Solving ODEs in the wild: Scalable pharmacometrics with Stan

Sebastian Weber, sebastian.weber@novartis.com

20th May 2018

Abstract

Pharmacometric modeling involve nonlinear hierarchical models which are most naturally expressed as ordinary differential equations. These class of models lead to a number of challenges which complicate a practical modeling work-flow in Stan. At the example of the public Warfarin data-set I will present solutions to these. I will first illustrate the concept of forcing functions and how to use these efficiently in Stan. Moreover, I will show how the hierarchical structure of the problem is most efficiently taken advantage of to parallelize the model computations which expedites the model evaluation time and allows to scale Stan's performance to large data-sets given sufficient computing resources.

Introduction

A drug therapy aims to treat a disease state. The drug is administered through some route in order to reach some location in the human body where the drug is active. The the human body is exposed to the drug and a readily accessible measure for drug exposure is the drug concentration in blood which is often taken as surrogate for exposure at the target tissue. The key to a successful therapy is to attain quickly an adequate exposure which is high enough for a sufficient effect to be reached, but low enough to avoid undesired adverse affects. This defines the so-called therapeutic window within which one aims to maintain patients using an adequate dosing regimen.

Pharmacometrics now aims to model the longitudinal history of a patient being under drug treatment. That is, one models how drug intake translates into a drug concentration (pharmacokinetics) and how drug concentration drives some effect (pharmacodynamics) over time. These processes are most naturally stated as ordinary differential equations (ODEs) where the phamacokinetics (PK) follows mass-action kinetics and the pharmacodynamics (PD) is modeled using a link process relating drug concentration (PK) with drug effect (PD).

This translates mathematically into a forced ODE problem. The PK model describes the so-called absorption, distribution, metabolism and elimination process (ADME). The PK model itself is externally driven by the input into the system which is the intake of drug amount over time. The intake of drug amount is referred to as dosing regimen and is known as data in advance or recorded during the trial. The PD model then describes the effect of the drug concentration on some measure of efficacy or safety.

Oftentimes these problems are approached in a two-step manner. First the PK model is informed using blood concentration data. Then the model parameters are fixed and the PD model is informed. Thus, the PK model becomes a forcing function to the PD model. The "clean" Bayesian solution is a joint fit of the combined model, but oftentimes a two-step approach is much quicker to implement and it gives a great example for forcing functions here.

In the following I will first introduce the warfarin data-set, then illustrate the concept of forcing functions and how to introduce them efficiently in Stan. Finally, I will show how the hierarchical problem structure can be used most efficiently to parallelize computations and expedite the model evaluation time for the example at hand, but most importantly enable Stan to scale it's performance to large data sets when given sufficient computing resources.

Anticoagulant Warfarin

Pharmacokinetics

Warfarin is an anticoagulant (blood thinner) used to prevent strokes in certain conditions. The pharmacokinetics of Warfarin (O'Reilly, Aggeler, and Leong 1963,Holford (1986)) is well-described by a one compartmental model which states that drug amounts (mass) is cleared from a compartment via a first order process,

$$d(C_i(t)\,V_i)/dt = -Cl_i\,C_i(t) + f_{in,i}(t).$$

Here the index i labels patients and the quantities introduced are

- $C_i(t)$: Drug concentration in the central compartment at time-point t.
- V_i: Apparent volume of the central compartment.
- Cl_i: Clearance of the drug from the central compartment in units of mass per volume.
- $f_{in,i}(t)$: Uptake of drug into the system which is determined by the route of administration.

Note that while we measure *concentrations* $C_i(t)$ of some patient i, the mass-action kinetics holds for *mass*. Moreover, the uptake of drug may occur through various routes like intra-venous injections, intra-venous infusions or oral administration. The example data set studied a single orally administered Warfarin dose of 1.5mg/kg. Oral administration is usually well-described by a first order process itself (input rate is proportional to $k_{a,i}$) such that the full PK system is most conveniently written as

$$\begin{aligned} d(a_i(t))/dt &= -k_{a,i}\,a_i(t) \\ d(C_i(t)\,V_i)/dt &= -Cl_i\,C_i(t) + k_{a,i}\,a_i(t). \end{aligned}$$

Fortunately, this ODE system can solved analytically. After administration of a single dose at $t = 0$ the solution is

