Accessing PKtools a vignette

M. Suzette Blanchard

February 2, 2007

1 Introduction

PKtools is a package for analysis of single dose population PK data. It gives the analyst of population PK data access to frequentist and Bayesian estimation methods, by providing an interface to the well established software packages NLME (Pinheiro and Bates, 2000), NONMEM (Beal and Sheiner, 1989) and WinBUGS (Spiegelhalter et al., 2001). Additional tools are also supplied to simplify the implementation of comparative or hybrid analyses.

The core functions are RunNLME, RunNM, and RunWB, that interface to NLME, NONMEM and WinBUGS, respectively. To access one of the three 'Run' functions the user only needs to identify 1) a pharmacokinetic dataset, including; patient id, dose, draw times and concentrations, 2) the statistical model to be fit, and 3) the list of variable names. The RunWB program requires some additional WinBUGS arguments. RunNLME, RunNM, and RunWB, respectively output PKNLME, NONMEM, and WinBUGS objects which contain parameter estimates, residuals, and predicted values, for use in other PKtools functions, for further manipulation in R, and for export to other environments.

2 Implementing PKtools:RunNLME

As an example lets start by implementing RunNLME to analyze the Theophyline (Theoph) data provided in the nlme package (Pinheiro and Bates, 2000). We need to define a dataset that includes the pk data and covariates. In our example Theo is the pk dataset and wt.v defines the covariate data vector.

```
> library(PKtools)
> library(nlme)
> data(Theoph)
> Theoph <- Theoph[Theoph$Time != 0, ]
> id <- as.numeric(as.character(Theoph$Subject))
> dose <- Theoph$Dose
> time <- Theoph$Time
> conc <- round(sqrt(Theoph$conc), 4)
> Theo <- data.frame(cbind(id, dose, time, conc))
> names(Theo) <- c("id", "dose", "time", "conc")
> wt.v <- Theoph$Wt
> data <- list(pkvar = Theo, cov = wt.v)
```

nameData is a list of the labels including the names of the covariates in the order they are given in the covariate dataset, y (concentration) and x (time) variables, the random

parameters (reparams - should match the list for random.model in the model.def), fixed parameters (params - should match the list for fixed.model in the model.def), and the transformed parameters (in the Theo example the model parameters are on a log scale so tparam=c("log(Ka)","log(V)","log(Cl)"). Note the names of the variance parameters are not required for `RunNLME`.

```
> nameData <- list(covnames = c("wt"), yvarlab = "Sqrt(Theop. Conc.) (mg/L)",
+     xvarlab = "Time since dose (hrs)", reparams = c("Ka", "V",
+         "Cl"), params = c("Ka", "V", "Cl"), tparams = c("log(Ka)",
+         "log(V)", "log(CL)"))
```

The inputStructure for `RunNLME` is the model.def. The model.def lays out the model definition including the model form, fixed and random effects, the starting values and gives the user access to the control arguments for the nlme function. The following is an example. We are using a one compartment model with a square root transformation (sonecpmt) written with all three parameters on a log scale (lKa, lVol and lCl) and lVol and lCl defined to be random. Note conc is the square root of concentration. The starting values are lKa=.3, lVol=-.6, and lCl=-3. The control statement implements the nlm optimizer and only produces a result if convergence is achieved.

```
> model.def <- list(fixed.model = c(lKa + lV + lCl ~ 1), random.model = lKa +
+     lV + lCl ~ 1, start.lst = c(0.5, -0.6, -3), form = conc ~
+     sonecpmt(dose, time, lV, lKa, lCl), control = nlmeControl(returnObject = FALSE,
+     opt = c("nlm")))
```

