Statistical Tutorial for Using Bayesian Optimal Interval (BOIN) Design for Phase I Single-Agent and Drug-Combination Trials

Ying Yuan* and Suyu Liu
Department of Biostatistics
The University of Texas MD Anderson Cancer Center
Houston, Texas 77030, U.S.A.
**email: yyuan@mdanderson.org*

Version: 2.4
August 7, 2016

Introduction

The Bayesian optimal interval (BOIN) design is a novel phase I trial design for finding the maximum tolerated dose (MTD). **It can be used to conduct both single-agent and drug-combination trials**. The BOIN design is motivated by the top priority and concern of clinicians, which is to effectively treat patients and minimize the chance of exposing them to subtherapeutic or overly toxic doses. The most important advantage of the BOIN design is that it is easy to implement in practice and has superior operating characteristics. The BOIN design is algorithm-based and thus can be implemented in a way similar to the traditional "3+3" design. The performance of the BOIN design is comparable to that of the continual reassessment method (CRM, one of the best performing model-based phase I trial designs) in terms of selecting the MTD, but has a lower risk of assigning patients to subtherapeutic or overly toxic doses (i.e., better patient ethics).

The idea behind the BOIN design is straightforward. The conduct of phase I trials can be viewed as a sequence of decision-making steps of dose assignment for patients who are sequentially enrolled into a trial. At each moment of decision making, based on the observed data, we take one of three actions: escalate, deescalate or retain the current dose. Under the standard assumption that efficacy monotonically increases with toxicity (for cytotoxic agents), an ideal trial design would escalate the dose when the current dose is below the MTD in order to avoid treating a patient at a subtherapeutic dose level; deescalate the dose when the current dose is above the MTD in order to avoid exposing a patient to an overly toxic dose; and retain the same dose level when the current dose is equal (or close) to the MTD. However, such an ideal design is not available in practice because we do not know whether the current dose is actually below, above or equal (or close) to the MTD, and need to infer that information and make decisions based on the data collected from patients who have been enrolled and treated in the trial. Due to the randomness of the observed data and small sample sizes of phase I trials, the decisions of dose assignment we make are often incorrect, e.g., we may erroneously escalate (or deescalate) the dose when it is actually higher (or lower) than the MTD, which results in overly aggressive (or conservative) dose assignments and treating excessive numbers of patients at dose levels above (or below) the MTD. From a practical and ethical viewpoint, it is highly desirable to minimize these decision errors, such that the actual design behaves as closely as possible to the ideal (error-free) design. The BOIN design is proposed to achieve this goal.

The BOIN design is very simple to implement in practice. In this design, dose transition is defined by the relative location of the observed toxicity rate (i.e., the number of patients who experienced toxicity divided by the total number of patients treated) at the current dose with respect to a prespecified toxicity tolerance interval. If the observed toxicity rate is located within the interval, we retain the current dose; if the observed toxicity rate is greater than the upper boundary of the interval, we deescalate the dose; and if the observed toxicity rate is smaller than the lower boundary of the interval, we escalate the dose.

In other words, to use the BOIN design, we need to specify only the interval (or dose escalation/deescalation) boundaries at the trial design phase, as they are the only design parameters that control dose escalation/deescalation. During the trial conduct, no additional software is needed, and clinicians can simply count the number of patients who experience toxicity and compare the observed toxicity rate with the prespecified dose escalation/deescalation boundaries to determine dose assignment until the trial is completed. In the section that follows, we describe how to obtain the dose escalation/deescalation boundaries using the R package we have provided.

Software

The R package "BOIN" is freely available from CRAN. It contains functions for implementing the BOIN design for both single-agent and drug-combination dose-finding trials.

Single-agent trial

- `get.boundary(···)`; This function is used to generate escalation and deescalation boundaries for conducting trials;
- `select.mtd(···)`; This function is used to select the MTD at the end of the trial based on isotonically transformed estimates;
- `get.oc(···)`; This function is used to generate the operating characteristics of the BOIN design.

