Keyboard: An R Package Suite for Early Phase Dose-finding Designs

Hongying Sun
University of Rochester Medical Center Department of Biostatistics and Computational Biology
Chen Li (lc.biosta@qq.com)
Fourth Military Medical University
Cheng Cheng
St Jude Children's Research Hospital Department of Biostatistics
Tang Li
St Jude Children's Research Hospital Department of Biostatistics
Haitao Pan
St. Jude Children's Research Hospital

Software

Keywords: Maximum tolerated dose, optimal biological dose, dose-finding, phase I/II trials, Bayesian adaptive design.

Posted Date: April 8th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-400107/v1

Keyboard: An R Package Suite for Early Phase Dose-finding Designs

Hongying Sun[Δ]
Department of Biostatistics and Computational Biology, University of Rochester Medical Center
Chen Li[Δ]
Department of Health Statistics, School of Preventive Medicine, Fourth Military Medical University
Cheng Cheng, **Li Tang**[*], **Haitao Pan**[*]
Department of Biostatistics, St. Jude Children's Research Hospital

Abstract

Background: Phase I and/or I/II oncology trials are conducted to find the maximum tolerated dose (MTD) and/or optimal biological dose (OBD) of a new drug or treatment. In these trials, for cytotoxic agents, the primary aim of the single-agent or drug-combination is to find the MTD with a certain target toxicity rate, while for the cytostatic agents, a more appropriate target is the OBD, which is often defined by consideration of toxicity and efficacy simultaneously. However, there still lacks accessible software packages to achieve both yet.

Results: Objective of this work is to develop a software package that can provide tools for both MTD- and OBD-finding trials, which implements the Keyboard design for single-agent MTD-finding trials by Yan et al., the Keyboard design for drug-combination MTD-finding trials by Pan et al., and phase I/II OBD-finding method by Li et al., in a single R package, called `Keyboard`. For each of the designs, the `Keyboard` package provides corresponding functions that begins with `get.boundary(···)` to determine the optimal dose escalation and de-escalation boundaries, that begins with `select.mtd(···)` to select the MTD when the trial is completed, that begins with `select.obd(···)` to select the OBD at the end of a trial, and that begins with `get.oc(···)` to generate the operating characteristics.

Conclusions: The developed `Keyboard` R package provides convenient tools for designing, conducting and analyzing single-agent, drug-combination and phase I/II dose-finding trials, which supports Bayesian designs of innovative dose-finding studies.

Keywords: Maximum tolerated dose; optimal biological dose; dose-finding; phase I/II trials; Bayesian adaptive design.

1. Introduction

A phase I clinical trial is critical in new drug/treatment development because it determines the dose that will be further investigated in the subsequent phase II or III trials. For the *cytotoxic* agent, one primary objective of a phase I dose-finding trial is to find the *maximum tolerated dose* (MTD), which is defined as the highest dose that has a dose-limiting toxicity (DLT) rate less than or close to a prespecified target rate. The identified MTD will then be employed in later phases, for example, phase II clinical trials. Statistical methods for the single-agent MTD-finding designs include the algorithm-based designs, such as the 3 + 3 design[1], the biased-coin design[2, 3, 4]; the model-based designs, such as the continual reassessment method (CRM)[5] and related methods[6], and the model- assisted design designs, such as mTPI[7], BOIN[8], Keyboard[9] designs, etc.

It should be noted that all the above methods were developed for the *cytotoxic* agent to find the MTD. The cytotoxic drug development, however, is rested on the premise that agents must be "cytotoxic" to be effective. By equating efficacy with the toxicity, traditional drug development, progressing from phase I through phase III clinical studies, has sought the highest effective dose that does not induce intolerable levels of toxicity, which essentially justifies the goal of finding the MTD for the phase I dose-finding trials. This assumption may be reasonable for cytotoxic agents, however, may not hold for the *cytostatic* or *molecularly targeted* agents. For example, the chimeric antigen receptor (CAR) T therapies require a balance of a boosting of the immune system to combat cancer while avoiding over-stimulation. In this case, preliminary dose exploration should aim to capture effective biologic activity rather than dose-limiting toxicity alone. Therefore, the *optimal biological dose* (a.k.a, OBD), which is defined as the lowest dose with the highest rate of efficacy while safe, is a more appropriate endpoint than the MTD since it takes accounts of both the toxicity and efficacy without the above toxicity-and-efficacy-simultaneously-increase-with-dose assumption. In literature, the associated method is termed as the phase I/II OBD-finding design. There are many proposed methods for the OBD-finding trials, for example the model-based methods[10, 11, 12, 13, 14] and model-assisted methods[15, 16, 17, 18, 19, 20] Among these designs, Li et al.[15] proposed

a toxicity and efficacy probability interval (TEPI) design, which was shown to have desirable operating characteristics and is simple and transparent to clinicians with a physician-elicited decision table that maps the two-dimensional probability intervals to a set of dosing decisions. The `Keyboard` package develops multiple R functions to implement this design since the method behind this design is also interval-based which is the core for the Keyboard design.

Except the single-agent trials, nowadays treating patients with a combination of agents is prevalent in clinical trials, especially for the oncology treatments as they are more effective and less susceptible to drug resistance than are single-agent trials. Trial designs for drug-combination studies involve several distinct features that are beyond the scope of methods for single-agent studies. A major challenge in designing combination trials is that dose combinations are only partially ordered in terms of toxicity probability. For instance, consider a trial combining m doses of agent A and n doses of agentthat is, we now have a $n \times m$ dose matrix, a major challenge in designing combination trials is that dose combinations are only partially ordered in terms of toxicity probability, that is, a priori, we cannot fully rank $n \times m$ dose matrix from low to high by their toxicity probabilities. Various designs have been proposed for the drug-combination MTD-finding trials: a design based on the order of the restricted inference[4], a copula-type regression model [19], latent contingency tables[20], the partial order CRM method (POCRM)[21], sequential dose-finding strategy[22, 23], a Bayesian optimal interval design[24], Keyboard combination design[27], and Bayesian data augmentation for late-onset toxicity[28], and among others. The `Keyboard` package implements the Keyboard combination design proposed by Pan et al.[27].

In sum, we develop an R package, `Keyboard`, to include three methods in [9, 15, 27] for practitioners' easy access to these methods. Since all of these methods use the Bayesian interval-based method, which is the underlying theoretical basis for the Keyboard design, therefore, we collect these three methods in one package.

The paper is organized as follows. We concisely introduce the three designs, Keyboard single-agent method for the MTD[9], Keyboard combination design [27], and Keyboard phase I/II OBD design[15] in Section 2. Section 3 describes the functions for implementing these methods. Section 4 provides three exemplary trials and demonstrates how to use the `Keyboard`

package to design trials for single agent, drug combination of MTD- or OBD-dose finding. Section 5 ends in a conclusion.

2. Implementation

2.1. Design for the MTD dose-finding single-agent trials

The design in this package for the single-agent MTD-finding was proposed by Yan et al. [9], and its theoretical properties were further explored by Pan et al. [27]. We call this design single agent Keyboard design, or simply, the Keyboard design. This design uses the toxicity probability's posterior distribution to guide dose transition. To decide whether to escalate or de-escalate the dose, we need to first find the strongest key, which is the interval with the maximum posterior probability. If this key is to the left of the "target key", then we escalate the dose because the data suggest that the current dose is most likely too low; if this key is to the right of the target key, then we de-escalate the dose because the observed data suggest that the current dose is likely too toxic; and if the strongest key is the target key, then we stay at the current dose because the observed data support the notion that the current dose is most likely to be in the right dose interval (See Figure 1).

Let ϕ be the target toxicity rate specified by the investigator and $p_d \in (0,1)$ denote toxicity probability of dose level $d \in \{1, \cdots, D\}$. At any time point, assume n_d patients have been treated at the current dose d, and y_d of them experienced dose-limiting toxicity (DLT). Assuming that $D_d = (n_d, y_d)$ given the observed data, a beta-binomial model is used: $y_d \mid p_d \sim Binomial(p_d, n_d)$, $p_d \sim Uniform(0,1)$. Given D_d, the toxicity distribution is $p_d|D_d \sim Beta(y_d + 1, n_d - y_d + 1)$. By specifying a target toxicity interval (is also called toxicity equivalent interval in the mTPI design) $\mathcal{I}_{\text{target}} = (\phi - \epsilon_1, \phi + \epsilon_2)$, which is named as a *target key* in the Keyboard design, here ϵ_1 and ϵ_2 are small positive values, for instance, both are 0.05. Then, using the width of the target key divides [0,1] to a series of equally-wide keys/intervals, denoted by $\mathcal{I}_k$.

Dose-transition rules of the Keyboard design are as follows:

Table 1: Pre-tabulated decision rules of Keyboard design for single-agent

	Number of patients treated at the current dose															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\phi = 0.2$ with the target key = (0.17, 0.23)																
Escalate if number of DLTs $\leq$	0	0	0	0	0	1	1	1	1	1	1	2	2	2	2	2
de-escalate if number of DLTs $\geq$	1	1	1	1	2	2	2	2	3	3	3	3	3	4	4	4
$\phi = 0.3$ with the target key=(0.25, 0.35)																
Escalate if number of DLTs $\leq$	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3
de-escalate if number of DLTs $\geq$	1	1	2	2	2	3	3	3	4	4	4	5	5	5	6	6

- To decide whether to escalate or de-escalate the dose, the Keyboard design calculates the *strongest key* $\mathcal{I}_{\max}$, which is defined as:

$$\mathcal{I}_{\max} = \underset{k\in\{1,\ldots,K\}}{\operatorname{argmax}} \{\Pr(p_d \in \mathcal{I}_k \mid D_d)\}.$$

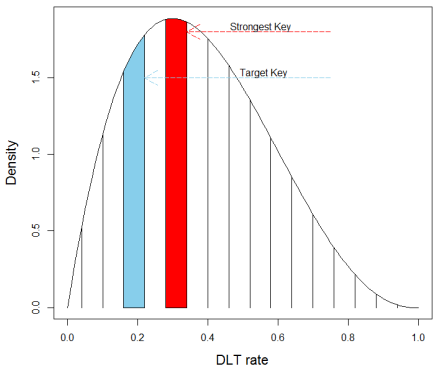

Figure 1: Keyboard design dose-escalation schema

  - if $\mathcal{I}_{max} \succ \mathcal{I}_{\text{target}}$, then p_d is most likely to be overdosing
  - if $\mathcal{I}_{max} \prec \mathcal{I}_{\text{target}}$, then p_d is most likely to be underdosing
  - if $\mathcal{I}_{max} \equiv \mathcal{I}_{\text{target}}$, then p_d is most likely to be the proper dose

A desirable feature of the Keyboard design is that the dose assignment rules can be provided before conducting the trial. Therefore, no real-time computation is required during the trial. Decisions to escalate or de-escalate the dose are solely based on the pre-tabulated

decision rules, which need just number of patients and number of patients experiencing DLT at various dose levels. Table 1 gives an example.

For the patient safety, the following **safety rules** is applied to the Keyboard design in practice:
(i) if at least 3 patients have been treated at a given dose and the observed data indicate that the probability of the current dose's toxicity rate exceeding the target toxicity rate by more than 95%, then we eliminate the current dose and any higher doses from the trial to avoid exposing future patients to unacceptably toxic doses. (ii) if the lowest dose is unacceptably toxic, then the trial terminates early and no dose is selected as the MTD.

2.2. Keyboard design for drug-combination MTD-finding trials

Drug-combination trials are challenging due to possible partial orderings among the dose combination (see Figure 2) and the larger search space (e.g., the dimension of the dose space expands in a multiplicative manner).

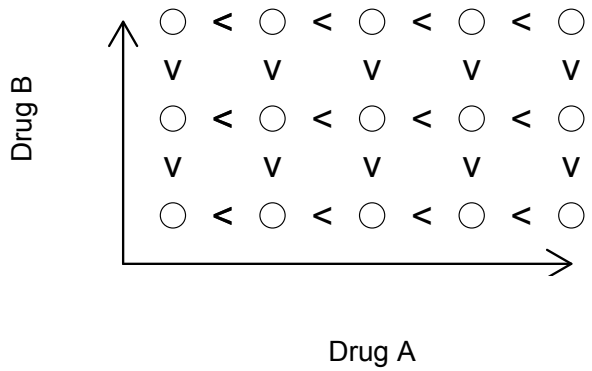

Figure 2: Two dimensional drug-combination trial

Though there are several proposed combination designs in literature [12, 20, 27] and their associated statistical performances demonstrated superior theoretically and empirically, their usages are still limited in practice possibly due to either statistical or computational complexity. The Keyboard combination design proposed by Pan et al. in [27] circumvents the

challenges by extending the above-introduced single-agent Keyboard design to the dual-agent setting.

Let p_{jk} be toxicity probability of the dual agents agent A at level j and agent B at level k, n_{jk} is the number of subjects treated at the dose combination (j,k) and y_{jk} is the number of subjects who experienced DLT at the dose combination (j,k).

Algorithm of the Keyboard combination design is as below:

Step 1. Treat the first cohort of subjects at the dose combination (1, 1).

Step 2. At the dose combination (j,k), given the observed data $D_{jk} = (n_{jk}, y_{jk})$, find the strongest key $\mathcal{I}_{\max}$ based on the posterior distribution of p_{jk}:

If $\mathcal{I}_{\max} \prec \mathcal{I}_{\text{target}}$, then we escalate the dose; if $\mathcal{I}_{\max} \succ \mathcal{I}_{\text{target}}$, then we deescalate the dose;

otherwise, if $\mathcal{I}_{\max} \equiv \mathcal{I}_{\text{target}}$, then we stay at the current dose.

