Package 'seqDesign'

May 23, 2019

Version 1.2

Date 2019-05-14

Title Simulation and Group Sequential Monitoring of Randomized
Two-Stage Treatment Efficacy Trials with Time-to-Event
Endpoints

URL https://github.com/mjuraska/seqDesign

Description A modification of the preventive vaccine efficacy trial de-
sign of Gilbert, Grove et al. (2011, Statistical Communications in Infectious Diseases) is imple-
mented, with application generally to individual-randomized clinical trials with multiple ac-
tive treatment groups and a shared control group, and a study endpoint that is a time-to-
event endpoint subject to right-censoring. The design accounts for the issues that the effi-
cacy of the treatment/vaccine groups may take time to accrue while the multiple treatment ad-
ministrations/vaccinations are given; there is interest in assessing the durability of treatment effi-
cacy over time; and group sequential monitoring of each treatment group for poten-
tial harm, non-efficacy/efficacy futility, and high efficacy is warranted. The design di-
vides the trial into two stages of time periods, where each treatment is first evaluated for effi-
cacy in the first stage of follow-up, and, if and only if it shows significant treatment effi-
cacy in stage one, it is evaluated for longer-term durability of efficacy in stage two. The pack-
age produces plots and tables describing operating characteristics of a specified design includ-
ing an unconditional power for intention-to-treat and per-protocol/as-treated analyses; trial dura-
tion; probabilities of the different possible trial monitoring outcomes (e.g., stop-
ping early for non-efficacy); unconditional power for comparing treatment efficacies; and distri-
butions of numbers of endpoint events occurring after the treatments/vaccinations are given, use-
ful as input parameters for the design of studies of the association of biomarkers with a clini-
cal outcome (surrogate endpoint problem). The code can be used for a single active treat-
ment versus control design and for a single-stage design.

BugReports https://github.com/mjuraska/seqDesign/issues

Depends R (>= 2.16), survival

License GPL-2

Encoding UTF-8

LazyLoad yes

VignetteBuilder knitr, R.rsp

Suggests knitr, R.rsp

RoxygenNote 6.1.1

NeedsCompilation no

Author Michal Juraska [aut, cre],
    Doug Grove [aut],
    Xuesong Yu [ctb],
    Peter Gilbert [ctb],
    Stephanie Wu [ctb]

Maintainer Michal Juraska <mjuraska@fredhutch.org>

Repository CRAN

Date/Publication 2019-05-22 22:10:35 UTC

R topics documented:

---

censTrial	*Generation of Pre-Unblinded Follow-Up Data-Sets by Applying the Monitoring Outcomes*

Description

censTrial 'correctly censors' treatment arms in data-sets generated by simTrial by including pre-unblinded follow-up data only according to the monitoring conclusions as reported by monitorTrial.

Usage

```
censTrial(dataFile, monitorFile, stage1, stage2, saveFile = NULL,
  saveDir = NULL, verbose = TRUE)
```

Arguments

dataFile	if saveDir = NULL, a list returned by simTrial; otherwise a name (character string) of an .RData file created by simTrial
monitorFile	if saveDir = NULL, a list returned by monitorTrial; otherwise a name (character string) of an .RData file created by monitorTrial
stage1	the final week of stage 1 in a two-stage trial
stage2	the final week of stage 2 in a two-stage trial, i.e., the maximum follow-up time

saveFile	a character string specifying the name of the output .RData file. If NULL (default), a default file name will be used.
saveDir	a character string specifying a path for both dataFile and monitorFile. If supplied, the output is also saved as an .RData file in this directory; otherwise the output is returned as a list.
verbose	a logical value indicating whether information on the output directory and file name should be printed out (default is TRUE)

Details

All time variables use week as the unit of time. Month is defined as 52/12 weeks.

The following censoring rules are applied to each data-set generated by simTrial:

- If no vaccine arm registers efficacy or high efficacy in Stage 1, the placebo arm is censored on the date when the last vaccine arm hits the harm or non-efficacy boundary.
- If a vaccine arm hits the harm boundary, censor the arm immediately.
- If a vaccine arm hits the non-efficacy boundary, censor the arm on the earliest date of the two events: (1) the last vaccine arm hits the harm or non-efficacy boundary (if applicable); and (2) all subjects in the vaccine arm have completed the final stage1 visit.

Value

If saveDir is specified, the output list (named trialListCensor) is saved as an .RData file in saveDir (the path to saveDir is printed); otherwise it is returned. The output object is a list of length equal to the number of simulated trials, each of which is a data.frame with at least the variables trt, entry, exit, and event storing the treatment assignments, enrollment times, correctly censored study exit times, and event indicators, respectively. If available, indicators belonging to the per-protocol cohort (named pp1, pp2, etc.) are copied from the uncensored data-sets.