Drug-combination trial

- `next.comb(···)`; This function is used to determine the dose combination for the next cohort of new patients for drug-combination trials that aim to find a single MTD;
- `select.mtd.comb(···)`; This function is used to select a MTD or the MTD contour at the end of the trial based on isotonically transformed estimates;
- `get.oc.comb(···)`; This function is used to generate the operating characteristics of the BOIN design or waterfall design for drug combination trials. The waterfall design is an extension of the BOIN design that aims to find the MTD contour (i.e., multiple MTDs in the dose matrix);
- `next.subtrial(···)`; This function is used to obtain the starting dose and the dose-searching space for next subtrial when the current subtrial is completed under the waterfall design.
- `get.oc.comb.phase12`; This function is used to obtain the operating characteristics of phase I-II waterfall design, which aims to find the optimal dose combination (ODC), defined as the combination that has the highest efficacy among the doses in the MTD contour.

1 Single-Agent Trials

1.1 Trial design

To design a single-agent trial, we only need to run the function `get.boundary(·)` to obtain the dose escalation and deescalation boundaries, which are all we need to run the trial. This function has the following arguments:

- `target` the target toxicity rate
- `ncohort` the total number of cohorts
- `cohortsize` the cohort size
- `n.earlystop` the early stopping parameter. If the number of patients treated at the current dose reaches `n.earlystop`, stop the trial early and select the MTD based on the observed data. The default value of `n.earlystop = 100` essentially turns off this type of early stopping.
- `p.saf` the highest toxicity probability that is deemed subtherapeutic (i.e., below the MTD) such that dose escalation should be made. The default value of `p.saf = 0.6 × target`.
- `p.tox` the lowest toxicity probability that is deemed overly toxic such that dose deescalation is required. The default value of `p.tox = 1.4 × target`.
- `cutoff.eli` the cutoff to eliminate the overly toxic dose for safety. We recommend the default value `cutoff.eli = 0.95` for general use.
- `extrasafe` set `extrasafe = TRUE` to impose a stricter stopping rule.
- `offset` a small positive number (between 0 and 0.5) to control how strict the stopping rule is when `extrasafe = TRUE`. A larger value leads to a stricter stopping rule. The default value `offset = 0.05` generally works well.
- `print` to print out the boundary results.

In practice, we should avoid setting the values of `p.saf` and `p.tox` very close to the `target`. This is because the small sample sizes of typical phase I trials prevent us from being able to discriminate the target toxicity rate from the rates close to it. For example, at the significance level of 0.1, there is only 7% power to distinguish 0.25 from 0.35 with a total of 30 patients given just two doses. In addition, in most clinical applications, the target toxicity rate is often a rough guess, and finding a dose level with toxicity rates reasonably close to the target rate is of interest to the investigator. Based on our experience with phase I oncology trials, and also the operating characteristics of the proposed design, we find that $\phi_1 \in [0.5\phi, 0.7\phi]$ and $\phi_2 \in [1.3\phi, 1.5\phi]$ are reasonable values for most clinical applications. As default values, we recommend $\phi_1 = 0.6\phi$ and $\phi_2 = 1.4\phi$ (i.e., 40% deviation from the target) for general use. Although this looks like a large deviation from the target, under typical phase I sample sizes, 40% deviation from the target actually is a rather small difference, and the power to detect such a difference is quite limited. For example, if $\phi = 0.25$, then $\phi_1 = 0.15$ and $\phi_2 = 0.35$. Given a sample size of 30 patients and only two doses, we only have 7% power to distinguish 0.25 from 0.35, and 9% power to distinguish 0.25 from 0.15, based on Fisher's exact test with a significance level of 0.1.

The BOIN design has two built-in stopping rules. (1) Stop the trial if the lowest dose is eliminated due to toxicity. In this case, no dose should be selected as the MTD. (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`), this means that the dose-finding algorithm has approximately converged. Thus, we can stop the trial early and select the MTD to save the sample size and reduce the trial duration. The default value `n.earlystop = 100` essentially turns off this type of early stopping rule. Note that setting `n.earlystop` at a value like 12 can potentially save the sample size and finish the trial early. 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` to `18`. Our experience is that this stopping rule is particularly useful when there is strong prior knowledge that the first dose is safe, since 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 BOIN 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 `p.saf`, `p.tox`, `design` and `cutoff.eli` automatically provided by the function, we can design the trial by running `get.boundary(·)`, as follows:

```
> get.boundary(target=0.3, ncohort=10, cohortsize=3)
Escalate the dose if the observed toxicity rate at the current dose <= 0.2364907
Deescalate the dose if the observed toxicity rate at the current dose >= 0.3585195

This is equivalent to the following 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 4 5 6 7
Deescalate 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 by

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 4 5 5 5
5 6 6 6 6 7
Deescalate if # of DLT >=       1 1 2 2 2 3 3 3 4 4 4 5 5 6 6 6 7 7 7 8 8 8 9
9 9 10 10 11 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

Default stopping rule:  Stop the trial if the lowest dose is eliminated.
```