Step 3. The process continues until the pre-specified maximum sample size N is achieved.

Here, $\mathcal{I}_1 \prec (\succ) \mathcal{I}_2$ means that $\mathcal{I}_1$ is at the left (right) of the $\mathcal{I}_2$.

However, for combination trials with dual-agents, the difficulty is that when we decide to escalate/de-escalate a dose, there usually exist more than one option, for example, if the decision is to escalate, we may move either dose level of agent A, dose level of agent B, or both simultaneously, to it (their) adjacent higher dose level(s).

In [27], the authors defined the admissible dose escalation/de-escalation sets as:

- $\mathcal{A}_{E_1} = \{(j+1,k),(j,k+1)\}$ and $\mathcal{A}_{D_1} = \{(j-1,k),(j,k-1)\}$
- $\mathcal{A}_{E_2} = \{(j+1,k),(j,k+1),(j+1,k+1)\}$ and $\mathcal{A}_{D_2} = \{(j-1,k),(j,k-1),(j-1,k-1)\}$

For instance, $\mathcal{A}_{E_1}$ means that escalation can be allowed to move the agent A from j to $j+1$, or move the agent B from k to $k+1$. We can see that the diagonal movement is prohibited in escalation/de-escalation due to safety concerns.

The dose assignment algorithms is then as follows:

- Escalation: escalate to the dose combination that belongs to $\mathcal{A}_{E_1}$ and has the highest value of $\Pr(p_{j'k'} \in \mathcal{I}_{\text{target}}|D_{jk})$ where $(j', k') \in \mathcal{A}_{E_1}$.

- De-escalation: de-escalate to the dose combination that belongs to $\mathcal{A}_{D_1}$ and has the highest value of $\Pr(p_{j'k'} \in \mathcal{I}_{\text{target}}|D_{jk})$ where $(j', k') \in \mathcal{A}_{D_1}$.

If there are multiple optimal dose combinations with the same value of $\Pr(p_{j'k'} \in \mathcal{I}_{\text{target}}|D_{jk})$, then we randomly choose one. The trial is completed when the maximum sample size is reached. Given all observed data at the completion of a trial, we use the matrix isotonic regression to obtain the estimates of p_{jk}'s value and select the MTD as the combination with a estimated toxicity probability that is closest to the target. Similar **safety rules** to the single-agent Keyboard design are also used here.

2.3. Design for the single-agent OBD-finding trials

Traditionally, the purpose of a dose-finding design in cancer is to find the maximum tolerated dose (MTD) based solely on the toxicity. However, for the molecular targeted agents, little toxicity may arise within the therapeutic dose range and the dose-response curves may not be strictly monotonic. This challenges the conventional principle of "more is better". Instead, the optimal biological dose (OBD), which is defined as the lowest dose with the highest rate of efficacy while safe, is a more appropriate endpoint. A study by Corbaux et al. [29] showed that the dose approved by the FDA is consistent with the OBD for 83% of the drugs in a total of 87 completed trials for evaluating molecular targeted agents.

The design for finding the OBD can also base on the posterior distributions of the toxicity and efficacy rates jointly to guide the dose-transition. Similar to the above Keyboard paradigm, a highest joint unit probability mass (JUPM) which is defined by the joint probabilities falling into the pre-specified equivalence target intervals for the toxicity and efficacy targets (the formal definition will be introduced shortly) and is also referred to as the winner

key, is used as the cornerstone for making dose-assignment decisions for the OBD-finding studies. The `Keyboard` package implements a method proposed by Li et al. in [15].

To use this design, a boundary table for both the toxicity and efficacy needs to be pre-specified by the investigators prior to the study. To be specific, for the toxicity, taking a four-subinterval schema as an example, it can be categorized as: low, moderate, high, and unacceptable. Generating these four intervals requires the following three input parameters, `toxicity.low, toxicity.moderate, toxicity.high`, to form the sub-intervals of (0, toxicity.low), (toxicity.low, toxicity.moderate), (toxicity.moderate, toxicity.high), and (toxicity.high, 1). These parameters are specified prior to a study by investigators and an example will be shown shortly.

Similarly, a four-subinterval for the efficacy can be generated if three parameters, `efficacy.low, efficacy.moderate, efficacy.high`, are given.

Assume there are d doses in the trial and the current dose is i, number of patients at this dose level is n_i, the number of patients who experienced toxicity is x_i, and the number of responses is y_i. The trial data can be represented as follows:

$$D = (n_i, x_i, y_i), i = 1, ..., d$$

Assuming that the toxicity probability is p_i and the efficacy probability is q_i at dose level i, the probability unit intervals for toxicity and efficacy can be partitioned into subintervals (a, b) and (c, d). Here, (a, b) is the subinterval for the toxicity probability, and (c, d) is the subinterval for the efficacy probability. The (a, b) $\times$ (c, d) represents a combination interval for joint toxicity and efficacy probability.

For instance, given the above four-subinterval for toxicity and efficacy probability, there are 16 combination intervals in total. Investigators would then be required to provide associated 16 decisions corresponding to the combination intervals based on the clinical rationale, which is quite similar to pre-specify utility scores strategy adopting by various authors [19, 20]. Decision "D, E, and S" denotes de-escalation, escalation, and stay. An exemplary pre-specified boundary table with a target toxicity rate of 0.2 and a target efficacy rate of 0.4 is given in

Table 2.

Table 2: An example of pre-specified decision table for the OBD-finding design

		Efficacy.low (0,0.25)	Efficacy.moderate (0.25,0.45)	Efficacy.high (0.45,0.65)	Efficacy.superb (0.65,1)
Toxicity.low	(0,0.15)	E	E	E	E
Toxicity.moderate	(0.15,0.25)	E	E	E	S
Toxicity.high	(0.25,0.35)	D	S	S	S
Toxicity.unacceptable	(0.35,1.0)	D	D	D	D

From the above Table 2, for example, the clinical team may think that the combination interval (0.25,0.35) $\times$ (0,0.25), which corresponds to high toxicity and low efficacy, should invoke a decision "D", e.g., De-escalation, that is, the next cohort of patients will be treated at the next lower level if the estimated toxicity rate of the current dose falls in (0.25,0.35) and meanwhile the estimated efficacy rate falls in (0,0.25).

For making the decision, we need to calculate the joint unit probability mass (JUPM) and the JUPM is defined as follows:

$$JUPM_{(a,b)}^{(c,d)} = \frac{Pr(p_j \in (a,b), q_j \in (c,d)|D)}{(b-a) \times (d-c)}$$

Here, $Pr(p_j \in (a,b), q_j \in (c,d)|D)$ is the posterior probability of p_i and q_i falling in the subinterval (a,b) and (c,d). Assume the priors for both p_i and q_i follow independent beta distributions $Beta(\alpha_p, \beta_p)$ and $Beta(\alpha_q, \beta_q)$ independently. The posterior distributions for p_i and q_i are $Beta(\alpha_p + x_i, \beta_p + n_i - x_i)$ and $Beta(\alpha_q + y_i, \beta_q + n_i - y_i)$. Using these posterior distributions to update the JUPMs for all sixteen combination intervals, we can find the winning combination interval (a*, b*) and (c*, d*), which has the largest JUPM value, which can guide the dose-transition to treat the next cohort of patients.

Two dose exclusion rules in terms of safety and futility are recommended in real practices similar to designs introduced in previous sections:

Safety rule: Similar to the previous methods, if at least 3 patients have been treated at a given dose and the observed data indicate that the probability of the current dose's toxicity rate exceeding the target toxicity rate by more than 95%, then we eliminate the current dose

and any higher doses from the trial to avoid exposing future patients to unacceptably toxic doses. If the lowest dose is unacceptably toxic, then the trial terminates early and no dose is selected as the OBD. This corresponds to a dose assignment of "`DUT`" output by the software, which means to de-escalate because of unacceptable high toxicity and exclude the current dose and any dose higher than this dose from the trial.

The probability threshold can be specified with `cutoff.eli.toxicity` in the software shown in the next section.

Futility rule: if at least 3 patients have been treated at a given dose and the observed data indicate that the probability of the current dose's efficacy rate exceeding the target efficacy rate by less than 30%, then we eliminate this dose from the trial to avoid exposing future patients to these futile doses. The probability threshold can be specified with `cutoff.eli.efficacy`. The software will output two possible dose assignments: "EUE" and "DUE", both of which exclude the current dose from the trial. "EUE" denotes escalation because of unacceptable low efficacy; "DUE" denotes de-escalation because of unacceptable low efficacy.

An attractive feature of this design is that its dose escalation and de-escalation rule can also be tabulated prior to the study. Thus, when implementing a trial, no real-time calculation or complicated model fitting is needed, and we need only to count the number of patients, the number of DLTs, and the number of responses observed at the current dose, and the decision to escalate or des-escalate the dose is based on the pre-tabulated decision rules shown in Table 2.

For the final OBD selection after completing a trial, a utility score is evaluated to quantify the desirability of all admissible doses. Calculation of utility scores requires the posterior probabilities for toxicity p_i and efficacy q_i, which can be computed using $Beta(\alpha_p + x_i, \beta_p + n_i - x_i)$ and $Beta(\alpha_q + y_i, \beta_q + n_i - y_i)$, assuming that the priors for both p_i and q_i follow independent beta distributions $Beta(\alpha_p, \beta_p)$ and $Beta(\alpha_q, \beta_q)$. In this package, we provide the three options for the utility functions.

The first utility function is a function of the toxicity $f_1(p)$, where p denotes the toxicity

rate, and the efficacy $f_2(q)$, where q denotes the efficacy rate. $f_1(p)$ is shown as in equation (1).

$$f_1(p) = \begin{cases} 1, & p \in [0, p_1) \\ 1 - (p - p_1)/(p_2 - p_1) & p \in [p_1, p_2) \\ 0 & p \in [p_2, 1] \end{cases} \tag{1}$$

Here, p_1 is the cutoff lower limit and p_2 is the cutoff upper limit for safety utility function is $f_1(p)$, which was used in [15].

Similarly, $f_2(q)$ is shown in Equation (2).

$$f_2(p) = \begin{cases} 0, & q \in [0, q_1) \\ 1 - (p - q_1)/(q_2 - q_1) & q \in [q_1, q_2) \\ 1 & q \in [q_2, 1] \end{cases} \tag{2}$$

Here, q_1 is the cutoff lower limit and q_2 is the cutoff upper limit for safety utility function is $f_2(q)$.

Assuming the toxicity and efficacy are independent of each other, the first utility function that quantifies benefit-risk trade-off at current dose i is defined as follows:

$$U(p_i, q_i) = f_1(p_i) \times f_2(q_i) \tag{3}$$

The second utility function depends on a marginal toxicity probability $\pi_{T,i} = Pr(Y_T = 1|d = i)$ and a marginal efficacy probability $\pi_{E,i} = Pr(Y_E = 1|d = i)$, which is defined as follows:

$$U(p_i, q_i) = \pi_{E,i} - w_1 \times \pi_{T,i} \tag{4}$$

where, w_1 is a pre-specified weight. This trade-off function describes how much patients are willing to trade an increase of w_1 in the DLT rate for a unit increase in the efficacy rate. If w_1 =0, we obtain a special case that the dose with the highest efficacy is the most desirable.

The third utility function also depends on the marginal toxicity probability $\pi_{T,i} = Pr(Y_T = 1|d = i)$ and efficacy probability $\pi_{E,i} = Pr(Y_E = 1|d = i)$, but it puts an additional penalty for over-toxicity and is defined as follows:

$$U(p_i, q_i) = \pi_{E,i} - w_1 \times \pi_{T,i} - w_2 * \pi_{T,i} * I(\pi_{T,i} > \rho) \tag{5}$$

where w_1 and w_2 are pre-specified weights, $I(\cdot)$ is an indicator function, and ρ is a pre-specified toxicity threshold deemed of substantial concern and can be choosen as the target toxicity rate. Compared to the above second utility function (4), this trade-off function is more flexible and allows to impose a higher penalty (i.e., $w_1 + w_2$) when the true DLT rate $\pi_{T,j}$ exceeds the threshold ρ.

Once the utility score is computed for all the doses, then the optimal biological dose (OBD) can be estimated by:

$$d = argmax_{i \in d} \ [U(p_i, q_i)|D]$$

3. Results

The R package `Keyboard` contains functions that implement the designs introduced in the above section for the single-agent, drug-combination MTD-finding trials and the phase I/II OBD-finding trials.

Single-agent MTD-finding design

- `get.boundary.kb(···)`: This function is used to generate the optimal dose-escalation and de-escalation boundaries for conducting a single-agent trial with the Keyboard design.

- `select.mtd.kb(···)`: This function is used to select the MTD after the single-agent trial is completed.

- `get.oc.kb(···)`: This function is used to generate the operating characteristics of the Keyboard design for single-agent trials.

Drug-combination MTD-finding design

- `get.boundary.comb.kb(···)`: This function is used to generate the optimal dose escalation and de-escalation boundaries for conducting a drug-combination trial with the Keyboard design.
- `next.comb.kb(···)`: This function is used to determine the dose combination for the next cohort of patients in drug-combination trials that aim to find one MTD.
- `select.mtd.comb.kb(···)`: This function is used to select the MTD after the drug-combination trial is completed.
- `get.oc.comb.kb(···)`: This function is used to generate the operating characteristics of the Keyboard design for drug-combination trials.