See Also

simTrial, monitorTrial, and rankTrial

Examples

```
simData <- simTrial(N=c(1000, rep(700, 2)), aveVE=seq(0, 0.4, by=0.2),
                    VEmodel="half", vePeriods=c(1, 27, 79), enrollPeriod=78,
                    enrollPartial=13, enrollPartialRelRate=0.5, dropoutRate=0.05,
                    infecRate=0.04, fuTime=156,
                    visitSchedule=c(0, (13/3)*(1:4), seq(13*6/3, 156, by=13*2/3)),
                    missVaccProb=c(0,0.05,0.1,0.15), VEcutoffWeek=26, nTrials=5,
                    stage1=78, randomSeed=300)

monitorData <- monitorTrial(dataFile=simData, stage1=78, stage2=156,
                            harmMonitorRange=c(10,100), alphaPerTest=NULL,
                            nonEffStartMethod="FKG", nonEffInterval=20,
                            lowerVEnoneff=0, upperVEnoneff=0.4, highVE=0.7,
                            stage1VE=0, lowerVEuncPower=0, alphaNoneff=0.05,
                            alphaHigh=0.05, alphaStage1=0.05,
```

```
                                    alphaUncPower=0.05, estimand="cuminc", lagTime=26)

   censData <- censTrial(dataFile=simData, monitorFile=monitorData, stage1=78, stage2=156)

   ### alternatively, to save the .RData output file (no '<-' needed):
   ###
   ### simTrial(N=c(1400, rep(1000, 2)), aveVE=seq(0, 0.4, by=0.2), VEmodel="half",
   ###          vePeriods=c(1, 27, 79), enrollPeriod=78, enrollPartial=13,
   ###          enrollPartialRelRate=0.5, dropoutRate=0.05, infecRate=0.04, fuTime=156,
   ###          visitSchedule=c(0, (13/3)*(1:4), seq(13*6/3, 156, by=13*2/3)),
   ###          missVaccProb=c(0,0.05,0.1,0.15), VEcutoffWeek=26, nTrials=30,
   ###          stage1=78, saveDir="./", randomSeed=300)
   ###
   ### monitorTrial(dataFile=
   ###              "simTrial_nPlac=1400_nVacc=1000_1000_aveVE=0.2_0.4_infRate=0.04.RData",
   ###              stage1=78, stage2=156, harmMonitorRange=c(10,100), alphaPerTest=NULL,
   ###              nonEffStartMethod="FKG", nonEffInterval=20, lowerVEnoneff=0,
   ###              upperVEnoneff=0.4, highVE=0.7, stage1VE=0, lowerVEuncPower=0,
   ###              alphaNoneff=0.05, alphaHigh=0.05, alphaStage1=0.05, alphaUncPower=0.05,
   ###              estimand="cuminc", lagTime=26, saveDir="./")
   ###
   ### censTrial(dataFile=
   ###          "simTrial_nPlac=1400_nVacc=1000_1000_aveVE=0.2_0.4_infRate=0.04.RData",
   ###          monitorFile=
   ###        "monitorTrial_nPlac=1400_nVacc=1000_1000_aveVE=0.2_0.4_infRate=0.04_cuminc.RData",
   ###          stage1=78, stage2=156, saveDir="./")
```

---

getBlockSize   *Determine block size for use in blocked randomization*

Description

getBlockSize returns the minimum block size (possibly within a specified range) that is compatible with a trial's overall treatment assignment totals.

Usage

```
getBlockSize(nvec, range = c(0, Inf))
```

Arguments

nvec   vector specifying the number of participants to be assigned to each treatment
                group. The vector should have one component per group, so that its length
                equals number of groups. The sum of nvec should equal the total enrollment for
                the trial.

range   (Optional) vector of length two giving the lower and upper bounds (respectively)
                on block sizes that the user wishes to consider.

Details

The ordering of the components of nvec is not important, so using nvec = c(x,y,z) will produce the same results as using nvec = c(z,x,y).

In block randomization one does not necessarily want the smallest block size, which is the reason for the existance of the range argument. For example, a trial with a 1:1 randomization allocation between two groups would have a minimum block size of 2, which most people would consider to be too small. So a typical usage of getBlockSize would be to use range to set a minimum acceptable block size, through use of vector of form c(lowerBound, Inf). A large trial should probably have a block size on the order of 10-20 or larger, depending on factors including the total trial size and speed of enrollment, so setting a minimum is a good idea.

Value

An integer or NA. If the user does not specify range, then the function will always return an integer, which is the smallest block size compatible with the specified vector of treatment group sizes. If the user *has* specified the range, then the function adds the further constraint that the block size must lie in the closed interval given by range (i.e., the block size must be greater-than-or-equal-to range[1] and less-than-or-equal-to range[2]). If there are no compatible block sizes that lie in the given interval, then an NA is returned.

Note that the value returned is the **minimum** block size that is compatible, not necessarily the only one. Any other compatible block sizes (if any exist) will be integer multiples of the minimum size. You can check the feasibility of various integer multiples by seeing if they divide evenly into the total trial size (i.e., into the sum of nvec).

Examples

```
getBlockSize(nvec = c(375, 375) )
## specify a minimum block size of 10 (no maximum)
getBlockSize(nvec = c(375, 375), range = c(10, Inf) )

getBlockSize( nvec = c(30, 510, 390) )
## require a minimum block size of 10 and maximum of 30
## (not possible with this nvec, so function returns NA)
getBlockSize( nvec = c(30, 510, 390), range = c(10, 30) )
```

---

monitorTrial   *Group Sequential Monitoring of Simulated Efficacy Trials for the Event of Potential Harm, Non-Efficacy, and High Efficacy*

Description

monitorTrial applies a group sequential monitoring procedure to data-sets generated by simTrial, which may result in modification or termination of each simulated trial.

Usage

```
monitorTrial(dataFile, stage1, stage2, harmMonitorRange,
  harmMonitorAlpha = 0.05, alphaPerTest = NULL,
  nonEffStartMethod = c("FKG", "fixed", "?", "old"),
  nonEffStartParams = NULL, nonEffInterval,
  nonEffIntervalUnit = c("counts", "time"), lowerVEnoneff = NULL,
  upperVEnoneff, highVE, stage1VE, lowerVEuncPower = NULL, alphaNoneff,
  alphaHigh, alphaStage1, alphaUncPower = NULL,
  estimand = c("combined", "cox", "cuminc"), laggedMonitoring = FALSE,
  lagTime, saveFile = NULL, saveDir = NULL, verbose = TRUE)
```

Arguments

dataFile   if saveDir = NULL, a list returned by simTrial; otherwise a name (character string) of an .RData file created by simTrial

stage1   the final week of stage 1 in a two-stage trial

stage2   the final week of stage 2 in a two-stage trial, i.e., the maximum follow-up time

harmMonitorRange

a 2-component numeric vector specifying the range of the pooled number of infections (pooled over the placebo and vaccine arm accruing infections the fastest) over which the type I error rate, specified in harmMonitorAlpha, will be spent (per vaccine arm). Note that harmMonitorRange does not specify a range for which potential-harm stopping boundaries will be computed; instead, it specifies when potential-harm monitoring will start, and the range over which harmMonitorAlpha will be spent.

harmMonitorAlpha

a numeric value (0.05 by default) specifying the overall type I error rate for potential-harm monitoring (per vaccine arm). To turn off potential-harm monitoring, set harmMonitorAlpha equal to 0.00001.

