Analysis of a Group Sequential Trial with a Survival Endpoint using rpact

Marcel Wolbers, Gernot Wassmer, and Friedrich Pahlke

Last change: 15 November, 2022

Contents

Su	Immary	1
1	Introduction	1
2	Inference for the Gallium trial which was stopped at an interim analysis2.1Gallium design background2.2Conventional analyses at the first and second interim analysis2.3Analysis accounting for the group sequential design	1 2 2 2
3	Example: Monitoring an ongoing trial	5

Summary

This R Markdown document provides examples how to analyse a group sequential trial with a survival endpoint and provide inference throughout and at the end of the trial with rpact.

1 Introduction

This tutorial provides two examples:

- The first example illustrates how to get inference (point estimate, confidence interval, and p-value) which respect the group sequential design after rejecting the null hypothesis at the interim or final analysis.
- The second example illustrates tools which are relevant for monitoring interim results of an ongoing trial: repeated confidence intervals and conditional power.

For a general introduction to "Inference in group sequential designs", please refer to the book "Group Sequential and Confirmatory Adaptive Designs in Clinical Trials" by Gernot Wassmer & Werner Brannath.

This tutorial only covers survival endpoints. Code for other endpoints is similar but the dataset needs to be provided in a different format (see **?getDataset** for details).

2 Inference for the Gallium trial which was stopped at an interim analysis

For details about the Gallium trial, we refer to the primary study publication: Marcus et al, N Engl J Med 2017; 377:1331-1344.

2.1 Gallium design background

Trial characteristics:

- Population: Treatment-naive follicular lymphoma patients.
- Comparison: Rituximab + chemotherapy vs. Obinutuzumab + chemotherapy. Rituximab: Rituxan, Mabthera. Obinutuzumab: Gazyva(ro).
- Phase III, 1:1 randomized, open-label clinical trial.
- Primary endpoint: investigator-assessed progression-free survival (PFS).

Group-sequential design:

- O'Brien & Fleming boundary with interim analyses after 30% and 67% of PFS events.
- Non-binding futility after 30% of PFS events (if estimated HR > 1).
- Target HR 0.74, 80% power at two-sided 5% significance level \Rightarrow final analysis at 370 PFS events.
- Target sample size is 1202 subjects.

2.2 Conventional analyses at the first and second interim analysis

Results from standard inference at the futility interim analysis after 113 events:

- Stratified HR 0.69 (95% CI 0.47 to 1.01). \Rightarrow Trial continues.
- $\log(\text{HR}) = \log(0.69)$ with standard error 0.20.
- Corresponding Z-score: $\log(0.69)/0.20 = -1.86$.

Results from standard inference at the efficacy interim analysis after 245 events:

- Stratified HR 0.66 (95% CI 0.51 to 0.85).
- $\log(\text{HR}) = \log(0.66)$ with standard error 0.13.
- Corresponding Z-score: $\log(0.66)/0.13 = -3.225$.
- Two-sided *p*-value is 0.0012 which was smaller than the critical value from the O'Brien-Fleming boundary of 0.012. ⇒ **Trial stopped early for efficacy**.

2.3 Analysis accounting for the group sequential design

First, load the rpact package and **define the group sequential boundaries** using the function getDesignGroupSequential. Note that while the Gallium protocol specified a two-sided significance level of 5%, we implement this via a one-sided significance level of 2.5% as rpact (sensibly) only supports one-sided designs if futility interim analyses are specified.

First, load the rpact package

```
library(rpact)
packageVersion("rpact") # version should be version 2.0.5 or later
```

[1] '3.3.2'

and define the design:

```
# FutilityBounds = c(0,-6) are on the Z-scale; a value of Z = 0 implies futility
# if the interim estimate is "in the wrong direction" (i.e., HR >= 1 here),
# a value of Z = -6 is essentially the same as Z = -Inf and implies no futility
# boundary for the second interim as per the Gallium design
design <- getDesignGroupSequential(
    informationRates = c(113, 245, 370) / 370,
    typeOfDesign = "asOF", sided = 1, alpha = 0.025,
    futilityBounds = c(0, -6), bindingFutility = FALSE
)
```

Note that bindingFutility = FALSE has no impact because it is the default, so actually this could be omitted (same holds for sided = 1 and alpha = 0.025).