Implementing the RunNLME function provides the user with the PKNLME object MM. Summary(MM$mm) gives the user the basic estimation results.

```
> MM <- RunNLME(inputStructure = model.def, data = data, nameData = nameData)
> summary(MM$mm)

Nonlinear mixed-effects model fit by maximum likelihood
  Model: form
 Data: pkdata
        AIC      BIC  logLik
  0.4275005 28.30242 9.78625

Random effects:
 Formula: list(lKa ~ 1, lV ~ 1, lCl ~ 1)
 Level: id
 Structure: General positive-definite, Log-Cholesky parametrization
         StdDev    Corr
lKa      0.6414976 lKa    lV
lV       0.1194798 -0.146
lCl      0.2455727 -0.121  0.999
Residual 0.1683417

Fixed effects: inputStructure$fixed.model
        Value  Std.Error  DF   t-value p-value
```

```
lKa  0.351867 0.19865594 106   1.77124  0.0794
lV  -0.787758 0.04367628 106 -18.03630  0.0000
lCl -3.214108 0.07679422 106 -41.85352  0.0000
 Correlation:
    lKa    lV
lV   0.024
lCl -0.137  0.677

Standardized Within-Group Residuals:
         Min           Q1          Med           Q3          Max
-3.359046251 -0.474004597 -0.008880566  0.325871602  3.141481129

Number of Observations: 120
Number of Groups: 12
```

Typing names(MM) gives the contents of the "PKNLME" object including "nlme" object, pkdata file that includes the PK data and covariates, and the nameData file.

```
> names(MM)

[1] "mm"       "cov.id"   "pkdata"   "nameData"
```

Typing names(MM$mm) gives the objects contained in the "nlme" object including the coefficients, residuals, fitted values etc.

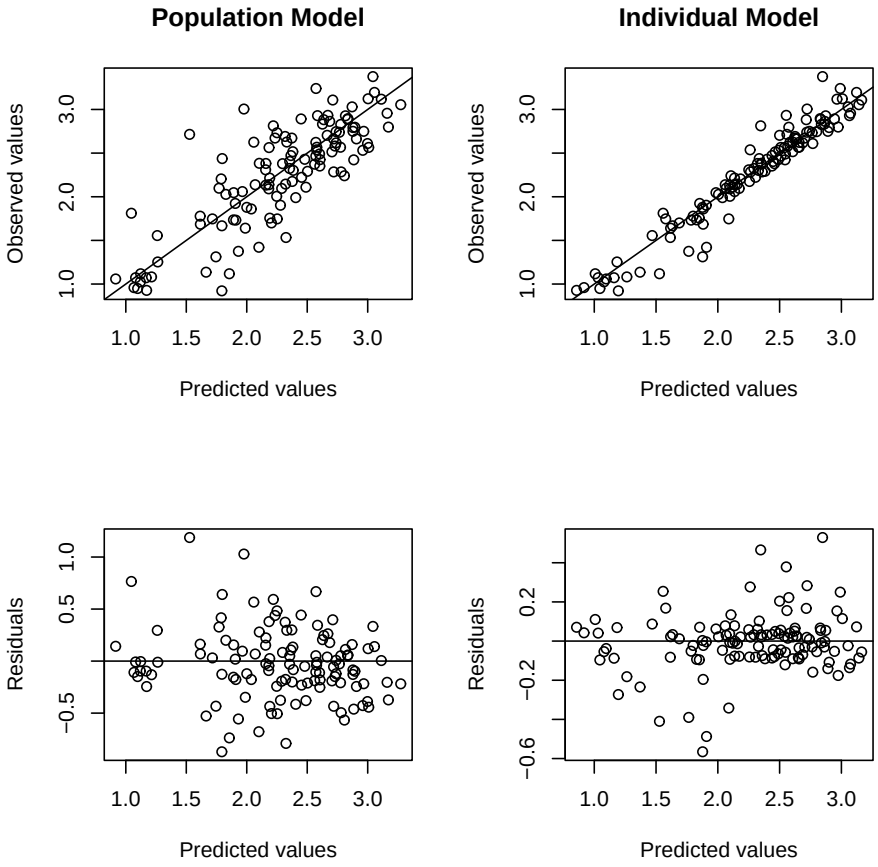
```
> names(MM$mm)

 [1] "modelStruct"  "dims"         "contrasts"    "coefficients" "varFix"
 [6] "sigma"        "apVar"        "logLik"       "numIter"      "groups"
[11] "call"         "method"       "fitted"       "residuals"    "plist"
[16] "map"          "fixDF"
```