alphaPerTest   a per-test nominal/unadjusted alpha level for potential-harm monitoring. If NULL, a per-test alpha level is calculated that yields a cumulative alpha of harmMonitorAlpha at the end of harmMonitorRange.

nonEffStartMethod

a character string specifying the method used for determining when non-efficacy monitoring is to start. The default method of Freidlin, Korn, and Gray (2010) ("FKG") calculates the minimal pooled infection count (pooled over the placebo and vaccine arm accruing infections the fastest) such that a hazard-ratio-based VE point estimate of 0% would result in declaring non-efficacy, i.e., the upper bound of the two-sided (1-alphaNoneff) x 100% confidence interval for VE based on the asymptotic variance of the log-rank statistic is (barely) below the non-efficacy threshold specified as component upperVEnonEff in the list nonEffStartParams. If this list component is left unspecified, the argument upperVEnonEff is used as the non-efficacy threshold. The alternative method ("fixed") starts non-efficacy monitoring at a fixed pooled infection count (pooled over the placebo and vaccine arm accruing infections the fastest) specified by component N1 in the list nonEffStartParams.

nonEffStartParams

a list with named components specifying parameters required by nonEffStartMethod (NULL by default)

nonEffInterval   a numeric value (a number of infections or a number of weeks) specifying the interval between two adjacent non-efficacy interim analyses

nonEffIntervalUnit

a character string specifying whether intervals between two adjacent non-efficacy interim analyses should be event-driven (default option "counts") or calendar time-driven (option "time")

lowerVEnoneff   specifies criterion 1 for declaring non-efficacy: the lower bound of the two-sided (1-alphaNoneff) x 100% confidence interval(s) for the VE estimand(s) lie(s) below lowerVEnoneff (typically set equal to 0). If NULL (default), this criterion is ignored.

upperVEnoneff   specifies criterion 2 for declaring non-efficacy: the upper bound of the two-sided (1-alphaNoneff) x 100% confidence interval(s) for the VE estimand(s) lie(s) below upperVEnoneff (typically a number in the 0–0.5 range)

highVE   specifies a criterion for declaring high-efficacy: the lower bound of the two-sided (1-alphaHigh) x 100% confidence interval for the VE estimand lies above highVE (typically a number in the 0.5–1 range). To turn off high efficacy monitoring, set highVE equal to 1.

stage1VE   specifies a criterion for advancement of a treatment's evaluation into Stage 2: the lower bound of the two-sided (1-alphaStage1) x 100% confidence interval for the VE estimand lies above stage1VE (typically set equal to 0)

lowerVEuncPower

a numeric vector with each component specifying a one-sided null hypothesis H0: VE(0–stage1) ≤ lowerVEuncPower x 100%. Unconditional power (i.e., accounting for sequential monitoring) to reject each H0 is calculated, where the rejection region is defined by the lower bound of the two-sided (1-alphaUncPower) x 100% confidence interval for the VE estimand being above the respective component of lowerVEuncPower (typically values in the 0–0.5 range).

alphaNoneff   one minus the nominal confidence level of the two-sided confidence interval used for non-efficacy monitoring

alphaHigh   one minus the nominal confidence level of the two-sided confidence interval used for high efficacy monitoring

alphaStage1   one minus the nominal confidence level of the two-sided confidence interval used for determining whether a treatment's evaluation advances into Stage 2

alphaUncPower   one minus the nominal confidence level of the two-sided confidence interval used to test one-sided null hypotheses H0: VE(0-stage1) ≤ lowerVEuncPower x 100% against alternative hypotheses H1: VE(0–stage1) > lowerVEuncPower x 100%. The same nominal confidence level is applied for each component of lowerVEuncPower.

estimand   a character string specifying the choice of VE estimand(s) used in non- and high efficacy monitoring, advancement rule for Stage 2, and unconditional power calculations. Three options are implemented: (1) the 'pure' Cox approach ("cox"), where VE is defined as 1-hazard ratio (treatment/control) and estimated by the

maximum partial likelihood estimator in the Cox model; (2) the 'pure' cumulative incidence-based approach (`"cuminc"`), where VE is defined as 1-cumulative incidence ratio (treatment/control) and estimated by the transformation of the Nelson-Aalen estimator for the cumulative hazard function; and (3) the combined approach (`"combined"`), where both aforementioned VE estimands are used for non-efficacy monitoring while the cumulative VE estimand is used for all other purposes. Only the first three characters are necessary.

laggedMonitoring
a logical value (`FALSE` by default) indicating whether "per-protocol" non-efficacy monitoring should additionally be conducted for events occurring after `lagTime` weeks as a more conservative non-efficacy monitoring approach. If `TRUE` and `estimand = "combined"`, the cumulative VE estimand is considered only for non-efficacy monitoring.

lagTime
a time point (in weeks) defining the per-protocol VE estimand, i.e., VE(lagTime–stage1). This VE estimand is also used in "per-protocol" non-efficacy monitoring if `laggedMonitoring` equals `TRUE`. It is typically chosen as the date of the last immunization or the date of the visit following the last immunization.

saveFile
a character string specifying the name of the output `.RData` file. If `NULL` (default), a default file name will be used.

saveDir
a character string specifying a path for `dataFile`. If supplied, the output is also saved as an `.RData` file in this directory; otherwise the output is returned as a list.

verbose
a logical value indicating whether information on the output directory, file name, and monitoring outcomes should be printed out (default is `TRUE`)

Details

All time variables use week as the unit of time. Month is defined as 52/12 weeks.

Potential harm monitoring starts at the `harmMonitorRange[1]`-th infection pooled over the placebo group and the vaccine regimen that accrues infections the fastest. The potential harm analyses continue at each additional infection until the first interim analysis for non-efficacy. The monitoring is implemented with exact one-sided binomial tests of H0: $p \leq p0$ versus H1: $p > p0$, where p is the probability that an infected participant was assigned to the vaccine group, and $p0$ is a fixed constant that represents the null hypothesis that an infection is equally likely to be assigned vaccine or placebo. Each test is performed at the same prespecified nominal/unadjusted alpha-level (`alphaPerTest`), chosen based on simulations such that, for each vaccine regimen, the overall type I error rate by the `harmMonitorRange[2]`-th arm-pooled infection (i.e., the probability that the potential harm boundary is reached when the vaccine is actually safe, $p = p0$) equals `harmMonitorAlpha`.

