An R Package for Simulation Experiments Evaluating Clinical Trial Designs

Yuanyuan Wang[1] and Roger Day[1, 2]
[1]Department of Biostatistics
[2] Department of Biomedical Informatics
University of Pittsburgh, Pittsburgh, PA 15213, USA

Abstract: This paper presents an open-source application for evaluating competing clinical trial (CT) designs using simulations. The S4 system of classes and methods is utilized. Using object-oriented programming provides extensibility through careful, clear interface specification; using R, an open-source widely-used statistical language, makes the application extendible by the people who design CTs: biostatisticians. Four key classes define the specifications of the population models, CT designs, outcome models and evaluation criteria. Five key methods define the interfaces for generating patient baseline characteristics, stopping rule, assigning treatment, generating patient outcomes and calculating the criteria. Documentation of their connections with the user input screens, with the central simulation loop, and with each other faciliates the extensibility. New subclasses and instances of existing classes meeting these interfaces can integrate immediately into the application. To illustrate the application, we evaluate the effect of patient pharmacokinetic heterogeneity on the performance of a common Phase I "3+3" design.

1. Introduction

In the past few decades, CT designers have proposed hundreds of designs for trials at different stages of drug development, ranging from preclinical to Phase IV trials. This abundance of designs mandates a question for the investigators and statisticians planning a trial: what design would be the "best" for their trial? Determining the answer must begin with careful consideration of the criterion by which to judge designs, which should reflect the goals of the trial.

CT simulation is used by academic research centers and pharmaceutical companies to improve the efficiency and informativeness of drug development [1-4]. Sophisticated commercial software for trial simulations is available for those with resources to cover fees and with design challenges matching the software's capabilities. Academic research centers usually use locally developed software mainly due to cost and flexibility considerations. Cost issues are obvious. Flexibility is needed primarily to explore novel designs and novel evaluation criteria. This local software development focuses on answering specific research questions in compressed time frames, and is not routinely sharable. Inspired by the success of open-source software development projects, we are building an open-source simulation experiment platform with the intention of harnessing the power of distributed peer review and transparency of process. Techniques in S4 classes and methods [5] are utilized to make our package trustworthy, clear and extendible.

In the spirit of personalized therapy, our simulation experiment platform acknowledges differences among study participants in three aspects. The population model for patients' baseline characteristics may be a mixture model (for example, reflecting pharmacogenetic heterogeneity). The outcome model may depend on patient-specific characteristics, and trial designs may allow treatment assignments to depend on measured or assayed patient baseline characteristics (for example, clinical, pathological, physiologic, genetic, or pharmacokinetic characteristic).

Our vision for this simulation experiment platform is to provide a framework in which new trial designs, new models, and new evaluation criteria are easily accommodated. The scope of this framework is reviewed in the discussion section. In the following sections, we will describe the framework and its implementation using S4 classes and methods in detail, and to illustrate the application, we evaluate the effect of patient pharmacokinetic heterogeneity on the performance of the standard Phase I "3+3" design without dose de-escalation. This example introduces a new evaluation criterion for Phase I trials mapping from "recommended dose" to the chance of future success in the drug development.

2. CT Simulation Experiment Framework

To begin a CT simulation experiment, besides the number of simulations of a CT, the experimenter needs to provide specifications for four major components: population models, CT designs, outcome models and evaluation criteria.

The population model specification contains all the necessary information to obtain the values for a participant's baseline characteristics which may affect either the treatment assignment decision or the patient's outcome. The design specification contains all design parameter values for a specific design. The outcome model specification provides the model

parameter values for a specific outcome model which generates each participant's clinical outcome based on his/her baseline characteristics and treatment assignments. The evaluation specification includes parameter values if any for a specific evaluation criterion.

An experiment consists of three loops over the population models, designs, and outcome models of interest. Within these outermost loops is a loop over CT simulations, and within this loop is the loop over patients simulated for a single CT simulation. The results are evaluated by the specified criteria at the end of simulations under a particular design, population model and outcome model.