The user can get a report including parameter estimates and model diagnostic plots using either the `tex` or `HTMLtools` functions for any of the 3 estimation methods including, generalized least squares (NLME), maximum likelihood (NONMEM) and Bayesian (WinBUGS) methods. To run `HTMLtools`, in addition to identifying the model object, the user must give the nameData list (discussed above), a nameFile list which gives the name of the output file for the HTML report file (i.e. nlme.output)and the names for each of the plots being output to an .html file. descStructure identifies the percentiles the user would like to see with the number of significant figures for the desc function. nameDir allows the user to define the directory where the HTMLfiles are placed. In this case we sent nlme.output.html to the standard NONMEM directory `C:\nmv\run` .

```
> nameFile <- list(file = "nlme.output", file1 = "trplt.nl", file2 = "diagplt.nl",
+     file3 = "qqploti.nl", file4 = "qqnormre.nl", file5 = "covre.nl",
+     file6 = "diagtrplti.nl", file7 = "diagtrpltp.nl")
> descStructure <- list(pcts = c(0.025, 0.05, 0.95, 0.975), nsig = 4)
> HTMLtools(x = MM, nameData = nameData, nameDir = "C:/nmv/run",
+     nameFile = nameFile, descStructure = descStructure)
```

```
> diagplot(MM)
```




```
Results written to C:/nmv/run
```

The plots provided in the report functions can also be run individually (see diagplot(), diagtrplot(), trplot(), obvsprplot(), residplot()). For example the figure shows diagplot(x=MM).

3 Implementing the model comparison and estimation method comparison functions in PKtools

PKtools also includes functions for model comparison and estimation method comparison. Before we can run the functions for model and estimation method comparison, we need to run our model in RunNM to get maximum likelihood estimates, and RunWB to get Bayesian estimates. RunNM uses the data set up for RunNLME so all that is required is a nameData list and the inputStructure, which for RunNM is a NONMEM control file. An example NONMEM control file is included in the appendix.

```
> nameData <- list(covnames = c("wt"), yvarlab = "Sqrt(Theop. Conc.) Sqrt(mg/L)",
+     xvarlab = "Time since dose (hrs)", reparams = c("Ka", "V",
+         "Cl"), params = c("Ka", "V", "Cl"), tparams = c("log(Ka)",
+         "log(V)", "log(Cl)"), varnames = c("D[1,1]", "D[1,2]",
```

```
+         "D[2,2]", "D[1,3]", "D[2,3]", "D[3,3]"))
> NM <- RunNM(inputStructure = "control.model5", data = data, nameData = nameData)
```

