# GroupSeq: Designing clinical trials using group sequential designs

*by Roman Pahl, Andreas Ziegler, and Inke R. König*

## Introduction

Clinical trials are the gold standard to prove superiority of new interventions. An attractive variant to these studies is to use group sequential designs (Ziegler et al., 2003). Widely used since their introduction in the 1970s and 1980s, many different versions of group sequential designs have been developed. Common to all these designs is their complex computation for which software tools are required. In this context, existing software tools like ADDPLAN (`http://www.addplan.com/`) often are not free of charge, making it valuable to consider freely distributed software alternatives.

**GroupSeq** provides a set of several methods for computing group sequential boundaries of well established group sequential designs, among others the designs by Pocock (1977) and O'Brien and Fleming (1979). More precisely, the original designs are approximated by the *α-spending* function approach which was introduced by Lan and DeMets (1983) and Kim and DeMets (1987), for which a Fortran program by Reboussin et al. (2003) has been available. For **GroupSeq**, the latter implementation was completely recoded in R, and, in addition, new features were added. For instance, **GroupSeq** allows one to calculate exact Pocock bounds beside the estimated ones obtained by the *α-spending* approach.

Furthermore, as an important feature, this application is embedded in a graphical user interface (GUI) using the Tcl/Tk interface in the R **tcltk** package. Note that there is also a graphical user interface by Reboussin et al. (2003) although it only runs under Windows. In contrast, the **GroupSeq** GUI is available on several platforms, and, moreover, provides customization possibilities; i.e., the user may create as many windows as desired. Thus, he or she may perform multiple tasks such as computing and comparing several designs at the same time.

The computations yield valid results for any test which is based on normally distributed test statistics with independent increments, survival studies, and certain longitudinal designs. Using the *α*-spending function approach, interim analyses need not be equally spaced, and their number need not be specified in advance.

## Group sequential tests

This paper only gives a basic introduction to group sequential designs. For a comprehensive overview of this topic, see, e.g., Jennison and Turnbull (2000) and Wassmer (2001).

The classical theory of testing goes back to Neyman and Pearson (1928) and is based on a sample of fixed size. In this sample, the null hypothesis $H_0$ is tested against an alternative hypothesis $H_1$. A significance level $\alpha$ has to be defined *a priori*, which implies the probability of the null hypothesis being falsely rejected. Importantly, the evaluation always takes place at the end after *all* observations have been made.

By contrast, group sequential designs allow consecutive testing, with possible rejection of the null hypothesis after observation of each group in a sequence of groups. Therefore, the significance levels corresponding to each group have to be adjusted appropriately. Many different group sequential designs have been developed. The main difference among these is the manner of allocating the specific significance levels. A further distinction consists in equally sized groups as in classical group sequential testing (Pocock, 1977; O'Brien and Fleming, 1979) versus unequally sized groups (Kim and DeMets, 1987).

## Determining critical regions

Normally one wants to compare two population mean values $\mu_1$ and $\mu_2$—i.e., one wants to tests the null hypothesis $H_0 : \mu_1 = \mu_2$ against the alternative hypothesis $H_1 : \mu_1 \neq \mu_2$. Consider a standard normally distributed $Z$ statistic

$$Z = \frac{\overline{X}_1 - \overline{X}_2}{\sqrt{\sigma_1^2/n_1 + \sigma_2^2/n_2}} \qquad (1)$$

with $\overline{X}_1$, $\overline{X}_2$ being the means of two independent samples of sizes $n_1, n_2$ which are distributed as $N(\mu_1, \sigma_1^2)$ and $N(\mu_2, \sigma_2^2)$. Assuming $n_1 = n_2 = n$ and $\sigma_1^2 = \sigma_2^2 = \sigma^2$, analogously to (1), one can define a statistic *per group* $k$ :

$$Z_k = \frac{\overline{X}_{1k} - \overline{X}_{2k}}{\sigma}\sqrt{\frac{n_k}{2}}$$

The standardized overall statistic until group $k$ then is defined as

$$Z_k^* = \frac{\sum_{j=1}^k \sqrt{n_j} Z_j}{\sqrt{\sum_{j=1}^k n_j}}$$

In contrast to $Z_1, Z_2, ..., Z_K$, the $Z_1^*, Z_2^*, ..., Z_K^*$ are statistically dependent on each other. The joint multivariate normal distribution of $Z_1^*, Z_2^*, ..., Z_K^*$ therefore has to be computed numerically.

