Interactive Simulation and Visualization of Drug/Disease Models

Michael Heathman (1), Donald Jennings (1), and Bernett Lee (2)
(1) Eli Lilly and Company, Indianapolis, Indiana, USA; (2) Lilly Singapore Center for Drug Discovery, Republic of Singapore

Introduction

Drug/disease modeling enables quantitative decision making in the drug development process, through codification of scientific information about disease states, comparators, and new molecular entities. Model-based simulations are a powerful tool to leverage this information, facilitating open discussion with clinicians and other key members of the drug-development team, particularly when coupled with data visualization techniques.

Current mechanisms for generating and interpreting these simulations require significant expertise in multiple complex software applications. The ability to answer clinically-relevant questions is further impeded due to the long turnaround times required to program and run these simulations. This makes poor use of modeling resources and limits the total benefit gained from model development.

Objectives

To address these problems, we have undertaken the development of an interactive system for model-based simulation and visualization, including the following capabilities:

- A standardized model-specification language, compatible with mixed effect models developed in common software.
- An interface which allows user-defined input of model parameters, drug properties, patient characteristics, and study design parameters.
- A real-time simulation engine, designed to generate virtual patients and/or study data in a parallel fashion on a cluster of networked computers.
- An interface for visualization of simulation results, calculation of summary statistics, and output of virtual patients for further analysis in other statistical software.

Methods

Model Specification Language

- Standardized model specification language developed in R, for speed and portability.
- Model specification file is an annotated R script designed to generate a set of model predictions for a single patient, given a set of parameters, dosing regimen, observation timepoints, and covariate values.
- Header information defines inputs to model function.
- Model function returns endpoint predictions for the virtual patient.
- Template library available to assist users with model translation.

Example Model Specification File (Indirect Response Model)

```
# Author: Mike Heathman
# Created: June 2010
# Data source: Final data transfer, study ABCD
# Description: Two compartment PK, first-order absorption,
#              Type III indirect response model
#              Qualified visual predictive check
# Function: model
# Theta:  KA      1.25     (1/hour)
# Theta:  CL      111      (L/hour)
# Theta:  V2      1410     (L)
# Theta:  V3      1870     (L)
# Theta:  Q       167      (L/hour)
# Theta:  E0      99.7     (pM)
# Theta:  EMAX    0.870    (Fraction)
# Theta:  EC50    95.0     (ng/mL)
# Theta:  HILL    1.00     (Power)
# Theta:  KOUT    0.0993   (1/hour)
# Theta:  AGE_CL  -0.0215  (Fraction)
# Theta:  WT_V2   0.233    (Power)
# Omega:  CL    CL    0.0641
# Omega:  CL    V2    0.0571
# Omega:  V2    V2    0.142
# Omega:  CL    V3    0.0621
# Omega:  V2    V3    0.112
# Omega:  V3    V3    0.147
# Omega:  E0    E0    0.0111
# Omega:  EMAX  EMAX  0.0423
# Omega:  EC50  EC50  0.0455
# Omega:  KOUT  KOUT  0.0421
# Sigma:  PROP.PK  0.0392  (variance)
# Sigma:  PROP.PD  0.0100  (variance)
# Therapy:  drug 1 (1=LY)
# Therapy:  dose [50] (mg)
# Therapy:  dose.times [0] (hours)
# Measurement:  timepoints [0,1,2,3,4,5,6,7,8,9,10,11,12,
#               13,14,15,16,17,18,19,20,21,22,23,24] (hours)
# Covariate:  age 40 (years)
# Covariate:  weight 80 (kg)
# Endpoint:  PK   (ng/mL)
# Endpoint:  PD   (pM)
# Endpoint:  CFB  (pM)
# END
```

```
library("MASS")
library("deSolve")

#Differential Equations for indirect response model
model.DE <- function(t,state,parms)  {
  with(as.list(c(state,parms)),{
    da1 <- -ka*a1
    da2 <- ka*a1 - (cl/v2)*a2 - (q/v2)*a2 + (q/v3)*a3
    da3 <- (q/v2)*a2 - (q/v3)*a3
    da4 <- (kout*e0)*(1+emax*(a2/v2)^hill/((a2/v2)^hill+ec50^hill))-kout*a4
    list(c(da1,da2,da3,da4))})
  }

#model:  Function which is called by simulation system
model <- function(theta,omega,sigma,drug,dose,dose.times,
                  timepoints,age,weight)  {

  #check that omega matrix is positive definite
  if(min(eigen(omega)$values) < 0)  return(list(SUCCESS=F))

  #Calculate individual parameter values
  etas <- mvrnorm(1,mu=rep(0,length(theta)),Sigma=omega)
  parms <- list(ka=theta[1]*exp(etas[1]),
                cl=theta[2]*(1 + (age-40)*theta[11])*exp(etas[2]),
                v2=theta[3]*((weight/80)^theta[12])*exp(etas[3]),
                v3=theta[4]*exp(etas[4]),q=theta[5]*exp(etas[5]),
                e0=theta[6]*exp(etas[6]),
                emax=theta[7]*exp(etas[7]),ec50=theta[8]*exp(etas[8]),
                hill=theta[9]*exp(etas[9]),kout=theta[10]*exp(etas[10]))

  #generate predictions
  state <- c(a1=(dose*1000),a2=0,a3=0,a4=parms$e0)
  results <- vode(state,c(timepoints),model.DE,parms=parms)
  pk <- results[,"a2"]/parms$v2
  pd <- results[,"a4"]

  #apply residual error
  pk <- pk*(1 + rnorm(length(pk),mean=0,sd=sqrt(sigma[1])))
  pd <- pd*(1 + rnorm(length(pd),mean=0,sd=sqrt(sigma[2])))
  cfb <- pd - parms$e0

  return(list(PK=pk,PD=pd,CFB=cfb,SUCCESS=T))
  }
```