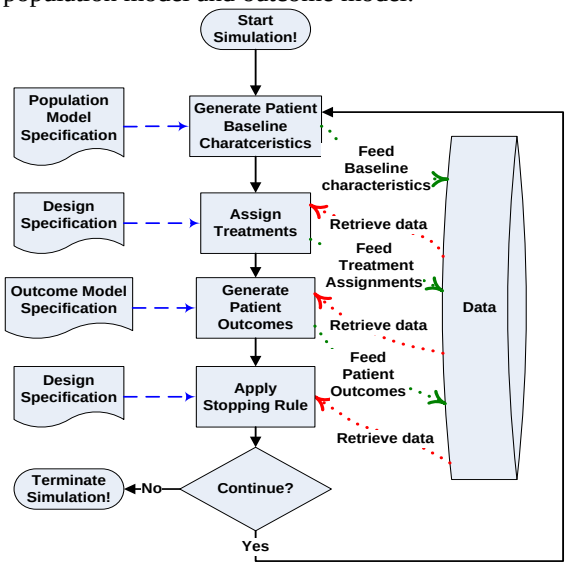

Figure 2.1 Central Simulation Loop

We display the central simulation loop in Figure 2.1. The solid black arrows trace the overall simulation flow. The dashed blue arrows show how the specifications are associated with the actions on the flow. The dotted red and green arrows show exchange of data between actions and the loop's temporary data repository (which is **not** intended to represent a CT database). As the simulation starts, a new patient (or in some cases *n* patients as a group) enters the CT. The loop generates this new patient's baseline characteristics and then assigns treatments based on the design, new patient's baseline characteristics, and (if applicable) accumulated data from previous patients. A patient's outcomes are generated from his/her baseline characteristics and treatment assignments. After the loop simulates outcome data for this new patient, it applies the stopping rule to decide whether to terminate the simulation or continue the simulation by enrolling another new patient. The simulation continues until the stopping rule concludes that it must terminate.

3. Framework Implementation

3.1 *S4 Classes and Methods*

R is a language and environment for statistical computing and graphics. The R language has functional-programming semantics, whereas it supports the object-oriented programming (OOP) style. R has two different OOP systems, known as S3 and S4. Compared to S3, S4 is more formal, rigorous and closer to the "traditional" OOP other languages like Java and Python follow [5]. The main difference between S4 and "traditional" OOP is that the method definitions in the S4 system do not reside in a class definition and methods sharing common conceptual properties and thus the same name are stored within the generic function according to their signature, a named list of classes with the names corresponding to the formal arguments of the function. With the promise of making software trustworthy, clear and extendible, S4 classes and methods are highly encouraged in the R software development [5].

The classes and methods discussed next are the ones participating in the central simulation loop as seen in the Figure 2.1, and the ones for evaluation. For better illustration, we describe methods under the classes they are closely associated with. We summarize the key classes along with their definitions in the Table 3.1.

3.1.1 *Class: "DesignSpecifier"*

"DesignSpecifier" is a class union with member classes (a special case of subclasses) representing the objects of specification for a specific CT design. These classes contain slots for the design parameters. For example, a class "APlusBNoDeEscSpecifier" represents objects of specification for A+B without dose de-escalation design described in Lin and Shih's paper [6], and it contains slots for tier doses, the initial cohort size (A), the additional cohort size (B) and several slots (C, D, E) for the dose-limiting toxicity (DLT) counts that are associated with stopping and dose assignments for the next group of patients.

Two methods are closely associated with the subclasses of "DesignSpecifier": "applyStoppingRule" and "assignTrt", which both have a subclass of "DesignSpecifier" and a list representing accumulated CT data in signature. "applyStoppingRule" produces a list with three elements: a Boolean decision on whether to stop the trial, conclusions if the trial is to stop and the number of additional patients to enroll as a group if the trial is to continue. The "assignTrt" method has one additional argument corresponding to current patient baseline characteristics and this method generates treatment assignments for the current patient.

3.1.2 *Class: "BaseCharModelSpecifier"*

"BaseCharModelSpecifier" is a class that represents objects of model specification for generating a specific baseline characteristic. It contains slots for this baseline characteristic name, the names of other baseline characteristics which are involved with the generation of this baseline characteristic in the model, and a string of user-defined R generating function by which we assume nothing about the model for generating this baseline characteristic.

The "generateBaseChar" method has two arguments corresponding to an instance of "BaseCharModelSpecifier" and a list of values for other baseline characteristics that are involved with the generation. This method generates a single baseline characteristic for a patient.

3.1.3 *Class: "PopModelSpecifier"*

"PopModelSpecifier" is a class that represents objects of a population model specification. It has a slot to hold a list of "BaseCharModelSpecifier" objects.

The "generateBaseChars" method has only one argument corresponding to an instance of "PopModelSpecifier". When generating a new selected patient's baseline characteristics, each of the "BaseCharModelSpecifier" objects is utilized by the method "generateBaseChar" in turn. We facilitate expression of the total joint distribution of characteristics by means of sequential conditional distributions.

3.1.4 *Class: "OutcomeModelSpecifier"*

"OutcomeModelSpecifier" is a class union with member classes representing objects of specification for a specific outcome model. The member classes contain slots for outcome model parameters if any and one auxiliary slot if otherwise (S4 regards a class with no slots as a virtual class and does not allow for the instantiation from a virtual class). For example, class "ToxDoseThresholdModelSpecifier" is one of the member classes of "OutcomeModelSpecifier", which represents objects of specification for the toxicity dose threshold model. This model can be described using the following equation:

$$y = \begin{cases} T & \text{if } z > \theta_T \\ t & \text{if } z \le \theta_T \end{cases}$$

where y denotes the outcome, T denotes toxicity response, t denotes non-toxicity response, z refers to dose assignment, θ_T is the dose threshold for toxicity response respectively. No parameters are in this model.

The "generateOutcomes" method has arguments corresponding to an instance of member class of "OutcomeModelSpecifier", current patient's baseline characteristics and treatment assignments. It produces outcomes for a patient.

3.1.5 *Class: "EvalSpecifier"*

"EvalSpecifier" is a class union with member classes either directly representing or being the superclasses of the classes that represent the objects of specification for a specific evaluation criterion. The member classes or subclasses of member classes have slots for the evaluation criterion parameters if any and one auxiliary slot if otherwise. For example, "EvalNPatAtTierDose" class represents objects of specification for evaluating the number of patients at some tier dose. This class contains one slot for this tier dose.

The "evaluateDesign" method utilizes information from an instance of a member class of "EvalSpecifier", simulated data and conclusions from simulated CT to produce the result from a single evaluation criterion.

Class	Definition
"DesignSpecifier"	A class union with member classes representing the objects of specification for a specific CT design
"BaseCharModelSpecifier"	A class that represents objects of model specification for generating a specific baseline characteristic
"PopModelSpecifier"	A class that represents objects of a model specification for generating a patient's baseline characteristics which may affect either the treatment assignment decision or the patient's outcome
"OutcomeModelSpecifier"	A class union with member classes representing objects of specification for a specific outcome model
"EvalSpecifier"	A class union with member classes either directly representing or being the superclasses of the classes that represent the objects of specification for a specific evaluation criterion.

Table 3.1 Class Definition

3.2 ***Interface: requirements and provisions***

Extensibility of our platform requires that a user, with a novel design, novel model (population or outcome) or novel evaluation criteria challenge, can develop new subclasses of the key classes and their associated methods, assuring that they will work together. The classes and methods described in the previous section pass information packets whose exact structure will vary in different experiments. We describe below two examples that show the "requirements and provisions" relationship among classes and methods.

Example 1: The "applyStoppingRule" method associated with the "APlusBNoDeEscSpecifier" class requires toxicity outcomes for the last A or A+B patients, experiment will be halted with an

appropriate error message if the "generateOutcomes" method fails to provide toxicity outcome.

Example 2: The "generateOutcomes" method associated with "ToxDoseThresholdModelSpecifier" class requires toxicity dose threshold from the "generateBaseChars" method, error will come out if the "generateBaseChars" method fails to provide the toxicity dose threshold.

Thus, a central component of the open-source strategy is to document these requirements and provisions for each class and method, to facilitate and encourage sharing of innovation.

