Solving ODEs in the wild: Scalable pharmacometrics with Stan

Sebastian Weber, Eero Siivola

Statistical Methodology Group, Novartis Pharma AG

StanCon Helsinki, 29-31 Aug 2018

Outline

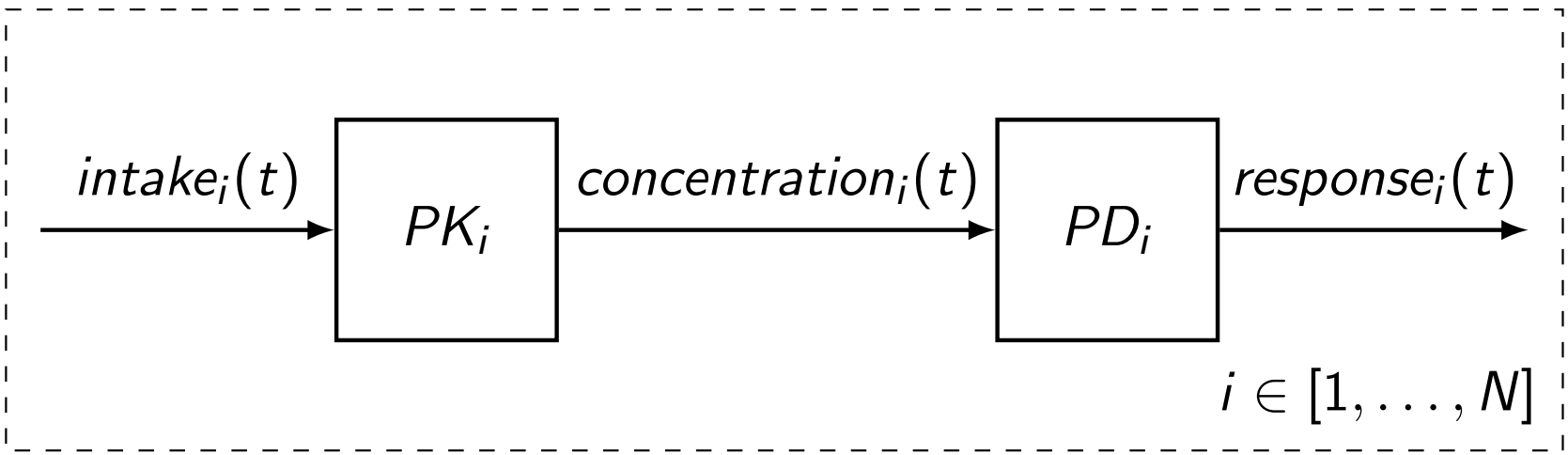
Outline

Pharmacometric data

- Drug therapy aims to treat a disease
  - Drug is administered someway in order to reach some location where it is active.
  - Drug concentration in blood is often taken as a surrogate for exposure at the target tissue.
- Drug research data contains observations of drug doses and concentration measurements for multiple patients over a period of time.

Pharmacometric models

- Pharmacometrics aims to model:
  - Pharmacokinetics (PK): Relation of drug admission and drug concentration. "PK is what the body does to the drug".
  - Pharmacodynamics (PD): Relation of drug concentration and drug effect. "PD is what the drug does to the body".
- These processes are stated as ordinary differential equations (ODEs) where there is a link process between PK and PD and each of N patients has different ODE parameters.