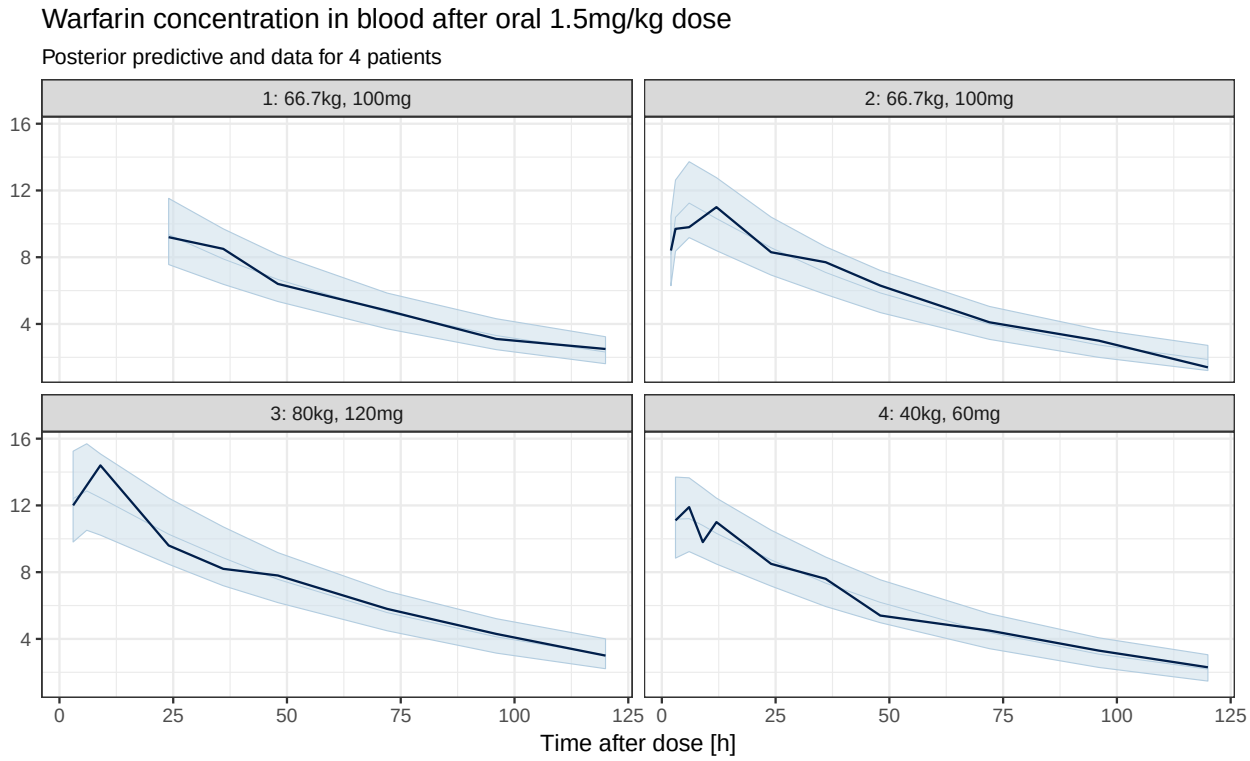
$$C_i(t) = \frac{D_i}{V_i}\,\frac{k_{a,i}}{k_{a,i} - Cl_i/V_i}\left(\exp\left(-Cl_i/V_i\,t\right) - \exp\left(-k_{a,i}\,t\right)\right).$$