Remarks:

- The output presents the dose escalation/deescalation rule in two forms: based on the observed toxicity rate, and based on the observed dose-limiting toxicity (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 various reasons. If that is the case, then it

is more appropriate to present the completed version of the decision boundaries in the protocol such that clinicians can make the decision of 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 BOIN design, in that it allows the cohort size to vary from one cohort to another and for the decision of dose escalation/deescalation to be made at any time during the trial conduct.

- 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.

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

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

   ......

In addition to the default stopping rule (i.e., stop the trial if the lowest dose
is eliminated), the following stricter safety stopping rule will be used for extra
safety:
   Stop the trial if (1) the number of patients treated at the lowest dose >=
3 AND (2) Pr(the toxicity rate of the lowest dose > 0.3 | data) > 0.9, which
corresponds to the following stopping boundary:

The number of patients treated at the lowest dose    3 6 9 12 15 18 21 24 27 30
Stop the trial if # of DLT >=                      3 4 5 6 7 8 9 10 12 13
```

1.2 Obtain the operating characteristics

For protocol preparation, it is often useful to obtain the operating characteristics of the design. The function `get.oc(·)` can be used for this purpose. This function shares the same set of arguments as the function `get.boundary(·)` described previously, with three additional arguments:

- `p.true` 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 number of trials to be simulated.

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

```
> get.oc(target=0.3, p.true=c(0.05, 0.15, 0.3, 0.45, 0.6), ncohort=10, cohortsize=3,
ntrial=1000)

selection percentage at each dose level (%):  1.1 23.4 54.2 20.2 1.1
number of patients treated at each dose level:  4.2 9.3 11.0 4.9 0.7
number of toxicities observed at each dose level:  0.2 1.4 3.3 2.2 0.4
average number of toxicities:  7.4
average number of patients:  30.0
percentage of early stopping due to toxicity:  0.0%
risk of poor allocation:  17.9%
risk of overdosing (>60% of patients treated above the MTD): 2.9 %
risk of overdosing (>80% of patients treated above the MTD): 0.0 %
```

Remarks: In the output, *the risk of poor allocation* is defined as the percentage of simulation runs in which the number of patients allocated to the MTD (say n_{MTD}) is less than that of a standard non-sequential design, which assigns equal numbers of patients to each dose, i.e., $\Pr(n_{\text{MTD}} < n/J)$. *The risk of overdosing* is defined as the percentage of simulated trials in which a large percentage (e.g., more than 60% or 80%) of patients are treated at doses above the MTD. These risk measures are of great practical importance because they gauge the likelihood of a trial turning out to be an unsafe trial. This important aspect of trial design has been largely overlooked by the existing literature, which typically focuses only on the mean or average performance of a design.

1.3 Select the MTD when the trial is completed

When the trial is completed, based on the observed data, we can select the MTD using the function `select.mtd(···)`. This function has six arguments: `target`, `npts`, `ntox`, `cutoff.eli`, `extrasafe` and `offset`, where

- `npts` the vector recording the total number of patients treated at each dose level.

- `ntox` the vector recording the number of patients who experienced toxicity at each dose level.

Arguments `cutoff.eli`, `extrasafe` and `offset` are the same as (and should be consistent with) those in functions `get.boundary(·)` and `get.oc(·)`, 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(·)`. 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 $y = (0, 0, 4, 4, 0)$.

```
> n<-c(3, 3, 15, 9, 0)
> y<-c(0, 0, 4, 4, 0)
> select.mtd(target=0.3, npts=n, ntox=y)
The MTD is dose level 3

