Stan Functions for Bayesian Pharmacometric Modeling

Charles C. Margossian and William R. Gillespie

Metrum Research Group, Tariffville, CT

METRUM RESEARCH GROUP

mc-stan.org

Introduction

Stan is a very flexible open source probabilistic programming language designed primarily to perform Bayesian data analysis [1]. The Stan No U-Turn Sampler (NUTS), an adaptive Hamiltonian Monte Carlo simulation algorithm, is more efficient than more commonly used MCMC samplers for complex high dimensional problems. Stan also includes a penalized maximum likelihood method. The primary objective of the presented work is to develop a set of new Stan functions we call the Torsten library to perform pharmacometric modeling tasks, including implementation of compartmental PKPD models and schedules of discrete events, e.g., dosing.

Stan and Torsten are open source projects.

Implemented PKPD Model Functions

- The current prototype Torsten library contains functions implementing:
  - Specific linear compartment models:
    - One compartment model with first order absorption
    - Two compartment model with elimination from and first order absorption into central compartment
  - Linear compartment model described by a system of first-order linear ODE's
  - General compartment model described by a system of first-order ODE's
  - Non-stiff ODE solver: Runge-Kutta 4th/5th order
  - Stiff ODE solver: Backward differentiation formula (CVODES BDF)
- All functions are integrated in the Stan software and can be used in a manner identical to built-in Stan functions
- The models and data format are based on NONMEM®/NMTRAN/PREDPP conventions including:
  - Recursive calculation of model predictions, which permits piecewise constant covariate values
  - Bolus or constant rate inputs into any compartment
  - Single dose, multiple doses, and steady state dosing histories
  - Implemented NMTRAN data items include: TIME, EVID, CMT, AMT, RATE, ADDL, II, SS

Implementation Details

- All functions are coded in C++ and directly included in Stan
- One and Two Compartment Models:
  - Analytical Solutions programmed in C++
- Linear Compartment Model
  - User provides constant rate matrix
  - Solves an ODE system by computing a Matrix Exponential, using a Pade approximation coupled with scaling and squaring [2]
- General Linear Compartment Model
  - User provides system of ODE's
  - User tunes the parameters of the ODE integrator
- The One, Two, and Linear compartment models handle steady state dosing history. The General model currently doesn't.

Simulated Examples

To describe and illustrate the use of Torsten, simulated results for the following scenario have been generated. The 3 different simulated outcomes (plasma drug concentrations and 2 different PD measurements) exercise the 3 methods Torsten implements to solve ODE's: analytical, semi-analytical, and numerical.

- Phase I study
  - Single dose and multiple doses
  - Parallel dose-escalation design
  - 8 subjects per dose arm
  - Single doses: 1.25, 5, 10, 20, and 40 mg
  - PK: plasma concentrations of parent drug (c)
  - PD: 2 different responses:
    - Response 1: Emax function of effect compartment concentration (R)
    - Response 2: Neutrophil count in circulation compartment (ANC) of Friberg-Karlsson semi-mechanistic model [3]
- Phase IIa trial in patients
  - Multiple doses: 20 mg for Response 1, 80000 mg for Response 2
  - 100 patients for Response 1, 5 patients for Response 2
  - Sparse PK and PD data (3-6 samples per patient) for Response 1
- The model used to fit the data is the same as the models used to simulate the data
- Priors:
  - Weakly informative for PK parameters and PD Response 1 (R)
  - Strongly informative for PD response 2 (ANC)

