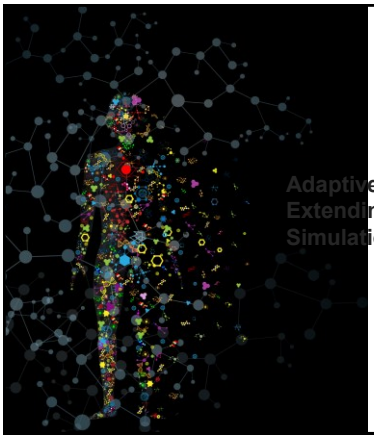


Adaptive Simulations: Extending PML to Trial Simulations

Problem Description: pre-clinical example

- Indirect-Response-Model developed that describes animal data
  - Population estimates for PK and PD
    - V
    - Cl
    - Emax
    - EC50
  - Between-Subject-Variabilities:
    - nV
    - nCl
    - nEmax
    - nEC50
  - ➢ Simulation will yield variabilities in concentrations and effects

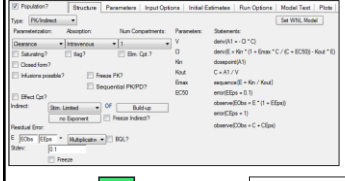
Problem Description: Desirable Dose

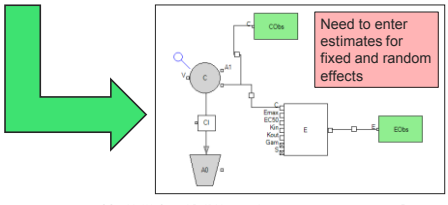
- Question:
  - What is the optimal dose - a dose for which 95% of animals reach a threshold response?
    - Determine the probability for animals to reach a certain response
  - How can we simulate the variability in doses?
    - Need a function of concentration and effect that yields a lognormal distribution of the outcome of different doses
- Not possible in Phoenix Model
  - –> need Trial Simulator Engine (TSE)

Adaptive Simulation: Desired Outcome & Protocol

- Desired Outcome:
  - Achieve minimum probability of 95% that animals will reach a therapeutic threshold
    - For indirect response model: ABEC>3000
    - Where ABEC = Area between Baseline and Effective Curve
- Protocol
  - Start with initial dose
  - Simulate PK/PD
  - Check PD response
    - If target achieved:
      - Stop
    - Else
      - Increase dose (incrementally)
      - Go back to Start

Indirect Response: From Built-In to Graphical



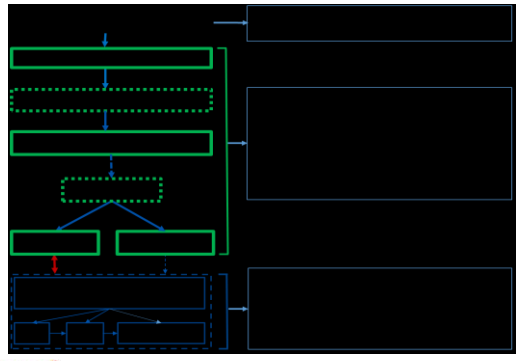


Indirect Response: From Graphical to Textual

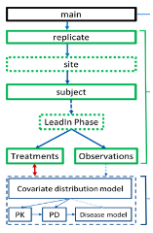
```
PKPD() {
  deriv(A1 = - Cl * C)                                  # differential equation for central compartment
  deriv(E = Kin * (1 + Emax * C / (C + EC50)) - Kout * E)  # ditto for indirect response model with stimulation of buildup
  deriv(ABEC = E - Kin / Kout)                          # ditto for calculation of area between baseline and effective curve
  dosepoint(A1)
  C = A1 / V
  sequence(E = Kin / Kout)                              # initialize the Effect, baseline response
  error(EEps = 0.1)                                     # estimate for standard deviation of residual error (Concentration)
  observe(CObs = C * (1 + CEps))                        # multiplicative residual error model for Concentration
  error(EEps = 1)                                       # estimate for standard deviation of residual error (Effect)
  observe(EObs = E + EEps)                              # additive residual error model for Effect
  stparm(V = tvV * exp(nV))                             # specification of structural parameters with
  stparm(Cl = tvCl * exp(nCl))                          # typical values (fixed effects)
  stparm(Kin = tvKin * exp(nKin))                       # and between-subject variabilities (random effects)
  stparm(Kout = tvKout * exp(nKout))
  stparm(Emax = tvEmax)
  stparm(EC50 = tvEC50)
  fixef(tvV = c(, 0.27, ))                              # specification of fixed effects in vector format:
  fixef(tvCl = c(, 0.15, ))                             # c(lower bound, initial estimate, upper bound)
  fixef(tvKin = c(, 1, ))
  fixef(tvKout = c(, 0.34, ))
  fixef(tvEmax = c(, 340, ))
  fixef(tvEC50 = c(, 6.4, ))
  ranef(diag(nV, nCl, nKin, nKout) = c(.1, .1, .1, .1))  # specification of random effects in diagonal structure
                                                        # output variable for Trial Simulator Engine (TSE)
```


