Supporting drug development as a Bayesian in due time?!

NOVARTIS

Sebastian Weber[1]

[1]Advanced Exploratory Analytics, Novartis Pharmaceuticals AG Basel, Switzerland

Background

Bayesian approaches can offer several interesting advantages for a pharmacometrician

- Formal framework to add external information through the use of priors
- Additional control over model fit in terms of regularization by weakly-informative priors which aids in numerical stability
- Has advantages in terms of propagating uncertainty through the analysis (avoids the need for bootstraps)
- Available open-source tools such as Stan [1] allow for a vast flexibility in terms of modeling choices which is relevant for specialty applications

... but there is no free lunch:

- Bayesian approaches oftentimes lead to excessive computation times which render the approach practically in-applicable for iterative modeling

Objectives

At the example of combining aggregate and individual patient data in the context of a non-linear population model the goal is to illustrate different strategies which turn running times of days into less than one hour.

The example chosen was key for a Novartis development program and did require the extension of an existing non-linear population model for 1300 patients to also include aggregate data from the literature, see [2].

The key steps taken to expedite the computations are

- Analytical shortcuts
- Approximation of the turn-over ordinary differential equation (ODE)
- Within-chain parallelization using the Message Passing Interface (MPI) in the upcoming Stan 2.18 version (machine local threads are an alternative)

Conclusion

- Major improvements in the execution times for ODE based pharmacometrics models through the use of multiple CPUs on a MPI cluster or via threads on a local machine
- The performance gain for ODE based models does scale linearly over multiple orders of magnitude with high efficiency – 61x speedup using 80 cores on 8 machines.
- Analytical shortcuts, which avoid the need for an ODE integrator, can significantly reduce computation times already. However, these require a high level of technical skills and need a case-by-case implementation.

Outlook:

- MPI and threading based within-chain parallelization are becoming available in the upcoming 2.18 version of Stan
- User-friendly pharmacometric Stan libraries like Torsten will make these function readily accessible.

Methods

Data

- Internal patient trial data included >1300 patients from 3 studies
- Monthly observation of pharmacodynamic effect
- External information in literature reported on 2400 patients which are enrolled into 2 studies with each 4 arms and on average 300 patients per arm.

Model

- Latent one-compartment model due to the unavailability of concentration measurements.
- Stimulation of kin turn-over model linking the latent pharmacokinetic model to the pharmacodynamic effect.
- Runtime of ODE model in Stan 2.5 days on full dataset

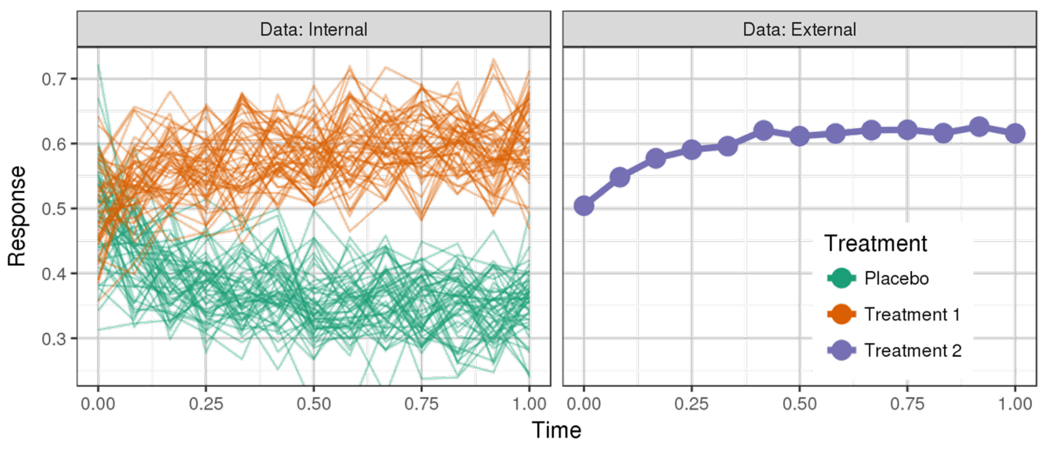
Bayesian Aggregation of Average Data (BAAD) [2]

- Expands an existing model $p(\phi|y)$ to include likelihood contribution of external summary data $p(\bar{y}'|\phi,\delta)$

$$p(\phi|y) \to p(\phi,\delta|y,\bar{y}') \propto p(\phi|y)\, p(\delta)\, p(\bar{y}'|\phi,\delta)$$

- Requires assumption that external data differs only in a few parameters, but many model parameters can be shared
- Approach is simulation based which increases numerical burden substantially (in addition to already 2.5 days runtime!)