Phase I/II trial OBD-finding design

- `get.decision.obd.kb(···)`: This function is used to generate the boundary table and decision matrix for single-agent phase I/II trials designed to find the OBD.
- `select.obd.kb(···)`: This function is used to select the OBD at the end of a trial.
- `get.oc.obd.kb(···)`: This function is used to generate operating characteristics.

Phase I/II trial-automatically generating decision tables

Different from the above, the following functions do not rely on pre-specified decision tables (Details and example are introduced shortly).

- `get.decision.obd2.kb(···)`: This function is used to provid the boundary table and decision matrix for the phase I/II trials automatically.
- `select.obd.kb(···)`: This function is used to select the optimal biological dose (OBD) at the end of a single-agent phase I/II trial.

- `get.oc.obd2.kb(···)`: This function is used to generate the operating characteristics, and the decision tables is generated automatically.

3.1. Single-agent trials

Design and conduct the trial

To design a single-agent trial, we can run the function `get.boundary.kb(·)` to obtain the dose escalation and de-escalation boundaries. This function has the following arguments:

- `target` The target dose-limiting toxicity (DLT) rate.
- `ncohort` The total number of cohorts.
- `cohortsize` The number of patients in the cohort.
- `marginL` The difference between the target and the lower bound of the "target key" (proper dosing interval) to be defined. The default is 0.05.
- `marginR` The difference between the target and the upper bound of the "target key" (proper dosing interval) to be defined. The default is 0.05.
- `n.earlystop` The early stopping parameter. If the number of patients treated at the current dose reaches `n.earlystop`, then stop the trial and select the MTD based on the observed data. The default value is 100.
- `cutoff.eli` The cutoff value to eliminate an overly toxic dose and all higher doses for safety. The recommended value is 0.95.
- `extrasafe` Set `extrasafe = TRUE` to impose a stricter stopping rule for extra safety, expressed as the stopping boundary value in the result.

As introduced previously, there are two built-in stopping rules:
(1) Stop the trial if the lowest dose is eliminated due to unacceptably high toxicity. In this case, no dose should be selected as the MTD/OBD;

(2) Stop the trial and select the MTD if the number of patients treated at the current dose reaches `n.earlystop`.

The first stopping rule is a safety rule to protect patients from the case in which all doses are overly toxic. The rationale for the second stopping rule is that when the number of patients assigned to a dose is large (i.e., `n.earlystop`), the dose-finding algorithm has approximately converged, or due to limited sample size of rare diseases, we can sometimes stop the trial if a certain number of patients have been treated at a dose level so that we can stop the trial early and select the MTD to save sample size and reduce the trial's duration. The trade-off is that it may affect the MTD selection percentage and decrease the rate of stopping for safety if the first dose is overly toxic. The value of `n.earlystop` should be calibrated by simulation to obtain desirable operating characteristics. In general, we recommend `n.earlystop = 9` or 12 often has desirable operating characteristics. Our experience is that this stopping rule is particularly useful when there is strong prior knowledge that the first dose is safe because a major side effect of using the stopping rule is that it decreases the rate of stopping for safety when the first dose is actually overly toxic.

Although the Keyboard design has a built-in safety stopping rule (i.e., stopping rule (1) described above), for some applications, investigators may prefer a stricter stopping rule for extra safety when the lowest dose is possibly overly toxic. Setting `extrasafe = TRUE` imposes the following stronger stopping rule:

> Stop the trial if (1) the number of patients treated at the lowest dose ≥ 3, and (2) Pr(toxicity rate of the lowest dose $>$ `target` | data) $>$ `cutoff.eli - offset`.

Note that as a trade-off, the stricter stopping rule will decrease the MTD selection percentage when the lowest dose actually is the true MTD. When using the option `extrasafe = TRUE`, we recommend the default value `offset = 0.05`, but users can calibrate the value of `offset` to obtain desired operating characteristics. In practice, `offset` is rarely greater than 0.2.

As an example, suppose we want to conduct a phase I trial with $J = 5$ dose levels and a target toxicity rate of $\phi = 0.3$. The maximum sample size is 30 patients, and patients are treated in cohorts of size 3. Using the default values of `marginL`, `marginR`, and `cutoff.eli` au-

tomatically provided by the function, we can design the trial by running `get.boundary.kb(·)`:

```
R> get.boundary.kb(target=0.3, ncohort=10, cohortsize=3)

The following are decision boundaries:

Number of patients treated    3 6 9 12 15 18 21 24 27 30
Escalate if # of DLT <=       0 1 2  2  3  4  5  5  6  7
De-escalate if # of DLT >=    2 3 4  5  6  7  8  9 10 11
Eliminate if # of DLT >=      3 4 5  7  8  9 10 11 12 14

A more complete version of the decision boundaries is given as follows:

Number of patients treated  1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
18 19 20 21 22 23 24 25 26 27 28 29 30
Escalate if # of DLT <=     0 0 0 0 1 1 1 1 2  2  2  2  3  3  3  3  4
4  4  4  5  5  5  5  6  6  6  6  7  7
De-escalate if # of DLT >=  1 1 2 2 2 3 3 3 4  4  4  5  5  5  6  6  6
7  7  7  8  8  9  9  9 10 10 10 11 11
Eliminate if # of DLT >=    NA 2 3 3 4 4 5 5 5  6  6  7  7  8  8  8  9
9  9 10 10 11 11 11 12 12 12 13 13 14
```

Remarks:

- The output presents the dose escalation/de-escalation rules in two forms: based on the observed toxicity rate, and based on the observed DLT. We recommend the latter because clinical researchers often find it easier to use.
- For convenience, two versions of the decision boundaries are displayed: one is based on the cohort size (3, 6, 9, ..., 30), the other is for all possible sample sizes (1, 2, 3, ..., 30). Although the design assumes a constant cohort size of 3, in practice, the actual cohort

size may vary during the trial for different reasons. If that is the case, then it is more appropriate to present the complete version of the decision boundaries in the protocol such that investigators can decide dose assignment at any time during the trial for any given number of patients who have been treated at the current dose level. This is one of the important advantages of the Keyboard design: it allows the cohort size to vary from one cohort to another and for the decision of dose escalation/de-escalation to be made at any time during the trial's conduction.

- The elimination boundaries are used to avoid treating patients at overly toxic doses, based on the following Bayesian safety rule: if $\text{pr}(p_j > \phi | m_j, n_j) > 0.95$ and $n_j \geq 3$, dose levels j and higher are eliminated from the trial; and the trial is terminated if the first dose level is eliminated. Here, m_j is the number of patients who expericed toxicity and n_j is the number of the patients treated at the current dose level.

If we set `extrasafe = TRUE` to turn on the `extrasafe` feature, the output will include the extra stopping boundaries, as follows,

```
R> get.boundary.kb(target=0.3, ncohort=10, cohortsize=3, extrasafe=T)

The stopping boundaries are shown as follows:

Number of patients treated at the lowest dose    1  2 3 4 5 6 7 8 9 10 11 12
13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
Stop the trial if # of DLT >=                    NA NA 2 3 3 4 4 4 5 5  6  6
6  7  7  8  8  8  9  9  9 10 10 10 11 11 12 12 12 13
```

Obtain the operating characteristics

For the protocol preparation, it is often useful to obtain the operating characteristics of the design. The function `get.oc.kb(·)` can be used for this purpose. This function shares the

same set of arguments as the function `get.boundary.kb`(·) described previously, with three additional arguments:

- `p.true` A vector containing the true toxicity probabilities of the investigational dose levels.
- `startdose` The starting dose level for treating the first cohort of patients. The default value is `startdose = 1`, i.e., starting from the lowest dose.
- `ntrial` The total number of trials to be simulated. The default value is 1000.

Using the same setting as above and assuming that the true toxicity scenario is `p.true` = $(0.05, 0.15, 0.30, 0.45, 0.6)$, here we show how to obtain the operating characteristics based on 1000 simulated trials.

```
R> get.oc.kb(target=0.3, p.true=c(0.05, 0.15, 0.3, 0.45, 0.6),
+           ncohort=20, cohortsize=3, ntrial=1000)
```

```
selection percentage at each dose level (%):     1.1 23.2 64.2 11.3  0.1
number of patients treated at each dose level:    4.6 17.6 27.5  9.2  1.1
number of toxicities observed at each dose level: 0.2 2.6 8.2 4.2 0.7
average number of toxicities: 15.9
average number of patients: 59.9
percentage of early stopping because of toxicity: 0.1%
risk of overdosing 60% or more of patients:  5.1%
risk of overdosing 80% or more of patients : 0.5%
```

Select the MTD when the trial is completed

When the trial is completed, we can select the MTD based on the observed data using

the function `select.mtd.kb(···)`. This function has six arguments: `target`, `npts`, `ntox`, `cutoff.eli`, `extrasafe` and `offset`, where

- `npts` A vector containing the number of patients treated at each dose level.
- `ntox` A vector containing the number of patients at each dose level who experienced a DLT at each dose level.

Arguments `cutoff.eli`, `extrasafe` and `offset` are the same as (and should be consistent with) those in functions `get.boundary.kb(·)` and `get.oc.kb(·)`, with default values `cutoff.eli = 0.95`, `extrasafe = TRUE` and `offset = 0.05`. When the default values are used, there is no need to specify the arguments in `select.mtd.kb(·)`. Assume that the number of patients treated at five doses is $n = (3, 3, 15, 9, 0)$ and the corresponding number of patients who experienced toxicity is $y = (0, 0, 4, 4, 0)$.

```
R> n<-c(3, 3, 15, 9, 0)
R> y<-c(0, 0, 4, 4, 0)
R> select.mtd(target=0.3, ntox=y, npts=n)

The selected MTD is dose level 3.
```

Dose Level	Posterior DLT Estimate	95% Credible Interval	Pr(toxicity>0.3|data)
1	0.02	(0.00 , 0.20)	0.01
2	0.02	(0.00 , 0.20)	0.01
3	0.27	(0.09 , 0.51)	0.36
4	0.45	(0.16 , 0.75)	0.66
5	----	(----------)	----

The result is that dose level 3 is selected as the MTD. No estimates are provided for dose level 5 because dose level 5 is not used in this trial.

We note here that the above functionals can also be implemented by Keyboard design module via Shiny app at www.trialdesign.org.

3.2. Drug-combination trials aiming to find a single MTD

Design and conduct the trial

To design a drug-combination trial, we can run the function `get.boundary.comb.kb(·)` to obtain the dose escalation and de-escalation boundaries, which are used to run the trial. This function has the following arguments:

- `target` The target DLT rate.
- `ncohort` The total number of cohorts.
- `cohortsize` The number of patients in the cohort.
- `marginL` The difference between the target and the lower boundary of the "target key" (proper dosing interval) to be defined. The default is 0.05.
- `marginR` The difference between the target and the upper boundary of the "target key" (proper dosing interval) to be defined. The default is 0.05.
- `n.earlystop` The early stopping parameter. If the number of patients treated at the current dose reaches `n.earlystop`, then stop the trial and select the MTD based on the observed data. The default value is 100.
- `cutoff.eli` The cutoff value to eliminate an overly toxic dose and all higher doses for safety. The recommended value is 0.95.
- `extrasafe` Set `extrasafe = TRUE` to impose a stricter stopping rule for extra safety, expressed as the stopping boundary value in the result.

As an example, suppose we want to conduct a phase I trial with a target toxicity rate of $\phi = 0.3$. The maximum sample size is 30 patients, and patients are treated in cohorts of size 3. Using the default values of `marginL`, `marginR`, and `cutoff.eli` automatically provided by the function, we can design the trial by running `get.boundary.kb(·)`:

```
R>  get.boundary.comb.kb(target=0.3, ncohort=10, cohortsize=3)

The following are decision boundaries:


Number of patients treated  1  2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
18 19 20 21 22 23 24 25 26 27 28 29 30
Escalate if # of DLT <=     0 0 0 0 1 1 1 1 2  2  2  2  3  3  3  3  4
4  4  4  5  5  5  5  6  6  6  6  7  7
de-escalate if # of DLT >=  1 1 2 2 2 3 3 3 4  4  4  5  5  5  6  6  6
7  7  7  8  8  9  9  9 10 10 10 11 11
Eliminate if # of DLT >=   NA NA 3 3 4 4 5 5 5  6  6  7  7  8  8  8  9
9  9 10 10 11 11 11 12 12 12 13 13 14
```

Obtain the operating characteristics

The function `get.oc.comb.kb(·)` can be used to obtain the operating characteristics of the Keyboard drug-combination design. This function shares the same set of arguments as the function `get.oc.kb`(·), except that `p.true` is now a matrix (rather than a vector) and `startdose` is a vector of length 2 (rather than a scalar).