Learning pharmacometric models

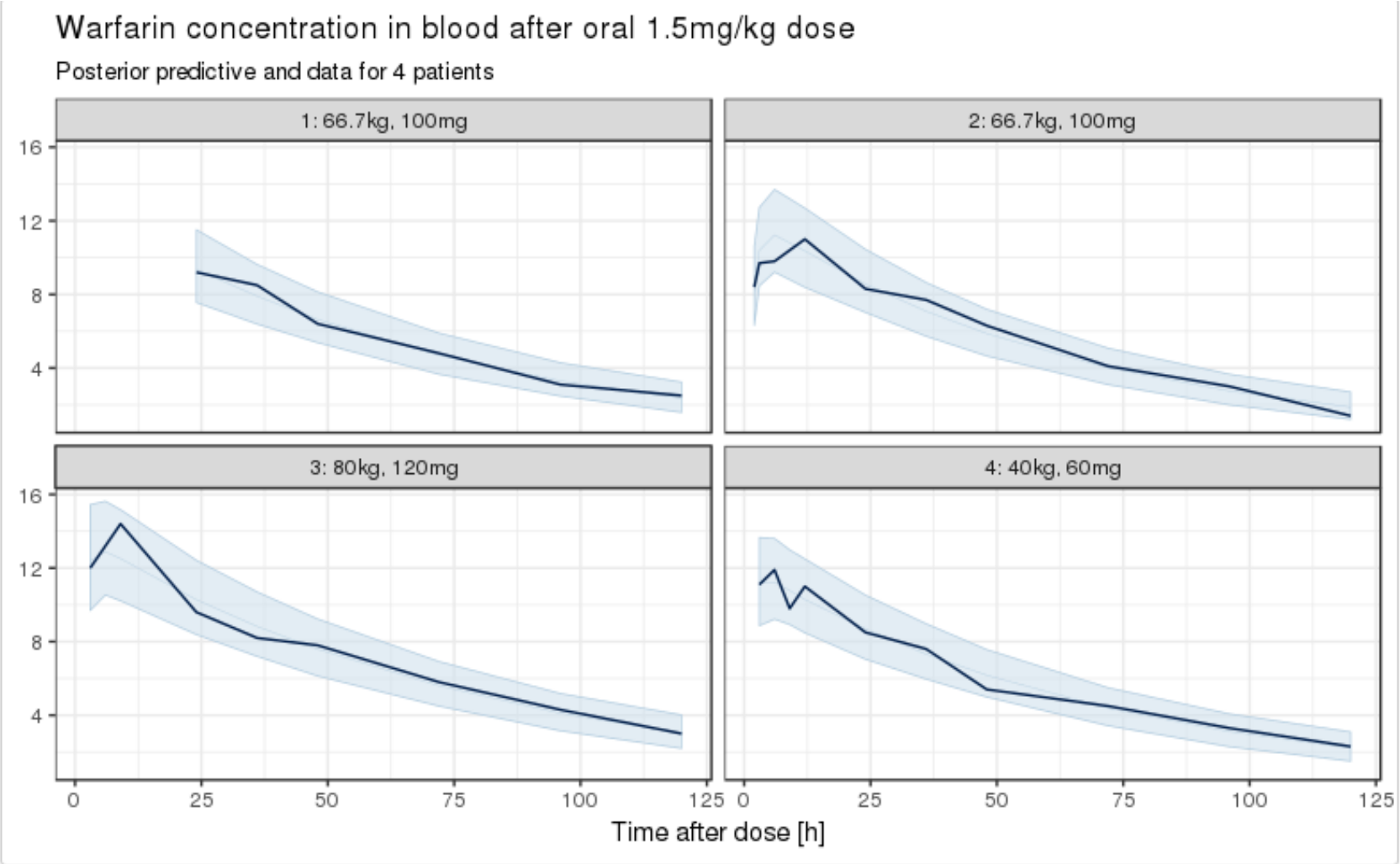
- Learning the joint fit of both ODEs at once is expensive and the modeling requires lots of time and effort.
- Often learning PK parameters first, fixing them and using it as a forcing function for the PD model is used.
- In this work we concentrate on speeding up PD models as they give a great example of forcing functions and cannot be solved analytically.

