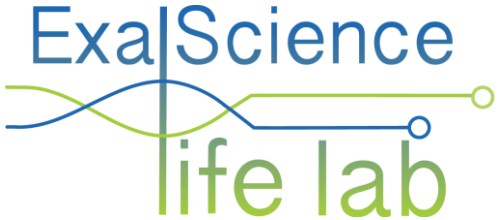
Hierarchical ODE Models Using Stan

20.05.2015

Bayes Pharma 2015

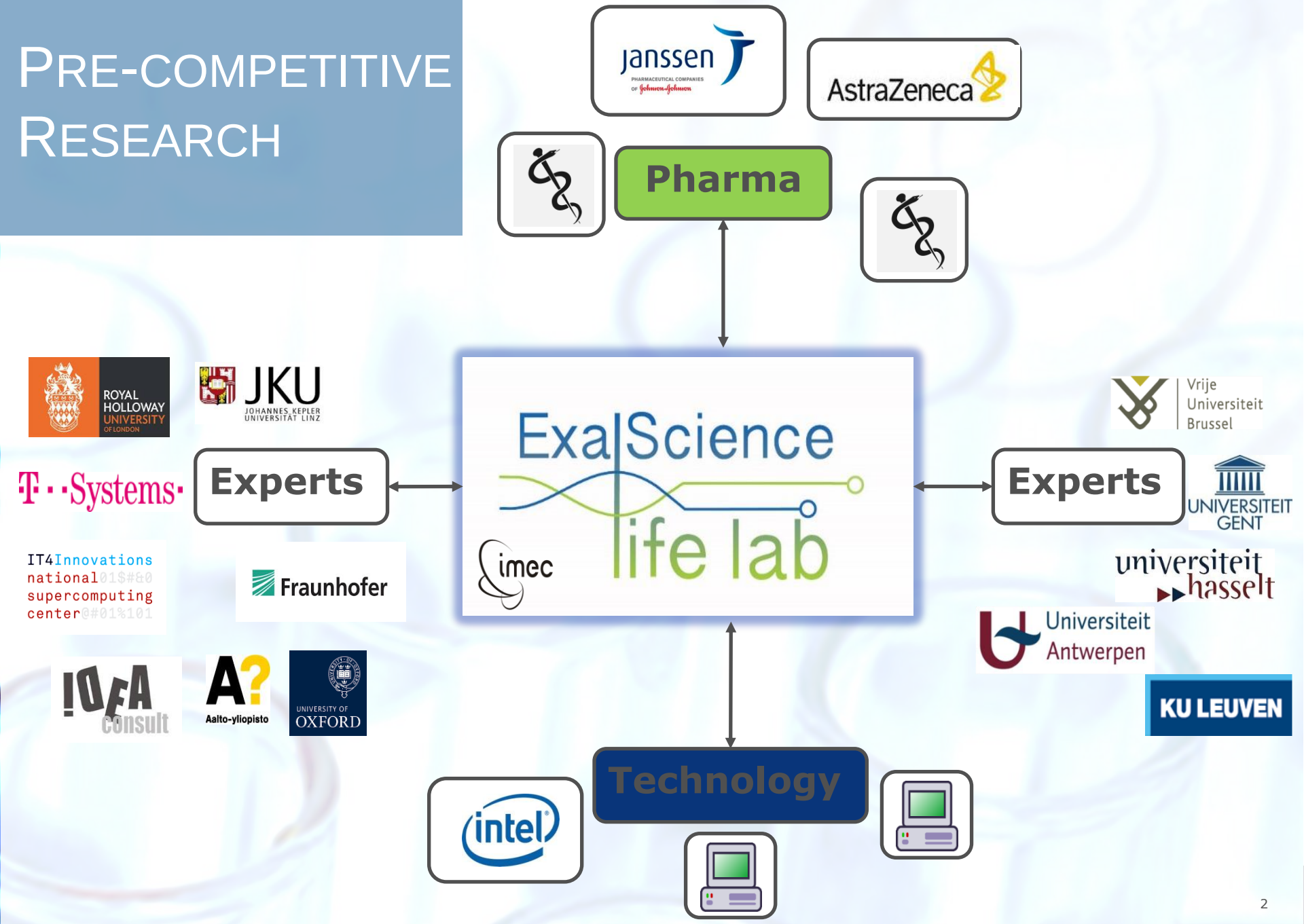
Jan Serroyen, Nele Goeyvaerts, Tom Haber