Model for plasma drug concentration (c)

$$\log(c_{ij}) \sim N\left(\log(\widehat{c}_{ij}), \sigma^2\right)$$

$$\widehat{c}_{ij} = f_{CptPar}(t_{ij}, D_j, \tau_j, CL_j, Q_j, V_{1j}, V_{2j}, k_{aj})$$

$$\log\left(CL_j, Q_j, V_{1j}, V_{2j}, k_{aj}\right) \sim N\left(\log\left(\widehat{CL}\left(\frac{bw_j}{70}\right)^{0.75}, \widehat{Q}\left(\frac{bw_j}{70}\right)^{0.75}, \widehat{V1}\left(\frac{bw_j}{70}\right), \widehat{V2}\left(\frac{bw_j}{70}\right), \widehat{k_a}\right), \Omega_{PK}\right)$$

$$\left(\widehat{CL}, \widehat{Q}, \widehat{V_1}, \widehat{V_2}, \widehat{k_a}\right) = \left(10\,\mathrm{L/h}, 15\,\mathrm{L/h}, 35\,\mathrm{L}, 105\,\mathrm{L}2\,\mathrm{h}^{-1}, 0.25\right)$$

$$\Omega_{PK} = \begin{pmatrix} 0.25^2 & 0 & 0 & 0 & 0 \\ 0 & 0.25^2 & 0 & 0 & 0 \\ 0 & 0 & 0.25^2 & 0 & 0 \\ 0 & 0 & 0 & 0.25^2 & 0 \\ 0 & 0 & 0 & 0 & 0.25^2 \end{pmatrix}, \quad \sigma = 0.1$$

Effect compartment model for PD response 1 (R)

$$R_{ij} \sim N\left(\widehat{R}_{ij}, \sigma_R^2\right)$$

$$\widehat{R}_{ij} = \frac{E_{max} c_{e\,ij}}{EC_{50\,j} + c_{e\,ij}}$$

$$c'_{e\,ij} = k_{e0j}\left(c_j - c_{e\,ij}\right)$$

$$\log\left(EC_{50\,j}, k_{e0\,j}\right) \sim N\left(\log\left(\widehat{EC}_{50}, \widehat{k}_{e0}\right), \Omega_R\right)$$

$$\left(E_{max}, \widehat{EC}_{50}, \widehat{k}_{e0}\right) = (100, 100.7, 1)$$

$$\Omega_R = \begin{pmatrix} 0.2^2 & 0 \\ 0 & 0.25^2 \end{pmatrix}, \quad \sigma_R = 10$$

Friberg-Karlsson Model for drug-induced myelosuppression, i.e. Response 2 (ANC)

$$\log(ANC_{ij}) \sim N(Circ_{ij}, \sigma_{ANC}^2)$$

$$\log\left(MTT_j, Circ_{0\,j}, \alpha_j\right) \sim N\left(\log\left(\widehat{MTT}, \widehat{Circ_0}, \widehat{\alpha}\right), \Omega_{ANC}\right)$$

$$\left(\widehat{MTT}, \widehat{Circ_0}, \widehat{\alpha}, \gamma\right) = (125, 5.2, 0.17)$$

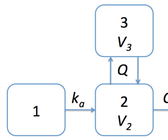
$$\Omega_{ANC} = \begin{pmatrix} 0.2^2 & 0 & 0 \\ 0 & 0.35^2 & 0 \\ 0 & 0 & 0.2^2 \end{pmatrix}, \quad \sigma_{ANC} = 0.1$$

$$\Omega_{PK} = \begin{pmatrix} 0.25^2 & 0 & a0 & 0 & 0 \\ 0 & 0.4^2 & 0 & 0 & 0 \\ 0 & 0 & 0.25^2 & 0 & 0 \\ 0 & 0 & 0 & 0.4^2 & 0 \\ 0 & 0 & 0 & 0 & 0.25^2 \end{pmatrix}$$