Outline

Outline

Pharmacokinetics of blood thinner Warfarin

- The pharmacokinetics of Warfarin can be described by a first order process that can be solved analytically:



Pharmacodynamic model for Warfarin

- Quantitative measure of Warfarin is change in the prothrombin complex levels.
- Concentration over time has fixed parameters learned from the PK model.
- The pharmacodynamics is described by a semi-mechanistic process:

$$\frac{dR_i(t)}{dt} = k_{in,i}\left(1 - \text{logit}^{-1}\left(\log\left(C_i(t)\right) - \log\left(EC50_i\right)\right)\right) - k_{out,i}R_i(t)$$

  - $R_i(t)$ is the response
  - $k_{in,i}$ and $k_{out,i}$ are influx and outflux constants
  - $EC50_i$ is concentration when response is 50% of maximum.
- This turn-over model cannot be solved analytically.

Outline

ODE function in Stan for the turn-over function

- The Stan code for the turn-over function could look something like:

```
real[] turnover_kin_inhib_1(real t, real[] R, real[] theta, real[] x_r, int[] x_i) {
    real ldose = x_r[1];
    real llag = x_r[2];
    real lka = x_r[3];
    real lCl = x_r[4];
    real lV = x_r[5];
    real lconc = pk_1cmt_oral_tlag([t]', ldose, llag, lka, lCl, lV)[1];
    real lkout = -theta[2];
    real lkin = theta[1] + lkout;
    real lEC50 = theta[3];
    real lS = log_inv_logit(lconc - lEC50);
    // dRdt = kin * (1 - C/(C + EC50)) - R * kout
    return { exp(lkin + log1m_exp(lS)) - R[1] * exp(lkout) };
}
```

- Stan run with 250 warmup and sampling for 32 patients:

	warmup	sample	sum
chain:1	12.67	2.12	14.80
chain:2	13.16	2.07	15.23
chain:3	13.14	2.09	15.23
chain:4	15.4	2.04	17.45

- This is bad!
  - Variable definitions (ldose/llag/...) inside a function are considered as parameters.
  - Autodiff stack grows a lot mostly by the call to pk_1cmt_oral_tlag and this makes Stan slow.

New ODE in Stan

- Better way of defining the function:

```
real[] turnover_kin_inhib_2(real t, real[] R, real[] theta, real[] x_r, int[] x_i) {
    //real ldose = x_r[1];
    //real llag = x_r[2];
    //real lka = x_r[3];
    //real lCl = x_r[4];
    //real lV = x_r[5];
    real lconc = pk_1cmt_oral_tlag_t(t, x_r[1], x_r[2], x_r[3], x_r[4], x_r[5]);
    real lkout = -theta[2];
    real lkin = theta[1] + lkout;
    real lEC50 = theta[3];
    real lS = log_inv_logit(lconc - lEC50);
    // dRdt = kin * (1 - C/(C + EC50)) - R * kout
    return { exp(lkin + log1m_exp(lS)) - R[1] * exp(lkout) };
}
```

- Stan run with 250 warmup and sampling for 32 patients:

	warmup	sample	sum
chain:1	6.10	0.96	7.06
chain:2	6.22	0.94	7.16
chain:3	6.17	0.94	7.11
chain:4	6.13	0.96	7.09