A specific group sequential design is characterized by its regions $\xi_k$ with $k$=1, 2, ..., $K$-1, where $\xi_k$ denotes the region in which the study continues; i.e., $H_0$ has not been rejected by then. The last *acceptance region* in the sequence is denoted $\xi_K$, and the probability of type I error for test $K$ is given by

$$1 - P_{H_0}\left(\bigcap_{k=1}^{K} \{Z_k^* \in \xi_k\}\right),$$

where $P_{H_0}$ denotes the distribution of $Z_k^*$ under $H_0$. One obtains the corresponding power with regard to the alternative hypothesis $H_1$ by

$$1 - P_{H_1}\left(\bigcap_{k=1}^{K} \{Z_k^* \in \xi_k\}\right).$$

## The $\alpha$-spending function approach

The *$\alpha$-spending function*, or *use function*, approach was introduced by Lan and DeMets (1983) and Kim and DeMets (1987). It allows analyses at arbitrary points of time in the study and hence of arbitrarily sized groups. This is achieved via the $\alpha$-spending function $\alpha^*(t_k)$ which determines the type I error "spent" until time point $t_k$, with $k$ denoting the $k$-th interim analysis. Usually, the entire period of analysis is standardized to one so that $0 < t_1 < t_2 < ... < t_K = 1$, with $t_k = \sum_{i=1}^{k} n_i/N$.

Given the maximum sample size $N$ and the $\alpha$-spending function $\alpha^*(t_k)$, the critical regions are obtained by recursive calculation. Setting $t_1 = n_1/N$, one obtains

$$P_{H_0}(|Z_1^*| \geq u_1) = \alpha^*(t_1) \qquad (2)$$

for the first critical region. Analogously to (2), the remaining probabilites are obtained by

$$P_{H_0}(\bigcap_{i=1}^{k-1} \{|Z_i^*| < u_i\} \cap |Z_k^*| \geq u_k) = \alpha^*(t_k) - \alpha^*(t_{k-1}).$$

The $\alpha$-spending function approach thus is a very flexible tool for designing various studies because neither group size nor number of interim analyses have to be specified in advance. All calculations performed by **GroupSeq** are based on the $\alpha$-spending approach. In this context, the classic designs are estimated as special cases by the following $\alpha$-spending functions:

- Pocock:
  $\alpha_1^*(t_k) = \alpha \ln[1 + (e-1)t_k]$
- O'Brien and Fleming:
  $\alpha_2^*(t_k) = 4\left\{1 - \Phi\left[\Phi^{-1}(1-\alpha/4)/\sqrt{t_k}\,\right]\right\}$

## Implementation

The program by Reboussin et al. (2003) constituted the basis for **GroupSeq**. Recoding in R followed the software engineering principles of unitization and re-usability; i.e., the tasks were divided into many subfunctions, which also allows improvement in further development. One goal consisted in maintaining the performance of the application. This was achieved by converting the "low level syntax" of Fortran into R specific functions as well as vector and matrix operations. Some algorithms also were slightly improved, e.g., by using Newton's method for optimization, replacing a bisection method. The interested reader is referred to the source code which is commented in detail. As it turned out, the performance of **GroupSeq** is generally comparable with the Fortran implementation. Notable differences occur only when the number of interim analyses rises to more than ten. This, however, will usually not occur in practice. Still, efficiency could be improved by outsourcing some computations into compiled C/C++ code.

A main focus was on improving the user interface. At the time of development of **GroupSeq**, the Fortran program only provided a command line interface. Because of this, **GroupSeq** was embedded in a GUI using Tcl/Tk, making it available for every platform on which R runs. The customization of the GUI is left to the user who may create a separate window for each task, which allows any arrangement and positioning on the desktop. For those who are familiar with the program by Reboussin et al. (2003), the menu structure of that program was retained in **GroupSeq**.

Additionally, **GroupSeq** provides new features that are not part of the program by Reboussin et al. (2003). Specifically, new functionality is included that allows the user to compute exact Pocock bounds.

## Package features

**GroupSeq** provides a GUI using the Tcl/Tk interface in the R **tcltk** package. Hence, **GroupSeq**'s usage is almost self-explanatory. After loading the package, the program starts automatically. The user will see the menu shown in Figure 1.

A task is chosen by selecting an appropriate line and confirming with "Perform Selected Task". For the purpose of multitasking, the user may open as many task windows as desired. Every task is processed independently. Choosing task "*-1- Compute Bounds*" will lead to the dialog box given in Figure 2.