Consider a 3×4 combination trial with the true toxicity probabilities

$$\texttt{p.true} = \begin{pmatrix} 0.02 & 0.04 & 0.08 & 0.14 \\ 0.08 & 0.25 & 0.42 & 0.48 \\ 0.25 & 0.45 & 0.50 & 0.60 \end{pmatrix},$$

and the target toxicity rate of 0.25. Here, we show how to obtain the operating characteristics based on 1000 simulated trials.

```
R> p.true <- matrix(c(0.01, 0.03, 0.10, 0.20, 0.30, 0.03, 0.05, 0.15, 0.30, 0.60,
+         0.08, 0.10, 0.30, 0.60, 0.75), byrow=TRUE, ncol=5)
R> get.oc.comb.kb(target=0.3, p.true, ncohort=20, cohortsize=3, n.earlystop=12,
+       startdose=c(1, 1), ntrial=100)

true toxicity rate of dose combinations:
0.01 0.03 0.10  0.20 0.30
0.03 0.05 0.15  0.30 0.60
0.08 0.10 0.30  0.60 0.75

selection percentage at each dose combination (%):
0    0    2    3    5
0    0   10   19    1
0   13   43    4    0

number of patients treated at each dose combination:
3.06 1.65 0.96 0.99 0.87
1.62 1.83 3.06 2.76 1.20
0.99 3.21 6.96 2.94 0.15

number of toxicities observed at each dose combination:
0.01 0.06 0.09 0.18 0.23
0.03 0.05 0.41 0.71 0.76
0.07 0.35 2.18 1.81 0.13

average number of toxicities: 7.1
average number of patients: 32.2
```

```
selection percentage of MTD: 67%
percentage of patients treated at the MTD: 32.8%
percentage of early stopping because of toxicity: 0.00%
```

Conduct the trial

The function `next.comb.kb(·)` is used to conduct phase I drug-combination trials that aim to find a single MTD. It takes the data from patients who have been enrolled into the trial as the input, and outputs the dose combination for treating the next cohort of new patients. The function `next.comb.kb(·)` shares a similar set of arguments with the function `get.boundary.kb`(·) described previously, with three additional arguments:

- `npts` A matrix recording the number of patients treated at each dose combination.
- `ntox` A matrix recording the number of patients who experienced toxicity at each dose combination.
- `dose.curr` The current dose combination (the dose combination that was used to treat the most recently enrolled cohort of patients).

Suppose that we conduct a 3×4 drug-combination trial with 3 dose levels of agent A and 4 dose levels of agent B, aiming to find a MTD that has a target toxicity rate of 0.3. The maximum sample size is 48 patients, and patients are treated in cohort sizes of 3. Let (j, k) denote the combination of the jth dose level of agent A and the kth dose level of agent B. The trial can be conducted as follows. We start the trial by treating the first cohort of 3 patients at the lowest dose $(2, 2)$. Assuming that none of the patients experienced DLT, the data from the first cohort of patients are given by

$$n = \begin{pmatrix} 3 & 0 & 0 & 0 & 0 \\ 7 & 6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \qquad y = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

where n records the number of patients treated at each dose combination, and y records the

number of patients who experienced toxicity at each dose combination. In matrixes y and n, entry (j, k) records the data associated with combination (j, k). To determine the dose for the second cohort of patients, we call function `next.comb.kb(·)`:

```
R> n<-matrix(c(3, 0, 0, 0, 0,7, 6, 0, 0, 0, 0, 0, 0, 0, 0), ncol=5, byrow=TRUE)
R> y<-matrix(c(0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0), ncol=5, byrow=TRUE)
R> next.comb.kb(target=0.3, npts=n, ntox=y, dose.curr=c(2, 2))
```

```
The recommended dose combination for the next cohort of patients is (2, 3)
```

Therefore, we escalate the dose and treat the second cohort of patients at dose combination (2, 3). Suppose that one patient in the second cohort experienced DLT; the data matrixes then become

$$n = \begin{pmatrix} 3 & 0 & 0 & 0 & 0 \\ 7 & 6 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \qquad y = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

To determine the dose for the third cohort of patients, we again call `next.comb.kb(·)` with updated `y`, `n` and `dose.curr`, as follows:

```
R> n<-matrix(c(3, 0, 0, 0, 0,7, 6, 3, 0, 0, 0, 0, 0, 0, 0), ncol=5, byrow=TRUE)
R> y<-matrix(c(0, 0, 0, 0, 0, 1, 1,3, 0, 0, 0, 0, 0, 0, 0), ncol=5, byrow=TRUE)
R> next.comb.kb(target=0.3, npts=n, ntox=y, dose.curr=c(2, 2))
```

```
The recommended dose combination for the next cohort of patients is (2, 2)
```

Therefore, we should de-escalate the dose and treat the third cohort of patients at dose (2, 2). We repeat this procedure until the maximum sample size is reached.

Select a MTD when the trial is completed

When the trial is completed, based on the observed data, we can select a MTD using the function `select.mtd.comb.kb(·)`. This function has seven arguments: `target`, `npts`, `ntox`, `cutoff.eli`, `extrasafe`, and `offset`, where the descriptions of `cutoff.eli`, `extrasafe`, and `offset` are the same as those in `get.boundary.kb(·)`. Assume that the number of patients treated at each dose combination and the corresponding number of patients who experienced toxicity at each dose combination are

$$n = \begin{pmatrix} 6 & 3 & 0 & 0 \\ 6 & 24 & 9 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad y = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 5 & 4 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

```
R> n<-matrix(c(6, 3, 0, 0,  6, 24, 9, 0,  0, 0, 0, 0), ncol=4, byrow=TRUE)
R> y<-matrix(c(0, 0, 0, 0,  1, 5, 4, 0,  0, 0, 0, 0), ncol=4, byrow=TRUE)
R> select.mtd.comb.kb(target=0.25, npts=n, ntox=y)

The MTD is dose combination ( 2 ,  2 )
Isotonic estimates of the toxicity rates are
0.01  0.02   NA   NA
0.17  0.21  0.45  NA
NA    NA     NA   NA
```

The result is that dose combination (2, 2) is selected as the MTD. This combination design can also be implemented at www.trialdesign.org.

3.3. Phase I/II trials aiming to find the OBD

Design and conduct the trial

To design a phase I/II trial for finding the OBD, we need to run the function `get.decision.obd.kb(·)` to obtain the boundary table and decision table. This function has the following arguments to input:

- `toxicity.low` The upper boundary for the low toxicity interval.
- `toxicity.moderate` The upper boundary for the moderate toxicity interval.
- `toxicity.high` The upper boundary for the high toxicity interval.
- `efficacy.low` The upper boundary for the low efficacy interval.
- `efficacy.moderate` The upper boundary for the moderate efficacy interval.
- `efficacy.high` The upper boundary for the high efficacy interval.
- `target.toxicity` The target DLT rate.
- `target.efficacy` The target efficacy rate.
- `ncohort` The total number of cohorts.
- `cohortsize` The number of patients in the cohort.
- `cutoff.eli.toxicity` The cutoff value to eliminate a dose with an unacceptably high toxicity for safety. The default value is 0.95.
- `cutoff.eli.efficacy` The cutoff value to eliminate a dose with unacceptably low efficacy. The default value is 0.3.

As an example, suppose we want to conduct a phase I/II trial with $J = 5$ dose levels, a target toxicity rate of $target.toxicity = 0.20$, a target efficacy rate of $target.efficacy = 0.40$. Patients are treated in cohorts of size 3 and ncohort is 10. Using the default values of `cutoff.eli.toxicity`, and `cutoff.eli.efficacy` provided by the function, we can design a trial by running `get.decision.obd.kb(·)`, as follows:

```
R> toxicity.low <- 0.15
R> toxicity.moderate <- 0.25
R> toxicity.high <- 0.35
R> efficacy.low <- 0.25
R> efficacy.moderate <- 0.45
```

```
R> efficacy.high <- 0.65
R> target.toxicity=0.20
R> target.efficacy=0.40
R> cohortsize=3
R> ncohort=10
R> decision.obd <- get.decision.obd.kb( toxicity.low = toxicity.low,
+               toxicity.moderate= toxicity.moderate,
+               toxicity.high = toxicity.high,
+               efficacy.low = efficacy.low,
+               efficacy.moderate = efficacy.moderate,
+               efficacy.high = efficacy.high,
+               target.toxicity=target.toxicity,
+               target.efficacy=target.efficacy,
+               cohortsize=cohortsize, ncohort=ncohort)
```

T1	T2	EF1	EF2	DECISION
0.00	0.15	0.00	0.25	E
0.00	0.15	0.25	0.45	E
0.00	0.15	0.45	0.65	E
0.00	0.15	0.65	1.00	E
0.15	0.25	0.00	0.25	E
0.15	0.25	0.25	0.45	E
0.15	0.25	0.45	0.65	E
0.15	0.25	0.65	1.00	S
0.25	0.35	0.00	0.25	D
0.25	0.35	0.25	0.45	S
0.25	0.35	0.45	0.65	S
0.25	0.35	0.65	1.00	S
0.35	1.00	0.00	0.25	D

0.35	1.00	0.25	0.45	D
0.35	1.00	0.45	0.65	D
0.35	1.00	0.65	1.00	D

Partial output of the decision matrix is given as the follows:

N	T	R	Decision
3	0	0	EUE
3	0	1	E
3	0	2	E
3	0	3	E
3	1	0	DUE
3	1	1	S
3	1	2	S
3	1	3	S
3	2	0	DUE
3	2	1	D
3	2	2	D
3	2	3	D
3	3	0	DUT
3	3	1	DUT
3	3	2	DUT
3	3	3	DUT
6	0	0	EUE
6	0	1	EUE
6	0	2	E
6	0	3	E
6	0	4	E
6	0	5	E
6	0	6	E

```
6   1   0       EUE
6   1   1       EUE
6   1   2         E
6   1   3         E
6   1   4         E
6   1   5         S
6   1   6         S
6   2   0       DUE
6   2   1       DUE
6   2   2         S
6   2   3         S
6   2   4         S
6   2   5         S
6   2   6         S
6   3   0       DUE
6   3   1       DUE
6   3   2         S
6   3   3         S
6   3   4         S
6   3   5         S
6   3   6         S
6   4   0       DUE
6   4   1       DUE
6   4   2         D
6   4   3         D
6   4   4         D
6   4   5         D
6   4   6         D
6   5   0       DUT
```

```
6   5   1       DUT
6   5   2       DUT
6   5   3       DUT
6   5   4       DUT
6   5   5       DUT
6   5   6       DUT
6   6   0       DUT
6   6   1       DUT
6   6   2       DUT
6   6   3       DUT
6   6   4       DUT
6   6   5       DUT
6   6   6       DUT
```

- D: deescalate to the previous dose level j-1, assuming that the current dose level is j and j is not the lowest dose level. Otherwise, treat the next cohort of patients at the current dose level j.
- E: escalate to the next dose level j+1 assuming that the curent dose level is j and j is not the highest dose level. Otherwise, treat the next cohort of patients at the current dose level j.
- S: stay at the current dose level j (treat the next cohort of patients at dose level j).
- EUE: escalate to the next higher dose level because of unacceptably low efficacy.
- DUE: deescalate to the previous lower admissible dose level because of unacceptably low efficacy.
- DUT: deescalate to the previous lower admissible dose level because of unacceptably high toxicity.

- Default stopping rule: if $Pr(p_i > p_T|D)$ is too high (the default is 0.95), then any dose higher than i or equal to i is excluded from the trial.

- Default futility rule: if $Pr(q_i > q_E|D)$ is too low(the default value is 0.3), then dose i is excluded from the trial.

From the above output, we can firstly have a pre-specified decision table like the one shown by Table 2 in the previous section and then get the decision rules for all combinations of toxicity and response. For example, we can see that if there are 3 patients on a specific dose level with 1 toxicity and 1 response, the decision would be "S", that is, we will continue to administer the current dose to the next cohort of patients.

Obtain the operating characteristics

The function `get.oc.obd.kb(·)` can be used to obtain the operating characteristics of the design. This function takes these following arguments:

- `toxicity.low` The upper boundary for the low toxicity interval.
- `toxicity.moderate` The upper boundary for the moderate toxicity interval.
- `toxicity.high` The upper boundary for the high toxicity interval.
- `efficacy.low` The upper boundary for the low efficacy interval.
- `efficacy.moderate` The upper boundary for the moderate efficacy interval.
- `efficacy.high` The upper boundary for the high efficacy interval.
- `target.toxicity` The target DLT rate.
- `target.efficacy` The target efficacy rate.
- `ncohort` The total number of cohorts.
- `cohortsize` The number of patients in the cohort.

- `n.early` The early stopping parameter. If the number of patients treated at the current dose reaches $n.early$, then we stop the trial and select the MTD based on the observed data. The default value is 100.
- `startdose` The starting dose level.
- `p.true` A vector containing the true toxicity probabilities of the investigational dose levels.
- `q.true` A vector containing the true efficacy probabilities of the investigational dose levels.
- `ntrial` The total number of trials to be simulated.
- `seed` The random seed for simulation.
- `p1` The cutoff lower limit for safety utility funciton (3): $U = f(p) \times f(q)$. f(p) is the function for toxicity probability, and f(q) is the function for efficacy probability.
- `p2` The cutoff upper limit for safety utility function (3).
- `q1` The cutoff lower limit for efficacy utility function (3).
- `q2` The cutoff upper limit for efficacy utility function (3).
- `cutoff.eli.toxicity` The cutoff value to eliminate a dose with unacceptably high toxicity for safety. The default value is 0.95.
- `cutoff.eli.efficacy` The cutoff value for the futility rule, the acceptably lowe efficacy. The default value is 0.30.
- `w1.toxicity` The weight for toxicity utility function (4) and (5) The recommended ρ is the target toxicity rate.
- `w2.toxicity` The weight for toxicity utility funciton (5).
- `indicator` The indicator cutoff value for utility funciton (5).