Example 1: Two compartment model with 1st order absorption

Stan language for fitting a two compartment model

```
transformed parameters {
  :
  for (i  in  1:nSubjects) {
    parms[1][1] = exp(logtheta[i,  1]);  # CL
    parms[1][2] = exp(logtheta[i,  2]);  # Q
    parms[1][3] = exp(logtheta[i,  3]);  # V1
    parms[1][4] = exp(logtheta[i,  4]);  # V2
    parms[1][5] = exp(logtheta[i,  5]);  # ka
    parms[1][6] = 1;  # F1
    parms[1][7] = 1;  # F2
    parms[1][8] = 1;  # F3
    parms[1][9] = 0;  # tlag1
    parms[1][10] = 0;  # tlag2
    parms[1][11] = 0;  # tlag3

    x[ start[i,j]:end[i,j]]  = PKModelTwoCpt(parms,
                                             time[ start[i,j]:end[i,j]],
                                             amt[start[i,j]:end[i,j]],
                                             rate[ start[i,j]:end[i,j]],
                                             ii [ start[i,j]:end[i,j]],
                                             evid[ start[i,j]:end[i,j]],
                                             cmt[start[i,j]:end[i,j]],
                                             addl[start[i,j]:end[i,j]],
                                             ss[start[i,j]:end[i,j]] ) ;

    cHat[start[i,j]:end[i,j]]  = col(x[ start[i,j]:end[i,j]],  2)  /  theta [1][3];
  }
}
  :
```

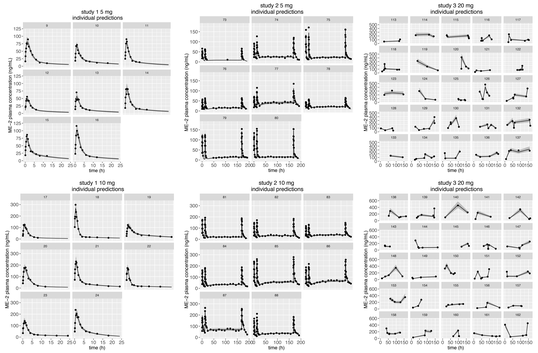


Priors (weakly informative)

$$\widehat{CL} \sim \text{half-}N(0, 20) \qquad \widehat{Q} \sim \text{half-}N(0, 20) \qquad \widehat{V1} \sim \text{half-}N(0, 100)$$

$$\widehat{V2} \sim \text{half-}N(0, 1000) \qquad \widehat{ka} \sim \text{half-}N(0, 5) \qquad \sigma \sim \text{half-cauchy}(0, 5)$$

$$\Omega = \text{diag}(\omega) P \text{diag}(\omega) \qquad \omega \sim \text{half-cauchy}(0, 2) \qquad P \sim \text{LKJCorr}(1)$$

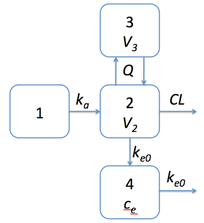


Example 1: Predicted (posterior median 90% credible intervals) and observed plasma drug concentrations for 5 and 10 mg dose regimes in Phase I study, and for the first 50 patients of Phase IIa study.

Example 2: General linear compartment model

Stan language for fitting an effect compartment model

```
transformed parameters {
  for (i  in  1:nSubjects){
    :
    ke0[i]  = exp(logKe0[i]);
    EC50[i]  = exp(logEC50[i]);

    k10 = CL[i]  /  V1[i];
    k12 = Q[i]  /  V1[i];
    k21 = Q[i]  /  V2[i];

    K = rep_matrix(0.  4.  4);

    K[1,  1]  = -ka[i];
    K[2,  1]  = ka[i];
    K[2,  2]  = -(k10 + k12);
    K[2,  3]  = k21;
    K[3,  2]  = k12;
    K[3,  3]  = -k21;
    K[4,  2]  = ke0[i];
    K[4,  4]  = -ke0[i];

    x[ start[i,j]:end[i,j]]  = linCptModel(K, parms,
                                           time[ start[i,j]:end[i,j]],
                                           amt[start[i,j]:end[i,j]],
                                           rate[ start[i,j]:end[i,j]],
                                           ii [ start[i,j]:end[i,j]],
                                           evid[ start[i,j]:end[i,j]],
                                           cmt[start[i,j]:end[i,j]],
                                           addl[start[i,j]:end[i,j]],
                                           ss[start[i,j]:end[i,j]] ) ;

    cHat[start[i,j]:end[i,j]]  = 1000 * x[start[i,j]:end[i,j],  2]  /  V1[i];
    ceHat[start[i,j]:end[i,j]]  = 1000 * x[start[i,j]:end[i,  4]  /  V1[i];
    respHat[start[i,j]:end[i,j]]  = 100 + ce/Hat[start[i,j]:end[i,j]]  /
        (EC50[i] + ceHat[start[i,j]:end[i,j]]) ;
  }

  for (i  in  1:nObs){
    cHatObs[i] = cHat[iObs[i]];
    respHatObs[i] = respHat[iObs[i]];
  }
}
  :
```