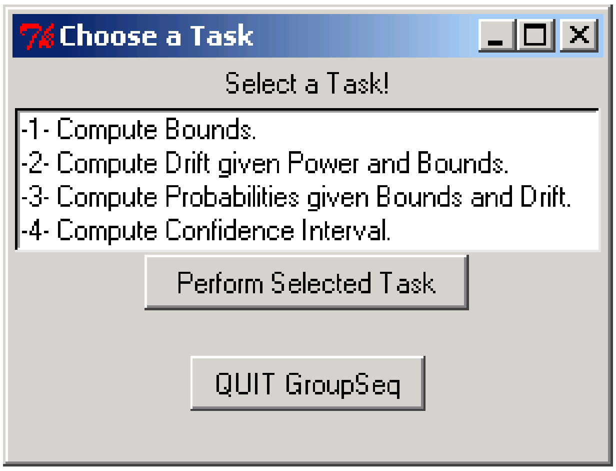
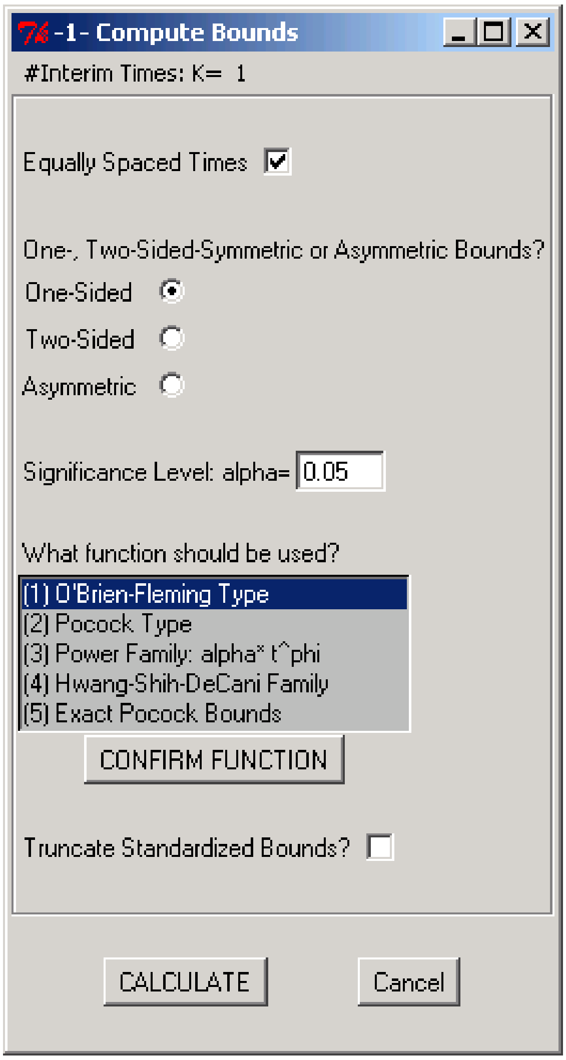

Choose a Task
Select a Task!
-1- Compute Bounds.
-2- Compute Drift given Power and Bounds.
-3- Compute Probabilities given Bounds and Drift.
-4- Compute Confidence Interval.
Perform Selected Task
QUIT GroupSeq

Figure 1: *Menu on program start.*

The number of interim analyses is selected by a drop down menu at the top of the window. An $\alpha$-spending function can be selected and must be confirmed by the "CONFIRM FUNCTION" button.


-1- Compute Bounds
One-, Two-Sided-Symmetric or Asymmetric Bounds?
What function should be used?
[1] O'Brien-Fleming Type
[2] Pocock Type
[3] Power Family: alpha* t^phi
[4] Hwang-Shih-DeCani Family
[5] Exact Pocock Bounds
CONFIRM FUNCTION
CALCULATE
Cancel

