PK/PD Modeling with RsNLME : : CHEAT SHEET

Basics

RsNLME is an R package to define NLME models as R native objects.

- **NLME engine is used for Fitting/Simulation**
- **R style help available via ?method**
- **All methods have sensible default values**

Model Creation

```
pk = pkmodel(numComp = 1,
                    absorption=Intravaneous,…)
pd = emaxmodel(checkBaseline=TRUE,…)
pd = linearmodel(type = LinearAlpha,… )
pkpd = pkemaxmodel(numComp = 2,
                    absorption=Extravascular
                    isTlag = TRUE,…)
pkpd = pkindirectmodel(isClosedForm=FALSE,
                    indirectType = LimitedStimulation,…)
pkpd = pklinearmodel(parameterization= Micro,
                    linearModelType=LinearBeta, …)
model = blankmodel(modelName, …)
```

Add more components to the model

```
model2 = addCountObservation(model, observationName,
          expression, structuralParameters)
model2 = addCategoricalObservation(model,
          observationName, offsetArray,
          structuralParameters)
model2 = addContinuousObservation(model,
          observationName,effect, hasRandomEffect)
model2 = addLLObservation(model,observationName,
          expression,dobefore,doafter, structuralParameters,
          isFrozen, hasRandomEffect,simulationCode)
model2 = addEventObservation(model,observationName,
          epression,dobefore,doafter,structuralParameters,
          isFrozen, hasRandomEffect)
model2 = addParameters(model,name,…)
model2= addExpression(model,blockName
          ,structuralParameters, codeLine, isFrozen, override)
```

Input Options

```
model2 = addReset(model, low, hi)
model2 = addExtraDose(model, doseType, doses)
```

Dose Types : SteadyStateDose | AddlDose

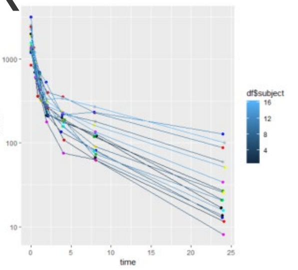
Model covariates

```
sex=categoricalCovariate("sex",c(1,2),c("female","male"))
weight=NlmeCovariateParameter("weight",centerValue="70",
                    continuousType  =CovarNumber,
                    direction=Forward)
age=NlmeCovariateParameter("age")
occ=occasionCovariate("OCC", c(1,2),  c("OCC1","OCC2"),
                 direction=Forward)
model=addCovariates(model,  c(sex,weight,age,occ),
                 c("V"="weight,age",  "Cl"="sex,weight"))
newModel = resetCovariateEffects(model)
covariateEffect(model,"wt","Cl")=COVAR_EFF_YES
```

Input dataset

Visualize the data with R

```
input=read.csv("16subjects.csv")
ggplot(data=input,aes(x=time,y=conc))+
scale_y_log10()+
geom_point(colour=df$subject)+
geom_line(aes(x=df$time,y=df$conc,
group=df$subject,colour=df$subject))
```



Map columns to model variables

```
dataset=NlmeDataset()
initColMapping(model)=input
print(modelColumnMapping(model))
modelColumnMapping(model)=c(id="Subject",
                    CObs="Conc",A1="Amount")
```

Customizing the model

Residual error model

```
residualEffect(model,"C")=c(errorType=Multiplicative,SD="0.16“,
          isFrozen=FALSE,isBQL=TRUE,bqlStatic=0.75,
          doBefore=“”,doAfter=“”))
```

Error model types :

Additive | LogAdditive | Multiplicative | AdditiveMultiplicative | MixRatio | Power | Custom

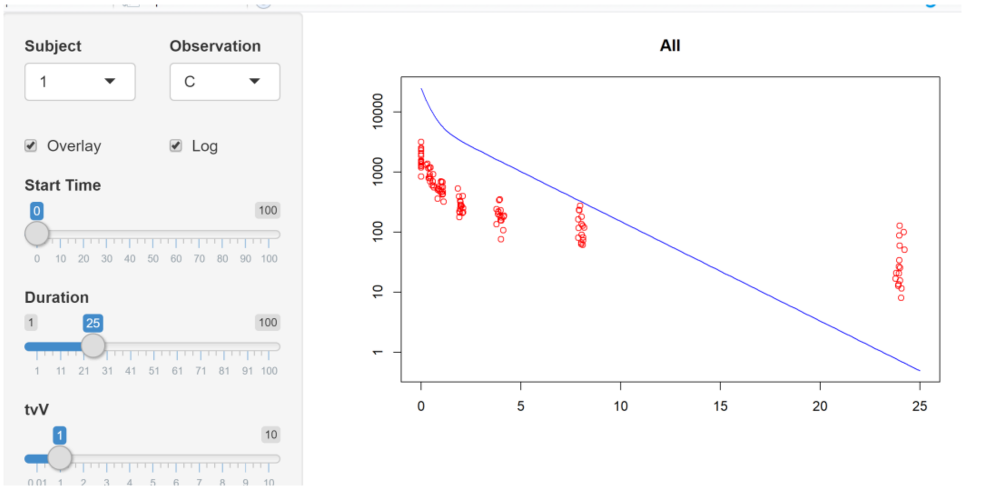
```
structuralParam(model,“V”) = c(style=Custom,
          code="stparm(V=10^(tvlog10V + nlog10V))")
```

