Prototype Model Library for Bayesian Pharmacokinetic/Pharmacodynamic (PKPD) Modeling in Winbugs

William R Gillespie and Marc R Gastonguay

Metrum Institute, Tariffville, CT 06081 USA

Background

- There is increasing interest in and use of Bayesian methods for PKPD modeling.
- A barrier to wider use of Bayesian PKPD modeling is the amount of custom programming required for each application.
- PKBugs and Pharmaco [1] partially address this problem by providing functions for linear 1, 2 and 3 compartment PK models and data management tools for WinBUGS [2],
- But they are limited to a small number of models and do not properly handle time-varying covariates.
- There is a need for a more comprehensive model library comparable in scope to NONMEM®/PREDPP [3].
- The combination of the BUGS language and a flexible PKPD model library would facilitate modeling and simulation tasks where the outcomes are causally related to drug exposure, e.g., models for clinical outcomes, adverse events, dropouts, and clinical trial simulations.

Objective

- To support fully Bayesian PKPD modeling by developing a collection of BUGS language functions that implement or facilitate user implementation of nonlinear-mixed effects population PKPD models.

Implemented PKPD Models

- The current prototype BUGS PKPD model library contains functions implementing:
  - Specific linear compartmental models:
    - One compartment model with first order absorption
    - Two compartment model with elimination from and first order absorption into central compartment
  - General linear compartment models described by a matrix exponential function
  - General compartment models described by a system of first order ODE's.
- The one and two compartment models are precompiled.
- General linear or nonlinear compartment models require user specification of a rate constant matrix or ODE's in a template Component Pascal procedure that must be compiled using the BlackBox Component Builder [4].
- The models and data format are based on NONMEM®/NMTRAN/PREDPP conventions including:
  - Recursive calculation of model predictions
    - This permits piecewise constant covariate values
  - Bolus or constant rate inputs into any compartment
  - Handles single dose, multiple dose and steady-state dosing histories
  - Implemented NMTRAN data items include:
    - TIME, EVID, CMT, AMT, RATE, ADDL, II, SS

Implementation Details

- Core model library procedures are programmed in Component Pascal and compiled with the BlackBox Component Builder.
- An extensible object-oriented programming approach is used to facilitate development of additional models.
- The BUGS language interface to WinBUGS 1.4.3 is implemented using WBDev [5].
- One and two compartment models:
  - Analytical calculations programmed in Component Pascal.
- General linear compartmental models:
  - Component Pascal wrapper procedures that call DLLs programmed in FORTRAN.
  - The DLLs are generated from Expokit [6] for matrix exponential calculations and LAPACK [7] for matrix calculations.
- General compartmental models:
  - Numerical solution of ODEs via Runge-Kutta 4th/5th order method.
  - Uses ODE solver code provided (but not documented) in WinBUGS.
  - Steady-state calculations based on solution of boundary value problem. Requires numerical solution of algebraic equations. Uses procedures from LibSolve [8].
  - Programmed in Component Pascal.

Simulated Examples

- Phase 1 study in healthy volunteers
  - Parallel dose-escalation design
  - 25 subjects per dose arm
  - Single doses
    - Placebo, 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_1)
    - Response 2: Indirect effect model with drug effect on kout (inhibitory Emax) (R_2)
  - PK and PD measured at 0, 0.125, 0.25, 0.5, 0.75, 1, 2, 3, 4, 6, 8, 12, 18 and 24 hours after dose.
- Phase IIa trial in patients
  - Multiple doses
    - 20 mg bid (q12h) x 7 days
  - 100 patients per treatment arm
  - Sparse PK data (3-6 samples/patient)

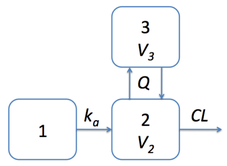
$$
\begin{aligned}
\log(c_{ij}) &\sim N(\log(\widehat{c}_{ij}),\sigma^2)\\
\widehat{c}_{ij} &= f_{2cpt}(t_{ij},D_j,\tau_j,CL_j,Q_j,V_{ij},V_{2j},k_{aj})\\
\log(CL_j,Q_j,V_{xxj},k_{aj}) &\sim \\
&N\left(\log\left(\widehat{CL}\left(\frac{bw_j}{70}\right)^{0.75},\widehat{Q}\left(\frac{bw_j}{70}\right)^{0.75},\widehat{V}_{xx}\left(\frac{bw_j}{70}\right),\widehat{k}_a\right),\Omega\right)\\
V_{ij} &= f_{V_1}V_{xxj}\quad V_{2j}=(1-f_{V_1})V_{xxj}\\
\left(\widehat{CL},\widehat{Q},\widehat{V}_{xx},\widehat{k}_a,f_{V_1}\right) &= (10\text{ L/h},15\text{ L/h},140\text{ L},2\text{ h}^{-1},0.25)\\
\Omega &= \begin{pmatrix}0.25^2&0&0&0\\0&0.25^2&0&0\\0&0&0.25^2&0\\0&0&0&0.25^2\end{pmatrix}\quad \sigma=0.1
\end{aligned}
$$