Demo 1

Structure of Trial Simulation Model



Adaptive Simulation: PML for TSE

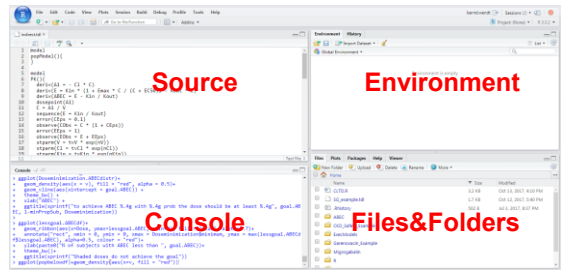



```
model                           # POPULATION MODEL
popModel(){}                    # no need to specify covariate distributions (demographics)

join {                          # JOIN PROCEDURE
  pop:   popModel()             # brings together the population model (distributions)
  pk:    PKPD()                 # with PKPD model
}

proc obstrt() {                 # OBSERVATION-TREATMENT PROCEDURE
  observe(pk$CObs)              # read observed concentration from PKPD model
  sleep(0.1)                    # wait for a small time increment
  bolus(pk$Al, Dose)            # apply an IV bolus dose, where dose is variable
  while (t < 24){               # while time is little than dosing interval
    sleep(1)                    # wait another time unit
    observe(pk$CObs)            # read predicted concentrations from model
  }                             # at the end of dosing interval:
  observe(pk$ABEC)              # read ABEC value from model
}
subject(iSub, iArm){            # SUBJECT PROCEDURE
  StartSubject()                # Initialize generation of subjects
  start obstrt()                # Initialize observations and treatments
}
proc replicate(){               # REPLICATE PROCEDURE
  int (iSub, iArm)              # specify variables for subject and treatments
  iArm = 0                      # in our case - there is no iteration over treatments
  InitReplicate()               # Initialize replication
  for (iSub in 1:nSub){         # loop over the desired number of subjects
    start subject(iSub, iArm)   # generate a subject
  }
}
double(Dose)                    # take value for Dose from input
int(Nsub)                       # take value for the number of subjects from input
proc main(){                    # MAIN PROCEDURE
  call replicate();             # execute simulation by starting the replicate
}
```


User Interface to Trial Simulation Engine: RStudio



Running Trial Simulation Engine through an R-Script

```
###
# setup work environment
###
setwd("~")
OutputFolder <- "~/ExecModels" # the executable file will be located here
resultDirName <- "~/Results" # the results will be located here
DeleteOldResults <- TRUE # change to FALSE if not intended
###
# setup TSE environment
###
InstallPath = Sys.getenv("TSERoot")                    # path to the compilation routines
source(paste0(InstallPath, "/RExamples/tools.R"))      # load predefined functions
BatchFile <- paste0(InstallPath, "/InstallDir/MakeExecTSE.sh")  # the compilation shell-script
###
# Choose model and compile
###
                                                       # PML Model
Model = normalizePath(ModelFile)                       # full path to the model file
batchRunStatus <- try(batchRun(BatchFile, Model, OutputFolder)) # compile model
###
# load R-libraries for post analysis
###
library(data.table)
library(dplyr)
library(ggplot2)
library(grid)
###
# Clear results directory
###
resDirStatus <- try(createResultDirectory(resultDirName, DeleteOldResults))
cat(resDirStatus)
###
# run the trial simulation
###
###
# read in and filter data
###
repdata <- fread("./Results/rep_1_1.csv", col.names = c("type", "iScen", "iRep", "iArm",
                 "iSub", "iPar", "t", "tUnit", "name", "y", "yUnit", "r"))
```


