BASKEXACT: AN R PACKAGE FOR ANALYTICAL CALCULATION OF BASKET TRIAL OPERATING CHARACTERISTICS

Lukas Baumann
Institute of Medical Biometry
University of Heidelberg
69120 Heidelberg, Germany
baumann@imbi.uni-heidelberg.de

March 26, 2024

ABSTRACT

Basket trials are a new type of clinical trial in which a treatment is investigated in several subgroups. For the analysis of these trials, information is shared between the subgroups based on the observed data to increase the power. Many approaches for the analysis of basket trials have been suggested, but only a few have been implemented in open source software packages. The R package baskexact facilitates the evaluation of two basket trial designs which use empirical Bayes techniques for sharing information. With baskexact, operating characteristics for single-stage and two-stage designs can be calculated analytically and optimal tuning parameters can be selected.

1 Motivation and significance

1.1 Introduction

Basket trials are a relatively new type of clinical trial in which a new treatment is tested in different subgroups. They are most commonly used in uncontrolled early-phase oncology trials, where a binary endpoint such as tumour response is investigated. The subgroups, also called baskets, are usually defined based on the tumour histology. All patients in the trial typically share a common genetic feature that is targeted by the treatment under investigation [1]. Many designs for the analysis of basket trials have been proposed, often using Bayesian techniques such as Bayesian hierarchical modelling or Bayesian model averaging. By utilising these methods, information is shared between the subgroups based on the observed data, which is the defining characteristic of a basket trial from a statistical point of view [2]. By sharing information, power is increased as compared to a separate analysis in each basket. This is essential, as sample sizes in the subgroups are often small due to their definition based on a certain genetic feature and tumour histology.

In their comprehensive review, Pohl et al. [2] categorised around 20 different basket trial designs and further designs were published in the meantime. Since many basket trial designs are methodology complex, reliable software implementation is vital for a design to be considered by researchers. However, only a fraction of the published designs are implemented in open source software [see 3, for an overview]. Searching the package list of the Comprehensive R Archive Network (CRAN) for the term "basket trial" only results in three hits (besides baskexact, the package presented in this article). basket [4, 5] implements the multisource exchangeability model design [6]. bhmbasket [7] implements two designs based on Bayesian hierarchical modelling, one by Berry et al. [8] and the EXNEX design by Neuenschwander et al. [9]. bmabasket [10] implements a design based on Bayesian model averaging by Psioda et al. [11]. These methods are computationally relatively complex and thus operating characteristics are evaluated by simulation in these R packages. The methods implemented in bhmbasket additionally require Markov Chain Monte Carlo sampling to calculate posterior probabilities as the posterior distributions are not available in closed form when Bayesian hierarchical modelling is applied.

In this paper, the R package baskexact [12], which implements the power prior basket trial design [13] and a design by Fujikawa et al. [14] (henceforth referred to as Fujikawa's design) is presented. Both designs share information

between baskets in a similar way using empirical Bayes methods. The power prior design is based on the power prior methodology which was initially proposed to borrow strength from historical data [15]. Since both designs are not fully Bayesian, computation of posterior distributions is much cheaper than that of many other basket trial design. Posterior probabilities are available in closed form and can therefore easily be calculated even for a large number of baskets. Even analytical computation of operating characteristics is feasible in some settings. `baskexact` facilitates analytical computation of operating characteristics for single-stage and two-stage designs in basket trials with the power prior and Fujikawa's design, with equal sample sizes per subgroup and up to 5 baskets.

1.2 Statistical background

An uncontrolled basket trial with a binary endpoint and $K \geq 2$ baskets is considered. Let n_k be the sample size in basket k. The number of responses in basket k is denoted by r_k, the vector of responses in all baskets is denoted by $\boldsymbol{r}$. The objective of the trial is to identify the baskets in which the response probabilities, denoted by p_k, $0 \leq k \leq K$, is larger than a null response rate p_0, that is of no clinical interest. Thus, the null hypotheses $H_{0,k} : p_k \leq p_0$ are tested against the alternatives $H_{1,k} : p_k > p_0$.

