Diabetes Toolkit:
A Clinical Trial Simulation Tool for Diabetes Trials

Dongwoo Kang & Raymond Miller
Modeling and Simulation
Daiichi Sankyo Pharma Development

Introduction

- Requirements of clinical trial simulation
  - Several procedures need to be done using various computing tools
  - Intermediate results have to be seamlessly passed down to the next stage
- Diabetes Toolkit to do clinical trial simulation with ease
  - Data generation, analysis, summary, and plotting are done in one place using MSToolkit
  - External NONMEM run can be called for advanced simulation
- Collaboration with Mango Solutions
  - All-in-one R code package

MSToolkit

- An R package for modeling and simulation
- Developed by a few Pfizer colleagues, now freeware
- Major component of Diabetes Toolkit
- generateData(…)
  - Calls other low level functions to generate simulated data
  - Models can be specified by R function or NONMEM control stream
  - Stores data in ‘replicateData’ directory in the working directory using filenames of ‘replicateNNNN.csv’
- analyzeData(…)
  - Requires a valid R code or external file (*.R or *.SAS) for analysis
  - Requires functions to perform micro-evaluation and macro-evaluation summary
  - analysisCode must return MEAN, SE, LOWER, UPPER, DOSE

Model for FPG and HbA1c

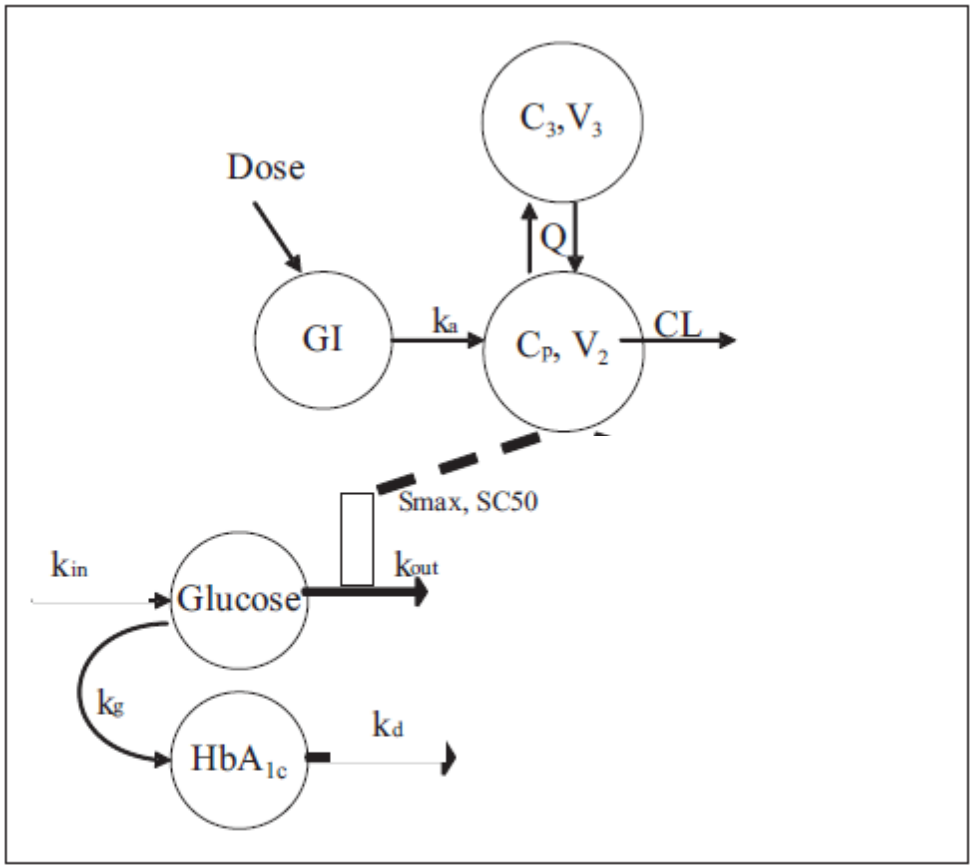
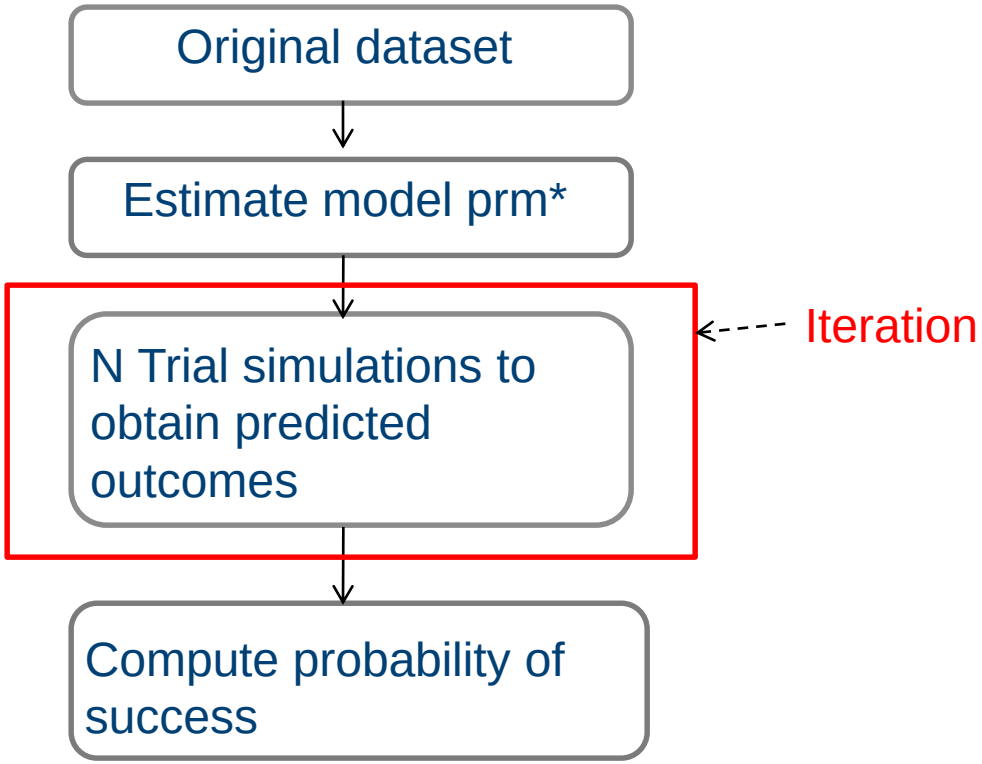