Simulation Interface

Graphical user interface for input of simulation specifications, model parameters, and covariate values.

Simulation Specifications

–Patient- vs. study-level simulation

–Number of replicates / patients

–Treatment regimens

–Simulation timepoints

Model Parameters

–Discrete values

–Parameter uncertainty sampled from

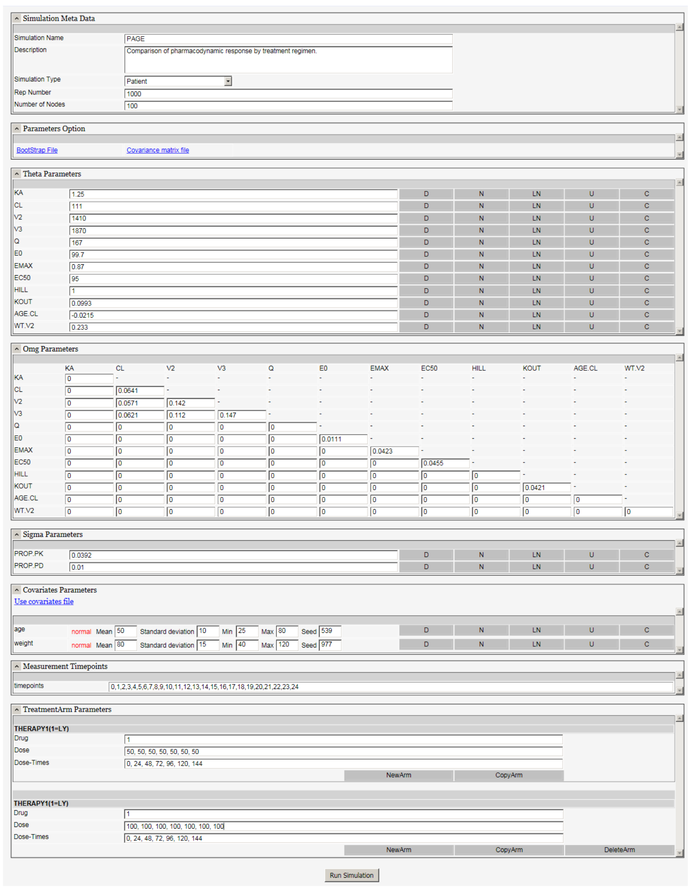
- normal, log-normal, uniform, or categorical distributions
- Bootstrap file (CSV)
- Multivariate normal distribution (covariance matrix)