RunWB like the other run functions requires a data set, a nameData list and the inputStructure. The data includes a list of required data files, including the basic pk data, as well as, a vector/matrix of covariates. The RunWB data file also includes any fixed parameters like number of subjects, the number of values per person required to define the model (see inputStructure below) and values for the hyperparameters.

```
> library(nlme)
> data(Theoph)
> Theoph <- Theoph[Theoph$Time != 0, ]
> id <- as.numeric(as.character(Theoph$Subject))
> dose <- Theoph$Dose
> time <- Theoph$Time
> conc <- round(sqrt(Theoph$conc), 4)
> sid <- split(id, id)
> hist <- sapply(sid, length)
> n.ind <- 12
> off.data <- matrix(NA, n.ind + 1, 1)
> off.data[1, 1] <- 1
> for (i in 2:(n.ind + 1)) off.data[i, 1] <- off.data[i - 1, 1] + hist[i - 1]
> off.data <- c(off.data)
> mean <- c(0.5, -0.6, -3)
> R <- structure(.Data = diag(rep(0.1, 3)))
> prec <- structure(.Data = diag(rep(1e-06, 3)))
> data <- list(n.ind = n.ind, off.data = off.data, dose = dose, conc = conc, time = time,
+     mean = mean, R = R, prec = prec)
> wt.v <- Theoph$Wt
> data <- list(data = data, cov = wt.v, id = id)
> nameData <- list(covnames = c("wt"), yvarlab = "Sqrt(Theop. Conc.) Sqrt(mg/L)",
+     xvarlab = "Time since dose (hrs)", coef = c("Ka", "V", "Cl"), params = c("Ka",
+         "V", "Cl"), reparams = c("Ka", "V", "Cl"), tparams = c("log(Ka)", "log(V)",
+         "log(CL)"), varnames = c("D[1,1]", "D[1,2]", "D[1,3]", "D[2,1]", "D[2,2]",
+         "D[2,3]", "D[3,1]", "D[3,2]", "D[3,3]"))
```

nameData is similar to that described above and the inputStructure for RunWB is the WinBUGs text model file. An example is provided in the appendix. Finally there is a list of additional WinBUGS arguments (WBargs). The WinBUGS arguments (WBargs) includes a list of the parameters to be sampled (parameters), the initial values for the parameters to be estimated (inits), the number of chains (n.chains), the number of iterations (n.iter), the burnin (n.burnin) and the number of samples to thin (n.thin). There is also an option called debug when set debug=T stops the program at the WinBUGS level. The user can then use WinBUGS in the standard Windows format to assess convergence and mixing and debug any problems. Convergence and mixing are assessed to determine if the MCMC sample has reached the stationary distribution (i.e. the target distribution). Further information on this and other topics on the application of MCMC for fully Bayesian analyses are provided in the WinBUGS Manual (Spiegelhalter et al., 2001), Lunn et al. (2002), Gelman et al. (2003), and Chib (2003).

```
> parameters <- c("sigma2", "ka", "cl", "v", "beta", "mu", "re", "itau", "ipredwb",
+     "ppredwb")
> inits <- function() {
+     list(beta = structure(.Data = c(rep(0.5, 12), rep(-0.6, 12), rep(-3, 12)),
+         .Dim = c(12, 3)), mu = c(0.5, -0.6, -3), tau = structure(.Data = c(0.1,
+         0, 0, 0, 0.1, 0, 0, 0, 0.1), .Dim = c(3, 3)), tauC = 20)
+     list(beta = structure(.Data = c(rep(-0.5, 12), rep(-0.8, 12), rep(-3.5,
+         12)), .Dim = c(12, 3)), mu = c(-0.5, -0.8, -3.5), tau = structure(.Data = c(0.1,
+         0, 0, 0, 0.1, 0, 0, 0, 0.1), .Dim = c(3, 3)), tauC = 20)
+     list(beta = structure(.Data = c(rep(1.5, 12), rep(-0.4, 12), rep(-2.8, 12)),
+         .Dim = c(12, 3)), mu = c(1.5, -0.4, -2.8), tau = structure(.Data = c(0.1,
+         0, 0, 0, 0.1, 0, 0, 0, 0.1), .Dim = c(3, 3)), tauC = 20)
+ }
> WBargs <- list(parameters = parameters, inits = inits, n.chains = 3, n.iter = 12000,
+     n.burnin = 4000, n.thin = 3, debug = T)
> WB <- RunWB(inputStructure = "theosw.txt", data = data, nameData = nameData,
+     WBargs = WBargs)
```

paramEst provides the user with a table of population parameter estimates for the three estimation methods.

```
> paramEst(PKNLMEobject = MM, NMobject = NM, WBobject = WB)
```