Figure 2. Depiction of FPG-HbA_{1c} and hemodilution exposure-response models. FPG, fasting plasma glucose; GI, gastrointestinal tract.

- Rivoglitazone model (Rohatagi et al., J Clin Pharmacol 2008) was used to assess the feasibility of short term study of a new compound

Daiichi-Sankyo

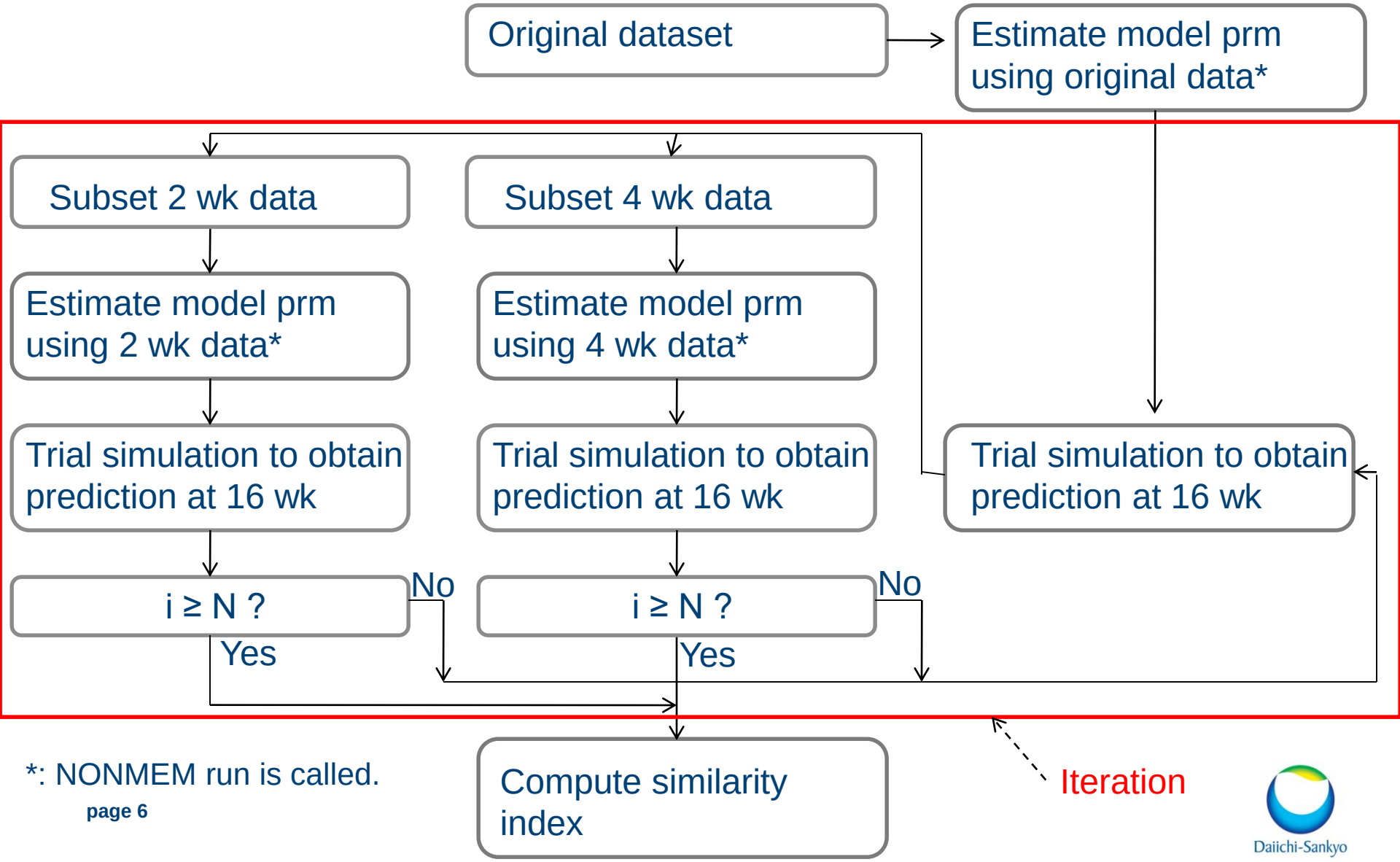
Flow diagram (Module 1): Comparison with respect to baseline



*: NONMEM run is required. Currently this step is pre-run and output files are provided within the toolkit.

Daiichi-Sankyo

Flow diagram (Module 2): Comparison with respect to long term study

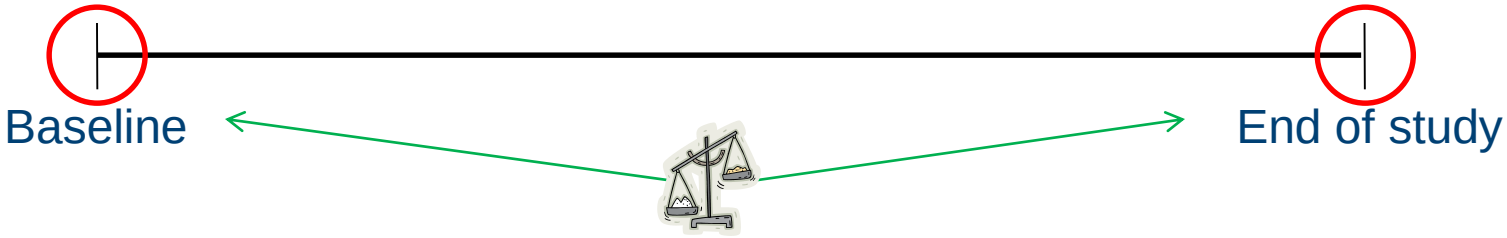


*: NONMEM run is called.