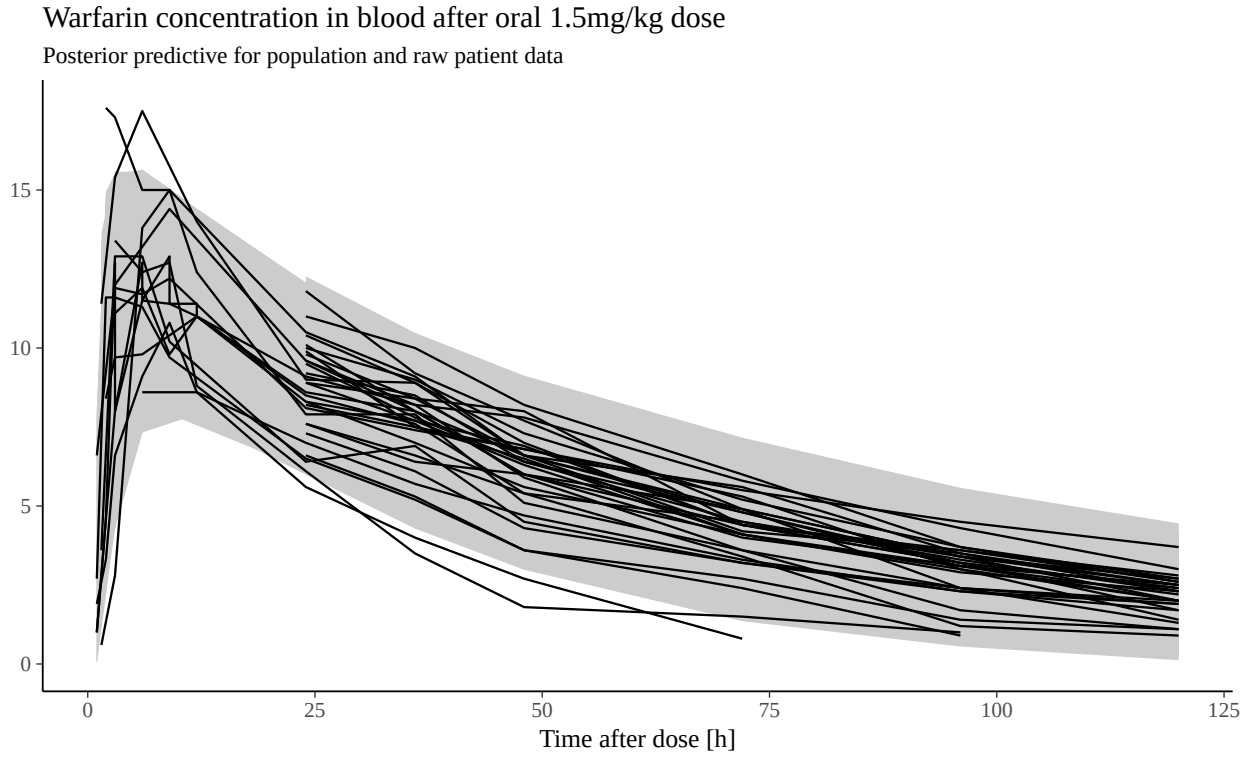
However, Warfarin has (as many drugs) an additional important property which is a delay of the drug-uptake. Thus, administering the drug at $t = t_0$ does not cause an immediate diffusion of the drug to the central compartment, but rather it takes some amount of time t_{lag} to see a non-zero concentration in the main compartment. The need for a time lag suggests that the actual absorbtion process is likely more involved than a simple first process, but a lag time is adequate here to account for it given that the interest is in time-scales much larger than the lag time. One approach to account for the lag time is to infer t_{lag} and then shift the analytic solution without a lag time by this amount. So the starting time of drug uptake is shifted by t_{lag}. While common in practice, this introduces the need for a discontinuous if-statement which depends on a parameter. This is not ideal for the performance of any gradient based approach like Stan's NUTS.

The data set we consider comprises 32 patients. For 14 patients blood samples measuring the concentration in the central compartement have been taken starting from 0.5h post oral dose administration while for 18 patients sampling started at 24h post oral dose administration. The hierarchical model for the parameters is

$$\begin{aligned} \text{logit}^{-1}(t_{lag,i}/t_{lag,max}) &\sim N(\nu_1, \sigma_1^2) \\ \log(k_{a,i}) &\sim N(\nu_2, \sigma_2^2) \\ \log(Cl_i) &\sim N(\nu_3 + 3/4\,\log(wt_i/70kg), \sigma_3^2) \\ \log(V_i) &\sim N(\nu_4 + \log(wt_i/70kg), \sigma_4^2). \end{aligned}$$

The model takes allometric scaling into account for the clearance and volume. This scaling accounts for a size dependent metabolic activity of the organs involved in eliminating the drug and the slopes are derived from first principles. The fit of the PK model is shown below for 4 patients as an example and for the full population. All codes are provided in this notebook, but in the following we will concentrate on the pharmacodynamic model of Warfarin. The parameters of the PK model are in the following assumed known and they are set to the median values of the PK model fit.





Pharmacodynamics

The effect of Warfarin is to thin blood and a quantitative measure is its effect on the prothrombin complex levels in relation to normal levels.

In the following, the pharmacodynamics will be conditioned on the pharmacokinetic model and since we have fixed the model parameters, the PK model is effectively a forcing function to the PD model. Thus, the concentration $C_i(t)$ at time t for patient i can be treated as data within the context of the PD model. An appropriate PD model for the effect of Warfarin on prothrombin complex levels is the turn-over model. The turn-over model is semi-mechanistic since it is certainly an over-simplification of the biological processes, but it is still motivated by biological rationale. Specifically, the response $R_i(t)$ is considered to be represented by a compartment which has a zero-order influx $k_{in,i}$ and a first order out-flux $k_{out,i}$. The drug-effect may enter this model through an inhibition or stimulation of either process. For Warfarin (O'Reilly and Aggeler 1968,Holford (1986)), an inhibition of the zero-order k_{in} is an adequate choice,