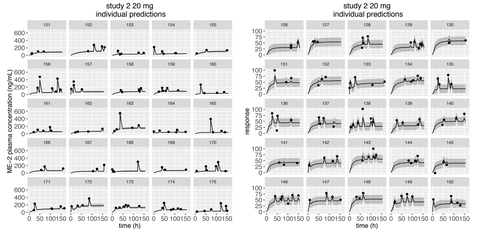


Priors (weakly informative)

$$\widehat{CL} \sim \text{half-}N(0, 20) \qquad \widehat{Q} \sim \text{half-}N(0, 40) \qquad \widehat{V1} \sim \text{half-}N(0, 150) \qquad \widehat{V2} \sim \text{half-}N(0, 150) \qquad \widehat{ka} \sim \text{half-}N(0, 5)$$

$$\widehat{ke_0} \sim \text{half-}N(0, 2) \qquad \widehat{EC_{50}} \sim \text{half-}N(0, 200) \qquad \sigma \sim \text{half-cauchy}(0, 2) \qquad \sigma_R \sim \text{half-cauchy}(0, 5) \qquad \omega_{k_{e0}} \sim \text{half-cauchy}(0, 2)$$

$$\omega_{EC_{50}} \sim \text{half-cauchy}(0, 5) \qquad \Omega = \text{diag}(\omega) P \text{diag}(\omega) \qquad \omega \sim \text{half-cauchy}(0, 2) \qquad P \sim \text{LKJCorr}(1)$$

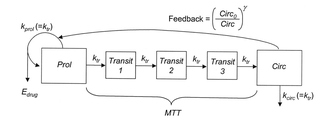


Example 2: Predicted (posterior median 90% credible intervals) and observed plasma drug concentrations and response 1 measurements for the first 50 patients.

Example 3: Friberg-Karlsson semi-mechanistic model

Stan language for coding an ODE system, with Figure for Friberg-Karlsson model [3]

```
functions {
  real []  twoCptNeutModelODE(real t, real[] x, real[]  parms,  real []  rdummy, int[]  idummy) {
    :
    k10 = CL / V1;
    k12 = Q / V1;
    k21 = Q / V2;
    ktr = 4 / mtt;

    dxdt[1]  = -ka * x[1];
    dxdt[2]  = ka * x[1]  -  (k10 + k12) * x[2]  + k21 * x[3];
    dxdt[3]  = k12 * x[2]  -  k21 * x[3];
    conc = x[2]/V1;

    EDrug = alpha + conc;
    prol  = x[4]  +  circ0;  //    x[4],  x[5],  x[6],  x[7]  and  x[8]  are  differences  from  circ0.
    transit1  = x[5]  +  circ0;
    transit2  = x[6]  +  circ0;
    transit3  = x[7]  +  circ0;
    circ  = fmax(machine_precision(),  x[8]  +  circ0);  // Device for implementing a modeled initial condition
    dxdt[4]  = ktr  *  prol  *  (1  -  EDrug)  *  (circ0  /  circ)^(gamma)  -  1);
    dxdt[5]  =  ktr  *  (prol  -  transit1 );
    dxdt[6]  =  ktr  *  (transit1  -  transit2);
    dxdt[7]  =  ktr  *  (transit2  -  transit3);
    dxdt[8]  =  ktr  *  (transit3  -  circ);

    return  dxdt;
  }
}
```