	NMparams	NMse	NLparams	NLse	WBparams	WBse
log(Ka)	0.3594780	0.40571700	0.35186706	0.19865594	0.37420	0.212300
log(V)	-0.7795500	0.10409500	-0.78775850	0.04367628	-0.77910	0.060500
log(Cl)	-3.1984400	0.22354300	-3.21410810	0.07679422	-3.22100	0.089210
D[1,1]	0.4261940	0.74846700	0.41151915	NA	0.50130	0.297200
D[1,2]	-0.0133228	0.04634360	-0.01122226	NA	-0.01007	0.050640
D[2,2]	0.0140928	0.04371580	0.01427543	NA	0.03503	0.019630
D[1,3]	-0.0195110	0.15941900	-0.01906321	NA	-0.02083	0.075360
D[2,3]	0.0294696	0.05444040	0.02932079	NA	0.03115	0.022620
D[3,3]	0.0617928	0.05697050	0.06030594	NA	0.08391	0.045300
sigma^2	0.0285694	0.00423325	0.02833893	NA	0.02992	0.004622

indEst provides the user with a table of individual level parameters for the three estimation methods.

```
> indEst(PKNLMEobject = MM, NMobject = NM, WBobject = WB)

[1] "Individual Estimates - Ka"
        NONMEM        NLME     WinBUGS
1   0.38730114  0.38524628  0.45256523
2   0.56338012  0.56536915  0.50820229
3   0.76285979  0.76725025  0.80260144
4   0.07881118  0.07928331  0.06849518
5   0.17134487  0.17401358  0.21215355
6   0.01123663  0.01721571  0.05847529
7  -0.44951104 -0.44420411 -0.40496065
```

```
8   0.22999999  0.23649425  0.30701222
9   1.84580564  1.83655268  1.79900360
10 -0.38409019 -0.38488004 -0.30973102
11  1.09192330  1.10136231  1.17156650
12 -0.11192647 -0.11129867 -0.18774183
[1] "Individual Estimates - V"
       NONMEM       NLME    WinBUGS
1  -1.0507082 -1.0602663 -1.0191608
2  -0.7563426 -0.7631127 -0.8043770
3  -0.7684312 -0.7739943 -0.7581178
4  -0.8150663 -0.8239786 -0.8376640
5  -0.7362217 -0.7423779 -0.7179590
6  -0.6751501 -0.6839600 -0.6584368
7  -0.6700363 -0.6792036 -0.6540040
8  -0.7110057 -0.7182591 -0.6775253
9  -0.9219819 -0.9279946 -0.9673334
10 -0.8510181 -0.8605991 -0.8157360
11 -0.6182067 -0.6208514 -0.5843091
12 -0.7899224 -0.7985043 -0.8598322
[1] "Individual Estimates - Cl"
      NONMEM      NLME   WinBUGS
1  -3.770046 -3.776149 -3.821468
2  -3.145370 -3.160932 -3.132358
3  -3.166829 -3.181633 -3.194042
4  -3.279051 -3.291435 -3.279811
5  -3.110829 -3.122343 -3.142608
6  -2.985168 -3.003357 -3.020747
7  -2.983665 -2.998119 -3.016982
8  -3.056417 -3.072004 -3.102619
9  -3.468973 -3.488399 -3.442369
10 -3.364277 -3.371817 -3.402946
11 -2.843166 -2.862512 -2.897935
12 -3.229861 -3.240598 -3.202471
```

`AICcomp` calculates and or prints the AIC, AICc (small sample AIC) and the loglikelihood from NONMEM and NLME for each of any number of models.

```
> PKNLMEobjects = list(MM)
> NONMEMobjects = list(NM)
> AICcomp(PKNLMEobjects = PKNLMEobjects, NONMEMobjects = NONMEMobjects)

   NM AIC NLME AIC  NM AICc NLME AICc   NM of NLME of  K
1 -217.92    0.428 -215.902     2.446 -237.92 -19.572 10
```