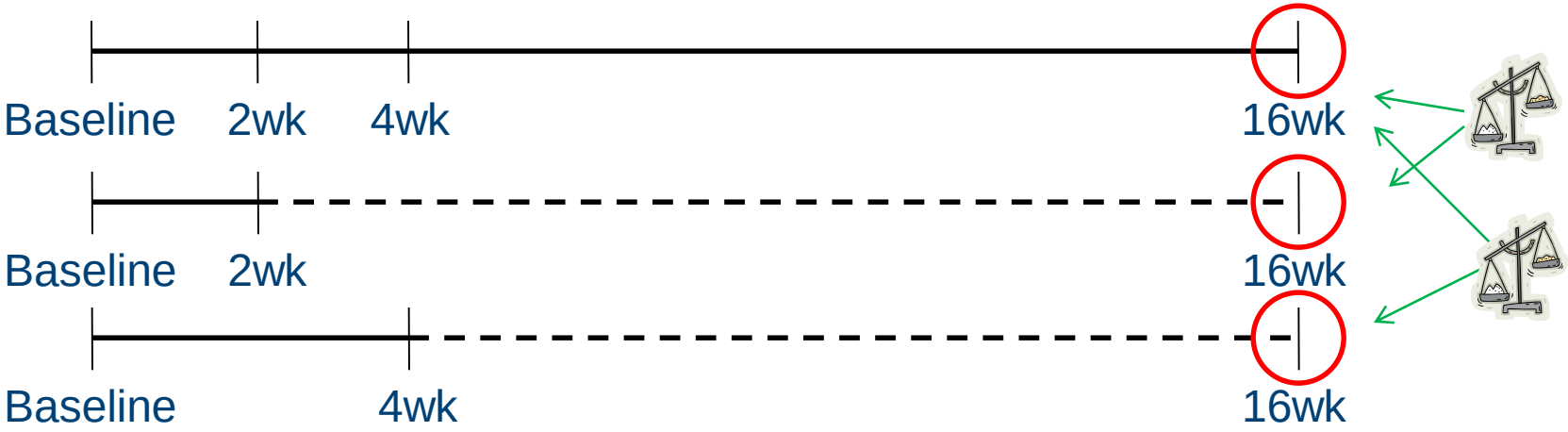
Daiichi-Sankyo

Chronological diagrams of Module 1 and 2

- Module I



- Module II



Daiichi-Sankyo

Definition of trial success

- Module 1: Comparison with respect to baseline
  - Success if significant difference
  - Mean ΔFPG $\geq$ 10 mg/dL for all doses, or 3, 5 mg (ie, high doses)
  - Mean ΔHBA $\geq$ 0.3% for all doses, or 3, 5 mg
- Module 2: Comparison with respect to long term study
  - Success if acceptable similarity
  - |Mean ΔFPG| < 5 mg/dL for all doses, or 0.5, 2, 5 mg (ie, key doses for dose-response estimation)
  - |Mean ΔHBA| < 0.15% for all doses, or 0.5, 2, 5 mg

Module 1

```
require(diabetes)

# Whether to write the outputs to files
dowrite <- TRUE

# Set working directory
# options(diabetes.wkdir = getwd())
wkdir <- .verifyFolder(file.path(getOption("diabetes.wkdir"), "Module1"))
ctsdir <- .verifyFolder(wkdir, "cts")
resultdir <- .verifyFolder(wkdir, "output")

# set criteria values:
#   endpoint - baseline <= criteria.CFB
#   abs(prediction - original) <= criteria.PRED
options(criteria.CFB = c(FPG = -10, HBA = -0.3))
options(criteria.PRED = c(FPG = 5, HBA = 0.15))

# Get parameters from NONMEM outputs
pars <- paras.run2(file.path(system.file(package = "diabetes"), "unitTests", "testdata", "nmtran", "run2"))

# simulate data
simulateData( modelEqn = model.run2,
                    replicateN = 2, subjects = 50, treatSubj = c(10, 10, 10, 10, 10),
                    treatDoses = c(500, 1000, 2000, 3000, 5000), treatPeriod = c(0:112) * 24,
                    genParNames = names(pars$thetaMean), genParMean = pars$thetaMean, genParVCov = pars$thetaVCov,
                    genParCrit = paste(names(pars$thetaMean), ">= 0"),
                    genParBtwNames = pars$omega2theta, genParBtwMean = 0, genParBtwVCov = pars$omega2Mean,
                    conCovNames = names(pars$conCovMean), conCovMean = pars$conCovMean, conCovVCov = pars$conCovVCov,
                    conCovCrit = paste(names(pars$conCovMean), "> 0"),
                    disCovNames = pars$disCovNames, disCovVals = pars$disCovVals, disCovProb = pars$disCovProb,
                    respVCov = 0, workingPath = ctsdir
)
```