In the power prior design, at first a beta prior with parameters $s_{1,k}$ and $s_{2,k}$ is specified for each baskets. Decisions are based on the following posterior distributions, which share the information between baskets:

$$\pi(p_k|\boldsymbol{r}, \boldsymbol{\omega_k}) = \text{Beta}(s_{1_k} + \sum_{i=1}^{K} \omega_{k,i} r_i, s_{2,k} + \sum_{i=1}^{K} \omega_{k,i}(n_i - r_i))$$

where $\omega_{k,i} \in [0,1]$ are the sharing weights and $\boldsymbol{\omega_k} = (\omega_{k,1}, \ldots, \omega_{k,K})$, with $\omega_{k,k} = 1$. $\omega_{k,i}$ is thus the proportion of information from basket i that is used in the analysis of basket k. A null hypothesis is rejected if $P(p_k > p_0|\boldsymbol{r}, \boldsymbol{\omega_k}) \geq \lambda$, where $\lambda \in (0,1)$ is prespecified.

The information sharing mechanism in the power prior design is very similar to that of Fujikawa's design. The difference between the sharing approaches in the two designs is solely that in Fujikawa's design prior parameters are also included in the weighted sums, and thus the prior information is also considered for the information sharing.

The sharing weights can be computed in various ways. In Fujikawa's design, they are derived from the pairwise Jensen-Shannon divergences (JSD) between the individual posterior distributions of the baskets in the following way:

$$\omega_{k,i} = \begin{cases} (1 - \text{JSD}(\pi(p_k|r_k), \pi(p_i|r_i))^\varepsilon & \text{if } (1 - \text{JSD}(\pi(p_k|r_k), \pi(p_i|r_i))^\varepsilon > \tau, \\ 0 & \text{otherwise,} \end{cases}$$

where $\pi(p_k, r_k)$ is the individual posterior distribution of basket k (derived from a beta-binomial model) and ε and τ are tuning parameters. Weights based on the JSD can of course also be used in the power prior design, i.e. without sharing the prior information as in Fujikawa's design.

Another way to calulcate the weights is derived from the so-called calibrated power prior (CPP) approach [16]. CPP weights are based on the Kolmogorov-Smirnov test statistic, which for binary data is simply the absolute difference in response rates. CPP weights are calculated as follows:

$$\omega_{k,i} = \frac{1}{1 + \exp(a + b \log(d_{k,i} \max(n_k, n_i)^{1/4}))},$$

where $d_{k,i}$ denotes the absolute difference in response rates between basket k and basket i.

In a two-stage design, an interim analysis is performed after a certain number of observations $n_{1,k}$ is available in each basket. Fujikawa et al. [14] suggest interim analyses based on the posterior predictive probability, which is the probability (based on prior information and interim data) that a result for which the null hypothesis can be rejected will be observed at the end of the study. If this probability is below or above a certain prespecified level in basket k, then that basket is stopped for futility or efficacy. Interim decisions may also be based on the same posterior distribution that is used for the final analysis.

For details about the two designs, the reader is referred to Baumann et al. [13] and Fujikawa et al. [14].

2 Software description

2.1 Software architecture

`baskexact` is an R package, available on CRAN [13] and GitHub (https://github.com/lbau7/baskexact). It is written using R's S4 and S3 methods. `baskexact` introduces two S4 classes, `OneStageDesign` and

`TwoStageDesign`, corresponding to a single-stage and a two-stage basket trial design, respectively. This enables easy extension of the package. A vignette is provided which explains how `baskexact` can be extended.

To reduce computation times, some internal functions are implemented in C++, utilising the R packages Rcpp [17] and RcppArmadillo [18]. The doFuture package [19, 20] is used to enable parallelisation of some functions. With doFuture only a single function call is necessary to initialise parallelisation. The parallel backend can be chosen by the user.

2.2 Software functionalities

`baskexact` facilitates the evaluation of a basket trial with the power prior design and Fujikawa's design. It enables the analytical computation of the operating characteristics and selection of optimal tuning parameter values for single-stage and two-stage basket trials with equal sample sizes in all baskets. The most important functions are listed in Table 1. `baskexact` provides functions to compute basic operating characteristics such as the type 1 error rate (TOER), power and the expected sample size. In basket trials, also the expected number of correct decisions (ECD) is important as it in a sense combines TOERs and power of all baskets to a single number which is useful for comparing different designs and tuning parameter values.