The object oriented nature of the output in the R environment makes writing simulations and saving the results easy and `PKtools` provides the function `desc` for calculating simple statistics including mean, median, standard deviation and percentiles on these results.

```
> desc(Theo$conc)
```

```
       N      Mean       Med         S     0.025      0.05      0.95     0.975       Min       Max
120.0000    2.2550    2.3800    0.6042    0.9589    1.0720    3.0570    3.1240    0.9220    3.3760
```

4 Appendix

NONMEM: Example Control file The following is a standard control file for NONMEM. In the control file the user defines the model ($SUBROUTINE,$PK,$ERROR), identifies the data file ($DATA) and variable names ($INPUT), specifies the estimation method ($ESTIMATION) and covariance matrix definition ($COV), provides starting values for the parameters ($THETA,$OMEGA,$SIGMA) and identifies the output data tables ($TABLE). The standard output data tables include population and individual level residuals and predicted values (NMPR.TBL),as well as, individual parameter estimates (NMIP.TBL) and estimates of the individual random effects (NMRE.TBL). We also output a table of the individual subject covariates (NMCOV.TBL). Two additional subroutine files infnx5u.for and wrtab5msb.for indicate to NONMEM to create the tables of the population parameter estimates (PAR.TAB), the correlation matrix (COR.TAB) and the covariance matrix (COV.TAB). We made some minor changes in the wrtab5.for file provided by GLOBOMAX to make PAR.TAB a 3 column table that is readable in R. These files are included in the appendix. In addition we added four lines of code to $PK that read the minimum value of the object function we then created an output table MVOF.TBL.

Looking at the control file, the $SUBROUTINE ADVAN2 TRANS2 identifies a one compartment model with first order absorption parameterized using (Ka, V/F, CL/F). The THETA's represent fixed parameters and the ETA's represent the random effects. In our example the individual PK parameters are assumed to follow a log normal distribution, and the concentrations are normal after transformation by taking the square root. For more information on the NONMEM CONTROL file see the NONMEM Manuals (Boeckmann et al., 1994).

```
$PROBLEM  control.model5
$INPUT ID DOSE=AMT TIME CONC=DV WT EVID
$DATA NMdata
$SUBROUTINE ADVAN2 TRANS2 INFN=infnx5u.for OTHER=wrtab5msb.for
$PK
"FIRST
"      COMMON /ROCM8/ OBJECT
"      DOUBLE PRECISION OBJECT
MVOF=OBJECT

   TLKA=THETA(1)
   TVKA=EXP(THETA(1))
   KA=TVKA*EXP(ETA(1))

   TLV=THETA(2)
   TVV=EXP(THETA(2))
   V=TVV*EXP(ETA(2))

   TLCL=THETA(3)
   TVCL=EXP(THETA(3))
   CL=TVCL*EXP(ETA(3))

   K=CL/V
   S2=V

   HL=.693/K
   TVHL=.693*TVV/TVCL
```

```
$ERROR
  IPRED=SQRT(F+.05)
  IRES=IPRED-DV
  IWRES=IRES/.169    ; std dev=.169
  Y=SQRT(F+.05) + ERR(1)

$THETA  (-3, .5, 1.5) (-3,-.6,-.1)  (-5,-3,1); (-1,.01,1) (-1,.01,1)
$OMEGA  BLOCK(3) .5 -.01 .02 -.01 .01 .02
$SIGMA  .05
$ESTIMATION  MAXEVAL=9999 PRINT=3 METHOD=CONDITIONAL LAPLACIAN SIGDIGITS=5
$COV   MATRIX=S
$TABLE ID ETA1 ETA2 ETA3 FIRSTONLY ONEHEADER NOPRINT NOAPPEND FILE=NMRE.TBL
$TABLE ID KA V CL FIRSTONLY ONEHEADER NOPRINT NOAPPEND FILE=NMIP.TBL
$TABLE ID WT FIRSTONLY ONEHEADER NOPRINT NOAPPEND FILE=NMCOV.TBL
$TABLE EVID ID TIME IPRED IRES IWRES ONEHEADER NOPRINT FILE=NMPR.TBL
$TABLE MVOF FIRSTONLY NOPRINT NOHEADER NOAPPEND FILE=MVOF.TBL
```