#Interim Times: K= 1

Equally Spaced Times ☑

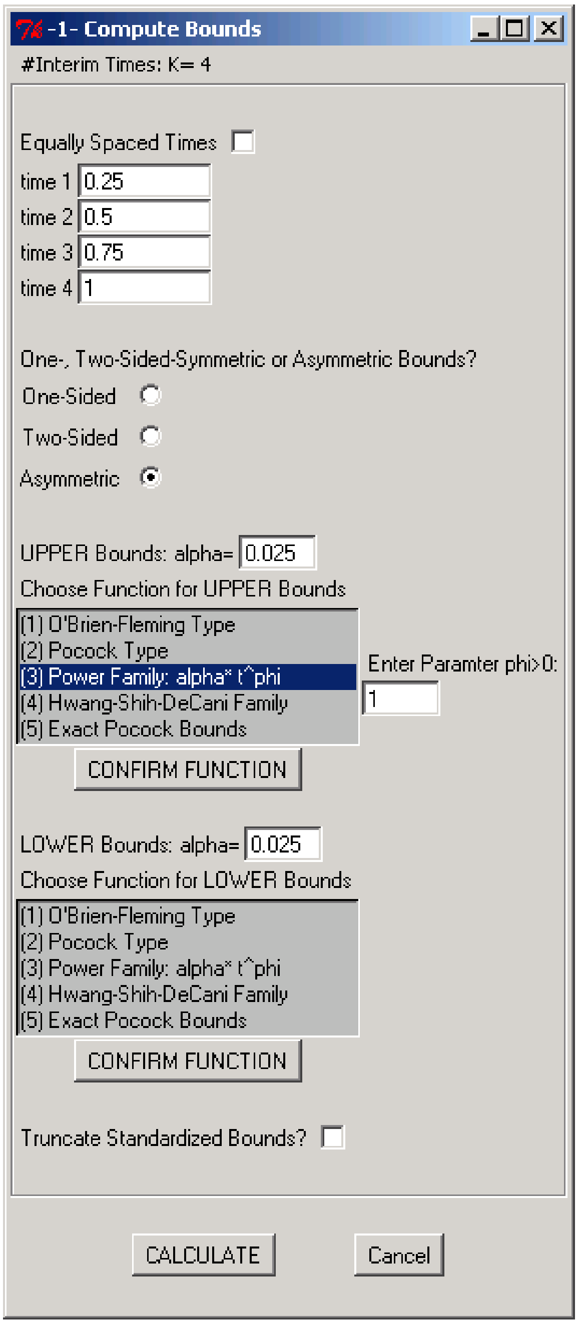
● One-Sided
○ Two-Sided
○ Asymmetric

Significance Level: alpha= 0.05

Truncate Standardized Bounds? □

Figure 2: *Window after "-1- Compute Bounds" was chosen.*

The user may choose between (1) and (2) which are estimates of the classic designs by Pocock (1977) and O'Brien and Fleming (1979), respectively. (3) corresponds to the design proposed by Kim and DeMets (1987), and (4) to a family of functions proposed by Hwang et al. (1990). Finally, (5) calculates the exact bounds for the Pocock design. All **GroupSeq** operating windows are designed so that irrelevant information is concealed. For example, if the user chooses to use unequal time points and different spending functions for upper and lower bounds, the window will appear as shown in Figure 3, with **GroupSeq** providing meaningful default values.


-1- Compute Bounds
One-, Two-Sided-Symmetric or Asymmetric Bounds?
Choose Function for UPPER Bounds
[1] O'Brien-Fleming Type
[2] Pocock Type
[3] Power Family: alpha* t^phi
[4] Hwang-Shih-DeCani Family
[5] Exact Pocock Bounds
CONFIRM FUNCTION
Choose Function for LOWER Bounds
[1] O'Brien-Fleming Type
[2] Pocock Type
[3] Power Family: alpha* t^phi
[4] Hwang-Shih-DeCani Family
[5] Exact Pocock Bounds
CONFIRM FUNCTION
CALCULATE
Cancel

#Interim Times: K= 4

Equally Spaced Times □

time 1 0.25
time 2 0.5
time 3 0.75
time 4 1

○ One-Sided
○ Two-Sided
● Asymmetric

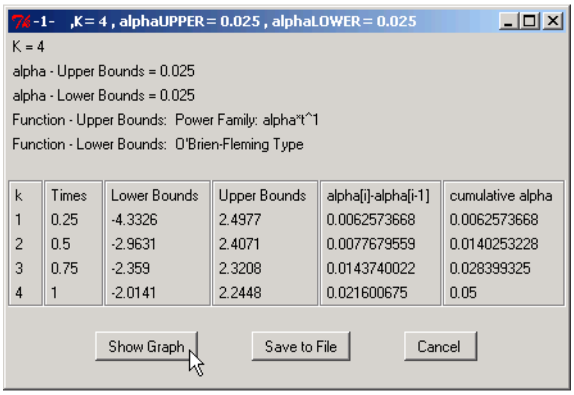
UPPER Bounds: alpha= 0.025

Enter Paramter phi>0: 1

LOWER Bounds: alpha= 0.025

Truncate Standardized Bounds? □

Figure 3: *User specification in "-1- Compute Bounds".*



-1- ,K= 4 , alphaUPPER= 0.025 , alphaLOWER= 0.025

K = 4
alpha - Upper Bounds = 0.025
alpha - Lower Bounds = 0.025
Function - Upper Bounds: Power Family: alpha*t^1
Function - Lower Bounds: O'Brien-Fleming Type

| k | Times | Lower Bounds | Upper Bounds | alpha[i]-alpha[i-1] | cumulative alpha |
|---|---|---|---|---|---|
| 1 | 0.25 | -4.3326 | 2.4977 | 0.0062573668 | 0.0062573668 |
| 2 | 0.5 | -2.9631 | 2.4071 | 0.0077679559 | 0.0140253228 |
| 3 | 0.75 | -2.359 | 2.3208 | 0.0143740022 | 0.028399325 |
| 4 | 1 | -2.0141 | 2.2448 | 0.021600675 | 0.05 |

Figure 4: *Displaying results of calculated bounds.*

Pressing the “CALCULATE” button at the bottom of the window initiates the calculation, and each of the results is shown in a new, separate window. Figure 4 gives an example. The parameter values that were used in the computation are shown at the top part of the window, and the most important parameters can also be seen in the blue window title bar. This feature helps the user to keep track of many windows when elaborating many designs at once. The results of the computation are presented in a table. In the case of task -1-, the resulting bounds, $\alpha$ spent per stage, and $\alpha$ spent in total are displayed. In a next step, the resulting critical bounds can be visualized by choosing “Show Graph” (see Figure 5) which requires the R graphics environment.