Table 1: Main functions of `baskexact`

Function	Description
`toer`	computation of family-wise and basket-wise TOERs
`pow`	computation of family-wise and basket-wise power
`ecd`	computation of expected number of correct decisions
`ess`	computation of expected sample size
`estim`	computation of mean posterior means and mean squared errors
`adjust_lambda`	find λ that protects FWER at a certain level
`opt_design`	find optimal tuning paramter values based on ECD
`plot_weights`	plot a weight function

To select the type of weights that are used for information sharing, a weight function has to be passed to the argument `weight_fun` in each function. The key weight functions that can be selected are `weights_cpp`, for the power prior design with CPP weights and `weights_fujikawa` for Fujikawa's design, i.e. with weights based on the JSD. Both weight functions have additional tuning parameters which are passed as a list to the argument `weight_params`. With the the function `opt_design` the optimal tuning parameter values from a grid of values can be selected. Optimisation is based on the mean ECD across one or several scenarios, while protecting the family-wise TOER (FWER) under the global null hypothesis at a certain level.

In a two-stage design, additionally the type of interim analysis has to be specified using the argument `interim_fun`. Available options are `interim_postpred` for an interim analysis based on the posterior predictive probability and `interim_posterior` for an interim analysis based on the posterior probability. To specify the probability stopping boundaries, both functions have two arguments: `prob_futstop` and `prob_effstop` for the futility and efficacy probability boundaries, respectively. These are passed as a list to `interim_params`. Note that by setting `prob_effstop` to 1, a design that only allows stopping for futility can be implemented.

3 Illustrative examples

The first step to use `baskexact` is always to create a design object corresponding either to a single-stage or a two-stage design (i.e. with one interim analysis) using the functions `setupOneStageBasket` and `setupTwoStageBasket`, respectively. For example:

```
> library(baskexact)
> design <- setupTwoStageBasket(k = 3, shape1 = 1, shape2 = 1, p0 = 0.2)
```

k corresponds to the number of baskets, `shape1` and `shape2` to the two prior parameters of the beta prior distribution. Note that in `baskexact` equal priors for all baskets are assumed. `p0` refers to the response probability under the null hypothesis. The design object is the first argument of most function calls in `baskexact` to select the appropriate S4 method. For example, to compute the TOER:

```
> toer(
>   design = design,
>   n = 20,
>   n1 = 10,
>   lambda = 0.95,
>   interim_fun = interim_postpred,
>   interim_params = list(prob_futstop = 0.1, prob_effstop = 0.9),
>   weight_fun = weights_cpp,
>   weight_params = list(a = 1, b = 1),
>   results = "group"
> )

$rejection_probabilities
[1] 0.0569416 0.0569416 0.0569416

$fwer
[1] 0.1181975
```

`toer` returns the basket-wise as well as the family-wise TOERs. `n` and `n1` refer to the maximum total sample size per basket and the sample size per basket for the interim analysis, respectively. `lambda` corresponds to the probability threshold λ that is used in the final analysis to determine whether a null hypothesis is rejected. `interim_fun` and `interim_params` define the type of interim analysis. With `interim_postpred`, interim decisions are based on the posterior predictive probability to reach a significant result at the end of the trial. `prob_futstop = 0.1` and `prob_effstop = 0.9` define that a basket is stopped for futility if this probability is below 0.1 and is stopped for efficacy if this probability is above 0.9. The other functions for the calculation of operating characteristics, `pow`, `ecd`, `ess` and `estim`, work analogously.

In the planning stage of a basket trial, it may be desirable to select the probability threshold λ, such that the FWER is controlled at a certain level. This can be achieved with `adjust_lambda`, which finds λ such that the one-sided FWER under the global null hypothesis is smaller than a certain level `alpha`.

```
> adjust_lambda(
>   design = design,
>   alpha = 0.05,
>   n = 20,
>   n1 = 10,
>   interim_fun = interim_postpred,
>   interim_params = list(prob_futstop = 0.1, prob_effstop = 0.9),
>   weight_fun = weights_cpp,
>   weight_params = list(a = 1, b = 1),
>   prec_digits = 3
> )

$lambda
[1] 0.982

$toer
[1] 0.04807536
```

Most arguments are the same as in `toer`. The only new arguments are `alpha` and `prec_digits`. The latter specifies the number of decimal places of `lambda`. As the outcomes are binary, in general there is no λ such that the FWER is exactly α, as is seen in the output. Sometimes increasing `prec_digits` results in a FWER closer the nominal level. Note that changing any of the design parameters (tuning parameters of the weight function or of the interim function) affects the FWER, i.e. the value for `lambda` is only valid for the given set of parameter values.