Model equations used to simulate the plasma drug concentrations (*c*) and the two PD responses (R_1 and R_2).

$$
\begin{aligned}
R_{1ij} &\sim N\left(\widehat{R}_{1ij},\sigma^2_{R_1}\right)\\
\widehat{R}_{1ij} &= \frac{E_{max}c_{ij}}{EC^{R_1}_{50j}+c_{ij}}\\
\dot{c}_{ij} &= k_{e0j}(c_j-c_{e,j})\\
\log(EC_{50j},k_{e0j}) &\sim N\left(\log\left(\widehat{EC}_{50},\widehat{k}_{e0}\right),\Omega_{R_1}\right)\\
\left(E_{max},\widehat{EC}_{50},\gamma,\widehat{k}_{e0}\right) &= (100,100.7,1.07,1)\\
\Omega_{R_1} &= \begin{pmatrix}0.2^2&0\\0&0.25^2\end{pmatrix}\quad \sigma_{R_1}=10
\end{aligned}
$$

$$
\begin{aligned}
\log(R_{2ij}) &\sim N\left(\log\left(\widehat{R}_{2ij}\right),\sigma^2_{R_2}\right)\\
\widehat{R}_{2,j} &= k_{out,j}\left(R_{20j}-\left(1-\frac{\widehat{c}_j}{EC^{R_2}_{50j}+\widehat{c}_j}\right)\widehat{R}_{2j}\right)\\
\log\left(R_{20j},EC^{R_2}_{50j},k_{out,j}\right) &\sim N\left(\log\left(\widehat{R}_{20},\widehat{EC}^{R_2}_{50},\widehat{k}_{out}\right),\Omega_{R_2}\right)\\
\left(\widehat{R}_{20},\widehat{EC}^{R_2}_{50},\widehat{k}_{out}\right) &= (100,200,0.5)\\
\Omega_{R_2} &= \begin{pmatrix}0.25^2&0&0\\0&0.25^2&0\\0&0&0.25^2\end{pmatrix}\quad \sigma_{R_2}=0.15
\end{aligned}
$$