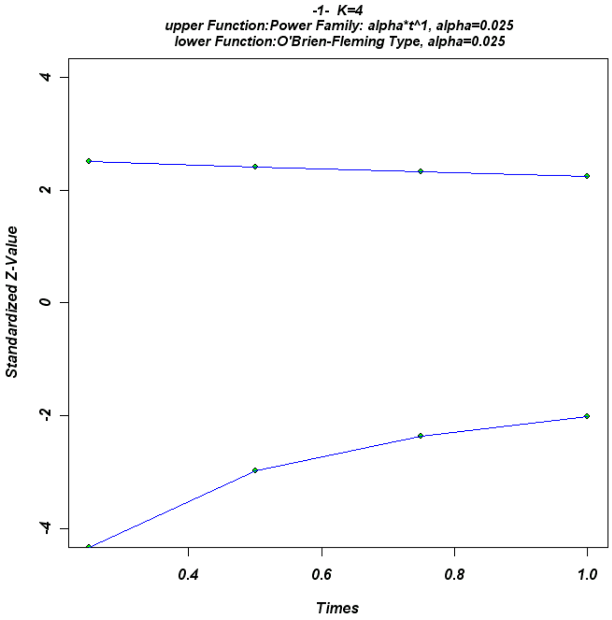
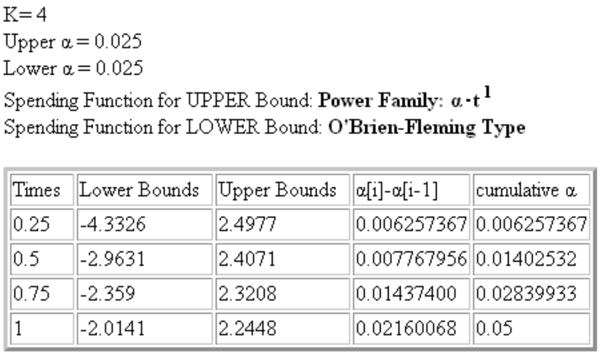


Figure 5: *Graphical visualization of results.*

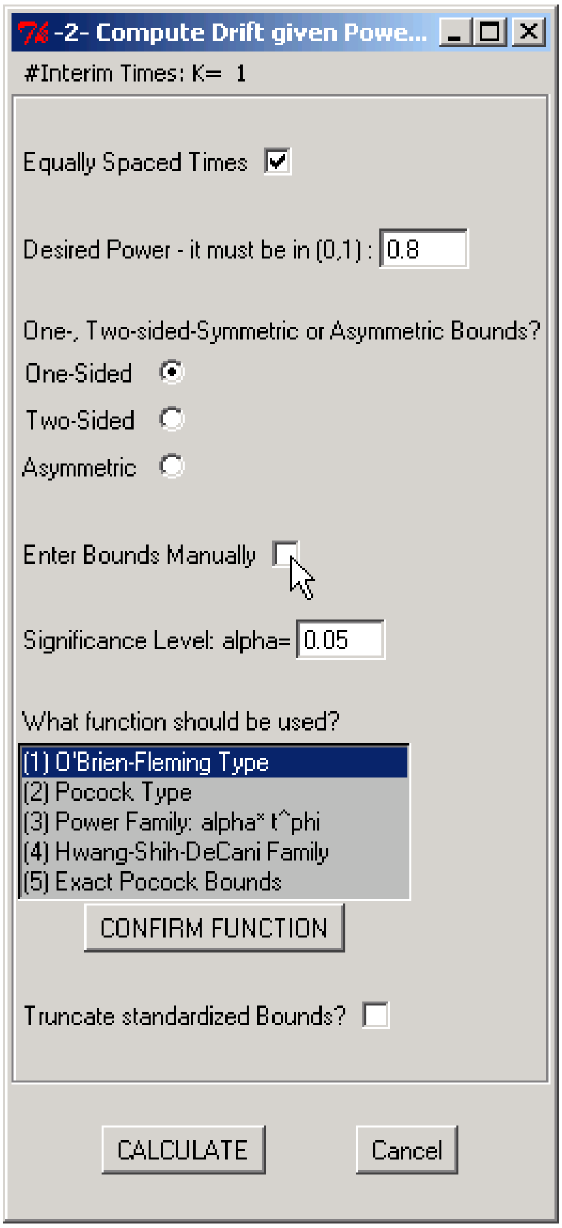
By using the option “Save to file”, the results may be saved into a table in common *.html-file format, which conserves meta information and therefore provides the possibility of further processing (see Figure 6).