With `opt_design`, tuning parameter values are sorted by their performance in terms of the mean ECD across a set of scenarios. A default set of scenarios can be created using `get_scenarios`:

```
> get_scenarios(design = design, p1 = 0.5)

    0 Active 1 Active 2 Active 3 Active
```

```
[1,]    0.2     0.2     0.2     0.5
[2,]    0.2     0.2     0.5     0.5
[3,]    0.2     0.5     0.5     0.5
```

This creates a matrix of response probability scenarios with an increasing number of baskets that are truly active. The null response probability is taken from the `design` object and p1 specifies the response probability of the active baskets. Of course this set of scenarios can be extended or modified, e.g. to include scenarios in which active baskets have different response probabilities.

In `opt_design`, for each combination of tuning parameter values that is passed to `weight_params`, at first λ is selected to control the FWER at a certain level and then, using the selected posterior threshold, the ECD is calculated for each scenario. Tuning parameters are then sorted by their mean ECD, where the mean over all scenarios is calculated. `opt_design` can be run in parallel. For example, for a single-stage design:

```
> design <- setupOneStageBasket(k = 3, p0 = 0.2)
> plan(multisession, workers = 4)

> opt_design(
>    design = design,
>    n = 20,
>    alpha = 0.05,
>    weight_fun = weights_cpp,
>    weight_params = list(a = 1:3, b = 1:3),
>    scenarios = get_scenarios(design = design, p1 = 0.5),
>    prec_digits = 3
> )

  a b Lambda 0 Active 1 Active 2 Active 3 Active Mean_ECD
1 2 1  0.981 2.932813 2.639612 2.636642 2.923344 2.783103
2 3 2  0.984 2.926667 2.655575 2.683766 2.859488 2.781374
3 3 3  0.983 2.928806 2.606198 2.661209 2.923073 2.779822
4 3 1  0.984 2.938167 2.703022 2.668577 2.803763 2.778382
5 2 2  0.978 2.919353 2.544335 2.590948 2.958013 2.753162
6 2 3  0.974 2.914952 2.438605 2.542111 2.976533 2.718050
7 1 1  0.973 2.917011 2.463110 2.468328 2.980259 2.707177
8 1 2  0.974 2.917205 2.365146 2.371869 2.989490 2.660927
9 1 3  0.971 2.888808 2.253843 2.360286 2.992850 2.623947
```

Since the doFuture package is used for parallelisation, calling `plan` is enough to initialise evaluation in parallel. The code in the example makes all code run on 4 cores on the local machine.

For each combination of tuning parameter values, the output contains the value for λ which controls the FWER at the specified level, the ECD for each scenario and the mean ECD. The first row corresponds to the results of the tuning parameter values which achieve the highest mean ECD, the tuning parameter values are sorted in decreasing order by mean ECD.

The amount of information that is shared between two baskets can be visualised with `plot_weights`:

```
> plot_weights(
>    design = design,
>    n = 20,
>    r1 = 10,
>    weight_fun = weights_cpp,
>    weight_params = list(a = 1:3, b = 1:3)
> )
```

Figure 1 shows the created plot. The argument `r1` refers to the number of responses observed in one of the two baskets. The responses in the other basket are varied on the x-axis. The weights resulting from different choices of the tuning parameters a and b are shown on the y-axis. As `r1` is set to 10, the weight is maximal when 10 responses are also observed in the other basket. The weights decline as the difference in responses increases, depending on the choice of tuning parameters values.