Non-efficacy is defined as evidence that it is highly unlikely that the vaccine has a beneficial effect measured as VE(0–stage1) of upperVEnoneff x 100% or more. The non-efficacy analyses for each vaccine regimen will start at the first infection (pooled over the vaccine and placebo arm) determined by `nonEffStartMethod`. Stopping for non-efficacy will lead to a reported two-sided (1-alphaNoneff) x 100% CI for VE(0–stage1) with, optionally, the lower confidence bound below lowerVEnoneff and the upper confidence bound below upperVEnoneff, where `estimand` determines the choice of the VE(0–stage1) estimand. This approach is similar to the inefficacy

monitoring approach of Freidlin, Korn, and Gray (2010). If estimand = "combined", stopping for non-efficacy will lead to reported (1-alphaNoneff) x 100% CIs for both VE parameters with, optionally, lower confidence bounds below lowerVEnoneff and upper confidence bounds below upperVEnoneff. If laggedMonitoring = TRUE, stopping for non-efficacy will lead to reported (1-alphaNoneff) x 100% CIs for both VE(0–stage1) and VE(lagTime–stage1) with, optionally, lower confidence bounds below lowerVEnoneff and upper confidence bounds below upperVEnoneff.

High efficacy monitoring allows early detection of a highly protective vaccine if there is evidence that VE(0–stage2) > highVE x 100%. It is synchronized with non-efficacy monitoring during Stage 1, and a single high-efficacy interim analysis during Stage 2 is conducted halfway between the end of Stage 1 and the end of the trial. While monitoring for potential harm and non-efficacy restricts to stage1 infections, monitoring for high efficacy counts all infections during stage1 or stage2, given that early stopping for high efficacy would only be warranted under evidence for durability of the efficacy.

The following principles and rules are applied in the monitoring procedure:

- Exclude all follow-up data from the analysis post-unblinding (and include all data pre-unblinding).
- The monitoring is based on modified ITT analysis, i.e., all subjects documented to be free of the study endpoint at baseline are included and analyzed according to the treatment assigned by randomization, ignoring how many vaccinations they received (only pre-unblinding follow-up included).
- If a vaccine hits the harm boundary, immediately discontinue vaccinations and accrual into this vaccine arm, and unblind this vaccine arm (continue post-unblinded follow-up until the end of Stage 1 for this vaccine arm).
- If a vaccine hits the non-efficacy boundary, immediately discontinue vaccinations and accrual into this vaccine arm, keep blinded and continue follow-up until the end of Stage 1 for this vaccine arm.
- If and when the last vaccine arm hits the non-efficacy (or harm) boundary, discontinue vaccinations and accrual into this vaccine arm, and unblind (the trial is over, completed in Stage 1).
- Stage 1 for the whole trial is over on the earliest date of the two events: (1) all vaccine arms have hit the harm or non-efficacy boundary; and (2) the last enrolled subject in the trial reaches the final stage1 visit.
- Continue blinded follow-up until the end of Stage 2 for each vaccine arm that reaches the end of stage1 with a positive efficacy (as defined by stage1VE) or high efficacy (as defined by highVE) result.
- If at least one vaccine arm reaches the end of stage1 with a positive efficacy or high efficacy result, continue blinded follow-up in the placebo arm until the end of Stage 2.
- Stage 2 for the whole trial is over on the earliest date of the two events: (1) all subjects in the placebo arm and each vaccine arm that registered efficacy or high efficacy in stage1 have failed or been censored; and (2) all subjects in the placebo arm and each vaccine arm that registered efficacy or high efficacy in stage1 have completed the final stage2 visit.

The above rules have the following implications:

- If a vaccine hits the non-efficacy boundary but Stage 1 for the whole trial is not over, then one includes in the analysis all follow-up through the final stage1 visit for that vaccine regimen,

including all individuals accrued up through the date of hitting the non-efficacy boundary (which will be the total number accrued to this vaccine arm).

- If a vaccine hits the harm boundary, all follow-up information through the date of hitting the harm boundary is included for this vaccine; no follow-up data are included after this date.
- If and when the last vaccine arm hits the non-efficacy (or harm) boundary, all follow-up information through the date of hitting the non-efficacy (or harm) boundary is included for this vaccine; no follow-up data are included after this date.

Value

If saveDir (and, optionally saveFile) is specified, the output list (named out) is saved as an .RData file in saveDir (the path to saveDir is printed); otherwise it is returned. The output object is a list of length equal to the number of simulated trials, each of which is a list of length equal to the number of treatment arms, each of which is a list with (at least) the following components:

- boundHit: a character string stating the monitoring outcome in this treatment arm, i.e., one of "Harm", "NonEffInterim", "NonEffFinal", "Eff", or "HighEff". The first four outcomes can occur in Stage 1, whereas the last outcome can combine data over Stage 1 and Stage 2.
- stopTime: the time of hitting a stopping boundary since the first subject enrolled in the trial
- stopInfectCnt: the pooled number of infections at stopTime
- summObj: a data.frame containing summary information from each non-/high efficacy interim analysis
- finalHRci: the final CI for the hazard ratio, available if estimand!="cuminc" and there is at least 1 infection in each arm
- firstNonEffCnt: the number of infections that triggered non-efficacy monitoring (if available)
- totInfecCnt: the total number of stage1 (stage2 if boundHit = "HighEff") infections
- totInfecSplit: a table with the numbers of stage1 (stage2 if boundHit = "HighEff") infections in the treatment and control arm
- lastExitTime: the time between the first subject's enrollment and the last subject's exiting from the trial

References

Freidlin B., Korn E. L., and Gray R. (2010), A general inefficacy interim monitoring rule for randomized clinical trials. *Clinical Trials* 7(3):197-208.