Covariate Values

–Discrete values

–Normal, log-normal, uniform, or categorical distributions

–Sampled from external file of observed patient values

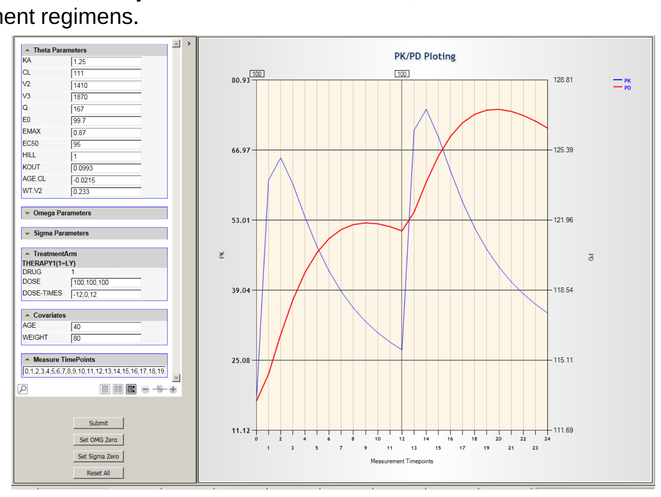


Simulation Engine

- Simulations distributed by application server using Condor, to run in parallel on 1136 node linux cluster.
- Results automatically aggregated by application server, patient-level data and mean values from study-level simulations.
- Aggregated files of virtual patients may be exported or viewed in visualization interface.

Visualization Interface

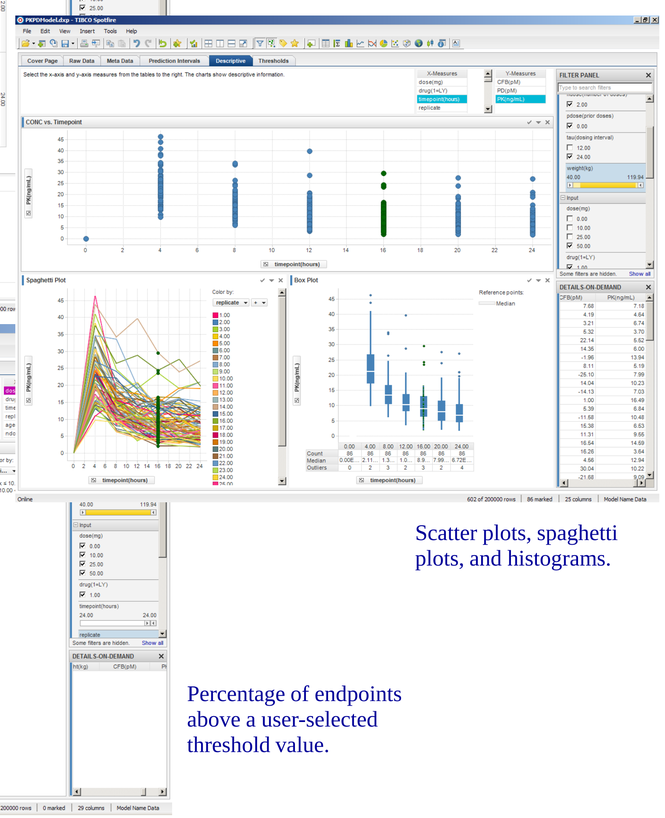
"Typical Response" may be viewed directly within simulation interface, with interactive control of model parameters, covariate values, and treatment regimens.



More extensive visualization environment developed using TICBO Spotfire®

- Automated import of simulation results
- Templates for standard views of simulations results
- Subset, filter, and trellis by parameter value, covariate or treatment regimen
- Webplayer version of software allows corporate-wide access via any web browser

Prediction intervals of simulated endpoints.



Scatter plots, spaghetti plots, and histograms.

Percentage of endpoints above a user-selected threshold value.

Results

–The system has substantially lowered the technical hurdles associated with running complex simulations to inform the drug development process.

–Large simulations can now be run in a matter of minutes, rather than hours or days.

–Interactive visualization of simulation results facilitates effective communication of model-derived information and projected outcomes in a team environment.

–The system provides a central repository for models and simulations, fostering information sharing and collaboration between scientists of different disciplines.

Conclusions

The system achieves two key objectives: (1) generation of simulation results to support real-time collaborative analysis, and (2) expansion of simulation capability to a broader non-technical audience for increased exploration of drug/disease models.

The resulting knowledge supports decision-making related to compound selection, dose selection, and study design optimization. The improvements in process gained from this initiative are essential for the integration of quantitative pharmacology into the drug development process.

Acknowledgments

Pei Yen Chia[1], Dong Liang Guo[1], Qing Yuan[1], Karunakaran Thagappillai[1], Umesh Manocha[1], Seth Williams[2], Andrew Kaczorek[3], and Sandy Allerheiligen[4]

[1]Lilly Singapore Center for Drug Discovery; [2]Tibco Software Inc.; [3]Eli Lilly and Company; [4]Merck, Inc.

Lilly
Answers That Matter.