Second, the results after the first and second interim are specified using the function getDataset():

```
# overallLogRanks: One-sided logrank statistic or Z-score ( = log(HR)/SE) from Cox regression
results <- getDataSet(
    overallEvents = c(113, 245),
    overallLogRanks = c(-1.86, -3.225),
    overallAllocationRatio = c(1, 1)
)</pre>
```

Since rpact version 3.2, the prefix cumulative[Capital case of first letter of variable name]... or cum[Capital case of first letter of variable name]... can alternatively be used for this. That is,

```
results <- getDataSet(
    cumulativeEvents = c(113, 245),
    cumulativeLogRanks = c(-1.86, -3.225),
    cumulativeAllocationRatio = c(1, 1)
)</pre>
```

defines the same survival dataset. This is used for creating the **adjusted inference** using the function **getAnalysisResults()** (directionUpper = FALSE is specified because the power is directed towards negative values of the logrank statistics):

```
adj_result <- getAnalysisResults(
    design = design,
    dataInput = results,
    directionUpper = FALSE
)
kable(adj_result)</pre>
```

Analysis results (survival of 2 groups, group sequential design)

Design parameters

- Information rates: 0.305, 0.662, 1.000
- Critical values: 3.891, 2.520, 1.992
- Futility bounds (non-binding): 0.000, -Inf
- Cumulative alpha spending: 0.00004995, 0.00587877, 0.02499999
- Local one-sided significance levels: 0.00004995, 0.00586101, 0.02318178
- Significance level: 0.0250
- Test: one-sided

User defined parameters

• Direction upper: FALSE

Default parameters

- Normal approximation: TRUE
- Theta H0: 1

Stage results

- Cumulative effect sizes: 0.7047, 0.6623, NA
- Stage-wise test statistics: -1.860, -2.673, NA
- Stage-wise p-values: 0.031443, 0.003762, NA
- Overall test statistics: -1.860, -3.225, NA
- Overall p-values: 0.0314428, 0.0006299, NA

Analysis results

- Actions: continue, reject and stop, NA
- Conditional rejection probability: 0.1373, 0.8616, NA
- Conditional power: NA, NA, NA
- Repeated confidence intervals (lower): 0.3389, 0.4799, NA
- Repeated confidence intervals (upper): 1.4653, 0.9139, NA
- Repeated p-values: 0.234459, 0.005409, NA
- Final stage: 2
- Final p-value: NA, 0.0006656, NA
- Final CIs (lower): NA, 0.5157, NA
- Final CIs (upper): NA, 0.8515, NA
- Median unbiased estimate: NA, 0.6626, NA

The output is explained as follows:

- Critical values are group sequential efficacy boundary values on the *z*-scale, Local one-sided significance levels are the corresponding one-sided local significance levels.
- Cumulative effect sizes refer to hazard ratio estimates that are based on the overall test statistics.
- Test statistics and p-values refer to z-scores and p-values obtained from the first interim analysis and results which would have been obtained after the second interim analysis if not all data up to the second interim analysis but only new data since the first interim had been included (i.e., per-stage results).
- Overall test statistics are the given (overall, not per-stage) z-scores from each interim and Overall p-value the corresponding one-sided p-values.
- Repeated confidence intervals provide valid (but conservative) inference at any stage of an ongoing or stopped group sequential trial. Repeated p-values are the corresponding p-values.
- Final p-value is the final one-sided adjusted *p*-value based on the stagewise ordering of the sample space.
- Median unbiased estimate and Final CIs are the corresponding median-unbiased adjusted treatment effect estimate and the confidence interval for the hazard ratio at the interim analysis where the trial was stopped.

