ppseq: An R Package for Sequential Predictive Probability Monitoring

by Emily C. Zabor, Brian P. Hobbs, and Michael J. Kane

Abstract Advances in drug discovery have produced numerous biomarker-guided therapeutic strategies for treating cancer. Yet the promise of precision medicine comes with the cost of increased complexity. Recent trials of targeted treatments have included expansion cohorts with sample sizes far exceeding those in traditional early phase trials of chemotherapeutic agents. The enlarged sample sizes raise ethical concerns for patients who enroll in clinical trials, and emphasize the need for rigorous statistical designs to ensure that trials can stop early for futility while maintaining traditional control of type I error and power. The R package **ppseq** provides a framework for designing early phase clinical trials of binary endpoints using sequential futility monitoring based on Bayesian predictive probability. Trial designs can be compared using interactive plots and selected based on measures of efficiency or accuracy.

Introduction

Statistical methods for early phase oncology trials were developed in the context of cytotoxic treatments. Most cytotoxic treatments exhibit increases in both efficacy and toxicity with increasing dose. As a result, phase I trials were centered on identifying the maximum tolerated dose (MTD), defined as the highest dose that did not exceed a pre-specified toxicity threshold. Phase I dose-escalation trials were designed using the rule-based 3+3 method or the model-based continual reassessment method, among others. But advances in drug discovery have produced numerous biomarker-guided non-cytotoxic therapies such as small molecule inhibitors, antibody drug conjugates, immune checkpoint inhibitors, and monoclonal antibodies. These therapies typically do not exhibit the same pattern of increases in both efficacy and toxicity with increasing dose, so the MTD as traditionally defined may not exist. Instead, lower doses may have equivalent efficacy but lower toxicity as compared to higher doses. As a result, these therapies can be difficult to study with traditional dose-escalation designs (Pestana et al., 2020). To address this issue, recent phase I trials have included large dose-expansion cohorts, in which additional patients are enrolled in phase 1 after the dose-escalation phase is complete. In this setup, the dose-escalation phase is considered phase 1a and used to assess the initial safety of multiple doses, then the dose-expansion phase is considered phase 1b and can have a variety of aims including to further refine the safety of one or more doses, to assess preliminary efficacy, to explore the treatment in various disease-specific subtypes, or to further characterize the pharmacokinetics and/or pharmacodynamics. The use of dose-expansion cohorts increased from 12% in 2006 to 38% in 2011 (Manji et al., 2013) and trials with dose-expansion cohorts led to higher response rates and more frequent success in phase 2 trials (Bugano et al., 2017).

But despite these successes, recent dose-expansion cohorts have not always been planned in advance, leading to uncertain statistical properties, and have at times included samples sizes that far exceed those typically seen in early phase trials of cytotoxic treatments. For example, the KEYNOTE-001 trial of pembrolizumab, initially designed as a 3+3 dose-escalation trial, included multiple protocol amendments and ultimately enrolled a total of 655 patients across five melanoma expansion cohorts and 550 patients across four non-small-cell lung cancer expansion cohorts (Khoja et al., 2015). In a basket trial of atezolizumab, an anti-PD-L1 treatment in patients with a variety of cancers and both with and without PD-L1 expression, an expansion cohort in metastatic urothelial bladder cancer ultimately enrolled 97 patients and evaluated 95, despite the fact that no expansion cohort in this disease subtype was originally planned in the trial protocol. The expansion cohort in metastatic urothelial carcinoma was rather added later in a protocol amendment in which the sample size was increased from what was initially planned (Petrylak et al., 2018; Powles et al., 2014). These enlarged sample sizes raise ethical concerns for patients who enroll in clinical trials, and emphasize the need for rigorous statistical designs to ensure that trials can stop early for futility while maintaining traditional control of type I error and power.

Bayesian predictive probability has been proposed as an approach for sequential monitoring in early phase oncology trials (Dmitrienko and Wang, 2006; Lee and Liu, 2008; Hobbs et al., 2018; Saville et al., 2014). However, in order to be useful to investigators designing such trials, software must be made available to calibrate the design for the desired statistical properties. To our knowledge no such software currently exists in the R programming language. This paper introduces the **ppseq** package for the R software language (R Core Team, 2020), which provides functions to design early phase clinical trials of binary endpoints using sequential predictive probability monitoring for futility. Interactive plots produced using the **ggplot2** package (Wickham, 2016) and the **plotly** package (Sievert, 2020) compare designs based on different thresholds for decision making. Moreover, we demonstrate criteria for selecting an ideal predictive probability monitoring design using the **ppseq** package. While the **ppseq** package was developed with early phase oncology clinical trials in mind, the methodology is general and can be applied to any application of sequential futility monitoring in clinical trial design.

Predictive probability monitoring

Consider the setting of an expansion cohort with a binary outcome, such as tumor response as measured by the RECIST (Response Evaluation Criteria in Solid Tumors) criteria. Here we will focus on the one-sample setting, in which all patients in the trial are enrolled onto a single arm and given the experimental treatment of interest. Functionality is also available for the two-sample setting, in which patients are enrolled onto two different treatment arms, or a treatment and a control arm, for comparative purposes. Each patient, denoted by i, enrolled in the trial either has a response such that $x_i = 1$ or does not have a response such that $x_i = 0$. Then $X = \sum_{i=1}^{n} x_i$ represents the number of responses out of n currently observed patients up to a maximum of N total patients. Let p represent the true probability of response. Fix p_0 as the threshold for unacceptable response rate and p_1 as the threshold for acceptable response rate. Most dose-expansion studies with an efficacy aim will wish to test the null hypothesis $H_0 : p \leq p_0$ versus the alternative hypothesis $H_1 : p \geq p_1$.

The Bayesian paradigm of statistics is founded on Bayes' theorem, which is a mathematical theory that specifies how to combine the prior distributions that define prior beliefs about parameters, such as the true response rate p, with the observed data, such as the total number of responses X, yielding a posterior distribution. Here the prior distribution of the response rate $\pi(p)$ has a beta distribution $\text{Beta}(a_0, b_0)$ and our data X have a binomial distribution $\text{Bin}(n, p)$. Combining the likelihood function for the observed data $L_x(p) \propto p^x(1-p)^{n-x}$ with the prior, we obtain the posterior distribution of the response rate, which follows the beta distribution $p|x \sim \text{Beta}(a_0 + x, b_0 + n - x)$. A posterior probability threshold θ would be pre-specified during the trial design stage. At the end of the trial if the posterior probability exceeded the pre-specified threshold, i.e. if $\Pr(p > p_0|\text{X}) > \theta$, the trial would be declared a success.

The posterior predictive distribution of the number of future responses X^* in the remaining $n^* = N - n$ future patients follows a beta-binomial distribution $\text{Beta-binomial}(n^*, a_0 + x, b_0 + n - x)$. Then the posterior predictive probability (PPP), is calculated as $PPP = \sum_{x^*=0}^{n^*} \Pr(X^* = x^*|x) \times I(\Pr(p > p_0|X, X^* = x^*) > \theta)$. The posterior predictive probability represents the probability that, at any given interim monitoring point, the treatment will be declared efficacious at the end of the trial when full enrollment is reached, conditional on the currently observed data and the specified priors. We would stop the trial early for futility if the posterior predictive probability dropped below a pre-specified threshold θ^*, i.e. $PPP < \theta^*$. Predictive probability thresholds closer to 0 lead to less frequent stopping for futility whereas thresholds near 1 lead to frequent stopping unless there is almost certain probability of success. Predictive probability provides an intuitive interim monitoring strategy for clinical trials that tells the investigator what the chances are of declaring the treatment efficacious at the end of the trial if we were to continue enrolling to the maximum planned sample size, based on the data observed in the trial to date.

Package overview

The **ppseq** package facilitates the design of clinical trials utilizing sequential predictive probability monitoring for futility. The goal is to establish a set of decision rules at the trial planning phase that would be used for interim monitoring during the course of the trial. The main computational challenge in designing such a trial is joint calibration of the posterior probability and posterior predictive probability thresholds to be used in the trial in order to achieve the desired levels of frequentist type I error and power. The main function for achieving this aim is the `calibrate_thresholds()` function, which will evaluate a grid of posterior thresholds θ and predictive thresholds θ^* provided by the user as vector inputs specified with the arguments `pp_threshold` and `ppp_threshold`, respectively. Other required arguments include the unacceptable response rate p_0 specified by `p_null`, the acceptable response rate p_1 specified by `p_alt`, a vector of sample sizes at which interim analyses are to be performed `n`, and the maximum total sample size `N`. The direction of the alternative hypothesis is specified with the argument `direction` and defaults to `"greater"`, which corresponds to the alternative hypothesis $H_1 : p \geq p_1$. The hyperparameters of the prior beta distribution are specified with the argument `prior` and default to `c(0.5, 0.5)`, which denotes a $\text{Beta}(0.5, 0.5)$ distribution. The number of posterior samples are specified with the argument `S`, which defaults to `5000` samples, and the number of simulated trial datasets is specified with the argument `nsim`, which defaults to `1000`. The additional argument `delta`, which defaults to `NULL` for the one-sample setting, can specify a clinically meaningful difference between groups $\delta = p_1 - p_0$ in the case of a two-sample trial design.

The `calibrate_thresholds()` function conducts the following algorithm, given the default arguments:

1. Generate `nsim` datasets, denoted by j, containing the cumulative number of responses x at each interim sample size n from a binomial distribution under the unacceptable (i.e. null) response rate p_0, specified by `p_null`, and under the acceptable (i.e. alternative) response rate p_1, specified by `p_alt`, for each interim look, denoted by l.
2. For dataset j and posterior threshold θ_k, draw `S` samples, denoted by s, from the posterior distribution $p|x_{ls} \sim \text{Beta}(a_0 + x_l, b_0 + n_l - x_l)$ where x_l is the number of responses at interim look l and n_l is the number of enrolled patients at interim look l. Use each $p|x_{ls}$ as the response probability in a binomial distribution to generate the number of future responses X^*_{ls} in the remaining $n^*_l = N - n_l$ future patients at interim look l.

    a. Then for each $X^*_{ls'}$, generate `S` posterior probabilities, denoted by s', at the end of the trial: $PP^*_{lss'} \sim \text{Beta}(a_0 + X^*_{ls} + x_l, b_0 + n^*_l - (X^*_{ls} + x_l))$ and calculate $PP^*_{ls} = \Pr(p > p_0|X^*_{ls}) = \frac{\sum_1^{S'} PP^*_{lss'} > p_0}{S'}$.
    b. Estimate the predictive probability at posterior threshold k as $PPP_{lk} = \frac{\sum_1^{S} PP^*_{ls} > \theta_k}{S}$.
    c. Stop the trial for dataset j at interim look l and predictive threshold m if $PPP_{lk} < \theta^*_m$. Otherwise continue enrolling.

3. Repeat (2) over all combinations of datasets j, posterior thresholds k, and predictive thresholds m.
4. If dataset j was stopped early for futility then we do not reject the null hypothesis. If dataset j reached full enrollment, we reject the null hypothesis $H_0 : p \leq p_0$ at posterior threshold k if $PPP_{lk} > \theta_k$.

The function returns a list, the first element of which is a `tibble` containing the posterior threshold θ, the predictive threshold θ^*, the mean sample size under the null and the alternative, the proportion of positive trials under the null and alternative, and the proportion of trials stopped early under the null and alternative. The proportion of trials simulated under the null hypothesis for which the null hypothesis was rejected is an estimate of the type I error, and the proportion of trials simulated under the alternative hypothesis for which the null hypothesis was rejected is an estimate of the power. The `print()` option will print the results summary for each combination of thresholds, filtered by an acceptable range of type I error and minimum power, if desired. Note that the results will be sensitive to the choice of `nsim` and `S`. We have set what we believe are reasonable defaults and would caution users against reducing these values without careful consideration.

Design selection

After obtaining results for all combinations of evaluated posterior and predictive thresholds, the next step is to select the ideal design from among the various options. The **ppseq** package introduces two criteria to assist users in making a selection. The first, called the "optimal accuracy" design, identifies the design that minimizes the Euclidean distance to 0 type I error probability and a power of 1. To accomplish this, the accuracy Euclidean distance (AED) for the design with posterior threshold k and predictive threshold m is calculated as $AED_{km} = w_\alpha * (\alpha_{km} - 0)^2 + w_(1 - \beta) * ((1 - \beta)_{km} - 1)^2$, where w_α and $w_{(1-\beta)}$ are optional weights on the type I error and power, respectively, and α_{km} denotes the estimated type I error and $(1 - \beta)_{km}$ denotes the estimated power. The design with the smallest value of AED_{km} is selected as optimal. The second criteria, called the "optimal efficiency" design, identifies the design that minimizes the Euclidean distance to minimal average sample size under the null and maximal average sample size under the alternative. To accomplish this, the efficiency Euclidean distant (EED) for the design with posterior threshold k and predictive threshold m is calculated as $EED_{km} = w_{\bar{N}_{H_0}} * (\bar{N}_{H_0km} - min(\bar{N}_{H_0}))^2 + w_{\bar{N}_{H_1}} * (\bar{N}_{H_1km} - max(\bar{N}_{H_1}))^2$, where $w_{\bar{N}_{H_0}}$ and $w_{\bar{N}_{H_1}}$ are optional weights on the average sample size under the null and alternative, respectively, $\bar{N}_{H_0km}$ and $\bar{N}_{H_1km}$ denote the average sample sizes under the null and alternative, respectively, and $min(\bar{N}_{H_0})$ and $max(\bar{N}_{H_1})$ denote the minimum average sample size under the null and the maximum average sample size under the alternative alternative, respectively, across all combinations of k and m. The design with the smallest value of EED_{km} is selected as optimal. The `optimize_design()` function returns a list that contains the details of each of the two optimal designs.

Decision rules

To ease the implementation of clinical trials designed with sequential predictive probability monitoring, once a design has been selected, a table of decision rules can be produced using the `calc_decision_rules()` function. The function takes the sample sizes `n` at which interim analyses are to be performed as well as the maximum total sample size `N`, the null value to compare to in the one-sample case `p0` (set to `NULL` in the two-sample case), the posterior threshold of the selected design `theta`, and the predictive threshold of the selected design `ppp`. Arguments `direction`, `prior`, `S`, `delta` are as described in the Package Overview section, with the same defaults. The function results in a `tibble`. The trial would stop at a given look if the number of observed responses is less than or equal to r, otherwise the trial would continue enrolling if the number of observed responses is greater than r. At the end of the trial when the maximum planned sample size is reached, the treatment would be considered promising if the number of observed responses is greater than r. In the one-sample case, the resulting `tibble` includes a column for the sample size `n` at each interim look, r at each look, and a column for the associated posterior predictive probability `ppp`. In the two-sample case, the `tibble` includes columns for `n0` and `n1`, the sample size at each interim analysis in the control and experimental arms, respectively. There are also columns for `r0` and `r1`, the number of responses in the control arm and experimental arm, respectively, leading to the decision to stop or continue. Finally, there is a column for the posterior predictive probability associated with that decision `ppp`.

Visualizations

Finally, to assist users in comparing the results of the various design options, a `plot()` option is available for the results of `calibrate_thresholds` that allows creation of static plots using the **ggplot2** package (Wickham, 2016) or interactive plots using the **plotly** package (Sievert, 2020). Two plots are produced, one plotting type I error by power and indicating the optimal accuracy design, and one plotting the average sample size under the null by the average sample size under the alternative and indicating the optimal efficiency design. The motivation for including an interactive graphics option was the utility of the additional information available when hovering over each point. Instead of simply eyeballing where points fall along the axes, users can see the specific type I error, power, average sample size under the null, average sample size under the alternative, the posterior and predictive thresholds associated with the design, as well as the distance to the upper left point on the plot. A `plot()` option is also available for the results of `calc_decision_rules`. In the one-sample case it produces a single plot showing the sample size at each interim analysis on the x-axis and the possible number of responses at each interim analysis on the y-axis. In the two-sample case a grid of plots is produced, with one plot for each interim analysis. The x-axis shows the number of possible responses in the control group and the y-axis shows the number of possible responses in the experimental group. In both cases, the boxes are colored green for a "proceed" decision and red for a "stop" decision for each combination and the hover box produced by **plotly** provides the details.

Toy example

In this section I will present a toy example to demonstrate the functionality, and the following section will present a more in-depth case study that is more computationally intensive.

First we install and load the **ppseq** package.

```
install.packages("ppseq")
```

```
library(ppseq)
```

Consider the case where we are interested in designing a trial to investigate how a new treatment impacts tumor response measured as a binary outcome of response versus no response. We know the current standard of care treatment results in a tumor response rate of 10%, and we wish to improve this by 30%. So we wish to test $H_0 : p \leq 0.1$ versus $H_1 : p \geq 0.4$, so we set `p_null = 0.1` and `p_alt = 0.4`. This is a rare disease so our maximum sample size is 15, so we set `N = 15`, and we will do interim analyses after every 5 patients, so we set `n = seq(5, 15, 5)`. We wish to examine designs based on combinations of posterior thresholds $\theta = 0.85, 0.90$ and predictive thresholds $\theta^* = 0.1, 0.2$, so we set `pp_threshold = c(0.85, 0.9)` and `ppp_threshold = c(0.1, 0.2)`. Finally, for computational speed in this toy example, we set `S=50` and `nsim=50`, but in practice we would want to use much larger values, in the thousands.

```
set.seed(123)
```

```
cal_thresh <-
  calibrate_thresholds(
    p_null = 0.1,
    p_alt = 0.4,
    n = seq(5, 15, 5),
    N = 15,
    pp_threshold = c(0.85, 0.9),
    ppp_threshold = c(0.1, 0.2),
    S = 50,
    nsim = 50
    )
```

Since there are only four design options in this toy example, we print the entire results table using a call to `print()`:

```
print(cal_thresh)
```

```
#> # A tibble: 4 x 8
#>   pp_threshold ppp_threshold mean_n1_null prop_pos_null
#>          <dbl>         <dbl>        <dbl>         <dbl>
#> 1         0.85           0.1         12.1          0.1
#> 2         0.85           0.2          9.5          0.06
#> 3         0.9            0.1         10.8          0.04
#> 4         0.9            0.2          8.9          0.04
#>   prop_stopped_null mean_n1_alt prop_pos_alt prop_stopped_alt
#>               <dbl>       <dbl>        <dbl>            <dbl>
#> 1              0.42        14.8         0.92             0.02
#> 2              0.64        14.8         0.92             0.02
#> 3              0.5         14.8         0.9              0.02
#> 4              0.74        14.7         0.88             0.04
```

We use `optimize_design()` to identify the optimal accuracy and optimal efficiency designs, subject to type I error between 0.025 and 0.1, specified by `type1_range = c(0.025, 0.1)`, and power of at least 0.75, specified by `minimum_power = 0.75`:

```
optimize_design(cal_thresh, type1_range = c(0.025, 0.1), minimum_power = 0.75)
```

```
#> $`Optimal accuracy design:`
#> # A tibble: 1 x 6
#>   pp_threshold ppp_threshold `Type I error` Power
#>          <dbl>         <dbl>          <dbl> <dbl>
#> 1         0.85           0.2           0.06  0.92
#>   `Average N under the null` `Average N under the alternative`
#>                        <dbl>                             <dbl>
#> 1                        9.5                              14.8
#>
#> $`Optimal efficiency design:`
#> # A tibble: 1 x 6
#>   pp_threshold ppp_threshold `Type I error` Power
#>          <dbl>         <dbl>          <dbl> <dbl>
#> 1          0.9           0.2           0.04  0.88
#>   `Average N under the null` `Average N under the alternative`
#>                        <dbl>                             <dbl>
#> 1                        8.9                              14.7
```

We can compare all of the design options graphically with a call to `plot()` and with `plotly = TRUE` to obtain interactive plots or `plotly = FALSE` to obtain static plots.

```
plot(
  cal_thresh,
  type1_range = c(0.025, 0.1),
  minimum_power = 0.75,
  plotly = FALSE
)
```

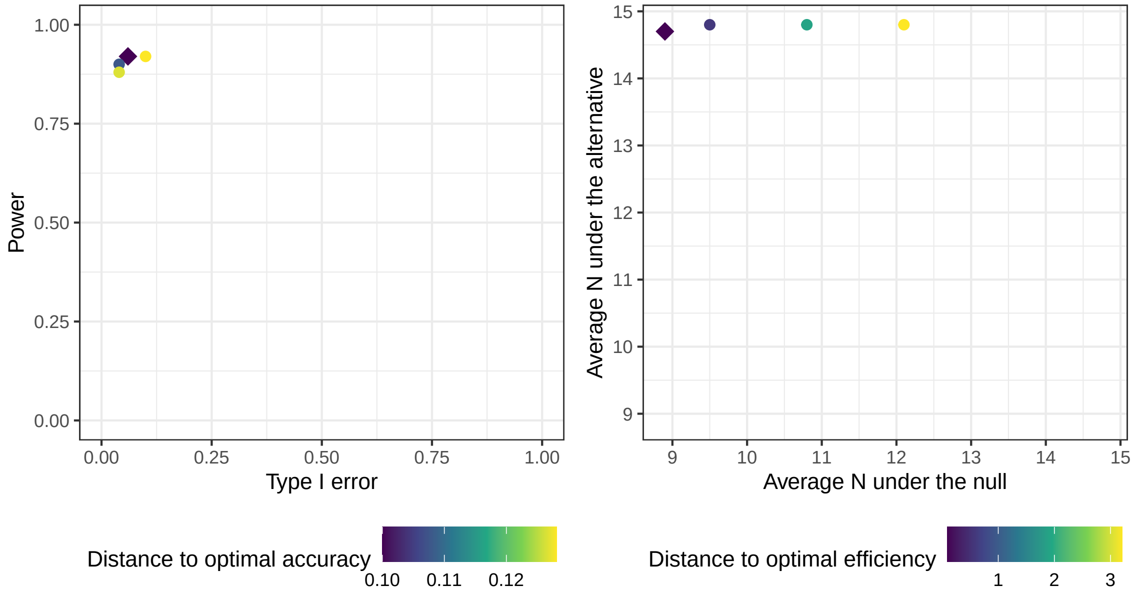

Figure 1: Plot of design options made with ggplot2. The accuracy designs are in the plot on the left and the efficiency designs are on the right. The color represents the Euclidean distance to the top left point and the optimal design is indicated by a diamond.
(#fig:figstatic)

Case study

To demonstrate the functionality of the **ppseq** package, we focus on a re-design of a dose-expansion cohort for the study of atezolizumab in metastatic urothelial carcinoma patients (mUC) using sequential predictive probability monitoring. Atezolizumab is an anti-PD-L1 treatment that was originally tested in the phase 1 setting in a basket trial across a variety of cancer sites harboring PD-L1 mutations. The atezolizumab expansion study in mUC had the primary aim of further evaluating safety, pharmacodynamics and pharmacokinetics and therefore was not designed to meet any specific criteria for type I error or power. An expansion cohort in mUC was not part of the original protocol design, but was rather added later through a protocol amendment. The expansion cohort in mUC ultimately evaluated a total of 95 participants (Powles et al., 2014). Other expansion cohorts that were included in the original protocol, including in renal-cell carcinoma, non-small-cell lung cancer, and melanoma, were planned to have a sample size of 40. These pre-planned expansion cohorts were designed with a single interim analysis for futility that would stop the trial if 0 responses were seen in the first 14 patients enrolled. According to the trial protocol, this futility rule is associated with at most a 4.4% chance of observing no responses in 14 patients if the true response rate is 20% or higher. The protocol also states the widths of the 90% confidence intervals for a sample size of 40 if the observed response rate is 30%. There was no stated decision rule for efficacy since efficacy was not an explicit aim of the expansion cohorts. In the re-design we assume a null, or unacceptable, response rate of 0.1 and an alternative, or acceptable, response rate of 0.2. We plan a study with up to a total of 95 participants. In our sequential predictive probability design we will check for futility after every 5 patients are enrolled. We consider posterior thresholds of 0, 0.7, 0.74, 0.78, 0.82, 0.86, 0.9, 0.92, 0.93, 0.94, 0.95, 0.96, 0.97, 0.98, 0.99, 0.999, 0.9999, 0.99999, and 1, and predictive thresholds of 0.05, 0.1, 0.15, and 0.2.

We use the `calibrate_thresholds()` function to obtain the operating characteristics of designs based on on each combination of posterior and predictive thresholds. Because of the inherent computation intensity in these calculations, this function relies on the **future** (Bengtsson, 2020) and **furrr** (Vaughan and Dancho, 2021) packages to parallelize computations. The user will be responsible for setting up a call to `future::plan()` that is appropriate to their operating environment and simulation setting. The example in this case study was run on a Unix server with 192 cores, and we wished to use 76 cores to accommodate the 76 distinct designs that result from the 19 posterior by 4 predictive threshold grid. Because the code takes some time to run, the results of the below example code are available as a dataset called `one_sample_cal_tbl` included in the **ppseq** package.

```
library(future)
```

```
set.seed(123)

plan(multicore(workers = 76))

one_sample_cal_tbl <-
  calibrate_thresholds(
    p_null = 0.1,
    p_alt = 0.2,
    n = seq(5, 95, 5),
    N = 95,
    pp_threshold = c(0, 0.7, 0.74, 0.78, 0.82, 0.86, 0.9, 0.92, 0.93, 0.94,
                     0.95, 0.96, 0.97, 0.98, 0.99, 0.999, 0.9999, 0.99999, 1),
    ppp_threshold = seq(0.05, 0.2, 0.05),
    direction = "greater",
    delta = NULL,
    prior = c(0.5, 0.5),
    S = 5000,
    nsim = 1000
  )
```

Next we print the results table using the `print()` option, and limited to designs with type I error between 0.05 and 0.1, and a minimum power of 0.7. We find that 35 of the 76 designs meet these criteria for type I error and power.

```
print(
  one_sample_cal_tbl,
  type1_range = c(0.05, 0.1),
  minimum_power = 0.7
)

#> # A tibble: 8 x 8
#>   pp_threshold ppp_threshold mean_n1_null prop_pos_null
#>          <dbl>         <dbl>        <dbl>         <dbl>
#> 1         0.82          0.2          35.9         0.096
#> 2         0.86          0.2          36.0         0.097
#> 3         0.9           0.05         50.7         0.082
#> 4         0.9           0.1          38.8         0.073
#> 5         0.9           0.15         35.5         0.065
#> 6         0.92          0.05         50.7         0.081
#> 7         0.92          0.1          38.8         0.073
#> 8         0.92          0.15         35.6         0.066
#>   prop_stopped_null mean_n1_alt prop_pos_alt prop_stopped_alt
#>               <dbl>       <dbl>        <dbl>            <dbl>
#> 1             0.891        79.4        0.781            0.214
#> 2             0.89         79.4        0.782            0.213
#> 3             0.865        89.8        0.872            0.097
#> 4             0.884        81.7        0.791            0.185
#> 5             0.919        79.4        0.76             0.232
#> 6             0.866        89.9        0.874            0.095
#> 7             0.884        81.8        0.793            0.183
#> 8             0.917        79.7        0.765            0.227
```

We use the `optimize_design()` function to obtain the details of the optimal accuracy and optimal efficiency designs, limited to our desired range of type I error and minimum power. We find that the optimal accuracy design is the one with posterior threshold 0.9 and predictive threshold 0.05. It has a type I error of 0.072, power of 0.883, average sample size under the null of 51, and average sample size under the alternative of 91. The optimal efficiency design is the one with posterior threshold of 0.92 and predictive threshold of 0.1. It has a type I error of 0.06, power of 0.796, average sample size under the null of 39, and average sample size under the alternative of 82. For comparison, the original design of the atezolizumab expansion cohort in mUC, with a single look for futility after the first 14 patients, has a type I error of 0.005, power of 0.528, average sample size under the null of 76, and average sample size under the alternative of 92.

```
optimize_design(
  one_sample_cal_tbl,
```

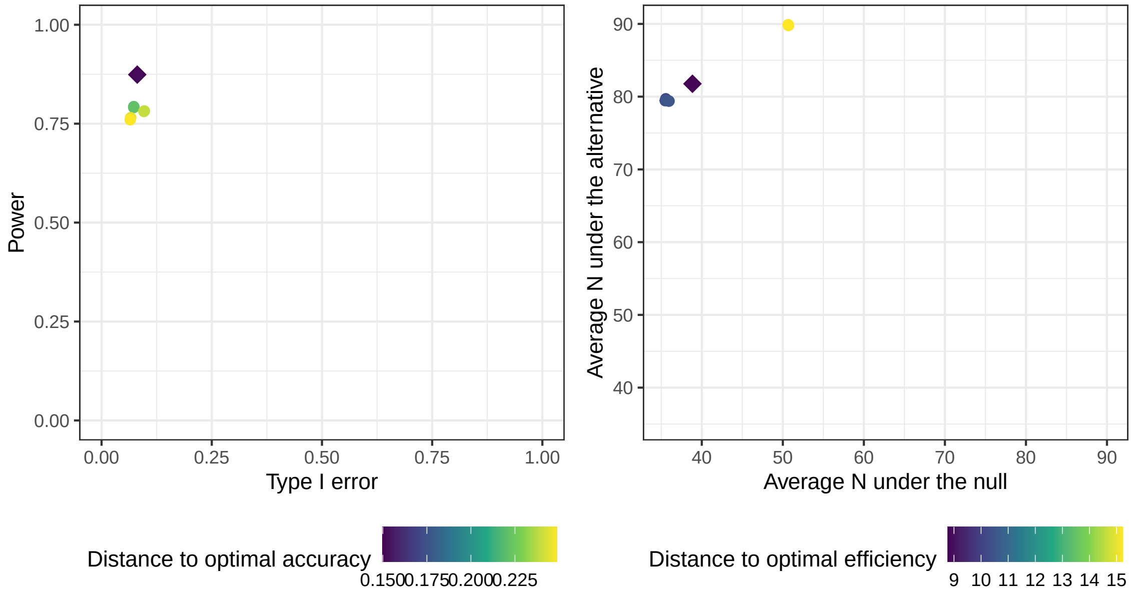

Figure 2: Plot of design options made with ggplot2. The accuracy designs are in the plot on the left and the efficiency designs are on the right. The color represents the Euclidean distance to the top left point and the optimal design is indicated by a diamond.
(#fig:unnamed-chunk-12)

```
  type1_range = c(0.05, 0.1),
  minimum_power = 0.7
)

#> $`Optimal accuracy design:`
#> # A tibble: 1 x 6
#>   pp_threshold ppp_threshold `Type I error` Power
#>          <dbl>         <dbl>          <dbl> <dbl>
#> 1         0.92          0.05          0.081 0.874
#>   `Average N under the null` `Average N under the alternative`
#>                        <dbl>                             <dbl>
#> 1                       50.7                              89.9
#>
#> $`Optimal efficiency design:`
#> # A tibble: 1 x 6
#>   pp_threshold ppp_threshold `Type I error` Power
#>          <dbl>         <dbl>          <dbl> <dbl>
#> 1         0.92           0.1          0.073 0.793
#>   `Average N under the null` `Average N under the alternative`
#>                        <dbl>                             <dbl>
#> 1                       38.8                              81.8
```

To compare these optimal designs with all other designs, we can use the `plot()` function with the `plotly = TRUE` option to obtain interactive visualizations, or the `plotly = FALSE` option to obtain static visualizations.

```
plot(
  one_sample_cal_tbl,
  type1_range = c(0.05, 0.1),
  minimum_power = 0.7,
  plotly = TRUE
)
```

In this case we may choose to use the optimal efficiency design, which has the desirable trait of a very small average sample size under the null of just 39 patients, while still maintaining reasonable type I error of 0.06 and power of 0.796. This design would allow us to stop early if the treatment were inefficacious, thus preserving valuable financial resources for use in studying more promising treatments and preventing our human subjects from continuing an ineffective treatment. Finally, we generate the decision table associated with the selected design for use in making decision at each interim analysis during the conduct of the trial. Because of the computational time involved, the results of the below example code are available as a dataset called `one_sample_decision_tbl` included in the **ppseq** package. In the results table, we see that at the first interim futility look after just 5 patients, we would not stop the trial. After the first 10 patients we would stop the trial if there were 0 responses, and so on. At the end of the trial when all 95 patients have accrued, we would declare the treatment promising of further study if there were greater than or equal to 14 responses.

```
set.seed(123)

one_sample_decision_tbl <-
  calc_decision_rules(
    n = seq(5, 95, 5),
    N = 95,
    theta = 0.92,
    ppp = 0.1,
    p0 = 0.1,
    direction = "greater",
    delta = NULL,
    prior = c(0.5, 0.5),
    S = 5000
  )

one_sample_decision_tbl

#> # A tibble: 19 x 3
#>         n     r     ppp
#>     <dbl> <int>   <dbl>
#>  1      5    NA  NA
#>  2     10     0  0.0634
#>  3     15     0  0.022
#>  4     20     1  0.0844
#>  5     25     1  0.0326
#>  6     30     2  0.0702
#>  7     35     2  0.0288
#>  8     40     3  0.0536
#>  9     45     4  0.0708
#> 10     50     4  0.0344
#> 11     55     5  0.0478
#> 12     60     6  0.0622
#> 13     65     7  0.0888
#> 14     70     8  0.095
#> 15     75     8  0.033
#> 16     80     9  0.0326
#> 17     85    10  0.0346
#> 18     90    11  0.0162
#> 19     95    13  0

plot(one_sample_decision_tbl, plotly = FALSE)
```

Summary

With the focus of early stage clinical trial research in oncology shifting away from the study of cytotoxic treatments and toward immunotherapies and other non-cytotoxic treatments, new approaches to clinical trial design are needed that move beyond the traditional search for the maximum tolerated dose (Hobbs et al., 2019). Bayesian sequential predictive probability monitoring provides a natural and flexible way to expand the number of patients studied in phase 1 or to design phase 2 trials that allow for efficient early stopping for futility while maintaining control of type I error and power. The **ppseq** package implements functionality to evaluate a range of posterior and predictive thresholds for a given study design and identify the optimal design based on accuracy (i.e. type I error and power) or efficiency (i.e. average sample sizes under the null and alternative). Interactive visualization options are provided to ease comparison of the resulting design options. Once an ideal design is selected, a table of decision rules can be obtained to make trial conduct simple and straightforward.

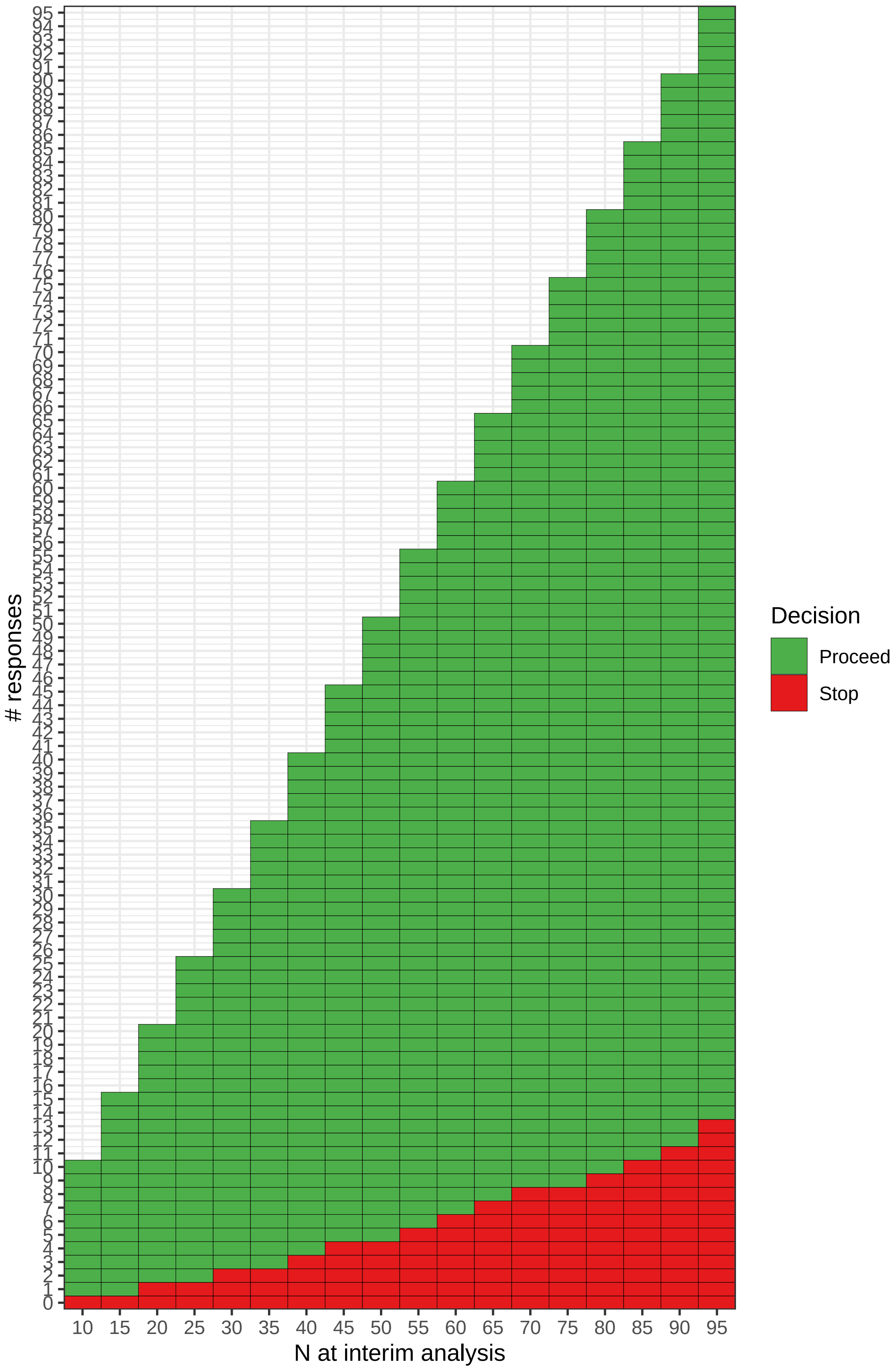

Figure 3: Plot of decision rules made with ggplot. The color indicates whether the trial should stop or proceed for a given number of responses at each interim analysis.
(#fig:unnamed-chunk-17)

Bibliography

H. Bengtsson. A unifying framework for parallel and distributed processing in r using futures, aug 2020. URL https://arxiv.org/abs/2008.00553. [p6]

D. D. G. Bugano, K. Hess, D. L. F. Jardim, A. Zer, F. Meric-Bernstam, L. L. Siu, A. R. A. Razak, and D. S. Hong. Use of expansion cohorts in phase i trials and probability of success in phase ii for 381 anticancer drugs. *Clin Cancer Res*, 23(15):4020–4026, 2017. ISSN 1078-0432 (Print) 1078-0432. doi: 10.1158/1078-0432.Ccr-16-2354. [p1]

A. Dmitrienko and M. D. Wang. Bayesian predictive approach to interim monitoring in clinical trials. *Stat Med*, 25(13):2178–95, 2006. ISSN 0277-6715 (Print) 0277-6715. doi: 10.1002/sim.2204. [p1]

B. P. Hobbs, N. Chen, and J. J. Lee. Controlled multi-arm platform design using predictive probability. *Stat Methods Med Res*, 27(1):65–78, 01 2018. [p1]

B. P. Hobbs, P. C. Barata, Y. Kanjanapan, C. J. Paller, J. Perlmutter, G. R. Pond, T. M. Prowell, E. H. Rubin, L. K. Seymour, N. A. Wages, T. A. Yap, D. Feltquate, E. Garrett-Mayer, W. Grossman, D. S. Hong, S. P. Ivy, L. L. Siu, S. A. Reeves, and G. L. Rosner. Seamless Designs: Current Practice and Considerations for Early-Phase Drug Development in Oncology. *J Natl Cancer Inst*, 111(2):118–128, 02 2019. [p9]

L. Khoja, M. O. Butler, S. P. Kang, S. Ebbinghaus, and A. M. Joshua. Pembrolizumab. *J Immunother Cancer*, 3:36, 2015. [p1]

J. J. Lee and D. D. Liu. A predictive probability design for phase ii cancer clinical trials. *Clin Trials*, 5(2): 93–106, 2008. ISSN 1740-7745 (Print) 1740-7745. doi: 10.1177/1740774508089279. [p1]

A. Manji, I. Brana, E. Amir, G. Tomlinson, I. F. Tannock, P. L. Bedard, A. Oza, L. L. Siu, and A. R. Razak. Evolution of clinical trial design in early drug development: systematic review of expansion cohort use in single-agent phase i cancer trials. *J Clin Oncol*, 31(33):4260–7, 2013. ISSN 0732-183x. doi: 10.1200/jco.2012.47.4957. [p1]

R. C. Pestana, S. Sen, B. P. Hobbs, and D. S. Hong. Histology-agnostic drug development - considering issues beyond the tissue. *Nat Rev Clin Oncol*, 17(9):555–568, 09 2020. [p1]

D. P. Petrylak, T. Powles, J. Bellmunt, F. Braiteh, Y. Loriot, R. Morales-Barrera, H. A. Burris, J. W. Kim, B. Ding, C. Kaiser, M. Fassò, C. O'Hear, and N. J. Vogelzang. Atezolizumab (mpdl3280a) monotherapy for patients with metastatic urothelial cancer: Long-term outcomes from a phase 1 study. *JAMA Oncol*, 4(4):537–544, 2018. ISSN 2374-2437 (Print) 2374-2437. doi: 10.1001/jamaoncol.2017.5440. [p1]

T. Powles, J. P. Eder, G. D. Fine, F. S. Braiteh, Y. Loriot, C. Cruz, J. Bellmunt, H. A. Burris, D. P. Petrylak, S. L. Teng, X. Shen, Z. Boyd, P. S. Hegde, D. S. Chen, and N. J. Vogelzang. Mpdl3280a (anti-pd-l1) treatment leads to clinical activity in metastatic bladder cancer. *Nature*, 515(7528):558–62, 2014. ISSN 0028-0836. doi: 10.1038/nature13904. [p1, 6]

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2020. URL http://www.R-project.org/. ISBN 3-900051-07-0. [p1]

B. R. Saville, J. T. Connor, G. D. Ayers, and J. Alvarez. The utility of bayesian predictive probabilities for interim monitoring of clinical trials. *Clin Trials*, 11(4):485–493, 2014. ISSN 1740-7745 (Print) 1740-7745. doi: 10.1177/1740774514531352. [p1]

C. Sievert. *Interactive Web-Based Data Visualization with R, plotly, and shiny*. Chapman and Hall/CRC, 2020. ISBN 9781138331457. URL https://plotly-r.com. [p2, 4]

D. Vaughan and M. Dancho. *furrr: Apply Mapping Functions in Parallel using Futures*, 2021. URL https://CRAN.R-project.org/package=furrr. R package version 0.2.2. [p6]

H. Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN 978-3-319-24277-4. URL https://ggplot2.tidyverse.org. [p2, 4]

Emily C. Zabor
Department of Quantitative Health Sciences & Taussig Cancer Institute, Cleveland Clinic
9500 Euclid Ave. CA-60
Cleveland, OH 44195 USA
http://www.emilyzabor.com/
ORCiD: 0000-0002-1402-4498
zabore2@ccf.org

Brian P. Hobbs
Dell Medical School, The University of Texas at Austin
true
Austin, TX 78712
ORCiD: 0000-0003-2189-5846
brian.hobbs@austin.utexas.edu

Michael J. Kane
Department of Biostatistics, Yale University
60 College Street
New Haven, CT 06511
ORCiD: 0000-0003-1899-6662
michael.kane@yale.edu