Using the same setting as above and assuming that the true toxicity scenario is `p.true` $= (0.16, 0.20, 0.25, 0.30)$, the true efficacy scenario is `q.true` $= (0.05, 0.10, 0.15, 0.18)$; here we show how to obtain the operating characteristics based on 1000 simulated trials.

```
R> toxicity.low <- 0.15
R> toxicity.moderate <- 0.25
R> toxicity.high <- 0.35
R> efficacy.low <- 0.25
R> efficacy.moderate <- 0.45
R> efficacy.high <- 0.65
R> target.toxicity=0.20
R> target.efficacy=0.40
R> p.true <-c(0.08,0.20,0.60,0.80)
R> q.true <- c(0.25,0.40,0.25,0.50)
R> oc.obd.kb <- get.oc.obd.kb(toxicity.low = toxicity.low,
+            toxicity.moderate= toxicity.moderate,
+            toxicity.high = toxicity.high,
+            efficacy.low = efficacy.low,
+            efficacy.moderate = efficacy.moderate,
+            efficacy.high = efficacy.high,
+            target.toxicity=target.toxicity,
+            target.efficacy= target.efficacy,
+            p.true= p.true, q.true= q.true)

selection percentage (%) at each dose level using the 1st utility function:
27.7 44.1  0.9  0.0
selection percentage (%) at each dose level using the 2nd utility function:
27.2 46.0  1.7  0.1
selection percentage (%) at each dose level using the 3rd utility function:
```

```
29.7 44.3  1.0  0.0
number of patients treated at each dose level:
6.822 13.770  8.694  0.642
number of toxicities observed at each dose level:
0.557 4.168 5.242 0.499
number of responses observed at each dose level:
1.691 5.469 2.192 0.314
average number of toxicities: 10.466
average number of responses:  9.666
average number of patients: 29.928
the percentage of early stop (%): 0.3
the percentage of trial termination using the 1st utility function (%): 27.3
the percentage of trial termination using the 2nd utility function (%): 25
the percentage of trial termination using the 3rd utility function (%): 25
```

Select the OBD when the trial is completed

When the trial is completed, we can select the OBD based on the observed data using the function `select.obd.kb(···)`. This function has these arguments:

- `target.toxicity` The target DLT rate.
- `target.efficacy` The target efficacy rate.
- `npts` The vector containing the total number of patients treated at each dose level.
- `ntox` The vector containing the number of subjects at each dose level who experienced toxicities.
- `neff` The vector containing the number of subjects at each dose level who experienced efficacies.
- `p1` The cutoff lower limit for safety utility funciton (3): $U = f(p) \times f(q)$. f(p) is the function for toxicity probability, and f(q) is the function for efficacy probability.

- `p2` The cutoff upper limit for safety utility function (3).
- `q1` The cutoff lower limit for efficacy utility function (3).
- `q2` The cutoff upper limit for efficacy utility function (3).
- `cutoff.eli.toxicity` The cutoff value to eliminate a dose with unacceptable high toxicity for safety. The default value is 0.95.
- `cutoff.eli.efficacy` The cutoff value for the futility rule, the acceptable lowest efficacy. The default value is 0.30.
- `w1.toxicity` The weight for toxicity utility function (4) and (5) The recommended ρ is the target toxicity rate.
- `w2.toxicity` The weight for toxicity utility funciton (5).
- `indicator` The indicator cutoff for utility funciton (5).

Assume that the number of patients treated at four doses is $npts = (6, 6, 12, 3)$, the corresponding number of patients who experienced toxicity is $ntox = (0, 1, 2, 2)$, and the corresponding number of patients who experienced efficacy is neff=(4,3,6,1)

```
R> target.toxicity=0.3
R> target.efficacy=0.4
R> npts <- c(3,6,12,3,3)
R> ntox <-  c(1,2,4,2,3)
R> neff <-  c(0,0,5,1,1)
R> obd <- select.obd.kb (target.toxicity=target.toxicity,
+      target.efficacy= target.efficacy, npts = npts,
+      ntox = ntox, neff =  neff)

The OBD using the 1st utility function is dose level  3.
The OBD using the 2nd utility function is dose level  3.
The OBD using the 3rd utility function is dose level  3.
```

3.4. Option 2: Phase I/II trials aiming to find OBD

We also provide a function, without requiring the investigator's efforts to pre-specify multiple input parameters/arguments, e.g., `toxicity.low, toxicity.moderate, ···, efficacy.moderate, efficacy.high` in the function `get.decision.obd.kb(.)`, here, the investigator is only required to provide the DLT target rate, efficacy target rate, the cohort size, total number of cohorts to automatically generate the boundary and decision table and meanwhile provide , which are generated based on both toxicity and efficacy dose escalation and de-escalation rules.

To generate a default parameter set of `toxicity.low, toxicity.moderate, ···, efficacy.moderate, efficacy.high`, here, we used the Bayesian optimal interval design (BOIN) as a tool to automatically divide the toxicity and efficacy intervals. To be specific, given the toxicity/DLT target rate, the BOIN algorithm will produce two cutoffs to divide the toxicity interval into `toxicity.low, toxicity-acceptable, toxicity-high` sub-intervals. For example, if the toxicity target rate is 0.2, by executing the following code, the toxicity interval can be divided into three sub-intervsls, (0, 0.16), (0.16, 0.24), and (0.24, 1), corresponding to the `toxicity.low, toxicity-acceptable, toxicity-high` sub-intervals.

```
R> > library(BOIN)
> bound <- get.boundary(target=0.2, ncohort=10, cohortsize=3)
> bound$lambda_e
[1] 0.1572423
> bound$lambda_d
[1] 0.2384624
```

For getting the sub-intervals for the efficacy, we would firstly input the efficacy failure rate to the above `get.boundary(.)` function to get the two cutoffs. The efficacy failure rate is defined as 1-`efficacy.target.rate`. Then, we use 1 minus the two computed cutoffs and reverse to increasing order to find the sub-intervals. The reason we take this approach is that the BOIN algorithm was originally proposed to find the MTD based on the rationale of higher dose indicating more toxicity, which is undesirable while in the setting of the efficacy, higher

the efficacy is desirable. Actually, a similar strategy was also used in Zohar's research [30].

To give an example, if the target efficacy rate is 0.4, by executing the following code, the efficacy interval can be divided into: (0, 1-0.73), (1-0.73, 1-0.48), (1-0.48, 1), that is, (0, 0.27), (0.27, 0.52), and (0.52, 1) correspond to the `efficacy.low, efficacy-acceptable, efficacy-high` sub-intervals.

```
R> bound <- get.boundary(target=0.6, ncohort=10, cohortsize=3)
> bound$lambda_e
[1] 0.4791901
> bound$lambda_d
[1] 0.7314159
```

From the above, we can see that by using this default approach, there are three sub-intervals for toxicity:

- The low interval for toxicity is (0, toxicity.lower.boundary).
- The acceptable interval for toxicity is (toxicity.lower.boundary, toxicity.upper.boundary).
- The high interval for toxicity is (toxicity.upper.boundary, 1).

and, there are three sub-intervals for efficacy:

- The low interval for efficacy is (0, efficacy.lower.boundary).
- The acceptable interval for efficacy is (efficacy.lower.boundary, efficacy.upper.boundary).
- The high interval for efficacy is (efficacy.upper.boundary, 1).

Shown here is an example of boundary tables designed using this method with a target toxicity rate of 0.2 and a target efficacy rate of 0.4:

		Efficacy.low	Efficacy.moderate	Efficacy.high
		(0,0.27)	(0.27,0.52)	(0.52,1)
Toxicity.low	(0,0.16)	E	E	S
Toxicity.moderate	(0.16,0.24)	S	S	S
Toxicity. high	(0.24,1)	D	D	D

For example, the interval combination (0, 0.16) × (0.27,0.52) corresponds to a decision "S". This means that the next cohort of patients will be treated at the current dose level if the observed toxicity rate of current dose falls in (0, 0.16) and the observed efficacy rate falls in (0.27,0.52) .

Design and conduct the trial

To design a phase I/II trial for finding the OBD, we need to run the function `get.decision.obd2.kb(·)` to obtain the boundary table and decision table, which are we need to run the trial. This function has the following arguments:

- `target.toxicity` The target DLT rate.
- `target.efficacy` The target efficacy rate.
- `cohortsize` The number of patients in the cohort.
- `ncohort` The total number of cohorts.
- `decision` The pre-specified decisions.
- `cutoff.eli.toxicity` The cutoff value to eliminate a dose with unacceptably high toxicity for safety. The default value is 0.95.
- `cutoff.eli.efficacy` The cutoff value to eliminate a dose with unacceptably low efficacy. The default value is 0.3.

As an example, suppose we want to conduct a phase I/II trial with $J = 5$ dose levels, a target toxicity rate of $target.toxicity = 0.2$, and a target efficacy rate of $target.efficacy = 0.4$.

Patients are treated in cohorts of size 3 and ncohort is 10. Using the default values of `cutoff.eli.toxicity`, and `cutoff.eli.efficacy` provided by the function, we can design a trial by running `get.decision.obd2.kb(·)`, as follows:

```
R> decision <- get.decision.obd2.kb(target.toxicity=0.2,
+           target.efficacy=0.4, cohortsize=3, ncohort=10,
+           decision=c("E", "E", "S", "S", "S", "S", "D", "D", "D"))
```

T1	T2	EF1	EF2	DECISION
0.00	0.16	0.00	0.27	E
0.00	0.16	0.27	0.40	E
0.00	0.16	0.40	0.52	E
0.00	0.16	0.52	1.00	E
0.16	0.25	0.00	0.27	E
0.16	0.25	0.27	0.40	E
0.16	0.25	0.40	0.52	E
0.16	0.25	0.52	1.00	S
0.25	0.40	0.00	0.27	D
0.25	0.40	0.27	0.40	S
0.25	0.40	0.40	0.52	S
0.25	0.40	0.52	1.00	S
0.40	1.00	0.00	0.27	D
0.40	1.00	0.27	0.40	D
0.40	1.00	0.40	0.52	D
0.40	1.00	0.52	1.00	D
0.40	1.00	0.52	1.00	D

Partial output of the decision matrix is given as the follows:

N	T	R	Decision

3	0	0	EUE
3	0	1	E
3	0	2	E
3	0	3	E
3	1	0	S
3	1	1	S
3	1	2	E
3	1	3	E
3	2	0	DUT
3	2	1	DUT
3	2	2	DUT
3	2	3	DUT
3	3	0	DUT
3	3	1	DUT
3	3	2	DUT
3	3	3	DUT
6	0	0	EUE
6	0	1	EUE
6	0	2	E
6	0	3	E
6	0	4	E
6	0	5	E
6	0	6	E
6	1	0	S
6	1	1	S
6	1	2	S
6	1	3	S
6	1	4	E
6	1	5	E

6	1	6	E
6	2	0	S
6	2	1	S
6	2	2	S
6	2	3	S
6	2	4	E
6	2	5	E
6	2	6	E
6	3	0	DUT
6	3	1	DUT
6	3	2	DUT
6	3	3	DUT
6	3	4	DUT
6	3	5	DUT
6	3	6	DUT
6	4	0	DUT
6	4	1	DUT
6	4	2	DUT
6	4	3	DUT
6	4	4	DUT
6	4	5	DUT
6	4	6	DUT
6	5	0	DUT
6	5	1	DUT
6	5	2	DUT
6	5	3	DUT
6	5	4	DUT
6	5	5	DUT
6	5	6	DUT

```
6     6     0       DUT
6     6     1       DUT
6     6     2       DUT
6     6     3       DUT
6     6     4       DUT
6     6     5       DUT
6     6     6       DUT
```

- D: deescalate to the previous dose level j-1, assuming that the current dose level is j and j is not the lowest dose level. Otherwise, treat the next cohort of patients at the current dose level j.
- E: escalate to the next dose level j+1, assuming that the current dose level is j and j is not the highest dose level. Otherwise, treat the next cohort of patients at the current dose level j.
- S: stay at the current dose level j (i.e., treat the next cohort of patients at dose level j).
- EUE: escalate to the next higher dose level because of unacceptably low efficacy.
- DUE: deescalate to the previous lower admissible dose level because of unacceptably low efficacy.
- DUT: deescalate to the previous lower admissible dose level because of unacceptably high toxicity.
- Default stopping rule: if $Pr(p_i > p_T|D)$ is too high (the default is 0.95), then any dose higher than i or equal to i is excluded from the trial.
- Default futility rule: if $Pr(q_i > q_E|D)$ is too low (the default value is 0.3), then dose i is excluded from the trial.

Obtain the operating characteristics

The function `get.oc.obd2.kb(·)` can be used to obtain the operating characteristics of

the design. This function takes the following arguments:

- `target.toxicity` The target DLT rate.
- `target.efficacy` The target efficacy rate.
- `ncohort` The total number of cohorts.
- `cohortsize` The number of patients in the cohort.
- `n.early` The early stopping parameter. If the number of patients treated at the current dose reaches $n.early$, then we stop the trial and select the MTD based on the observed data. The default value is 100.
- `startdose` The starting dose level.
- `p.true` A vector containing the true toxicity probabilities of the investigational dose levels.
- `q.true` A vector containing the true efficacy probabilities of the investigational dose levels.
- `ntrial` The total number of trials to be simulated.
- `seed` The random seed for simulation.
- `p1` The cutoff lower limit for safety utility funciton (3): $U = f(p) \times f(q)$. f(p) is the function for toxicity probability, and f(q) is the function for efficacy probability.
- `p2` The cutoff upper limit for safety utility function .
- `q1` The cutoff lower limit for efficacy utility funciton (3).
- `q2` The cutoff upper limit for efficacy utility funciton (3).
- `cutoff.eli.toxicity` The cutoff value to eliminate a dose with unacceptable high toxicity for safety. The default value is 0.95.