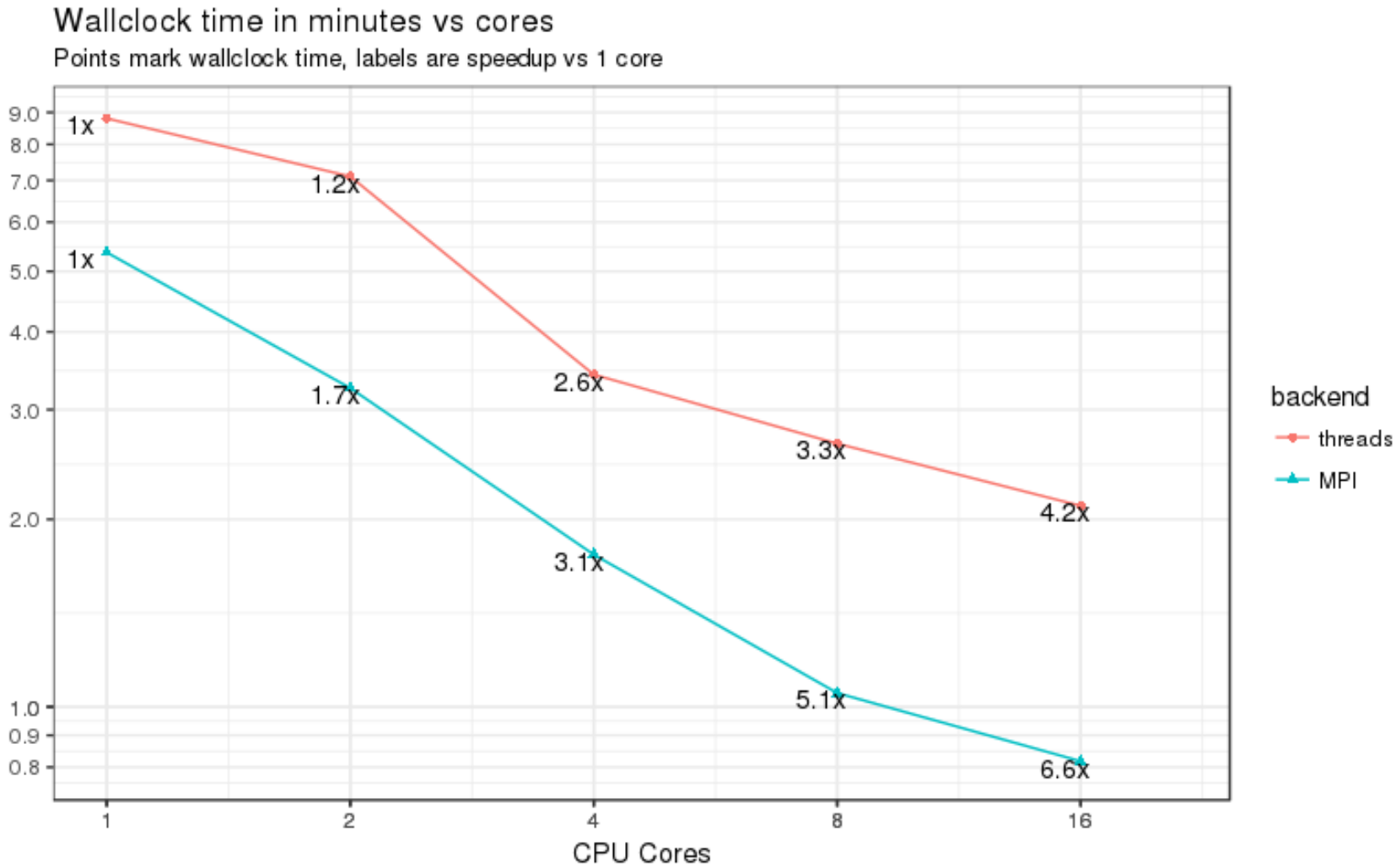
- Better!

Outline

Problem formulation is embarrassingly parallel

- Pharmacometric models are by default hierarchical so that all patients are exchangeable.
  - The likelihood of a given patient can be evaluated in independence of all other patients.
- This hierarchical structure can be taken advantage of using the new *map_rect* function of Stan.
  - This function applies a user-defined function to a set of parameters which are in rectangular data storage format.
  - Evaluations can be performed in parallel using either threading or the message passing interface (MPI).

Computation time as a function of CPU cores



Outline

Conclusion

- We can go from 15 minutes to 45 seconds!
- Stan has all required components for realistic pharmacometric problems.
- Stan 3 will allow the user to declare which function parameters are data and which are not. This will allow the code to be more readable in the future.
- *map_rect*-function gives huge performance gains for large problems.