Daiichi-Sankyo

Module 1

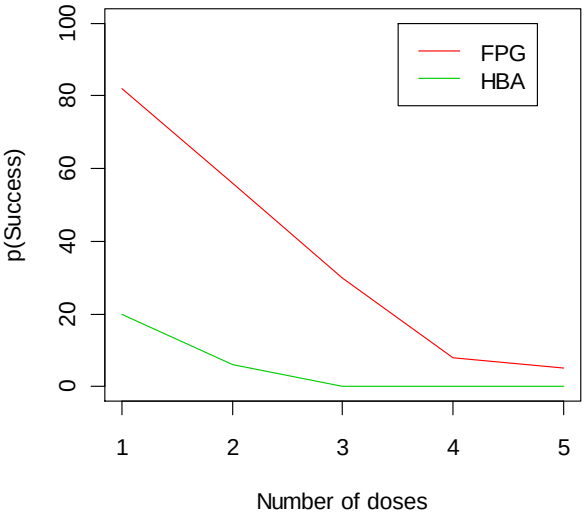
```
# Analysis data by using default functions. You can define 'analysisCode' and 'macroCode' functions here
analyzeData(analysisCode = .microCode.run2.module1 , macroCode = .macroCode.run2.module1 , workingPath = ctsdir)
file.copy(file.path(ctsdir, "microSummary.csv"), resultdir, overwrite = TRUE)
file.copy(file.path(ctsdir, "macroSummary.csv"), resultdir, overwrite = TRUE)

# Read data
allDf <- readReplicate(workingdir = ctsdir)
microDf <- readMicrodata(workingdir = ctsdir)

# Write plots to file
if(dowrite){
      pdf(file = file.path(resultdir, "out.pdf"))
}

# Analysis
cfb.HBA <- tableCFBCont(outvar = 'HBA', allDf = allDf, needwrite = dowrite, needplot = TRUE, resultdir = resultdir)
cfb.FPG <- tableCFBCont(outvar = 'FPG', allDf = allDf, needwrite = dowrite, needplot= TRUE, resultdir = resultdir)
cfbtest.HBA <- tableCFBTest(outvar = 'HBA', allDf = allDf, needwrite = dowrite, resultdir = resultdir)
cfbtest.FPG <- tableCFBTest(outvar = 'FPG', allDf = allDf, needwrite = dowrite, resultdir = resultdir)

plotBias(outvar = 'HBA', allDf = allDf)
plotBias(outvar = 'FPG', allDf = allDf)

plotTime(outvar = 'HBA', allDf = allDf, byCol = "REP")
plotTime(outvar = 'FPG', allDf = allDf, byCol = "REP")

plotDoseResponse(outvar = 'HBA', microDf = microDf)
plotDoseResponse(outvar = 'FPG', microDf = microDf)

if(dowrite) dev.off()
```


Daiichi-Sankyo

Module 1 run results: probability of success

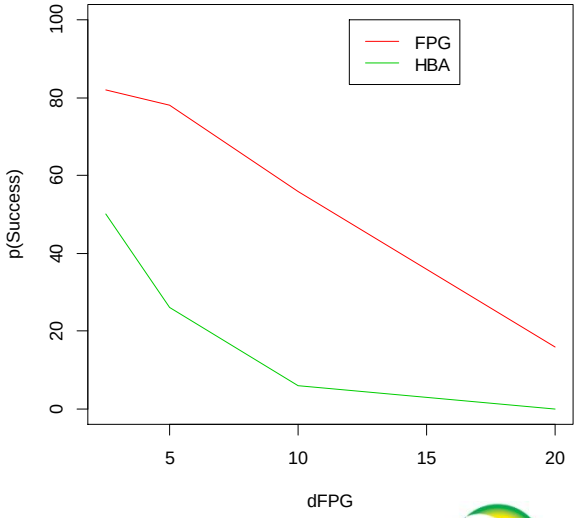
- Success if significant difference
  - Mean ΔFPG ≥ 10 mg/dL
  - Mean ΔHBA ≥ 0.3%