Example simulation data set for BAAD



Basic Serial Stan Program

```
data {
  int<lower=1> N;
  real y[N,5];
}
parameters {
  real mu;
  real<lower=0> sigma;
  real<lower=0> omega;
  vector[N] theta;
}
model {
  // data likelihood
  for(i in 1:N)
    y[i] ~ normal(theta[i], sigma);

  // priors
  mu ~ normal(0, 10);
  omega ~ normal(0, 1);
  theta ~ normal(mu, omega);
  sigma ~ normal(0, 1);
}
```

$$y_i|\theta_i,\sigma \sim \text{Normal}(\theta_i,\sigma^2)$$

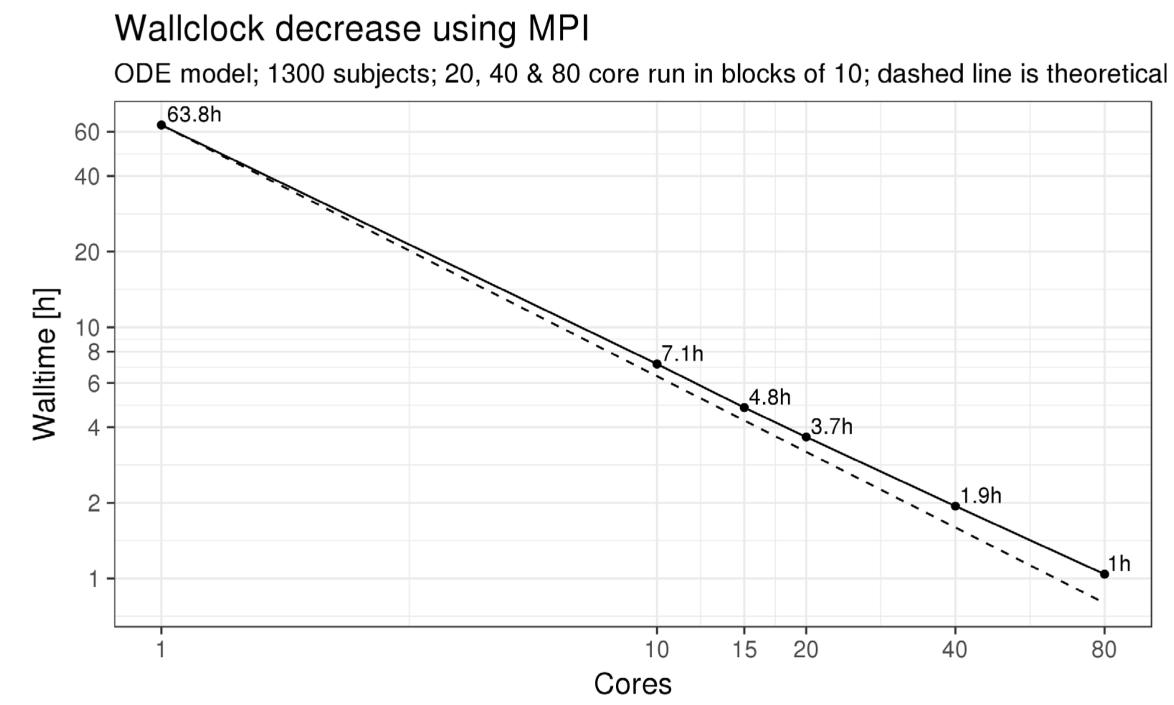
$$\theta_i|\mu,\omega \sim \text{Normal}(\mu,\omega^2)$$

