MSToolkit – An R library for simulating and evaluating clinical trial designs and scenarios

Mike K Smith[1], Richard Pugh[2], Romain Francois[2] 1: Pharmacometrics, Pfizer Ltd, 2: Mango Solutions

Objectives

The ability to quickly and easily simulate clinical trial scenarios is a vital tool in the modern statisticians' inventory. Advances in clinical trial design mean that sample sizing and evaluation of trial performance metrics and operating characteristics often require simulation.

Methods

MSToolkit is an R package that allows the user to quickly simulate clinical trial data and then apply analytical methods to this simulated data to evaluate trial performance. Code associated with assessing trial designs via simulations involves piecing together many different basic elements. Figure 1 shows the six elements of model-based drug development as defined by Lalonde et al (2007).

MSToolkit takes as input the PK/PD, disease models, meta-analysis results, or hypothesised dose-response models. Parameters of these data generation models can vary between simulated trial replicates. Parameters can also be set to vary between individuals. Values are either generated assuming parametric distributions acknowledging correlations between parameters, or they can be sampled from external parameter files.

MSToolkit also defines the trial design characteristics: treatment arms - how many doses, what amounts, how often, crossover sequences of treatments; allocating subjects to those treatment arms. Continuous and discrete covariate values can be generated for subjects within the trial, either assuming parametric distributions or sampled from existing data sources.

Based on all of these input factors, MSToolkit generates response values for each subject. Responses can be continuous, binary or count data; but other forms of response outcome can be generated using user-defined link functions with random value generating functions.

To simulate realistic data one should also consider missing data and dropout processes and the ability to partition generated data into interim analysis subsets. MSToolkit allows for missing data and dropout mechanisms through user-generated response functions, as well as missing at random.

Analytical methods need to be applied to the generated data, results extracted and decision criteria applied in order to assess operating characteristics. The analytical methods may need to be applied once at the end of the trial, or in the case of interim analyses or adaptive trials after fixed proportions of patients have completed. MSToolkit applies the analytical method separately to the data generation process, allowing users to compare several analytical methods against the same generated data. Code is written as though for one single data set and MSToolkit applies this against all of the replicated datasets. Although R is used as the platform for data generation, other tools can be used for data analysis – MSToolkit provides easy linking to SAS as the analytic engine, and by using libraries such as BRugs or R2WinBUGS, MSToolkit can easily be used to assess Bayesian design and analysis problems.

The simulation framework needs to be able to analyse the data and apply decision criteria multiple times, potentially making decisions to "stop" trials or drop doses. Through specifying user-defined code for applying decision criteria this is easily handled by MSToolkit.

Finally, assessments of overall trial performance are evaluated at the end of each trial. These may be as simple as Go / No Go criteria, but can include quantitative assessments of trial-based parameter estimates versus "true" values used in generating data, allowing quantification of bias and precision.

Figure 1: Six elements of model-based Drug development

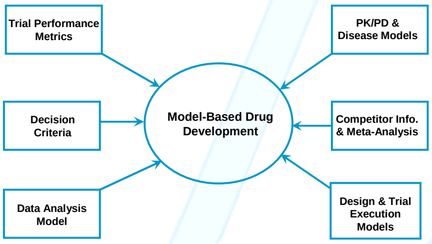

Trial Performance Metrics
PK/PD & Disease Models
Decision Criteria
Model-Based Drug Development
Competitor Info. & Meta-Analysis
Data Analysis Model
Design & Trial Execution Models

Figure 2: MSToolkit can use R, SAS, WinBUGS or other batch-mode software for analysis.