WinBUGS: Example Model file without censoring (theosw.txt) The model file gives the definition of the model function (in our case a one compartment model with first order absorption), and associated parameter distributions and priors. We also define the population (ppredwb) and individual (ipredwb) level predicted values. We defined the concentrations to be normal after taking the square root and the individual pk parameters are assumed to be log normal. We used vague priors. (`TauC~gamma(.001,.001)`, `mu~mnorm(mean[],prec[,])`, `tau~wish(R[,],3)` the values for the hyperparameters are given in the data listing. Note that WinBUGS defines the distributions using precision instead of variance, so for example tauC is precision and sigma=1/tauC.

```
  model {
        for (i in 1:n.ind) {
            for (j in off.data[i]:(off.data[i+1]-1)) {
            conc[j] ~ dnorm(ipredwb[j], tauC)
            ipredwb[j] <- sqrt(dose[j]*ka[i]*(exp(-(cl[i]/v[i])*time[j] )-
                          exp(-ka[i]*time[j]))/( v[i]*(ka[i]-(cl[i]/v[i]))))
            ppredwb[j] <- sqrt(dose[j]*exp(mu[1])*(exp(-(exp(mu[3])/exp(mu[2]))*time[j] )-
                          exp(-exp(mu[1])*time[j]))/(exp(mu[2])*(exp(mu[1])-(exp(mu[3])/exp(mu[2])))))
            }
            ka[i] <- exp(beta[i, 1])
            v[i]  <- exp(beta[i, 2])
            cl[i] <- exp(beta[i, 3])
            hl[i] <- .693*v[i]/cl[i]
        beta[i, 1:3] ~ dmnorm(mu[ ],tau[,])
        re[i,1] <- beta[i,1] - mu[1]
        re[i,2] <- beta[i,2] - mu[2]
        re[i,3] <- beta[i,3] - mu[3]
        }
        tauC ~ dgamma(1.0E-3, 1.0E-3)
        sigma2<-1/tauC
        mu[1:3] ~ dmnorm(mean[ ], prec[ , ])
        tau[1:3,1:3 ] ~ dwish(R[ , ], 3)
        itau[1:3,1:3]<-inverse(tau[ , ])
        tvhl<-.693*exp(mu[2])/exp(mu[3])
}
```

References

Beal, S. and Sheiner, L. (1989). *NONMEM Users Guide - Part I.* NONMEM Project Group, UCSF.

Boeckmann, A., Sheiner, L., and Beal, S. (1994). *NONMEM Users Guide- Part V, Introductory Guide.* NONMEM Project Group, UCSF.

Chib, S. (2003). *Subjective and Objective Bayesian Statistics: Principles, Models and Applications*, chapter Markov Chain Monte Carlo Methods, pages 119–171. Wiley-Interscience, Hoboken,New Jersey.

Gelman, A., Carlin, J., Stern, H., and Rubin, D. (2003). *Bayesian Data Analysis*. Chapman & Hall/CRC, New York, 2nd edition.

Lunn, D., Best, N., Thomas, A., Wakefield, J., and Spiegelhalter, D. (2002). Bayesian analysis of population pk/pd models: General concepts and software. *Journal of Pharmacokinetics and Pharmacodynamics*, 29(3):271–307.

Pinheiro, J. and Bates, D. (2000). *Mixed-Effects Models in S and SPLUS*. Springer, New York.

Spiegelhalter, D., Thomas, A., Best, N., and Lunn, D. (2001). *Winbugs Version 1.4 User Manual*. Imperial College School of Medicine, London.