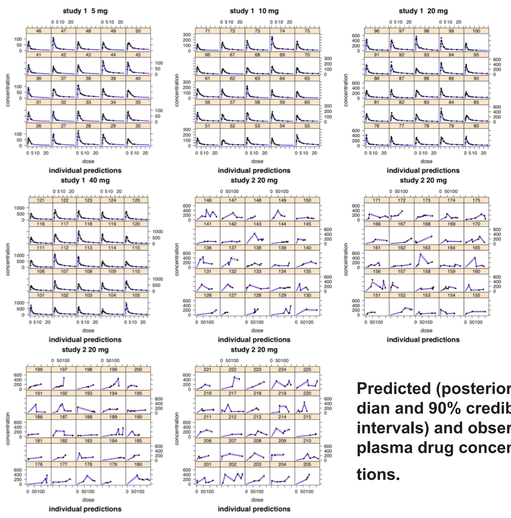
Example 1



Two compartment PK model with first order absorption

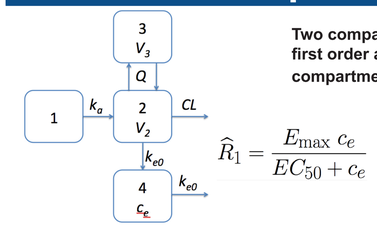
BUGS language model for fitting a two compartment model.

```
model
{

   for(i in 1:nsub){

      # Inter-patient variation
      logtheta[i, 1:6] ~ dmnorm(logthetaMean[i, 1:6], omega.inv[1:6, 1:6])
      logthetaMean[i, 1] <- logCLHat + 0.75*log(weight[start[i]]/70) # CL
      logthetaMean[i, 2] <- logQHat + 0.75*log(weight[start[i]]/70)  # Q
      logthetaMean[i, 3] <- logV1Hat + log(weight[start[i]]/70)          # V1
      logthetaMean[i, 4] <- logV2Hat + log(weight[start[i]]/70)          # V2
      logthetaMean[i, 5] <- logDkaHat                                    # ka - alpha1
      theta[i,6] <- 1 # F1
      theta[i,7] <- 1 # F2
      theta[i,8] <- 1 # F3
      theta[i,9] <- 0 # tlag1
      theta[i,10] <- 0 # tlag2
      theta[i,11] <- 0 # tlag3

      for(j in 1:5){
         log(theta[i,j]) <- logtheta[i,j]
      }

      # Call to PK model library to calculate amount in each compartment at each time
      xhat[start[i]:end[i],1:3] <- TwoCptModel(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]], theta[i,])

   }

   for(i in 1:nobs){

      logCobs[i] ~ dnorm(logCHat[i],tau)
      CHat[i] <- 1000*xhat[i,2]/theta[subject[i],3]
      logCHat[i] <- log(max(CHat[i],eps))

   }

   # Prior distributions
   logCLHat ~ dnorm(0,1.0E-6)
   logQHat ~ dnorm(0,1.0E-6)
   logV1Hat ~ dnorm(0,1.0E-6)
   logV2Hat ~ dnorm(0,1.0E-6)
   logDkaHat ~ dnorm(0,1.0E-6)
   log(CLHat) <- logCLHat
   log(QHat) <- logQHat
   log(V1Hat) <- logV1Hat
   log(V2Hat) <- logV2Hat
   log(DkaHat) <- logDkaHat
   tau <- 1/(sigma*sigma)
   sigma ~ dunif(0,1000)
   omega.inv[1:6, 1:6] ~ dwish(omega.inv.prior[1:6, 1:6], 6)
   omega[1:6, 1:6] <- inverse(omega.inv[1:6, 1:6])
   eps <- 1.0E-6

}
```



Predicted (posterior median and 90% credible intervals) and observed plasma drug concentrations.

Example 2



Two compartment PK model with first order absorption + effect compartment PKPD model

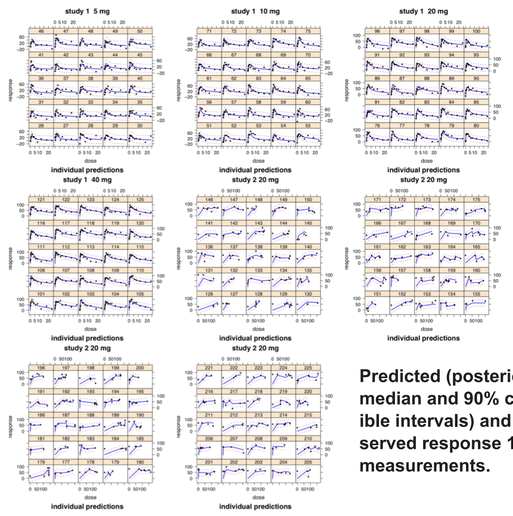
$$\widehat{R}_1 = \frac{E_{\max}\, c_e}{EC_{50} + c_e}$$

```
model
{

   for(i in 1:nsub){

      # Inter-patient variation
      logtheta[i, 1:8] ~ dmnorm(logthetaMean[i, 1:8], omega.inv[1:6, 1:6])
      logthetaMean[i, 1] <- logCLHat + 0.75*log(weight[start[i]]/70) # CL
      logthetaMean[i, 2] <- logQHat + 0.75*log(weight[start[i]]/70)  # Q
      logthetaMean[i, 3] <- logV1Hat + log(weight[start[i]]/70)          # V1
      logthetaMean[i, 4] <- logV2Hat + log(weight[start[i]]/70)          # V2
      logthetaMean[i, 5] <- logDkaHat                                    # ka - alpha1
      logtheta[i,6] ~ dnorm(logKeGHat,tauKeG)                            # keG
      theta[i,7] <- 1 # F1
      theta[i,8] <- 1 # F2
      theta[i,9] <- 1 # F3
      theta[i,10] <- 0 # tlag1
      theta[i,11] <- 0 # tlag2
      theta[i,12] <- 0 # tlag3

      logEC50[i] ~ dnorm(logEC50Hat,tauEC50)
      log(EC50[i]) <- logEC50[i]

      for(j in 1:6){
         log(theta[i,j]) <- logtheta[i,j]
      }

      # Call to PK model library to calculate amount in each compartment at each time
      xhat[start[i]:end[i],1:4] <- TwoCptEffFOpCModel(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]], theta[i,])

   }

   for(i in 1:nobs){

      logCobs[i] ~ dnorm(logCHat[i],tauC)
      CHat[i] <- 1000*xhat[i,2]/theta[subject[i],3]
      logCHat[i] <- log(max(CHat[i],eps))

      fxa[i] ~ dnorm(fxaMean[i],tauFxa)
      fxaMean[i] <- Emax*CeHat[i]/(EC50[subject[i]]+CeHat[i])
      CeHat[i] <- 100*xhat[i,4]/theta[subject[i],3]
      logCeHat[i] <- log(max(CeHat[i],eps))

   }

   # Prior distributions
   logCLHat ~ dnorm(0,1.0E-6)
   logQHat ~ dnorm(0,1.0E-6)
   logV1Hat ~ dnorm(0,1.0E-6)
   logV2Hat ~ dnorm(0,1.0E-6)
   logDkaHat ~ dnorm(0,1.0E-6)
   logKeGHat ~ dnorm(0,1.0E-6)
   logEC50Hat ~ dnorm(0,1.0E-6)
   log(CLHat) <- logCLHat
   log(QHat) <- logQHat
   log(V1Hat) <- logV1Hat
   log(V2Hat) <- logV2Hat
   log(DkaHat) <- logDkaHat
   log(KeGHat) <- logKeGHat
   log(EC50Hat) <- logEC50Hat
   tauC <- 1/(sigmaC*sigmaC)
   sigmaC ~ dunif(0,1000)
   tauFxa <- 1/(sigmaFxa*sigmaFxa)
   sigmaFxa ~ dunif(0,1000)
   omega.inv[1:6, 1:6] ~ dwish(omega.inv.prior[1:6, 1:6], 6)
   omega[1:6, 1:6] <- inverse(omega.inv[1:6, 1:6])
   omegaKeG ~ dunif(0,1000)
   tauKeG <- 1/(omegaKeG*omegaKeG)
   omegaEC50 ~ dunif(0,1000)
   tauEC50 <- 1/(omegaEC50*omegaEC50)
   eps <- 1.0E-6
   Emax <- 100

}
```