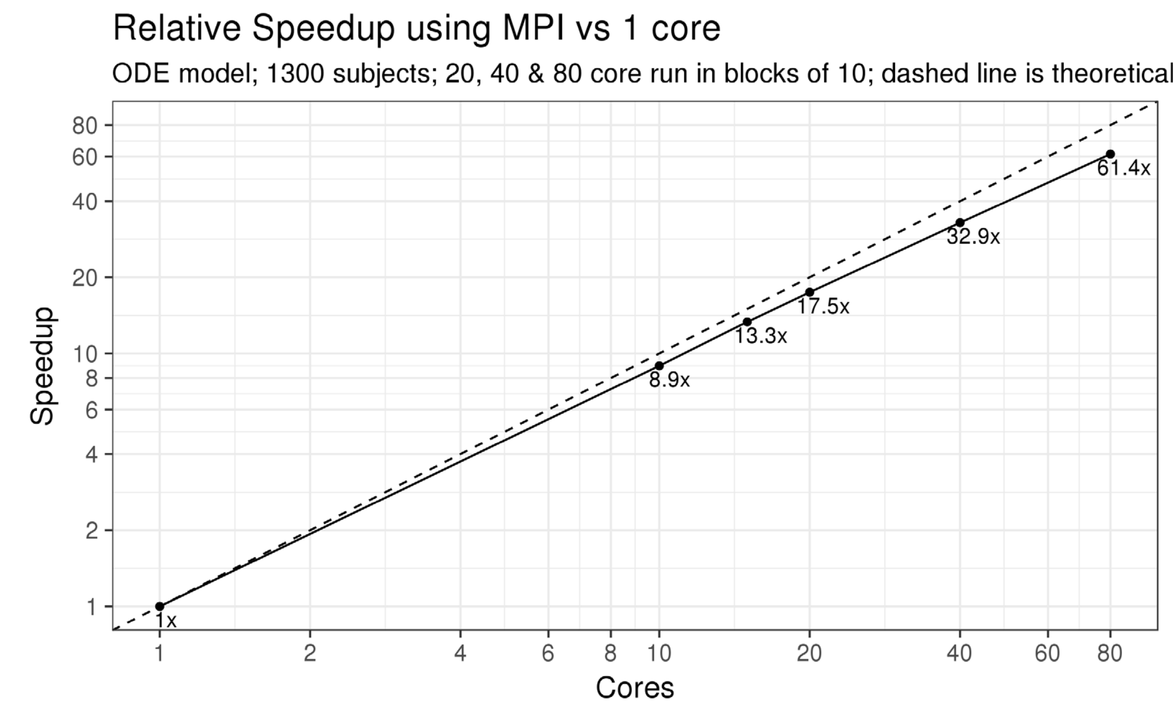
Parallelized Stan Program (using MPI or threading)

```
functions {
vector unit_logLik(vector shared,
                   vector unit_specific,
                   real[] y_data,
                   int[] data_int) {
  real mu = unit_specific[1];
  real sigma = shared[1];
  return [ normal_lpdf(y_data | mu, sigma) ]';
}
}
data {
  int<lower=1> N;
  real y[N,5];
}
transformed data {
  int x_i[N,0];
}
parameters {
  real mu;
  real<lower=0> sigma;
  real<lower=0> omega;
  vector[N] theta;
}
model {
  vector[1] shared = [ sigma ]';
  vector[1] unit[N];

  for(i in 1:N) unit[i,1] = theta[i];

  // data likelihood
  target += map_rect(unit_logLik, shared, unit, y, x_i);

  // priors
  mu ~ normal(0, 10);
  omega ~ normal(0, 1);
  theta ~ normal(mu, omega);
  sigma ~ normal(0, 1);
}
```

Results

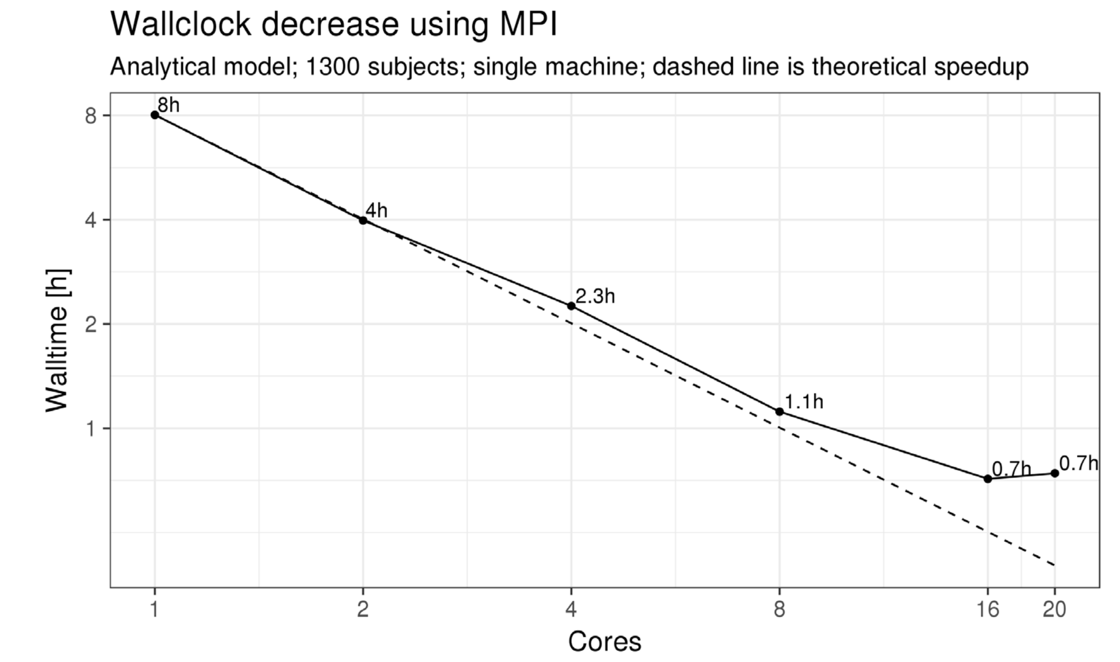
Reduction in runtime for ODE based model using MPI:





Analytic approximation of turn-over ODE

Due to a slow elimination half-life of 9days it is possible to approximate the pharmacokinetic concentration with a step-wise constant function. A step-wise constant input to a turn-over ODE can be solved (step-wise) using an analytic solution.



References: [1] Stan Development Team (2018). Stan: A C++ library for probability and sampling.

[2] Annals of Applied Statistics. 2017. Weber et al.

Poster presented at the Population Approach Group Europe (PAGE) Meeting, May 29th - June 1st, 2018, Montreux, Switzerland.