5
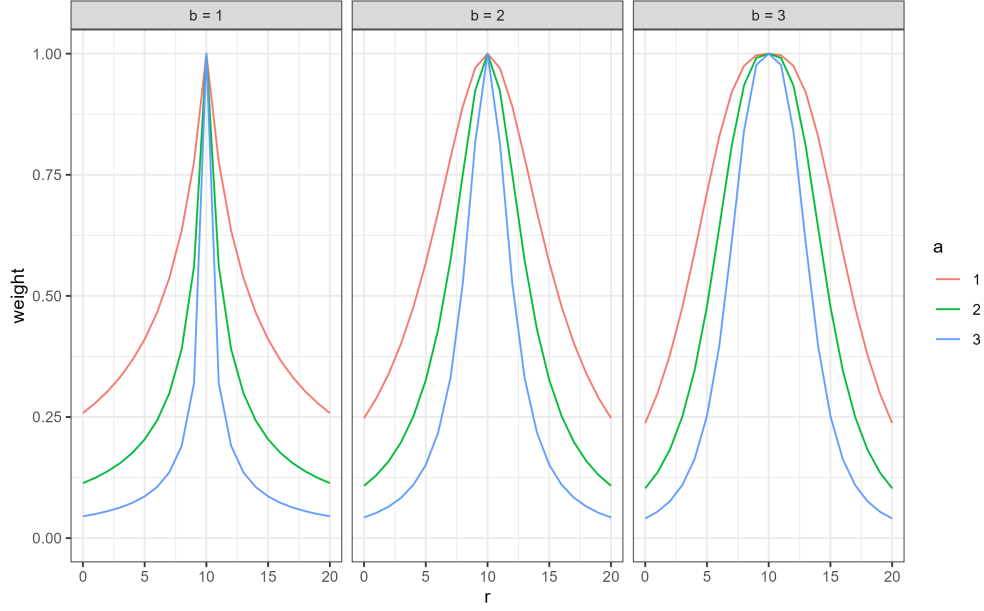
Figure 1: Weight function created with `plot_weights`

4 Impact

While many designs for the analysis of basket trials have been proposed in the literature, open source software implementation is the exception. However, without reliable, user-friendly and well documented software, it is unlikely that a design will be widely adopted. Software implementation is also necessary to enable comparison studies between different designs. Many basket trial designs were only evaluated in the manuscripts in which they were proposed. Thus, there is a large gap in the literature [2]. Neutral comparison studies are necessary [21] to determine which designs work best under which scenarios. Many questions regarding the application of basket trial designs are still open, e.g. the optimal type and timing of interim analyses. `baskexact` can thus be used to investigate the performance of the power prior and Fujikawa's design. As the package is easily extendable, new weight functions for the power prior design can be implemented and evaluated which may further increase its impact.

5 Conclusions

In this manuscript, the R package `baskexact` was presented which implements two closely related basket trial designs, the power prior design and Fujikawa's design, which both share information using an empirical Bayes method and are therefore computationally much cheaper than other Bayesian basket trial designs. `baskexact` enables analytical computation of operating characteristics for up to 5 baskets and single-stage and two-stage designs. The main limitation of `baskexact` is that currently only equal sample sizes per basket are supported. Nevertheless, `baskexact` is a valuable tool for the evaluation of the two implemented designs and can be used by researchers to further compare the performance of different basket trial designs.

Acknowledgements

References

[1] Akihiro Hirakawa, Junichi Asano, Hiroyuki Sato, and Satoshi Teramukai. Master protocol trials in oncology: review and new trial designs. *Contemporary clinical trials communications*, 12:1–8, 2018.

[2] Moritz Pohl, Johannes Krisam, and Meinhard Kieser. Categories, components, and techniques in a modular construction of basket trials for application and further research. *Biometrical Journal*, 63(6):1159–1184, 2021.

[3] Elias Laurin Meyer, Peter Mesenbrink, Tobias Mielke, Tom Parke, Daniel Evans, Franz König, and EU-PEARL (EU Patient cEntric clinicAl tRial pLatforms) Consortium. Systematic review of available software for multi-arm multi-stage and platform clinical trial design. *Trials*, 22:1–14, 2021.

[4] Nan Chen, Brian Hobbs, Alex Kaizer, and Michael J. Kane. *basket: Basket Trial Analysis*, 2021. URL `https://CRAN.R-project.org/package=basket`. R package version 0.10.11.