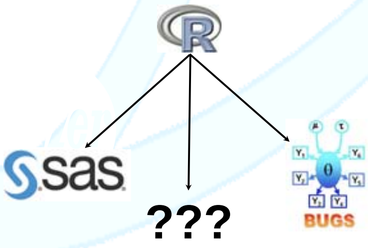
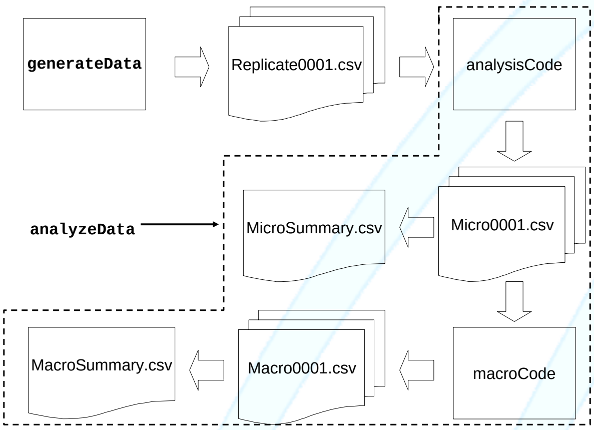


Figure 3 – Schematic of MSToolkit process



System description

MSToolkit has two main functions: generateData(…) and analyzeData(…). The generataData function is a wrapper function for many lower-level functions which generate treatment arms, allocate subjects to treatments, define covariate and model parameter distributions and how these vary between subjects or between trial replicates, or sample values of these from external files. The user-defined response function describes how the various inputs combine to define the expected response for an individual on a given treatment at a given time. To this residual variability is added if appropriate and random outcomes generated. This process generates comma-separated variable (CSV) files. These files are easily read into Excel or any other software for analysis, facilitating QC as well as portability across analytical software.

The user specifies their analysis code within R, which can also call other batch-mode analysis engines, or via SAS. If SAS is used for analysis, MSToolkit pre-processes data files to make them easily accessible within SAS. In either case the user need only write code for the analysis of one dataset at a time. In the case of trials with interim analyses, analysis is first performed on the complete dataset, as well as on individual interim analysis datasets. This allows comparison of trial designs with dose-selection algorithms or stopping rules and the complete dataset without interim analysis decision rules.

MSToolkit handles the data input/output and collation of results – micro-evaluation presenting treatment-level results of analysis and macro-evaluation evaluating the individual trial outcome against decision criteria for Go / No Go decisions. Individual trial CSV files are created for each replicate, as well as simulation scenario level summaries across all replicates.

Data generation in MSToolkit is a single step for all simulation replicates. Analysis of large numbers of replicates can be divided across GRID computing nodes – 1000 simulation analyses can be divided into batches of 100 across 10 GRID nodes, parallelising the analytic step and speeding up the analysis process. MSToolkit handles the splitting of the job and collation of the output from these 10 jobs. Having separate CSV data and output files facilitates this process.

Example Code

```
generateData(replicateN = 100, subjects=398,
treatSubj = c(78,81,81,81,77),
treatDoses = c(0, 5, 25, 50, 100),
genParNames = "ED50,E0,EMAX",
genParMean = c(67.5,2.46,15.13),
genParVCov=c(2423.66, 19.48, 0.56, 227.71, 0.931, 24.71),
genParCrit="0<ED50",
respEqn = "E0+(EMAX*DOSE)/(ED50+DOSE)",
respVCov = 63,
interimSubj="0.33,0.66", seed=12345)

emaxCode<-function(data){
## ANALYSIS function returning
## DOSE, MEAN, SE, LOWER, UPPER, N
with( data, {
uniDoses <- sort( unique(DOSE))
eFit <- emaxalt( RESP, DOSE )
outDf <- data.frame( DOSE = uniDoses, MEAN = eFit$fitpred,
SE = eFit$sdpred)
outDf$LOWER <- outDf$MEAN - 1.96 * outDf$SE
outDf$UPPER <- outDf$MEAN + 1.96 * outDf$SE
outDf$N <- as.vector(table(DOSE))
outDf$DIFF <- eFit$fitpred-eFit$fitpred[uniDoses==0]
outDf$DIFFLOWER <- outDf$DIFF-1.96*eFit$sddif
outDf }) }

interimcode<-function(data){
dropdose <- with( data,{
         DOSE [ DIFF<5 & DOSE != 0] })
outList <- list()
if( length(dropdose) > 0 ) outList$DROP <- dropdose
outList$STOP <- length(dropdose)==nrow(data)-1
outList }

macrocode<-function(data) {
temp<-data[data$INTERIM==0,]
fullsuccess<-temp$DIFFLOWER[temp$DOSE==100]>5
interimsuccess<-min(data$DIFF[data$DOSE==100]>5)>0
success<-fullsuccess&interimsuccess
data.frame( SUCCESS = success, FULL=fullsuccess,
INTERIM=interimsuccess ) }

analyzeData(analysisCode = emaxCode,
interimCode=interimcode, macroCode = macrocode)
```