Dose     Posterior DLT         95%
Level       Estimate     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             ----       ( ----------- )               ----
NOTE: no estimate is provided for the doses at which no patient was treated.
```

The result is that dose level 3 is selected as the MTD. Note that no estimate is provided for dose level 5 because that dose has never been used to treat patients (i.e., `n[5]=0`).

2 Drug-Combination Trials

A fundamental feature of drug-combination trials is that multiple MTDs, i.e., the MTD contour, may exist in the dose matrix. Therefore, when designing drug-combination trials, it is important to distinguish two types of design objective: to find a single MTD or to find the MTD contours (i.e., multiple MTDs). These two objectives require different design strategies. In what follows, we first describe the design for finding a single MTD, followed the design for finding the MTD contour.

2.1 BOIN design to find a single MTD

The BOIN drug-combination design for finding a single MTD and that for single-agent trials are based on the same statistical principle and share a similar set of design parameters. We strongly recommend that users read the description of the BOIN design for single-agent trials (i.e., Section 1) before reading the following drug-combination design.

2.1.1 Obtain the operating characteristics

For protocol preparation, it is often useful to obtain the operating characteristics of the design. The function `get.oc.comb(·)` can be used for this purpose. This function shares the same set of arguments as the function `get.oc`(·) described on page 7 (for single-agent trials), except that `p.true` is now a matrix (rather than a vector) and `startdose` is a vector of length 2 (rather than a scalar). Note that `get.oc.comb(·)` includes an additional argument: `MTD.contour`, which is used to indicate whether we are interested in finding a MTD or the MTD contour. To find a single MTD, we should set `MTD.contour=FALSE`.

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

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

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

```
> p.true = matrix(c(0.02, 0.04, 0.08, 0.14, 0.08, 0.25, 0.42, 0.48, 0.25, 0.45,
0.50, 0.60), ncol=4, byrow=TRUE)
> get.oc.comb(target=0.25, p.true=p.true, ncohort=16, cohortsize=3, ntrial=1000,
MTD.contour=FALSE)

true toxicity rate of dose combinations:
0.02 0.04 0.08 0.14
0.08 0.25 0.42 0.48
0.25 0.45 0.50 0.60

selection percentage at each dose combination (%):
0.00 1.00 2.80 19.90
4.40 37.60 7.10 1.40
21.80 3.80 0.20 0.00

number of patients treated at each dose combination:
```

```
4.06 3.73 3.22 4.18
6.08 10.04 4.19 1.91
5.85 3.65 0.80 0.30

number of toxicities observed at each dose combination:
0.10 0.15 0.25 0.58
0.47 2.57 1.80 0.89
1.48 1.65 0.39 0.18

average number of toxicities:  10.5
average number of patients:  48.0
selection percentage of MTD: 59.4
percentage of patients treated at the MTD: 33.1
percentage of early stopping due to toxicity:  0.00
```

2.1.2 Conduct the trial

The function `next.comb(·)` 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(·)` shares a similar set of arguments with the function `get.boundary`(·) described previously, with three additional arguments:

- `npts`   the matrix recording the number of patients treated at each dose combination.
- `ntox`   the matrix recording the number of patients who experienced toxicity at each dose combination.
- `dose.curr`   the current dose combination, i.e., 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 drug A and 4 dose levels of drug B, aiming to find the MTD that has the target toxicity rate of 0.25. 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 drug A and the kth dose level of drug B. The trial can be conducted as follows. We start the trial by treating the first cohort of 3 patients at the lowest dose $(1, 1)$. Assuming that none of the patients experienced DLT, the data from the first cohort of patients are given by

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

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(·)`:

```
> n<-matrix(c(3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), ncol=4, byrow=TRUE)
> y<-matrix(c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), ncol=4, byrow=TRUE)
> next.comb(target=0.25, npts=n, ntox=y, dose.curr=c(1, 1))
The recommended dose combination for the next cohort of patients is ( 2 , 1 )
```

Therefore, we escalate the dose and treat the second cohort of patients at dose combination (2, 1). Suppose that one patient in the second cohort experienced DLT, the data matrixes then become

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

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