See Also

[simTrial](#), [censTrial](#), and [rankTrial](#)

Examples

```
simData <- simTrial(N=c(1000, rep(700, 2)), aveVE=seq(0, 0.4, by=0.2),
                    VEmodel="half", vePeriods=c(1, 27, 79), enrollPeriod=78,
                    enrollPartial=13, enrollPartialRelRate=0.5, dropoutRate=0.05,
                    infecRate=0.04, fuTime=156,
```

```
                    visitSchedule=c(0, (13/3)*(1:4), seq(13*6/3, 156, by=13*2/3)),
                    missVaccProb=c(0,0.05,0.1,0.15), VEcutoffWeek=26, nTrials=5,
                    stage1=78, randomSeed=300)

monitorData <- monitorTrial(dataFile=simData, stage1=78, stage2=156,
                            harmMonitorRange=c(10,100), alphaPerTest=NULL,
                            nonEffStartMethod="FKG", nonEffInterval=20,
                            lowerVEnoneff=0, upperVEnoneff=0.4, highVE=0.7,
                            stage1VE=0, lowerVEuncPower=0, alphaNoneff=0.05,
                            alphaHigh=0.05, alphaStage1=0.05, alphaUncPower=0.05,
                            estimand="cuminc", lagTime=26)

### alternatively, to save the .RData output file (no '<-' needed):
###
### simTrial(N=c(1400, rep(1000, 2)), aveVE=seq(0, 0.4, by=0.2), VEmodel="half",
###          vePeriods=c(1, 27, 79), enrollPeriod=78, enrollPartial=13,
###          enrollPartialRelRate=0.5, dropoutRate=0.05, infecRate=0.04, fuTime=156,
###          visitSchedule=c(0, (13/3)*(1:4), seq(13*6/3, 156, by=13*2/3)),
###          missVaccProb=c(0,0.05,0.1,0.15), VEcutoffWeek=26, nTrials=30,
###          stage1=78, saveDir="./", randomSeed=300)
###
### monitorTrial(dataFile=
###              "simTrial_nPlac=1400_nVacc=1000_1000_aveVE=0.2_0.4_infRate=0.04.RData",
###              stage1=78, stage2=156, harmMonitorRange=c(10,100), alphaPerTest=NULL,
###              nonEffStartMethod="FKG", nonEffInterval=20, lowerVEnoneff=0,
###              upperVEnoneff=0.4, highVE=0.7, stage1VE=0, lowerVEuncPower=0,
###              alphaNoneff=0.05, alphaHigh=0.05, alphaStage1=0.05, alphaUncPower=0.05,
###              estimand="cuminc", lagTime=26, saveDir="./")
```

---

rankTrial	*Ranking and Selection, and Head-to-Head Comparison of Individual and Pooled Treatment Arms*

Description

rankTrial assesses the probability of correctly selecting the winning most efficacious (individual and/or pooled) treatment arm, and assesses power to detect relative treatment efficacy in head-to-head comparisons of (individual and/or pooled) treatment arms.

Usage

```
rankTrial(censFile, idxHighestVE, headHead = NULL, poolHead = NULL,
   lowerVE, stage1, stage2, alpha, saveDir = NULL, verbose = TRUE)
```

Arguments

censFile   if saveDir = NULL, a list returned by censTrial; otherwise a name (character
                string) of an .RData file created by censTrial

idxHighestVE	an integer value identifying the treatment (vaccine) arm with the true highest VE(0–stage2)
headHead	a matrix (ncol = 2) of treatment arm indices for head-to-head comparisons, where the treatment with higher efficacy is listed first in each row
poolHead	a matrix (ncol equals 3 or 4) of treatment arm indices for pooled-arm comparisons, where the pooled treatment with higher efficacy pooled over the first two columns is compared with the (pooled) treatment defined by columns 3 and onward. Ranking and selection of pooled arms is performed separately for each row of poolHead.
lowerVE	a numeric value defining a 'winning' treatment arm as one with sufficient evidence for rejecting the null hypothesis H0: VE(0–stage1) $\leq$ lowerVE x 100% (typically set equal to 0)
stage1	the final week of stage 1 in a two-stage trial
stage2	the final week of stage 2 in a two-stage trial, i.e., the maximum follow-up time
alpha	one minus the nominal confidence level of the two-sided confidence interval used for testing a null hypothesis H0: VE(0–stage1) $\leq b$ x 100% against an alternative hypothesis H1: VE(0–stage1) $> b$ x 100%
saveDir	a character string specifying a path for censFile. If supplied, the output is also saved as an .RData file in this directory; otherwise the output is returned as a list.
verbose	a logical value indicating whether information on the output directory and file name should be printed out (default is TRUE)

Details

All time variables use week as the unit of time. Month is defined as 52/12 weeks.

The probability of correct treatment selection is defined as the probability that the treatment arm with the highest estimated VE(0–stage2) is the one with the true highest VE(0–stage2) and, for this treatment arm, the null hypothesis H0: VE(0–stage1) $\leq$ lowerVE x 100% is rejected. If poolHead is specified, the probability of correct pooled treatment selection is assessed for each set of two pooled treatment arms.

VE(0–t) is estimated as one minus the ratio of Nelson-Aalen-based cumulative incidence functions. The null hypothesis H0: VE(0–t) $\leq b$ x 100% is rejected if the lower bound of the two-sided (1-alpha) x 100% confidence interval for VE(0–t) lies above b.

For head-to-head individual and pooled treatment comparisons, powers to reject the null hypotheses that relative VE(0–stage1) $\leq$ 0% and relative VE(0–stage2) $\leq$ 0% are assessed using the aforementioned testing rule.