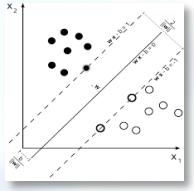
Overview

- ExaScience Life Lab
- Stan
- Application: PK-PD model
- Stan-CVODE
- Truncating the derivative

Pre-competitive Research



Opportunities for Computing



Better simulations and statistics

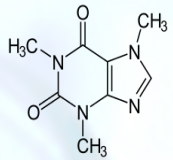
- E.g. Biostatistics, systems biology, PBPK, molecular dynamics

Better predictions

- E.g. Chemogenomics, toxicology

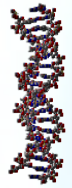
Analysis of very large amounts of data

- E.g. Genomics, large compound libraries, sensor data



Aims**:**

- Improve **disease knowledge** and **effectiveness** of medicines
  - Pharma provide the *challenges* and the *data*
- Leverage **broad computing expertise** to do this
- **Pool resources** to improve capabilities



What is Stan?

- "Probabilistic programming language implementing full Bayesian statistical inference"
  - MCMC sampling (Hamiltonian MC, NUTS)
  - Maximum likelihood estimation (BFGS)
- Coded in C++ and runs on all major platforms
- Open-source software (+ maintained): http://mc-stan.org/
- Standalone software, or interfaces with R, Python, Matlab, Julia
- HMC uses gradient information → less affected by correlations between parameters than random walk MC

Application: Pre-Clinical PK-PD Experiment

- Screening experiment on enzyme blocker
- Data:
  - 12 animals
  - 3 dose groups (oral)
  - 7 PK measurements over time per animal (drug concentration)
  - 3 PD measurements over time per animal (enzyme concentration)
- PK-PD model:
  - Pharmacokinetics: first-order one-compartment model
  - Pharmacodynamic: turnover model

Pharmacokinetic Model

First-order one-compartment model:

$$\begin{cases} \dfrac{dC_g}{dt} = -k_a C_g \\ \dfrac{dC_p}{dt} = (k_a C_g - k_e C_p)/V_f \end{cases}$$

Where

- C_g is the drug concentration in the GI tract (latent)
- C_p is the drug concentration in the blood plasma (observed)
- k_a is the absorption rate
- k_e is the elimination rate
- V_f is the (apparent) volume of distribution
- Initial conditions (t_0): $C_g(0)$ = dose, $C_p(0) = 0$

Pharmacodynamic Model

Turnover model:

$$\frac{dR}{dt} = k_{in}\left(1 - \frac{I_{max}C_p}{IC_{50}C_p}\right) - k_{out}R$$

Where

- R is the enzyme concentration (observed)
- k_{in} is the production rate
- k_{out} is the elimination rate
- I_{max} is the maximal inhibition (fixed to 100%)
- C_p corresponds to the plasma concentration
- IC_{50} is the plasma concentration required to obtain 50% inhibition
- k_{in}/k_{out} is animal (species) dependent → historical data

Bayesian PK-PD Model

- Model parameters:
  - Fixed effects (6): muKa, muKe, muVf, muKin, muKout, muIC50
  - Random (subject-specific) effects (6x12)
  - Variance components (6+2): random effect variances + resid. error

- Priors:
  - Fixed effects: Normal(0, 100)
  - Random effects: Normal(Fixed, VarComp)
  - Variance components: Cauchy(0, 2.5)
  - Note: priors are automatically truncated by Stan based on user-specified bounds on parameters

First Attempts in Stan

- Problem: Stan seems to easily get stuck (not updating) when fitting ODE models
- Attempted solutions that didn't work:
  - Specify "well-chosen" starting values for model parameters
  - Put upper bounds on model parameters (besides lower bounds)
  - Modify NUTS tuning parameter settings
- Underlying issue: no stopping criterion for step-halving