- `cutoff.eli.efficacy` The cutoff value for the futility rule, the acceptable lowest efficacy. The default value is 0.30.

- `w1.toxicity` The weight for toxicity utility functions (4) and (5) , utility function (4): $U_j = \pi_{E,j} - w_1 \times \pi_{T,j}$, and utility function (5): $U_j = \pi_{E,j} - w_1 \times \pi_{T,j} - w_2 \times \pi_{T,j} \times (\pi_{T,j} > \rho)$. The recommended ρ is the target toxicity rate.

- `w2.toxicity` The weight for toxicity utility funciton (5).

- `indicator` The indicator cutoff value for utility funciton (5).

Using the same setting as above and assuming that the true toxicity scenario is `p.true` $= (0.08, 0.30, 0.60, 0.80)$, the true efficacy scenario is `q.true` $= (0.25, 0.40, 0.25, 0.50)$, here we show how to obtain the operating characteristics based on 1000 simulated trials.

```
R> target.toxicity <- 0.30
R> target.efficacy <- 0.40
R> p.true <-c(0.08,0.30,0.60,0.80)
R> q.true <- c(0.25,0.40,0.25,0.50)
R> oc.obd2.kb <- get.oc.obd2.kb(target.toxicity=target.toxicity,
+             target.efficacy= target.efficacy, ncohort=20,
+             cohortsize= 3,  p.true= p.true, q.true= q.true)

selection percentage (%) at each dose level using the 1st utility:
30.0 46.7  1.0  0.0
selection percentage (%) at each dose level using the 2nd utility :
31.0 47.0  1.5  0.1
selection percentage (%) at each dose level using the 3rd utility:
34.7 44.1  0.8  0.0
number of patients treated at each dose level:  9.384 27.789 20.553  2.166
number of toxicities observed at each dose level:  0.737  8.374 12.237  1.756
number of responses observed at each dose level:  2.339 11.218  5.024  1.055
```

```
average number of toxicities: 23.104
average number of responses:   19.636
average number of patients: 59.892
the percentage of early stop (%): 0.2
the percentage of trial termination using the 1st utility function (%): 22.3
the percentage of trial termination using the 2nd utility function (%): 20.4
the percentage of trial termination using the 3rd utility function (%): 20.4
```

Select the OBD when the trial is completed

The selection of the OBD uses the same function as in section 3.3.3.

4. Trial examples

4.1. Single-agent phase I trial

Consider a single-agent phase I trial with 5 dose levels, in which the objective is to find the MTD with a target DLT rate of 0.3. The maximum sample size is 30 patients, treated in cohorts with the size of 3. To design and conduct this trial, we first ran function `get.boundary.kb(target=0.3, ncohort=10, cohortsize=3)`, yielding the dose escalation and de-escalation boundaries as shown in Table 3 .

Table 3: Dose escalation and de-escalation rule for the Keyboard design.

	Number of patients treated									
	3	6	9	12	15	18	21	24	27	30
Escalate if number of DLT <=	0	1	2	2	3	4	5	5	6	7
De-escalate if number of DLT >=	2	3	4	5	6	7	8	9	10	11
Eliminate if number of DLT >=	3	4	5	7	8	9	10	11	12	14

The trial started by treating the first cohort of 3 patients at dose level 1, and none of the patients had DLT. According to the dose escalation and de-escalation rule provided in Table 3, we escalated the dose to level 2 to treat the second cohort of 3 patients; none of whom experienced DLT. Thus, we escalated the dose to level 3 and treated the third cohort

of patients, 2 of whom experienced DLT. On the basis of Table 3, we de-escalated the dose back to level 2 and treated the fourth cohort of patients, one of whom experienced DLT. We then escalated the dose to level 3 and treated the fifth cohort of patients, none of whom experienced DLT. Therefore, the sixth cohort was also treated at dose level 3. Figure 3 shows the dose assignment path for all 30 patients.

At the end of the trial, the number of patients and the number of DLTs at the 5 doses were `n=c(3, 6, 18, 3, 0)` and `y=c(0, 1, 5, 3, 0)`, respectively. We called function `select.mtd.kb(target=0.3, ntox=y, npts=n)`, which recommended dose level 3 as the MTD, with the estimated DLT rate is 28.0% and the 95% CI =is (10%, 50%).

4.2. Drug-combination trial to find a single MTD

Consider a drug-combination trial that combines 3 doses of agent A and 5 doses of agent B. The objective is to find a MTD with a target DLT rate of 0.3. The maximum sample size is 30 patients, treated in cohorts with the size of 3. The trial started by treating the first cohort of 3 patients at the lowest dose combination (1,1), at which no DLT was observed. The observed data were

$$n = \begin{pmatrix} 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \qquad y = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

where n records the number of patients treated at each dose combination, and y records the number of patients who experienced DLT at each dose combination. In matrices y and n, entry (j, k) records the data associated with combination (j, k). To determine the dose for the second cohort of patients, we called function `next.comb.kb(target=0.3, npts=n, ntox=y, dose.curr=c(1, 1))`, which recommended escalating the dose to combination (2,1). Therefore, we treated the second cohort of patients at dose combination (2,1). In the second

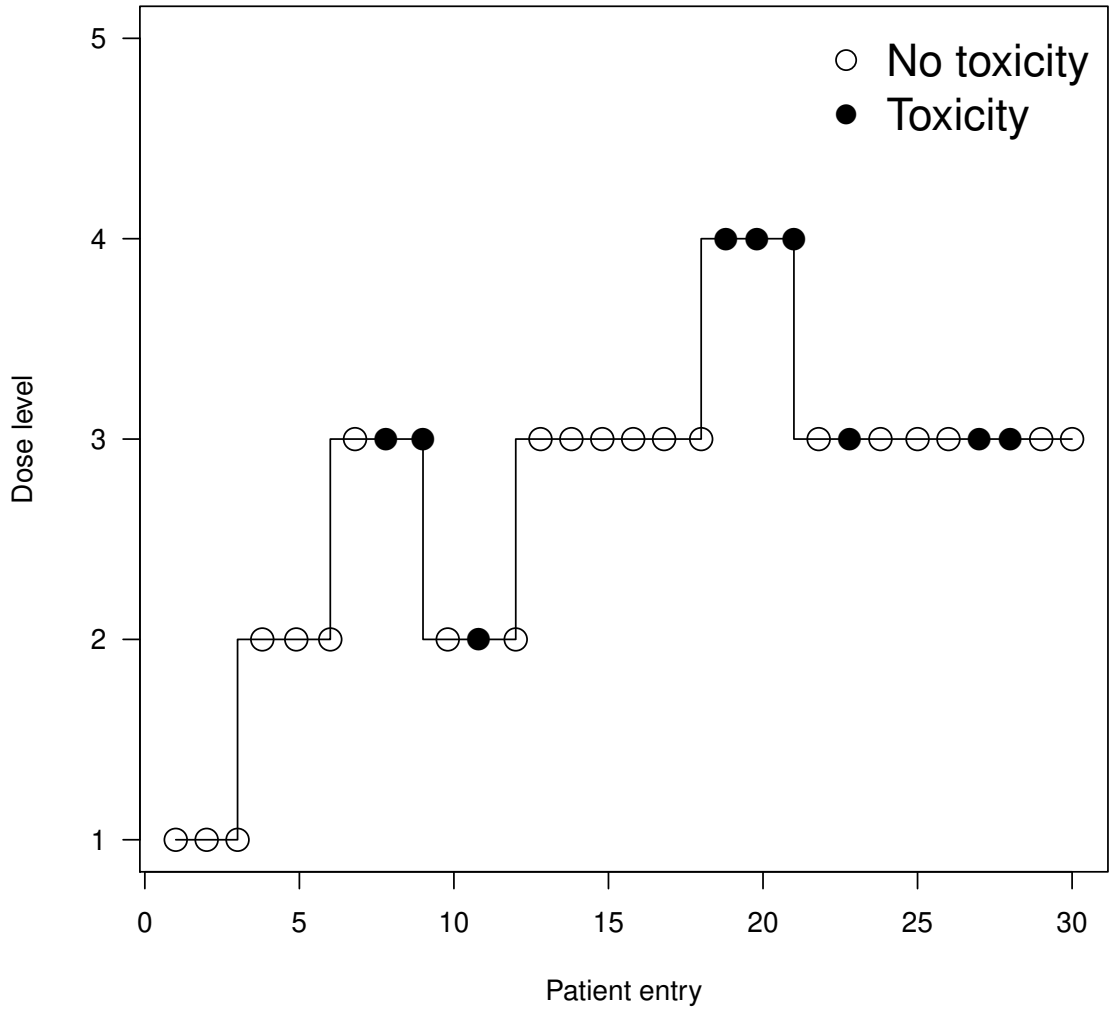

Figure 3: Illustration of a single-agent phase I trial using the Keyboard design for single-agent MTD-finding trials

cohort, none experienced DLT, so the updated data matrices became

$$n = \begin{pmatrix} 3 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \qquad y = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

To determine the dose for the third cohort of patients, we again call `next.comb.kb(target=0.3, npts=n, ntox=y, dose.curr=c(2,1))` with updated `y`, `n` and `dose.curr`. The function recommended escalating the dose to (3,1) for treating the third cohort of patients. We repeated this procedure until the maximum sample size was reached. Figure 4 shows the dose assignments path for all 30 patients. For example, at dose combination (3,2) , there were 0 DLTs; the function recommended escalating the dose to combination (3,3). When the trial was completed, the number of patients treated at each dose combination and the corresponding number of patients who experienced toxicity at each dose combination were

$$n = \begin{pmatrix} 3 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 \\ 3 & 3 & 3 & 12 & 6 \end{pmatrix}, \qquad y = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 4 & 0 \end{pmatrix}.$$

We called function `select.mtd.comb.kb(target=0.3, npts=n, ntox=y)`, which recommended dose combination (3, 3) as the MTD.

4.3. Single-agent Phase I/II trial to find the OBD

Consider a single-agent phase I trial with 5 dose levels, in which the objective is to find the OBD with a target DLT rate of 0.20 and a target efficacy rate of 0.40. The maximum sample size is 30 patients, treated in cohorts with the size of 3. To design and conduct this trial, we first ran function `get.decision.obd.kb(toxicity.low =0.15, toxicity.moderate = 0.25, toxicity.high = 0.35, efficacy.low = 0.25, efficacy.moderate = 0.45, efficacy.high =0.65,target.toxicity=0.20, target.efficacy=0.40, ncohort=10, cohortsize=3)`, yielding the dose escalation and de-escalation boundaries, part of which is

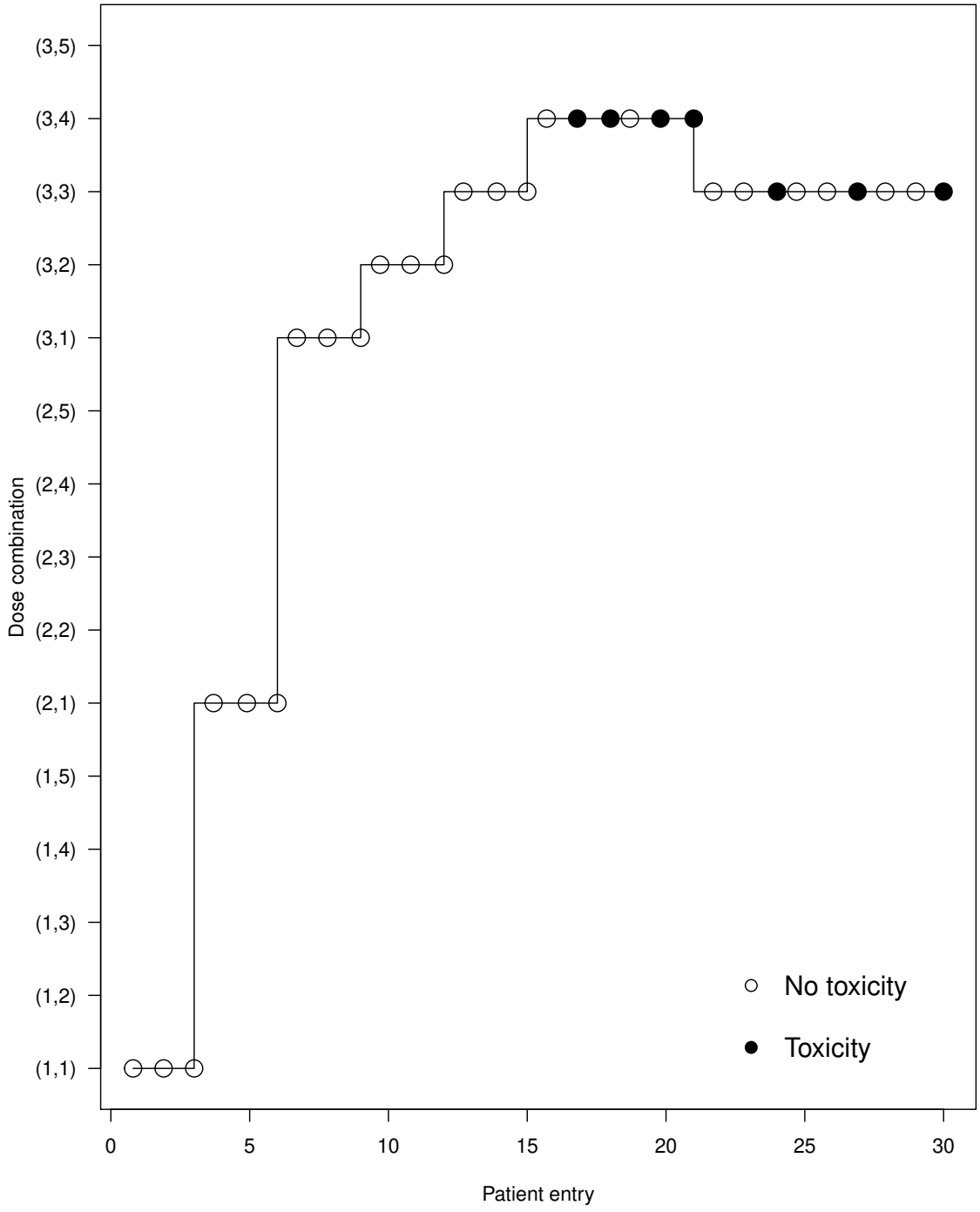

Figure 4: Illustration of the dual-agent combination design of finding the MTD for a 3×5 combination trial with a cohort of 3. Open circles indicate patients without toxicity, and solid circles denote patients with toxicity.