K= 4
Upper $\alpha$ = 0.025
Lower $\alpha$ = 0.025
Spending Function for UPPER Bound: **Power Family: $\alpha \cdot t^1$**
Spending Function for LOWER Bound: **O'Brien-Fleming Type**

| Times | Lower Bounds | Upper Bounds | α[i]-α[i-1] | cumulative α |
|---|---|---|---|---|
| 0.25 | -4.3326 | 2.4977 | 0.006257367 | 0.006257367 |
| 0.5 | -2.9631 | 2.4071 | 0.007767956 | 0.01402532 |
| 0.75 | -2.359 | 2.3208 | 0.01437400 | 0.02839933 |
| 1 | -2.0141 | 2.2448 | 0.02160068 | 0.05 |

Figure 6: *Saved results (viewed in Mozilla Firefox).*



-2- Compute Drift given Powe...

#Interim Times: K= 1

Equally Spaced Times ☑

Desired Power - it must be in (0,1) : 0.8

One-, Two-sided-Symmetric or Asymmetric Bounds?
One-Sided ●
Two-Sided ○
Asymmetric ○

Enter Bounds Manually □

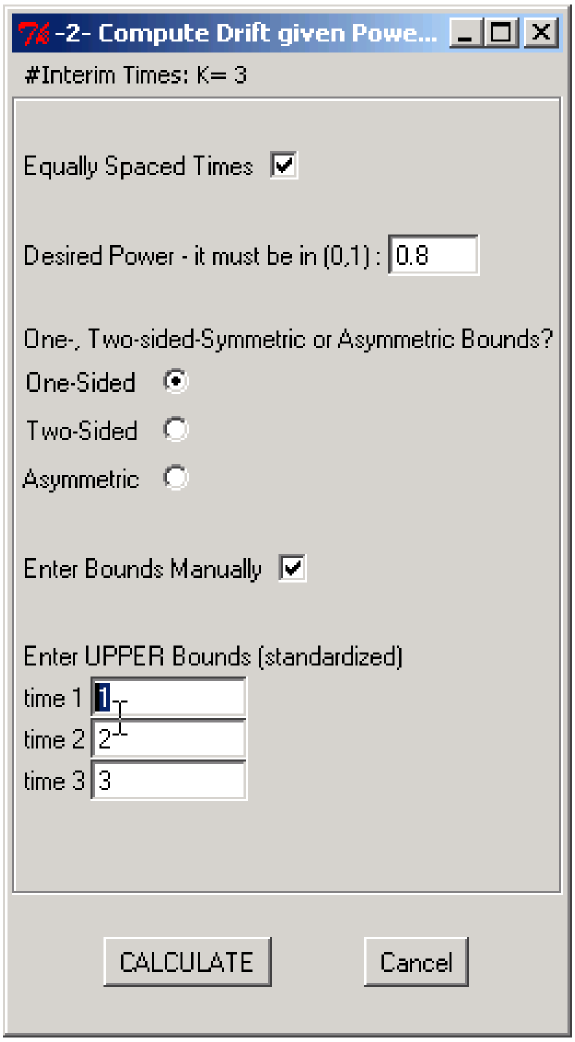
Significance Level: alpha= 0.05

What function should be used?
(1) O'Brien-Fleming Type
(2) Pocock Type
(3) Power Family: alpha* t^phi
(4) Hwang-Shih-DeCani Family
(5) Exact Pocock Bounds

Truncate standardized Bounds? □

Figure 7: *“-2- Compute drift given power and bounds” was chosen.*

Choosing task “*-2- Compute drift given power and bounds*” from the main menu will lead to the window shown in Figure 7. Besides specifying the desired power, the user may enter specific bounds instead of computing them with a spending function. Activating the corresponding check box changes the current window as shown in Figure 8. Calculating the effect under the alternative hypothesis leads to the window shown in Figure 9.


-2- Compute Drift given Powe...

#Interim Times: K= 3

Equally Spaced Times ☑

Desired Power - it must be in (0,1) : 0.8

One-, Two-sided-Symmetric or Asymmetric Bounds?