  → current ODE solver (Boost) is unstable

Contributions from ExaScience Lab

- More complex models
- Bug fixes:
  - Memory leak (later incorporated into Stan 2.6)
  - Initial condition ODE (t0): removed restriction (timepoints ≠ t0)
- Implemented better ODE solver: **CVODE** (Sundials)
  - Currently in Stan: only Runge-Kutta (simple/non-stiff)
  - CVODE: can deal with difficult (stiff/unstable) models
  - Jacobian: built using the auto-diff system of Stan
- Stan development team (Daniel Lee) is currently looking at Stan-CVODE implementation

Results and Impressions

- Correlations induced by hierarchical setting, combined with
- Non-linearity of ODE system:
  - Difficult to navigate parameter space

    → HMC: small step sizes & many leapfrog steps (e.g. 1023 steps)

    → Many calls to ODE solver per iteration
  - Long warmup phase needed

- Elapsed time :

  ~4 hours for 650 iterations (including 150 warm-up iterations)

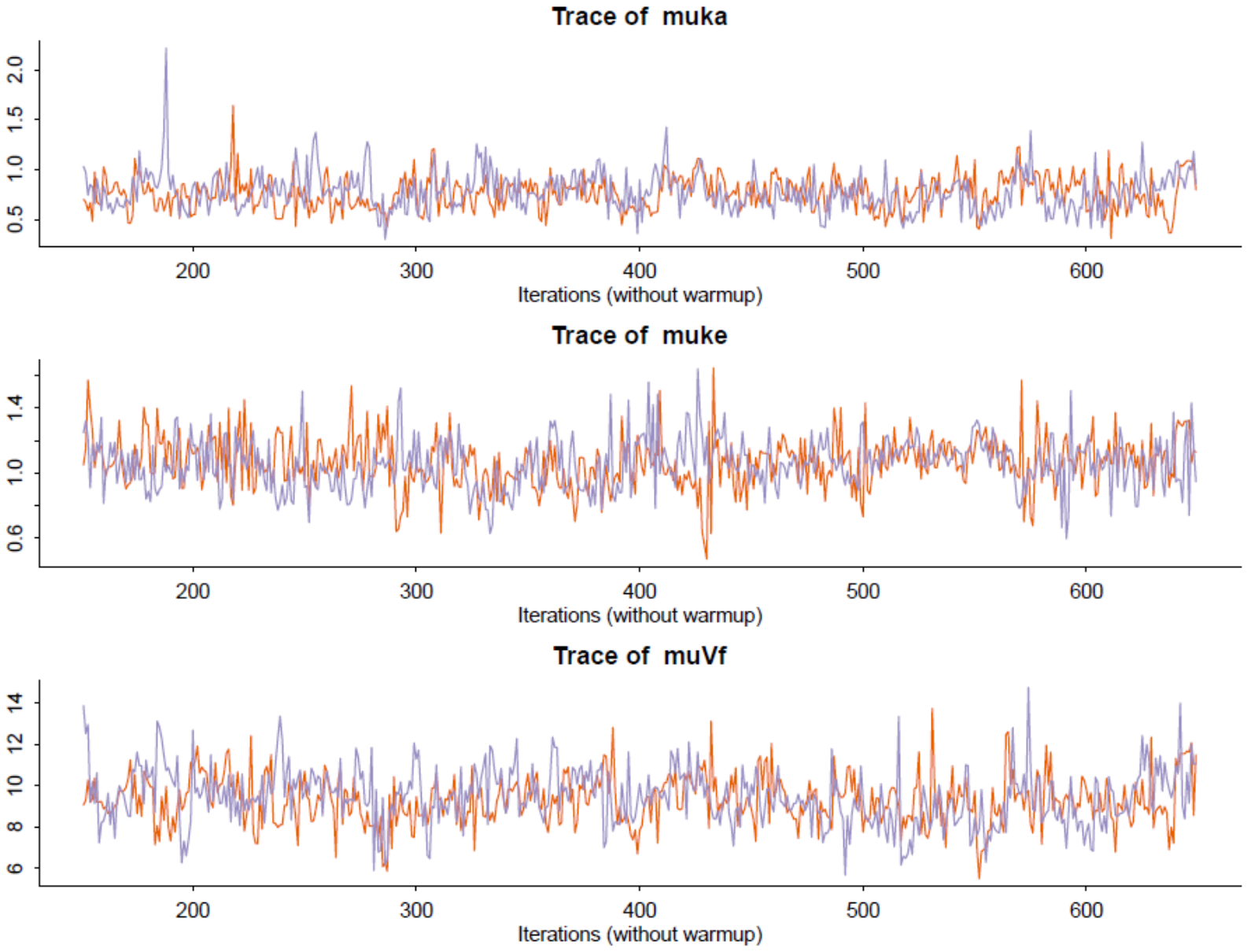
  Limited options for speedup: parallel chains