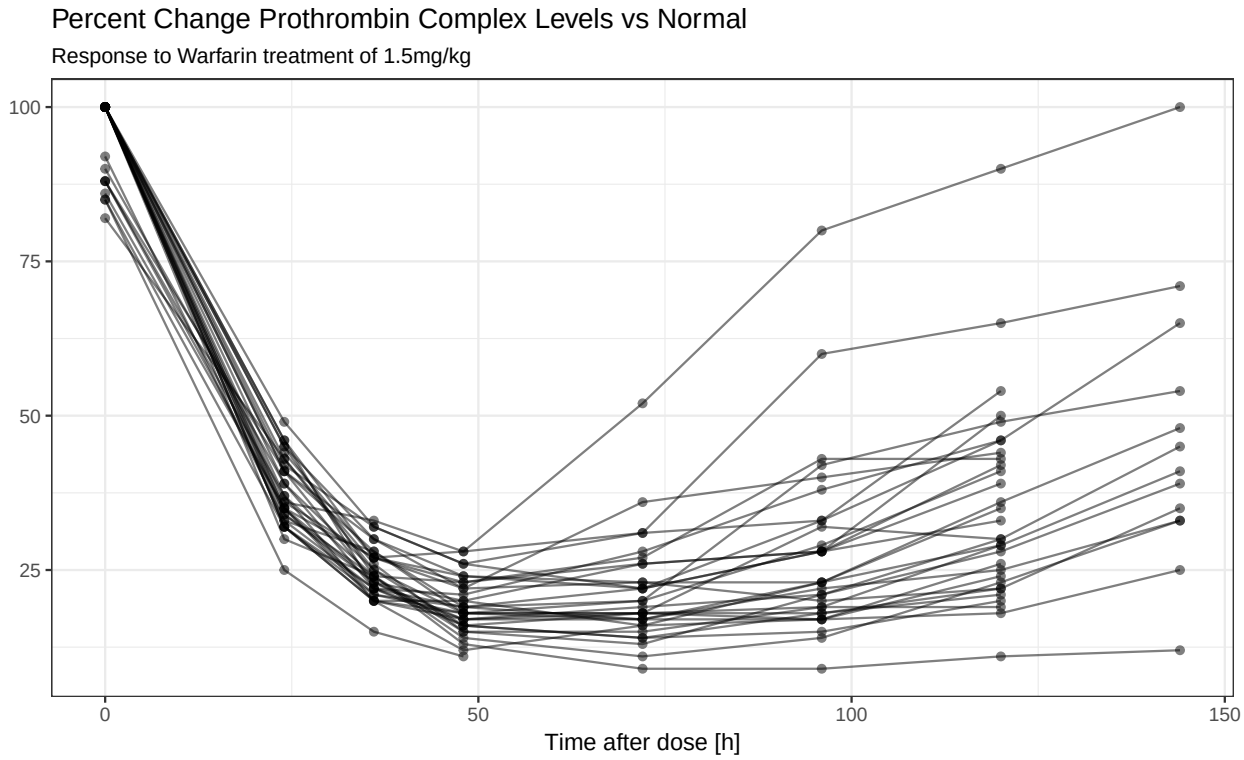
$$dR_i/dt = k_{in,i}\,(1 - \frac{C_i(t)}{C_i(t) + EC50_i}) - k_{out,i}\,R_i$$
$$\Leftrightarrow dR_i/dt = k_{in,i}\,\left\{1 - \text{logit}^{-1}[\log(C_i(t)) - \log(EC50_i)]\right\} - k_{out,i}\,R_i.$$

Whenever the drug concentration is equal to the $EC50_i$ value, then half of the full drug effect is reached. Furthermore, in absence of drug, the response R_i will reach a steady-state which is $R_{i,ss} = \frac{k_{in,i}}{k_{out,i}}$ (just set $dR_i/dt = 0$). Since at $t = 0$ the human body is free of any drug and we assume that at $t = 0$ the system is in steady-state such that the response at $t = 0$ is used to set $k_{in,i}$ by multiplication with $k_{out,i}$ (so we do not fit k_{in} as an independent parameter). The hierarchial model structure for the PD model is

$$\log(R_{0,i}) \sim N(\theta_1, \omega_1^2)$$
$$\log(1/k_{out,i}) \sim N(\theta_2, \omega_2^2)$$

$$\log(EC50_i) \sim N(\theta_3, \omega_3^2).$$

In this form, the turn-over model describes a reduction of the response variable as a result of drug treatment, since the value of k_{in} is decreased through the drug. This is in line with the raw data:



The ODE functor which defines the turn-over model could look like shown below when coded in Stan:

```
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) };
}
```