Value

If saveDir is specified, the output list (named out) is saved as an .RData file in saveDir (the path to saveDir is printed); otherwise it is returned. The output object is a list with the following components:

- rankSelectPw: the probability of correct selection of the winning most efficacious individual treatment

- headHeadPw: if headHead is specified, a matrix of powers to detect relative VE(0–stage1) (column 1) and relative VE(0–stage2) (column 2) in head-to-head comparisons of individual treatment arms

- poolRankSelectPw: if poolHead is specified, a numeric vector of the probabilities of correct selection of the winning most efficacious pooled treatment for each set of pooled treatments

- poolHeadPw: if poolHead is specified, a matrix of powers to detect relative VE(0–stage1) (column 1) and relative VE(0–stage2) (column 2) in head-to-head comparisons of pooled treatment arms

See Also

[simTrial](#), [monitorTrial](#), and [censTrial](#)

Examples

```
simData <- simTrial(N=c(1000, rep(700, 2)), aveVE=seq(0, 0.4, by=0.2),
                    VEmodel="half", vePeriods=c(1, 27, 79), enrollPeriod=78,
                    enrollPartial=13, enrollPartialRelRate=0.5, dropoutRate=0.05,
                    infecRate=0.04, fuTime=156,
                    visitSchedule=c(0, (13/3)*(1:4), seq(13*6/3, 156, by=13*2/3)),
                    missVaccProb=c(0,0.05,0.1,0.15), VEcutoffWeek=26, nTrials=5,
                    stage1=78, randomSeed=300)

monitorData <- monitorTrial(dataFile=simData, stage1=78, stage2=156,
                            harmMonitorRange=c(10,100), alphaPerTest=NULL,
                            nonEffStartMethod="FKG", nonEffInterval=20,
                            lowerVEnoneff=0, upperVEnoneff=0.4,
                            highVE=0.7, stage1VE=0, lowerVEuncPower=0,
                            alphaNoneff=0.05, alphaHigh=0.05, alphaStage1=0.05,
                            alphaUncPower=0.05, estimand="cuminc", lagTime=26)

censData <- censTrial(dataFile=simData, monitorFile=monitorData, stage1=78, stage2=156)

rankData <- rankTrial(censFile=censData, idxHighestVE=2,
                      headHead=matrix(2:1, nrow=1, ncol=2), lowerVE=0, stage1=78,
                      stage2=156, alpha=0.05)

### alternatively, to save the .RData output file (no '<-' needed):
###
### simTrial(N=c(1400, rep(1000, 2)), aveVE=seq(0, 0.4, by=0.2), VEmodel="half",
###          vePeriods=c(1, 27, 79), enrollPeriod=78, enrollPartial=13,
###          enrollPartialRelRate=0.5, dropoutRate=0.05, infecRate=0.04, fuTime=156,
###          visitSchedule=c(0, (13/3)*(1:4), seq(13*6/3, 156, by=13*2/3)),
###          missVaccProb=c(0,0.05,0.1,0.15), VEcutoffWeek=26, nTrials=30,
###          stage1=78, saveDir="./", randomSeed=300)
###
### monitorTrial(dataFile=
###          "simTrial_nPlac=1400_nVacc=1000_1000_aveVE=0.2_0.4_infRate=0.04.RData",
###          stage1=78, stage2=156, harmMonitorRange=c(10,100), alphaPerTest=NULL,
###          nonEffStartMethod="FKG", nonEffInterval=20,
```

```
###            lowerVEnoneff=0, upperVEnoneff=0.4, highVE=0.7, stage1VE=0,
###            lowerVEuncPower=0, alphaNoneff=0.05, alphaHigh=0.05, alphaStage1=0.05,
###            alphaUncPower=0.05, estimand="cuminc", lagTime=26, saveDir="./")
###
### censTrial(dataFile=
###   "simTrial_nPlac=1400_nVacc=1000_1000_aveVE=0.2_0.4_infRate=0.04.RData",
###   monitorFile=
###   "monitorTrial_nPlac=1400_nVacc=1000_1000_aveVE=0.2_0.4_infRate=0.04_cuminc.RData",
###   stage1=78, stage2=156, saveDir="./")
###
### rankTrial(censFile=
###   "trialDataCens_nPlac=1400_nVacc=1000_1000_aveVE=0.2_0.4_infRate=0.04_cuminc.RData",
###   idxHighestVE=2, headHead=matrix(2:1, nrow=1, ncol=2), lowerVE=0, stage1=78,
###   stage2=156, alpha=0.05, saveDir="./")
```

| simTrial | *Simulation of Multi-Arm Randomized Phase IIb/III Efficacy Trials with Time-to-Event Endpoints* |

Description

`simTrial` generates independent time-to-event data-sets according to a user-specified trial design. The user makes assumptions about the enrollment, dropout, and infection processes in each treatment arm.

Usage

```
simTrial(N, aveVE, VEmodel = c("half", "constant"), vePeriods,
  enrollPeriod, enrollPartial, enrollPartialRelRate, dropoutRate,
  infecRate, fuTime, visitSchedule, missVaccProb = NULL, VEcutoffWeek,
  nTrials, blockSize = NULL, stage1, saveFile = NULL, saveDir = NULL,
  verbose = TRUE, randomSeed = NULL)
```

Arguments

N	a numeric vector specifying the numbers of enrolled trial participants per treatment arm. The length of N equals the total number of treatment arms, and the first component of N represents the control arm.
aveVE	a numeric vector containing, for each treatment arm in N, a time-averaged vaccine efficacy (VE), defined as the weighted average of VEs in the time intervals specified by vePeriods. If VEmodel = "half", VE is halved in the initial interval, the full VE is applied in the second interval, and aveVE is applied thereafter. The components of N and aveVE correspond to each other.
VEmodel	a character string specifying whether VE is assumed constant over time (option "constant") or halved in the initial time interval as defined in vePeriods (option "half"). Only the first character is necessary.

vePeriods	a numeric vector defining start times (in weeks) of time intervals with (potentially) distinct VE levels depending on the choice of the `VEmodel`
enrollPeriod	the final week of the enrollment period
enrollPartial	the final week of the portion of the enrollment period with a reduced enrollment rate defined by `enrollPartialRelRate`
enrollPartialRelRate	
	a non-negative value characterizing the fraction of the weekly enrollment rate governing enrollment from week 1 until week `enrollPartial`
dropoutRate	a (prior) annual dropout rate
infecRate	a (prior) annual infection rate in the control arm
fuTime	a follow-up time (in weeks) of each participant
visitSchedule	a numeric vector listing the visit weeks at which testing for the endpoint is conducted
missVaccProb	a numeric vector with conditional probabilities of having missed a vaccination given the follow-up time exceeds `VEcutoffWeek` weeks. For each component, a separate per-protocol indicator is generated. Each per-protocol cohort includes subjects with (i) a non-missing vaccination, and (ii) follow-up time exceeding `VEcutoffWeek` weeks. If `NULL`, no per-protocol indicators are included.
VEcutoffWeek	a time cut-off (in weeks); the follow-up time exceeding `VEcutoffWeek` weeks is required for inclusion in the per-protocol cohort
nTrials	the number of trials to be simulated
blockSize	a constant block size to be used in permuted-block randomization. The choice of `blockSize` requires caution to achieve the desired balance of treatment assignments within a block.
stage1	the final week of stage 1 in a two-stage trial
saveFile	a character string specifying the name of the output `.RData` file. If `NULL` (default), a default file name will be used.
saveDir	a character string specifying a path for the output directory. If supplied, the output is saved as an `.RData` file in the directory; otherwise the output is returned as a list.
verbose	a logical value indicating whether information on the output directory and file name should be printed out (default is `TRUE`)
randomSeed	sets seed of the random number generator for simulation reproducibility