Results

MSToolkit provides flexibility in specifying trial design components coupled with the ability to analyse data using a variety of analytical engines using the power of parallel processing in a GRID environment.

One of the main strengths of MSToolkit is that it provides a common framework for simulations allowing statisticians to share code and quickly understand the mechanisms and analytical techniques used by others in simulations. There is flexibility in the type of designs that can be simulated, the models used to generate the data, and the type of data generated. Balancing flexibility and ease of use often proves difficult, but MSToolkit seems to strike a balance – allowing generation of simple data cases with very few lines of code, while more complex scenarios can be specified through user-defined functions and combinations of function arguments.

While the data generation functionality defined within MSToolkit is deep, the analytic functionality is sparse – users must specify their own analytical methods. This gives users much greater flexibility, but also forces them to consider the robustness of the analytical method and how it applies to the generated data. Similarly the decision criteria applied to the analysis results for Go / No Go decision making, dropping doses or stopping studies at interim analyses are left to the users to define. Analytical methods and decision criteria are key to trial performance metrics and it is right for users to spend the bulk of their time considering these aspects. Code used to analyse simulation results can easily be used in the analysis of the "real" trial, so effort is not wasted.

The ability to use GRID architecture means that users benefit from parallelising the time consuming task of analysis. Thus initial simulations may be performed on desktop machines, while full evaluations utilising 1000s of simulation replicates can easily be submitted to high performance computing resources.

Figure 4: ReplicateData stored in CSV files

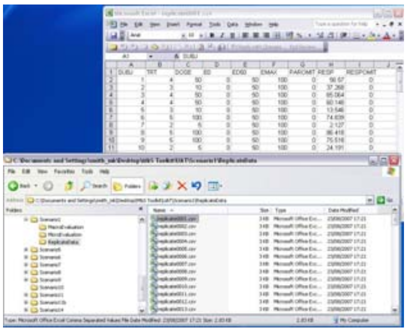
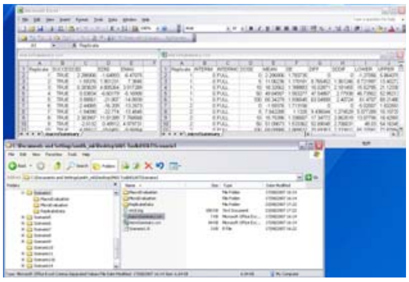


Figure 5: Micro- and Macro-evaluation summaries



Discussion and conclusions

The MSToolkit library was designed to facilitate sharing of code by providing a common framework for data generation allowing flexibility in defining trial designs and data generation models, and a data analysis engine that applies user-defined analytical methods and decision criteria and takes care of the data handling and house-keeping aspects. The main aim of the MSToolkit was to reduce the overheads in coding for statisticians performing simulations to allow them to concentrate on design features, analytical methods and decision criteria. Our experience in using this package at Pfizer is that this aim has been achieved.

References

Lalonde et al (2007) Model-based drug development. Clin Pharm Ther. **82**, pp21-32.