```
> n<-matrix(c(3, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0), ncol=4, byrow=TRUE)
> y<-matrix(c(0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0), ncol=4, byrow=TRUE)
> next.comb(target=0.25, npts=n, ntox=y, dose.curr=c(2, 1))
The recommended dose combination for the next cohort of patients is ( 1 , 1 )
```

Therefore, we should deescalate the dose and treat the third cohort of patients at dose (1, 1). We repeat this procedure until the maximum sample size is reached.

2.1.3 Select a MTD when the trial is completed

When the trial is completed, based on the observed data, we can select the MTD using the function `select.mtd.comb(·)`. This function has seven arguments: `target`, `npts`, `ntox`, `cutoff.eli`, `extrasafe`, `offset` and `MTD.contour` where the descriptions of `cutoff.eli`, `extrasafe` and `offset` are the same as those in `get.boundary(·)` described on page 4 (for single-agent trials). Since we are interested in finding a single MTD, we should set

`MTD.contour=FALSE`. 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

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

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

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. Note that no estimate is provided for dose combinations that have never been used to treat patients, e.g., (1, 3) or (3, 3).

2.2 Waterfall design to find the MTD contour

The waterfall design (Zhang and Yuan, 2016) is a phase I trial design to find the MTD contour (i.e., multiple MTDs) for drug combination trials. The waterfall design is built upon the sequential design (Yuan and Yin, 2008) and the BOIN design. The basic idea of the waterfall design is to divide the two-dimensional drug-combination dose finding into a series of one-dimensional dose findings, referred to as subtrials. Each subtrial is then conducted using the BOIN design in a sequential fashion. Figure 1 illustrates the waterfall design, see Zhang and Yuan (2016) for more details. The waterfall design inherits the simplicity and robust performance of the BOIN design.

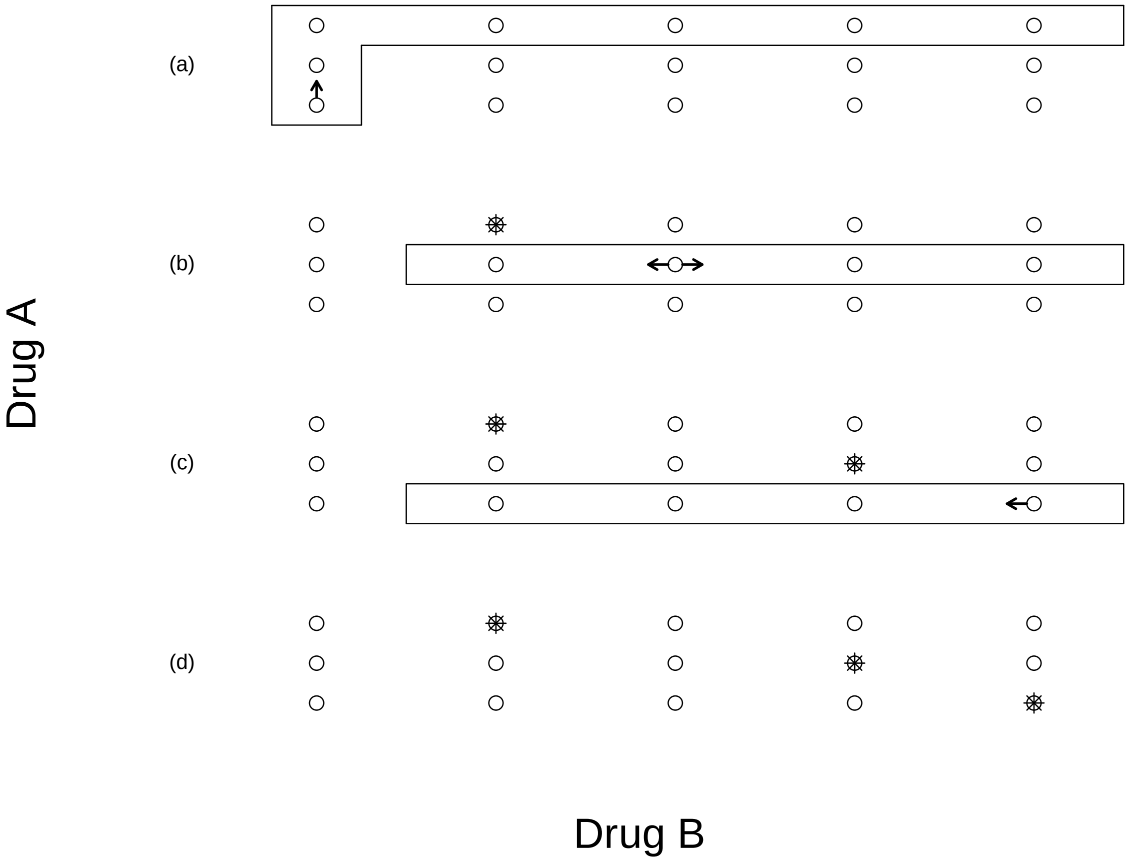

Figure 1: Illustration of the waterfall design for a 3 × 5 combination trial. The doses in the rectangle form a subtrial, and the asterisk denotes the candidate MTD. As shown in panel (a), the trial started by conducting the first subtrial with the starting dose A_1B_1. After the first subtrial identified A_3B_2 as the candidate MTD, we conducted the second subtrial with the starting dose A_2B_3 (see panel (b)). After the second subtrial identified A_2B_4 as the candidate MTD, we conducted the third subtrial with the starting dose A_1B_5 (see panel (c)). After all the subtrials were completed, we selected the MTD contour based on the data from all the subtrials, as shown in panel (d).

2.2.1 Obtain the operating characteristics

The function get.oc.comb(·) can be used to obtain the operating characteristics of the design. This function shares the same set of arguments as the function get.oc(·) described on page 7 (for single-agent trials), except that p.true is now a $J \times K$ matrix (rather than a vector) and startdose is a vector of length 2 (rather than a scalar). We require that $J \leq K$ (i.e., the number of rows $\leq$ the number of columns), such that the dose matrix is in a "landscape" orientation. This is because arranging the dose matrix in the "landscape" orientation leads to fewer subtrials, which simplifies the practical implementation of the waterfall design. If drug A has more dose levels than drug B (i.e., $J > K$), we recommend rotating the dose matrix to make it in the "landscape" orientation. In get.oc.comb(·), argument MTD.contour is used to indicate whether we are interested in finding a MTD or the MTD contour. To find the MTD contour, we should set MTD.contour=TRUE. In addition, the argument ncohort now is a vector of length K that specifies the number of cohorts used for each of K subtrials. As a rule of thumb, we recommend ncohort $\geq$ (the number of doses in the subtrials) $\times$ 4 / cohortsize. For example, for a 5×3 combination, as shown in Figure 1, the first subtrial contains 7 doses, and the second and third subtrials contain 4 doses each. The recommended sample sizes are (at least) 28, 16 and 16 for three subtrials, respectively. If cohortsize=3, then ncohort = c(9, 5, 5) or c(10, 6, 6)

Consider a 3×5 drug combination trial with true toxicity probabilities as follows

$$\texttt{p.true} = \begin{pmatrix} 0.01 & 0.03 & 0.10 & 0.20 & 0.30 \\ 0.03 & 0.05 & 0.15 & 0.30 & 0.60 \\ 0.05 & 0.10 & 0.30 & 0.60 & 0.75 \end{pmatrix},$$

and the target toxicity rate is 30%. Here, we show how to obtain the operating characteristics based on 1000 simulated trials.