Posterior: Summary Statistics

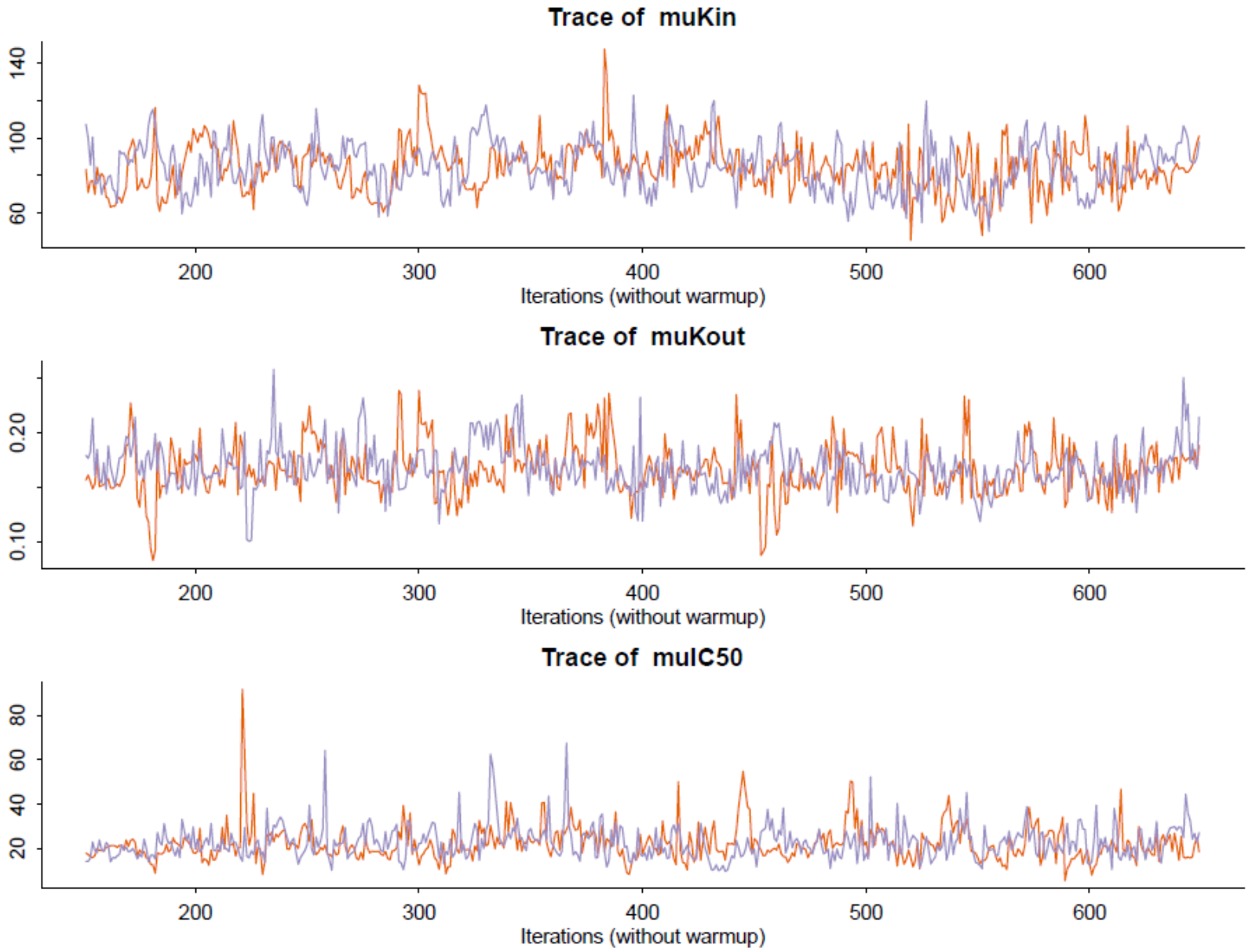
```
Inference for Stan model: stanPKPD.
2 chains, each with iter=650; warmup=150; thin=1;
post-warmup draws per chain=500, total post-warmup draws=1000.
```

