extravasation-limited)

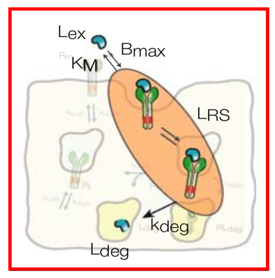
Simulation result: 7E3 (mAb)

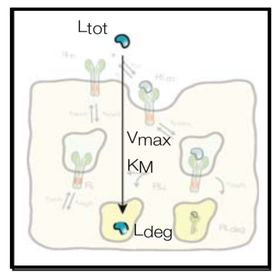


Receptor-mediated endocytosis of therapeutic proteins [8]

Question: When does Michaelis-Menten adequately describe receptor-mediated endocytosis?

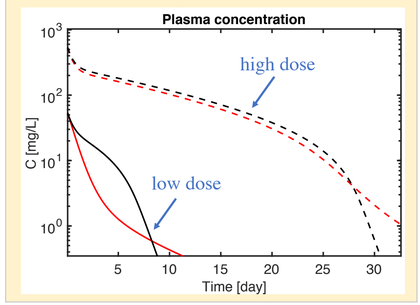
Saturable distribution





Michaelis-Menten

Simulation result: Zalutumumab



References

[1] Huisinga W et al. CPT:PSP (2012) 1:e4
[2] Fronton L et al. J PKPD (2014) 41(2):87–107
[3] Huisinga W et al. (2016) "Target-driven pharmacokinetics of biotherapeutics" in: Zhou, Theil (Eds.), Wiley
[4] Fuhrmann S et al. J PKPD (2017) 44(4):351–374
[5] https://www.pharmetrx.de/
[6] Physical Units toolbox, version 4.1.0.0 https://de.mathworks.com/matlabcentral/fileexchange/38977-physical-units-toolbox
[7] Pilari S, Huisinga W. J PKPD (2010) 37:365–405
[8] Krippendorff BF et al. J PKPD (2009) 36:239–260

Contact

For additional information, please contact Niklas Hartung,
niklas.hartung@uni-potsdam.de