```
> 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), ncol=5, byrow=TRUE)
> get.oc.comb(target=0.3, p.true, ncohort=c(10,5,5), cohortsize=3, n.earlystop=12,
startdose=c(1,1), ntrial=1000, MTD.contour=TRUE)

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 (%):
0.00    0.00     1.80    26.40    71.80
```

```
0.20    0.60    22.30    69.60    7.50
3.00   21.30    68.60    6.90    0.00

number of patients treated at each dose combination:
3.10    0.00    0.45    3.46    9.67
3.45    0.28    3.09    8.20   3.23
4.11    6.05    8.82    3.12    0.11

number of toxicities observed at each dose combination:
0.03    0.00    0.05    0.66    2.89
0.11    0.01    0.46    2.42    1.92
0.35    0.56    2.67    1.85    0.09

average number of toxicities:  14.1
average number of patients:  57.1
percentage of patients treated at the MTD contour:  46.7%
percentage of patients treated above the MTD contour:  11.3%
percentage of patients treated below the MTD contour:  42%
percentage of correct selection of the the MTD contour:  36.3%
```

2.2.2 Conduct the trial

The function `next.subtrial(·)` is used to conduct phase I drug-combination trials that aim to find the MTD contour. Specifically, `next.subtrial(·)` outputs the starting dose and the dose-searching space for the next subtrial after the current subtrial is completed. Once we know the staring dose and the dose-searching space for a subtrial, the subtrial can be conducted easily as a single-agent trial using the BOIN design described in Section 1.

As described in Zhang and Yuan (2016) and Figure 1, the first subtrial is prespecified and starts from the lowest dose (1, 1). When the current subtrial is completed, we call `next.subtrial(·)` to determine the starting dose and the dose-searching space for the next subtrial. The function `next.subtrial(·)` shares a similar set of arguments with the function `get.boundary`(·) described previously, with two additional arguments:

- `npts` the $J \times K$ matrix ($J \leq K$) recording the number of patients treated at each dose combination.
- `ntox` the $J \times K$ matrix ($J \leq K$) recording the number of patients who experienced toxicity at each dose combination.

Assume that when the current subtrial is completed, the number of patients treated at each dose combination (i.e., matrix n) and the corresponding number of patients who experienced toxicity at each dose combination (i.e., matrix y) are

$$n = \begin{pmatrix} 6 & 0 & 0 & 0 \\ 6 & 0 & 0 & 0 \\ 9 & 12 & 0 & 0 \end{pmatrix} \qquad y = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 2 & 3 & 0 & 0 \end{pmatrix}.$$

We call `next.subtrial(·)` to determine the dose space for the next subtrial and its starting dose, as follows,