	mean	se_mean	n_eff	Rhat	
muKa	0.770	0.012	241.090	1.012	Fixed effects
muKe	1.061	0.010	228.917	1.014	
muVf	9.374	0.089	213.036	1.017	
muKin	84.419	1.022	158.511	1.009	
muKout	0.167	0.001	225.177	1.010	
muIC50	22.575	0.474	263.675	1.000	
sigmaEps[1]	67.044	0.188	1000.000	1.001	Variance components
sigmaEps[2]	2.331	0.040	198.920	1.007	
sigmaKa	0.334	0.010	232.765	1.000	
sigmaKe	0.307	0.025	43.428	1.034	
sigmaVf	2.530	0.076	218.520	1.009	
sigmaKin	13.444	2.789	21.563	1.109	
sigmaKout	0.049	0.002	144.723	1.005	
sigmaIC50	11.457	0.706	160.463	1.000	

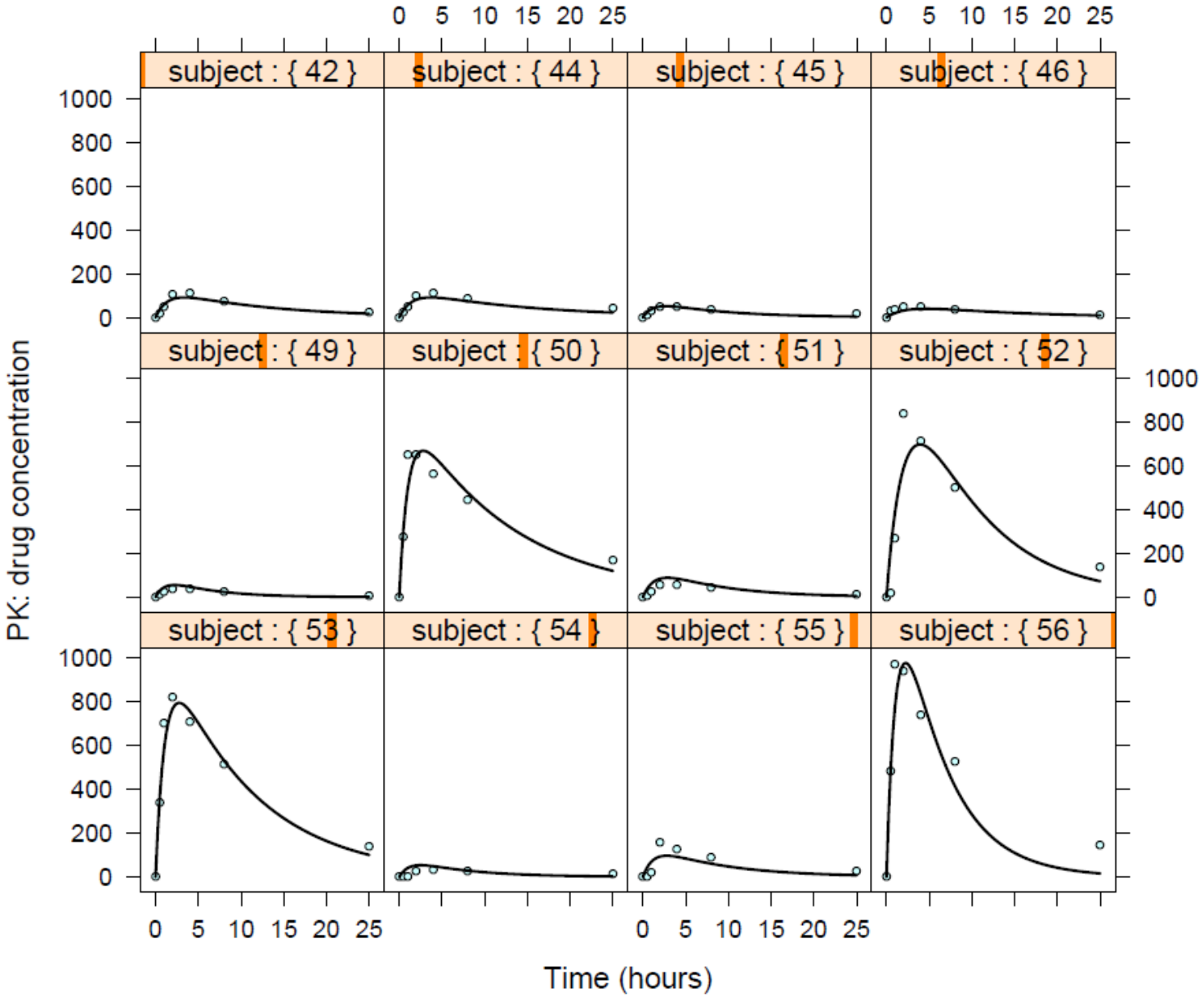
Trace Plots: PK model (fixed effects)