One-Sided ●

Two-Sided ○

Asymmetric ○

Enter Bounds Manually ☑

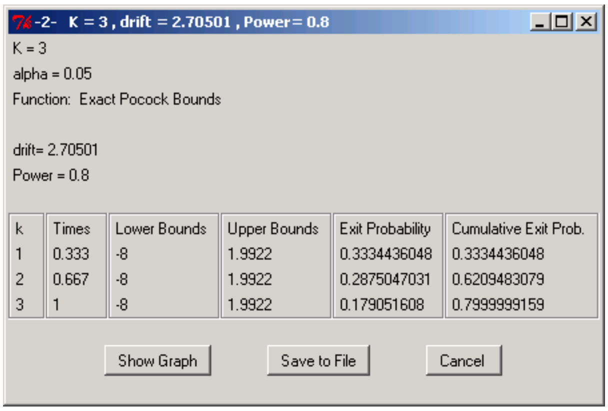
Enter UPPER Bounds (standardized)

time 1 1

time 2 2

time 3 3

Figure 8: *User enters specific bounds.*


-2- K = 3 , drift = 2.70501 , Power= 0.8

K = 3

alpha = 0.05

Function: Exact Pocock Bounds

drift= 2.70501

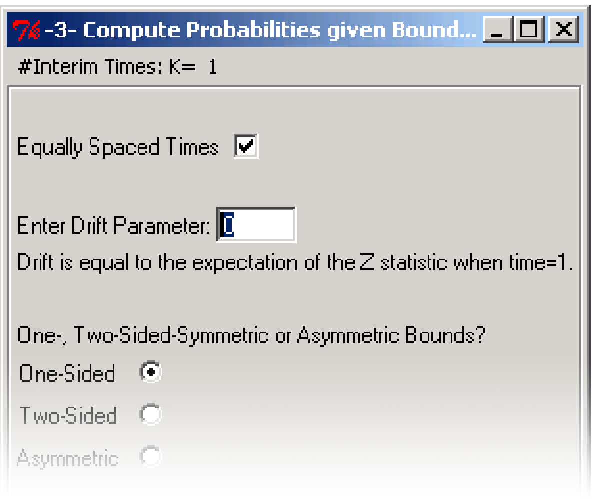
Power = 0.8

| k | Times | Lower Bounds | Upper Bounds | Exit Probability | Cumulative Exit Prob. |
|---|---|---|---|---|---|
| 1 | 0.333 | -8 | 1.9922 | 0.3334436048 | 0.3334436048 |
| 2 | 0.667 | -8 | 1.9922 | 0.2875047031 | 0.6209483079 |
| 3 | 1 | -8 | 1.9922 | 0.179051608 | 0.7999999159 |

Figure 9: *Displaying results of calculated drift under re-specified power (here: Exact Pocock Bounds were used).*

The table contains exit probabilities (i.e., $1 - \beta$) corresponding to the probability per stage of rejecting the null hypothesis under the alternative. The effect (drift) hereby is implicitly given by the pre-specified power. The cumulative exit probabilities, in turn, sum up to the power at last analysis.

Task “*-3- Compute drift given power and bounds*” works the other way around. Specifying a certain drift, which, again, corresponds to the expected effect size at last analysis (see Figure 10), exit probabilities, resulting in the overall power at last analysis, are calculated. The computation, of course, also depends on the given bounds and interim times.


-3- Compute Probabilities given Bound...

#Interim Times: K= 1

Equally Spaced Times ☑

Enter Drift Parameter: 0

Drift is equal to the expectation of the Z statistic when time=1.

One-, Two-Sided-Symmetric or Asymmetric Bounds?

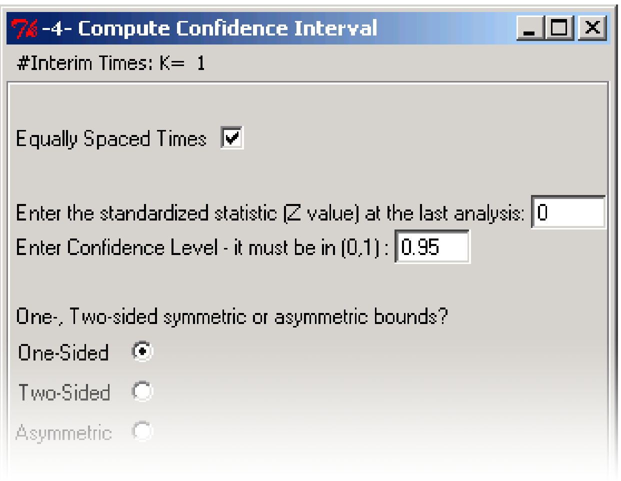
One-Sided ●

Two-Sided ○

Asymmetric ○

Figure 10: *User enters drift in Task -3- window (detail).*

Finally, users may select task “*-4- Compute confidence interval*”. As shown in Figure 11, the confidence level and the overall effect (as a $Z$-value) have to be entered.