Table 4: Dose escalation and de-escalation rules for the phase I/II OBD-finding design

N	T	R	Decision
3	0	0	EUE
3	0	1	E
3	0	2	E
3	0	3	E
3	1	0	DUE
3	1	1	S
3	1	2	S
3	1	3	S
3	2	0	DUE
3	2	1	D
3	2	2	D
3	2	3	D
3	3	0	DUT
3	3	1	DUT
3	3	2	DUT
3	3	3	DUT
6	0	0	EUE
6	0	1	EUE
6	0	2	E
6	0	3	E
6	0	4	E
6	0	5	E
6	0	6	E
6	1	0	EUE
6	1	1	EUE
6	1	2	E
6	1	3	E
6	1	4	E
6	1	5	S
6	1	6	S

shown in Table 4.

The trial started by treating the first cohort of 3 patients at dose level 1; none of the patients had dose-limiting toxicity (DLT), and two patients showed response. We then ran function `decision.finding(out.matrix = output.matrix, n=3, t=0,r=2)` to find the next recommend dose level.

The output was "E", so we escalated the dose to level 2 to treat the second cohort of 3 patients, 1 of whom experienced DLT and 2 of whom experienced an efficacy response. To be noted, `output.matrix` is the decision.matrix from the above Table 4, which can be extracted by `$decision.matrix`. Up-to-now, there are total 6 patients with 2 DLTs and

3 responses, we ran again the function `decision.finding(out.matrix = output.matrix, n=6,t=1, r=4)`. The output was "E", so we escalated the dose to level 3 to treat the third cohort of 3 patients, 1 of whom experienced DLT and 3 of whom experienced an efficacy response. We repeat this process until maximum sample size reached or stopping rules are satisfied. The complete dose-transition path is shown in Figure 5.

At the end of the trial, the number of patients, the number of DLTs, and the number of responses at the 5 doses were `n=c(3,3,15,9,0)`, and `t=c(0,1,3,8,0)`, and `r=c(2,2,11,3,0)`, respectively. We called function `select.obd.kb(target.toxicity=0.2, target,efficacy=0.4, npts=n,ntox=t, neff=r)`. The recommended dose level using utility function (3), (4), and (5) is dose level 1, 3, 1, respectively.

5. Conclusion

The Keyboard design, as a model-assisted design, combines the advantages of both the algorithm-based design (simplicity) and the model-assisted design (superior performances). The Keyboard designs are novel early-phase trial designs that can be implemented simply and transparently, similar to the 3+3 design, but yield excellent performance comparable to those of more complicated, model-based designs. The developed Keyboard package provides tools for designing, conducting, and analyzing single-agent, drug-combination, and phase I/II dose-finding clinical trials.

6. Availability and requirements

Project name: Keyboard

Project home page: https://cran.r-project.org/web/packages/Keyboard/index.html

Operating system(s): Platform independent

Programming language: R

Other requirements: R 3.4.0 or above

License: GPL-2

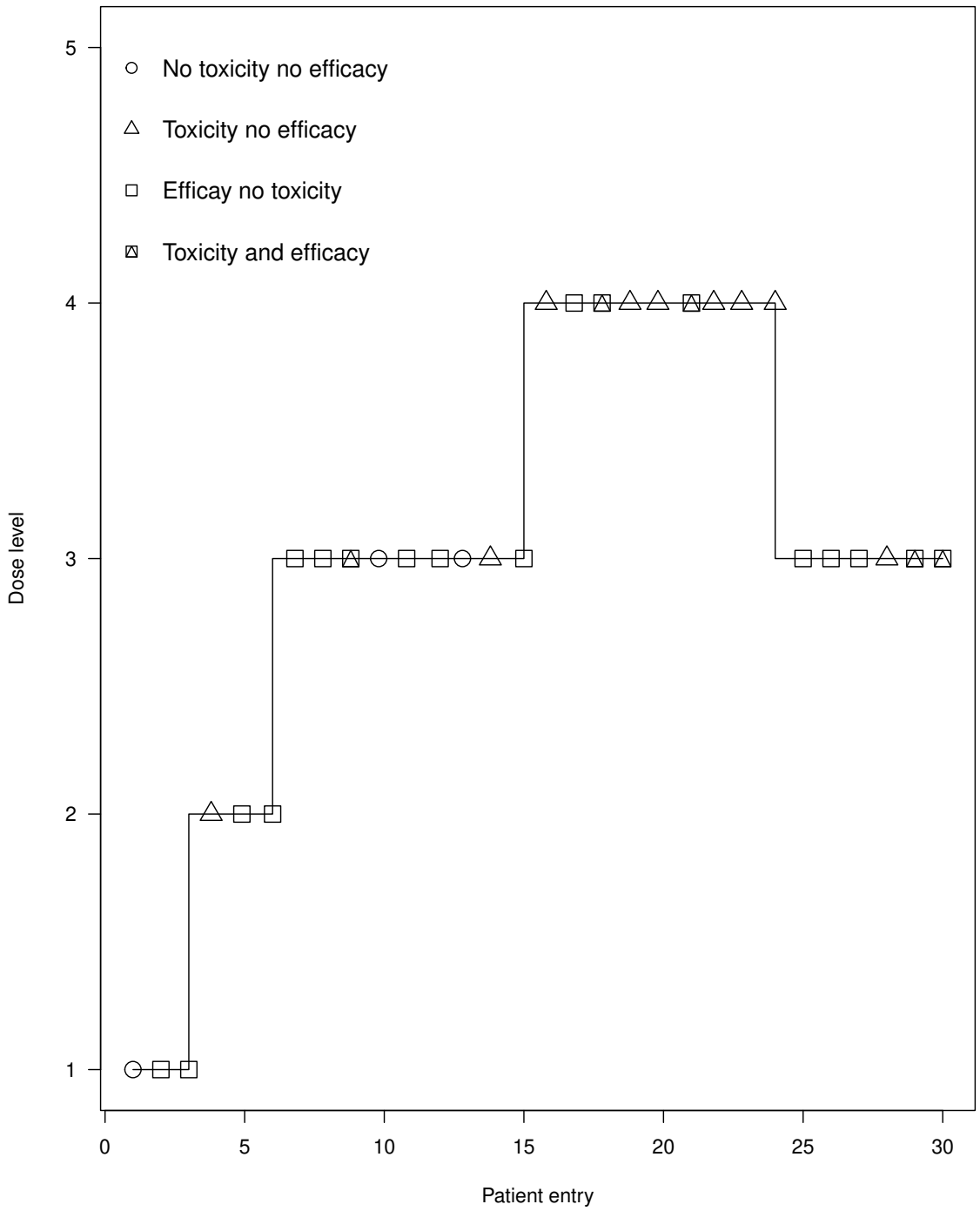

Figure 5: Illustration of a single-agent phase I/II trial OBD-finding design

Any restrictions to use by non-academics: none

7. Abbreviations

MTD:max-imum tolerated dose; OBD:optimal biological dose; DLT: dose-limiting toxicity; CRM:continual reassessment method; TEPI:toxicity and efficacy probability interval; JUPM: joint unit probability mass; BOIN: bayesian optimal interval design

8. Authors' contributions

PHt and LT contributed to the conception and design of the work. SHy, LC and CC coded the software. SHy and LC drafted the first version of the manuscript. All five authors (SHy, LC, CC, LT and PHt) substantially revised it and approved the manuscript.

9. Funding

Cheng's work was partially supported by NIH/NCI cancer Center Support Grant CA21765. Li's work was partially supported by National Natural Science Foundation of China Grant 81803328. Pan, Tang, Sun, and Cheng's work were partially supported by American Lebanese Syrian Associated Charities (ALSAC).

10. Competing interests

The authors declare that they have no competing of interests.

11. Contact

Hongying Sun and Chen Li are co-first authors.
Contact Haitao Pan at haitao.pan@stjude.org or li.tang@stjude.org with any

questions.

References

[1] Storer BE: Design and analysis of phase I clinical trials. Biometrics 1989, 45(3):925-937.

[2] Durham SD, Rosenberger FWF: A random walk rule for phase I clinical trials. Biometrics 1997, 53(2):745-760.

[3] Ivanova A, Montazer-Haghighi A, Mohanty SG, Durham SD: Improved up-and-down designs for phase I trials. Statistics in Medicine 2003, 22(1):69-82.

[4] Stylianou M, Follmann DA: The accelerated biased coin up-and-down design in phase I trials. Journal of Biopharmaceutical Statistics 2004, 14(1):249-260.

[5] O'Quigley J, Pepe M, Fisher L: Continual reassessment method: a practical design for phase 1 clinical trials in cancer. Biometrics 1990, 46(1):33-48.

[6] Cheung K, Ying: Dose Finding by the Continual Reassessment Method; 2011.

[7] Ji Y, Liu P, Li Y, Nebiyou Bekele B: A modified toxicity probability interval method for dose-finding trials. Clinical Trials 2010, 7(6):653.

[8] Liu S, Yuan Y: Bayesian optimal interval designs for phase I clinical trials. Journal of the Royal Statistical Society 2015, 64.

[9] Fangrong, Yan, Sumithra, Mandrekar, Ying, Yuan: Keyboard: A Novel Bayesian Toxicity Probability Interval Design for Phase I Clinical Trials. Clinical Cancer Research An Official Journal of the American Association for Cancer Research 2017.

[10] Thall PF, Cook JD: Dose-Finding Based on Efficacy-Toxicity Trade-Offs. Biometrics 2004, 60(3).

[11] Yin G, Ji LY: Bayesian Dose-fiinding in Phase I/II Clinical Trials Using Toxicity and Efficacy Odds Ratios. Biometrics 2006, 62(3):777-784.

[12] Zang, Lee, J. J, Yuan: Adaptive designs for identifying optimal biological dose for molecularly targeted agents. Clinical Trials Journal of the Society for Clinical Trials 2014, 11(3):319-327.

[13] Suyu L, Johnson VE: A robust Bayesian dose-finding design for phase I/II clinical trials. Biostatistics (2):2.

[14] Hiroyuki, Sato, Akihiro, Hirakawa, Chikuma, Hamada: An adaptive dose-finding method using a change-point model for molecularly targeted agents in phase I trials. Statistics in Medicine 2016.

[15] Li DH, Whitmore JB, Guo W, Ji Y: Toxicity and Efficacy Probability Interval Design for Phase I Adoptive Cell Therapy Dose-Finding Clinical Trials. Clinical cancer research : an official journal of the American Association for Cancer Research 2017, 23(1):13-20.

[16] Lin R, Yin G: STEIN: A simple toxicity and efficacy interval design for seamless phase I/II clinical trials. Statistics in Medicine 2017.

[17] Lin R, Zhou Y, Yan F, Li D, Yuan Y: BOIN12: Bayesian Optimal Interval Phase I/II Trial Design for Utility-Based Dose Finding in Immunotherapy and Targeted Therapies. JCO precision oncology 2020, 4.

[18] Li DH, Whitmore J, Ji Y: Abstract 423: Toxicity and efficacy probability intervals design for phase I dose-finding in oncology trials. Cancer Research 2016, 76(14 Supplement):423-423.

[19] Zhou Y, Lee JJ, Yuan Y: A utility-based Bayesian optimal interval

(U-BOIN) phase I/II design to identify the optimal biological dose for targeted and immune therapies. Statistics in Medicine 2019, 38(28).

[20] Shi H, Cao J, Yuan Y, Lin R: uTPI: A utility-based toxicity probability interval design for phase I/II dose-finding trials. Statistics in Medicine 2021(28).

[21] Yin G, Yuan Y: Bayesian dose finding in oncology for drug combinations by copula regression. Journal of the Royal Statistical Society: Series C (Applied Statistics) 2009.

[22] Yuan YY: A Latent Contingency Table Approach to Dose Finding for Combinations of Two Agents. Biometrics 2010, 65(3):866-875.

[23] Wages NA, Conaway MR, O'Quigley J: Continual Reassessment Method for Partial Ordering. Biometrics 2015, 67(4).

[24] Yuan Y, Yin G: Sequential continual reassessment method for two-dimensional dose finding. Statistics in Medicine 2010, 27(27):5664-5678.

[25] Zhang L, Yuan Y: A practical Bayesian design to identify the maximum tolerated dose contour for drug combination trials. Statistics in Medicine 2016, 35.

[26] Lin R, Yin G: Bayesian optimal interval design for dose finding in drug-combination trials. Stat Methods Med Res 2017, 26(5):2155-2167.

[27] Pan H, Lin R, Zhou Y, Yuan Y: Keyboard design for phase I drug-combination trials. Contemp Clin Trials 2020, 92:105972.