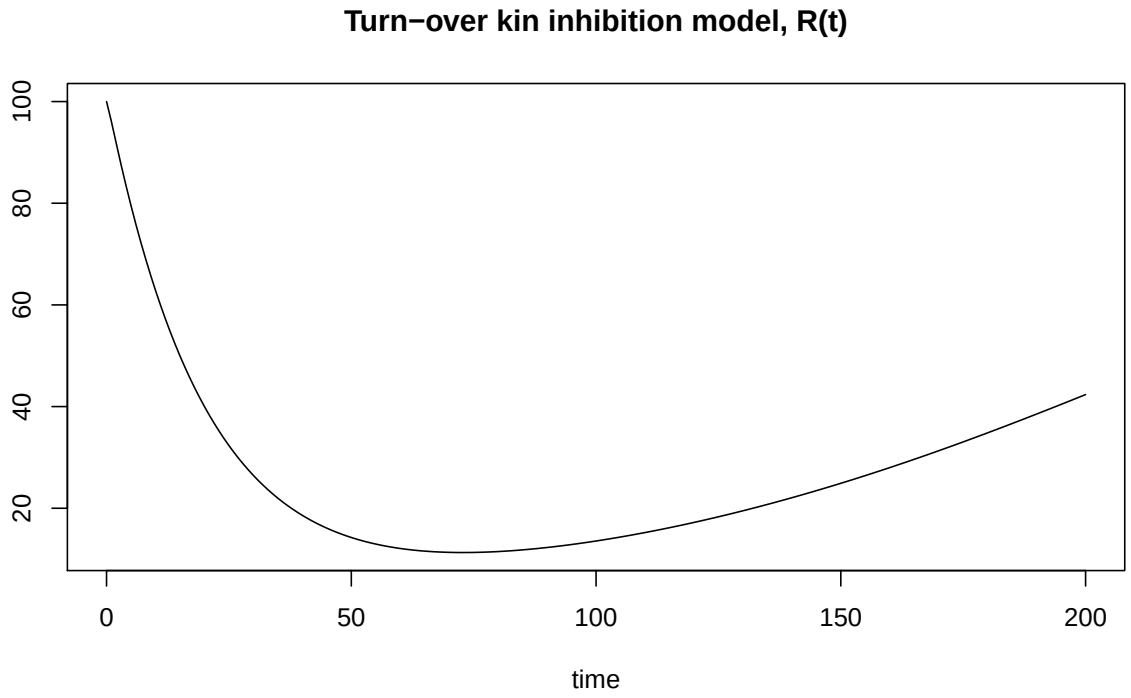
It is a good practice to visualize the ODE solution with typical values (for example those values used to define prior means) before using the ODE functor inside Stan. In R this can be achieved through the use of `deSolve` and `expose_stan_functions`:

```
library(deSolve)
## ensure we cache the binaries of the compiled functions
options(rcpp.cache.dir=".")
expose_stan_functions(pd_model1_gen)
```

```
## model function passed to deSolve
model <- function(time, y, theta) {
    ## use dR/dt ODE functor defined in Stan model
    dR <- turnover_kin_inhib_1(time, y[1], theta, x_r, x_i)
    list(c(dR))
}

theta <- log(c(100, 20, 0.5))
x_r <- log(c(100, 0.7, 1.1, 0.13, 8))
x_i <- c(0)
initial <- c(R=100)
times <- seq(0, 200, by=1)

sim <- ode(y=initial, times=times, func=model, parms=theta)
plot(sim, main="Turn-over kin inhibition model, R(t)")
```



This shape of the function roughly resembles the raw data. However, the time to fit this relatively small data set is rather excessive:

	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	15.40	2.04	17.45

In words, it takes for only 32 patients about 15 minutes for 250 warmup and 250 sampling iterations on a modern machine used inside a high-performance cluster.

The above formulation of the problem tickles a few issues of the current Stan language which we can avoid. In the current Stan language all variable definitions inside a function are automatically considered as parameters if the output of the function is involving parameters of the model, which happens whenever any of the arguments of a function is a parameter. Recall, that the major numerical cost in any Stan program is the gradient calculation of the log-likelihood wrt to all parameters. Inspecting the ODE functor above reveals that all the parameters of the PK model are implicitly turned into parameters due to their declaration in a function which returns a result involving parameters. So, let's reformulate the ODE functor without these intermediate declarations as shown below:

```
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) };
}
```

Doing so leads to less readable code, but it does avoid that the PK model is evaluated with variables which are considered as parameters. In effect, the evaluation of the PK model is much cheaper within this ODE functor. Running now the same model with this modified ODE functor yields a more than doubling of the execution speed:

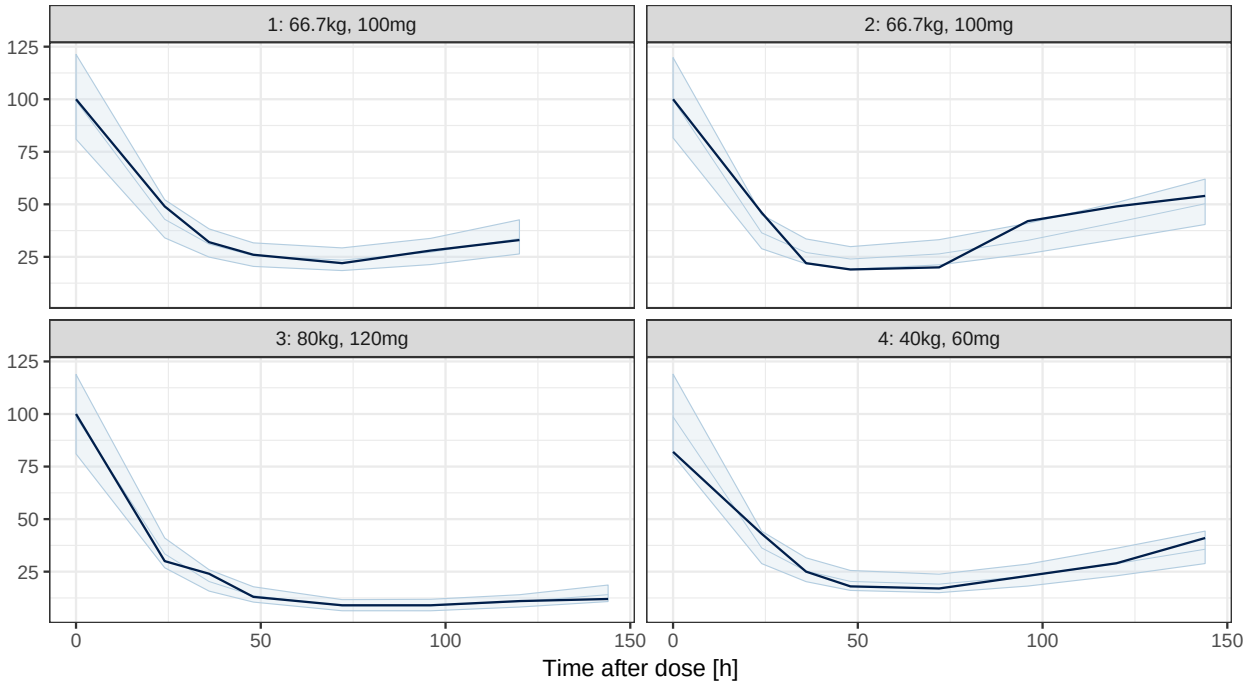
	warmup	sample	Sum
chain:1	6.10	0.96	7.06
chain:2	6.22	0.94	7.16
chain:3	6.17	0.94	7.11
chain:4	6.13	0.96	7.09

Fortunately, the Stan 3 language proposal currently worked on will allow to deliberately declare which variables are data for sure and which ones not. Thus, the rather cumbersome second version shouldn't be needed in the future with Stan 3.

Finally, here are the posterior predictive distributions for the first 4 patients in the data set and below the posterior predictive for the population with the raw data of all patients.

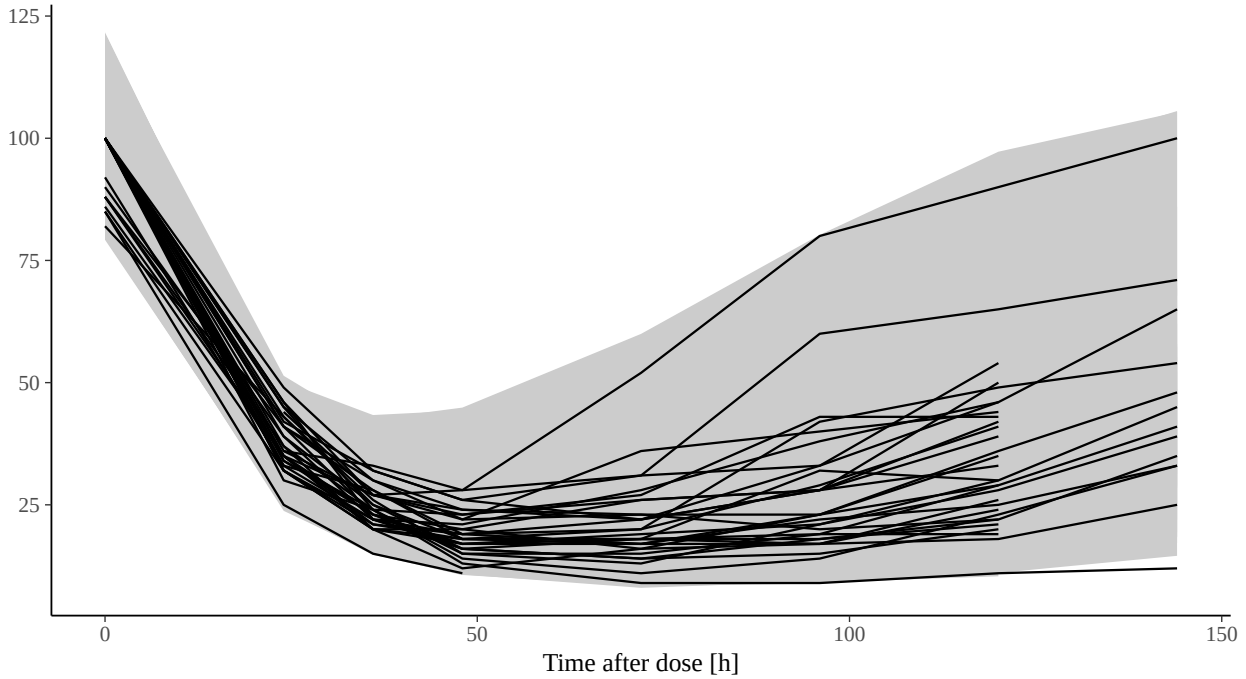
Percent Change Prothrombin Complex Levels vs Normal