Dose	FPG	HBA
0.5, 1, 2, 3, 5	5	0
1, 2, 3, 5	8	0
2, 3, 5	30	0
3, 5	56	6
5	82	20



- Success if significant difference
  - For 3mg and 5mg

ΔFPG ≥	ΔHBA ≥	FPG	HBA
2.5	0.075	76	50
5	0.15	78	26
10	0.3	56	6
20	0.6	16	0



Daiichi-Sankyo

Module 2

```
require(diabetes)


doWrite <- TRUE      # Whether to write the outputs to files
doNONMEM <- FALSE    # Whether to invoke NONMEM to estimate parameters


# Set working directory
# options(diabetes.wkdir = getwd())
studydir <- .verifyFolder(file.path(getOption("diabetes.wkdir"), "Module2"))
study1dir <- .verifyFolder(studydir, "study1")
study2dir <- .verifyFolder(studydir, "study2")
if(!file.exists(file.path(study1dir, "ReplicateData"))) dir.create(file.path(study1dir, "ReplicateData"))
if(!file.exists(file.path(study2dir, "ReplicateData"))) dir.create(file.path(study2dir, "ReplicateData"))
# set criteria values:
#   endpoint - baseline <= criteria.CFB
#   abs(prediction - original) <= criteria.PRED
options(criteria.CFB = c(FPG = -10, HBA = -0.3))
options(criteria.PRED = c(FPG = 5, HBA = 0.15))
```


Daiichi-Sankyo

Module 2

```
##########################################################################
## Step 1: Simulate FPG and HAB for a long term study (eg. 16 wk)  ##
##########################################################################

# Get parameters from NONMEM outputs
pars <- paras.run2(file.path(system.file(package = "diabetes"), "unitTests", "testdata", "nmtran", "run2"))

# simulate data
simulateData( modelEqn = model.run2,
                    replicateN = 2, subjects = 50, treatSubj = c(10, 10, 10, 10, 10),
                    treatDoses = c(500, 1000, 2000, 3000, 5000), treatPeriod = c(0:112) * 24,
                    genParNames = names(pars$thetaMean), genParMean = pars$thetaMean, genParVCov = pars$thetaVCov,
                    genParCrit = paste(names(pars$thetaMean), ">= 0"),
                    genParBtwNames = pars$omega2theta, genParBtwMean = 0, genParBtwVCov = pars$omega2Mean,
                    conCovNames = names(pars$conCovMean), conCovMean = pars$conCovMean, conCovVCov = pars$conCovVCov,
                    conCovCrit = paste(names(pars$conCovMean), "> 0"),
                    disCovNames = pars$disCovNames, disCovVals = pars$disCovVals, disCovProb = pars$disCovProb,
                    respVCov = 0, workingPath = studydir
)

obsData <- readReplicate(workingdir = studydir)
```


Daiichi-Sankyo

Module 2

```
##############################################################################
## Step 2: 2 wks analysis (study1)                                          ##
##############################################################################


# read and subset observed data (the simulated 16 wks data in step 1)
obsData1 <- obsData[obsData$TIME <= 2*7*24, ]
write.csv(obsData1, file.path(study1dir, "study1.obs.csv"), row.names = FALSE)

# Re-estimate the parameters (if doNONMEM = FALSE, use the default 2 wk parameters)
if (doNONMEM) {
    pars1 <- list()
    for (i in 1:length(unique(obsData1$REP))) {
            study1NMdir <- .verifyFolder(study1dir, "NM", paste("rep", unique(obsData1$REP)[i], sep = ""))
            # NONMEM data preparing -- copy control file of 'run2'
            ctlfile <- file.path(system.file(package = "diabetes"), "unitTests", "testdata", "nmtran", "run2", "run2.ctl")
            file.copy(ctlfile, study1NMdir)
            # NONMEM data preparing -- generate data file by using the observed data
            convNMData(obsData1[obsData1$REP == unique(obsData1$REP)[i], setdiff(names(obsData1), "REP")],
                                    NMNameMap = c(X.ID = "SUBJ", SID = "0", THR = "TIME", STDY = "0", DATE = "0", TIME =
    "TIME", AMT = "DOSE",
                                                    CMT = "CMT", EVID = "0", MDV = "0", DV = "RESP", HV =
    "0", AGE = "0", RACE = "0", SEX = "SEX",
                                                    WT = "0", HT = "0", CRE = "0", CL = "CLI", V2 = "V2I", V3 =
    "V3I", KA = "KAI", Q = "QI",
                                                    ALAG = "ALAGI", FPG = "FPG", HBA = "HBA", NONN =
    "NONN", DGRP = "0", LNDV = "log(RESP)"),
                                    RespName = c("FPG", "HBA"), RespCMT = c(4, 5), OrderName = c("X.ID", "THR", "CMT"),
                                    filename = file.path(study1NMdir, "new.csv"))

            # invoke NONMEM to run the model and estimate the parameters
            runNONMEM(ctlFile = file.path(study1NMdir, "run2.ctl"), nmDir = getOption("nmfe"), cleanFile = TRUE)
            pars1[[i]] <- paras.run2(study1NMdir)
    }
```