Stan language for coding a Friberg-Karlsson model

```
transformed parameters {
  :
  thetaM = (rep_matrix(thetaHat, nSubjects)  .*  exp(diag_pre_multiply(omega, L  *  etaStd)))';  ## use Matt's trick

  for(i  in  1:nSubjects) {
    parms[1][1]  =  thetaM[i,  1]  *  (weight[i ]  /  70)^0.75;  # CL

    x[ start[ i]:end[ i],] = generalCptModel_rk45(twoCptNeutModelODE, 8,
                                                  parms,
                                                  time[ start[ i]:end[ i]],
                                                  amt[start[ i]:end[ i]],
                                                  rate[ start[ i]:end[ i]],
                                                  ii [start[ i]:end[ i]],
                                                  evid[ start[ i]:end[ i]],
                                                  cmt[start[ i]:end[ i]],
                                                  addl[ start[ i]:end[ i]],
                                                  ss[ start[ i]:end[ i]],
                                                  1e-6, 1e-6, 1e6);

    cHat[start[ i]:end[ i]]  =  x[ start[ i]:end[ i],  2]  /  parms[1][3];  # divide by V1
    neutHat[start[ i]:end[ i]]  =  x[ start[ i]:end[ i],  8]  +  parms[1][7];  # Add baseline
  }

  for(i  in  1:nObsPK) cHatObs[i] = cHat[iObsPK[i]];
  for(i  in  1:nObsPD) neutHatObs[i] = neutHat[iObsPD[i]];
}
```

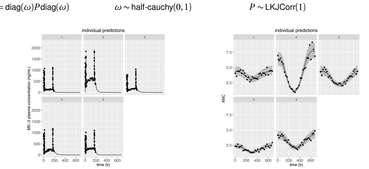
Priors (strongly informative)

$$\widehat{CL} \sim \text{logNormal}(\log(10), 0.10) \qquad \widehat{Q} \sim \text{logNormal}(\log(15), 0.18) \qquad \widehat{V1} \sim \text{logNormal}(\log(35), 0.14) \qquad \widehat{V2} \sim \text{logNormal}(\log(105), 0.17)$$

$$\widehat{ka} \sim \text{logNormal}(\log(2), 0.16) \qquad \widehat{MTT} \sim \text{logNormal}(\log(125), 0.2) \qquad \widehat{Circ_0} \sim \text{logNormal}(\log(5), 0.2) \qquad \widehat{\alpha} \sim \text{logNormal}(\log(2), 0.2)$$

$$\gamma \sim \text{logNormal}(log(0.17), 0.2) \qquad \sigma_{ANC} \sim \text{half-cauchy}(0, 1) \qquad \sigma_{ANC} \sim \text{half-cauchy}(0, 1)$$

$$\Omega = \text{diag}(\omega) P \text{diag}(\omega) \qquad \omega \sim \text{half-cauchy}(0, 1) \qquad P \sim \text{LKJCorr}(1)$$



Example 3: Predicted (posterior median 90% credible intervals) and observed plasma drug concentration and neutrophil count (ANC).

Development Plans

Current plans for further development of the Torsten PKPD library include:

- Extension of the linCptModel function to handle piece-wise constant parameters (parameters are currently required to be constant)
- Extension of the generalCptModel_rk45 and generalCptModel_bdf functions to handle steady-states
- Numerical solver of nonlinear algebraic equations (aka root solver)
- Optimized Matrix Exponential functions (particularly its action on a vector)

Distribution

- Torsten is an active, open source project of Metrum Research Group (`metrumrg.com`).
- Torsten may be downloaded free of charge. See `https://github.com/charlesm93/example-models/blob/feature/issue-70-PKPDexamples-torsten/PKPD/torsten/README.md`.
- **Warning:** The current version of Torsten is a *prototype*. It is being released for review and comment, and to support limited research applications. It has not been rigorously tested and should not be used for critical applications without further testing or cross-checking by comparison with other methods.
- Stan is an active, open source project of the Stan group (`mc-stan.org`).

References

[1] Carpenter, B., Gelman, A., Hoffman, M., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M.A., Guo, J., Li, P. and Riddel, A. Stan: A probabilistic programming language. *Journal of Statistical Software (in press)* (2016).

[2] Moler, C. and Van Loan, C. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review* (2003).

[3] Friberg, L.E. and Karlsson, M.O. Mechanistic models for myelosuppression. *Invest New Drugs* **21** (2003):183–194.

For more information, check out our user manual: