Journal of Statistical Software

November 2017, Volume 81, Issue 8. *doi: 10.18637/jss.v081.i08*

OneArmPhaseTwoStudy: An R Package for Planning, Conducting, and Analysing Single-Arm Phase II Studies

Meinhard Kieser
University of Heidelberg

Marius Wirths
University of Heidelberg

Stefan Englert
University of Heidelberg

Cornelia Ursula Kunz
University of Warwick

Geraldine Rauch
University of Heidelberg

Abstract

In clinical phase II studies, the efficacy of a promising therapy is tested in patients for the first time. Based on the results, it is decided whether the development programme should be stopped or whether the benefit-risk profile is promising enough to justify the initiation of large phase III studies. In oncology, phase II trials are commonly conducted as single-arm trials with planned interim analyses to allow for an early stopping for futility. The specification of an adequate study design that guarantees control of the type I and II error rates is a key task in the planning stage of such a trial. A variety of statistical methods exists which can be used to optimise the planning and analysis of such studies. However, there are currently neither commercial nor non-commercial software tools available that support the practical application of these methods comprehensively. The R package **OneArmPhaseTwoStudy** was implemented to fill this gap. The package allows determining an adequate study design for the particular situation at hand as well as monitoring the progress of the study and evaluating the results with valid and efficient analyses methods. This article describes the features of the R package and its application.

Keywords: clinical trial, single-arm study, two-stage design, binary data, exact method, adaptive design.

1. Introduction

In phase II clinical trials, the activity of a new therapy is evaluated to decide whether it warrants further investigation in large-scale phase III trials. In oncology, these trials are

frequently performed in a single-arm design (Gan, Grothey, Pond, Moore, Siu, and Sargent 2010; Baghdadi and Laffler 2013). The primary endpoint is commonly a binary outcome measuring therapy response based on tumor shrinkage (Eisenhauer *et al.* 2009). For ethical and economical reasons, these trials are usually performed with interim analyses to allow for an early termination in case of a low observed response rate ("stop for futility"). Due to logistic restrictions and limited gain in statistical efficiency when increasing the number of interim analyses (Chen 1997), two-stage designs are most commonly applied. Early termination due to overwhelming activity ("stopping for efficacy") plays no major role for these phase II studies as there is no ethical imperative for early stopping in this situation. Furthermore, the collection of sufficient information on efficacy and safety in phase II is important before initiating a large phase III program.

We consider two-stage designs with the option of early stopping for futility where the null hypothesis $H_0 : \pi \leq \pi_0$ is tested at one-sided level α and where the power $1 - \beta$ is evaluated at a response rate $\pi_1 > \pi_0$. By searching algorithms based on the exact binomial distribution, designs fulfilling the type I and type II error restrictions can be identified. These designs are characterized by the sample size for the first and second stage, n_1 and $n - n_1$, respectively, and by the boundary values r_1 and r ($r_1 < r$). The study is continued after the first stage if the number of responses at the interim analysis is greater than r_1, and the null hypothesis can be rejected after stage 2 if the total number of responses exceeds r. Usually, several solutions (n_1, r_1, n, r) exist and additional criteria are required for selecting a specific design. Simon (1989) proposed the "minimax design" minimizing the maximum sample size n and the "optimal design" minimizing the expected sample size under the null hypothesis among those two-stage designs satisfying the constraints. Other criteria are available such as "admissible designs" (Jung, Lee, Kim, and George 2004) that are compromises between the minimax and the optimal design. By the choice of an adequate design, control of the type I and type II error rate is assured. A further important demand from a biostatistical viewpoint is the appropriate estimation of the treatment effect in the analyses. Treatment effect estimates obtained from phase II studies are used to compare the activity of different therapies under investigation and are the basis of planning the proceeding phase III studies. However, the common maximum likelihood estimator (MLE) is typically biased in multi-stage designs that allow for early stopping (Kunz and Kieser 2012). Similarly, calculation of the "naive" confidence interval without taking the sequential nature of the trial into account does usually not guarantee the desired coverage probability. Therefore, proper methods are required that are tailored to the design applied.

The above described two-stage designs foresee early stopping only after the first stage. However, it may become evident during the course of the trial that, based on the currently observed results, it is very unlikely or even impossible to reject the null hypothesis after the second stage. This disadvantage of standard two-stage designs can be resolved by implementing a statistical monitoring of the results and a curtailment procedure. The idea is to stop the trial if the probability to reject the null hypothesis, given the observed number of responses, falls below a pre-specified threshold ("stochastic curtailment") or is zero ("non-stochastic curtailment"). Curtailment can be restricted to the second stage only (Ayanlowo and Redden 2007) or can be performed in both stages (Kunz and Kieser 2012). Whether or not to implement a curtailment procedure depends on the balance between the reduction of sample size and the loss in power to be expected in the specific situation at hand. Again, appropriate methods and related software are needed.

The "classical" two-stage designs described above require conduct of the study exactly as pre-defined by specification of (n_1, r_1, n, r). Changing the design mid-course includes the risk of compromising the type I error rate (Englert and Kieser 2012a). However, if, for example, an unexpected high response rate is observed at the interim analysis, it may be desirable to reduce the sample size for the second stage. Recently, flexible single-arm two-stage designs have been developed that allow (data-driven) modifications of the initially specified design while still controlling the type I error rate (Englert and Kieser 2012a,b). Furthermore, it turns out that using these designs may even lead to an increased statistical efficiency (Englert and Kieser 2012b). Application of these methods is therefore highly attractive but requires related software. The same holds true for the so-called subset designs that allow a simultaneous assessment of two nested endpoints.

Currently available software for single-arm phase II studies are restricted to "classical " two-stage-designs and focus on specific aspects. For example, there are a number of non-commercial (e.g., Kirk and Fay 2014; Seshan 2015; Southwest Oncology Group 2015) and commercial software packages (e.g., Cytel 2015, NCSS 2015, SAS Institute Inc. 2015 and StataCorp 2015) providing the feature of determining Simon's optimal and minimax design. Admissible designs are implemented in two other programs (Jung *et al.* 2004; Kunz and Kieser 2011b). In addition, there are a number of web-based software tools. These are not explicitly referenced here, because a quality assessment of the source code is not directly possible. Until now, there exists no software dealing with flexible phase II designs. Furthermore, no software package is currently available that allows a comprehensive support of all aspects when performing single-arm phase II studies, namely planning, conduct and analysis. The R (R Core Team 2017) package **OneArmPhaseTwoStudy** (Wirths 2017) that is described below fills this gap. The package is available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/package=OneArmPhaseTwoStudy`. As a helpful supplement, we recommend the validated, web-based software tool implemented by Englert (`https://imbi.shinyapps.io/phaseII-app/`). This tool is based on the work of Englert and Kieser (2015) which allows for a proper dealing with over- and underrunning.

The paper is organized as follows. In Section 2, we outline the statistical methods for single-arm two-stage designs. Implementation of these methods in the **OneArmPhaseTwoStudy** package as well as its features are described in Section 3. In Section 4, its application is demonstrated by an example, and we provide a brief discussion in Section 5.

2. Methods

2.1. Classical single-arm two-stage designs

Identification of designs

For a two-stage design defined by (n_1, r_1, n, r) the probability of rejecting the null hypothesis in case of a true response rate π^* is given by

$$1 - \left(B(n_1, r_1, \pi^*) + \sum_{x=r_1+1}^{\min(n_1, r_1)} b(n_1, x, \pi^*) B(n - n_1, r - x, \pi^*) \right), \tag{1}$$

where $B(\cdot)$ denotes the cumulative binomial distribution and $b(\cdot)$ the binomial probability mass function (see e.g., Simon 1989). Evaluating (1) at $\pi^* = \pi_0$ or $\pi^* = \pi_1$ provides the type I error rate and power, respectively. The probability of early stopping (PET) and the expected sample size (EN) under the null hypothesis are given by

$$PET(\pi_0) = B(n_1, r_1, \pi_0), \tag{2}$$

$$EN(\pi_0) = n_1 + (1 - PET(\pi_0))(n - n_1). \tag{3}$$

Under all designs fulfilling the type I and type II error constraints, the optimal design is defined as the one with the smallest $EN(\pi_0)$ and the minimax design is the one with smallest total sample size n. If more than one design exists with smallest n, the one with smallest $EN(\pi_0)$ is selected as minimax design (Simon 1989). Admissible designs minimize the Bayes risk $qn + (1-q)EN(\pi_0)$ for a given weight $q \in [0, 1]$ (Jung *et al.* 2004). In general, a design is admissible not only for a single value but for a range of values for q. Admissible designs show a higher total sample size than the minimax design but a smaller total sample size than the optimal design and vice versa with respect to $EN(\pi_0)$.

The algorithm to determine two-stage designs fulfilling the requirements with respect to the type I and II error and among the optimal, minimal, and admissible designs follows the description in Kunz and Kieser (2011b). A crude algorithm searches for each value of n over $n_1 \in [1, n-1]$, $r_1 \in [0, n_1 - 1]$ and $r \in [r_1 + 1, n - 1]$ for those designs for which expression (1) is less than or equal to α at $\pi^* = \pi_0$ and at least $1 - \beta$ at $\pi^* = \pi_1$. This approach can be improved as follows. The starting values of the searching procedure are determined based on the following considerations. As $(1 - \pi)n_1 \leq B(n_1, r_1, \pi_1)$, the constraint with respect to the type II error rate leads to $n_1 > \log(\beta)/\log(1 - \pi_1)$. Hence, for $\beta \neq 1 - \pi_1$ the starting value for n_1 is $\text{ceil}(\log(\beta)/\log(1 - \pi_1))$ while it is 2 for $\beta = 1 - \pi_1$. Here, $\text{ceil}(x)$ denotes the function which returns the smallest integer value that is not less than x. As n has to be larger than n_1, the starting value for n is $n_1 + 1$. For every pair (r_1, n_1) with $r_1 \in [0, n_1 - 1]$ it is checked whether $B(n_1, r_1, \pi_1) \geq 1 - \beta$, as it can be shown that this inequality has to be true in order to find a parameter set (n_1, r_1, n, r) which fulfills the type II error condition. If this is not the case, n_1 is increased by 1 and the search continues; if the inequality holds true, the algorithm searches over r in the range of $[r_1 + 1, n - n_1 + r_1]$. For every value of r, $B(n, r, \pi_1) < \beta$ has to hold true for any solution. Therefore, this condition is checked and the search over r is stopped and continued with the next r_1 whenever the inequality does not hold true. Otherwise, the type I error rate and power are calculated via (1) and the algorithm continues.

To improve the search algorithm, the **OneArmPhaseTwoStudy** package provides a method to approximate a maximal sample size $maxN$ such that the search algorithm stops if $n = maxN$. The parameter $maxN$ is approximated in a way that the optimal design is included among the identified designs. To determine $maxN$, n_1 is set to the minimal possible value (as described above) and r_1 is set to 0. After that, an algorithm searches over all possible values of $maxN$ and $r \in [1, maxN]$, where $maxN$ is increased until a combination of r_1, n_1, r, and $maxN$ is found such that the corresponding error constraints are fulfilled. The idea behind this approach is that $maxN$ has to be very large when n_1 is set to the minimal possible value and r_1 is set to 0. There is no formal proof that the search algorithm will always find the optimal design when using $maxN$ as upper boundary for the total sample size. However, in the multitude of examples we considered there was no case indicating this approach may be wrong.

If not all possible designs should be identified but only optimal, minimax, and admissible designs, the algorithm can be further improved by taking into account the inequalities $EN(\pi_0)_{\text{optimal}} \leq EN(\pi_0)_{\text{admissible}} \leq EN(\pi_0)_{\text{minimax}}$ and $n_1 < EN(\pi_0)$ for $\pi_0 > 0$. Therefore, n_1 is smaller than $EN(\pi_0)_{\text{admissible}}$ for admissible designs and smaller than $EN(\pi_0)_{\text{optimal}}$ for the optimal design. Consequently, the above described algorithm starts searching for n_1 up to a maximal value of $n - 1$ until the first solution is identified. The maximal value for n_1 is then replaced by $EN(\pi_0)$ of this design, and whenever another solution is identified the upper bound of the range of n_1 is replaced with the smallest $EN(\pi_0)$ found so far.

Point estimation, confidence intervals, and p values

In an early work by Girshick, Mosteller, and Savage (1946), unbiased estimators for several samples from binomial distributions were developed. Based on this approach, Jung and Kim (2004) derived an unbiased estimator for the true response rate and proved that it is the uniformly minimum variance unbiased estimator (UMVUE). Let t_1 denote the number of responses in the first stage and t denote the cumulative number of responses when the trial is continued to the second stage, then this estimator is given by

$$\hat{\pi}_{\text{UMVUE}} = \begin{cases} \dfrac{\binom{n_1 - 1}{t_1 - 1}}{\binom{n_1}{t_1}} = \frac{t_1}{n_1} & \text{if } t_1 \leq r_1, \\ \dfrac{\sum_{i=\max[r_1, t-1-n+n_1]}^{\min[t-1, n_1-1]} \binom{n_1 - 1}{i} \binom{n - n_1}{t - 1 - i}}{\sum_{i=\max[r_1+1, t-n+n_1]}^{\min[t, n_1]} \binom{n_1}{i} \binom{n - n_1}{t - i}} & \text{if } t_1 > r_1. \end{cases} \tag{4}$$

For the derivation of appropriate p values and confidence intervals that match the test decision and that take into account the sequential nature of the design, an ordering of the sample space has to be defined. Armitage (1957) suggested a stage-wise ordering of the sample space. In the case of Simon's design, stage-wise ordering means that outcomes observed in the second stage of the trial are more extreme than outcomes observed in the first stage of the trial. Another option would be to sort the sample space based on the UMVUE. For Simon's design this approach, however, leads to the same ordering. The corresponding p value is given by Jung, Owzar, and George (2006) and Koyama and Chen (2008). For the stage-wise ordering, Koyama and Chen (2008) derived the p value for testing $H_0 : \pi \leq \pi_0$ by

$$p = \begin{cases} 1 - \sum_{x_1=0}^{t-1} \binom{n_1}{x_1} \pi_0^{x_1} \cdot (1 - \pi_0)^{n_1 - x_1} & \text{if } t \leq r_1, \\ \sum_{x_1=r_1+1}^{n_1} \binom{n_1}{x_1} \pi_0^{x_1} (1 - \pi_0)^{n_1 - x_1} \cdot \sum_{x_2=\max[0, t-x_1]}^{n-n_1} \binom{n - n_1}{x_2} \pi_0^{x_2} (1 - \pi_0)^{n - n_1 - x_2} & \text{if } t > r_1. \end{cases} \tag{5}$$

This p value reflects the decision rule of the underlying design in that the null hypothesis can be rejected at level α if and only if $p \leq \alpha$. Furthermore, a two-sided $(1 - 2\alpha)$-confidence interval $[\hat{\pi}_L, \hat{\pi}_U]$ can be obtained by inverting the test: $[\hat{\pi}_L, \hat{\pi}_U]$ includes all values π_0^* for which the p value for testing $H_0^* : \pi \leq \pi_0^*$ within the given two-stage design lays within the

interval $[\alpha, 1-\alpha]$. Note that this $(1-2\alpha)$-confidence interval (CI) matches the test decision as the null hypothesis H_0 is rejected if and only if $\hat{\pi}_L > \pi_0$. Jovic and Whitehead (2010) give a nice overview on the computation and evaluation of point estimates and confidence intervals for single-arm two-stage designs.

Non-stochastic and stochastic curtailment

Non-stochastic and stochastic curtailment are based on the conditional power, i.e., the probability to reject $H_0 : \pi \leq \pi_0$ after the second stage under the alternative $\pi = \pi_1$ given the results observed so far. If we denote by $\tilde{n}, 0 \leq \tilde{n} \leq n$, the number of patients for which the outcome has been observed and by k the number of responses that occurred for these patients, the null hypothesis cannot be rejected after the second stage if $r_1 - k + 1 > n_1 - \tilde{n}$ or $r - k + 1 > n - \tilde{n}$, independently on any result that may be observed for future patients. The conditional power is thus zero in those cases, and stopping the trial for this reason is referred to non-stochastic curtailment. Stochastic curtailment means to stop the study for futility if the conditional power falls below a pre-defined threshold θ $(0 < \theta < 1)$. A formula for the conditional power when a stochastic curtailment procedure is applied in both stages of the study is given in the Appendix of Kunz and Kieser (2012). This formula was implemented in our package to allow a simulation-based investigation of the effect of including stochastic curtailment with a defined threshold θ. Monte Carlo simulations are used to generate possible study outcomes based on the corresponding binomial distribution under H_0 and H_1, respectively. The type I and type II error rates are estimated by the relative frequencies of rejection or acceptance of H_0 over all simulated data sets. Furthermore, $PET(\pi_0)$ and $EN(\pi_0)$ are simulated analogously.

2.2. Adaptive single-arm two-stage designs

The "classical" two-stage designs discussed in Section 2.1 require strict adherence to the sample sizes and decision rules pre-specified in the planning stage of the study. In case of deviations from these values, control of the type I error rate is no longer guaranteed (Englert and Kieser 2012a). For practical applications, this is a severe restriction. Englert and Kieser (2012b) defined the conditional error function for two-stage designs with discrete outcomes based on the approach initially introduced for continuous test statistics (see, e.g., Proschan and Hunsberger 1995; Posch and Bauer 1999; Müller and Schäfer 2001). It can be shown that any "classical" one-arm two-stage design with a binary endpoint can be re-written in terms of the conditional error function CE, which is given by

$$CE(k) = \begin{cases} 0 & \text{if } k \leq r_1, \\ 1 - B(n_2, r - k, \pi_0) & \text{if } r_1 < k \leq r, \\ 1 & \text{if } k > r, \end{cases} \tag{6}$$

where k defines the number of responses observed in the first stage. The p values p_1 and p_2 of the first and second stage, respectively, are given by $p_1(k) = 1 - B(n_1, k-1, \pi_0)$ and $p_2(l) = 1 - B(n - n_1, l - 1, \pi_0)$, where k and l denote the number of observed responses at stage 1 and 2. Then the null hypothesis can be rejected if $p_2(l) \leq CE(k)$. Furthermore, the type I error rate is controlled when applying this decision rule even if arbitrary design modifications are performed after the first stage, e.g., a recalculation of the sample size based on the results of the interim analysis (Englert and Kieser 2012b). Due to the discreteness

of the outcome and with it the test statistic, the available type I error rate α is usually not exhausted but the actual level

$$\alpha' = \sum_{k=0}^{n_1} CE(k) \mathsf{P}_{H_0}(P_1 = p_1(k)) \tag{7}$$

generally falls below α. By increasing the boundaries of the "natural" conditional error function $CE(k)$ given above and thereby implementing the remaining level $\alpha - \alpha'$, the conservatism can be reduced and the efficiency can be increased (Englert and Kieser 2012b). Such a modification of the conditional error function can be done in a multitude of ways. The software package includes the options of increasing the boundaries equally, proportionally to the probability of observing $p_1(k)$, or increasing only the smallest value of the conditional error function that is unequal to zero.

2.3. Subset designs

Lin, Allred, and Andrews (2008) proposed a single-arm phase II design which is based on two endpoints where a response for endpoint 1 implies a response for endpoint 2. Thus, endpoint 1 defines a subset of endpoint 2, as, e.g., in case of disease-free survival and overall survival as endpoints 1 and 2. These designs are also called "Simon's designs with ordinal outcomes". However, due to ease of readability we use the term "subset design". The decision to continue to the second stage is based only on one endpoint. In our package we implemented a subset design where the decision to proceed is based on endpoint 1.

The global test problem for the subset design is given by

$$\begin{aligned} & H_0 : (H_0^1 : \pi_{\text{sub}} \leq \pi_{\text{sub}_0}) \cap (H_0^2 : \pi_{\text{super}} \leq \pi_{\text{super}_0}), \\ \text{versus} \quad & \\ & H_a : (H_a^1 : \pi_{\text{sub}} > \pi_{\text{sub}_a}) \cup (H_a^2 : \pi_{\text{super}} > \pi_{\text{super}_a}), \end{aligned} \tag{8}$$

where π_{sub} and π_{super} denote the true response rates for endpoint 1 (subset) and endpoint 2 (superset), respectively. The probabilities π_{sub_0} and π_{super_0} denote the response rates for endpoint 1 and endpoint 2 under the global null hypothesis, and π_{sub_a} and π_{super_a} denote the response rates under the global alternative hypothesis.

Identification of designs

The subset design implemented in the **OneArmPhaseTwoStudy** package is defined by the parameters (n_1, r_1, n, r, s), where more than r_1 responses for endpoint 1 under the first n_1 patients are needed to proceed to the second stage. To reject H_0 after the second stage, more than r responses for endpoint 1 or more than s responses for endpoint 2 under all n enrolled patients are needed.

The probability of rejecting the global null hypothesis for true response rates π^*_{sub} and π^*_{super} is given by

$$\begin{aligned} 1 - \Bigg[& \sum_{x_1=0}^{r_1} b(n_1, x_1, \pi^*_{\text{sub}}) + \sum_{x_1}^{\min[r,n_1]} \sum_{y_1=x_1}^{\min[s,n_1]} m(n_1, x_1, y_1 - x_1, \pi^*_{\text{sub}}, \pi^*_{\text{super}} - \pi^*_{\text{sub}}) \\ & \cdot \sum_{x_2=0}^{\min[r-x_1, s-y_1]} \sum_{y_2=x_2}^{\min[n_2, s-y_1]} m(n_2, x_2, y_2 - x_2, \pi^*_{\text{sub}}, \pi^*_{\text{super}} - \pi^*_{\text{sub}}) \Bigg], \end{aligned} \tag{9}$$

where $m(\cdot)$ denotes the multinomial probability mass function and $n_2 = n - n_1$ (see e.g., Kunz and Kieser 2011a).

Evaluating (9) at $\pi^*_{\text{sub}} = \pi_{\text{sub}_0}$ and $\pi^*_{\text{super}} = \pi_{\text{super}_0}$ or $\pi^*_{\text{sub}} = \pi_{\text{sub}_1}$ and $\pi^*_{\text{super}} = \pi_{\text{super}_a}$ provides the type I error rate and power, respectively. The probability of early termination and the expected sample size under the global null hypothesis are based on the subset endpoint only and are therefore given by

$$PET(\pi_{\text{sub}_0}) = \sum_{x_1=0}^{r_1} b(n_1, x_1, \pi_{\text{sub}_0}), \tag{10}$$

$$EN(\pi_{\text{sub}_0}) = n_1 + (1 - PET(\pi_{\text{sub}_0})) \cdot n_2. \tag{11}$$

Under all designs fulfilling the type I and type II error constraints, the optimal design is defined as the one with the smallest $EN(\pi_{\text{sub}_0})$. If this solution is not unique, the one with the highest power is chosen. The minimax design is the one with smallest total sample size n. If more than one design with smallest n exists, the one with smallest $EN(\pi_{\text{sub}_0})$ is selected as minimax design (Kunz and Kieser 2011a). Admissible designs can be derived in the same way as described in Section 2.1.

The algorithm to determine subset designs fulfilling the type I and II error requirements and to choose among them the optimal, minimal, and admissible designs follows the description in Kunz (2011). This algorithm has many similarities with the algorithm described in Section 2.1 to detect Simon's designs but includes only a few changes. As before, a naive algorithm to detect subset designs searches for each value of n over $n_1 \in [1, n-1]$ and $r_1 \in [0, n_1 - 1]$ as well as $r \in [r_1 + 1, n - n_1 + r_1]$ and $s \in [r, n]$ for those designs for which expression (9) is less than or equal to α at $\pi^*_{\text{sub}} = \pi_{\text{sub}_0}$ and $\pi^*_{\text{super}} = \pi_{\text{super}_0}$ and equal or larger than $1 - \beta$ at $\pi^*_{\text{sub}} = \pi_{\text{sub}_1}$ and $\pi^*_{\text{super}} = \pi_{\text{super}_1}$. This approach can be improved in a similar way as described in Section 2.1. The starting value for n_1 is determined in the same way as for the Simon's design but is based on the response rate for endpoint 1 under the alternative hypothesis. Therefore, n_1 is ceil$(\log(\beta)/\log(1 - \pi_{\text{sub}_1}))$ while it is 2 for $\beta = 1 - \pi_{\text{sub}_1}$. As n has to be larger than n_1, the starting value for n is $n_1 + 1$. As for the Simon's two-stage designs, the same inequality $B(n_1, r_1, \pi_{\text{sub}_a}) \geq 1 - \beta$ has to hold true for every pair (r_1, n_1) with $r_1 \in [0, n_1 - 1]$. If the inequality does not hold true, n_1 is increased by 1 and the search continues; otherwise, the algorithm searches backwards over r in the range of $[n - n_1 + r_1, r_1 + 1]$. Because of the fact that the actual type I error rate for Simon's design α'_{simon} for the parameter set (n_1, r_1, n, r) is smaller or equal to the actual type I error rate of a subset design with the same parameters, the condition $\alpha'_{\text{simon}}(n_1, r_1, n, r) \leq \alpha$ is checked for every value of r. If it holds true, the algorithm searches over s in the range of $[r, n-1]$, else r_1 is increased and the search continues. The last step of the algorithm is to check whether the type II error rate of the multinomial test for the parameter set (n_1, r_1, n, r, s) under the alternative is less than β. If this condition holds true, the type I error rate and power are calculated via (9) and the algorithm continues. Otherwise, s is skipped and r is decreased by 1. As for Simon's design, the inequalities $EN(\pi_{\text{sub}_0})_{\text{optimal}} \leq EN(\pi_{\text{sub}_0})_{\text{admissible}} \leq EN(\pi_{\text{sub}_0})_{\text{minimax}}$ and $n_1 < EN(\pi_{\text{sub}_0})$ hold true for subset designs, which leads to the same improvement of the algorithm as described in Section 2.1.

Point estimation, confidence intervals, and p values

Based on the work of Girshick *et al.* (1946), the uniformly minimum variance unbiased esti-

mator for endpoint 1 ($\hat{\pi}_{\text{sub,UMVUE}}$) can be obtained by (4), and for endpoint 2 ($\hat{\pi}_{\text{super,UMVUE}}$) the estimator is given by

$$\widehat{\pi}_{\text{super,UMVUE}} = \begin{cases} \frac{t_1}{n_1} + \frac{\sum_{i=0}^{r_1} \binom{n_1-1}{i}\binom{n_1-1-i}{u_1-t_1-1}}{\sum_{i=0}^{r_1} \binom{n_1}{i}\binom{n_1-i}{u_1-t_1}} & \text{if } t_1 \leq r_1, \\ \frac{\sum_{i=\max[r1,t-1-n+n_1]}^{\min[t-1,n_1-1]} \binom{n_1-1}{i}\binom{n-n_1}{t-1-i}}{\sum_{\max[r_1+1,t-n+n_1]}^{\min[t,n_1]} \binom{n_1}{i}\binom{n-n_1}{t-i}} + & \\ \qquad \frac{\sum_{i=\max[r_1+1,t-n+n_1]}^{\min[n_1-1,t]} \binom{n_1-1}{i}\binom{n-n_1}{t-i}\binom{n-t-1}{u-t-1}}{\sum_{i=\max[r_1+1,t-n+n_1]}^{\min[n_1,t]} \binom{n_1}{i}\binom{n-n_1}{t-i}\binom{n-t}{u-t}} & \text{if } t_1 > r_1, \end{cases} \tag{12}$$

where t_1 denotes the number of observed responses for endpoint 1 in the first stage, u_1 denotes the observed responses for endpoint 2 in the first stage, and t and u denote the observed responses for endpoint 1 and 2 in the whole trial.

For the derivation of appropriate p values that match the test decision, Kunz (2011) derived the following formula

$$p_{\text{exact}} = \sum_{x_1=r_1+1}^{n_1} \sum_{y_1=x_1}^{n_1} \sum_{x_2=\max[0,t-x_1]}^{n-n_1} \sum_{y_2=\max[x_2,u-y_1]}^{n-n_1} \binom{n_1}{y_1}\binom{y_1}{x_1}\binom{n-n_1}{y_2}\binom{y_2}{x_2} \cdot \pi_{\text{sub}_0}^{x_1+x_2}(\pi_{\text{super}_0} - \pi_{\text{sub}_0})^{y_1+y_2-x_1-x_2}(1-\pi_{\text{super}_0})^{n-y_1-y_2}. \tag{13}$$

Since the exact p value depends on π_{sub_0} and π_{super_0}, the confidence interval for the response rate of endpoint 1 depends on the response rate of endpoint 2 and vice versa. This results in a one-sided confidence area which is called the confidence set (Reiczigel, Abonyi-T'oth, and Singer 2008). The boundary of this area is given by all combinations of $\hat{\pi}_{\text{sub,lower}}$ and $\hat{\pi}_{\text{super,lower}}$ with $p_{\text{exact}}(\hat{\pi}_{\text{sub,lower}}, \hat{\pi}_{\text{super,lower}}) = \alpha$.

The idea and implementation of (non-)stochastic curtailment can be applied to subset designs in the same way as described in Section 2.1 for the Simon's designs.

3. Structure of the package

3.1. Package overview

The **OneArmPhaseTwoStudy** package consists of 25 functions. These functions are implemented for the purpose of planning, monitoring, and analyzing one-arm phase II studies with binary outcomes. The supported designs are two-stage designs with a single endpoint as well

as subset designs, where for the single endpoint designs the "classical" as well as the adaptive variants are available. In the following, all functions will be outlined. Each section is separated into the three parts planning, monitoring, and analysis.

3.2. Classical two-stage designs

The **OneArmPhaseTwoStudy** package implements 8 functions, which are provided solely for the "classical" two-stage designs. In the following each of these functions will be described.

Planning classical two-stage designs

In the planning stage, a main task is to identify an adequate design for the given study situation at hand. Our package implements three functions to fulfill this purpose, which will be described below.

The algorithm to find possible designs for given values of α, β, π_0, and π_1 requires a high computational effort. Therefore, the package uses the programming language C++ internally. Because C++ is a compiled language, computations can be performed up to 80 times faster. The R package **Rcpp** (Eddelbuettel and François 2011) is used to establish a link between R and C++ code. This link is established by calling the function `setupSimon`.

```
setupSimon(alpha = 0.05, beta = 0.05, p0 = 0.1, p1 = 0.3)
```

returns what we will reference as a 'simon' object. This 'simon' object allows access to an internally used C++ object. The arguments `alpha`, `beta`, `p0`, and `p1` correspond to α, β, π_0, and π_1, respectively. The parameters can be changed any time by invoking the function `setSimonParams`.

Once a 'simon' object is generated, the function `getSolutions` can be used to start the search algorithm described in Section 2.1 to identify possible designs for given α, β, π_0, and π_1

```
getSolutions(simon = setupSimon(), useCurtailment = FALSE,
  curtail_All = FALSE, cut = 0, replications = 10000, upperBorder = 0)
```

The first argument passed to `getSolutions` must be a pre-specified 'simon' object. To investigate the effect of (non-)stochastic curtailment, the argument `useCurtailment` must be set to `TRUE`. By this, the function `getSolutions` will determine the changes in the type I and II error rate for all identified designs as well as the impact on $PET(\pi_0)$ and $EN(\pi_0)$. The threshold θ for the conditional power can be specified by the argument `cut` and has to be chosen as a value between 0 and 1. To evaluate the effect of different thresholds simultaneously, the argument `curtail_All` can be set to `TRUE`. In this case, the algorithm will calculate the effect of curtailment for all values from the value of `cut` to 1 in steps of 0.05. This allows the user to get an impression which threshold is weighing the best decrease in sample size and the loss in power. The argument `replications` determines how many studies are simulated to evaluate the effect of curtailment. Due to the fact that C++ is used internally, even large values for `replications` (like 100,000) lead to results within a couple of seconds (tested on a computer with a dual core processor with 2.8GHz).

The function `getSolutions` returns a list object containing several data frames which summarize all identified designs and the consequences of curtailment. The application of function `getSolutions` is described in Section 4.

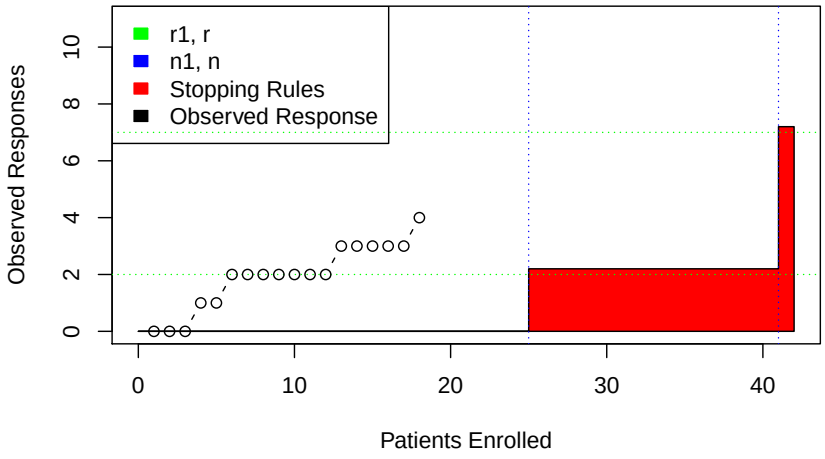

Figure 1: Example of a graphics generated by the function `plot_simon_study_state` without curtailment.

Monitoring classical two-stage designs

The **OneArmPhaseTwoStudy** package includes the two functions `plot_simon_study_state` and `getCP_simon` which are dedicated to monitoring purposes. The first function allows the user to get a graphical overview of the current status of the study. An exemplary call of this function is given below.

```
R> set.seed(25)
R> design <- getSolutions()$Solutions[3, ]
R> stoppingRules <- data.frame(Enrolled_patients = c(design$n1, design$n),
+    Needed_responses_ep1 = c(design$r1, design$r))
R> enrolledPat <- data.frame(ep1 = rbinom(18, 1, design$p1))
R> plot_simon_study_state(stoppingRules, enrolledPat, design$r1, design$n1,
+    design$r, design$n)
```

This call results in the output shown in Figure 1. The horizontal green dashed lines indicate the critical values (r_1, r) for the given two-stage design, whereas the blue lines denote the sample size for the interim and the final analysis (n_1, n). The black circles depict the patients which have already been enrolled. The red area illustrates the stopping rules for the given design. If the black circles enter the red area, the study has to be stopped. Moreover, the user can easily see when the interim analysis has to be performed and which number of responses is required for continuation. When the design is planned without curtailment, the stopping rules are simply defined by r_1, n_1, r, and n. If curtailment is applied, these stopping rules change as there are more options to stop for futility. Figure 2 shows the same design as illustrated in Figure 1 but with stochastic curtailment for a threshold of $\theta = 0.2$ which can be generated through the call given below.

```
R> set.seed(25)
R> tmp <- getSolutions(useCurtailment = TRUE, cut = 0.2)
R> design <- tmp$Solutions[3, ]
R> stoppingRules <-
```

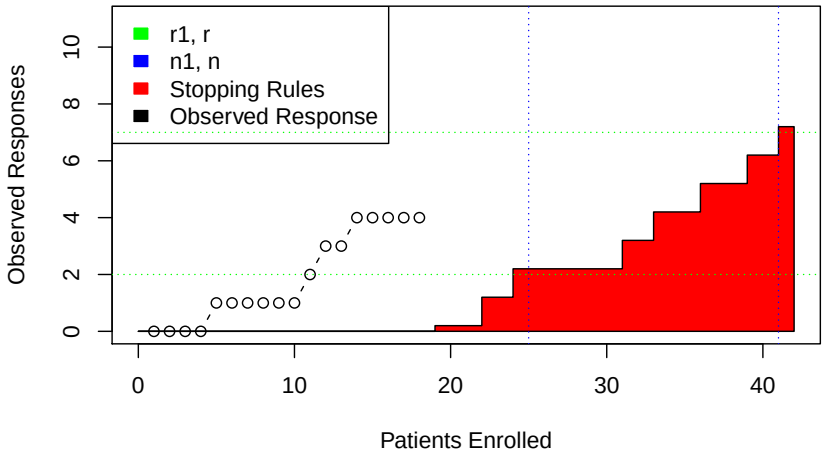

Figure 2: Example of a graphics generated by the function `plot_simon_study_state` with stochastic curtailment.

```
+    tmp$Curtailment_Results$`StoppingrulesForID:2`$`Stoppingrules_for_Row:1`
R> names(stoppingRules) <- c("Needed_responses_ep1", "Enrolled_patients")
R> enrolledPat <- data.frame(ep1 = rbinom(18, 1, design$p1))
R> plot_simon_study_state(stoppingRules, enrolledPat, design$r1,
+    design$n1, design$r, design$n)
```

The second function dedicated to monitoring purposes is the function `getCP_simon` which allows calculating the conditional power at any time point of an ongoing study. This function can also be used to decide whether a study should be stopped for futility when (non-)stochastic curtailment is applied. As arguments, this function requires specification of the number of observed responses, the number of enrolled patients, and the design parameters r_1, n_1, n, and π_1.

Analyzing classical two-stage designs

As the conduct of interim analysis is included in the monitoring procedure, this section focuses on the functions provided for the final analysis of a "classical" two-stage designs, which are `get_p_KC`, `get_CI`, and `get_UMVUE_GMS`.

The function `get_p_KC` calculates the exact p value based on the approach of Koyama and Chen (2008) according to (5) given in Section 2.1. With this tool, the user can decide whether to reject or accept H_0. Based on the function `get_p_KC` it is possible to derive the $(1-2\alpha)$-CI given by $[\hat{\pi}_L, \hat{\pi}_U]$, which can be calculated by calling the function `get_CI`. Internally, this function performs a stage-wise ordering by iterating over different values for $\hat{\pi}_L$, which is increased with every iteration step. Analogously, $\hat{\pi}_U$ is decreased with every iteration step as long as the values of `get_p_KC`$(\hat{\pi}_L)$ and `get_p_KC`$(\hat{\pi}_U)$ are less than α. As mentioned in Section 2.1, H_0 can be rejected if and only if π_0 is less than $\hat{\pi}_L$. The definition of this function is illustrated below.

```
get_CI(k, r1, n1, n, alpha = 0.05, precision = 4)
```

The first argument has to be set to the number of observed responses. The following four

arguments correspond to r_1, n_1, n, and α, respectively. The argument `precision` can be used to select to which digit the result of `get_CI` should be accurate.

Besides a correct test decision, the estimated response rate plays a major role in the planning of proceeding phase III studies. Therefore, the package includes the function `get_UMVUE_GMS` which implements the UMVUE of the true response rate based on the work of Jung and Kim (2004) (see Section 2.1). The listing below illustrates the definition of this function

```
get_UMVUE_GMS(k, r1, n1, n),
```

where `k`, `r1`, `n1`, and `n` correspond to the number of observed responses, the critical value for the first stage, the number of patients enrolled in the first stage, and the total number of patients enrolled in the whole trial, respectively. The calculation of the UMVUE is illustrated in Section 4.3.

3.3. Adaptive two-stage designs

As described in Section 2.2, every "classical" two-stage design presented in Section 2.1 can be "translated" into an adaptive design and may furthermore be improved with respect to efficiency. The **OneArmPhaseTwoStudy** package provides eight functions which implement the algorithms described in Section 2.2.

Planning adaptive two-stage designs

To plan an adaptive two-stage design, the first step is to identify a "classical" two-stage design using the functions described in Section 3.2. After that, a rule to increase the boundaries of the conditional error function $CE(k)$ must be specified. For this purpose, the package implements four functions denoted by `getD_none`, `getD_equally`, `getD_proportional`, and `getD_distributeToOne`. These functions return data frames with all possible values of k (number of observed responses at the interim analysis) and the corresponding value of the conditional error function. The function `getD_none` corresponds to the case where the remaining level $\alpha - \alpha'$ is not used to modify the conditional error function but where the original function $CE(k)$ is used. The functions `getD_equally`, `getD_proportional`, and `getD_distributeToOne` spend the remaining level $\alpha - \alpha'$ by increasing the boundaries returned by $CE(k)$ either equally, proportionally to the probability of observing $p_1(k)$, or to the smallest value of $CE(k)$ that is unequal to zero, respectively.

Monitoring adaptive two-stage designs

For monitoring purposes, the same functions as described in Section 3.2 can be used. The main differences to a "classical" design occur during the conduct of the interim analysis, which is described in the next section.

Analyzing adaptive two-stage designs

As mentioned in Section 2.2, adaptive designs allow to modify the number of patients to be enrolled in the second stage based on the results of the interim analysis. The package implements three functions which are dedicated for this purpose and which are denoted by `getCP`, `getN2`, and `get_r2_flex`.

The function `getCP` returns the conditional power of the study if the number of patients to be enrolled in the second stage is changed to `n2` when `k` responses were observed at the interim analysis and under the assumption that `p1` is the true response rate.

```
getCP(n2, p1, design, k, mode = 0, alpha = 0.05)
```

The argument `design` is to be specified as a data frame containing the columns `r1`, `n1`, `r`, `n`, and `p0` which correspond to the values of r_1, n_1, r, n, and π_0. The assumed true response rate is that under the alternative hypothesis and is given by `p1`. The argument `mode` dedicates in which way the remaining level $\alpha - \alpha'$ is spent to modify the boundaries returned by $CE(k)$. `mode` has to be a value in $\{0, 1, 2, 3\}$ where 0 indicates that the remaining level was not spent (`getD_none`), 1 indicates a proportional spending (`getD_proportional`), 2 indicates an equal spending (`getD_equally`), and 3 stands for an allocation to the smallest value of $CE(k)$ which is unequal to zero. The argument `alpha` specifies the overall type I error rate of the study.

The function `getN2` returns the number of patients to be enrolled in the second stage in order to achieve the specified conditional power.

```
getN2(cp, p1, design, k, mode = 0, alpha = 0.05)
```

The arguments of `getN2` are exactly the same as for `getCP` with the only difference that the first argument specifies the conditional power the study should achieve.

Changing the number of patients to be enrolled after the interim analysis results in a different critical value to be applied to the number of responses observed in the second stage of the study. To calculate the new value for r_2, the function `get_r2_flex` can be used. This function requires three arguments: The first argument is the conditional error (value of the modified $CE(k)$), the second is π_0, and the last argument is n_2. As outlined in Section 2.2, H_0 can be rejected if $p_2(l) \leq CE(k)$, where $p_2(l)$ is implemented in the function `getP`.

3.4. Subset designs

The following sections will outline all functions of the **OneArmPhaseTwoStudy** package which support the subset designs described in Section 2.3.

Planning subset designs

Planning a subset design follows similar steps as for the "classical" two-stage designs with a single endpoint described in Section 3.2. The corresponding functions supporting these steps for subset designs are given by `setupSub1Design`, `setSub1Params`, and `getSolutionsSub1`.

Similar to the procedure described in Section 3.2, at first a 'sub1' object has to be defined which establishes a link between C++ and R code. By this, it is possible to perform the calculations much faster as compared to plain R code. Nevertheless, the identification of subset designs is computationally more intensive than the identification of "classical" designs. Therefore, depending on the underlying parameter constellation it may take several minutes until all possible designs are identified. To generate a 'sub1' object, the function `setupSub1Design` is used which is illustrated below.

```
setupSub1Design(alpha = 0.1, beta = 0.1, pc0 = 0.6, pt0 = 0.7, pc1 = 0.8,
  pt1 = 0.9)
```

The arguments `alpha` and `beta` are used to specify the significance level and the type II error rate. The other arguments are used to set π_{sub_0}, π_{super_0}, π_{sub_a}, and π_{super_a}, respectively. The arguments can be changed any time by invoking the function `setSub1Params`.

The identification of subset designs is similar to the two-stage designs with a single endpoint. The function `getSolutionsSub1` starts the search algorithm described in Section 2.3. This function accepts the same arguments as `getSolutions` but uses a 'sub1' object instead of a 'simon' object. Moreover, the function `getSolutionsSub1` provides the additional arguments `skipS`, `skipR`, and `skipN1` which should be set either to `TRUE` or `FALSE`. These arguments instruct the search algorithm to skip the range of s, r or n_1 every time a design is identified which fulfills the type I and II error constraints. This results in a performance improvement. However, if one or more of these arguments are set to `TRUE` the algorithm will only be able to determine the minimax, admissible, and optimal design among the identified designs. This does not assure that the overall minimax, admissible, and optimal designs are found.

Monitoring subset designs

The **OneArmPhaseTwoStudy** package provides two functions for monitoring subset designs. The function `plot_sub1_study_state` generates a plot similar to the one as described in Section 3.2. The listing below illustrates the application of this function.

```
R> sub1 <- setupSub1Design(0.05, 0.1, 0.5, 0.6)
R> design <- getSolutionsSub1(sub1, FALSE, FALSE, FALSE)$Solutions[4, ]
R> sr <- data.frame(Enrolled_patients = c(design$n1, design$n),
+    Needed_responses_ep1 = c(design$r1, design$r),
+    Needed_responses_ep2 = c(0, design$s))
R> tmp_ep1 <- rbinom(8, 1, design$pc1)
R> tmp_ep2 <- tmp_ep1 | rbinom(8, 1, design$pt1)
R> enrolledPat <- data.frame(ep1 = tmp_ep1, ep2 = tmp_ep2)
R> plot_sub1_study_state(sr, enrolledPat, design$r1, design$n1, design$r,
+    design$s, design$n)
```

This call results in the plot shown in Figure 3. The dashed green lines indicate the critical values for endpoint 1 (r_1, r). The dark blue line represents the critical value for endpoint 2 (s). The red dashed lines indicate the sample sizes for the first stage and for the whole trial (n_1,n). The red and the blue area represent the stopping rules for endpoint 1 and 2, respectively. Finally, the connected black circles represent the number of responses for endpoint 1 and the x-es represent the number of responses for endpoint 2.

The second function dedicated to monitoring purposes is `get_conditionalPower` which can be used to calculate the conditional power for a given subset design in a similar manner as described in Section 3.2 for the "classical" two-stage designs. The function requires specification of the number of observed responses for endpoint 1 and 2 as well as the number of enrolled patients. Moreover, the parameter set (r_1, n_1, r, s, n, π_{sub_a}, π_{super_a}) has to be provided.

Analyzing subset designs

The **OneArmPhaseTwoStudy** package implements four functions which can be used to perform the final analysis of a subset design, three of which are designated to calculate the exact p value and the confidence set which are described in Section 2.3.

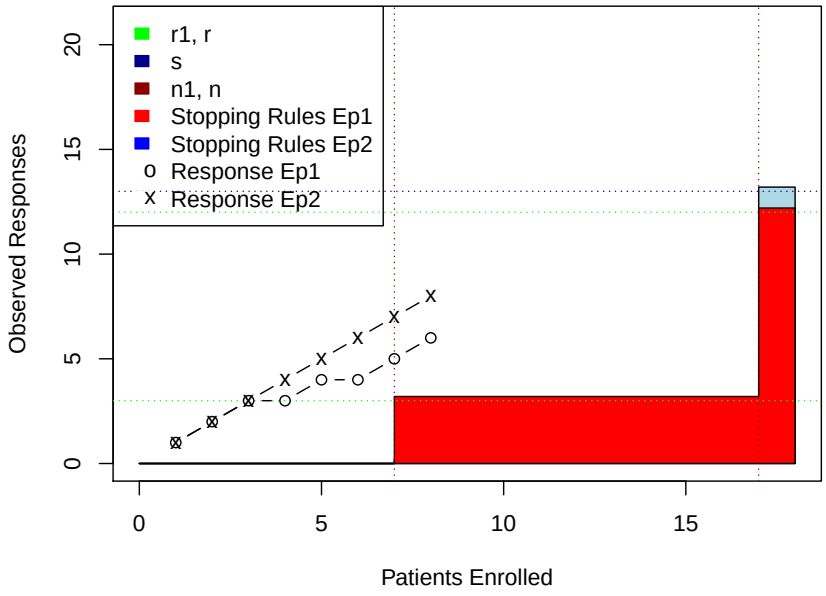

Figure 3: Example of a graphics generated by the function `plot_sub1_study_state`.

The function `get_p_exact_subset` computes the exact p value for a given subset design (see Equation 13). The function is defined as follows.

```
get_p_exact_subset(t, u, r1, n1, n, pc0, pt0, sub1 = setupSub1Design())
```

The first two arguments `t` and `u` have to be set equal to the number of responses observed for endpoint 1 and 2, respectively. The following arguments correspond to the values of r_1, n_1, n, π_{sub_0}, and π_{super_0}. As the decision to proceed to the second stage of the study is based on r_1 only, the function does not depend on the parameters r and s. The last argument `sub1` is internally used by the function `get_confidence_set` and should not be overwritten.

As described in Section 2.3, the confidence interval for the response rate of endpoint 1 depends on the response rate of endpoint 2 and vice versa which results in a so-called confidence set. The boundaries of this confidence set can be calculated through the function `get_confidence_set`. Internally, this function uses `get_p_exact_subset` for different values of `pc0` and `pt0` to determine different sets of $[\hat{\pi}_{\text{sub,lower}}, \hat{\pi}_{\text{super,lower}}]$ for which $p_{\text{exact}}(\hat{\pi}_{\text{sub,lower}}, \hat{\pi}_{\text{super,lower}}) \leq \alpha$. To illustrate the calculated confidence set, the function `plot_confidence_set` can be used. A call of this function results in a plot as shown in Figure 4 where the green area represents the confidence set. The black dot illustrates the point estimate of the true response rates of endpoint 1 and 2 given by $[\hat{\pi}_{\text{sub,UMVUE}}, \hat{\pi}_{\text{super,UMVUE}}]$. Finally, the red area indicates the acceptance area which means that H_0 cannot be rejected if the confidence set overlaps with this region.

The point estimates $\hat{\pi}_{\text{sub,UMVUE}}$ and $\hat{\pi}_{\text{super,UMVUE}}$ are provided by function `get_UMVUE_GMS` for the subset endpoint and function `get_UMVUE_GMS_subset_total` for the superset endpoint.

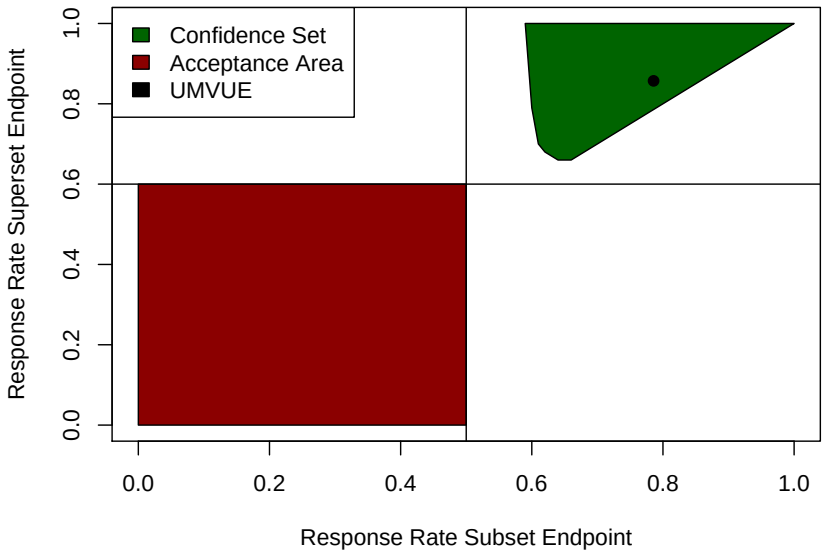

Figure 4: Example of a graphics generated by the function `plot_confidence_set`.

3.5. Graphical user interface

In addition to the R package **OneArmPhaseTwoStudy**, a graphical user interface (GUI) was developed in order to provide an easy to use application. The GUI is implemented in **Qt** (Nord and Chambe-Eng 2017) which is an extension to the C++ standard (for more information visit `http://qt-project.org/`). This extension is especially suited for the development of platform-independent graphical interfaces. The purpose of the GUI is to provide the full functionality of the **OneArmPhaseTwoStudy** package to users with no or limited knowledge in R. Internally, the GUI uses the R package **Rinside** (Eddelbuettel and François 2015) to access the **OneArmPhaseTwoStudy** package. A GUI installer for Windows can be downloaded at `http://www.klinikum.uni-heidelberg.de/fileadmin/inst_med_biometrie/Aktuelles/R-Paket/installer.exe`). Also the source code is available on GitHub at `https://github.com/imbi-heidelberg/OneArmPhaseTwoStudy_GUI`. The tools provided by the GUI are the same as described in the Sections 3.2 to 3.4. Therefore, the following subsections will only exemplarily illustrate the application of the GUI.

Study planning with the GUI

To plan a new study, the option "Create new study" in the "File" menu must be selected. After that, some general information like the study name, the principal investigator, and the name of the involved biometrician must be provided. Once the general information has been entered, the GUI displays a window with all available design options (Figure 5).

At first, a choice between Simon's two-stage design or the subset design must be made. If "Simon's Design" is selected (see section *a*) of Figure 5), all necessary design parameters (α, β, π_0, π_1) have to be entered in section *b*) of Figure 5. After that, the search algorithm described in Section 2.1 can be started by pressing the button "Start calculation" which internally invokes the function `getSolutions` provided by the R package. All identified designs are displayed in a table as illustrated in section *c*) of Figure 5. With a click into the

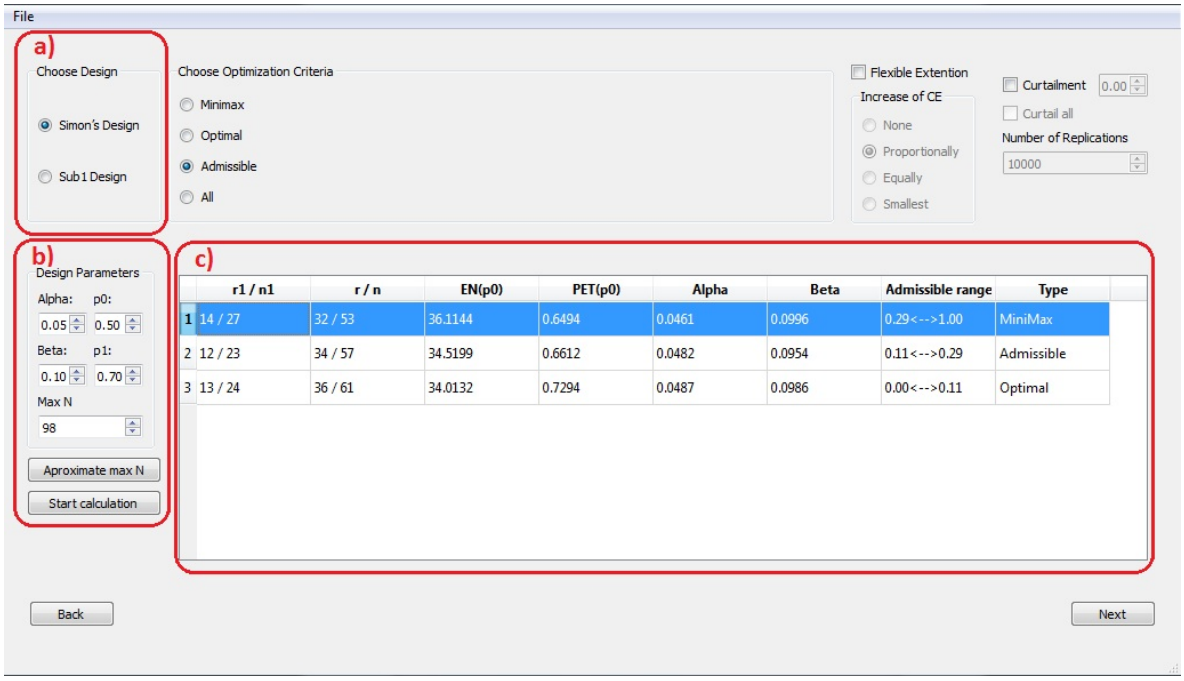

Figure 5: Design calculation with the GUI: *a)* Selection of design type; *b)* Area for entering the required design parameters (α, β, π_0, π_1); *c)* Table to display the identified designs.

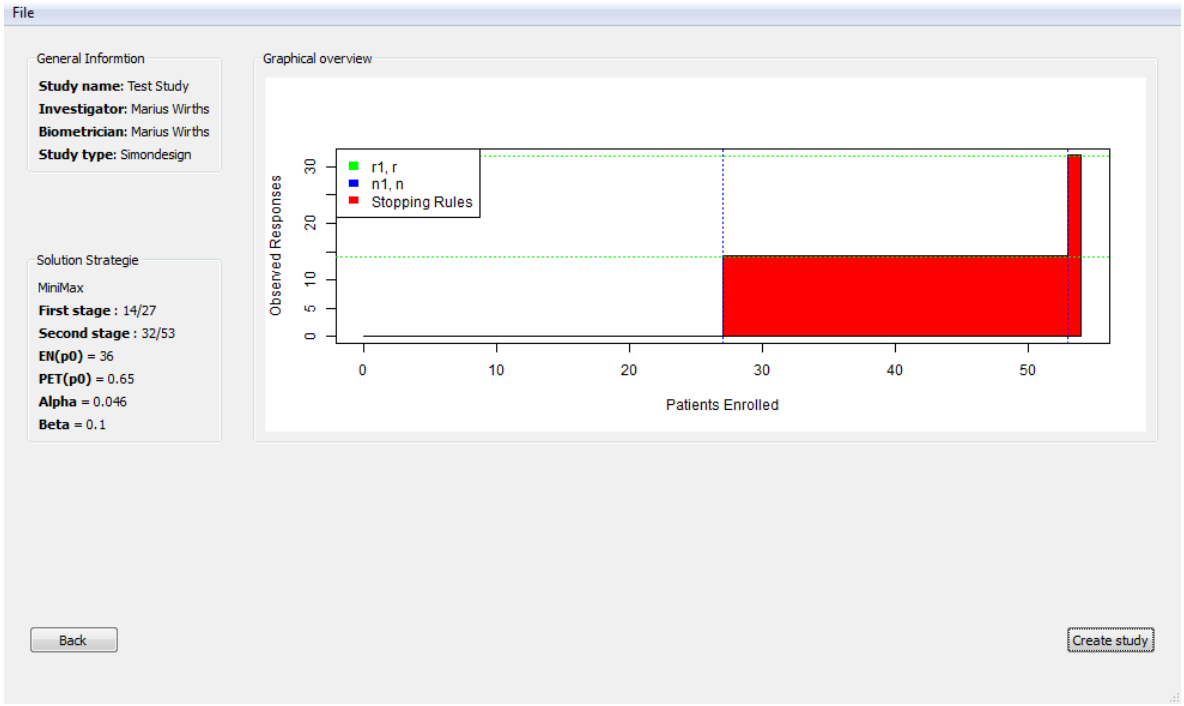

Figure 6: Overview of the characteristics of the selected design provided by the GUI as the final step of the planning phase.

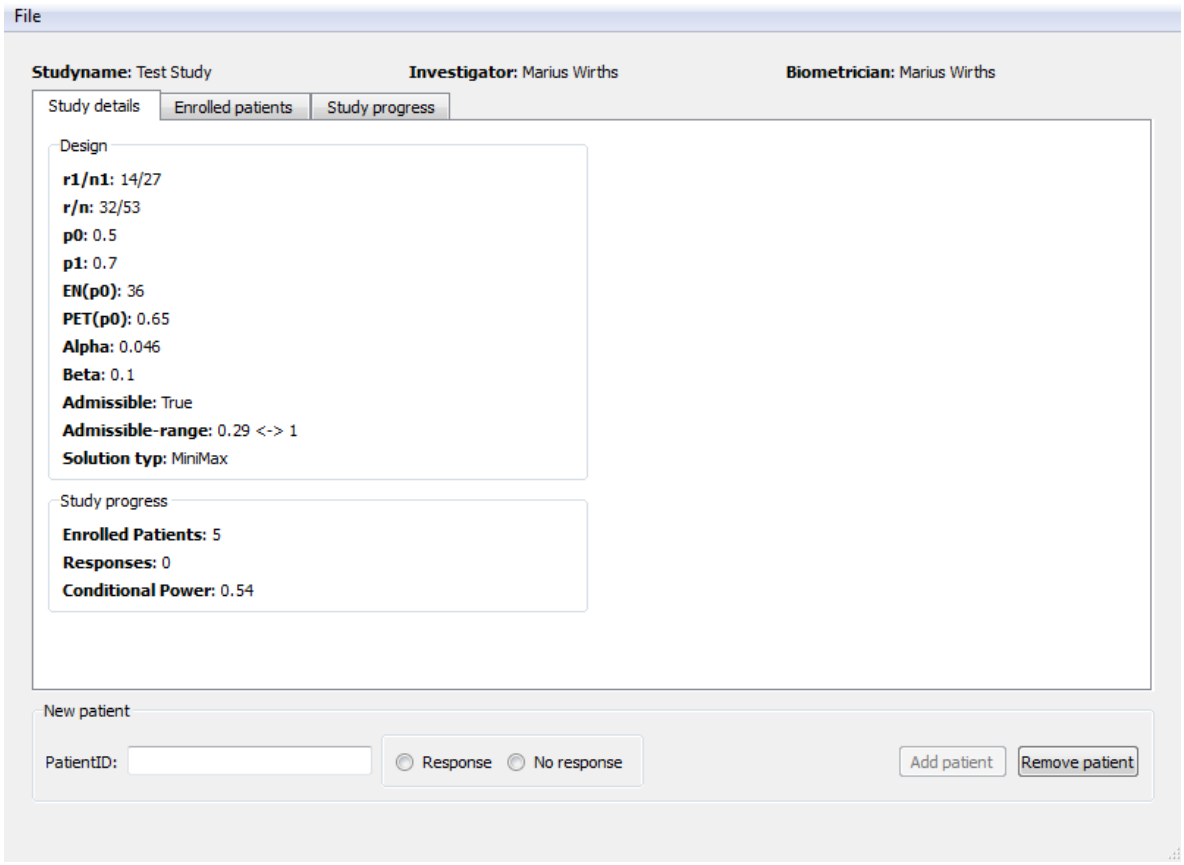

Figure 7: The "Study details" page of the monitoring mode of the GUI.

table, the user can select which design to use. The GUI provides an overview of all selections made during the planning of the study by clicking "Next" (see Figure 6).

If the selected design should be applied, the user has to click on the "Create study" button. Then a save menu will be provided so that this design can be re-used for monitoring and analysis of the study at any time later.

Monitoring with the GUI

After planning a new or opening a previously saved study, the GUI continues to the monitoring mode. To add a new patient to the study, a patient ID must be provided and the information whether or not a response was observed for this patient. With a click on "Add patient" the provided information is included into the study. It is possible to save the current study state at any time through the "File" menu.

Moreover, the monitoring mode of the GUI provides three different pages which sum up all available information up to the current study state. The first page "Study details" displays all design parameters as well as the number of enrolled patients, the number of observed responses, and the current conditional power (Figure 7). On the second page "Enrolled patients", all included patients are displayed in a table. The third page "Study progress" (see Figure 8) shows a graphical overview of the current study state, which is internally generated

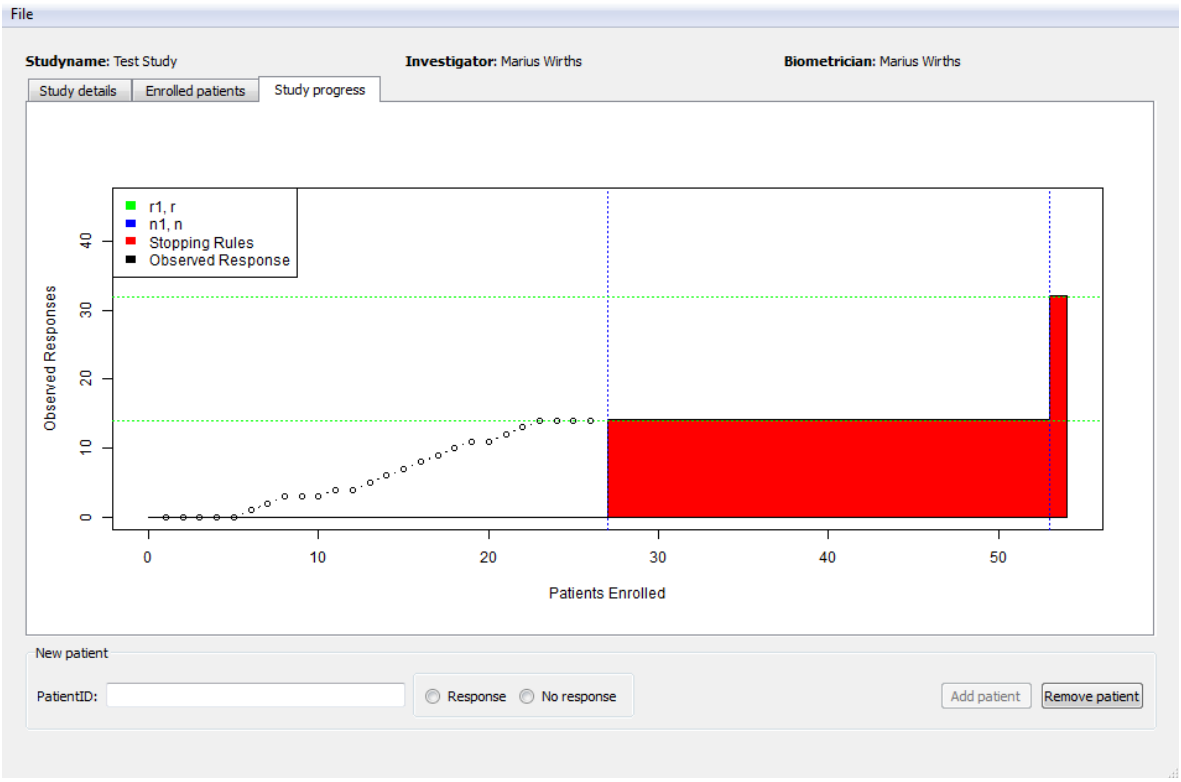

Figure 8: The "Study progress" page of the monitoring mode of the GUI.

through a call of the function `plot_simon_study_state` (see Section 3.2).

After having entered n_1 patients, a pop-up message gives the information that the interim analysis is to be performed. Depending on the number of observed responses, the pop-up message reports whether the study has to be stopped or can proceed to the second stage.

If the study was planned in an adaptive design, the number of patients to be enrolled in the second stage of the trial can be changed at the interim analysis. Note that it is impossible to change the number of patients to be enrolled in the second stage after more than n_1 patients are enrolled. If further patients are added to the study, the GUI switches back to the monitoring mode until n patients in total are included.

Analyzing with the GUI

After having entered a total of n patients, the GUI switches to the final analysis window which shows the results of the test decision as well as the estimated response rate together with the confidence interval (see Figure 9).

In the middle of the screen, the graphical overview of the study with all enrolled patients is displayed, which is internally obtained by a call of `plot_simon_study_state` (see Section 3.2). All design parameters are displayed in the group box "Study design". The group box "Final study state" contains the number of enrolled patients, the number of observed responses, and the exact p value which is internally calculated through invocation of `get_p_KC`. Moreover, two point estimators are provided, namely the MLE and the UMVUE. The MLE is simply calculated by the number of observed responses divided by the number of enrolled patients.

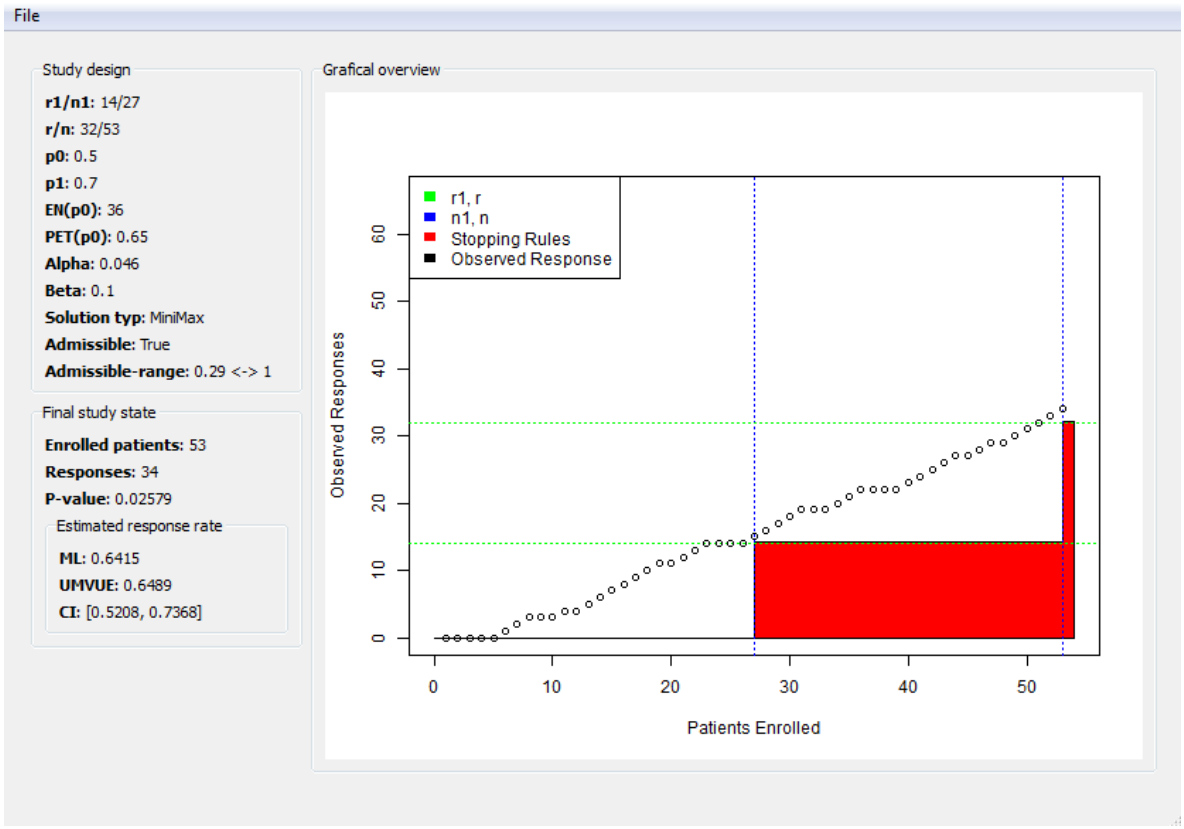

Figure 9: Final analysis window of the GUI.

For the calculation of the UMVUE, the GUI internally invokes the function `get_UMVUE_GMS`. Finally, the $(1 - 2\alpha)$-CI is displayed which is calculated with the function `get_CI`.

4. Example

4.1. Planning

Razak *et al.* (2013) conducted a single-arm phase II trial to investigate the clinical activity of a new orally administered agent in recurrent or metastatic squamous-cell cancer of the head and neck. Primary endpoint was objective response for which the null hypothesis $H_0 : \pi \leq \pi_0 = 0.05$ was assessed at one-sided level $\alpha = 0.05$. A power of $1 - \beta = 0.80$ should be reached for a true objective response rate of $\pi_1 = 0.15$. Simon's optimal two-stage design was implemented. This design is identified by calling the function `getSolutions` that provides the result $(n_1, r_1, n, r) = (23, 1, 56, 5)$. All designs fulfilling the constraints with respect to the type I and type II error rate with a maximum sample size of at most 100 are obtained by using the following code.

```
R> simon <- setupSimon(0.05, 0.20, 0.05, 0.15)
R> designs <- getSolutions(simon, upperBorder = 100)$Solutions
R> hidecol <- c(-1, -6, -7, -9, -11, -14, -15, -16)
R> designs[, hidecol]
```

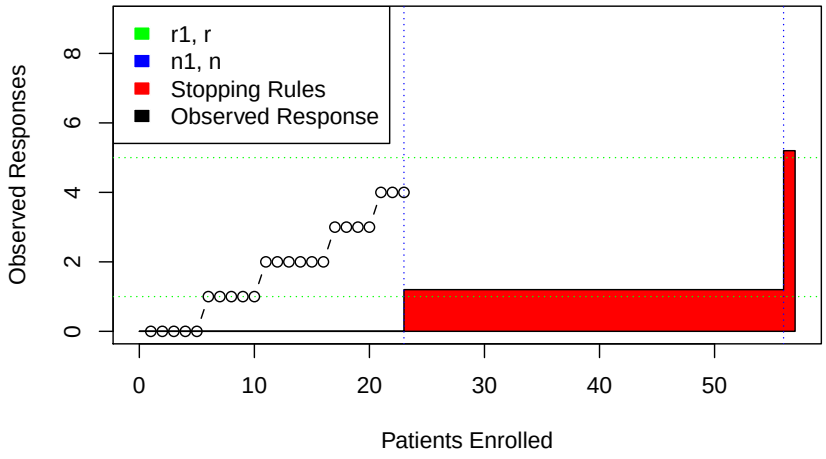

Figure 10: Graphical overview generated by the function `plot_simon_study_state`.

This generates the following output (note that some columns are hidden due to space limitations).

```
  r1 n1 r  n  enP0  petP0  Alpha   Beta        Type
1  0 21 5 52 41.44 0.3406 0.0432 0.1986
2  1 30 5 52 39.82 0.5535 0.0430 0.1980     MiniMax
3  0 18 5 53 39.10 0.3972 0.0453 0.1970
4  1 27 5 53 37.24 0.6061 0.0448 0.1968 Admissible
5  1 25 5 54 35.37 0.6424 0.0463 0.1987 Admissible
6  1 24 5 55 34.51 0.6608 0.0483 0.1977
7  1 23 5 56 33.58 0.6794 0.0500 0.1997     Optimal
```

The maximum sample size of the minimax design is by four lower than for the optimal design while its expected sample size is by more than six higher. Two admissible designs are identified with maximum and expected sample size in between the minimax and optimal ones.

4.2. Statistical monitoring

Four responses were observed in the 23 evaluable patients of the first stage and thus the study proceeded to stage two, which can be seen in the graphical overview (Figure 10) generated by calling

```
R> set.seed(20)
R> sr <- data.frame(Enrolled_patients = c(23, 56),
+    Needed_responses_ep1 = c(1, 5))
R> enrolledPat <- data.frame(ep1 = logical(23))
R> enrolledPat[sample.int(23, 4), ] <- TRUE
R> plot_simon_study_state(sr, enrolledPat, 1, 23, 5, 56)
```

Here, the black circles which represent the enrolled patients together with the observed responses do not fall into the red area at the interim analysis which is indicated by the vertical blue dotted line.

Let us assume that two responses were observed within the 23 patients of the first stage. The study could then be continued but the conditional power to reject the null hypothesis after the second stage amounts to

```
R> getCP_simon(2, 23, 1, 23, 5, 56, 0.15)
```

```
[1] 0.7504551
```

If after a total of 25 (30 / 35) patients still only two responses were observed, the conditional power amounts to 0.7039 (0.5615 / 0.3887) and one may think about stopping the trial for futility based on stochastic curtailment considerations.

4.3. Analysis

After the second stage, seven responses were observed for the 56 evaluable patients enrolled in the study performed by Razak *et al.* (2013). As $r = 5$, this leads to the rejection of the null hypothesis. The MLE of the response rate is $7/56 = 0.11$ and the related "naive" two-sided 90%-CI and one-sided p value that do not take into account the sequential design, are given by [0.0602, 0.2006] and $p = 0.0212$, respectively. To obtain the UMVUE as well as the 90%-CI and p value tailored to the sequential nature of the design, the functions `get_UMVUE_GMS`, `get_CI`, and `get_p_KC` have to be called.

```
R> results <- data.frame(UMVUE = 0, CI_low = 0, CI_high = 0, p_value = 0)
R> results$UMVUE <- get_UMVUE_GMS(7, 1, 23, 56)
R> results$CI_low <- get_CI(7, 1, 23, 56)$CI_low
R> results$CI_high <- get_CI(7, 1, 23, 56)$CI_high
R> results$p_value <- get_p_KC(7, 1, 23, 56, 0.05)
R> results
```

This results in the following output.

```
      UMVUE CI_low CI_high    p_value
1 0.1379133 0.0617 0.21439 0.01882311
```

As can be seen, the MLE underestimates the response rate which is a general feature in two-stage designs with the option of early stopping for futility. Consequently, reporting this estimate may then lead to an inappropriate judgment of the treatment effect.

4.4. Planning and performing adaptive designs

As an alternative to the "classical" Simon's optimal design, the study by Razak *et al.* (2013) could also have been planned within an adaptive framework that allows to react in a flexible way to unforeseen events by data-driven modifications while still controlling the type I error rate. The flexible counterpart of Simon's optimal design that exhausts the available significance level by equal allocation of the available undershoot in type I error rate to the conditional error function can be obtained by calling `getD_distibuteToOne`.

```
R> simon <- setupSimon(0.05, 0.20, 0.05, 0.15)
R> optimal_design <- getSolutions(simon)$Solutions[7, ]
R> optimal_design
R> getD_equally(optimal_design, 0.05)[1:15, ]
```

This results in the following output.

```
    k         ce
1   0 0.00000000
2   1 0.00000000
3   2 0.08085087
4   3 0.22730544
5   4 0.49677692
6   5 0.81810800
7   6 1.00000000
8   7 1.00000000
9   8 1.00000000
10  9 1.00000000
11 10 1.00000000
12 11 1.00000000
13 12 1.00000000
14 13 1.00000000
15 14 1.00000000
```

Values of the conditional error function of 0 or 1, respectively, mean that the study is to be stopped for futility (number of observed responses is less than or equal to r_1) or efficacy (number of observed responses is greater than r) after the first stage. Let us assume that five responses were observed within the 23 patients of the first stage. With the "classical" Simon's optimal design, further 33 patients must be included in stage two although only one additional response has to occur to reject the null hypothesis. In contrast, the adaptive design allows recalculation of the sample size taking into account the result observed in the interim analysis. Within the conditional error rate approach pursued in the adaptive design framework, a p value smaller or equal to 0.818108 has to be achieved in the second stage. Calling the function `getN2` results in

```
R> getN2(0.8, 0.15, optimal_design, 5, 2)
```

```
[1] 10
```

Based on these considerations the total sample size n could, for example, be changed from 56 $(23 + 33)$ to 33 $(23 + 10)$ maintaining a conditional power of 80%, which corresponds to the initial power the study was planned for. Choice of an adaptive design may therefore have led to a much smaller sample size and thus to considerable savings in time and financial resources.

5. Discussion

In this article, we presented an overview of the **OneArmPhaseTwoStudy** package to plan, monitor, and analyze single-arm two-stage clinical trials with a binary outcome. The theory behind the implemented methods is sketched, the package is described in detail, and practical application is illustrated by a real clinical study example. Although to our knowledge **OneArmPhaseTwoStudy** provides the most comprehensive spectrum of methods of available software tools in this field, there are several options for extension of the package. Such extensions may cover designs with more than two stages (Chen 1997) or alternative designs with more than one endpoint (Kunz and Kieser 2011a, 2012). One of the methodological research we are currently pursuing and whose results will be integrated in the package concerns construction of point estimates and confidence intervals for adaptive single-arm two-stage designs. Finally, we are working on the problem on how flexible designs can be used to deal with the situation that the initially specified sample sizes n_1 or n are not exactly met. As this is frequently the case in practice, the availability of related methods and software would be a further major step forward.

Acknowledgments

The first two authors contributed equally.

This work was supported by Bundesministerium für Bildung und Forschung (BMBF), grant 01EZ1206, and Deutsche Forschungsgemeinschaft grants KI708/1-1 and 1-3. The authors would like to thank Stephanie Kovalchik (Associate Editor) and two anonymous reviewers for their helpful and constructive comments on an earlier version of the manuscript.

References

Armitage P (1957). "Restricted Sequential Procedures." *Biometrika*, **44**(1–2), 9–26. doi:10.1093/biomet/44.1-2.9.

Ayanlowo AO, Redden DT (2007). "Stochastically Curtailed Phase II Clinical Trials." *Statistics in Medicine*, **26**(7), 1462–1472. doi:10.1002/sim.2653.

Baghdadi R, Laffler MJ (2013). "The Next Phase in Oncology: FDA's Pazdur Has New Vision for Drug Development." *The Pink Sheet*. doi:10.1097/00042871-200405000-00001.

Chen TT (1997). "Optimal Three-Stage Designs for Phase II Cancer Clinical Trials." *Statistics in Medicine*, **16**(23), 2701–2711. doi:10.1002/(sici)1097-0258(19971215)16:23<2701::aid-sim704>3.0.co;2-1.

Cytel (2015). *Exact Inference in **East** with **East EXACT***. Cambridge.

Eddelbuettel D, François R (2011). "**Rcpp**: Seamless R and C++ Integration." *Journal of Statistical Software*, **40**(8), 1–18. doi:10.18637/jss.v040.i08.

Eddelbuettel D, François R (2015). ***RInside**: C++ Classes to Embed R in C++ Applications*. R package version 0.2.13, URL https://CRAN.R-project.org/package=RInside.

Eisenhauer EA, Therasse P, Bogaerts J, Schwartz LH, Sargent D, Ford R, Dancey J, Arbuck S, Gwyther S, Mooney M, Rubinstein L, Shankar L, Dodd L, Kaplan R, Lacombe D, Verweij J (2009). "New Response Evaluation Criteria in Solid Tumours: Revised Recist Guideline (Version 1.1)." *European Journal of Cancer*, **45**(2), 228–247. doi:10.1016/j.ejca.2008.10.026.

Englert S, Kieser M (2012a). "Adaptive Designs for Single-Arm Phase II Trials in Oncology." *Pharmaceutical Statistics*, **11**(3), 241–249. doi:10.1002/pst.541.

Englert S, Kieser M (2012b). "Improving the Flexibility and Efficiency of Phase II Designs for Oncology Trials." *Biometrics*, **68**(3), 886–892. doi:10.1111/j.1541-0420.2011.01720.x.

Englert S, Kieser M (2015). "Methods for Proper Handling of Over- and Underrunning in Phase II Designs for Oncology Trials." *Statistics in Medicine*, **34**(13), 2128–2137. doi:10.1002/sim.6479.

Gan HK, Grothey A, Pond GR, Moore MJ, Siu LL, Sargent D (2010). "Randomized Phase II Trials: Inevitable or Inadvisable?" *Journal of Clinical Oncology*, **28**(15), 2641–2647. doi:10.1200/jco.2009.26.3343.

Girshick MA, Mosteller F, Savage LJ (1946). "Unbiased Estimates for Certain Binomial Sampling Problems with Applications." *The Annals of Mathematical Statistics*, **17**(1), 13–23. doi:10.1214/aoms/1177731018.

Jovic G, Whitehead J (2010). "An Exact Method for Analysis Following a Two-Stage Phase II Cancer Clinical Trial." *Statistics in Medicine*, **29**(30), 3118–3125. doi:10.1002/sim.3837.

Jung SH, Kim KM (2004). "On the Estimation of the Binomial Probability in Multistage Clinical Trials." *Statistics in Medicine*, **23**(6), 881–896. doi:10.1002/sim.1653.

Jung SH, Lee T, Kim K, George SL (2004). "Admissible Two-Stage Designs for Phase II Cancer Clinical Trials." *Statistics in Medicine*, **23**(4), 561–569. doi:10.1002/sim.1600.

Jung SH, Owzar K, George SL (2006). "p-Value Calculation for Multistage Phase II Cancer Clinical Trials." *Journal of Biopharmaceutical Statistics*, **16**(6), 765–775. doi:10.1080/10543400600825645.

Kirk JL, Fay MP (2014). "An Introduction to Practical Sequential Inferences via Single Arm Binary Response Studies Using the **binseqtest** R Package." *The American Statistician*, **68**(4), 230–242. doi:10.1080/00031305.2014.951126.

Koyama T, Chen H (2008). "Proper Inference from Simon's Two-Stage Designs." *Statistics in Medicine*, **27**(16), 3145–3154. doi:10.1002/sim.3123.

Kunz C (2011). *Two-Stage Designs for Phase II Trials with One or Two Endpoints*. Ph.D. thesis, Medical Faculty, Ruprechts-Karls-Universität Heidelberg.

Kunz C, Kieser M (2011a). "Optimal Two-Stage Designs for Single-Arm Phase II Oncology Trials with Two Binary Endpoints." *Methods of Information in Medicine*, **50**(4), 372–377. doi:10.3414/me10-01-0037.

Kunz CU, Kieser M (2011b). "Simon's Minimax and Optimal and Jung's Admissible Two-Stage Design with or without Curtailment." *The Stata Journal*, **11**(2), 240–254.

Kunz CU, Kieser M (2012). "Estimation of Secondary Endpoints in Two-Stage Phase II Oncology Trials." *Statistics in Medicine*, **31**(30), 4352–4368. doi:10.1002/sim.5585.

Lin X, Allred R, Andrews G (2008). "A Two-Stage Phase II Trial Design Utilizing Both Primary and Secondary Endpoints." *Pharmaceutical Statistics*, **7**(2), 88–92. doi:10.1002/pst.255.

Müller HH, Schäfer H (2001). "Adaptive Group Sequential Designs for Clinical Trials: Combining the Advantages of Adaptive and of Classical Group Sequential Approaches." *Biometrics*, **57**(3), 886–891. doi:10.1111/j.0006-341x.2001.00886.x.

NCSS (2015). ***PASS*** *Software, Version 14*. Kaysville. URL http://www.ncss.com/software/pass/.

Nord H, Chambe-Eng E (2017). ***Qt****: Cross-Platform Software Development for Embedded & Desktop*. The **Qt** Company. URL https://www.qt.io.

Posch M, Bauer P (1999). "Adaptive Two Stage Designs and the Conditional Error Function." *Biometrical Journal*, **41**(6), 689–696. doi:10.1002/(sici)1521-4036(199910)41:6<689::aid-bimj689>3.0.co;2-p.

Proschan MA, Hunsberger SA (1995). "Designed Extension of Studies Based on Conditional Power." *Biometrics*, **51**(4), 1315–1324. doi:10.2307/2533262.

Razak ARA, Soulieres D, Laurie SA, Hotte SJ, Singh S, Winquist E, Chia S, Le Tourneau C, Nguyen-Tan PF, Chen EX, Chan KK, Wang T, Giri N, Mormont C, Quinn S, Siu LL (2013). "A Phase II Trial of Dacomitinib, an Oral Pan-Human EGF Receptor (HER) Inhibitor, as First-Line Treatment in Recurrent and/or Metastatic Squamous-Cell Carcinoma of the Head and Neck." *The Annals of Oncology*, **24**(3), 761–769. doi:10.1093/annonc/mds503.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Reiczigel J, Abonyi-T'oth Z, Singer J (2008). "An Exact Confidence Set for Two Binomial Proportions and Exact Unconditional Confidence Intervals for the Difference and Ratio of Proportions." *Computational Statistics & Data Analysis*, **52**(11), 5046–5053. doi:10.1016/j.csda.2008.04.032.

SAS Institute Inc (2015). *SAS Software, Version 9.4*. Cary. URL http://www.sas.com/.

Seshan VE (2015). ***clinfun****: Clinical Trial Design and Data Analysis Functions*. R package version 1.0.11, URL https://CRAN.R-project.org/package=clinfun.

Simon R (1989). "Optimal Two-Stage Designs for Phase II Clinical Trials." *Controlled Clinical Trials*, **10**(1), 1–10. doi:10.1016/0197-2456(89)90015-9.

Southwest Oncology Group (2015). *Two-Stage Phase II Clinical Trials*. Kaysville. PASS Sample Size Software, URL https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/PASS/Two-Stage_Phase_II_Clinical_Trials.pdf.

StataCorp (2015). *Stata Statistical Software: Release 14*. Cambridge. URL http://www.stata.com/.

Wirths M (2017). ***OneArmPhaseTwoStudy***: *Planing, Monitoring and Evaluating Oncological Phase 2 Studies*. R package version 1.0.3, URL https://CRAN.R-project.org/package=OneArmPhaseTwoStudy.

Affiliation:

Meinhard Kieser, Marius Wirths
Institute of Medical Biometry and Informatics
University of Heidelberg
Im Neuenheimer Feld 305
69120 Heidelberg, Germany
E-mail: kieser@imbi.uni-heidelberg.de, wirths@imbi.uni-heidelberg.de

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
November 2017, Volume 81, Issue 8 *Submitted:* 2014-12-15
doi:10.18637/jss.v081.i08 *Accepted:* 2016-07-27