These results can also be displayed with the summary() function:

kable(summary(adj_result))

Analysis results for a survival endpoint

Sequential analysis with 3 looks (group sequential design). The results were calculated using a two-sample logrank test (one-sided). H0: hazard ratio = 1 against H1: hazard ratio < 1.

1	2	3	
0.305	0.662	1	
3.891	2.520	1.992	
0	-Inf		
< 0.0001	0.0059	0.0250	
< 0.0001	0.0059	0.0232	
0.705	0.662		
-1.860	-3.225		
0.0314	0.0006		
continue	reject and stop		
0.1373	0.8616		
	$ \begin{array}{c} 1 \\ 0.305 \\ 3.891 \\ 0 \\ < 0.0001 \\ < 0.0001 \\ 0.705 \\ -1.860 \\ 0.0314 \\ continue \\ 0.1373 \\ \end{array} $	12 0.305 0.662 3.891 2.520 0-Inf < 0.0001 0.0059 < 0.0001 0.0059 0.705 0.662 -1.860 -3.225 0.0314 0.0006 continuereject and stop 0.1373 0.8616	$\begin{array}{cccccccccccccccccccccccccccccccccccc$

Stage	1	2	3
95% repeated confidence interval	[0.339; 1.465]	[0.480; 0.914]	
Repeated p-value Final p-value Final confidence interval Median unbiased estimate	0.2345	$\begin{array}{c} 0.0054 \\ 0.0007 \\ [0.516; \ 0.852] \\ 0.663 \end{array}$	

Note that for this example, the adjusted final hazard ratio of 0.663 and the adjusted confidence interval of (0.516, 0.852) match the results from the conventional analysis almost exactly for the first two decimals. This is consistent with the finding that stopping a trial after 50% or more of the events had been collected has a negligible impact on estimation.

3 Example: Monitoring an ongoing trial

Monitoring ongoing trials is also possible with the function getAnalysisResults introduced above. Repeated confidence intervals which provide valid (but conservative) inference at any stage of an ongoing or stopped group sequential trial can be obtained using the same code as introduced in the previous example. Conditional power calculations require additional specification of the following arguments:

- The assumed true hazard ratio thetaH1.
- The planned number of additional events for future interim stages (nPlanned).
- The planned allocation ratio allocationRatioPlanned for future interim stages (default is 1).

We illustrate these capabilities by introducing **hypothetical interim results** for the Gallium trial.

Assume the same design as for the Gallium trial introduced above and the following interim results:

Hypothetical results from standard inference at the futility interim analysis after 113 events:

- Stratified HR 0.69 (95% CI 0.47 to 1.01). \Rightarrow Trial would continue.
- $\log(\text{HR}) = \log(0.69)$ with standard error 0.20.
- Corresponding Z-score: $\log(0.69)/0.20 = -1.86$.

Hypothetical results from standard inference at the efficacy interim analysis after 245 events:

- Stratified HR 0.80 (95% CI 0.62 to 1.03). \Rightarrow Trial would continue.
- $\log(\text{HR}) = \log(0.80)$ with standard error 0.13.
- Corresponding Z-score: $\log(0.80)/0.13 = -1.716$.

Calculation of repeated confidence intervals and conditional power:

```
# 1) Specify results so far using function getDataset as before
results <- getDataset(
    cumulativeEvents = c(113, 245),
    cumulativeLogRanks = c(-1.86, -1.716),
    cumulativeAllocationRatio = c(1, 1)
)
# 2) Calculate repeated confidence intervals and conditional power using
# the function getAnalysisResults as before
# Additional arguments for the conditional power calculation are
# - nPlanned: additional events from second interim until final analysis
# (370-245 for this trial)
# - thetaH1: True hazard ratio governing future stages</pre>
```

```
# (set to 0.74 here as per the original protocol assumptions)
interim_results <- getAnalysisResults(
    design = design,
    dataInput = results,
    directionUpper = FALSE,
    nPlanned = 370 - 245, thetaH1 = 0.74
)
kable(summary(interim_results))</pre>
```

Analysis results for a survival endpoint

Sequential analysis with 3 looks (group sequential design). The results were calculated using a two-sample logrank test (one-sided). H0: hazard ratio = 1 against H1: hazard ratio < 1. The conditional power calculation with planned sample size is based on assumed effect = 0.74.