4. Example

4.1 *Objective*

In this experiment, we are trying to evaluate the effect of patient pharmacokinetic heterogeneity on the performance of the standard Phase I "3+3" design without dose de-escalation. We represent pharmacokinetic heterogeneity in terms of bimodality of the distribution for toxicity dose threshold. The parameter values in the population model are selected only for illustration purposes.

4.2 *Experiment Set-up*

Design: Standard Phase I "3+3" design without dose de-escalation, with tier doses following the modified Fibonacci sequence: 3, 6, 10, 13, 15.

Population Model 1 (Unimodal): The one baseline characteristic produced is a toxicity dose threshold, which follows a lognormal distribution whose mean is 13, the 4th dose:

$\log\theta_T \sim N(\mu = \log(13), \sigma^2 = 0.1^2)$.

Population Model 2 (Mixture): The previous population model 1 is mixed with a small subpopulation at much higher risk of toxicity in the ratio 0.9 to 0.1. The toxicity dose threshold in this subpopulation follows a lognormal distribution:

$$\log\theta_T \sim N(\mu = \log(3), \sigma^2 = 0.1^2)$$

Outcome Model: The outcome has one Boolean component. Dose-limiting toxicity (DLT) occurs if the dose exceeds the patient's toxicity threshold.

Evaluation Criteria:

- Number of patients enrolled in the CT
- Number of patients experiencing DLT
- Probability of success in Phase II trial at recommended dose (RD), where the success is defined by concluding that the drug is worthy for further studies in the Phase II trial

The future Phase II trial used to represent a medical research beneficial outcome refers to a trial using a two-stage Bryant and Day design [7], where both toxicity and efficacy outcomes are considered. The design parameters are [8]: for the first stage, the sample size is 19 patients, the cut-off values for efficacy and non-toxicity outcomes are 5 and 11 respectively; for both stages together, the sample size is 33 patients, the cut-off values for efficacy and non-toxicity outcomes are 12 and 22 respectively. The underlying model for dose thresholds for toxicity and efficacy responses for Phase II patients are assumed to follow bi-lognormal distribution:

$$(\log\theta_T, \log\theta_E) \sim N(\mu = \begin{pmatrix}\log(13)\\ \log(10)\end{pmatrix}, \Sigma_\theta = \begin{pmatrix}1 & 0.4\\ 0.4 & 4\end{pmatrix})$$

We simulate 1000 replications of Phase I trials as described above under two scenarios where scenario 1 uses unimodal population model and scenario 2 uses mixture population model.

4.3 *Results*

	3	6	9	12	15	18	21	24
Scenario 1	0	0	0	508	430	61	1	0
Scenario 2	5	36	77	316	352	165	44	5

Table 4.1 Number of Patients Enrolled by Scenarios

	2	3	4	5	6
Scenario 1	537	383	79	1	0
Scenario 2	385	371	183	49	12

Table 4.2 Number of Patients Experiencing DLT by Scenarios

	0	3	6	10	13
Scenario 1	0	0	0	815	185
Scenario 2	21	85	80	699	115

Table 4.3 RD by Scenarios

RD	0	3	6	10	13
Pr(Success)	0	0.09	0.51	0.15	0.02

Table 4.4 Probability of Success at RD

The above Tables 4.1-4.3 list the distribution of number of patients enrolled, number of patients experiencing DLT, and RDs under two scenarios respectively. RD is set to be zero when lower dose than starting dose needs to be claimed as RD. The average number of patients enrolled is 13.7 under scenario 1, and 14.0 under scenario 2. The average number of patients experiencing DLT is 2.5 under scenario 1, and 2.9 under scenario 2. The addition of a small subpopulation of highly toxicity sensitive patients causes the Phase I trials to recommend a lower dose more frequently. Table 4.4 shows the probability of a successful phase II trial at each RD. Given our specified Phase II design and dose thresholds distribution for the patients in the Phase II trials, the dose of 6 is associated with the highest probability of a successful Phase II trial. Summarizing across the trials, the average probabilities of success in phase II trial for the two scenarios are 13% under scenario 1, and 16% under scenario 2. We do not observe any obvious different

results between two scenarios based on the specified evaluation criteria in this example. Therefore, this example doesn't provide evidence that patient pharmacokinetic heterogeneity affect the performance of the standard Phase I "3+3" design without dose de-escalation.