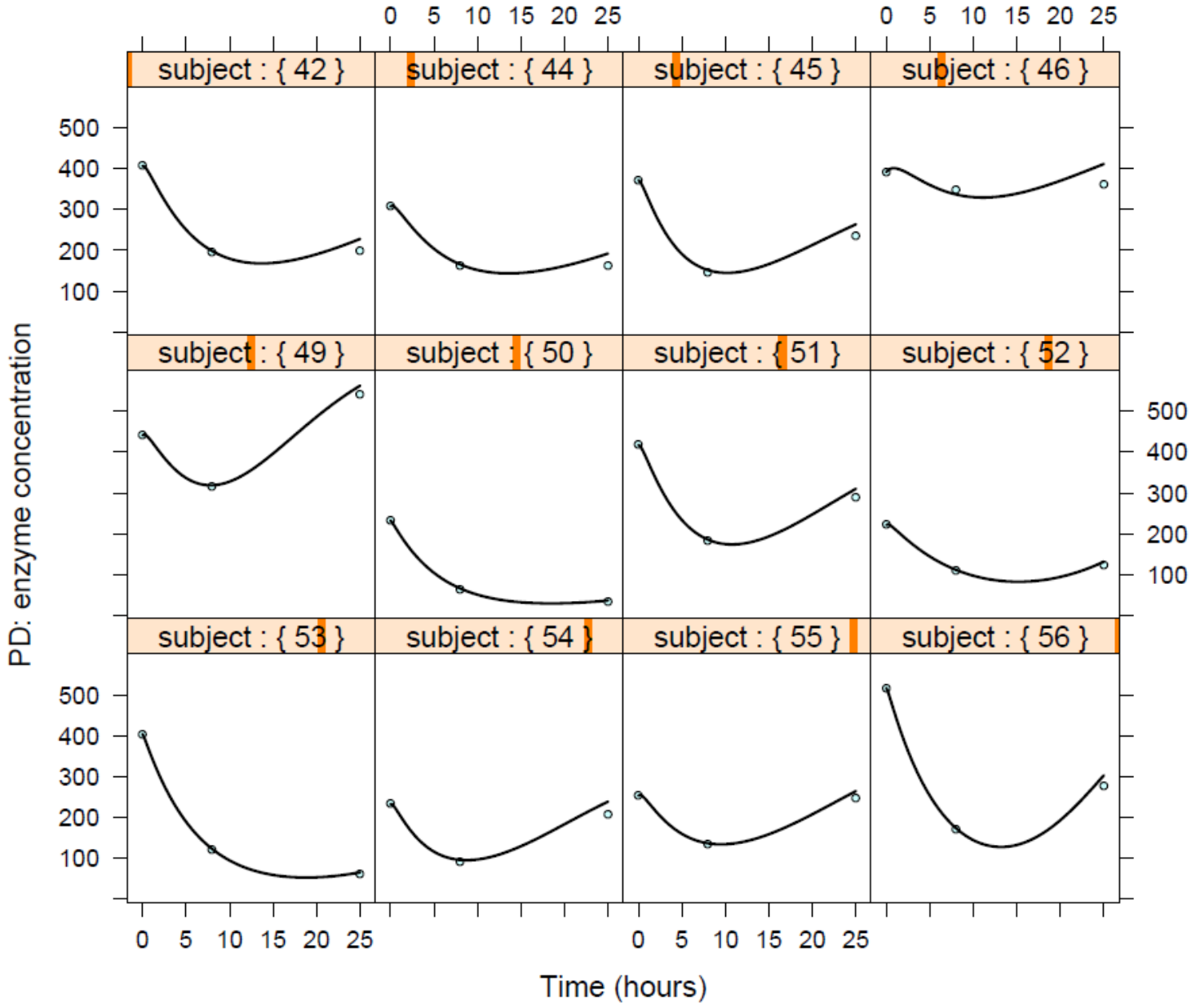
Trace Plots: PD model (fixed effects)