Daiichi-Sankyo

Module 2

```
} else {
      pars1 <- list()
      pars1_0 <- paras.run2(nmpath = file.path(system.file(package = "diabetes"), "unitTests", "testdata", "nmtran", "run2_2wk"),
                                    xmloutputfile = "run2_2wks.xml",
                                    csvdatafile = "new2wks.csv"
      )
      for (i in 1:length(unique(obsData1$REP))) {
                  pars1[[i]] <- pars1_0
                  pars1[[i]]$etaDf <- unique(obsData1[obsData1$REP == unique(obsData1$REP)[i], c("SUBJ", "ETA1", "ETA2", "ETA3",  "ETA4",
      "ETA5")])
      }
}

# Do prediction

obsData1 <- read.csv(file.path(study1dir, "study1.obs.csv"))
predictBySim(obsData1, obsData, nmTheta = lapply(pars1, FUN = function(X) X$thetaMean),
                  nmEta = lapply(pars1, FUN = function(X) X$etaDf), times = c(0:112) * 24 ,
                  filename = file.path(study1dir, "study1.pred.csv"))

# Analyze the predicted data
predData1 <- read.csv(file.path(study1dir, "study1.pred.csv"))
for(i in unique(predData1$REP))
      write.csv(predData1[predData1$REP %in% i, ], file = sprintf("%s/ReplicateData/replicate%04d.csv", study1dir, i), row.names = FALSE)
analyzeData(analysisCode = .microCode.run2.module , macroCode = .macroCode.run2.module1 , workingPath = study1dir)
```


Daiichi-Sankyo

Module 2

```
### summarization
con2wks.FPG <- tablePredSummary(outvar = c("FPG"),
                    allDf = predData1, trialvar = "REP", cuttime = 2*7*24,
                    needwrite = doWrite, resultdir = study1dir)

con2wks.HBA <- tablePredSummary(outvar = c("HBA"),
                    allDf = predData1, trialvar = "REP", cuttime = 2*7*24,
                    needwrite = doWrite, resultdir = study1dir)

con2wks.FPG
con2wks.HBA

### plot
if(doWrite) pdf(file = file.path(study1dir, sprintf("cfp[%s].pdf", "2wks")))
plotBox(obsDf = predData1, outvar = "HBA", trialvar = "REP")
plotBox(obsDf = predData1, outvar = "FPG", trialvar = "REP")
if(doWrite) dev.off()



#########################################################################
## Step 3: 4 wks analysis (study2)                                     ##
#########################################################################
```


Daiichi-Sankyo

Demonstration

Daiichi-Sankyo

Summary

- Brief overview of clinical trial simulation and MSToolkit
- Diabetes Toolkit modules are demonstrated
  - Module 1: To compute probability of trial success by evaluating the magnitude of drug effect on output variables (eg., FPG, HbA1c) with respect to their baseline values
  - Module 2: To assess if short term study can replace the required long term study by computing the similarity index using the output variables predicted from short term study data and the long term study values

Acknowledgements

- Mango Solutions
  - Andy Nicholls
  - Jim Ditchburn
  - Jian Li

Daiichi-Sankyo