Posterior predictive and data for 4 patients after 1.5mg/kg Warfarin oral dose



Percent Change Prothrombin Complex Levels vs Normal

Posterior predictive for population and raw patient datae after 1.5mg/kg Warfarin oral dose



Scalable Pharmacometrics

The example presented so far comprises a very small data set and used on of the simplest PD models. While the presentation so far is no over-simplification, the major issue is the inability to scale Stan's performance to larger data sets. However, pharmacometric models are by default hierarchical such that patients are modeled exchangeable to one another. Thus 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` facility in Stan. This function applies a user-defined defined function to a set of parameters which are in rectangular data storage format (hence the name of the function). Since each evaluation is independent of all others, these evaluations can be performed in parallel using either threading or the message passing interface (MPI). Threading has less operating system requirements in comparison to MPI, but is limited to the CPUs of the machine a given chain is run on while MPI can be used across multiple machines (typically part of a large high-performance cluster). The `map_rect` facility is used most efficiently if the function evaluated per subject (or whatever unit) directly returns the log-probability density contribution of the respective subject. This minimizes the communication between the different processes.

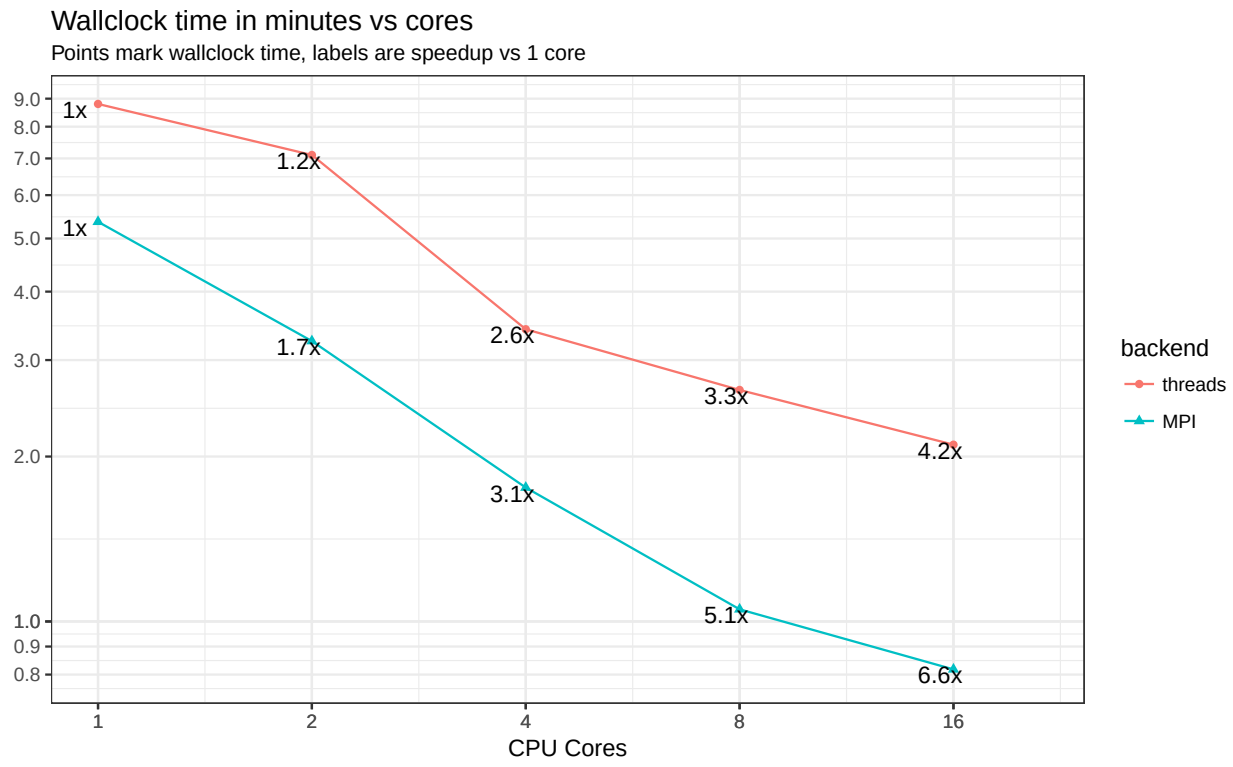
The function for the Warfarin example could look like this:

```
vector lpdf_subject(vector phi, vector eta, real[] x_r, int[] x_i) {
  vector[3] theta = phi[1:3];
  vector[3] sigma_eta = phi[4:6];
  real sigma_y = phi[7];
  real kappa = phi[8];
  real eta_cov[3] = to_array_1d(theta + eta .* sigma_eta);
  int start = x_i[1];
  int end = x_i[2]-1;
  int Ndv = x_i[3];
  int N = end-start+1;
  real mu_temp[N,1] = integrate_ode_rk45(turnover_kin_inhib_2,
                                         { exp(eta_cov[1]) }, -1E-4,
                                         x_r[7+start-1 : 7+end-1], // time
                                         eta_cov,
                                         x_r[1:5], // PK parameters
                                         x_i[1:0],
                                         1E-5, 1E-3, 500
                                         );
  vector[N] mu = to_vector(mu_temp[:,1]);
  return [ gamma2_overdisp_array_lpdf(x_r[7+Ndv+start-1 : 7+Ndv+end-1]| mu, sigma_y, kappa * x_r[6] )
}
```