Random effects

```
initRandomEffects(model)=c(Block, FALSE,
        "nV,nCl,nKa,nV2", "0.2, 0, 0.2, 0, 0, 0.2, 0, 0, 0, 0.1")
 initRandomEffects(model)=c(Diagonal,
        FALSE,"nV,nCl","0.1, 0.02")
initRandomEffects(model)=c(
        Diagonal, FALSE,"nV,nCl","0.1, 0.02",
        Block, TRUE,     "nCl2,nV2","0.2, 0, 0.2")
initOccasionRandomEffect(model,"Occasion") =
        c(0.1,0.02,0.1)
```

Initial estimates for fixed effects

```
estimatesUI(model,unique(input$Subject),host)
```



Subject: 1

Observation: C

☑ Overlay ☑ Log

Start Time: 0

Duration: 25

tvV: 1

```
effects=getInitialEstimates()
initFixedEffects(model) = effects
```

Platform/Engine defintion

```
method=NlmeParallelMethod(method="MULTICORE"|"MPI"|"TORQUE"|"TORQUE_MPI"|"SGE"|"SGE_MPI"|"LSF"|"LSF_MPI")
host = NlmeParallelHost(sharedDirectory="C:/Shared",
                        parallelMethod=method,
                        hostName="MPI",
                        numCores=4)
```

Fitting Engines Definition

```
engineParams = NlmeEngineExtraParams(
        PARAMS_METHOD=METHOD_FOCE_ELS,
        PARAMS_NUM_ITERATIONS=1000,
        PARAMS_SAND="TRUE")
```

```
METHOD_QRPEM          METHOD_FIRST_ORDER
METHOD_IT2S_EM        METHOD_LAPLACIAN
METHOD_FOCE_LB        METHOD_NAIVE_POOLED
```

Model Fitting

```
Job = fitmodel(hostPlatform, dataset, params , model,
        runInBackground )
```

Ex.

```
job = fitmodel(host,dataset,engineParams)
status = read.csv("Overall.csv")
print(status)
```

Model Simulation

```
Job = simmodel(hostPlatform, dataset, simParams ,
        model , runInBackground )
```

Ex.

```
SimTableObs = NlmeSimTableDef("SimTableObs.csv",
        "0,1,2,4,4.9,55.1,56,57,59,60", "C, CObs", FALSE)
simParams = NlmeSimulationParams(numReplicates = 50,
        seed = 3527, simulationTables = c(SimTableObs))
job = simmodel(defaultHost,dataset,simParams,model)
```

Covariate Search

```
stepwiseSearch(hostPlatform, dataset, params,
        covariateModel, stepwiseParams, model,
        runInBackground)
shotgunSearch(hostPlatform, dataset, params,
        covariateModel, model, runInBackground)
```

Example

```
sp = NlmeStepwiseParams(0.01, 0.001, "-2LL")
job=stepwiseSearch(host, dataset, params,
        covariateModel(model), sp, model)
```

Bootstrap

```
bootstrap(hostPlatform, dataset, params, bootParams,
        model, runInBackground)
```

Ex.

```
bootParams = NlmeBootstrapParams(
        numReplicates=5, randomNumSeed=1234)
job = bootstrap(defaultHost,dataset,params,
        bootParams,model,TRUE)
```

VPC

```
vpcmodel(hostPlatform, dataset, vpcParams, model, …)
```

Ex.

```
obsVars = GetObservationVariables(dataset)
observationParameters(obsVars[[1]])=c(xaxis=VPC_XAXIS_T,
        binningMethod=VPC_BIN_NONE,
        quantilesValues ="5,50,95") vpcParams =
        NlmeVpcParams(numReplicates=2, seed=1234,
        observationVars=obsVars)
job = vpcmodel(defaultHost,dataset,vpcParams,model)
```

Scenario Fitting

```
scenario1=NlmeScenario("SC0001","1")
scenario2=NlmeScenario("SC0002","1,2")
scenarios = c(scenario1,scenario2)
sortColumns=NlmeSortColumns("group,sex")
job = sortfit(defaultHost,dataset,params,
        sortColumns,scenarios,model)
```

Analyzing Results

VPC

```
library(vpc)
simData = getSimData(input,stratifyColumns="sex")
obsData = getObsData(input)
vpcdb = vpc(sim=simData, obs = obsData, vpcdb = TRUE)
plot_vpc(vpcdb,
        show = list(obs_dv = TRUE, obs_ci = FALSE),
        xlab = "Time(hours)", ylab = "Concentration",
        title = "VPC!")
```

Diagnostic plots

```
xp = xposeNlme(dir="./",modelName="Initial Model")
list_vars(xp)
doexpose(xp)
dv_vs_pred(xp)
res_vs_pred(xp,res="CWRES",type="ps")
ind_plots(xp)
eta_distrib(xp)
eta_qq(xp)
```

Bootstrap

```
out=read.csv("out.csv")
View(out)
overall=read.csv("BootOverall.csv")
View(overall)
theta=read.csv("BootTheta.csv")
View(theta)
varCovar=read.csv("BootVarCoVar.csv")
View(varCovar)
omega=read.csv("BootOmega.csv")
View(omega)
```

Covariate Search

```
overall = read.csv("Overall.csv")
View(overall)
stepwiseLines=readLines("Stepwise.txt)
View(stepwiseLines)
```