Demo 2

Trial Simulation Engine: Optimize Dose R-Script

```
###
Dose=5
goalachieved=0
while(goalachieved <=0.95)  {
  AddParam = paste0("Dose=", Dose)
  Scenarios(Model, OutputFolder, resultDirName, 1, 1, AddParam, 1234)
  repdata <- fread("./Results/rep_1_1.csv", col.names = c("type", "iScen", "iRep", "iArm",
                   "iSub", "iPer", "t", "tUnit", "name", "v", "vUnit", "n"))

  PKdata <-repdata %>% filter(name == "pk_CObs" & !is.na(v))
  PDdata <-repdata %>%filter(name == "pk_EObs" & !is.na(v))
  ABECdata <-repdata %>%filter(name == "pk_ABEC" & !is.na(v))
  populationachieved <-  ABECdata %>%  summarize(poptargetachieved = sum(v>3000)/n())
  goalachieved=populationachieved[1,1]
  Dose= Dose + 5
}
```

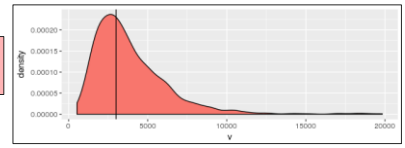
```
# set initial dose to 5 units
# initialize goal criterion to 0
# loop until goal criterion is established
# prepare a parameter string for Dose
# run replicate (simulation)
# read in the results

# filter results for PK observations
# filter results for PD observations
# filter results for summary ABEC value
# calculate population that achieved the target
# take first element of target frame
# incrementally increase Dose
```

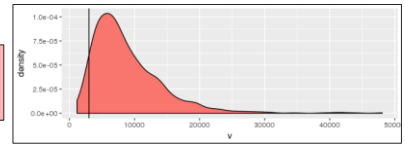
CERTARA

Adaptive Simulation: Results

Distribution of ABEC-Values for Dose=10



Distribution of ABEC-Values for optimal Dose (Dose=130)



CERTARA

Adaptive Simulations: Summary

- Introduction of a new PML-Tool: Trial Simulator Engine (TSE)
  - Simple pre-clinical example
    - What is the optimal dose for our next study?
  - Extension of Phoenix Modeling Language (PML)
    - Definition of trial simulation protocol, procedures, functions
  - RStudio as user interface to TSE
    - Requires scripting

CERTARA

Questions?

Trial Simulator Engine Introductory Program

- Trial Simulator Engine will launch in January 2018
- You can evaluate TSE with the Introductory Program
  - 2 months of complimentary access to TSE
  - Free training webinar training sessions
  - Online user guide
  - Online forum support
- At the end of the Introductory Program, we would like to get your feedback on your experience with a brief online survey
- If you would like to proceed with a purchase, you can do so at a special rate
- Contact sales@certara.com if you are interested in joining the TSE Introductory Program

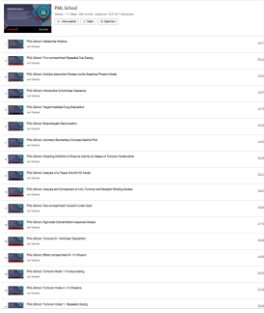
CERTARA

PML School: Materials

Forum: >30 Topics

https://support.certara.com/forums/forum/34-pml-school/

Youtube: 22 videos



https://www.youtube.com/user/CertaraLP/playlists

CERTARA

Coming up…

- New Series in 2018: NONMEM to PML Comparisons
  - Popular Models using NONMEM software
    - 1:1 translation into Phoenix Modeling Language
    - Setup and run NONMEM models in Phoenix
    - Setup and Run same model in Phoenix NLME
    - Compare Results

CERTARA