In the `model` block this can then be called via:

```
  target += map_rect(lpdf_subject, phi, Eta, x_r, x_i);
```

The speedup for this specific example is shown below when using threading or MPI on a machine with upto 16 cores.



As can be seen, the total wallclock time can be reduced to less than 1 minute when using 16 cores. Moreover, the MPI backend outperforms the threading backend considerably for a single core run and in terms of better scalability. Currently, the thread implementation in Stan relies on C++11 programming language standards. These do defer many implementation details to the compiler used. This will hopefully leave some room for future improvements of the code. From these experiments MPI should be preferred over threading despite the larger burden to setup MPI in comparison to threading.

What is important to emphasize is that the `map_rect` approach enables to **scale** the performance of a given Stan program and data set. This scalability allows to adapt performance needs in such a way that we get reasonable run times needed for a productive, iterative explorative modeling process (assuming availability of sufficient hardware ressources). As a rule of thumb, hierarchical models with a large computation cost per unit, like ODE models, greatly benefit from the use of `map_rect`. For a large problem involving > 1300 patients speedups of up to 61x on 80 cores haven been measured (2.5 days down to 1h computation time) (Weber 2018).

Conclusion

Stan has come a long way and has as of today all required components for application to realistic pharmacometric problems. These are numerically very challenging and have influenced the Stan development in some aspects considerably (non-stiff/stiff ODE solvers, matrix exponential and now within-chain parallelization). With the recent addition of within-chain parallelization Stan can finally be used as a practical tool for iterative modeling with realistic data-sets as used in pharmacometrics. The `map_rect` function is currently somewhat clunky to use due to the need for packing and unpacking of parameters and data, but the additional work is well offset by the massive performance gain for large problems whenever sufficient CPU cores are available to the Stan modeler. Once Stan 3 introduces closures for functions, the in-convenient packing/unpacking coding will hopefully be remedied to some extent.

References

Holford, Nicholas H.G. 1986. "Clinical Pharmacokinetics and Pharmacodynamics of Warfarin: Understanding the Dose-Effect Relationship." Springer International Publishing. doi:10.2165/00003088-198611060-00005.

O'Reilly, R A, P M Aggeler, and L S Leong. 1963. "Studies on the coumarin antocoagulant drugs: The pharmacodynamics of warfarin in man." *J. Clin. Invest.* 42 (10). American Society for Clinical Investigation: 1542–51. doi:10.1172/JCI104839.

O'Reilly, R A, and P M Aggeler. 1968. "Studies on coumarin anticoagulant drugs. Initiation of warfarin therapy without a loading dose." *Circulation* 38 (1). American Heart Association, Inc.: 169–77. doi:10.1161/01.CIR.38.1.169.

Weber, Sebastian. 2018. "Supporting drug development as a Bayesian in due time?!" In *PAGE Meet. 2018*. Montreux, Switzerland. https://www.page-meeting.org/default.asp?abstract=8735.