BUGS language model for fitting a two compartment PK model with first order absorption and an effect compartment PKPD model.

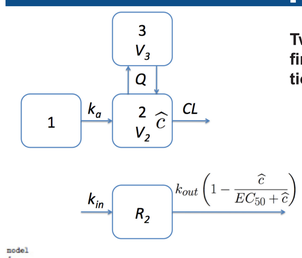
Specification of the matrix exponential form of the effect compartment model. This involves user specification of the non-zero elements of the rate constant matrix in the Component Pascal file TwoCptEffCptModel.odc.

```
PROCEDURE UserKMatrix(IN theta: ARRAY OF REAL; nCmt: INTEGER):
    POINTER TO ARRAY OF ARRAY OF REAL;
 VAR
    kMatrix: POINTER TO ARRAY OF ARRAY OF REAL;
    i, j: INTEGER;
    CL, Q, V2, V3, dka, ke0, k10, k12, k21, ksum, ka: REAL;
BEGIN
    NEW(kMatrix,nCmt,nCmt);
    (* Initialize to all zeros *)
    FOR i := 0 TO nCmt-1 DO;
    FOR j := 0 TO nCmt-1 DO;
       kMatrix[i,j] := 0;
    END; END;
    (* 2 cpt model with 1st order absorption *)
    CL := theta[0];
    Q := theta[1];
    V2 := theta[2];
    V3 := theta[3];
    dka := theta[4];
    ke0 := theta[5];
    k10 := CL/V2;
    k12 := Q/V2;
    k21 := Q/V3;
    ksum := k10 + k12 + k21;
    ka := dka + (ksum - Math.Sqrt(ksum*ksum-4.0*k10*k21))/2.0;
    (* Assign nonzero rate constants *)
    kMatrix[0,0] := -ka;
    kMatrix[1,0] := ka;
    kMatrix[1,1] := -(k10+k12);
    kMatrix[1,2] := k21;
    kMatrix[2,1] := k12;
    kMatrix[2,2] := -k21;
    kMatrix[3,1] := ke0;
    kMatrix[3,3] := -ke0;

    RETURN kMatrix;

 END UserKMatrix;

 PROCEDURE (m: MatExpModel) InitModel*;
 BEGIN
    m.nParameter := 12;
    m.f1Index := 6;
    m.tlag1Index := 10;
    m.nCmt := 4;
 END InitModel;
```



Predicted (posterior median and 90% credible intervals) and observed response 1 measurements.