[5] Michael J. Kane, Nan Chen, Alexander M. Kaizer, Xun Jiang, H. Amy Xia, and Brian P. Hobbs. Analyzing Basket Trials under Multisource Exchangeability Assumptions. *The R Journal*, 12(2):342–358, 2021. doi: 10.32614/RJ-2021-020. URL `https://journal.r-project.org/archive/2021/RJ-2021-020/RJ-2021-020.pdf`.

[6] Brian P Hobbs and Rick Landin. Bayesian basket trial design with exchangeability monitoring. *Statistics in medicine*, 37(25):3557–3572, 2018.

[7] Stephan Wojciekowski. *bhmbasket: Bayesian Hierarchical Models for Basket Trials*, 2022. URL `https://CRAN.R-project.org/package=bhmbasket`. R package version 0.9.5.

[8] Scott M Berry, Kristine R Broglio, Susan Groshen, and Donald A Berry. Bayesian hierarchical modeling of patient subpopulations: efficient designs of phase ii oncology clinical trials. *Clinical Trials*, 10(5):720–734, 2013.

[9] Beat Neuenschwander, Simon Wandel, Satrajit Roychoudhury, and Stuart Bailey. Robust exchangeability designs for early phase clinical trials with multiple strata. *Pharmaceutical statistics*, 15(2):123–134, 2016.

[10] Ethan Alt. *bmabasket: Bayesian Model Averaging for Basket Trials*, 2022. URL `https://CRAN.R-project.org/package=bmabasket`. R package version 0.1.2.

[11] Matthew A Psioda, Jiawei Xu, QI Jiang, Chunlei Ke, Zhao Yang, and Joseph G Ibrahim. Bayesian adaptive basket trial design using model averaging. *Biostatistics*, 22(1):19–34, 2021.

[12] Lukas Baumann. *baskexact: Analytical Calculation of Basket Trial Operating Characteristics*, 2024. URL `https://CRAN.R-project.org/package=baskexact`. R package version 1.0.0.

[13] Lukas Baumann, Lukas Sauer, and Meinhard Kieser. A basket trial design based on power priors. *arXiv preprint arXiv:2309.06988*, 2024.

[14] Kei Fujikawa, Satoshi Teramukai, Isao Yokota, and Takashi Daimon. A bayesian basket trial design that borrows information across strata based on the similarity between the posterior distributions of the response probability. *Biometrical Journal*, 62(2):330–338, 2020.

[15] Joseph G Ibrahim and Ming-Hui Chen. Power prior distributions for regression models. *Statistical Science*, pages 46–60, 2000.

[16] Haitao Pan, Ying Yuan, and Jielei Xia. A calibrated power prior approach to borrow information from historical data with application to biosimilar clinical trials. *Journal of the Royal Statistical Society. Series C, Applied statistics*, 66(5):979, 2017.

[17] Dirk Eddelbuettel, Romain Francois, JJ Allaire, Kevin Ushey, Qiang Kou, Nathan Russell, Inaki Ucar, Douglas Bates, and John Chambers. *Rcpp: Seamless R and C++ Integration*, 2024. URL `https://CRAN.R-project.org/package=Rcpp`. R package version 1.0.12.

[18] Dirk Eddelbuettel, Romain Francois, Doug Bates, Binxiang Ni, and Conrad Sanderson. *RcppArmadillo: 'Rcpp' Integration for the 'Armadillo' Templated Linear Algebra Library*, 2024. URL `https://CRAN.R-project.org/package=RcppArmadillo`. R package version 0.12.8.1.0.

[19] Henrik Bengtsson. *doFuture: Use Foreach to Parallelize via the Future Framework*, 2023. URL `https://CRAN.R-project.org/package=doFuture`. R package version 1.0.1.

[20] Henrik Bengtsson. A unifying framework for parallel and distributed processing in r using futures. *The R Journal*, 13(2):208–227, 2021. doi: 10.32614/RJ-2021-048. URL `https://doi.org/10.32614/RJ-2021-048`.

[21] Anne-Laure Boulesteix, Sabine Lauer, and Manuel JA Eugster. A plea for neutral comparison studies in computational sciences. *PloS one*, 8(4):e61562, 2013.