5. Discussion

The framework presented here covers most CT designs. There are some exceptions. Our framework cannot cover CTs where treatment assignments may occur more than once to a patient, for example, in studies with re-randomization of a subset depending on intermediate clinical outcomes. As for the other components, there are no obvious limitations beyond the willingness of contributors to construct subclasses reflecting the models and criteria they prefer.

CT designs should consider both the potential risks/benefits for the patients on the trial and the potential scientific/medical value of the knowledge to be gained. In our example, we assessed the scientific or medical value (U_S) by the probability of success of a subsequent Phase II trial at the RD produced from the Phase I trial. Since a Phase I trial is often a first-in-human trial, the ethical concerns are critical. Operating characteristics which reflect the effects on the enrolled patients focus primarily on tabulations of adverse events [9, 10]. We could and probably should consider the balance among various societal and patient utilities and disutilities. This simulation experiment platform provides a way to do that. For example, if the patient has two possible outcomes, *T* (toxicity) and *t* (non-toxicity), the patient utility can be represented by:

$$U_P = \sum_{i=1}^{n} \sum_{h_i=T}^{t} u(h_i)$$

where $u(h_i)$ is the utility for the outcome h_i. Total utility is measured by the sum of patient utility, societal utility and total sampling cost:

$$U_{Tot} = \lambda_P U_P + \lambda_S U_S + \lambda_C n u_C$$

where u_C is the cost per patient, and the λ's are multipliers which convert among the different kinds of utilities.

Commercial simulation platforms for evaluating CT designs are out of reach for many academic centers conducting clinical research. Academic centers also have innovative CT methodology researchers facing an extremely varied array of design challenges. To disseminate the innovation, expand design capabilities and enhance efficiency, an open source solution is urged.

Future development of this system will include publishing it as an open-source R package, setting up an archive site for users to contribute new subclasses and corresponding methods (new designs, population models, outcome models, and criteria), a user-friendly GUI, user evaluation by CT designers. Complex factorial evaluations will be computationally time-consuming. Parallelization for these demanding tasks is possible, for example, several R packages provide support for parallel processing [11].

References

1. Santen G, van Zwet E, Danhof M, Della Pasqua O. From Trial and Error to Trial Simulation. Part 1: The Importance of Model-Based Drug Development for Antidepressant Drugs. Clin Pharmacol Ther. 2009;86(3):248-54.
2. Hale M, Gillespie W, Gupta S, Tuk B, Holford N. Clinical Trial Simulation as a Tool for Increased Drug Development Efficiency. Applied Clinical Trials. 1996;5:35-40.
3. Holford NHG, Kimko HC, Monteleone JPR, Peck CC. Simulation of Clinical Trials. Annual Review of Pharmacology and Toxicology. 2000;40:209-34.
4. Lockwood P, Ewy W, Hermann D, Holford N. Application of clinical trial simulation to compare proof-of-concept study designs for drugs with a slow onset of effect; an example in Alzheimer's disease. Pharm Res. 2006;23(9):2050-9.
5. Chambers JM. Software for Data Analysis. New York: Springer Science+Business Media, LLC; 2008.
6. Lin Y, Shih WJ. Statistical Properties of the Traditional Algorithm-Based Designs for Phase I Cancer Clinical Trials. Biostatistics. 2001;2:203-15.
7. Bryant J, Day R. Incorporating Toxicity Considerations Into the Design of Two-Stage Phase II Clinical Trials. Biometrics. 1995;51(4):1372-83.
8. Bryant J, Day RS. Available from: http://www.upci.upmc.edu/bf/resources.cfm.
9. Reiner E, Paoletti X, O'Quigley J. Operating Characteristics of the Standard Phase I Clinical Trial Design. Computational Statistics & Data Analysis. 1999;30:303-15.
10. Ahn C. An Evaluation of Phase I Cancer Clinical Trial Designs. Statistics in Medicine. 1998;17:1537-49.
11. Eddelbuettel D. *High-Performance and Parallel Computing with R* Available from: http://cran.r-project.org/web/views/HighPerformanceComputing.html.