PK Model: Fitted Curves



PD Model: Fitted Curves



Installation Script RStan-CVODE (Linux)

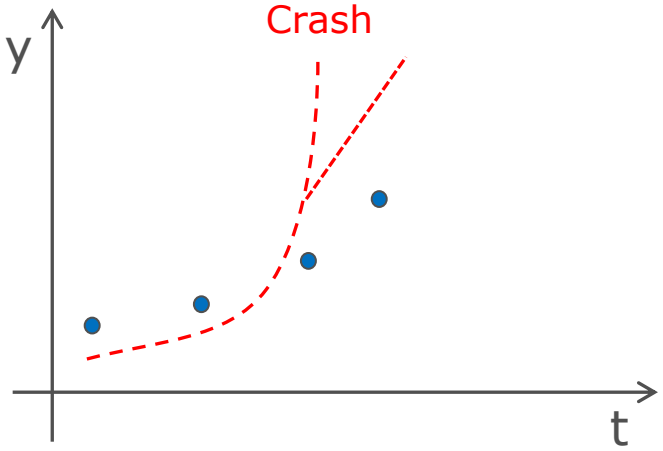
git clone https://github.com/stan-dev/rstan.git
cd rstan
git checkout -b cvode_rstan 7d6bf44c5b45b061e8e0f0b32f0a81050f37d4fb
git submodule update --init
cd stan
git remote add tomh https://github.com/tomhaber/stan.git
git fetch tomh
git checkout cvode
cd ../rstan
Make

# Install Sundials library (Ubuntu/Debian)
sudo apt-get install libsundials-serial libsundials-serial-dev

# Start RStudio with CVODE preloaded
LD_PRELOAD=/usr/lib/libsundials_cvode.so:/usr/lib/libsundials_nvecserial.so rstudio

Alternative Fix: Truncating the Derivative

- Aim: stabilize ODE solution
- dy* = min(max(dy, -a), a)
- Truncating ("clipping") is more straightforward than constraining all parameters individually



- Best used in combination with upper bounds on parameters

Alternative Fix: Truncating the Derivative

- Stan functions:

```
real clip(real dy, real a) {
    real dyclipped;
    dyclipped <- (fabs(dy + a) - fabs(dy - a)) / 2;
    return dyclipped;
    }

real[] ode(real t, real[] y, real[] theta, real[] x_r, int[] x_i) {
  real dydt[3];
  dydt[1] <- clip(-theta[1] * y[1], 10);
  dydt[2] <- clip((theta[1] * y[1] - theta[2] * y[2])/theta[3], 10) ;
  dydt[3] <- clip(theta[4] * (1 - y[2]/(theta[6] + y[2])) - theta[5] *
                  y[3], 10);
  return dydt;
  }
```

Summary and Conclusion

- Bayesian hierarchical ODE models were successfully fitted in Stan
- Default ODE solver is not stable

  → replaced by CVODE

  → Alternative: truncate the derivatives
- Our PK-PD model needs further refinement

  e.g. non-centered parametrization
- Checking quality and robustness of model fit for complex ODE models is not straightforward (and time consuming)

Acknowledgements

- Tom Jacobs, Janssen R&D
- Jeroen Tolboom, Janssen R&D

Thank you!