-4- Compute Confidence Interval

#Interim Times: K= 1

Equally Spaced Times ☑

Enter the standardized statistic (Z value) at the last analysis: 0

Enter Confidence Level - it must be in (0,1) : 0.95

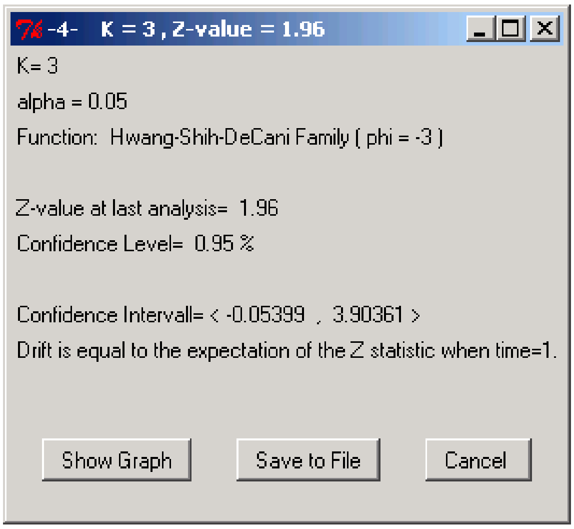
One-, Two-sided symmetric or asymmetric bounds?

One-Sided ●

Two-Sided ○

Asymmetric ○

Figure 11: *User enters confidence level and desired effect in Task -4- window (detail).*



K= 3

alpha = 0.05

Function: Hwang-Shih-DeCani Family ( phi = -3 )

Z-value at last analysis= 1.96

Confidence Level= 0.95 %

Confidence Intervall= < -0.05399 , 3.90361 >

Drift is equal to the expectation of the Z statistic when time=1.

Figure 12: *Resulting confidence interval (Two-Sided-Bounds).*

The window displaying the calculated confidence interval differs somewhat from the other windows (see Figure 12), as no table is presented. The bounds still can be visualized using "Show Graph", however.

## Future development

Future versions of **GroupSeq** will include additional software ergonomic improvements such as a status bar during calculation. Additionally, an "undo" function will be added, as well as a feature to let users specify their own default values, thereby achieving more efficiency in frequent use of **GroupSeq**. Furthermore, it is planned to implement *adaptive* group sequential designs.

## Bibliography

I. K. Hwang, W. J. Shih, and J. S. DeCani. Group sequential designs using a family of type I error probability spending functions. *Statistics in Medicine*, 9:1439–1445, 1990.

C. Jennison and B. W. Turnbull. *Group Sequential Methods with Applications to Clinical Trials*. Chapman & Hall/CRC, 2000.

K. Kim and D. DeMets. Design and analysis of group sequential tests based on the type I error spending rate function. *Biometrika*, 74:25–36, 1987.

K. K. G. Lan and D. L. DeMets. Discrete sequential boundaries for clinical trials. *Biometrika*, 70:659–663, 1983.

J. Neyman and E. S. Pearson. On the use and interpretation of certain test criteria for purposes of statistical interference. *Biometrika*, 20A:263–295, 1928.

P. C. O'Brien and T. R. Fleming. A multiple testing procedure for clinical trials. *Biometrics*, 35:549–556, 1979.

S. J. Pocock. Group sequential methods in the design and analysis of clinical trials. *Biometrika*, 64:191–199, 1977.

D. M. Reboussin, D. L. DeMets, K. Kim, and K. K. G. Lan. Programs for Computing Group Sequential Boundaries Using the Lan-DeMets Method, Version 2.1. 2003. URL http://www.biostat.wisc.edu/landemets/.

G. Wassmer. *Statistische Testverfahren für gruppensequentielle und adaptive Pläne in klassischen Studien*. Alexander Mönch Verlag, Köln, 2001.

A. Ziegler, G. Dahmen, I. R. König, G. Wassmer, P. Hildebrand, and M. Birth. Accelerating clinical trials by flexible designs. *Good Clinical Practice Journal*, 10:16–19, 2003.

*Roman Pahl*
roman.pahl@gmx.de
*Andreas Ziegler*
ziegler@imbs.uni-luebeck.de
*Inke R. König*
Inke.Koenig@imbs.uni-luebeck.de
*Institute of Medical Biometry and Statistics*
*University at Lübeck, Germany*

# Using R/Sweave in everyday clinical practice

*by Sven Garbade and Peter Burgard*

The Sweave package by Friedrich Leisch is a powerful tool to combine R with LaTeX facilities for text formatting (R Development Core Team, 2005; Leisch, 2002a,b). Sweave allows the dynamic generation of statistical reports by using literate data analysis. A further application of Sweave is the package vignette (Leisch, 2003). An introduction to the abilities of Sweave can be found on Friedrich Leisch's Home-