Details

All time variables use week as the unit of time. Month is defined as 52/12 weeks.

The prior weekly enrollment rate is calculated based on the duration of the enrollment periods with reduced/full enrollment rates and the total number of subjects to be enrolled.

The weekly enrollment, dropout and infection rates used for generating trial data are sampled from specified prior distributions (the prior annual dropout and infection probabilities are specified by the user). The default choice considers non-random point-mass distributions, i.e., the prior rates directly govern the accumulation of trial data.

Subjects' enrollment is assumed to follow a Poisson process with a time-varying rate (the argument enrollPartialRelRate characterizes a reduced enrollment rate applied to weeks 1 through enrollPartial, i.e., full enrollment starts at week enrollPartial+1). The number of enrolled subjects is determined by the vector N.

Dropout times are assumed to follow an exponential distribution where the probability of a dropout within 1 week is equal to dropoutRate/52.

Permuted-block randomization is used for assigning treatment labels. If left unspecified by the user, an appropriate block size, no smaller than 10, will computed and used. The function getBlockSize can be used to determine appropriate block sizes (see help(getBlockSize)).

Infection times are generated following the VE schedule characterized by aveVE, VEmodel and vePeriods. Independent exponential times are generated within each time period of constant VE, and their minimum specifies the right-censored infection time. Exponential rates are chosen that satisfy the user-specified requirements on the treatment- and time-period-specific probabilities of an infection within 1 week (in the control arm, the infection probability within 1 week uniformly equals infecRate/52).

Infection diagnosis times are calculated according to the visitSchedule. The observed follow-up time is defined as the minumum of the infection diagnosis time, dropout time, and fuTime.

Value

If saveDir is specified, the output list (named trialObj) is saved as an .RData file (the output directory path is printed); otherwise it is returned. The output object is a list with the following components:

- trialData: a list with nTrials components each of which is a data.frame with at least the variables trt, entry, exit, and event storing the treatment assignments, enrollment times, study exit times, and event indicators, respectively. The observed follow-up times can be recovered as exit - entry. Indicators of belonging to the per-protocol cohort (named pp1, pp2, etc.) are included if missVaccProb is specified.

- NinfStage1: a list whose components are numeric vectors with the numbers of stage1 infections by treatment ([1] = control arm) for each simulated trial

- nTrials: the number of simulated trials

- N: the total number of enrolled trial participants

- nArms: the number of treatment arms

- trtAssgnProbs: a numeric vector containing the treatment assignment probabilities

- blockSize: the block size used for treatment assignment

- fuTime: the follow-up time (in weeks) of each participant

- rates: a list with three components: the prior weekly enrollment rate (enrollment), the prior probability of dropout within 1 week (dropout), and the prior probability of infection within 1 week (infection)

- enrollSchedule: a data.frame summarizing information on enrollment periods and corresponding relative enrollment rates (relative to the weekly "base" enrollment rate). The column names are start, end, and relativeRates.

- VEs: a list with components being numeric vectors containing VE levels assumed within time periods defined by vePeriods for each active treatment arm

- infecRates: a data.frame summarizing information on time periods of distinct VE across all treatment arms. The variables trt, start, end, and relRate carry treatment assignment labels, first and last week of a time interval, and the pertaining assumed hazard ratio in the given interval.

- randomSeed: the set seed of the random number generator for simulation reproducibility

See Also

[monitorTrial](), [censTrial](), and [rankTrial]()

Examples

```
simData <- simTrial(N=c(1000, rep(700, 2)), aveVE=seq(0, 0.4, by=0.2),
                    VEmodel="half", vePeriods=c(1, 27, 79), enrollPeriod=78,
                    enrollPartial=13, enrollPartialRelRate=0.5, dropoutRate=0.05,
                    infecRate=0.04, fuTime=156,
                    visitSchedule=c(0, (13/3)*(1:4), seq(13*6/3, 156, by=13*2/3)),
                    missVaccProb=c(0,0.05,0.1,0.15), VEcutoffWeek=26, nTrials=5,
                    blockSize=30, stage1=78, randomSeed=300)

### alternatively, to save the .RData output file (no '<-' needed):
###
### simTrial(N=c(1400, rep(1000, 2)), aveVE=seq(0, 0.4, by=0.2), VEmodel="half",
###          vePeriods=c(1, 27, 79), enrollPeriod=78, enrollPartial=13,
###          enrollPartialRelRate=0.5, dropoutRate=0.05, infecRate=0.04, fuTime=156,
###          visitSchedule=c(0, (13/3)*(1:4), seq(13*6/3, 156, by=13*2/3)),
###          missVaccProb=c(0,0.05,0.1,0.15), VEcutoffWeek=26, nTrials=5,
###          blockSize=30, stage1=78, saveDir="./", randomSeed=300)
```

VEpowerPP	*Unconditional Power to Detect Positive Treatment Efficacy in a Per-Protocol Cohort*

Description

VEpowerPP computes unconditional power to detect positive treatment (vaccine) efficacy in per-protocol cohorts identified in simTrial-generated data-sets.