```
> n<-matrix(c(6, 0, 0, 0, 6, 0, 0, 0, 9, 12, 0, 0), ncol=4, byrow=TRUE)
> y<-matrix(c(0, 0, 0, 0, 1, 0, 0, 0, 2, 3, 0, 0), ncol=4, byrow=TRUE)
> next.subtrial(target=.3, npts=n, ntox=y)

Next subtrial includes doses:
(2, 2), (2, 3), (2, 4)

The starting dose for this subtrial is:
(2, 3)
```

That is, the next subtrial consists of three doses $\{(2,2),(2,3),(2,4)\}$. As these doses are monotonically ordered, the BOIN design described in Section 1 for single-agent trials can be directly used here to run this subtrial with the starting dose $(2,3)$, i.e., the second dose level.

2.2.3 Select the MTD contour when the trial is completed

When the trial is completed, based on the observed data, we can select the MTD contour using the function `select.mtd.comb(·)`. This function has seven arguments: `target`, `npts`, `ntox`, `cutoff.eli`, `extrasafe`, `offset` and `MTD.contour` where the descriptions of `cutoff.eli`, `extrasafe` and `offset` are the same as those in `get.boundary(·)` described on page 4 (for single-agent trials). Since we are interested in finding the MTD contour, we should set `MTD.contour=TRUE`.

Assume that the number of patients treated at each dose combination (i.e., matrix n) and the corresponding number of patients who experienced toxicity at each dose combination (i.e., matrix y) are

$$n = \begin{pmatrix} 6 & 9 & 24 & 0 \\ 6 & 24 & 9 & 0 \\ 12 & 18 & 0 & 0 \end{pmatrix} \qquad y = \begin{pmatrix} 0 & 1 & 5 & 0 \\ 1 & 5 & 4 & 0 \\ 1 & 5 & 0 & 0 \end{pmatrix}.$$

```
> n<-matrix(c(6, 9, 24, 0, 6, 24, 9, 0, 12, 18, 0, 0), ncol=4, byrow=TRUE)
> y<-matrix(c(0, 1, 5, 0, 1, 5, 4, 0, 1, 5, 0, 0), ncol=4, byrow=TRUE)
> select.mtd.comb(target=0.3, npts=n, ntox=y, print=TRUE, MTD.contour=TRUE)

The MTD contour includes dose combinations (1, 3) (2, 2) (3, 2)
Isotonic estimates of the toxicity rates are
0.01    0.12    0.21    NA
0.17    0.21    0.45    NA
0.12    0.28      NA     NA
```

The result is that dose combinations (1, 3), (2, 2) and (3, 2) are selected as the MTD contour. Note that no estimate is provided for dose combinations that have never been used to treat patients, e.g., (1, 4) or (3, 3).

3 Contact

If have any questions or comments, please contact Ying Yuan at yyuan@mdanderson.org. You can also visit `http://odin.mdacc.tmc.edu/~yyuan/index_code.html` for more information.

References

Liu S. and Yuan, Y. (2015). Bayesian Optimal Interval (BOIN) Designs for Phase I Clinical Trials, *Journal of the Royal Statistical Society: Series C*, **64**, 507-523.

Lin R. and Yin, G. (2016). Bayesian Optimal Interval Designs for Dose Finding in Drug-combination Trials, *Statistical Methods in Medical Research*, in press.

Yuan, Y. and Yin, G. (2008) Sequential Continual Reassessment Method for Two-dimensional Dose Finding. Statistics in Medicine 27, 5664-5678.

Zhang L. and Yuan, Y. (2016). A Simple Bayesian Design to Identify the Maximum Tolerated Dose Contour for Drug Combination Trials, under review.