Example 3



Two compartment PK model with first order absorption + indirect action PKPD model

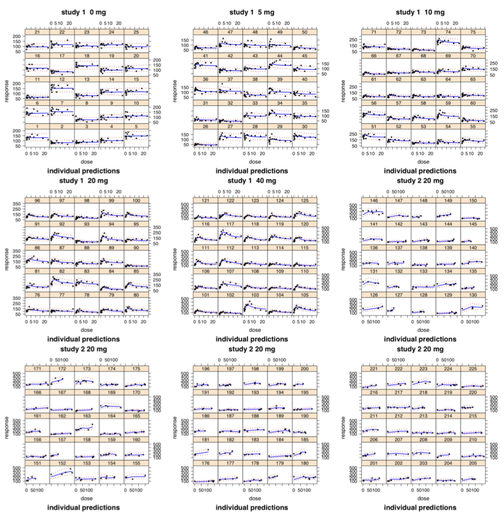
```
model
{

   for(i in 1:nsub){

    # Inter-patient variation
      logtheta[i, 1:5] ~ dmnorm(logthetaMean[i, 1:5], omega.inv[1:5, 1:5])
      logthetaMean[i, 1] <- logCLHat + 0.75*log(weight[start[i]]/70) # CL
      logthetaMean[i, 2] <- logQHat + 0.75*log(weight[start[i]]/70)  # Q
      logthetaMean[i, 3] <- logV1Hat + log(weight[start[i]]/70)          # V1
      logthetaMean[i, 4] <- logV2Hat + log(weight[start[i]]/70)          # V2
      logthetaMean[i, 5] <- logDkaHat                                    # ka - alpha1

      logtheta[i, 6:8] ~ dmnorm(logthetaMean[i, 6:8], omegaPD.inv[1:3, 1:3])
      logthetaMean[i, 6] <- logKoutHat                                   # kout
      logthetaMean[i, 7] <- logYeff0Hat                                  # yeff0
      logthetaMean[i, 8] <- logEC50Hat                                   # EC50

      theta[i,9] <- 1 # F1
      theta[i,10] <- 1 # F2
      theta[i,11] <- 1 # F3
      theta[i,12] <- 1 # F4
      theta[i,13] <- 0 # tlag1
      theta[i,14] <- 0 # tlag2
      theta[i,15] <- 0 # tlag3
      theta[i,16] <- 0 # tlag4

      for(j in 1:8){
         log(theta[i,j]) <- logtheta[i,j]
         log(thetaPred[i,j]) <- logthetaPred[i,j]
      }

      # Call to PK model library to calculate amount in each compartment at each time
      xhat[start[i]:end[i],1:4] <- TwoCptIndEff1Model(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]], theta[i,])

   }

   for(i in 1:nobs){

      logCobs[i] ~ dnorm(logCHat[i],tauC)
      CHat[i] <- 1000*xhat[i,2]/theta[subject[i],3]
      logCHat[i] <- log(max(CHat[i],eps))

      logYeff[i] ~ dnorm(logYeffHat[i],tauYeff)
      YeffHat[i] <- 1000*(xhat[i,4] + theta[subject[i],7])
      logYeffHat[i] <- log(max(YeffHat[i],eps))

   }

   logCLHat ~ dnorm(0,1.0E-6)
   logQHat ~ dnorm(0,1.0E-6)
   logV1Hat ~ dnorm(0,1.0E-6)
   logV2Hat ~ dnorm(0,1.0E-6)
   logDkaHat ~ dnorm(0,1.0E-6)
   log(CLHat) <- logCLHat
   log(QHat) <- logQHat
   log(V1Hat) <- logV1Hat
   log(V2Hat) <- logV2Hat
   log(DkaHat) <- logDkaHat

   logKoutHat ~ dnorm(0,1.0E-6)
   logYeff0Hat ~ dnorm(0,1.0E-6)
   logEC50Hat ~ dnorm(0,1.0E-6)
   log(koutHat) <- logKoutHat
   log(Yeff0Hat) <- logYeff0Hat
   log(EC50Hat) <- logEC50Hat
   tauC <- 1/(sigmaC*sigmaC)
   sigmaC ~ dunif(0,1000)
   tauYeff <- 1/(sigmaYeff*sigmaYeff)
   sigmaYeff ~ dunif(0,1000)
   omega.inv[1:5, 1:5] ~ dwish(omega.inv.prior[1:5, 1:5], 6)
   omega[1:5, 1:5] <- inverse(omega.inv[1:5, 1:5])
   omegaPD.inv[1:3, 1:3] ~ dwish(omegaPD.inv.prior[1:3, 1:3], 3)
   omegaPD[1:3, 1:3] <- inverse(omegaPD.inv[1:3, 1:3])
   eps <- 1.0E-6

}
```

BUGS language model for fitting a two compartment PK model with first order absorption and an indirect action PKPD model.