Usage

```
VEpowerPP(dataList, lowerVEuncPower, alphaUncPower, VEcutoffWeek, stage1,
  outName = NULL, saveDir = NULL, verbose = TRUE)
```

Arguments

dataList	if saveDir = NULL, a list of objects (lists) returned by censTrial; otherwise a list of .RData file names (character strings) generated by censTrial
lowerVEuncPower	
	a numeric value specifying a one-sided null hypothesis H0: VE(VEcutoffWeek–stage1) $\leq$ lowerVEuncPower x 100%. Unconditional power (i.e., accounting for sequential monitoring) to reject H0 in the per-protocol cohort is calculated, where the rejection region is defined by the lower bound of the two-sided (1-alphaUncPower) x 100% confidence interval for VE(VEcutoffWeek–stage1) being above lowerVEuncPower (typically a number in the 0–0.5 range).
alphaUncPower	one minus the nominal confidence level of the two-sided confidence interval used to test the one-sided null hypothesis H0: VE(VEcutoffWeek–stage1) $\leq$ lowerVEuncPower x 100% against the alternative hypothesis H1: VE(VEcutoffWeek–stage1) > lowerVEuncPower x 100%.
VEcutoffWeek	a cut-off time (in weeks). Only subjects with the follow-up time exceeding VEcutoffWeek are included in the per-protocol cohort.
stage1	the final week of stage 1 in a two-stage trial
outName	a character string specifying the output .RData file name. If outName = NULL but saveDir is specified, the output file is named VEpwPP.RData.
saveDir	a character string specifying a path for the output directory. If supplied, the output is saved as an .RData file named outName in the directory; otherwise the output is returned as a list.
verbose	a logical value indicating whether information on the output directory and file name should be printed out (default is TRUE)

Details

All time variables use week as the unit of time. Month is defined as 52/12 weeks.

A per-protocol cohort indicator is assumed to be included in the simTrial-generated data-sets, which is ensured by specifying the missVaccProb argument in simTrial.

VE(VEcutoffWeek–stage1) is estimated as one minus the ratio of Nelson-Aalen-based cumulative incidence functions. VEpowerPP computes power to reject the null hypothesis H0: VE(VEcutoffWeek–stage1) $\leq$ lowerVEuncPower x 100%. H0 is rejected if the lower bound of the two-sided (1-alphaUncPower) x 100% confidence interval for VE(VEcutoffWeek–stage1) lies above lowerVEuncPower.

Value

If saveDir is specified, the output list (named pwList) is saved as an .RData file named outName (or VEpwPP.RData if left unspecified); otherwise the output list is returned. The output object is a list (of equal length as dataList) of lists with the following components:

- VE: a numeric vector of VE(VEcutoffWeek–stage1) estimates for each missing vaccination probability in missVaccProb of simTrial

- VEpwPP: a numeric vector of powers to reject the null hypothesis H0: VE(VEcutoffWeek–stage1) $\leq$ lowerVEuncPower x 100% for each missing vaccination probability in missVaccProb of simTrial

See Also

simTrial

Examples

```
simData <- simTrial(N=rep(1000, 2), aveVE=c(0, 0.4), VEmodel="half",
                    vePeriods=c(1, 27, 79), enrollPeriod=78,
                    enrollPartial=13, enrollPartialRelRate=0.5, dropoutRate=0.05,
                    infecRate=0.04, fuTime=156,
                    visitSchedule=c(0, (13/3)*(1:4), seq(13*6/3, 156, by=13*2/3)),
                    missVaccProb=c(0,0.05,0.1,0.15), VEcutoffWeek=26, nTrials=5,
                    stage1=78, randomSeed=300)

monitorData <- monitorTrial(dataFile=simData, stage1=78, stage2=156,
                            harmMonitorRange=c(10,100), alphaPerTest=NULL,
                            nonEffStartMethod="FKG", nonEffInterval=20,
                            lowerVEnoneff=0, upperVEnoneff=0.4,
                            highVE=0.7, stage1VE=0, lowerVEuncPower=0,
                            alphaNoneff=0.05, alphaHigh=0.05, alphaStage1=0.05,
                            alphaUncPower=0.05, estimand="cuminc", lagTime=26)

censData <- censTrial(dataFile=simData, monitorFile=monitorData, stage1=78, stage2=156)

VEpwPP <- VEpowerPP(dataList=list(censData), lowerVEuncPower=0, alphaUncPower=0.05,
                    VEcutoffWeek=26, stage1=78)

### alternatively, to save the .RData output file (no '<-' needed):
###
### simTrial(N=rep(1000, 2), aveVE=c(0, 0.4), VEmodel="half",
###          vePeriods=c(1, 27, 79), enrollPeriod=78, enrollPartial=13,
###          enrollPartialRelRate=0.5, dropoutRate=0.05, infecRate=0.04, fuTime=156,
###          visitSchedule=c(0, (13/3)*(1:4), seq(13*6/3, 156, by=13*2/3)),
###          missVaccProb=c(0,0.05,0.1,0.15), VEcutoffWeek=26, nTrials=5,
###          stage1=78, saveDir="./", randomSeed=300)
###
### monitorTrial(dataFile=
###          "simTrial_nPlac=1000_nVacc=1000_aveVE=0.4_infRate=0.04.RData",
###          stage1=78, stage2=156, harmMonitorRange=c(10,100), alphaPerTest=NULL,
###          nonEffStartMethod="FKG", nonEffInterval=20,
###          lowerVEnoneff=0, upperVEnoneff=0.4, highVE=0.7, stage1VE=0,
###          lowerVEuncPower=0, alphaNoneff=0.05, alphaHigh=0.05, alphaStage1=0.05,
###          alphaUncPower=0.05, estimand="cuminc", lagTime=26, saveDir="./")
###
### censTrial(dataFile=
###   "simTrial_nPlac=1000_nVacc=1000_aveVE=0.4_infRate=0.04.RData",
###   monitorFile=
###   "monitorTrial_nPlac=1000_nVacc=1000_aveVE=0.4_infRate=0.04_cuminc.RData",
###   stage1=78, stage2=156, saveDir="./")
###
### VEpowerPP(dataList=
###   list("trialDataCens_nPlac=1000_nVacc=1000_aveVE=0.4_infRate=0.04_cuminc.RData"),
###   lowerVEuncPower=0, alphaUncPower=0.05, VEcutoffWeek=26, stage1=78, saveDir="./")
```

Index