[28] Liu S, Ning J: A Bayesian Dose-finding Design for Drug Combination Trials with Delayed Toxicities. Bayesian Analysis 2013, 8(3):703-722.

[29] Corbaux P, Madani ME, Tod M, Péron J, You B: Clinical efficacy of the optimal biological dose in early-phase trials of anti-cancer targeted therapies. European Journal of Cancer 2019, 120:40-46.

[30] Zohar S, Resche-Rigon M, Chevret S: Using the continual reassessment method to estimate the minimum effective dose in phase II dose-finding studies: a case study. Clinical Trials 2013, 10(3):414-421.

Figures

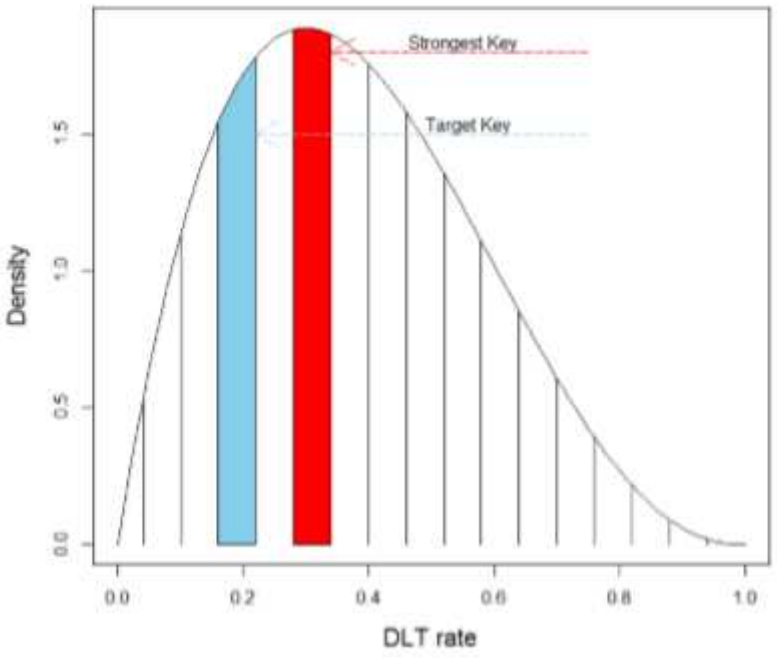

Figure 1

Keyboard design dose-escalation schema

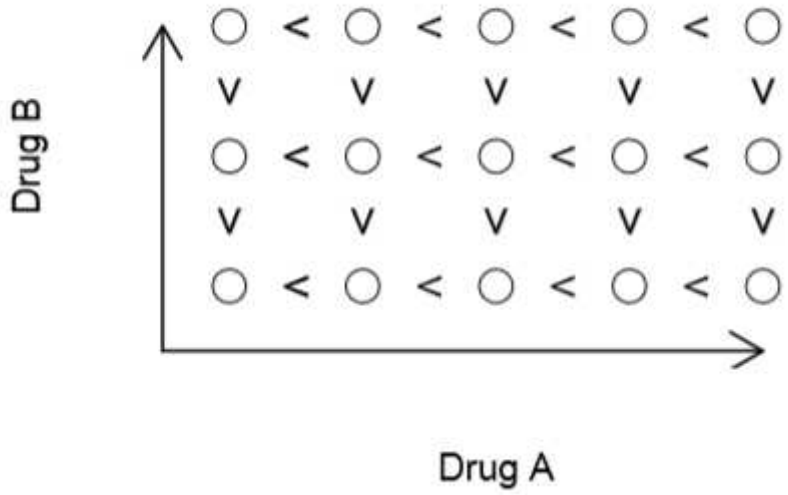

Figure 2

Two dimensional drug-combination trial

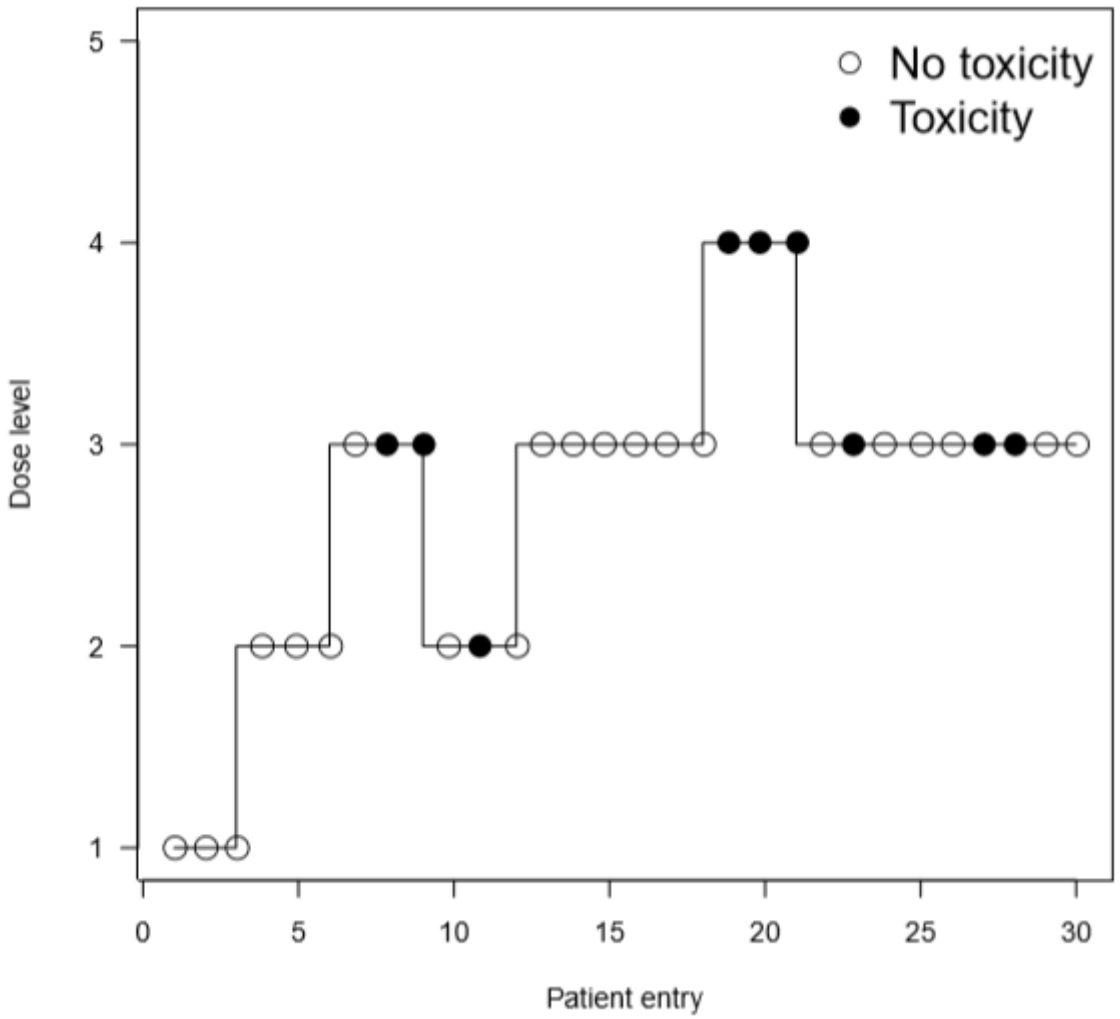

Figure 3

Illustration of a single-agent phase I trial using the Keyboard design for single-agent MTD-finding trials

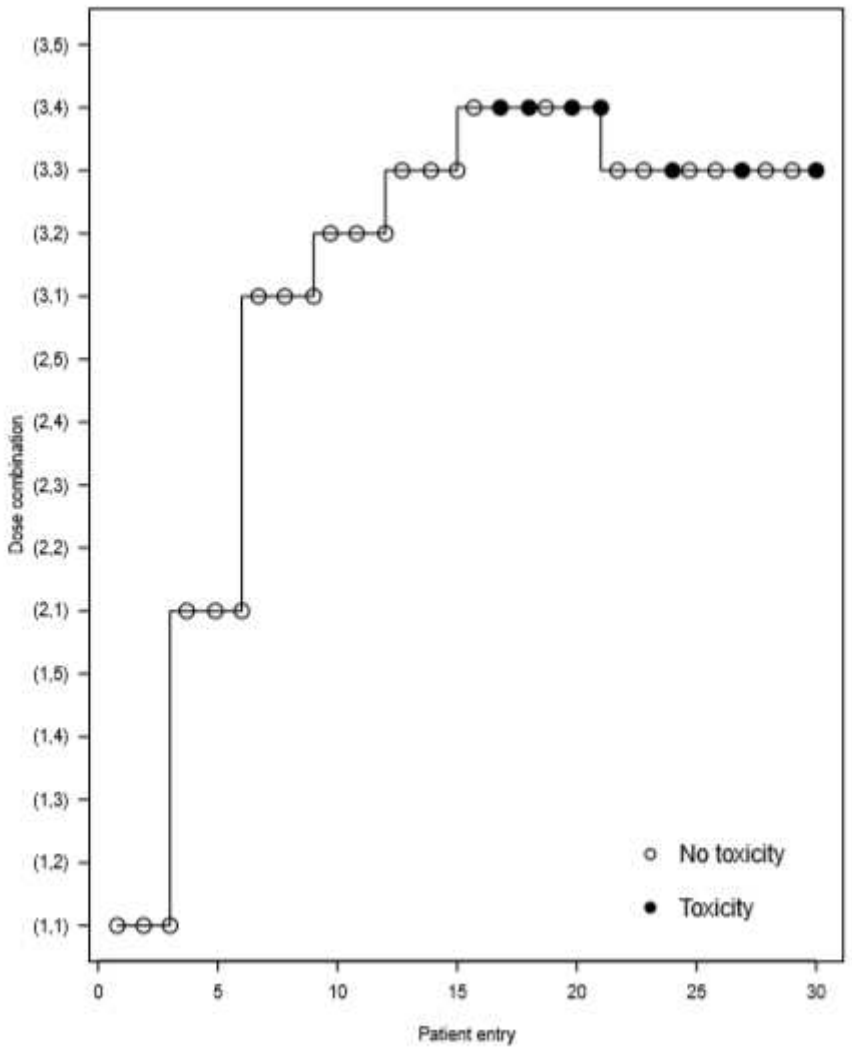

Figure 4

Illustration of the dual-agent combination design of finding the MTD for a 3×5 combination trial with a cohort of 3. Open circles indicate patients without toxicity, and solid circles denote patients with toxicity.

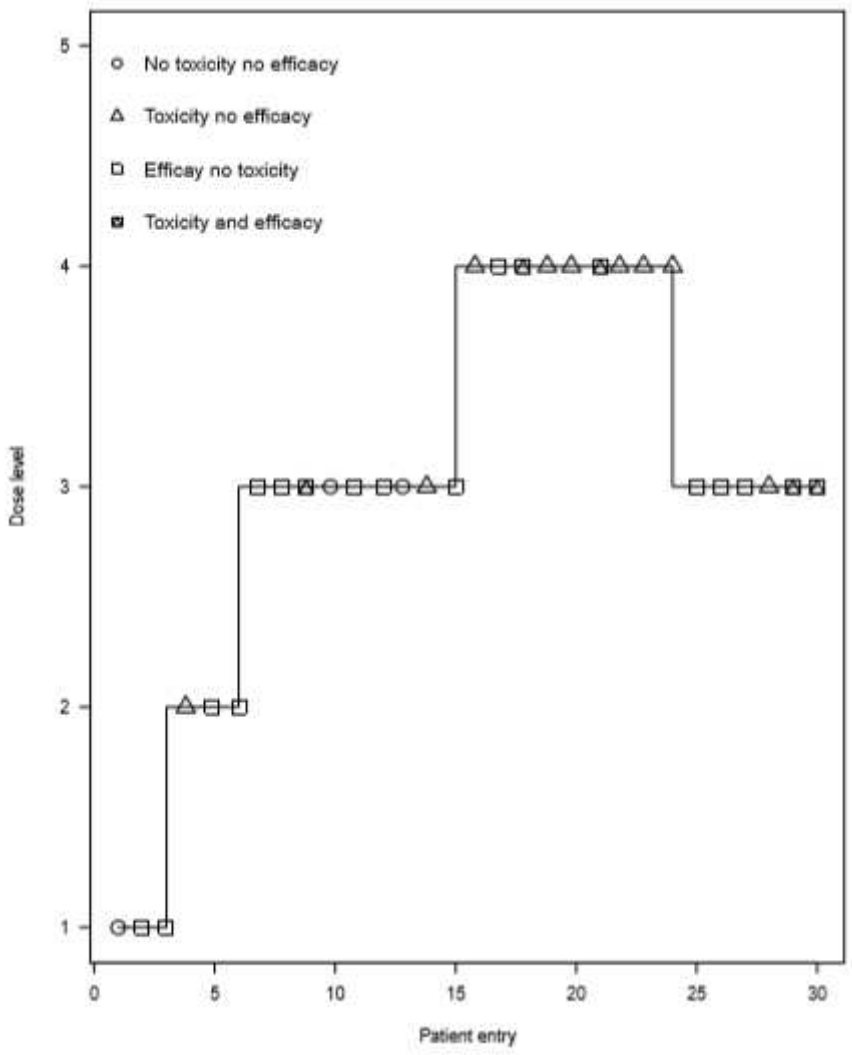

Figure 5

Illustration of a single-agent phase I/II trial OBD-finding design