Stage	1	2	3
Fixed weight	0.305	0.662	1
Efficacy boundary	3.891	2.520	1.992
(z-value scale)			
Futility boundary	0	-Inf	
(z-value scale)			
Cumulative alpha spent	< 0.0001	0.0059	0.0250
Stage level	< 0.0001	0.0059	0.0232
Cumulative effect size	0.705	0.803	
Overall test statistic	-1.860	-1.716	
Overall p-value	0.0314	0.0431	
Test action	continue	continue	
Conditional rejection	0.1373	0.1527	
probability			
Planned sample size			125
Conditional power			0.7448
95% repeated confidence	[0.339; 1.465]	[0.582; 1.108]	
interval			
Repeated p-value	0.2345	0.1013	

As per the output above, the recommended action after the second interim analysis of this hypothetical trial would be to continue the trial, a repeated confidence interval for the hazard ratio is (0.582 to 1.108), and the conditional power to reach significance at the final analysis under protocol assumptions is 0.745. Final estimates and *p*-values are still missing as the trial has not stopped yet.

To obtain a **plot of the conditional power over a range of alternatives** you might call the rpact **plot** function and specify the range for theta with **thetaRange**. This produces the conditional power curve together with the likelihood function over the specified range:

plot(interim_results, thetaRange = c(0.6, 1))



Note that nPlanned is from interim_results and can optionally be changed.

If one only wants to calculate the conditional power, then it is computationally more efficient to call the functions getStageResults and getConditionalPower instead. The code below illustrates this by plotting the conditional power curve depending on the true treatment effect. The dashed vertical lines in the plot correspond to the protocol hazard ratio of 0.74 and the observed interim hazard ratio of 0.8.

```
# get stage results so far
stageResults <- getStageResults(design, results, directionUpper = FALSE)</pre>
# calculate condition power for true HR ranging from 0.6 to 1
hr <- seq(0.6, 1, by = 0.01)
cpower <- rep(NA, length(hr))</pre>
for (i in 1:length(hr)) {
    cpower[i] <- getConditionalPower(stageResults,</pre>
        nPlanned = 370 - 245,
        thetaH1 = hr[i]
    )$conditionalPower[3]
}
# Alternatively, use the Vectorize() function for doing this:
cpowerVec <- Vectorize(function(x) {</pre>
    getConditionalPower(stageResults,
        nPlanned = 370 - 245,
        thetaH1 = x
    )$conditionalPower[3]
```

```
})
cpower <- cpowerVec(hr)
# Plot results
plot(hr, cpower,
    type = "1", xlab = "True hazard ratio",
    ylab = "Conditional power",
    lwd = 2, ylim = c(0, 1), axes = FALSE,
    main = "Conditional power after second interim analysis"
)
axis(1, at = seq(0.6, 1, by = 0.025))
axis(2, at = seq(0, 1, by = 0.1))
abline(v = seq(0.6, 1, by = 0.05), h = seq(0, 1, by = 0.1), col = gray(0.9))
abline(v = c(0.74, 0.8), lty = 2)</pre>
```

Conditional power after second interim analysis



System: rpact 3.3.2, R version 4.2.1 (2022-06-23 ucrt), platform: x86_64-w64-mingw32

To cite R in publications use:

R Core Team (2022). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Um Paket 'rpact' in Publikationen zu zitieren, nutzen Sie bitte:

Wassmer G, Pahlke F (2022). rpact: Confirmatory Adaptive Clinical Trial Design and Analysis. https://www.rpact.org, https://www.rpact.com/https://github.com/rpact-com/rpact.