Specification of differential equations in the Component Pascal file: TwoCptIndEff1Model.odc.

```
PROCEDURE UserDerivatives(IN theta, x: ARRAY OF REAL;
numEq: INTEGER; t: REAL; OUT dxdt: ARRAY OF REAL) ;
VAR
CL, Q, V2, V3, dka, kout, yeff0, EC50, ka, k10, k12, k21, ksum, conc: REAL;
BEGIN
CL := theta[0];
Q := theta[1];
V2 := theta[2];
V3 := theta[3];
dka := theta[4];
kout := theta[5];
yeff0 := theta[6];
EC50 := theta[7];
k10 := CL/V2;
k12 := Q/V2;
k21 := Q/V3;
ksum := k10 + k12 + k21;
ka := dka + (ksum - Math.Sqrt(ksum*ksum-4.0*k10*k21))/2.0;

(* Differential equations for the model excluding piecewise *)
(* constant  input rates provided in the data set *)

dxdt[0] := -ka * x[0];
dxdt[1] := ka * x[0] - (k10 + k12) * x[1] + k21 * x[2];
dxdt[2] := k12 * x[1] - k21 * x[2];
conc := x[1]/V2;
dxdt[3] := kout * (yeff0 - (1 - conc/(EC50+conc))*(x[3]+yeff0))
(* x[3] = yeff - yeff0 *)

END UserDerivatives;
```

Predicted (posterior median and 90% credible intervals) and observed response 2 measurements.



Development Plans

- Revise prototype library using better programming practices to enhance extensibility, maintainability, reliability and efficiency. Specific changes include:
  - Use of factory design pattern.
  - Adaptive allocation of arrays to eliminate limits imposed by fixed size arrays while also avoiding inefficiencies due to overuse of dynamic allocation.
  - Implement additional ODE solvers, e.g., LSODA, CVODE, etc..
  - Improve method used for general compartmental model steady-state calculations.
- Implement additional models to be determined in collaboration with potential users.
- Develop documentation for both users and developers.
- Develop OpenBUGS interface for the model library.
- Execute rigorous and well-documented testing of the model library components.
- Conduct beta testing

BUGSModelLibrary Distribution

- BUGSModelLibrary is an active, open-source project of Metrum Institute (http://www.metruminstitute.org).
- BUGSModelLibrary may be downloaded from http://bugsmodellibrary.googlecode.com.

References

1. DJ Lunn, J Wakefield, A Thomas, N Best, D Spiegelhalter. PKBugs User Guide, Dept. Epidemiology & Public Health, Imperial College School of Medicine, London (1999). (http://www.winbugs-development.org.uk/pkbugs/home.html & http://www.winbugs-development.org.uk/pharmaco.html)
2. DJ Lunn, A Thomas, N Best, D Spiegelhalter. WinBUGS -- a Bayesian modelling framework: concepts, structure, and extensibility. Statistics and Computing, 10:325—337 (2000). (http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/contents.shtml)
3. SL Beal, LB Sheiner NONMEM users guide (1998) (http://www.icondevsolutions.com/nonmem.htm)
4. Oberon Microsystems. BlackBox Component Builder 1.5 Users Manual (2005) (http://www.oberon.ch/blackbox.html)
5. DJ Lunn. WinBUGS Development Interface (WBDev). ISBA Bulletin, 10(3): 10-11 (2003) (http://www.winbugs-development.org.uk/wbdev.html)
6. RB Sidje. Expokit: A Software Package for Computing Matrix Exponentials. ACM Trans. Math. Softw., 24(1):130-156, 1998 (http://www.maths.uq.edu.au/expokit/)
7. E Anderson, Z Bai, C Bischof, S Blackford, J Demmel, J Dongarra, J Du Croz, A Greenbaum, S Hammarling, A McKenney, D Sorensen. LAPACK Users' Guide. SIAM (1999) (http://www.netlib.org/lapack/)
8. RD Campbell. LibSolve (http://www.zinnamturm.eu/downloadsIN.htm#Lib)

 copies of this poster are available at www.metruminstitute.com/publications