Package 'Surrogate'

February 21, 2016

Type Package

Title Evaluation of Surrogate Endpoints in Clinical Trials

Version 0.1-68

Date 2016-02-21

Author Wim Van der Elst, Paul Meyvisch, Ariel Alonso, Hannah M. Ensor, Christopher J. Weir & Geert Molenberghs

Maintainer Wim Van der Elst <Wim.vanderelst@gmail.com>

Description In a clinical trial, it frequently occurs that the most credible outcome to evaluate the effectiveness of a new therapy (the true endpoint) is difficult to measure. In such a situation, it can be an effective strategy to replace the true endpoint by a biomarker that is easier to measure and that allows for a prediction of the treatment effect on the true endpoint (a surrogate endpoint). The package 'Surrogate' allows for an evaluation of the appropriateness of a candidate surrogate endpoint based on the meta-analytic, information-theoretic, and causal-inference frameworks. Part of this software has been developed using funding provided from the European Union's 7th Framework Programme for research, technological development and demonstration under Grant Agreement no 602552.

Depends MASS, msm

Imports rgl, lattice, latticeExtra, survival, nlme, lme4, OrdinalLogisticBiplot, logistf, rms

License GPL (>= 2)

BugReports Wim Van der Elst <Wim.vanderelst@gmail.com>

Repository CRAN

NeedsCompilation no

Date/Publication 2016-02-21 19:51:43

R topics documented:

ARMD *Data of the Age-Related Macular Degeneration Study*

Description

These are the data of a clinical trial involving patients suffering from age-related macular degeneration (ARMD), a condition that involves a progressive loss of vision. A total of 181 patients from 36 centers participated in the trial. Patients' visual acuity was assessed using standardized vision charts. There were two treatment conditions (placebo and interferon-α). The potential surrogate endpoint is the change in the visual acuity at 24 weeks (6 months) after starting treatment. The true endpoint is the change in the visual acuity at 52 weeks.

Usage

```
data(ARMD)
```

Format

A `data.frame` with 181 observations on 5 variables.

`Id` The Patient ID.

`Center` The center in which the patient was treated.

`Treat` The treatment indicator, coded as -1 = placebo and 1 = interferon-α.

`Diff24` The change in the visual acuity at 24 weeks after starting treatment. This endpoint is a potential surrogate for `Diff52`.

`Diff52` The change in the visual acuity at 52 weeks after starting treatment. This outcome serves as the true endpoint .

`BifixedContCont` *Fits a bivariate fixed-effects model to assess surrogacy in the meta-analytic multiple-trial setting (Continuous-continuous case)*

Description

The function `BifixedContCont` uses the bivariate fixed-effects approach to estimate trial- and individual-level surrogacy when the data of multiple clinical trials are available. The user can specify whether a (weighted or unweighted) full, semi-reduced, or reduced model should be fitted. See the **Details** section below.

Usage

```
BifixedContCont(Dataset, Surr, True, Treat, Trial.ID, Pat.ID, Model=c("Full"),
Weighted=TRUE, Min.Trial.Size=2, Alpha=.05)
```

Arguments

`Dataset`	A `data.frame` that should consist of one line per patient. Each line contains (at least) a surrogate value, a true endpoint value, a treatment indicator, a patient ID, and a trial ID.
`Surr`	The name of the variable in `Dataset` that contains the surrogate endpoint values.
`True`	The name of the variable in `Dataset` that contains the true endpoint values.
`Treat`	The name of the variable in `Dataset` that contains the treatment indicators. The treatment indicator should either be coded as 1 for the experimental group and -1 for the control group, or as 1 for the experimental group and 0 for the control group.
`Trial.ID`	The name of the variable in `Dataset` that contains the trial ID to which the patient belongs.
`Pat.ID`	The name of the variable in `Dataset` that contains the patient's ID.
`Model`	The type of model that should be fitted, i.e., `Model=c("Full")`, `Model=c("Reduced")`, or `Model=c("SemiReduced")`. See the **Details** section below. Default `Model=c("Full")`.
`Weighted`	Logical. If `TRUE`, then a weighted regression analysis is conducted at stage 2 of the two-stage approach. If `FALSE`, then an unweighted regression analysis is conducted at stage 2 of the two-stage approach. See the **Details** section below. Default `TRUE`.
`Min.Trial.Size`	The minimum number of patients that a trial should contain in order to be included in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded from the analysis. Default 2.
`Alpha`	The α-level that is used to determine the confidence intervals around R^2_{trial}, R_{trial}, R^2_{indiv} and R_{indiv}. Default 0.05.

Details

When the full bivariate mixed-effects model is fitted to assess surrogacy in the meta-analytic framework (for details, Buyse & Molenberghs, 2000), computational issues often occur. In that situation, the use of simplified model-fitting strategies may be warranted (for details, see Burzykowski et al., 2005; Tibaldi et al., 2003).

The function `BifixedContCont` implements one such strategy, i.e., it uses a two-stage bivariate fixed-effects modelling approach to assess surrogacy. In the first stage of the analysis, a bivariate linear regression model is fitted. When a full or semi-reduced model is requested (by using the argument `Model=c("Full")` or `Model=c("SemiReduced")` in the function call), the following bivariate model is fitted:

$$S_{ij} = \mu_{Si} + \alpha_i Z_{ij} + \varepsilon_{Sij},$$

$$T_{ij} = \mu_{Ti} + \beta_i Z_{ij} + \varepsilon_{Tij},$$

where S_{ij} and T_{ij} are the surrogate and true endpoint values of subject j in trial i, Z_{ij} is the treatment indicator for subject j in trial i, μ_{Si} and μ_{Ti} are the fixed trial-specific intercepts for S and T, and α_i and β_i are the trial-specific treatment effects on S and T, respectively. When a reduced model is requested (by using the argument `Model=c("Reduced")` in the function call), the following bivariate model is fitted:

$$S_{ij} = \mu_S + \alpha_i Z_{ij} + \varepsilon_{Sij},$$

$$T_{ij} = \mu_T + \beta_i Z_{ij} + \varepsilon_{Tij},$$

where μ_S and μ_T are the common intercepts for S and T (i.e., it is assumed that the intercepts for the surrogate and the true endpoints are identical in all trials). The other parameters are the same as defined above.

In the above models, the error terms ε_{Sij} and ε_{Tij} are assumed to be mean-zero normally distributed with variance-covariance matrix $\mathbf{\Sigma}$:

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_{SS} & \\ \sigma_{ST} & \sigma_{TT} \end{pmatrix}.$$

Based on $\mathbf{\Sigma}$, individual-level surrogacy is quantified as:

$$R^2_{indiv} = \frac{\sigma^2_{ST}}{\sigma_{SS}\sigma_{TT}}.$$

Next, the second stage of the analysis is conducted. When a full model is requested by the user (by using the argument `Model=c("Full")` in the function call), the following model is fitted:

$$\widehat{\beta_i} = \lambda_0 + \lambda_1 \widehat{\mu_{Si}} + \lambda_2 \widehat{\alpha_i} + \varepsilon_i,$$

where the parameter estimates for β_i, μ_{Si}, and α_i are based on the full model that was fitted in stage 1.

When a reduced or semi-reduced model is requested by the user (by using the arguments `Model=c("Reduced")` or `Model=c("SemiReduced")` in the function call), the following model is fitted:

$$\widehat{\beta_i} = \lambda_0 + \lambda_1 \widehat{\alpha_i} + \varepsilon_i.$$

where the parameter estimates for β_i and α_i are based on the semi-reduced or reduced model that was fitted in stage 1.

When the argument `Weighted=FALSE` is used in the function call, the model that is fitted in stage 2 is an unweighted linear regression model. When a weighted model is requested (using the argument `Weighted=TRUE` in the function call), the information that is obtained in stage 1 is weighted according to the number of patients in a trial.

The classical coefficient of determination of the fitted stage 2 model provides an estimate of R^2_{trial}.

Value

An object of class `BifixedContCont` with components,

`Data.Analyze` Prior to conducting the surrogacy analysis, data of patients who have a missing value for the surrogate and/or the true endpoint are excluded. In addition, the data of trials (i) in which only one type of the treatment was administered, and (ii) in which either the surrogate or the true endpoint was a constant (i.e., all patients within a trial had the same surrogate and/or true endpoint value) are excluded. In addition, the user can specify the minimum number of patients that a trial should contain in order to include the trial in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded. `Data.Analyze` is the dataset on which the surrogacy analysis was conducted.

`Obs.Per.Trial` A `data.frame` that contains the total number of patients per trial and the number of patients who were administered the control treatment and the experimental treatment in each of the trials (in `Data.Analyze`).

`Results.Stage.1`
The results of stage 1 of the two-stage model fitting approach: a `data.frame` that contains the trial-specific intercepts and treatment effects for the surrogate and the true endpoints (when a full or semi-reduced model is requested), or the trial-specific treatment effects for the surrogate and the true endpoints (when a reduced model is requested).

`Residuals.Stage.1`
A `data.frame` that contains the residuals for the surrogate and true endpoints that are obtained in stage 1 of the analysis (ε_{Sij} and ε_{Tij}).

`Results.Stage.2`
An object of class `lm` (linear model) that contains the parameter estimates of the regression model that is fitted in stage 2 of the analysis.

`Trial.R2` A `data.frame` that contains the trial-level coefficient of determination (R^2_{trial}), its standard error and confidence interval.

`Indiv.R2` A `data.frame` that contains the individual-level coefficient of determination (R^2_{indiv}), its standard error and confidence interval.

`Trial.R` A `data.frame` that contains the trial-level correlation coefficient (R_{trial}), its standard error and confidence interval.

`Indiv.R` A `data.frame` that contains the individual-level correlation coefficient (R_{indiv}), its standard error and confidence interval.

`Cor.Endpoints` A `data.frame` that contains the correlations between the surrogate and the true endpoint in the control treatment group (i.e., ρ_{T0S0}) and in the experimental treatment group (i.e., ρ_{T1S1}), their standard errors and their confidence intervals.

`D.Equiv` The variance-covariance matrix of the trial-specific intercept and treatment effects for the surrogate and true endpoints (when a full or semi-reduced model is fitted, i.e., when `Model=c("Full")` or `Model=c("SemiReduced")` is used in the function call), or the variance-covariance matrix of the trial-specific treatment effects for the surrogate and true endpoints (when a reduced model is fitted, i.e., when `Model=c("Reduced")` is used in the function call). The variance-covariance matrix `D.Equiv` is equivalent to the $\boldsymbol{D}$ matrix that would be obtained when a (full or reduced) bivariate mixed-effect approach is used; see function `BimixedContCont`).

`Sigma` The 2 by 2 variance-covariance matrix of the residuals (ε_{Sij} and ε_{Tij}).

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Burzykowski, T., Molenberghs, G., & Buyse, M. (2005). *The evaluation of surrogate endpoints*. New York: Springer-Verlag.

Buyse, M., Molenberghs, G., Burzykowski, T., Renard, D., & Geys, H. (2000). The validation of surrogate endpoints in meta-analysis of randomized experiments. *Biostatistics, 1,* 49-67.

Tibaldi, F., Abrahantes, J. C., Molenberghs, G., Renard, D., Burzykowski, T., Buyse, M., Parmar, M., et al., (2003). Simplified hierarchical linear models for the evaluation of surrogate endpoints. *Journal of Statistical Computation and Simulation, 73,* 643-658.

See Also

`UnifixedContCont`, `UnimixedContCont`, `BimixedContCont`, `plot Meta-Analytic`

Examples

```
# Example 1, based on the ARMD data
data(ARMD)

# Fit a full bivariate fixed-effects model with weighting according to the
# number of patients in stage 2 of the two stage approach to assess surrogacy:
Sur <- BifixedContCont(Dataset=ARMD, Surr=Diff24, True=Diff52, Treat=Treat, Trial.ID=Center,
Pat.ID=Id, Model="Full", Weighted=TRUE)

# Obtain a summary of the results
summary(Sur)

# Obtain a graphical representation of the trial- and individual-level surrogacy
plot(Sur)
```

```
# Example 2
# Conduct a surrogacy analysis based on a simulated dataset with 2000 patients,
# 100 trials, and Rindiv=Rtrial=.8
# Simulate the data:
Sim.Data.MTS(N.Total=2000, N.Trial=100, R.Trial.Target=.8, R.Indiv.Target=.8,
Seed=123, Model="Reduced")

# Fit a reduced bivariate fixed-effects model with no weighting according to the
# number of patients in stage 2 of the two stage approach to assess surrogacy:
## Not run:  #time-consuming code parts
Sur2 <- BifixedContCont(Dataset=Data.Observed.MTS, Surr=Surr, True=True, Treat=Treat,
Trial.ID=Trial.ID, Pat.ID=Pat.ID, , Model="Reduced", Weighted=FALSE)

# Show summary and plots of results:
summary(Sur2)
plot(Sur2, Weighted=FALSE)
## End(Not run)
```

`BimixedContCont` *Fits a bivariate mixed-effects model to assess surrogacy in the meta-analytic multiple-trial setting (Continuous-continuous case)*

Description

The function `BimixedContCont` uses the bivariate mixed-effects approach to estimate trial- and individual-level surrogacy when the data of multiple clinical trials are available. The user can specify whether a full or reduced model should be fitted. See the **Details** section below.

Usage

```
BimixedContCont(Dataset, Surr, True, Treat, Trial.ID, Pat.ID, Model=c("Full"),
Min.Trial.Size=2, Alpha=.05, ...)
```

Arguments

`Dataset`	A `data.frame` that should consist of one line per patient. Each line contains (at least) a surrogate value, a true endpoint value, a treatment indicator, a patient ID, and a trial ID.
`Surr`	The name of the variable in `Dataset` that contains the surrogate endpoint values.
`True`	The name of the variable in `Dataset` that contains the true endpoint values.
`Treat`	The name of the variable in `Dataset` that contains the treatment indicators. The treatment indicator should either be coded as 1 for the experimental group and -1 for the control group, or as 1 for the experimental group and 0 for the control group.

`Trial.ID`	The name of the variable in `Dataset` that contains the trial ID to which the patient belongs.
`Pat.ID`	The name of the variable in `Dataset` that contains the patient's ID.
`Model`	The type of model that should be fitted, i.e., `Model=c("Full")` or `Model=c("Reduced")`. See the **Details** section below. Default `Model=c("Full")`.
`Min.Trial.Size`	The minimum number of patients that a trial should contain to be included in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded from the analysis. Default 2.
`Alpha`	The α-level that is used to determine the confidence intervals around R^2_{trial}, R_{trial}, R^2_{indiv} and R_{indiv}. Default 0.05.
`...`	Other arguments to be passed to the function `lmer` (of the R package `lme4`) that is used to fit the geralized linear mixed-effect models in the function `BimixedContCont`. For details, see the lme4 manual.

Details

The function `BimixedContCont` fits a bivariate mixed-effects model to assess surrogacy (for details, see Buyse et al., 2000). In particular, the following mixed-effects model is fitted:

$$S_{ij} = \mu_S + m_{Si} + (\alpha + a_i)Z_{ij} + \varepsilon_{Sij},$$

$$T_{ij} = \mu_T + m_{Ti} + (\beta + b_i)Z_{ij} + \varepsilon_{Tij},$$

where i and j are the trial and subject indicators, S_{ij} and T_{ij} are the surrogate and true endpoint values of subject j in trial i, Z_{ij} is the treatment indicator for subject j in trial i, μ_S and μ_T are the fixed intercepts for S and T, m_{Si} and m_{Ti} are the corresponding random intercepts, α and β are the fixed treatment effects for S and T, and a_i and b_i are the corresponding random treatment effects, respectively.

The vector of the random effects (i.e., m_{Si}, m_{Ti}, a_i and b_i) is assumed to be mean-zero normally distributed with variance-covariance matrix $\boldsymbol{D}$:

$$\boldsymbol{D} = \begin{pmatrix} d_{SS} & & & \\ d_{ST} & d_{TT} & & \\ d_{Sa} & d_{Ta} & d_{aa} & \\ d_{Sb} & d_{Tb} & d_{ab} & d_{bb} \end{pmatrix}.$$

The trial-level coefficient of determination (i.e., R^2_{trial}) is quantified as:

$$R^2_{trial} = \frac{\begin{pmatrix} d_{Sb} \\ d_{ab} \end{pmatrix}' \begin{pmatrix} d_{SS} & d_{Sa} \\ d_{Sa} & d_{aa} \end{pmatrix}^{-1} \begin{pmatrix} d_{Sb} \\ d_{ab} \end{pmatrix}}{d_{bb}}.$$

The error terms ε_{Sij} and ε_{Tij} are assumed to be mean-zero normally distributed with variance-covariance matrix $\boldsymbol{\Sigma}$:

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_{SS} & \\ \sigma_{ST} & \sigma_{TT} \end{pmatrix}.$$

Based on $\mathbf{\Sigma}$, individual-level surrogacy is quantified as:

$$R^2_{indiv} = \frac{\sigma^2_{ST}}{\sigma_{SS}\sigma_{TT}}.$$

Note

When the full bivariate mixed-effects approach is used to assess surrogacy in the meta-analytic framework (for details, see Buyse & Molenberghs, 2000), computational issues often occur. Such problems mainly occur when the number of trials is low, the number of patients in the different trials is low, and/or when the trial-level heterogeneity is small (Burzykowski et al., 2000).

In that situation, the use of a simplified model-fitting strategy may be warranted (for details, see Burzykowski et al., 2000; Tibaldi et al., 2003).

For example, a reduced bivariate-mixed effect model can be fitted instead of a full model (by using the `Model=c("Reduced")` argument in the function call). In the reduced model, the random-effects structure is simplified (i) by assuming that there is no heterogeneity in the random intercepts, or (ii) by assuming that the covariance between the random intercepts and random treatment effects is zero. Note that under this assumption, the computation of the trial-level coefficient of determination (i.e., R^2_{trial}) simplifies to:

$$R^2_{trial} = \frac{d^2_{ab}}{d_{aa}d_{bb}}.$$

Alternatively, the bivariate mixed-effects model may be abandonned and the user may fit a univariate fixed-effects model, a bivariate fixed-effects model, or a univariate mixed-effects model (for details, see Tibaldi et al., 2003). These models are implemented in the functions `UnifixedContCont`, `BifixedContCont`, and `UnimixedContCont`).

Value

An object of class `BimixedContCont` with components,

`Data.Analyze` Prior to conducting the surrogacy analysis, data of patients who have a missing value for the surrogate and/or the true endpoint are excluded. In addition, the data of trials (i) in which only one type of the treatment was administered, and (ii) in which either the surrogate or the true endpoint was a constant (i.e., all patients within a trial had the same surrogate and/or true endpoint value) are excluded. In addition, the user can specify the minimum number of patients that a trial should contain in order to include the trial in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded. `Data.Analyze` is the dataset on which the surrogacy analysis was conducted.

`Obs.Per.Trial` A `data.frame` that contains the total number of patients per trial and the number of patients who were administered the control treatment and the experimental treatment in each of the trials (in `Data.Analyze`).

`Trial.Spec.Results`
A `data.frame` that contains the trial-specific intercepts and treatment effects on the surrogate and the true endpoints when a full model is requested (i.e., $\mu_S +$

m_{Si}, $\mu_T + m_{Ti}$, $\alpha + a_i$, and $\beta + b_i$), or the trial-specific treatment effects on the surrogate and the true endpoints when a reduced model is requested (i.e., $\alpha + a_i$, and $\beta + b_i$). Note that the results that are contained in `Trial.Spec.Results` are equivalent to the results in `Results.Stage.1` that are obtained when the functions `UnifixedContCont`, `UnimixedContCont`, or `BifixedContCont` are used.

`Residuals` A `data.frame` that contains the residuals for the surrogate and true endpoints (ε_{Sij} and ε_{Tij}).

`Fixed.Effect.Pars`
A `data.frame` that contains the fixed intercept and treatment effects for the surrogate and the true endpoints (i.e., μ_S, μ_T, α, and β).

`Random.Effect.Pars`
A `data.frame` that contains the random intercept and treatment effects for the surrogate and the true endpoints (i.e., m_{Si}, m_{Ti}, a_i, and b_i) when a full model is fitted (i.e., when `Model=c("Full")` is used in the function call), or that contains the random treatment effects for the surrogate and the true endpoints (i.e., a_i and b_i) when a reduced model is fitted (i.e., when `Model=c("Reduced")` is used in the function call).

`Trial.R2` A `data.frame` that contains the trial-level coefficient of determination (R^2_{trial}), its standard error and confidence interval.

`Indiv.R2` A `data.frame` that contains the individual-level coefficient of determination (R^2_{indiv}), its standard error and confidence interval.

`Trial.R` A `data.frame` that contains the trial-level correlation coefficient (R_{trial}), its standard error and confidence interval.

`Indiv.R` A `data.frame` that contains the individual-level correlation coefficient (R_{indiv}), its standard error and confidence interval.

`Cor.Endpoints` A `data.frame` that contains the correlations between the surrogate and the true endpoint in the control treatment group (i.e., ρ_{T0S0}) and in the experimental treatment group (i.e., ρ_{T1S1}), their standard errors and their confidence intervals.

`D` The variance-covariance matrix of the random effects (the $\boldsymbol{D}$ matrix), i.e., a 4 by 4 variance-covariance matrix of the random intercept and treatment effects when a full model is fitted (i.e., when `Model=c("Full")` is used in the function call), or a 2 by 2 variance-covariance matrix of the random treatment effects when a reduced model is fitted (i.e., when `Model=c("Reduced")` is used in the function call).

`Sigma` The 2 by 2 variance-covariance matrix of the residuals (ε_{Sij} and ε_{Tij}).

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Burzykowski, T., Molenberghs, G., & Buyse, M. (2005). *The evaluation of surrogate endpoints*. New York: Springer-Verlag.

Buyse, M., Molenberghs, G., Burzykowski, T., Renard, D., & Geys, H. (2000). The validation of surrogate endpoints in meta-analysis of randomized experiments. *Biostatistics, 1,* 49-67.

Tibaldi, F., Abrahantes, J. C., Molenberghs, G., Renard, D., Burzykowski, T., Buyse, M., Parmar, M., et al., (2003). Simplified hierarchical linear models for the evaluation of surrogate endpoints. *Journal of Statistical Computation and Simulation, 73,* 643-658.

See Also

UnifixedContCont, BifixedContCont, UnimixedContCont, plot Meta-Analytic

Examples

```
# Open the Schizo dataset (clinial trial in schizophrenic patients)
data(Schizo)

## Not run:  #Time consuming (>5 sec) code part
# When a reduced bivariate mixed-effect model is used to assess surrogacy,
# the conditioning number for the D matrix is very high:
Sur <- BimixedContCont(Dataset=Schizo, Surr=BPRS, True=PANSS, Treat=Treat, Model="Reduced",
Trial.ID=InvestId, Pat.ID=Id)

# Such problems often occur when the total number of patients, the total number
# of trials and/or the trial-level heterogeneity
# of the treatment effects is relatively small

# As an alternative approach to assess surrogacy, consider using the functions
# BifixedContCont, UnifixedContCont or UnimixedContCont in the meta-analytic framework,
# or use the information-theoretic approach

## End(Not run)
```

CausalDiagramBinBin *Draws a causal diagram depicting the median informational coefficients of correlation (or odds ratios) between the counterfactuals for a specified range of values of the ICA in the binary-binary setting.*

Description

This function provides a diagram that depicts the medians of the informational coefficients of correlation (or odds ratios) between the counterfactuals for a specified range of values of the individual causal association in the binary-binary setting (R^2_H).

Usage

```
CausalDiagramBinBin(x, Values="Corrs", Theta_T0S0, Theta_T1S1,
Min=0, Max=1, Cex.Letters=3, Cex.Corrs=2, Lines.Rel.Width=TRUE,
Col.Pos.Neg=TRUE, Monotonicity)
```

Arguments

x	An object of class ICA.BinBin. See ICA.BinBin.
Values	Specifies whether the median informational coefficients of correlation or median odds ratios between the counterfactuals should be depicted, i.e., Values="Corrs" or Values="ORs".
Theta_T0S0	The odds ratio between T and S in the control group. This quantity is estimable based on the observed data. Only has to be provided when Values="ORs".
Theta_T1S1	The odds ratio between T and S in the experimental treatment group. This quantity is estimable based on the observed data. Only has to be provided when Values="ORs".
Min	The minimum value of R_H^2 that should be considered. Default=-1.
Max	The maximum value of R_H^2 that should be considered. Default=1.
Cex.Letters	The size of the symbols for the counterfactuals (S_0, S_1), T_0, T_1). Default=3.
Cex.Corrs	The size of the text depicting the median odds ratios between the counterfactuals. Default=2.
Lines.Rel.Width	Logical. When Lines.Rel.Width=TRUE, the widths of the lines that represent the odds ratios between the counterfactuals are relative to the size of the odds ratios (i.e., a smaller/thicker line is used for smaller/higher odds ratios. When Lines.Rel.Width=FALSE, the width of all lines representing the odds ratios between the counterfactuals is identical. Default=TRUE. Only considered when Values="ORs".
Col.Pos.Neg	Logical. When Col.Pos.Neg=TRUE, the color of the lines that represent the odds ratios between the counterfactuals is red for odds ratios below 1 and black for the ones above 1. When Col.Pos.Neg=FALSE, all lines are in black. Default=TRUE. Only considered when Values="ORs".
Monotonicity	Specifies the monotonicity scenario that should be considered (i.e., Monotonicity=c("No"), Monotonicity=c("True.Endp"), Monotonicity=c("Surr.Endp"), or Monotonicity=c("Surr.True.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., Molenberghs, G., Buyse, M., & Burzykowski, T. (submitted). On the relationship between the causal inference and meta-analytic paradigms for the validation of surrogate markers.

Van der Elst, W., Alonso, A., & Molenberghs, G. (submitted). An exploration of the relationship between causal inference and meta-analytic measures of surrogacy.

See Also

ICA.BinBin

Examples

```
# Compute R2_H given the marginals specified as the pi's
ICA <- ICA.BinBin.Grid.Sample(pi1_1_=0.2619048, pi1_0_=0.2857143,
pi_1_1=0.6372549, pi_1_0=0.07843137, pi0_1_=0.1349206, pi_0_1=0.127451,
Seed=1, Monotonicity=c("General"), M=1000)

# Obtain a causal diagram that provides the medians of the
# correlations between the counterfactuals for the range
# of R2_H values between 0.1 and 1
   # Assume no monotonicty
CausalDiagramBinBin(x=ICA, Min=0.1, Max=1, Monotonicity="No")

   # Assume monotonicty for S
CausalDiagramBinBin(x=ICA, Min=0.1, Max=1, Monotonicity="Surr.Endp")

# Now only consider the results that were obtained when
# monotonicity was assumed for the true endpoint
CausalDiagramBinBin(x=ICA, Values="ORs", Theta_T0S0=2.156, Theta_T1S1=10,
Min=0, Max=1,  Monotonicity="True.Endp")
```

`CausalDiagramContCont` *Draws a causal diagram depicting the median correlations between the counterfactuals for a specified range of values of ICA or MICA in the continuous-continuous setting*

Description

This function provides a diagram that depicts the medians of the correlations between the counterfactuals for a specified range of values of the individual causal association (ICA; ρ_Δ) or the meta-analytic individual causal association (MICA; ρ_M).

Usage

```
CausalDiagramContCont(x, Min=-1, Max=1, Cex.Letters=3, Cex.Corrs=2,
Lines.Rel.Width=TRUE, Col.Pos.Neg=TRUE)
```

Arguments

`x`	An object of class `ICA.ContCont` or `MICA.ContCont`. See `ICA.ContCont` or `MICA.ContCont`.
`Min`	The minimum values of (M)ICA that should be considered. Default=-1.
`Max`	The maximum values of (M)ICA that should be considered. Default=1.
`Cex.Letters`	The size of the symbols for the counterfactuals (S_0, S_1), T_0, T_1). Default=3.
`Cex.Corrs`	The size of the text depicting the median correlations between the counterfactuals. Default=2.

`Lines.Rel.Width`
: Logical. When `Lines.Rel.Width=TRUE`, the widths of the lines that represent the correlations between the counterfactuals are relative to the size of the correlations (i.e., a smaller line is used for correlations closer to zero whereas a thicker line is used for (absolute) correlations closer to 1). When `Lines.Rel.Width=FALSE`, the width of all lines representing the correlations between the counterfactuals is identical. Default=`TRUE`.

`Col.Pos.Neg`
: Logical. When `Col.Pos.Neg=TRUE`, the color of the lines that represent the correlations between the counterfactuals is red for negative correlations and black for positive ones. When `Col.Pos.Neg=FALSE`, all lines are in black. Default=`TRUE`.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., Molenberghs, G., Buyse, M., & Burzykowski, T. (submitted). On the relationship between the causal inference and meta-analytic paradigms for the validation of surrogate markers.

Van der Elst, W., Alonso, A., & Molenberghs, G. (submitted). An exploration of the relationship between causal inference and meta-analytic measures of surrogacy.

See Also

ICA.ContCont, MICA.ContCont

Examples

```
## Not run:  #Time consuming (>5 sec) code parts
# Generate the vector of ICA values when rho_T0S0=.91, rho_T1S1=.91, and when the
# grid of values {0, .1, ..., 1} is considered for the correlations
# between the counterfactuals:
SurICA <- ICA.ContCont(T0S0=.95, T1S1=.91, T0T1=seq(0, 1, by=.1), T0S1=seq(0, 1, by=.1),
T1S0=seq(0, 1, by=.1), S0S1=seq(0, 1, by=.1))

#obtain a plot of ICA

# Obtain a causal diagram that provides the medians of the
# correlations between the counterfactuals for the range
# of ICA values between .9 and 1 (i.e., which assumed
# correlations between the counterfactuals lead to a
# high ICA?)
CausalDiagramContCont(SurICA, Min=.9, Max=1)

# Same, for low values of ICA
CausalDiagramContCont(SurICA, Min=0, Max=.5)
## End(Not run)
```

CIGTS *Data of the Collaborative Initial Glaucoma Treatment Study*

Description

These are the data of the Collaborative Initial Glaucoma Treatment Study (CIGTS), a randomized clinical trial designed to compare the effects of surgery and medicine on intraocular pressure (IOP). Elevated IOP is an important risk factor for glaucoma, which is an occular disorder that leads to impaired vision. S and T are IOP at months 12 and 96, respectively. A total of $N = 228$ patients had IOP measured at both time points of interest. The continuous S and T were dichotomized in the following way: $S = 1$ if IOP at month 12 is < 18 mmHg and $S = 0$ if IOP at month 12 is $>= 18$ mmHg (and similarly for T).

Acknowledgment

David Musch (Coordinating Center Director) and Brenda Gillespie (Study Statistician) are gratefully acknowledged for providing data from the Collaborative Initial Glaucoma Treatment Study (CIGTS).

Usage

```
data(CIGTS)
```

Format

A `data.frame` with 228 observations on 4 variables.

`Id` The Patient ID.

`Treat` The treatment indicator, coded as -1 = medical treatment and 1 = surgery.

`IOP_12` The dichotomized IOP level 12 months following randomization.

`IOP_96` The dichotomized IOP level 96 months following randomization.

References

Musch, D. C., Lichter, P., R., Guire, K. E., Standardi, C. L., & CIGTS Investigators (1999). The Collaborative Initial Glaucoma Treatment Study (CIGTS): Study design, methods, and baseline characteristics of enrolled patients. Ophthalmology, 106, 653-662.

FixedBinBinIT *Fits (univariate) fixed-effect models to assess surrogacy in the binary-binary case based on the Information-Theoretic framework*

Description

The function FixedBinBinIT uses the information-theoretic approach (Alonso & Molenberghs, 2007) to estimate trial- and individual-level surrogacy based on fixed-effect models when both S and T are binary variables. The user can specify whether a (weighted or unweighted) full, semi-reduced, or reduced model should be fitted. See the **Details** section below.

Usage

```
FixedBinBinIT(Dataset, Surr, True, Treat, Trial.ID, Pat.ID,
Model=c("Full"), Weighted=TRUE, Min.Trial.Size=2, Alpha=.05,
Number.Bootstraps=50, Seed=sample(1:1000, size=1))
```

Arguments

Dataset	A data.frame that should consist of one line per patient. Each line contains (at least) a surrogate value, a true endpoint value, a treatment indicator, a patient ID, and a trial ID.
Surr	The name of the variable in Dataset that contains the surrogate endpoint values.
True	The name of the variable in Dataset that contains the true endpoint values.
Treat	The name of the variable in Dataset that contains the treatment indicators. The treatment indicator should either be coded as 1 for the experimental group and -1 for the control group, or as 1 for the experimental group and 0 for the control group.
Trial.ID	The name of the variable in Dataset that contains the trial ID to which the patient belongs.
Pat.ID	The name of the variable in Dataset that contains the patient's ID.
Model	The type of model that should be fitted, i.e., Model=c("Full"), Model=c("Reduced"), or Model=c("SemiReduced"). See the **Details** section below. Default Model=c("Full").
Weighted	Logical. In practice it is often the case that different trials (or other clustering units) have different sample sizes. Univariate models are used to assess surrogacy in the information-theoretic approach, so it can be useful to adjust for heterogeneity in information content between the trial-specific contributions (particularly when trial-level surrogacy measures are of primary interest and when the heterogeneity in sample sizes is large). If Weighted=TRUE, weighted regression models are fitted. If Weighted=FALSE, unweighted regression analyses are conducted. See the **Details** section below. Default TRUE.
Min.Trial.Size	The minimum number of patients that a trial should contain to be included in the analysis. If the number of patients in a trial is smaller than the value specified by Min.Trial.Size, the data of the trial are excluded from the analysis. Default 2.

`Alpha` The α-level that is used to determine the confidence intervals around R^2_h and R^2_{ht}. Default 0.05.

`Number.Bootstraps`
The standard errors and confidence intervals for R^2_h, $R^2_{b.ind}$ and $R^2_{h.ind}$ are determined based on a bootstrap procedure. `Number.Bootstraps` specifies the number of bootstrap samples that are used. Default 50.

`Seed` The seed to be used in the bootstrap procedure. Default $sample(1:1000, size = 1)$.

Details

Individual-level surrogacy

The following univariate generalised linear models are fitted:

$$g_T(E(T_{ij})) = \mu_{Ti} + \beta_i Z_{ij},$$

$$g_T(E(T_{ij}|S_{ij})) = \gamma_{0i} + \gamma_{1i} Z_{ij} + \gamma_{2i} S_{ij},$$

where i and j are the trial and subject indicators, g_T is an appropriate link function (i.e., a logit link when binary endpoints are considered), S_{ij} and T_{ij} are the surrogate and true endpoint values of subject j in trial i, and Z_{ij} is the treatment indicator for subject j in trial i. μ_{Ti} and β_i are the trial-specific intercepts and treatment-effects on the true endpoint in trial i. γ_{0i} and γ_{1i} are the trial-specific intercepts and treatment-effects on the true endpoint in trial i after accounting for the effect of the surrogate endpoint.

The -2 log likelihood values of the previous models in each of the i trials (i.e., L_{1i} and L_{2i}, respectively) are subsequently used to compute individual-level surrogacy based on the so-called Variance Reduction Factor (VFR; for details, see Alonso & Molenberghs, 2007):

$$R^2_h = 1 - \frac{1}{N} \sum_i exp\left(-\frac{L_{2i} - L_{1i}}{n_i}\right),$$

where N is the number of trials and n_i is the number of patients within trial i.

When it can be assumed (i) that the treatment-corrected association between the surrogate and the true endpoint is constant across trials, or (ii) when all data come from a single clinical trial (i.e., when $N = 1$), the previous expression simplifies to:

$$R^2_{h.ind} = 1 - exp\left(-\frac{L_2 - L_1}{N}\right).$$

The upper bound does not reach to 1 when T is binary, i.e., its maximum is 0.75. Kent (1983) claims that 0.75 is a reasonable upper bound and thus $R^2_{h.ind}$ can usually be interpreted without paying special consideration to the discreteness of T. Alternatively, to address the upper bound problem, a scaled version of the mutual information can be used when both S and T are binary (Joe, 1989):

$$R^2_{b.ind} = \frac{I(T,S)}{min[H(T), H(S)]},$$

where the entropy of T and S in the previous expression can be estimated using the log likelihood functions of the GLMs shown above.

Trial-level surrogacy

When a full or semi-reduced model is requested (by using the argument `Model=c("Full")` or `Model=c("SemiReduced")` in the function call), trial-level surrogacy is assessed by fitting the following univariate models:

$$S_{ij} = \mu_{Si} + \alpha_i Z_{ij} + \varepsilon_{Sij}, (1)$$
$$T_{ij} = \mu_{Ti} + \beta_i Z_{ij} + \varepsilon_{Tij}, (1)$$

where i and j are the trial and subject indicators, S_{ij} and T_{ij} are the surrogate and true endpoint values of subject j in trial i, Z_{ij} is the treatment indicator for subject j in trial i, μ_{Si} and μ_{Ti} are the fixed trial-specific intercepts for S and T, and α_i and β_i are the fixed trial-specific treatment effects on S and T, respectively. The error terms ε_{Sij} and ε_{Tij} are assumed to be independent.

When a reduced model is requested by the user (by using the argument `Model=c("Reduced")` in the function call), the following univariate models are fitted:

$$S_{ij} = \mu_S + \alpha_i Z_{ij} + \varepsilon_{Sij}, (2)$$
$$T_{ij} = \mu_T + \beta_i Z_{ij} + \varepsilon_{Tij}, (2)$$

where μ_S and μ_T are the common intercepts for S and T. The other parameters are the same as defined above, and ε_{Sij} and ε_{Tij} are again assumed to be independent.

When the user requested a full model approach (by using the argument `Model=c("Full")` in the function call, i.e., when models (1) were fitted), the following model is subsequently fitted:

$$\widehat{\beta}_i = \lambda_0 + \lambda_1 \widehat{\mu_{Si}} + \lambda_2 \widehat{\alpha}_i + \varepsilon_i, (3)$$

where the parameter estimates for β_i, μ_{Si}, and α_i are based on models (1) (see above). When a weighted model is requested (using the argument `Weighted=TRUE` in the function call), model (3) is a weighted regression model (with weights based on the number of observations in trial i). The -2 log likelihood value of the (weighted or unweighted) model (3) (L_1) is subsequently compared to the -2 log likelihood value of an intercept-only model ($\widehat{\beta}_i = \lambda_3$; L_0), and R^2_{ht} is computed based based on the Variance Reduction Factor (for details, see Alonso & Molenberghs, 2007):

$$R^2_{ht} = 1 - exp\left(-\frac{L_1 - L_0}{N}\right),$$

where N is the number of trials.

When a semi-reduced or reduced model is requested (by using the argument `Model=c("SemiReduced")` or `Model=c("Reduced")` in the function call), the following model is fitted:

$$\widehat{\beta}_i = \lambda_0 + \lambda_1 \widehat{\alpha}_i + \varepsilon_i,$$

where the parameter estimates for β_i and α_i are based on models (1) when a semi-reduced model is fitted or on models (2) when a reduced model is fitted. The -2 log likelihood value of this (weighted or unweighted) model (L_1) is subsequently compared to the -2 log likelihood value of an intercept-only model ($\widehat{\beta}_i = \lambda_3$; L_0), and R^2_{ht} is computed based on the reduction in the likelihood (as described above).

Value

An object of class `FixedBinBinIT` with components,

`Data.Analyze` Prior to conducting the surrogacy analysis, data of patients who have a missing value for the surrogate and/or the true endpoint are excluded. In addition, the data of trials (i) in which only one type of the treatment was administered, and (ii) in which either the surrogate or the true endpoint was a constant (i.e., all patients within a trial had the same surrogate and/or true endpoint value) are excluded. In addition, the user can specify the minimum number of patients that a trial should contain in order to include the trial in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded. `Data.Analyze` is the dataset on which the surrogacy analysis was conducted.

`Obs.Per.Trial` A `data.frame` that contains the total number of patients per trial and the number of patients who were administered the control treatment and the experimental treatment in each of the trials (in `Data.Analyze`).

`Trial.Spec.Results` A `data.frame` that contains the trial-specific intercepts and treatment effects for the surrogate and the true endpoints (when a full or semi-reduced model is requested), or the trial-specific treatment effects for the surrogate and the true endpoints (when a reduced model is requested).

`R2ht` A `data.frame` that contains the trial-level surrogacy estimate and its confidence interval.

`R2h.ind` A `data.frame` that contains the individual-level surrogacy estimate $R^2_{h.ind}$ (single-trial based estimate) and its confidence interval.

`R2h` A `data.frame` that contains the individual-level surrogacy estimate R^2_h (cluster-based estimate) and its confidence interval (based on a bootsrtrap).

`R2b.ind` A `data.frame` that contains the individual-level surrogacy estimate $R^2_{b.ind}$ (single-trial based estimate accounting for upper bound) and its confidence interval (based on a bootstrap).

`R2h.Ind.By.Trial` A `data.frame` that contains individual-level surrogacy estimates R^2_{hInd} (cluster-based estimates) and their confidence interval for each of the trials seperately.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A, & Molenberghs, G. (2007). Surrogate marker evaluation from an information theory perspective. *Biometrics, 63,* 180-186.

Joe, H. (1989). Relative entropy measures of multivariate dependence. *Journal of the American Statistical Association, 84,* 157-164.

Kent, T. J. (1983). Information gain as a general measure of correlation. *Biometrica, 70,* 163-173.

See Also

FixedBinContIT, FixedContBinIT, plot Information-Theoretic BinCombn

Examples

```
## Not run:  # Time consuming (>5sec) code part
# Generate data with continuous Surr and True
Sim.Data.MTS(N.Total=5000, N.Trial=50, R.Trial.Target=.9, R.Indiv.Target=.9,
             Fixed.Effects=c(0, 0, 0, 0), D.aa=10, D.bb=10, Seed=1,
             Model=c("Full"))
# Dichtomize Surr and True
Surr_Bin <- Data.Observed.MTS$Surr
Surr_Bin[Data.Observed.MTS$Surr>.5] <- 1
Surr_Bin[Data.Observed.MTS$Surr<=.5] <- 0
True_Bin <- Data.Observed.MTS$True
True_Bin[Data.Observed.MTS$True>.15] <- 1
True_Bin[Data.Observed.MTS$True<=.15] <- 0
Data.Observed.MTS$Surr <- Surr_Bin
Data.Observed.MTS$True <- True_Bin

# Assess surrogacy using info-theoretic framework
Fit <- FixedBinBinIT(Dataset = Data.Observed.MTS, Surr = Surr,
True = True, Treat = Treat, Trial.ID = Trial.ID,
Pat.ID = Pat.ID, Number.Bootstraps=100)

# Examine results
summary(Fit)
plot(Fit, Trial.Level = FALSE, Indiv.Level.By.Trial=TRUE)
plot(Fit, Trial.Level = TRUE, Indiv.Level.By.Trial=FALSE)

## End(Not run)
```

`FixedBinContIT` *Fits (univariate) fixed-effect models to assess surrogacy in the case where the true endpoint is binary and the surrogate endpoint is continuous (based on the Information-Theoretic framework)*

Description

The function `FixedBinContIT` uses the information-theoretic approach (Alonso & Molenberghs, 2007) to estimate trial- and individual-level surrogacy based on fixed-effect models when T is binary and S is continuous. The user can specify whether a (weighted or unweighted) full, semi-reduced, or reduced model should be fitted. See the **Details** section below.

Usage

```
FixedBinContIT(Dataset, Surr, True, Treat, Trial.ID, Pat.ID,
Model=c("Full"), Weighted=TRUE, Min.Trial.Size=2, Alpha=.05,
Number.Bootstraps=50,Seed=sample(1:1000, size=1))
```

Arguments

`Dataset`	A `data.frame` that should consist of one line per patient. Each line contains (at least) a surrogate value, a true endpoint value, a treatment indicator, a patient ID, and a trial ID.
`Surr`	The name of the variable in `Dataset` that contains the surrogate endpoint values.
`True`	The name of the variable in `Dataset` that contains the true endpoint values.
`Treat`	The name of the variable in `Dataset` that contains the treatment indicators. The treatment indicator should either be coded as 1 for the experimental group and -1 for the control group, or as 1 for the experimental group and 0 for the control group.
`Trial.ID`	The name of the variable in `Dataset` that contains the trial ID to which the patient belongs.
`Pat.ID`	The name of the variable in `Dataset` that contains the patient's ID.
`Model`	The type of model that should be fitted, i.e., `Model=c("Full")`, `Model=c("Reduced")`, or `Model=c("SemiReduced")`. See the **Details** section below. Default `Model=c("Full")`.
`Weighted`	Logical. In practice it is often the case that different trials (or other clustering units) have different sample sizes. Univariate models are used to assess surrogacy in the information-theoretic approach, so it can be useful to adjust for heterogeneity in information content between the trial-specific contributions (particularly when trial-level surrogacy measures are of primary interest and when the heterogeneity in sample sizes is large). If `Weighted=TRUE`, weighted regression models are fitted. If `Weighted=FALSE`, unweighted regression analyses are conducted. See the **Details** section below. Default `TRUE`.
`Min.Trial.Size`	The minimum number of patients that a trial should contain to be included in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded from the analysis. Default 2.
`Alpha`	The α-level that is used to determine the confidence intervals around R_h^2 and R_{ht}^2. Default 0.05.
`Number.Bootstraps`	The standard errors and confidence intervals for R_h^2 and $R_{h.ind}^2$ are determined based on a bootstrap procedure. `Number.Bootstraps` specifies the number of bootstrap samples that are used. Default 50.
`Seed`	The seed to be used in the bootstrap procedure. Default $sample(1:1000, size = 1)$.

Details

Individual-level surrogacy

The following univariate generalised linear models are fitted:

$$g_T(E(T_{ij})) = \mu_{Ti} + \beta_i Z_{ij},$$

$$g_T(E(T_{ij}|S_{ij})) = \gamma_{0i} + \gamma_{1i} Z_{ij} + \gamma_{2i} S_{ij},$$

where i and j are the trial and subject indicators, g_T is an appropriate link function (i.e., a logit link for binary endpoints and an identity link for normally distributed continuous endpoints), S_{ij} and T_{ij} are the surrogate and true endpoint values of subject j in trial i, and Z_{ij} is the treatment indicator for subject j in trial i. μ_{Ti} and β_i are the trial-specific intercepts and treatment-effects on the true endpoint in trial i. γ_{0i} and γ_{1i} are the trial-specific intercepts and treatment-effects on the true endpoint in trial i after accounting for the effect of the surrogate endpoint.

The -2 log likelihood values of the previous models in each of the i trials (i.e., L_{1i} and L_{2i}, respectively) are subsequently used to compute individual-level surrogacy based on the so-called Variance Reduction Factor (VFR; for details, see Alonso & Molenberghs, 2007):

$$R_h^2 = 1 - \frac{1}{N}\sum_i exp\left(-\frac{L_{2i} - L_{1i}}{n_i}\right),$$

where N is the number of trials and n_i is the number of patients within trial i.

When it can be assumed (i) that the treatment-corrected association between the surrogate and the true endpoint is constant across trials, or (ii) when all data come from a single clinical trial (i.e., when $N = 1$), the previous expression simplifies to:

$$R_{h.ind}^2 = 1 - exp\left(-\frac{L_2 - L_1}{N}\right).$$

The upper bound does not reach to 1 when T is binary, i.e., its maximum is 0.75. Kent (1983) claims that 0.75 is a reasonable upper bound and thus $R_{h.ind}^2$ can usually be interpreted without paying special consideration to the discreteness of T. Alternatively, to address the upper bound problem, a scaled version of the mutual information can be used when both S and T are binary (Joe, 1989):

$$R_{b.ind}^2 = \frac{I(T,S)}{min[H(T), H(S)]},$$

where the entropy of T and S in the previous expression can be estimated using the log likelihood functions of the GLMs shown above.

Trial-level surrogacy

When a full or semi-reduced model is requested (by using the argument `Model=c("Full")` or `Model=c("SemiReduced")` in the function call), trial-level surrogacy is assessed by fitting the following univariate models:

$$S_{ij} = \mu_{Si} + \alpha_i Z_{ij} + \varepsilon_{Sij}, (1)$$

$$T_{ij} = \mu_{Ti} + \beta_i Z_{ij} + \varepsilon_{Tij}, (1)$$

where i and j are the trial and subject indicators, S_{ij} and T_{ij} are the surrogate and true endpoint values of subject j in trial i, Z_{ij} is the treatment indicator for subject j in trial i, μ_{Si} and μ_{Ti} are the fixed trial-specific intercepts for S and T, and α_i and β_i are the fixed trial-specific treatment effects on S and T, respectively. The error terms ε_{Sij} and ε_{Tij} are assumed to be independent.

When a reduced model is requested by the user (by using the argument `Model=c("Reduced")` in the function call), the following univariate models are fitted:

$$S_{ij} = \mu_S + \alpha_i Z_{ij} + \varepsilon_{Sij}, (2)$$

$$T_{ij} = \mu_T + \beta_i Z_{ij} + \varepsilon_{Tij}, (2)$$

where μ_S and μ_T are the common intercepts for S and T. The other parameters are the same as defined above, and ε_{Sij} and ε_{Tij} are again assumed to be independent.

When the user requested a full model approach (by using the argument `Model=c("Full")` in the function call, i.e., when models (1) were fitted), the following model is subsequently fitted:

$$\widehat{\beta}_i = \lambda_0 + \lambda_1 \widehat{\mu_{Si}} + \lambda_2 \widehat{\alpha}_i + \varepsilon_i, (3)$$

where the parameter estimates for β_i, μ_{Si}, and α_i are based on models (1) (see above). When a weighted model is requested (using the argument `Weighted=TRUE` in the function call), model (3) is a weighted regression model (with weights based on the number of observations in trial i). The -2 log likelihood value of the (weighted or unweighted) model (3) (L_1) is subsequently compared to the -2 log likelihood value of an intercept-only model ($\widehat{\beta}_i = \lambda_3$; L_0), and R^2_{ht} is computed based based on the Variance Reduction Factor (for details, see Alonso & Molenberghs, 2007):

$$R^2_{ht} = 1 - exp\left(-\frac{L_1 - L_0}{N}\right),$$

where N is the number of trials.

When a semi-reduced or reduced model is requested (by using the argument `Model=c("SemiReduced")` or `Model=c("Reduced")` in the function call), the following model is fitted:

$$\widehat{\beta}_i = \lambda_0 + \lambda_1 \widehat{\alpha}_i + \varepsilon_i,$$

where the parameter estimates for β_i and α_i are based on models (1) when a semi-reduced model is fitted or on models (2) when a reduced model is fitted. The -2 log likelihood value of this (weighted or unweighted) model (L_1) is subsequently compared to the -2 log likelihood value of an intercept-only model ($\widehat{\beta}_i = \lambda_3$; L_0), and R^2_{ht} is computed based on the reduction in the likelihood (as described above).

Value

An object of class `FixedBinContIT` with components,

`Data.Analyze` Prior to conducting the surrogacy analysis, data of patients who have a missing value for the surrogate and/or the true endpoint are excluded. In addition, the data of trials (i) in which only one type of the treatment was administered, and (ii) in which either the surrogate or the true endpoint was a constant (i.e., all patients within a trial had the same surrogate and/or true endpoint value) are excluded. In addition, the user can specify the minimum number of patients that a trial should contain in order to include the trial in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded. `Data.Analyze` is the dataset on which the surrogacy analysis was conducted.

`Obs.Per.Trial` A `data.frame` that contains the total number of patients per trial and the number of patients who were administered the control treatment and the experimental treatment in each of the trials (in `Data.Analyze`).

`Trial.Spec.Results` A `data.frame` that contains the trial-specific intercepts and treatment effects for the surrogate and the true endpoints (when a full or semi-reduced model is requested), or the trial-specific treatment effects for the surrogate and the true endpoints (when a reduced model is requested).

`R2ht` A `data.frame` that contains the trial-level surrogacy estimate and its confidence interval.

`R2h.ind` A `data.frame` that contains the individual-level surrogacy estimate $R^2_{h.ind}$ (single-trial based estimate) and its confidence interval.

`R2h` A `data.frame` that contains the individual-level surrogacy estimate R^2_h (cluster-based estimate) and its confidence interval (bootstrap-based).

`R2b.ind` A `data.frame` that contains the individual-level surrogacy estimate $R^2_{b.ind}$ (single-trial based estimate accounting for upper bound) and its confidence interval (based on a bootstrap).

`R2h.Ind.By.Trial` A `data.frame` that contains individual-level surrogacy estimates R^2_h (cluster-based estimate) and their confidence interval for each of the trials seperately.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A, & Molenberghs, G. (2007). Surrogate marker evaluation from an information theory perspective. *Biometrics, 63,* 180-186.

Joe, H. (1989). Relative entropy measures of multivariate dependence. *Journal of the American Statistical Association, 84,* 157-164.

Kent, T. J. (1983). Information gain as a general measure of correlation. *Biometrica, 70,* 163-173.

See Also

`FixedBinBinIT`, FixedContBinIT, `plot Information-Theoretic BinCombn`

Examples

```
## Not run:  # Time consuming (>5sec) code part
# Generate data with continuous Surr and True
Sim.Data.MTS(N.Total=2000, N.Trial=100, R.Trial.Target=.8,
R.Indiv.Target=.8, Seed=123, Model="Full")

# Make T binary
Data.Observed.MTS$True_Bin <- Data.Observed.MTS$True
Data.Observed.MTS$True_Bin[Data.Observed.MTS$True>=0] <- 1
Data.Observed.MTS$True_Bin[Data.Observed.MTS$True<0] <- 0
```

```
# Analyze data
Fit <- FixedBinContIT(Dataset = Data.Observed.MTS, Surr = Surr,
True = True_Bin, Treat = Treat, Trial.ID = Trial.ID, Pat.ID = Pat.ID,
Model = "Full", Number.Bootstraps=50)

# Examine results
summary(Fit)
plot(Fit, Trial.Level = FALSE, Indiv.Level.By.Trial=TRUE)
plot(Fit, Trial.Level = TRUE, Indiv.Level.By.Trial=FALSE)

## End(Not run)
```

`FixedContBinIT` *Fits (univariate) fixed-effect models to assess surrogacy in the case where the true endpoint is continuous and the surrogate endpoint is binary (based on the Information-Theoretic framework)*

Description

The function `FixedContBinIT` uses the information-theoretic approach (Alonso & Molenberghs, 2007) to estimate trial- and individual-level surrogacy based on fixed-effect models when T is continuous normally distributed and S is binary. The user can specify whether a (weighted or unweighted) full, semi-reduced, or reduced model should be fitted. See the **Details** section below.

Usage

```
FixedContBinIT(Dataset, Surr, True, Treat, Trial.ID, Pat.ID,
Model=c("Full"), Weighted=TRUE, Min.Trial.Size=2, Alpha=.05,
Number.Bootstraps=50,Seed=sample(1:1000, size=1))
```

Arguments

`Dataset`	A `data.frame` that should consist of one line per patient. Each line contains (at least) a surrogate value, a true endpoint value, a treatment indicator, a patient ID, and a trial ID.
`Surr`	The name of the variable in `Dataset` that contains the surrogate endpoint values.
`True`	The name of the variable in `Dataset` that contains the true endpoint values.
`Treat`	The name of the variable in `Dataset` that contains the treatment indicators. The treatment indicator should either be coded as 1 for the experimental group and -1 for the control group, or as 1 for the experimental group and 0 for the control group.
`Trial.ID`	The name of the variable in `Dataset` that contains the trial ID to which the patient belongs.
`Pat.ID`	The name of the variable in `Dataset` that contains the patient's ID.
`Model`	The type of model that should be fitted, i.e., `Model=c("Full")`, `Model=c("Reduced")`, or `Model=c("SemiReduced")`. See the **Details** section below. Default `Model=c("Full")`.

`Weighted` Logical. In practice it is often the case that different trials (or other clustering units) have different sample sizes. Univariate models are used to assess surrogacy in the information-theoretic approach, so it can be useful to adjust for heterogeneity in information content between the trial-specific contributions (particularly when trial-level surrogacy measures are of primary interest and when the heterogeneity in sample sizes is large). If `Weighted=TRUE`, weighted regression models are fitted. If `Weighted=FALSE`, unweighted regression analyses are conducted. See the **Details** section below. Default `TRUE`.

`Min.Trial.Size` The minimum number of patients that a trial should contain to be included in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded from the analysis. Default 2.

`Alpha` The α-level that is used to determine the confidence intervals around R_h^2 and R_{ht}^2. Default 0.05.

`Number.Bootstraps`
The standard error and confidence interval for $R_{h.ind}^2$ is determined based on a bootstrap procedure. `Number.Bootstraps` specifies the number of bootstrap samples that are used. Default 50.

`Seed` The seed to be used in the bootstrap procedure. Default $sample(1:1000, size = 1)$.

Details

Individual-level surrogacy

The following univariate generalised linear models are fitted:

$$g_T(E(T_{ij})) = \mu_{Ti} + \beta_i Z_{ij},$$

$$g_T(E(T_{ij}|S_{ij})) = \gamma_{0i} + \gamma_{1i} Z_{ij} + \gamma_{2i} S_{ij},$$

where i and j are the trial and subject indicators, g_T is an appropriate link function (i.e., a logit link for binary endpoints and an identity link for normally distributed continuous endpoints), S_{ij} and T_{ij} are the surrogate and true endpoint values of subject j in trial i, and Z_{ij} is the treatment indicator for subject j in trial i. μ_{Ti} and β_i are the trial-specific intercepts and treatment-effects on the true endpoint in trial i. γ_{0i} and γ_{1i} are the trial-specific intercepts and treatment-effects on the true endpoint in trial i after accounting for the effect of the surrogate endpoint.

The -2 log likelihood values of the previous models in each of the i trials (i.e., L_{1i} and L_{2i}, respectively) are subsequently used to compute individual-level surrogacy based on the so-called Variance Reduction Factor (VFR; for details, see Alonso & Molenberghs, 2007):

$$R_h^2 = 1 - \frac{1}{N} \sum_i exp\left(-\frac{L_{2i} - L_{1i}}{n_i}\right),$$

where N is the number of trials and n_i is the number of patients within trial i.

When it can be assumed (i) that the treatment-corrected association between the surrogate and the true endpoint is constant across trials, or (ii) when all data come from a single clinical trial (i.e., when $N = 1$), the previous expression simplifies to:

$$R^2_{h.ind} = 1 - exp\left(-\frac{L_2 - L_1}{N}\right).$$

Trial-level surrogacy

When a full or semi-reduced model is requested (by using the argument `Model=c("Full")` or `Model=c("SemiReduced")` in the function call), trial-level surrogacy is assessed by fitting the following univariate models:

$$S_{ij} = \mu_{Si} + \alpha_i Z_{ij} + \varepsilon_{Sij}, (1)$$

$$T_{ij} = \mu_{Ti} + \beta_i Z_{ij} + \varepsilon_{Tij}, (1)$$

where i and j are the trial and subject indicators, S_{ij} and T_{ij} are the surrogate and true endpoint values of subject j in trial i, Z_{ij} is the treatment indicator for subject j in trial i, μ_{Si} and μ_{Ti} are the fixed trial-specific intercepts for S and T, and α_i and β_i are the fixed trial-specific treatment effects on S and T, respectively. The error terms ε_{Sij} and ε_{Tij} are assumed to be independent.

When a reduced model is requested by the user (by using the argument `Model=c("Reduced")` in the function call), the following univariate models are fitted:

$$S_{ij} = \mu_S + \alpha_i Z_{ij} + \varepsilon_{Sij}, (2)$$

$$T_{ij} = \mu_T + \beta_i Z_{ij} + \varepsilon_{Tij}, (2)$$

where μ_S and μ_T are the common intercepts for S and T. The other parameters are the same as defined above, and ε_{Sij} and ε_{Tij} are again assumed to be independent.

When the user requested a full model approach (by using the argument `Model=c("Full")` in the function call, i.e., when models (1) were fitted), the following model is subsequently fitted:

$$\widehat{\beta}_i = \lambda_0 + \lambda_1 \widehat{\mu_{Si}} + \lambda_2 \widehat{\alpha}_i + \varepsilon_i, (3)$$

where the parameter estimates for β_i, μ_{Si}, and α_i are based on models (1) (see above). When a weighted model is requested (using the argument `Weighted=TRUE` in the function call), model (3) is a weighted regression model (with weights based on the number of observations in trial i). The -2 log likelihood value of the (weighted or unweighted) model (3) (L_1) is subsequently compared to the -2 log likelihood value of an intercept-only model ($\widehat{\beta}_i = \lambda_3$; L_0), and R^2_{ht} is computed based based on the Variance Reduction Factor (for details, see Alonso & Molenberghs, 2007):

$$R^2_{ht} = 1 - exp\left(-\frac{L_1 - L_0}{N}\right),$$

where N is the number of trials.

When a semi-reduced or reduced model is requested (by using the argument `Model=c("SemiReduced")` or `Model=c("Reduced")` in the function call), the following model is fitted:

$$\widehat{\beta}_i = \lambda_0 + \lambda_1 \widehat{\alpha}_i + \varepsilon_i,$$

where the parameter estimates for β_i and α_i are based on models (1) when a semi-reduced model is fitted or on models (2) when a reduced model is fitted. The -2 log likelihood value of this (weighted or unweighted) model (L_1) is subsequently compared to the -2 log likelihood value of an intercept-only model ($\widehat{\beta}_i = \lambda_3$; L_0), and R^2_{ht} is computed based on the reduction in the likelihood (as described above).

Value

An object of class `FixedContBinIT` with components,

`Data.Analyze` Prior to conducting the surrogacy analysis, data of patients who have a missing value for the surrogate and/or the true endpoint are excluded. In addition, the data of trials (i) in which only one type of the treatment was administered, and (ii) in which either the surrogate or the true endpoint was a constant (i.e., all patients within a trial had the same surrogate and/or true endpoint value) are excluded. In addition, the user can specify the minimum number of patients that a trial should contain in order to include the trial in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded. `Data.Analyze` is the dataset on which the surrogacy analysis was conducted.

`Obs.Per.Trial` A `data.frame` that contains the total number of patients per trial and the number of patients who were administered the control treatment and the experimental treatment in each of the trials (in `Data.Analyze`).

`Trial.Spec.Results`
A `data.frame` that contains the trial-specific intercepts and treatment effects for the surrogate and the true endpoints (when a full or semi-reduced model is requested), or the trial-specific treatment effects for the surrogate and the true endpoints (when a reduced model is requested).

`R2ht` A `data.frame` that contains the trial-level surrogacy estimate and its confidence interval.

`R2h` A `data.frame` that contains the individual-level surrogacy estimate R^2_h (cluster-based estimate) and its confidence interval.

`R2h.ind` A `data.frame` that contains the individual-level surrogacy estimate $R^2_{h.ind}$ (single-trial based estimate) and its confidence interval based on a bootstrap. The $R^2_{h.ind}$ shown is the mean of the bootstrapped values.

`R2h.Ind.By.Trial`
A `data.frame` that contains individual-level surrogacy estimates R^2_h (cluster-based estimate) and their confidence interval for each of the trials seperately.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A, & Molenberghs, G. (2007). Surrogate marker evaluation from an information theory perspective. *Biometrics, 63,* 180-186.

See Also

FixedBinBinIT, FixedBinContIT, plot Information-Theoretic BinCombn

Examples

```
## Not run:  # Time consuming (>5sec) code part
# Generate data with continuous Surr and True
Sim.Data.MTS(N.Total=2000, N.Trial=100, R.Trial.Target=.8,
R.Indiv.Target=.8, Seed=123, Model="Full")

# Make S binary
Data.Observed.MTS$Surr_Bin <- Data.Observed.MTS$Surr
Data.Observed.MTS$Surr_Bin[Data.Observed.MTS$Surr>=0] <- 1
Data.Observed.MTS$Surr_Bin[Data.Observed.MTS$Surr<0] <- 0

# Analyze data
Fit <- FixedContBinIT(Dataset = Data.Observed.MTS, Surr = Surr_Bin,
True = True, Treat = Treat, Trial.ID = Trial.ID, Pat.ID = Pat.ID,
Model = "Full", Number.Bootstraps=50)

# Examine results
summary(Fit)
plot(Fit, Trial.Level = FALSE, Indiv.Level.By.Trial=TRUE)
plot(Fit, Trial.Level = TRUE, Indiv.Level.By.Trial=FALSE)

## End(Not run)
```

FixedContContIT *Fits (univariate) fixed-effect models to assess surrogacy in the continuous-continuous case based on the Information-Theoretic framework*

Description

The function FixedContContIT uses the information-theoretic approach (Alonso & Molenberghs, 2007) to estimate trial- and individual-level surrogacy based on fixed-effect models when both S and T are continuous variables. The user can specify whether a (weighted or unweighted) full, semi-reduced, or reduced model should be fitted. See the **Details** section below.

Usage

```
FixedContContIT(Dataset, Surr, True, Treat, Trial.ID, Pat.ID,
Model=c("Full"), Weighted=TRUE, Min.Trial.Size=2,
Alpha=.05, Number.Bootstraps=500, Seed=sample(1:1000, size=1))
```

Arguments

`Dataset`	A `data.frame` that should consist of one line per patient. Each line contains (at least) a surrogate value, a true endpoint value, a treatment indicator, a patient ID, and a trial ID.
`Surr`	The name of the variable in `Dataset` that contains the surrogate endpoint values.
`True`	The name of the variable in `Dataset` that contains the true endpoint values.
`Treat`	The name of the variable in `Dataset` that contains the treatment indicators. The treatment indicator should either be coded as 1 for the experimental group and -1 for the control group, or as 1 for the experimental group and 0 for the control group.
`Trial.ID`	The name of the variable in `Dataset` that contains the trial ID to which the patient belongs.
`Pat.ID`	The name of the variable in `Dataset` that contains the patient's ID.
`Model`	The type of model that should be fitted, i.e., `Model=c("Full")`, `Model=c("Reduced")`, or `Model=c("SemiReduced")`. See the **Details** section below. Default `Model=c("Full")`.
`Weighted`	Logical. In practice it is often the case that different trials (or other clustering units) have different sample sizes. Univariate models are used to assess surrogacy in the information-theoretic approach, so it can be useful to adjust for heterogeneity in information content between the trial-specific contributions (particularly when trial-level surrogacy measures are of primary interest and when the heterogeneity in sample sizes is large). If `Weighted=TRUE`, weighted regression models are fitted. If `Weighted=FALSE`, unweighted regression analyses are conducted. See the **Details** section below. Default `TRUE`.
`Min.Trial.Size`	The minimum number of patients that a trial should contain to be included in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded from the analysis. Default 2.
`Alpha`	The α-level that is used to determine the confidence intervals around R_h^2 and R_{ht}^2. Default 0.05.
`Number.Bootstraps`	The standard error and confidence interval for R_h^2 is determined based on a bootstrap procedure. `Number.Bootstraps` specifies the number of bootstrap samples that are used. Default 500.
`Seed`	The seed to be used in the bootstrap procedure. Default $sample(1:1000, size = 1)$.

Details

Individual-level surrogacy

The following univariate generalised linear models are fitted:

$$g_T(E(T_{ij})) = \mu_{Ti} + \beta_i Z_{ij},$$

$$g_T(E(T_{ij}|S_{ij})) = \gamma_{0i} + \gamma_{1i} Z_{ij} + \gamma_{2i} S_{ij},$$

where i and j are the trial and subject indicators, g_T is an appropriate link function (i.e., an identity link when a continuous true endpoint is considered), S_{ij} and T_{ij} are the surrogate and true endpoint values of subject j in trial i, and Z_{ij} is the treatment indicator for subject j in trial i. μ_{Ti} and β_i are the trial-specific intercepts and treatment-effects on the true endpoint in trial i. γ_{0i} and γ_{1i} are the trial-specific intercepts and treatment-effects on the true endpoint in trial i after accounting for the effect of the surrogate endpoint.

The -2 log likelihood values of the previous models in each of the i trials (i.e., L_{1i} and L_{2i}, respectively) are subsequently used to compute individual-level surrogacy based on the so-called Variance Reduction Factor (VFR; for details, see Alonso & Molenberghs, 2007):

$$R^2_{h.ind} = 1 - \frac{1}{N}\sum_i exp\left(-\frac{L_{2i} - L_{1i}}{n_i}\right),$$

where N is the number of trials and n_i is the number of patients within trial i.

When it can be assumed (i) that the treatment-corrected association between the surrogate and the true endpoint is constant across trials, or (ii) when all data come from a single clinical trial (i.e., when $N = 1$), the previous expression simplifies to:

$$R^2_{h.ind.clust} = 1 - exp\left(-\frac{L_2 - L_1}{N}\right).$$

Trial-level surrogacy

When a full or semi-reduced model is requested (by using the argument `Model=c("Full")` or `Model=c("SemiReduced")` in the function call), trial-level surrogacy is assessed by fitting the following univariate models:

$$S_{ij} = \mu_{Si} + \alpha_i Z_{ij} + \varepsilon_{Sij}, (1)$$

$$T_{ij} = \mu_{Ti} + \beta_i Z_{ij} + \varepsilon_{Tij}, (1)$$

where i and j are the trial and subject indicators, S_{ij} and T_{ij} are the surrogate and true endpoint values of subject j in trial i, Z_{ij} is the treatment indicator for subject j in trial i, μ_{Si} and μ_{Ti} are the fixed trial-specific intercepts for S and T, and α_i and β_i are the fixed trial-specific treatment effects on S and T, respectively. The error terms ε_{Sij} and ε_{Tij} are assumed to be independent.

When a reduced model is requested by the user (by using the argument `Model=c("Reduced")` in the function call), the following univariate models are fitted:

$$S_{ij} = \mu_S + \alpha_i Z_{ij} + \varepsilon_{Sij}, (2)$$

$$T_{ij} = \mu_T + \beta_i Z_{ij} + \varepsilon_{Tij}, (2)$$

where μ_S and μ_T are the common intercepts for S and T. The other parameters are the same as defined above, and ε_{Sij} and ε_{Tij} are again assumed to be independent.

When the user requested a full model approach (by using the argument `Model=c("Full")` in the function call, i.e., when models (1) were fitted), the following model is subsequently fitted:

$$\widehat{\beta}_i = \lambda_0 + \lambda_1\widehat{\mu_{Si}} + \lambda_2\widehat{\alpha}_i + \varepsilon_i, (3)$$

where the parameter estimates for β_i, μ_{Si}, and α_i are based on models (1) (see above). When a weighted model is requested (using the argument `Weighted=TRUE` in the function call), model (3) is a weighted regression model (with weights based on the number of observations in trial i). The -2 log likelihood value of the (weighted or unweighted) model (3) (L_1) is subsequently compared to the -2 log likelihood value of an intercept-only model ($\widehat{\beta}_i = \lambda_3$; L_0), and R^2_{ht} is computed based based on the Variance Reduction Factor (for details, see Alonso & Molenberghs, 2007):

$$R^2_{ht} = 1 - exp\left(-\frac{L_1 - L_0}{N}\right),$$

where N is the number of trials.

When a semi-reduced or reduced model is requested (by using the argument `Model=c("SemiReduced")` or `Model=c("Reduced")` in the function call), the following model is fitted:

$$\widehat{\beta}_i = \lambda_0 + \lambda_1 \widehat{\alpha}_i + \varepsilon_i,$$

where the parameter estimates for β_i and α_i are based on models (1) when a semi-reduced model is fitted or on models (2) when a reduced model is fitted. The -2 log likelihood value of this (weighted or unweighted) model (L_1) is subsequently compared to the -2 log likelihood value of an intercept-only model ($\widehat{\beta}_i = \lambda_3$; L_0), and R^2_{ht} is computed based on the reduction in the likelihood (as described above).

Value

An object of class `FixedContContIT` with components,

`Data.Analyze` Prior to conducting the surrogacy analysis, data of patients who have a missing value for the surrogate and/or the true endpoint are excluded. In addition, the data of trials (i) in which only one type of the treatment was administered, and (ii) in which either the surrogate or the true endpoint was a constant (i.e., all patients within a trial had the same surrogate and/or true endpoint value) are excluded. In addition, the user can specify the minimum number of patients that a trial should contain in order to include the trial in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded. `Data.Analyze` is the dataset on which the surrogacy analysis was conducted.

`Obs.Per.Trial` A `data.frame` that contains the total number of patients per trial and the number of patients who were administered the control treatment and the experimental treatment in each of the trials (in `Data.Analyze`).

`Trial.Spec.Results` A `data.frame` that contains the trial-specific intercepts and treatment effects for the surrogate and the true endpoints (when a full or semi-reduced model is requested), or the trial-specific treatment effects for the surrogate and the true endpoints (when a reduced model is requested).

`R2ht` A `data.frame` that contains the trial-level surrogacy estimate and its confidence interval.

`R2h.ind.clust` A `data.frame` that contains the individual-level surrogacy estimate and its confidence interval.

R2h.ind	A data.frame that contains the individual-level surrogacy estimate and its confidence interval under the assumption that the treatment-corrected association between the surrogate and the true endpoints is constant across trials or when all data come from a single clinical trial.
Boot.CI	A data.frame that contains the bootstrapped R2h.Single values.
Cor.Endpoints	A data.frame that contains the correlations between the surrogate and the true endpoint in the control treatment group (i.e., ρ_{T0S0}) and in the experimental treatment group (i.e., ρ_{T1S1}), their standard errors and their confidence intervals.
Residuals	A data.frame that contains the residuals for the surrogate and true endpoints (ε_{Sij} and ε_{Tij}) that are obtained when models (1) or models (2) are fitted (see the **Details** section above).

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A, & Molenberghs, G. (2007). Surrogate marker evaluation from an information theory perspective. *Biometrics, 63,* 180-186.

See Also

MixedContContIT, FixedContBinIT, FixedBinContIT, FixedBinBinIT, plot Information-Theoretic

Examples

```
# Example 1
# Based on the ARMD data

data(ARMD)
# Assess surrogacy based on a full fixed-effect model
# in the information-theoretic framework:
Sur <- FixedContContIT(Dataset=ARMD, Surr=Diff24, True=Diff52, Treat=Treat, Trial.ID=Center,
Pat.ID=Id, Model="Full", Number.Bootstraps=50)
# Obtain a summary of the results:
summary(Sur)

## Not run:  #time consuming code
# Example 2
# Conduct an analysis based on a simulated dataset with 2000 patients, 100 trials,
# and Rindiv=Rtrial=.8

# Simulate the data:
Sim.Data.MTS(N.Total=2000, N.Trial=100, R.Trial.Target=.8, R.Indiv.Target=.8,
             Seed=123, Model="Full")
# Assess surrogacy based on a full fixed-effect model
# in the information-theoretic framework:
Sur2 <- FixedContContIT(Dataset=Data.Observed.MTS, Surr=Surr, True=True, Treat=Treat,
Trial.ID=Trial.ID, Pat.ID=Pat.ID, Model="Full", Number.Bootstraps=50)
```

```
# Show a summary of the results:
summary(Sur2)
## End(Not run)
```

`FixedDiscrDiscrIT` *Investigates surrogacy for binary or ordinal outcomes using the Information Theoretic framework*

Description

The function `FixedDiscrDiscrIT` uses the information theoretic approach (Alonso and Molenberghs 2007) to estimate trial and individual level surrogacy based on fixed-effects models when the surrogate is binary and the true outcome is ordinal, the converse case or when both outcomes are ordinal (the user must specify which form the data is in). The user can specify whether a weighted or unweighted analysis is required at the trial level. The penalized likelihood approach of Firth (1993) is applied to resolve issues of separation in discrete outcomes for particular trials. Requires packages `OrdinalLogisticBiplot` and `logistf`.

Usage

```
FixedDiscrDiscrIT(Dataset, Surr, True, Treat, Trial.ID,
Weighted = TRUE, Setting = c("binord"))
```

Arguments

`Dataset`	A `data.frame` that should consist of one line per patient. Each line contains (at least) a surrogate value, a true outcome value, a treatment indicator and a trial ID.
`Surr`	The name of the variable in `Dataset` that contains the surrogate outcome values.
`True`	The name of the variable in `Dataset` that contains the true outcome values.
`Treat`	The name of the in `Dataset` that contains the treatment group values, 0/1 or -1/+1 are recommended.
`Trial.ID`	The name of the variable in `Dataset` that contains the trial ID to which the patient belongs.
`Weighted`	Logical. In practice it is often the case that different trials (or other clustering units) have different sample sizes. Univariate models are used to assess surrogacy in the information-theoretic approach, so it can be useful to adjust for heterogeneity in information content between the trial-specific contributions (particularly when trial-level surrogacy measures are of primary interest and when the heterogeneity in sample sizes is large). If `Weighted=TRUE`, weighted regression models are fitted. If `Weighted=FALSE`, unweighted regression analyses are conducted. See the **Details** section below. Default `TRUE`.
`Setting`	Specifies whether an ordinal or binary surrogate or true outcome are present in `Dataset`. `Setting=c("binord")` for a binary surrogate and ordinal true outcome, `Setting=c("ordbin")` for an ordinal surrogate and binary true outcome and `Setting=c("ordord")` where both outcomes are ordinal.

Details

Individual level surrogacy

The following univariate logistic regression models are fitted when `Setting=c("ordbin")`:

$$logit(P(T_{ij} = 1)) = \mu_{Ti} + \beta_i Z_{ij}, (1)$$

$$logit(P(T_{ij} = 1|S_{ij} = s)) = \gamma_{0i} + \gamma_{1i} Z_{ij} + \gamma_{2i} S_{ij}, (1)$$

where: i and j are the trial and subject indicators; S_{ij} and T_{ij} are the surrogate and true outcome values of subject j in trial i; and Z_{ij} is the treatment indicator for subject j in trial i; μ_{Ti} and β_i are the trial-specific intercepts and treatment-effects on the true endpoint in trial i; and γ_{0i} and γ_{1i} are the trial-specific intercepts and treatment-effects on the true endpoint in trial i after accounting for the effect of the surrogate endpoint. The -2 log likelihood values of the previous models in each of the i trials (i.e., L_{1i} and L_{2i}, respectively) are subsequently used to compute individual-level surrogacy based on the so-called Likelihood Reduction Factor (LRF; for details, see Alonso & Molenberghs, 2006):

$$R_h^2 = 1 - \frac{1}{N} \sum_i exp\left(-\frac{L_{2i} - L_{1i}}{n_i}\right),$$

where N is the number of trials and n_i is the number of patients within trial i.

At the individual level in the discrete case R_h^2 is bounded above by a number strictly less than one and is re-scaled (see Alonso & Molenberghs (2007)):

$$\widehat{R_h^2} = \frac{R_h^2}{1 - e^{-2L_0}},$$

where L_0 is the log-likelihood of the intercept only model of the true outcome ($logit(P(T_{ij} = 1) = \gamma_3$).

In the case of `Setting=c("binord")` or `Setting=c("ordord")` proportional odds models in (1) are used to accommodate the ordinal true response outcome, in all other respects the calculation of R_h^2 would proceed in the same manner.

Trial-level surrogacy

When `Setting=c("ordbin")` trial-level surrogacy is assessed by fitting the following univariate logistic regression and proportional odds models for the ordinal surrogate and binary true response variables regressed on treatment for each trial i:

$$logit(P(S_{ij} \leq W)) = \mu_{S_{wi}} + \alpha_i Z_{ij}, (2)$$

$$logit(P(T_{ij} = 1)) = \mu_{Ti} + \beta_i Z_{ij}, (2)$$

where: i and j are the trial and subject indicators; S_{ij} and T_{ij} are the surrogate and true outcome values of subject j in trial i; Z_{ij} is the treatment indicator for subject j in trial i; $\mu_{S_{wi}}$ are the trial-specific intercept values for each cut point w, where $w = 1, .., W - 1$, of the ordinal surrogate outcome; μ_{Ti} are the fixed trial-specific intercepts for T; and α_i and β_i are the fixed trial-specific treatment effects on S and T, respectively. The mean trial-specific intercepts for the surrogate are calculated, $\overline{\mu}_{S_{wi}}$.The following model is subsequently fitted:

$$\widehat{\beta}_i = \lambda_0 + \lambda_1 \widehat{\overline{\mu}}_{S_{wi}} + \lambda_2 \widehat{\alpha}_i + \varepsilon_i, (3)$$

where the parameter estimates for β_i, $\overline{\mu}_{S_{wi}}$, and α_i are based on models (2) (see above). When a weighted model is requested (using the argument `Weighted=TRUE` in the function call), model (2) is a weighted regression model (with weights based on the number of observations in trial i). The -2 log likelihood value of the (weighted or unweighted) model (2) (L_1) is subsequently compared to the -2 log likelihood value of an intercept-only model ($\widehat{\beta}_i = \lambda_3$; L_0), and R^2_{ht} is computed based on the Likelihood Reduction Factor (for details, see Alonso & Molenberghs, 2006):

$$R^2_{ht} = 1 - exp\left(-\frac{L_1 - L_0}{N}\right),$$

where N is the number of trials.

When separation (the presence of zero cells) occurs in the cross tabs of treatment and the true or surrogate outcome for a particular trial in models (2) extreme bias can occur in R^2_{ht}. Under separation there are no unique maximum likelihood for parameters β_i, $\overline{\mu}_{S_{wi}}$ and α_i, in (2), for the affected trial i. This typically leads to extreme bias in the estimation of these parameters and hence outlying influential points in model (3), bias in R^2_{ht} inevitably follows.

To resolve the issue of separation the penalized likelihood approach of Firth (1993) is applied. This approach adds an asymptotically negligible component to the score function to allow unbiased estimation of β_i, $\overline{\mu}_{S_{wi}}$, and α_i and in turn R^2_{ht}. The penalized likelihood R function `logitf` from the package of the same name is applied in the case of binary separation (Heinze and Schemper, 2002). The function `pordlogistf` from the package `OrdinalLogisticBioplot` is applied in the case of ordinal separation (Hern'andez, 2013). All instances of separation are reported.

In the case of `Setting=c("binord")` or `Setting=c("ordord")` the appropriate models (either logistic regression or a proportional odds models) are fitted in (2) to accommodate the form (either binary or ordinal) of the true or surrogate response variable. The rest of the analysis would proceed in a similar manner as that described above.

Value

An object of class `FixedDiscrDiscrIT` with components,

`Trial.Spec.Results`
: A `data.frame` that contains the trial-specific intercepts and treatment effects for the surrogate and the true endpoints. Also, the number of observations per trial; whether the trial was able to be included in the analysis for both R^2_h and R^2_{ht}; whether separation occurred and hence the penalized likelihood approach used for the surrogate or true outcome.

`R2ht`
: A `data.frame` that contains the trial-level surrogacy estimate and its confidence interval.

`R2h`
: A `data.frame` that contains the individual-level surrogacy estimate and its confidence interval.

Author(s)

Hannah M. Ensor & Christopher J. Weir

References

Alonso, A, & Molenberghs, G. (2007). Surrogate marker evaluation from an information theory perspective. *Biometrics, 63,* 180-186.

Alonso, A, & Molenberghs, G., Geys, H., Buyse, M. & Vangeneugden, T. (2006). A unifying approach for surrogate marker validation based on Prentice's criteria. *Statistics in medicine, 25,* 205-221.

Firth, D. (1993). Bias reduction of maximum likelihood estimates. *Biometrika, 80,* 27-38.

Heinze, G. & Schemper, M. 2002. A solution to the problem of separation in logistic regression. *Statistics in medicine, 21,* 2409-2419.

Hern'andez, J. C. V.-V. O., J. L. 2013. OrdinalLogisticBiplot: Biplot representations of ordinal variables. R.

See Also

FixedContContIT, plot Information-Theoretic, logistf

Examples

```
## Not run:  # Time consuming (>5sec) code part
# Example 1
# Conduct an analysis based on a simulated dataset with 2000 patients, 100 trials,
# and Rindiv=Rtrial=.8

# Simulate the data:
Sim.Data.MTS(N.Total=2000, N.Trial=100, R.Trial.Target=.8, R.Indiv.Target=.8,
Seed=123, Model="Full")

# create a binary true and ordinal surrogate outcome
Data.Observed.MTS$True<-findInterval(Data.Observed.MTS$True,
c(quantile(Data.Observed.MTS$True,0.5)))
Data.Observed.MTS$Surr<-findInterval(Data.Observed.MTS$Surr,
c(quantile(Data.Observed.MTS$Surr,0.333),quantile(Data.Observed.MTS$Surr,0.666)))

# Assess surrogacy based on a full fixed-effect model
# in the information-theoretic framework for a binary surrogate and ordinal true outcome:
SurEval <- FixedDiscrDiscrIT(Dataset=Data.Observed.MTS, Surr=Surr, True=True, Treat=Treat,
Trial.ID=Trial.ID, Setting="ordbin")

# Show a summary of the results:
summary(SurEval)
SurEval$Trial.Spec.Results
SurEval$R2h
SurEval$R2ht

## End(Not run)
```

ICA.BinBin *Assess surrogacy in the causal-inference single-trial setting in the binary-binary case*

Description

The function ICA.BinBin quantifies surrogacy in the single-trial causal-inference framework (individual causal association and causal concordance) when both the surrogate and the true endpoints are binary outcomes. See **Details** below.

Usage

```
ICA.BinBin(pi1_1_, pi1_0_, pi_1_1, pi_1_0, pi0_1_, pi_0_1,
Monotonicity=c("General"), Sum_Pi_f = seq(from=0.01, to=0.99, by=.01),
M=10000, Volume.Perc=0, Seed=sample(1:100000, size=1))
```

Arguments

pi1_1_	A scalar or vector that contains values for $P(T = 1, S = 1\|Z = 0)$, i.e., the probability that $S = T = 1$ when under treatment $Z = 0$. A vector is specified to account for uncertainty, i.e., rather than keeping $P(T = 1, S = 1\|Z = 0)$ fixed at one estimated value, a distribution can be specified (see **examples** below) from which a value is drawn in each run.
pi1_0_	A scalar or vector that contains values for $P(T = 1, S = 0\|Z = 0)$.
pi_1_1	A scalar or vector that contains values for $P(T = 1, S = 1\|Z = 1)$.
pi_1_0	A scalar or vector that contains values for $P(T = 1, S = 0\|Z = 1)$.
pi0_1_	A scalar or vector that contains values for $P(T = 0, S = 1\|Z = 0)$.
pi_0_1	A scalar or vector that contains values for $P(T = 0, S = 1\|Z = 1)$.
Monotonicity	Specifies which assumptions regarding monotonicity should be made: Monotonicity=c("General"), Monotonicity=c("No"), Monotonicity=c("True.Endp"), Monotonicity=c("Surr.Endp"), or Monotonicity=c("Surr.True.Endp"). See **Details** below. Default Monotonicity=c("General").
Sum_Pi_f	A scalar or vector that specifies the grid of values $G = g_1,\ g_2,\ ...,\ g_k$ to be considered when the sensitivity analysis is conducted. See **Details** below. Default Sum_Pi_f = seq(from=0.01, to=0.99, by=.01).
M	The number of runs that are conducted for a given value of Sum_Pi_f. This argument is not used when Volume.Perc=0. Default M=10000.
Volume.Perc	Note that the marginals that are observable in the data set a number of restrictions on the unidentified correlations. For example, under montonicity for S and T, it holds that $\pi_{0111} <= min(\pi_{0\cdot1\cdot}, \pi_{\cdot1\cdot1})$ and $\pi_{1100} <= min(\pi_{1\cdot0\cdot}, \pi_{\cdot1\cdot0})$. For example, when $min(\pi_{0\cdot1\cdot}, \pi_{\cdot1\cdot1}) = 0.10$ and $min(\pi_{1\cdot0\cdot}, \pi_{\cdot1\cdot0}) = 0.08$, then all valid $\pi_{0111} <= 0.10$ and all valid $\pi_{1100} <= 0.08$. The argument Volume.Perc specifies the fraction of the 'volume' of the paramater space that is explored. This volume is computed based on the grids G=0, 0.01, ..., maximum possible value for the counterfactual probability at hand. E.g., in the

previous example, the 'volume' of the parameter space would be $11 * 9 = 99$, and when e.g., the argument `Volume.Perc=1` is used a total of 99 runs will be conducted for each given value of `Sum_Pi_f`. Notice that when monotonicity is not assumed, relatively high values of `Volume.Perc` will lead to a large number of runs and consequently a long analysis time.

`Seed` The seed to be used to generate π_r. Default `Seed=sample(1:100000, size=1)`.

Details

In the continuous normal setting, surroagacy can be assessed by studying the association between the individual causal effects on S and T (see `ICA.ContCont`). In that setting, the Pearson correlation is the obvious measure of association.

When S and T are binary endpoints, multiple alternatives exist. Alonso et al. (2014) proposed the individual causal association (ICA; R^2_H), which captures the association between the individual causal effects of the treatment on S (Δ_S) and T (Δ_T) using information-theoretic principles.

The function `ICA.BinBin` computes R^2_H based on plausible values of the potential outcomes. Denote by $\boldsymbol{Y}' = (T_0, T_1, S_0, S_1)$ the vector of potential outcomes. The vector $\boldsymbol{Y}$ can take 16 values and the set of parameters $\pi_{ijpq} = P(T_0 = i, T_1 = j, S_0 = p, S_1 = q)$ (with $i, j, p, q = 0/1$) fully characterizes its distribution.

However, the parameters in π_{ijpq} are not all functionally independent, e.g., $1 = \pi_{\cdot\cdot\cdot\cdot}$. When no assumptions regarding monotonicity are made, the data impose a total of 7 restrictions, and thus only 9 proabilities in π_{ijpq} are allowed to vary freely (for details, see Alonso et al., 2014). Based on the data and assuming SUTVA, the marginal probabilites $\pi_{1\cdot1\cdot}$, $\pi_{1\cdot0\cdot}$, $\pi_{\cdot1\cdot1}$, $\pi_{\cdot1\cdot0}$, $\pi_{0\cdot1\cdot}$, and $\pi_{\cdot0\cdot1}$ can be computed (by hand or using the function `MarginalProbs`). Define the vector

$$\boldsymbol{b}' = (1, \pi_{1\cdot1\cdot}, \pi_{1\cdot0\cdot}, \pi_{\cdot1\cdot1}, \pi_{\cdot1\cdot0}, \pi_{0\cdot1\cdot}, \pi_{\cdot0\cdot1})$$

and $\boldsymbol{A}$ is a contrast matrix such that the identified restrictions can be written as a system of linear equation

$$\boldsymbol{A\pi} = \boldsymbol{b}.$$

The matrix $\boldsymbol{A}$ has rank 7 and can be partitioned as $\boldsymbol{A} = (\boldsymbol{A_r}|\boldsymbol{A_f})$, and similarly the vector $\boldsymbol{\pi}$ can be partitioned as $\boldsymbol{\pi}' = (\boldsymbol{\pi_r}'|\boldsymbol{\pi_f}')$ (where f refers to the submatrix/vector given by the 9 last columns/components of $\boldsymbol{A}/\boldsymbol{\pi}$). Using these partitions the previous system of linear equations can be rewritten as

$$\boldsymbol{A_r\pi_r} + \boldsymbol{A_f\pi_f} = \boldsymbol{b}.$$

The following algorithm is used to generate plausible distributions for $\boldsymbol{Y}$. First, select a value of the specified grid of values (specified using `Sum_Pi_f` in the function call). For $k = 1$ to M (specified using `M` in the function call), generate a vector π_f that contains 9 components that are uniformly sampled from hyperplane subject to the restriction that the sum of the generated components equals `Sum_Pi_f` (the function `RandVec`, which uses the `randfixedsum` algorithm written by Roger Stafford, is used to obtain these components). Next, $\boldsymbol{\pi_r} = \boldsymbol{A_r^{-1}(b - A_f\pi_f)}$ is computed and the π_r vectors where all components are in the $[0;\ 1]$ range are retained. This procedure is repeated for each of the `Sum_Pi_f` values. Based on these results, R^2_H is estimated. The obtained values can be used to conduct a sensitivity analysis during the validation exercise.

The previous developments hold when no monotonicity is assumed. When monotonicity for S, T, or for S and T is assumed, some of the probabilities of π are zero. For example, when montonicity is

assumed for T, then $P(T_0 <= T_1) = 1$, or equivantly, $\pi_{1000} = \pi_{1010} = \pi_{1001} = \pi_{1011} = 0$. When monotonicity is assumed, the procedure described above is modified accordingly (for details, see Alonso et al., 2014). When a general analysis is requested (using `Monotonicity=c("General")` in the function call), all settings are considered (no monotonicity, monotonicity for S alone, for T alone, and for both for S and T.)

To account for the uncertainty in the estimation of the marginal probabilities, a vector of values can be specified from which a random draw is made in each run (see **Examples** below).

Value

An object of class `ICA.BinBin` with components,

`Pi.Vectors`	An object of class `data.frame` that contains the valid π vectors.
`R2_H`	The vector of the R^2_H values.
`Theta_T`	The vector of odds ratios for T.
`Theta_S`	The vector of odds ratios for S.
`H_Delta_T`	The vector of the entropies of Δ_T.
`Monotonicity`	The assumption regarding monotonicity that was made.
`Volume.No`	The 'volume' of the parameter space when monotonicity is not assumed. Is only provided when the argument $Volume.Perc$ is used (i.e., when it is not equal to 0.
`Volume.T`	The 'volume' of the parameter space when monotonicity for T is assumed. Is only provided when the argument $Volume.Perc$ is used.
`Volume.S`	The 'volume' of the parameter space when monotonicity for S is assumed. Is only provided when the argument $Volume.Perc$ is used.
`Volume.ST`	The 'volume' of the parameter space when monotonicity for S and T is assumed. Is only provided when the argument $Volume.Perc$ is used.

Author(s)

Wim Van der Elst, Paul Meyvisch, Ariel Alonso & Geert Molenberghs

References

Alonso, A., Van der Elst, W., & Molenberghs, G. (2015). Validation of surrogate endpoints: the binary-binary setting from a causal inference perspective.

See Also

`ICA.ContCont`, `MICA.ContCont`

Examples

```
## Not run: # Time consuming code part
# Compute R2_H given the marginals specified as the pi's, making no
# assumptions regarding monotonicity (general case)
ICA <- ICA.BinBin(pi1_1_=0.2619048, pi1_0_=0.2857143, pi_1_1=0.6372549,
```

```
pi_1_0=0.07843137, pi0_1_=0.1349206, pi_0_1=0.127451, Seed=1,
Monotonicity=c("General"), Sum_Pi_f = seq(from=0.01, to=.99, by=.01), M=10000)

# obtain plot of the results
plot(ICA, R2_H=TRUE)

# Example 2 where the uncertainty in the estimation
# of the marginals is taken into account
ICA_BINBIN2 <- ICA.BinBin(pi1_1_=runif(10000, 0.2573, 0.4252),
pi1_0_=runif(10000, 0.1769, 0.3310),
pi_1_1=runif(10000, 0.5947, 0.7779),
pi_1_0=runif(10000, 0.0322, 0.1442),
pi0_1_=runif(10000, 0.0617, 0.1764),
pi_0_1=runif(10000, 0.0254, 0.1315),
Monotonicity=c("General"),
Sum_Pi_f = seq(from=0.01, to=0.99, by=.01),
M=50000, Seed=1)

# Plot results
plot(ICA_BINBIN2)

## End(Not run)
```

`ICA.BinBin.CounterAssum` *ICA (binary-binary setting) that is obtaied when the counterfactual correlations are assumed to fall within some prespecified ranges.*

Description

Shows the results of ICA (binary-binary setting) in the subgroup of results where the counterfactual correlations are assumed to fall within some prespecified ranges.

Usage

```
ICA.BinBin.CounterAssum(x, r2_h_S0S1_min, r2_h_S0S1_max, r2_h_S0T1_min,
r2_h_S0T1_max, r2_h_T0T1_min, r2_h_T0T1_max, r2_h_T0S1_min, r2_h_T0S1_max,
Monotonicity="General", Type="Freq", MainPlot=" ", Cex.Legend=1,
Cex.Position="topright", ...)
```

Arguments

`x`	An object of class `ICA.BinBin`. See `ICA.BinBin`.
`r2_h_S0S1_min`	The minimum value to be considered for the counterfactual correlation $r^2_h(S_0, S_1)$.
`r2_h_S0S1_max`	The maximum value to be considered for the counterfactual correlation $r^2_h(S_0, S_1)$.
`r2_h_S0T1_min`	The minimum value to be considered for the counterfactual correlation $r^2_h(S_0, T_1)$.
`r2_h_S0T1_max`	The maximum value to be considered for the counterfactual correlation $r^2_h(S_0, T_1)$.

r2_h_T0T1_min	The minimum value to be considered for the counterfactual correlation $r_h^2(T_0, T_1)$.
r2_h_T0T1_max	The maximum value to be considered for the counterfactual correlation $r_h^2(T_0, T_1)$.
r2_h_T0S1_min	The minimum value to be considered for the counterfactual correlation $r_h^2(T_0, S_1)$.
r2_h_T0S1_max	The maximum value to be considered for the counterfactual correlation $r_h^2(T_0, S_1)$.
Monotonicity	Specifies whether the all results in the fitted object ICA.BinBin should be shown (i.e., Monotonicity=c("General")), or a subset of the results arising under specific assumptions (i.e., Monotonicity=c("No"), Monotonicity=c("True.Endp"), Monotonicity=c("Surr.Endp"), or Monotonicity=c("Surr.True.Endp")). Default Monotonicity=c("General").
Type	The type of plot that is produced. When Type="Freq" or Type="Density", the Y-axis shows frequencies or densities of R_H^2. When Type="All.Densities" and the fitted object of class ICA.BinBin was obtained using a general analysis (i.e., conducting the analyses assuming no monotonicity, monotonicity for S alone, monotonicity for T alone, and for both S and T, so using Monotonicity=c("General") in the function call of ICA.BinBin), the density plots are shown for the four scenarios where different assumptions regarding monotonicity are made. Default "Freq".
MainPlot	The title of the plot. Default " ".
Cex.Legend	The size of the legend when Type="All.Densities" is used. Default Cex.Legend=1.
Cex.Position	The position of the legend, Cex.Position="topright" or Cex.Position="topleft". Default Cex.Position="topright".
...	Other arguments to be passed to the plot() function.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., Molenberghs, G., Buyse, M., & Burzykowski, T. (submitted). On the relationship between the causal inference and meta-analytic paradigms for the validation of surrogate markers.

Van der Elst, W., Alonso, A., & Molenberghs, G. (submitted). An exploration of the relationship between causal inference and meta-analytic measures of surrogacy.

See Also

ICA.BinBin

Examples

```
## Not run:  #Time consuming (>5 sec) code part
# Compute R2_H given the marginals specified as the pi's, making no
# assumptions regarding monotonicity (general case)
ICA <- ICA.BinBin.Grid.Sample(pi1_1_=0.261, pi1_0_=0.285,
pi_1_1=0.637, pi_1_0=0.078, pi0_1_=0.134, pi_0_1=0.127,
Monotonicity=c("General"), M=5000, Seed=1)
```

```
# Obtain a density plot of R2_H, assuming that
# r2_h_S0S1>=.2, r2_h_S0T1>=0, r2_h_T0T1>=.2, and r2_h_T0S1>=0
ICA.BinBin.CounterAssum(ICA, r2_h_S0S1_min=.2, r2_h_S0S1_max=1,
r2_h_S0T1_min=0, r2_h_S0T1_max=1, r2_h_T0T1_min=0.2, r2_h_T0T1_max=1,
r2_h_T0S1_min=0, r2_h_T0S1_max=1, Monotonicity="General",
Type="Density")

# Now show the densities of R2_H under the different
# monotonicity assumptions
ICA.BinBin.CounterAssum(ICA, r2_h_S0S1_min=.2, r2_h_S0S1_max=1,
r2_h_S0T1_min=0, r2_h_S0T1_max=1, r2_h_T0T1_min=0.2, r2_h_T0T1_max=1,
r2_h_T0S1_min=0, r2_h_T0S1_max=1, Monotonicity="General",
Type="All.Densities", MainPlot=" ", Cex.Legend=1,
Cex.Position="topright", ylim=c(0, 20))

## End(Not run)
```

`ICA.BinBin.Grid.Full` *Assess surrogacy in the causal-inference single-trial setting in the binary-binary case when monotonicity for S and T is assumed using the full grid-based approach*

Description

The function `ICA.BinBin.Grid.Full` quantifies surrogacy in the single-trial causal-inference framework (individual causal association and causal concordance) when both the surrogate and the true endpoints are binary outcomes. This method provides an alternative for `ICA.BinBin` and `ICA.BinBin.Grid.Sample`. It uses an alternative strategy to identify plausible values for π. See **Details** below.

Usage

```
ICA.BinBin.Grid.Full(pi1_1_, pi1_0_, pi_1_1, pi_1_0, pi0_1_, pi_0_1,
Monotonicity=c("General"), pi_1001=seq(0, 1, by=.02),
pi_1110=seq(0, 1, by=.02), pi_1101=seq(0, 1, by=.02),
pi_1011=seq(0, 1, by=.02), pi_1111=seq(0, 1, by=.02),
pi_0110=seq(0, 1, by=.02), pi_0011=seq(0, 1, by=.02),
pi_0111=seq(0, 1, by=.02), pi_1100=seq(0, 1, by=.02),
Seed=sample(1:100000, size=1))
```

Arguments

`pi1_1_`	A scalar that contains $P(T = 1, S = 1 \mid Z = 0)$, i.e., the proability that $S = T = 1$ when under treatment $Z = 0$.
`pi1_0_`	A scalar that contains $P(T = 1, S = 0 \mid Z = 0)$.
`pi_1_1`	A scalar that contains $P(T = 1, S = 1 \mid Z = 1)$.
`pi_1_0`	A scalar that contains $P(T = 1, S = 0 \mid Z = 1)$.

pi0_1_	A scalar that contains $P(T = 0, S = 1|Z = 0)$.
pi_0_1	A scalar that contains $P(T = 0, S = 1|Z = 1)$.
Monotonicity	Specifies which assumptions regarding monotonicity should be made: Monotonicity=c("General"), Monotonicity=c("No"), Monotonicity=c("True.Endp"), Monotonicity=c("Surr.Endp"), or Monotonicity=c("Surr.True.Endp"). When a general analysis is requested (using Monotonicity=c("General") in the function call), all settings are considered (no monotonicity, monotonicity for S alone, for T alone, and for both for S and T. Default Monotonicity=c("General").
pi_1001	A vector that specifies the grid of values that should be considered for π_{pi_1001}. Default pi_1001=seq(0, 1, by=.02).
pi_1110	A vector that specifies the grid of values that should be considered for π_{pi_1110}. Default pi_1110=seq(0, 1, by=.02).
pi_1101	A vector that specifies the grid of values that should be considered for π_{pi_1101}. Default pi_1101=seq(0, 1, by=.02).
pi_1011	A vector that specifies the grid of values that should be considered for π_{pi_1011}. Default pi_1011=seq(0, 1, by=.02).
pi_1111	A vector that specifies the grid of values that should be considered for π_{pi_1111}. Default pi_1111=seq(0, 1, by=.02).
pi_0110	A vector that specifies the grid of values that should be considered for π_{pi_0110}. Default pi_0110=seq(0, 1, by=.02).
pi_0011	A vector that specifies the grid of values that should be considered for π_{pi_0011}. Default pi_0011=seq(0, 1, by=.02).
pi_0111	A vector that specifies the grid of values that should be considered for π_{pi_0111}. Default pi_0111=seq(0, 1, by=.02).
pi_1100	A vector that specifies the grid of values that should be considered for π_{pi_1100}. Default pi_1100=seq(0, 1, by=.02).
Seed	The seed to be used to generate π_r. Default Seed=sample(1:100000, size=1).

Details

In the continuous normal setting, surroagacy can be assessed by studying the association between the individual causal effects on S and T (see ICA.ContCont). In that setting, the Pearson correlation is the obvious measure of association.

When S and T are binary endpoints, multiple alternatives exist. Alonso et al. (2014) proposed the individual causal association (ICA; R^2_H), which captures the association between the individual causal effects of the treatment on S (Δ_S) and T (Δ_T) using information-theoretic principles.

The function ICA.BinBin.Grid.Full computes R^2_H using a grid-based approach where all possible combinations of the specified grids for the parameters that are allowed that are allowed to vary freely are considered. When it is not assumed that monotonicity holds for both S and T, the computationally less demanding algorithm ICA.BinBin.Grid.Sample may be preferred.

Value

An object of class ICA.BinBin with components,

Pi.Vectors	An object of class data.frame that contains the valid π vectors.
R2_H	The vector of the R_H^2 values.
Theta_T	The vector of odds ratios for T.
Theta_S	The vector of odds ratios for S.
H_Delta_T	The vector of the entropies of Δ_T.

Author(s)

Wim Van der Elst, Paul Meyvisch, Ariel Alonso & Geert Molenberghs

References

Alonso, A., Van der Elst, W., & Molenberghs, G. (2014). Validation of surrogate endpoints: the binary-binary setting from a causal inference perspective.

Buyse, M., Burzykowski, T., Aloso, A., & Molenberghs, G. (2014). Direct estimation of joint counterfactual probabilities, with application to surrogate marker validation.

See Also

ICA.ContCont, MICA.ContCont, ICA.BinBin, ICA.BinBin.Grid.Sample

Examples

```
## Not run:  # time consuming code part
# Compute R2_H given the marginals,
# assuming monotonicity for S and T and grids
# pi_0111=seq(0, 1, by=.001) and
# pi_1100=seq(0, 1, by=.001)
ICA <- ICA.BinBin.Grid.Full(pi1_1_=0.2619048, pi1_0_=0.2857143, pi_1_1=0.6372549,
pi_1_0=0.07843137, pi0_1_=0.1349206, pi_0_1=0.127451,
pi_0111=seq(0, 1, by=.01), pi_1100=seq(0, 1, by=.01), Seed=1)

# obtain plot of R2_H
plot(ICA, R2_H=TRUE)

## End(Not run)
```

ICA.BinBin.Grid.Sample

Assess surrogacy in the causal-inference single-trial setting in the binary-binary case when monotonicity for S and T is assumed using the grid-based sample approach

---

Description

The function `ICA.BinBin.Grid.Sample` quantifies surrogacy in the single-trial causal-inference framework (individual causal association and causal concordance) when both the surrogate and the true endpoints are binary outcomes. This method provides an alternative for `ICA.BinBin` and `ICA.BinBin.Grid.Full`. It uses an alternative strategy to identify plausible values for π. See **Details** below.

Usage

```
ICA.BinBin.Grid.Sample(pi1_1_, pi1_0_, pi_1_1, pi_1_0, pi0_1_,
pi_0_1, Monotonicity=c("General"), M=100000,
Volume.Perc=0, Seed=sample(1:100000, size=1))
```

Arguments

`pi1_1_`	A scalar or vector that contains values for $P(T = 1, S = 1\|Z = 0)$, i.e., the probability that $S = T = 1$ when under treatment $Z = 0$. A vector is specified to account for uncertainty, i.e., rather than keeping $P(T = 1, S = 1\|Z = 0)$ fixed at one estimated value, a distribution can be specified (see **examples** below) from which a value is drawn in each run.
`pi1_0_`	A scalar or vector that contains values for $P(T = 1, S = 0\|Z = 0)$.
`pi_1_1`	A scalar or vector that contains values for $P(T = 1, S = 1\|Z = 1)$.
`pi_1_0`	A scalar or vector that contains values for $P(T = 1, S = 0\|Z = 1)$.
`pi0_1_`	A scalar or vector that contains values for $P(T = 0, S = 1\|Z = 0)$.
`pi_0_1`	A scalar or vector that contains values for $P(T = 0, S = 1\|Z = 1)$.
`Monotonicity`	Specifies which assumptions regarding monotonicity should be made: `Monotonicity=c("General")`, `Monotonicity=c("No")`, `Monotonicity=c("True.Endp")`, `Monotonicity=c("Surr.Endp")`, or `Monotonicity=c("Surr.True.Endp")`. When a general analysis is requested (using `Monotonicity=c("General")` in the function call), all settings are considered (no monotonicity, monotonicity for S alone, for T alone, and for both for S and T. Default `Monotonicity=c("General")`.
`M`	The number of random samples that have to be drawn for the freely varying parameters. Default `M=100000`. This argument is not used when `Volume.Perc=0`. Default `M=10000`.
`Volume.Perc`	Note that the marginals that are observable in the data set a number of restrictions on the unidentified correlations. For example, under montonicity for S and T, it holds that $\pi_{0111} <= min(\pi_{0\cdot1\cdot}, \pi_{\cdot1\cdot1})$ and $\pi_{1100} <= min(\pi_{1\cdot0\cdot}, \pi_{\cdot1\cdot0})$. For example, when $min(\pi_{0\cdot1\cdot}, \pi_{\cdot1\cdot1}) = 0.10$ and $min(\pi_{1\cdot0\cdot}, \pi_{\cdot1\cdot0}) = 0.08$, then all valid $\pi_{0111} <= 0.10$ and all valid $\pi_{1100} <= 0.08$. The argument `Volume.Perc` specifies the fraction of the 'volume' of the paramater space that is explored. This volume is computed based on the grids G=0, 0.01, ..., maximum possible value for the counterfactual probability at hand. E.g., in the previous example, the 'volume' of the parameter space would be $11 * 9 = 99$, and when e.g., the argument `Volume.Perc=1` is used a total of 99 runs will be conducted. Notice that when monotonicity is not assumed, relatively high values of `Volume.Perc` will lead to a large number of runs and consequently a long analysis time.

`Seed` The seed to be used to generate π_r. Default `M=100000`.

Details

In the continuous normal setting, surroagacy can be assessed by studying the association between the individual causal effects on S and T (see `ICA.ContCont`). In that setting, the Pearson correlation is the obvious measure of association.

When S and T are binary endpoints, multiple alternatives exist. Alonso et al. (2014) proposed the individual causal association (ICA; R^2_H), which captures the association between the individual causal effects of the treatment on S (Δ_S) and T (Δ_T) using information-theoretic principles.

The function `ICA.BinBin.Grid.Full` computes R^2_H using a grid-based approach where all possible combinations of the specified grids for the parameters that are allowed that are allowed to vary freely are considered. When it is not assumed that monotonicity holds for both S and T, the number of possible combinations become very high. The function `ICA.BinBin.Grid.Sample` considers a random sample of all possible combinations.

Value

An object of class `ICA.BinBin` with components,

`Pi.Vectors`	An object of class `data.frame` that contains the valid π vectors.
`R2_H`	The vector of the R^2_H values.
`Theta_T`	The vector of odds ratios for T.
`Theta_S`	The vector of odds ratios for S.
`H_Delta_T`	The vector of the entropies of Δ_T.
`Volume.No`	The 'volume' of the parameter space when monotonicity is not assumed.
`Volume.T`	The 'volume' of the parameter space when monotonicity for T is assumed.
`Volume.S`	The 'volume' of the parameter space when monotonicity for S is assumed.
`Volume.ST`	The 'volume' of the parameter space when monotonicity for S and T is assumed.

Author(s)

Wim Van der Elst, Paul Meyvisch, Ariel Alonso & Geert Molenberghs

References

Alonso, A., Van der Elst, W., & Molenberghs, G. (2014). Validation of surrogate endpoints: the binary-binary setting from a causal inference perspective.

Buyse, M., Burzykowski, T., Aloso, A., & Molenberghs, G. (2014). Direct estimation of joint counterfactual probabilities, with application to surrogate marker validation.

See Also

`ICA.ContCont`, `MICA.ContCont`, `ICA.BinBin`, `ICA.BinBin.Grid.Sample`

Examples

```
# Compute R2_H given the marginals,
# assuming monotonicity for S and T and grids
# pi_0111=seq(0, 1, by=.001) and
# pi_1100=seq(0, 1, by=.001)
ICA <- ICA.BinBin.Grid.Sample(pi1_1_=0.261, pi1_0_=0.285,
pi_1_1=0.637, pi_1_0=0.078, pi0_1_=0.134, pi_0_1=0.127,
Monotonicity=c("General"), M=2500, Seed=1)

# obtain plot of R2_H
plot(ICA, R2_H=TRUE)
```

`ICA.ContCont` *Assess surrogacy in the causal-inference single-trial setting (Individual Causal Association, ICA) in the Continuous-continuous case*

Description

The function `ICA.ContCont` quantifies surrogacy in the single-trial causal-inference framework. See **Details** below.

Usage

```
ICA.ContCont(T0S0, T1S1, T0T0=1, T1T1=1, S0S0=1, S1S1=1, T0T1=seq(-1, 1, by=.1),
T0S1=seq(-1, 1, by=.1), T1S0=seq(-1, 1, by=.1), S0S1=seq(-1, 1, by=.1))
```

Arguments

`T0S0`	A scalar or vector that specifies the correlation(s) between the surrogate and the true endpoint in the control treatment condition that should be considered in the computation of ρ_Δ.
`T1S1`	A scalar or vector that specifies the correlation(s) between the surrogate and the true endpoint in the experimental treatment condition that should be considered in the computation of ρ_Δ.
`T0T0`	A scalar that specifies the variance of the true endpoint in the control treatment condition that should be considered in the computation of ρ_Δ. Default 1.
`T1T1`	A scalar that specifies the variance of the true endpoint in the experimental treatment condition that should be considered in the computation of ρ_Δ. Default 1.
`S0S0`	A scalar that specifies the variance of the surrogate endpoint in the control treatment condition that should be considered in the computation of ρ_Δ. Default 1.
`S1S1`	A scalar that specifies the variance of the surrogate endpoint in the experimental treatment condition that should be considered in the computation of ρ_Δ. Default 1.

`T0T1` A scalar or vector that contains the correlation(s) between the counterfactuals T0 and T1 that should be considered in the computation of ρ_Δ. Default `seq(-1, 1, by=.1)`, i.e., the values $-1, -0.9, -0.8, \ldots, 1$.

`T0S1` A scalar or vector that contains the correlation(s) between the counterfactuals T0 and S1 that should be considered in the computation of ρ_Δ. Default `seq(-1, 1, by=.1)`.

`T1S0` A scalar or vector that contains the correlation(s) between the counterfactuals T1 and S0 that should be considered in the computation of ρ_Δ. Default `seq(-1, 1, by=.1)`.

`S0S1` A scalar or vector that contains the correlation(s) between the counterfactuals S0 and S1 that should be considered in the computation of ρ_Δ. Default `seq(-1, 1, by=.1)`.

Details

Based on the causal-inference framework, it is assumed that each subject *j* has four counterfactuals (or potential outcomes), i.e., T_{0j}, T_{1j}, S_{0j}, and S_{1j}. Let T_{0j} and T_{1j} denote the counterfactuals for the true endpoint (T) under the control ($Z = 0$) and the experimental ($Z = 1$) treatments of subject *j*, respectively. Similarly, S_{0j} and S_{1j} denote the corresponding counterfactuals for the surrogate endpoint (S) under the control and experimental treatments, respectively. The individual causal effects of Z on T and S for a given subject *j* are then defined as $\Delta_{T_j} = T_{1j} - T_{0j}$ and $\Delta_{S_j} = S_{1j} - S_{0j}$, respectively.

In the single-trial causal-inference framework, surrogacy can be quantified as the correlation between the individual causal effects of Z on S and T (for details, see Alonso et al., submitted):

$$\rho_\Delta = \rho(\Delta_{T_j}, \Delta_{S_j}) = \frac{\sqrt{\sigma_{S_0S_0}\sigma_{T_0T_0}}\rho_{S_0T_0} + \sqrt{\sigma_{S_1S_1}\sigma_{T_1T_1}}\rho_{S_1T_1} - \sqrt{\sigma_{S_0S_0}\sigma_{T_1T_1}}\rho_{S_0T_1} - \sqrt{\sigma_{S_1S_1}\sigma_{T_0T_0}}\rho_{S_1T_0}}{\sqrt{(\sigma_{T_0T_0} + \sigma_{T_1T_1} - 2\sqrt{\sigma_{T_0T_0}\sigma_{T_1T_1}}\rho_{T_0T_1})(\sigma_{S_0S_0} + \sigma_{S_1S_1} - 2\sqrt{\sigma_{S_0S_0}\sigma_{S_1S_1}}\rho_{S_0S_1})}},$$

where the correlations $\rho_{S_0T_1}$, $\rho_{S_1T_0}$, $\rho_{T_0T_1}$, and $\rho_{S_0S_1}$ are not estimable. It is thus warranted to conduct a sensitivity analysis (by considering vectors of possible values for the correlations between the counterfactuals – rather than point estimates).

When the user specifies a vector of values that should be considered for one or more of the counterfactual correlations in the above expression, the function `ICA.ContCont` constructs all possible matrices that can be formed as based on these values, identifies the matrices that are positive definite (i.e., valid correlation matrices), and computes ρ_Δ for each of these matrices. The obtained vector of ρ_Δ values can subsequently be used to examine (i) the impact of different assumptions regarding the correlations between the counterfactuals on the results (see also `plot Causal-Inference ContCont`), and (ii) the extent to which proponents of the causal-inference and meta-analytic frameworks will reach the same conclusion with respect to the appropriateness of the candidate surrogate at hand.

The function `ICA.ContCont` also generates output that is useful to examine the plausibility of finding a good surrogate endpoint (see `GoodSurr` in the **Value** section below). For details, see Alonso et al. (submitted).

Notes

A single ρ_Δ value is obtained when all correlations in the function call are scalars.

Value

An object of class `ICA.ContCont` with components,

`Total.Num.Matrices`
: An object of class `numeric` that contains the total number of matrices that can be formed as based on the user-specified correlations in the function call.

`Pos.Def`
: A `data.frame` that contains the positive definite matrices that can be formed based on the user-specified correlations. These matrices are used to compute the vector of the ρ_Δ values.

`ICA`
: A scalar or vector that contains the individual causal association (ICA; ρ_Δ) value(s).

`GoodSurr`
: A `data.frame` that contains the ICA (ρ_Δ), σ_{Δ_T}, and δ.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., Molenberghs, G., Buyse, M., & Burzykowski, T. (submitted). On the relationship between the causal-inference and meta-analytic paradigms for the validation of surrogate markers.

See Also

`MICA.ContCont`, `ICA.Sample.ContCont`, `Single.Trial.RE.AA`, `plot Causal-Inference ContCont`

Examples

```
# Generate the vector of ICA.ContCont values when rho_T0S0=rho_T1S1=.95,
# sigma_T0T0=90, sigma_T1T1=100,sigma_ S0S0=10, sigma_S1S1=15, and
# the grid of values {0, .2, ..., 1} is considered for the correlations
# between the counterfactuals:
SurICA <- ICA.ContCont(T0S0=.95, T1S1=.95, T0T0=90, T1T1=100, S0S0=10, S1S1=15,
T0T1=seq(0, 1, by=.2), T0S1=seq(0, 1, by=.2), T1S0=seq(0, 1, by=.2),
S0S1=seq(0, 1, by=.2))

# Examine and plot the vector of generated ICA values:
summary(SurICA)
plot(SurICA)

# Obtain the positive definite matrices than can be formed as based on the
# specified (vectors) of the correlations (these matrices are used to
# compute the ICA values)
SurICA$Pos.Def

# Same, but specify vectors for rho_T0S0 and rho_T1S1: Sample from
# normal with mean .95 and SD=.05 (to account for uncertainty
# in estimation)
SurICA2 <- ICA.ContCont(T0S0=rnorm(n=10000000, mean=.95, sd=.05),
```

```
T1S1=rnorm(n=10000000, mean=.95, sd=.05),
T0T0=90, T1T1=100, S0S0=10, S1S1=15,
T0T1=seq(0, 1, by=.2), T0S1=seq(0, 1, by=.2), T1S0=seq(0, 1, by=.2),
S0S1=seq(0, 1, by=.2))

# Examine results
summary(SurICA2)
plot(SurICA2)
```

`ICA.Sample.ContCont` *Assess surrogacy in the causal-inference single-trial setting (Individual Causal Association, ICA) in the Continuous-continuous case using the grid-based sample approach*

Description

The function `ICA.Sample.ContCont` quantifies surrogacy in the single-trial causal-inference framework. It provides a faster alternative for `ICA.ContCont`. See **Details** below.

Usage

```
ICA.Sample.ContCont(T0S0, T1S1, T0T0=1, T1T1=1, S0S0=1, S1S1=1, T0T1=seq(-1, 1, by=.001),
T0S1=seq(-1, 1, by=.001), T1S0=seq(-1, 1, by=.001), S0S1=seq(-1, 1, by=.001), M=50000)
```

Arguments

`T0S0`	A scalar or vector that specifies the correlation(s) between the surrogate and the true endpoint in the control treatment condition that should be considered in the computation of ρ_Δ.
`T1S1`	A scalar or vector that specifies the correlation(s) between the surrogate and the true endpoint in the experimental treatment condition that should be considered in the computation of ρ_Δ.
`T0T0`	A scalar that specifies the variance of the true endpoint in the control treatment condition that should be considered in the computation of ρ_Δ. Default 1.
`T1T1`	A scalar that specifies the variance of the true endpoint in the experimental treatment condition that should be considered in the computation of ρ_Δ. Default 1.
`S0S0`	A scalar that specifies the variance of the surrogate endpoint in the control treatment condition that should be considered in the computation of ρ_Δ. Default 1.
`S1S1`	A scalar that specifies the variance of the surrogate endpoint in the experimental treatment condition that should be considered in the computation of ρ_Δ. Default 1.
`T0T1`	A scalar or vector that contains the correlation(s) between the counterfactuals T0 and T1 that should be considered in the computation of ρ_Δ. Default `seq(-1, 1, by=.001)`.

`T0S1` A scalar or vector that contains the correlation(s) between the counterfactuals T0 and S1 that should be considered in the computation of ρ_Δ. Default `seq(-1, 1, by=.001)`.

`T1S0` A scalar or vector that contains the correlation(s) between the counterfactuals T1 and S0 that should be considered in the computation of ρ_Δ. Default `seq(-1, 1, by=.001)`.

`S0S1` A scalar or vector that contains the correlation(s) between the counterfactuals S0 and S1 that should be considered in the computation of ρ_Δ. Default `seq(-1, 1, by=.001)`.

`M` The number of runs that should be conducted. Default `50000`.

Details

Based on the causal-inference framework, it is assumed that each subject *j* has four counterfactuals (or potential outcomes), i.e., T_{0j}, T_{1j}, S_{0j}, and S_{1j}. Let T_{0j} and T_{1j} denote the counterfactuals for the true endpoint (T) under the control ($Z = 0$) and the experimental ($Z = 1$) treatments of subject *j*, respectively. Similarly, S_{0j} and S_{1j} denote the corresponding counterfactuals for the surrogate endpoint (S) under the control and experimental treatments, respectively. The individual causal effects of Z on T and S for a given subject *j* are then defined as $\Delta_{T_j} = T_{1j} - T_{0j}$ and $\Delta_{S_j} = S_{1j} - S_{0j}$, respectively.

In the single-trial causal-inference framework, surrogacy can be quantified as the correlation between the individual causal effects of Z on S and T (for details, see Alonso et al., submitted):

$$\rho_\Delta = \rho(\Delta_{T_j}, \Delta_{S_j}) = \frac{\sqrt{\sigma_{S_0S_0}\sigma_{T_0T_0}}\rho_{S_0T_0} + \sqrt{\sigma_{S_1S_1}\sigma_{T_1T_1}}\rho_{S_1T_1} - \sqrt{\sigma_{S_0S_0}\sigma_{T_1T_1}}\rho_{S_0T_1} - \sqrt{\sigma_{S_1S_1}\sigma_{T_0T_0}}\rho_{S_1T_0}}{\sqrt{(\sigma_{T_0T_0} + \sigma_{T_1T_1} - 2\sqrt{\sigma_{T_0T_0}\sigma_{T_1T_1}}\rho_{T_0T_1})(\sigma_{S_0S_0} + \sigma_{S_1S_1} - 2\sqrt{\sigma_{S_0S_0}\sigma_{S_1S_1}}\rho_{S_0S_1})}},$$

where the correlations $\rho_{S_0T_1}$, $\rho_{S_1T_0}$, $\rho_{T_0T_1}$, and $\rho_{S_0S_1}$ are not estimable. It is thus warranted to conduct a sensitivity analysis.

The function `ICA.ContCont` constructs all possible matrices that can be formed based on the specified vectors for $\rho_{S_0T_1}$, $\rho_{S_1T_0}$, $\rho_{T_0T_1}$, and $\rho_{S_0S_1}$, and retains the positive definite ones for the computation of ρ_Δ.

In contrast, the function `ICA.ContCont` samples random values for $\rho_{S_0T_1}$, $\rho_{S_1T_0}$, $\rho_{T_0T_1}$, and $\rho_{S_0S_1}$ based on a uniform distribution with user-specified minimum and maximum values, and retains the positive definite ones for the computation of ρ_Δ.

The obtained vector of ρ_Δ values can subsequently be used to examine (i) the impact of different assumptions regarding the correlations between the counterfactuals on the results (see also `plot Causal-Inference ContCont`), and (ii) the extent to which proponents of the causal-inference and meta-analytic frameworks will reach the same conclusion with respect to the appropriateness of the candidate surrogate at hand.

The function `ICA.Sample.ContCont` also generates output that is useful to examine the plausibility of finding a good surrogate endpoint (see `GoodSurr` in the **Value** section below). For details, see Alonso et al. (submitted).

Notes

A single ρ_Δ value is obtained when all correlations in the function call are scalars.

Value

An object of class `ICA.ContCont` with components,

`Total.Num.Matrices`
: An object of class `numeric` that contains the total number of matrices that can be formed as based on the user-specified correlations in the function call.

`Pos.Def`
: A `data.frame` that contains the positive definite matrices that can be formed based on the user-specified correlations. These matrices are used to compute the vector of the ρ_{Δ} values.

`ICA`
: A scalar or vector that contains the individual causal association (ICA; ρ_{Δ}) value(s).

`GoodSurr`
: A `data.frame` that contains the ICA (ρ_{Δ}), σ_{Δ_T}, and δ.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., Molenberghs, G., Buyse, M., & Burzykowski, T. (submitted). On the relationship between the causal-inference and meta-analytic paradigms for the validation of surrogate markers.

See Also

`MICA.ContCont`, `ICA.ContCont`, `Single.Trial.RE.AA`, `plot Causal-Inference ContCont`

Examples

```
# Generate the vector of ICA values when rho_T0S0=rho_T1S1=.95,
# sigma_T0T0=90, sigma_T1T1=100,sigma_ S0S0=10, sigma_S1S1=15, and
# min=-1 max=1 is considered for the correlations
# between the counterfactuals:
SurICA2 <- ICA.Sample.ContCont(T0S0=.95, T1S1=.95, T0T0=90, T1T1=100, S0S0=10,
S1S1=15, M=5000)

# Examine and plot the vector of generated ICA values:
summary(SurICA2)
plot(SurICA2)
```

`LongToWide` *Reshapes a dataset from the 'long' format (i.e., multiple lines per patient) into the 'wide' format (i.e., one line per patient)*

Description

Reshapes a dataset that is in the 'long' format into the 'wide' format. The dataset should contain a single surrogate endpoint and a single true endpoint value per subject.

Usage

```
LongToWide(Dataset, OutcomeIndicator, IdIndicator, TreatIndicator, OutcomeValue)
```

Arguments

Dataset	A data.frame in the 'long' format that contains (at least) five columns, i.e., one that contains the subject ID, one that contains the trial ID, one that contains the endpoint indicator, one that contains the treatment indicator, and one that contains the endpoint values.
OutcomeIndicator	The name of the variable in Dataset that contains the indicator that distinguishes between the surrogate and true endpoints.
IdIndicator	The name of the variable in Dataset that contains the subject ID.
TreatIndicator	The name of the variable in Dataset that contains the treatment indicator. For the subsequent surrogacy analyses, the treatment indicator should either be coded as 1 for the experimental group and -1 for the control group, or as 1 for the experimental group and 0 for the control group. The $-1/1$ coding is recommended.
OutcomeValue	The name of the variable in Dataset that contains the endpoint values.

Value

A data.frame in the 'wide' format, i.e., a data.frame that contains one line per subject. Each line contains a surrogate value, a true endpoint value, a treatment indicator, a patient ID, and a trial ID.

Author(s)

Wim Van der Elst, Ariel Alonso, and Geert Molenberghs

Examples

```
# Generate a dataset in the 'long' format that contains
# S and T values for 100 patients
Outcome <- rep(x=c(0, 1), times=100)
ID <- rep(seq(1:100), each=2)
Treat <- rep(seq(c(0,1)), each=100)
Outcomes <- as.numeric(matrix(rnorm(1*200, mean=100, sd=10),
                                    ncol=200))
Data <- data.frame(cbind(Outcome, ID, Treat, Outcomes))

# Reshapes the Data object
LongToWide(Dataset=Data, OutcomeIndicator=Outcome, IdIndicator=ID,
           TreatIndicator=Treat, OutcomeValue=Outcomes)
```

`MarginalProbs` *Computes marginal probabilities for a dataset where the surrogate and true endpoints are binary*

Description

This function computes the marginal probabilities associated with the distribution of the potential outcomes for the true and surrogate endpoint.

Usage

```
MarginalProbs(Dataset=Dataset, Surr=Surr, True=True, Treat=Treat)
```

Arguments

`Dataset`	A `data.frame` that should consist of one line per patient. Each line contains (at least) a binary surrogate value, a binary true endpoint value, and a treatment indicator.
`Surr`	The name of the variable in `Dataset` that contains the binary surrogate endpoint values. Should be coded as 0 and 1.
`True`	The name of the variable in `Dataset` that contains the binary true endpoint values. Should be coded as 0 and 1.
`Treat`	The name of the variable in `Dataset` that contains the treatment indicators. The treatment indicator should be coded as 1 for the experimental group and -1 for the control group.

Value

`Theta_T0S0`	The odds ratio for S and T in the control group.
`Theta_T1S1`	The odds ratio for S and T in the experimental group.
`Freq.Cont`	The frequencies for S and T in the control group.
`Freq.Exp`	The frequencies for S and T in the experimental group.
`pi1_1_`	The estimated $\pi_{1\cdot 1\cdot}$
`pi0_1_`	The estimated $\pi_{0\cdot 1\cdot}$
`pi1_0_`	The estimated $\pi_{1\cdot 0\cdot}$
`pi0_0_`	The estimated $\pi_{0\cdot 0\cdot}$
`pi_1_1`	The estimated $\pi_{\cdot 1\cdot 1}$
`pi_1_0`	The estimated $\pi_{\cdot 1\cdot 0}$
`pi_0_1`	The estimated $\pi_{\cdot 0\cdot 1}$
`pi_0_0`	The estimated $\pi_{\cdot 0\cdot 0}$

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

See Also

ICA.BinBin

Examples

```
# Open the ARMD dataset and recode Diff24 and Diff52 as 1
# when the original value is above 0, and 0 otherwise
data(ARMD)
ARMD$Diff24_Dich <- ifelse(ARMD$Diff24>0, 1, 0)
ARMD$Diff52_Dich <- ifelse(ARMD$Diff52>0, 1, 0)

# Obtain marginal probabilities and ORs
MarginalProbs(Dataset=ARMD, Surr=Diff24_Dich, True=Diff52_Dich,
Treat=Treat)
```

MaxEntICABinBin *Use the maximum-entropy approach to compute ICA in the binary-binary setting*

Description

In a surrogate evaluation setting where both S and T are binary endpoints, a sensitivity-based approach where multiple 'plausible values' for ICA are retained can be used (see functions ICA.BinBin, ICA.BinBin.Grid.Full, or ICA.BinBin.Grid.Sample). Alternatively, the maximum entropy distribution of the vector of potential outcomes can be considered, based upon which ICA is subsequently computed. The use of the distribution that maximizes the entropy can be justified based on the fact that any other distribution would necessarily (i) assume information that we do not have, or (ii) contradict information that we do have. The function MaxEntICABinBin implements the latter approach.

Usage

```
MaxEntICABinBin(pi1_1_, pi1_0_, pi_1_1,
pi_1_0, pi0_1_, pi_0_1, Method="BFGS",
Fitted.ICA=NULL)
```

Arguments

pi1_1_	A scalar that contains the estimated value for $P(T = 1, S = 1\|Z = 0)$, i.e., the probability that $S = T = 1$ when under treatment $Z = 0$.
pi1_0_	A scalar that contains the estimated value for $P(T = 1, S = 0\|Z = 0)$.
pi_1_1	A scalar that contains the estimated value for $P(T = 1, S = 1\|Z = 1)$.
pi_1_0	A scalar that contains the estimated value for $P(T = 1, S = 0\|Z = 1)$.
pi0_1_	A scalar that contains the estimated value for $P(T = 0, S = 1\|Z = 0)$.

pi_0_1	A scalar that contains the estimated value for $P(T = 0, S = 1|Z = 1)$.
Method	The maximum entropy frequency vector p^* is calculated based on the optimal solution to an unconstrained dual convex programming problem (for details, see Alonso et al., 2015). Two different optimization methods can be specified, i.e., Method="BFGS" and Method="CG", which implement the quasi-Newton BFGS (Broyden, Fletcher, Goldfarb, and Shanno) and the conjugent gradient (CG) methods (for details on these methods, see the help files of the optim() function and the references theirin). Alternatively, the π vector (obtained when the functions ICA.BinBin, ICA.BinBin.Grid.Full, or ICA.BinBin.Grid.Sample are executed) that is 'closest' to the vector π can be retained. Here, the 'closest' vector is defined as the vector where the sum of the squared differences between the components in the vectors π and π is smallest. The latter 'Minimum Difference' method can re requested by specifying the argument Method="MD" in the function call. Default Method="BFGS".
Fitted.ICA	A fitted object of class ICA.BinBin, ICA.BinBin.Grid.Full, or ICA.BinBin.Grid.Sample. Only required when Method="MD" is used.

Value

R2_H	The R2_H value.
Vector_p	The maximum entropy frequency vector p^*
H_max	The entropy of p^*

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., & Van der Elst, W. (2015). A maximum-entropy approach for the evluation of surrogate endpoints based on causal inference.

See Also

ICA.BinBin, ICA.BinBin.Grid.Sample, ICA.BinBin.Grid.Full, plot MaxEntICA BinBin

Examples

```
# Sensitivity-based ICA results using ICA.BinBin.Grid.Sample
ICA <- ICA.BinBin.Grid.Sample(pi1_1_=0.341, pi0_1_=0.119, pi1_0_=0.254,
pi_1_1=0.686, pi_1_0=0.088, pi_0_1=0.078, Seed=1,
Monotonicity=c("No"), M=5000)

# Maximum-entropy based ICA
MaxEnt <- MaxEntICABinBin(pi1_1_=0.341, pi0_1_=0.119, pi1_0_=0.254,
pi_1_1=0.686, pi_1_0=0.088, pi_0_1=0.078)

# Explore maximum-entropy results
summary(MaxEnt)
```

```
# Plot results
plot(x=MaxEnt, ICA.Fit=ICA)
```

`MaxEntSPFBinBin` *Use the maximum-entropy approach to compute SPF (surrogate predictive function) in the binary-binary setting*

Description

In a surrogate evaluation setting where both S and T are binary endpoints, a sensitivity-based approach where multiple 'plausible values' for vector π (i.e., vectors π that are compatible with the observable data at hand) can be used (for details, see `SPF.BinBin`). Alternatively, the maximum entropy distribution for vector π can be considered (Alonso et al., 2015). The use of the distribution that maximizes the entropy can be justified based on the fact that any other distribution would necessarily (i) assume information that we do not have, or (ii) contradict information that we do have. The function `MaxEntSPFBinBin` implements the latter approach.

Based on vector π, the surrogate predictive function (SPF) is computed, i.e., $r(i,j) = P(\Delta T = i|\Delta S = j)$. For example, $r(-1,1)$ quantifies the probability that the treatment has a negative effect on the true endpoint ($\Delta T = -1$) given that it has a positive effect on the surrogate ($\Delta S = 1$).

Usage

```
MaxEntSPFBinBin(pi1_1_, pi1_0_, pi_1_1,
pi_1_0, pi0_1_, pi_0_1, Method="BFGS",
Fitted.ICA=NULL)
```

Arguments

`pi1_1_`	A scalar that contains the estimated value for $P(T = 1, S = 1|Z = 0)$, i.e., the probability that $S = T = 1$ when under treatment $Z = 0$.
`pi1_0_`	A scalar that contains the estimated value for $P(T = 1, S = 0|Z = 0)$.
`pi_1_1`	A scalar that contains the estimated value for $P(T = 1, S = 1|Z = 1)$.
`pi_1_0`	A scalar that contains the estimated value for $P(T = 1, S = 0|Z = 1)$.
`pi0_1_`	A scalar that contains the estimated value for $P(T = 0, S = 1|Z = 0)$.
`pi_0_1`	A scalar that contains the estimated value for $P(T = 0, S = 1|Z = 1)$.
`Method`	The maximum entropy frequency vector p^* is calculated based on the optimal solution to an unconstrained dual convex programming problem (for details, see Alonso et al., 2015). Two different optimization methods can be specified, i.e., `Method="BFGS"` and `Method="CG"`, which implement the quasi-Newton BFGS (Broyden, Fletcher, Goldfarb, and Shanno) and the conjugent gradient (CG) methods (for details on these methods, see the help files of the `optim()` function and the references theirin). Alternatively, the π vector (obtained when the functions `ICA.BinBin`, `ICA.BinBin.Grid.Full`, or `ICA.BinBin.Grid.Sample` are executed) that is 'closest' to the vector π can be retained. Here, the 'closest'

vector is defined as the vector where the sum of the squared differences between the components in the vectors π and π is smallest. The latter 'Minimum Difference' method can re requested by specifying the argument `Method="MD"` in the function call. Default `Method="BFGS"`.

`Fitted.ICA` A fitted object of class `ICA.BinBin`, `ICA.BinBin.Grid.Full`, or `ICA.BinBin.Grid.Sample`. Only required when `Method="MD"` is used.

Value

`Vector_p`	The maximum entropy frequency vector p^*
`r_1_1`	The vector of values for $r(1,1)$, i.e., $P(\Delta T = 1 \vert \Delta S = 1)$.
`r_min1_1`	The vector of values for $r(-1,1)$.
`r_0_1`	The vector of values for $r(0,1)$.
`r_1_0`	The vector of values for $r(1,0)$.
`r_min1_0`	The vector of values for $r(-1,0)$.
`r_0_0`	The vector of values for $r(0,0)$.
`r_1_min1`	The vector of values for $r(1,-1)$.
`r_min1_min1`	The vector of values for $r(-1,-1)$.
`r_0_min1`	The vector of values for $r(0,-1)$.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., & Van der Elst, W. (2015). A maximum-entropy approach for the evluation of surrogate endpoints based on causal inference.

See Also

`ICA.BinBin`, `ICA.BinBin.Grid.Sample`, `ICA.BinBin.Grid.Full`, `plot MaxEntSPF BinBin`

Examples

```
# Sensitivity-based ICA results using ICA.BinBin.Grid.Sample
ICA <- ICA.BinBin.Grid.Sample(pi1_1_=0.341, pi0_1_=0.119, pi1_0_=0.254,
pi_1_1=0.686, pi_1_0=0.088, pi_0_1=0.078, Seed=1,
Monotonicity=c("No"), M=5000)

# Sensitivity-based SPF
SPFSens <- SPF.BinBin(ICA)

# Maximum-entropy based SPF
SPFMaxEnt <- MaxEntSPFBinBin(pi1_1_=0.341, pi0_1_=0.119, pi1_0_=0.254,
pi_1_1=0.686, pi_1_0=0.088, pi_0_1=0.078)

# Explore maximum-entropy results
```

```
summary(SPFMaxEnt)

# Plot results
plot(x=SPFMaxEnt, SPF.Fit=SPFSens)
```

`MICA.ContCont` *Assess surrogacy in the causal-inference multiple-trial setting (Meta-analytic Individual Causal Association; MICA) in the continuous-continuous case*

Description

The function `MICA.ContCont` quantifies surrogacy in the multiple-trial causal-inference framework. See **Details** below.

Usage

```
MICA.ContCont(Trial.R, D.aa, D.bb, T0S0, T1S1, T0T0=1, T1T1=1, S0S0=1, S1S1=1,
T0T1=seq(-1, 1, by=.1), T0S1=seq(-1, 1, by=.1), T1S0=seq(-1, 1, by=.1),
S0S1=seq(-1, 1, by=.1))
```

Arguments

`Trial.R`	A scalar that specifies the trial-level correlation coefficient (i.e., R_{trial}) that should be used in the computation of ρ_M.
`D.aa`	A scalar that specifies the between-trial variance of the treatment effects on the surrogate endpoint (i.e., d_{aa}) that should be used in the computation of ρ_M.
`D.bb`	A scalar that specifies the between-trial variance of the treatment effects on the true endpoint (i.e., d_{bb}) that should be used in the computation of ρ_M.
`T0S0`	A scalar or vector that specifies the correlation(s) between the surrogate and the true endpoint in the control treatment condition that should be considered in the computation of ρ_M.
`T1S1`	A scalar or vector that specifies the correlation(s) between the surrogate and the true endpoint in the experimental treatment condition that should be considered in the computation of ρ_M.
`T0T0`	A scalar that specifies the variance of the true endpoint in the control treatment condition that should be considered in the computation of ρ_M. Default 1.
`T1T1`	A scalar that specifies the variance of the true endpoint in the experimental treatment condition that should be considered in the computation of ρ_M. Default 1.
`S0S0`	A scalar that specifies the variance of the surrogate endpoint in the control treatment condition that should be considered in the computation of ρ_M. Default 1.
`S1S1`	A scalar that specifies the variance of the surrogate endpoint in the experimental treatment condition that should be considered in the computation of ρ_M. Default 1.

`T0T1` A scalar or vector that contains the correlation(s) between the counterfactuals T0 and T1 that should be considered in the computation of ρ_M. Default `seq(-1, 1, by=.1)`, i.e., the values -1, -0.9, -0.8, ..., 1.

`T0S1` A scalar or vector that contains the correlation(s) between the counterfactuals T0 and S1 that should be considered in the computation of ρ_M. Default `seq(-1, 1, by=.1)`.

`T1S0` A scalar or vector that contains the correlation(s) between the counterfactuals T1 and S0 that should be considered in the computation of ρ_M. Default `seq(-1, 1, by=.1)`.

`S0S1` A scalar or vector that contains the correlation(s) between the counterfactuals S0 and S1 that should be considered in the computation of ρ_M. Default `seq(-1, 1, by=.1)`.

Details

Based on the causal-inference framework, it is assumed that each subject *j* in trial *i* has four counterfactuals (or potential outcomes), i.e., T_{0ij}, T_{1ij}, S_{0ij}, and S_{1ij}. Let T_{0ij} and T_{1ij} denote the counterfactuals for the true endpoint (T) under the control ($Z = 0$) and the experimental ($Z = 1$) treatments of subject *j* in trial *i*, respectively. Similarly, S_{0ij} and S_{1ij} denote the corresponding counterfactuals for the surrogate endpoint (S) under the control and experimental treatments of subject *j* in trial *i*, respectively. The individual causal effects of Z on T and S for a given subject *j* in trial *i* are then defined as $\Delta_{T_{ij}} = T_{1ij} - T_{0ij}$ and $\Delta_{S_{ij}} = S_{1ij} - S_{0ij}$, respectively.

In the multiple-trial causal-inference framework, surrogacy can be quantified as the correlation between the individual causal effects of Z on S and T (for details, see Alonso et al., submitted):

$$\rho_M = \rho(\Delta_{Tij},\, \Delta_{Sij}) = \frac{\sqrt{d_{bb}d_{aa}}R_{trial} + \sqrt{V(\varepsilon_{\Delta Tij})V(\varepsilon_{\Delta Sij})}\rho_{\Delta}}{\sqrt{V(\Delta_{Tij})V(\Delta_{Sij})}},$$

where

$$V(\varepsilon_{\Delta Tij}) = \sigma_{T_0T_0} + \sigma_{T_1T_1} - 2\sqrt{\sigma_{T_0T_0}\sigma_{T_1T_1}}\rho_{T_0T_1},$$

$$V(\varepsilon_{\Delta Sij}) = \sigma_{S_0S_0} + \sigma_{S_1S_1} - 2\sqrt{\sigma_{S_0S_0}\sigma_{S_1S_1}}\rho_{S_0S_1},$$

$$V(\Delta_{Tij}) = d_{bb} + \sigma_{T_0T_0} + \sigma_{T_1T_1} - 2\sqrt{\sigma_{T_0T_0}\sigma_{T_1T_1}}\rho_{T_0T_1},$$

$$V(\Delta_{Sij}) = d_{aa} + \sigma_{S_0S_0} + \sigma_{S_1S_1} - 2\sqrt{\sigma_{S_0S_0}\sigma_{S_1S_1}}\rho_{S_0S_1}.$$

The correlations between the counterfactuals (i.e., $\rho_{S_0T_1}$, $\rho_{S_1T_0}$, $\rho_{T_0T_1}$, and $\rho_{S_0S_1}$) are not identifiable from the data. It is thus warranted to conduct a sensitivity analysis (by considering vectors of possible values for the correlations between the counterfactuals – rather than point estimates).

When the user specifies a vector of values that should be considered for one or more of the correlations that are involved in the computation of ρ_M, the function `MICA.ContCont` constructs all possible matrices that can be formed as based on the specified values, identifies the matrices that are positive definite (i.e., valid correlation matrices), and computes ρ_M for each of these matrices. An examination of the vector of the obtained ρ_M values allows for a straightforward examination of the impact of different assumptions regarding the correlations between the counterfactuals on the results (see also `plot Causal-Inference ContCont`), and the extent to which proponents of the

causal-inference and meta-analytic frameworks will reach the same conclusion with respect to the appropriateness of the candidate surrogate at hand.

Notes

A single ρ_M value is obtained when all correlations in the function call are scalars.

Value

An object of class `MICA.ContCont` with components,

`Total.Num.Matrices`
: An object of class `numeric` which contains the total number of matrices that can be formed as based on the user-specified correlations.

`Pos.Def`
: A `data.frame` that contains the positive definite matrices that can be formed based on the user-specified correlations. These matrices are used to compute the vector of the ρ_M values.

`ICA`
: A scalar or vector of the ρ_Δ values.

`MICA`
: A scalar or vector of the ρ_M values.

Warning

The theory that relates the causal-inference and the meta-analytic frameworks in the multiple-trial setting (as developped in Alonso et al., submitted) assumes that a reduced or semi-reduced modelling approach is used in the meta-analytic framework. Thus R_{trial}, d_{aa} and d_{bb} should be estimated based on a reduced model (i.e., using the `Model=c("Reduced")` argument in the functions `UnifixedContCont`, `UnimixedContCont`, `BifixedContCont`, or `BimixedContCont`) or based on a semi-reduced model (i.e., using the `Model=c("SemiReduced")` argument in the functions `UnifixedContCont`, `UnimixedContCont`, or `BifixedContCont`).

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., Molenberghs, G., Buyse, M., & Burzykowski, T. (submitted). On the relationship between the causal-inference and meta-analytic paradigms for the validation of surrogate markers.

See Also

`ICA.ContCont`, `MICA.Sample.ContCont`, `plot Causal-Inference ContCont`, `UnifixedContCont`, `UnimixedContCont`, `BifixedContCont`, `BimixedContCont`

Examples

```
# Generate the vector of MICA values when R_trial=.8, rho_T0S0=rho_T1S1=.8,
# sigma_T0T0=90, sigma_T1T1=100,sigma_ S0S0=10, sigma_S1S1=15, D.aa=5, D.bb=10,
# and when the grid of values {0, .2, ..., 1} is considered for the
# correlations between the counterfactuals:
```

```
SurMICA <- MICA.ContCont(Trial.R=.80, D.aa=5, D.bb=10, T0S0=.8, T1S1=.8,
T0T0=90, T1T1=100, S0S0=10, S1S1=15, T0T1=seq(0, 1, by=.2),
T0S1=seq(0, 1, by=.2), T1S0=seq(0, 1, by=.2), S0S1=seq(0, 1, by=.2))

# Examine and plot the vector of the generated MICA values:
summary(SurMICA)
plot(SurMICA)


# Same analysis, but now assume that D.aa=.5 and D.bb=.1:
SurMICA <- MICA.ContCont(Trial.R=.80, D.aa=.5, D.bb=.1, T0S0=.8, T1S1=.8,
T0T0=90, T1T1=100, S0S0=10, S1S1=15, T0T1=seq(0, 1, by=.2),
T0S1=seq(0, 1, by=.2), T1S0=seq(0, 1, by=.2), S0S1=seq(0, 1, by=.2))

# Examine and plot the vector of the generated MICA values:
summary(SurMICA)
plot(SurMICA)


# Same as first analysis, but specify vectors for rho_T0S0 and rho_T1S1:
# Sample from normal with mean .8 and SD=.1 (to account for uncertainty
# in estimation)
SurMICA <- MICA.ContCont(Trial.R=.80, D.aa=5, D.bb=10,
T0S0=rnorm(n=10000000, mean=.8, sd=.1),
T1S1=rnorm(n=10000000, mean=.8, sd=.1),
T0T0=90, T1T1=100, S0S0=10, S1S1=15, T0T1=seq(0, 1, by=.2),
T0S1=seq(0, 1, by=.2), T1S0=seq(0, 1, by=.2), S0S1=seq(0, 1, by=.2))
```

`MICA.Sample.ContCont` *Assess surrogacy in the causal-inference multiple-trial setting (Meta-analytic Individual Causal Association; MICA) in the continuous-continuous case using the grid-based sample approach*

Description

The function `MICA.Sample.ContCont` quantifies surrogacy in the multiple-trial causal-inference framework. It provides a faster alternative for `MICA.ContCont`. See **Details** below.

Usage

```
MICA.Sample.ContCont(Trial.R, D.aa, D.bb, T0S0, T1S1, T0T0=1, T1T1=1, S0S0=1, S1S1=1,
T0T1=seq(-1, 1, by=.001), T0S1=seq(-1, 1, by=.001), T1S0=seq(-1, 1, by=.001),
S0S1=seq(-1, 1, by=.001), M=50000)
```

Arguments

`Trial.R`	A scalar that specifies the trial-level correlation coefficient (i.e., R_{trial}) that should be used in the computation of ρ_M.

`D.aa` A scalar that specifies the between-trial variance of the treatment effects on the surrogate endpoint (i.e., d_{aa}) that should be used in the computation of ρ_M.

`D.bb` A scalar that specifies the between-trial variance of the treatment effects on the true endpoint (i.e., d_{bb}) that should be used in the computation of ρ_M.

`T0S0` A scalar or vector that specifies the correlation(s) between the surrogate and the true endpoint in the control treatment condition that should be considered in the computation of ρ_M.

`T1S1` A scalar or vector that specifies the correlation(s) between the surrogate and the true endpoint in the experimental treatment condition that should be considered in the computation of ρ_M.

`T0T0` A scalar that specifies the variance of the true endpoint in the control treatment condition that should be considered in the computation of ρ_M. Default 1.

`T1T1` A scalar that specifies the variance of the true endpoint in the experimental treatment condition that should be considered in the computation of ρ_M. Default 1.

`S0S0` A scalar that specifies the variance of the surrogate endpoint in the control treatment condition that should be considered in the computation of ρ_M. Default 1.

`S1S1` A scalar that specifies the variance of the surrogate endpoint in the experimental treatment condition that should be considered in the computation of ρ_M. Default 1.

`T0T1` A scalar or vector that contains the correlation(s) between the counterfactuals T0 and T1 that should be considered in the computation of ρ_M. Default `seq(-1, 1, by=.001)`.

`T0S1` A scalar or vector that contains the correlation(s) between the counterfactuals T0 and S1 that should be considered in the computation of ρ_M. Default `seq(-1, 1, by=.001)`.

`T1S0` A scalar or vector that contains the correlation(s) between the counterfactuals T1 and S0 that should be considered in the computation of ρ_M. Default `seq(-1, 1, by=.001)`.

`S0S1` A scalar or vector that contains the correlation(s) between the counterfactuals S0 and S1 that should be considered in the computation of ρ_M. Default `seq(-1, 1, by=.001)`.

`M` The number of runs that should be conducted. Default `50000`.

Details

Based on the causal-inference framework, it is assumed that each subject *j* in trial *i* has four counterfactuals (or potential outcomes), i.e., T_{0ij}, T_{1ij}, S_{0ij}, and S_{1ij}. Let T_{0ij} and T_{1ij} denote the counterfactuals for the true endpoint (T) under the control ($Z = 0$) and the experimental ($Z = 1$) treatments of subject *j* in trial *i*, respectively. Similarly, S_{0ij} and S_{1ij} denote the corresponding counterfactuals for the surrogate endpoint (S) under the control and experimental treatments of subject *j* in trial *i*, respectively. The individual causal effects of Z on T and S for a given subject *j* in trial *i* are then defined as $\Delta_{T_{ij}} = T_{1ij} - T_{0ij}$ and $\Delta_{S_{ij}} = S_{1ij} - S_{0ij}$, respectively.

In the multiple-trial causal-inference framework, surrogacy can be quantified as the correlation between the individual causal effects of Z on S and T (for details, see Alonso et al., submitted):

$$\rho_M = \rho(\Delta_{Tij},\, \Delta_{Sij}) = \frac{\sqrt{d_{bb}d_{aa}}R_{trial} + \sqrt{V(\varepsilon_{\Delta Tij})V(\varepsilon_{\Delta Sij})}\rho_{\Delta}}{\sqrt{V(\Delta_{Tij})V(\Delta_{Sij})}},$$

where

$$V(\varepsilon_{\Delta Tij}) = \sigma_{T_0T_0} + \sigma_{T_1T_1} - 2\sqrt{\sigma_{T_0T_0}\sigma_{T_1T_1}}\rho_{T_0T_1},$$

$$V(\varepsilon_{\Delta Sij}) = \sigma_{S_0S_0} + \sigma_{S_1S_1} - 2\sqrt{\sigma_{S_0S_0}\sigma_{S_1S_1}}\rho_{S_0S_1},$$

$$V(\Delta_{Tij}) = d_{bb} + \sigma_{T_0T_0} + \sigma_{T_1T_1} - 2\sqrt{\sigma_{T_0T_0}\sigma_{T_1T_1}}\rho_{T_0T_1},$$

$$V(\Delta_{Sij}) = d_{aa} + \sigma_{S_0S_0} + \sigma_{S_1S_1} - 2\sqrt{\sigma_{S_0S_0}\sigma_{S_1S_1}}\rho_{S_0S_1}.$$

The correlations between the counterfactuals (i.e., $\rho_{S_0T_1}$, $\rho_{S_1T_0}$, $\rho_{T_0T_1}$, and $\rho_{S_0S_1}$) are not identifiable from the data. It is thus warranted to conduct a sensitivity analysis (by considering vectors of possible values for the correlations between the counterfactuals – rather than point estimates).

When the user specifies a vector of values that should be considered for one or more of the correlations that are involved in the computation of ρ_M, the function `MICA.ContCont` constructs all possible matrices that can be formed as based on the specified values, and retains the positive definite ones for the computation of ρ_M.

In contrast, the function `MICA.Sample.ContCont` samples random values for $\rho_{S_0T_1}$, $\rho_{S_1T_0}$, $\rho_{T_0T_1}$, and $\rho_{S_0S_1}$ based on a uniform distribution with user-specified minimum and maximum values, and retains the positive definite ones for the computation of ρ_M.

An examination of the vector of the obtained ρ_M values allows for a straightforward examination of the impact of different assumptions regarding the correlations between the counterfactuals on the results (see also `plot Causal-Inference ContCont`), and the extent to which proponents of the causal-inference and meta-analytic frameworks will reach the same conclusion with respect to the appropriateness of the candidate surrogate at hand.

Notes

A single ρ_M value is obtained when all correlations in the function call are scalars.

Value

An object of class `MICA.ContCont` with components,

`Total.Num.Matrices`
: An object of class `numeric` which contains the total number of matrices that can be formed as based on the user-specified correlations.

`Pos.Def`
: A `data.frame` that contains the positive definite matrices that can be formed based on the user-specified correlations. These matrices are used to compute the vector of the ρ_M values.

`ICA`
: A scalar or vector of the ρ_{Δ} values.

`MICA`
: A scalar or vector of the ρ_M values.

Warning

The theory that relates the causal-inference and the meta-analytic frameworks in the multiple-trial setting (as developped in Alonso et al., submitted) assumes that a reduced or semi-reduced modelling approach is used in the meta-analytic framework. Thus R_{trial}, d_{aa} and d_{bb} should be estimated based on a reduced model (i.e., using the `Model=c("Reduced")` argument in the functions `UnifixedContCont`, `UnimixedContCont`, `BifixedContCont`, or `BimixedContCont`) or based on a semi-reduced model (i.e., using the `Model=c("SemiReduced")` argument in the functions `UnifixedContCont`, `UnimixedContCont`, or `BifixedContCont`).

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., Molenberghs, G., Buyse, M., & Burzykowski, T. (submitted). On the relationship between the causal-inference and meta-analytic paradigms for the validation of surrogate markers.

See Also

`ICA.ContCont`, `MICA.ContCont`, `plot Causal-Inference ContCont`, `UnifixedContCont`, `UnimixedContCont`, `BifixedContCont`, `BimixedContCont`

Examples

```
## Not run:  #Time consuming (>5 sec) code part
# Generate the vector of MICA values when R_trial=.8, rho_T0S0=rho_T1S1=.8,
# sigma_T0T0=90, sigma_T1T1=100,sigma_ S0S0=10, sigma_S1S1=15, D.aa=5, D.bb=10,
# and when the grid of values {-1, -0.999, ..., 1} is considered for the
# correlations between the counterfactuals:
SurMICA <- MICA.Sample.ContCont(Trial.R=.80, D.aa=5, D.bb=10, T0S0=.8, T1S1=.8,
T0T0=90, T1T1=100, S0S0=10, S1S1=15, T0T1=seq(-1, 1, by=.001),
T0S1=seq(-1, 1, by=.001), T1S0=seq(-1, 1, by=.001),
S0S1=seq(-1, 1, by=.001), M=10000)

# Examine and plot the vector of the generated MICA values:
summary(SurMICA)
plot(SurMICA, ICA=FALSE, MICA=TRUE)


# Same analysis, but now assume that D.aa=.5 and D.bb=.1:
SurMICA <- MICA.Sample.ContCont(Trial.R=.80, D.aa=.5, D.bb=.1, T0S0=.8, T1S1=.8,
T0T0=90, T1T1=100, S0S0=10, S1S1=15, T0T1=seq(-1, 1, by=.001),
T0S1=seq(-1, 1, by=.001), T1S0=seq(-1, 1, by=.001),
S0S1=seq(-1, 1, by=.001), M=10000)

# Examine and plot the vector of the generated MICA values:
summary(SurMICA)
plot(SurMICA)
```

```
## End(Not run)
```

`MinSurrContCont` *Examine the plausibility of finding a good surrogate endpoint in the Continuous-continuous case*

Description

The function `MinSurrContCont` examines the plausibility of finding a good surrogate endpoint in the continuous-continuous setting. For details, see Alonso et al. (submitted).

Usage

```
MinSurrContCont(T0T0, T1T1, Delta, T0T1=seq(from=0, to=1, by=.01))
```

Arguments

`T0T0`	A scalar that specifies the variance of the true endpoint in the control treatment condition.
`T1T1`	A scalar that specifies the variance of the true endpoint in the experimental treatment condition.
`Delta`	A scalar that specifies an upper bound for the prediction mean squared error when predicting the individual causal effect of the treatment on the true endpoint based on the individual causal effect of the treatment on the surrogate.
`T0T1`	A scalar or vector that contains the correlation(s) between the counterfactuals T0 and T1 that should be considered in the computation of ρ^2_{min}. Default `seq(0, 1, by=.1)`, i.e., the values $0, 0.10, 0.20, \ldots, 1$.

Value

An object of class `MinSurrContCont` with components,

`T0T1`	A scalar or vector that contains the correlation(s) between the counterfactuals T0 and T1 that were considered (i.e., $\rho_{T_0T_1}$).
`Sigma.Delta.T`	A scalar or vector that contains the standard deviations of the individual causal treatment effects on the true endpoint as a function of $\rho_{T_0T_1}$.
`Rho2.Min`	A scalar or vector that contains the ρ^2_{min} values as a function of $\rho_{T_0T_1}$.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., Molenberghs, G., Buyse, M., & Burzykowski, T. (submitted). On the relationship between the causal-inference and meta-analytic paradigms for the validation of surrogate markers.

See Also

ICA.ContCont, plot Causal-Inference ContCont, plot MinSurrContCont

Examples

```
# Assess the plausibility of finding a good surrogate when
# sigma_T0T0 = sigma_T1T1 = 8 and Delta = 1
## Not run:
MinSurr <- MinSurrContCont(T0T0 = 8, T1T1 = 8, Delta = 1)
summary(MinSurr)
plot(MinSurr)
## End(Not run)
```

MixedContContIT — *Fits (univariate) mixed-effect models to assess surrogacy in the continuous-continuous case based on the Information-Theoretic framework*

Description

The function MixedContContIT uses the information-theoretic approach (Alonso & Molenberghs, 2007) to estimate trial- and individual-level surrogacy based on mixed-effect models when both S and T are continuous endpoints. The user can specify whether a (weighted or unweighted) full, semi-reduced, or reduced model should be fitted. See the **Details** section below.

Usage

```
MixedContContIT(Dataset, Surr, True, Treat, Trial.ID, Pat.ID,
Model=c("Full"), Weighted=TRUE, Min.Trial.Size=2, Alpha=.05, ...)
```

Arguments

Dataset	A data.frame that should consist of one line per patient. Each line contains (at least) a surrogate value, a true endpoint value, a treatment indicator, a patient ID, and a trial ID.
Surr	The name of the variable in Dataset that contains the surrogate endpoint values.
True	The name of the variable in Dataset that contains the true endpoint values.
Treat	The name of the variable in Dataset that contains the treatment indicators. The treatment indicator should either be coded as 1 for the experimental group and -1 for the control group, or as 1 for the experimental group and 0 for the control group.
Trial.ID	The name of the variable in Dataset that contains the trial ID to which the patient belongs.
Pat.ID	The name of the variable in Dataset that contains the patient's ID.
Model	The type of model that should be fitted, i.e., Model=c("Full"), Model=c("Reduced"), or Model=c("SemiReduced"). See the **Details** section below. Default Model=c("Full").

`Weighted` Logical. In practice it is often the case that different trials (or other clustering units) have different sample sizes. Univariate models are used to assess surrogacy in the information-theoretic approach, so it can be useful to adjust for heterogeneity in information content between the trial-specific contributions (particularly when trial-level surrogacy measures are of primary interest and when the heterogeneity in sample sizes is large). If `Weighted=TRUE`, weighted regression models are fitted. If `Weighted=FALSE`, unweighted regression analyses are conducted. See the **Details** section below. Default `TRUE`.

`Min.Trial.Size` The minimum number of patients that a trial should contain to be included in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded from the analysis. Default 2.

`Alpha` The α-level that is used to determine the confidence intervals around R_h^2 and R_{ht}^2. Default 0.05.

`...` Other arguments to be passed to the function `lmer` (of the R package `lme4`) that is used to fit the geralized linear mixed-effect models in the function `BimixedContCont`. For details, see the lme4 manual.

Details

Individual-level surrogacy

The following generalised linear mixed-effect models are fitted:

$$g_T(E(T_{ij})) = \mu_T + m_{Ti} + \beta Z_{ij} + b_i Z_{ij},$$

$$g_T(E(T_{ij}|S_{ij})) = \theta_0 + c_{Ti} + \theta_1 Z_{ij} + a_i Z_{ij} + \theta_{2i} S_{ij},$$

where i and j are the trial and subject indicators, g_T is an appropriate link function (i.e., an identity link when a continuous true endpoint is considered), S_{ij} and T_{ij} are the surrogate and true endpoint values of subject j in trial i, and Z_{ij} is the treatment indicator for subject j in trial i. μ_T and β are a fixed intercept and a fixed treatment-effect on the true endpoint, while m_{Ti} and b_i are the corresponding random effects. θ_0 and θ_1 are the fixed intercept and the fixed treatment effect on the true endpoint after accounting for the effect of the surrogate endpoint, and c_{Ti} and a_i are the corresponding random effects.

The -2 log likelihood values of the previous models (i.e., L_1 and L_2, respectively) are subsequently used to compute individual-level surrogacy (based on the so-called Variance Reduction Factor, VFR; for details, see Alonso & Molenberghs, 2007):

$$R_{hind}^2 = 1 - exp\left(-\frac{L_2 - L_1}{N}\right),$$

where N is the number of trials.

Trial-level surrogacy

When a full or semi-reduced model is requested (by using the argument `Model=c("Full")` or `Model=c("SemiReduced")` in the function call), trial-level surrogacy is assessed by fitting the following mixed models:

$$S_{ij} = \mu_S + m_{Si} + (\alpha + a_i) Z_{ij} + \varepsilon_{Sij}, (1)$$

$$T_{ij} = \mu_T + m_{Ti} + (\beta + b_i)Z_{ij} + \varepsilon_{Tij}, (1)$$

where i and j are the trial and subject indicators, S_{ij} and T_{ij} are the surrogate and true endpoint values of subject j in trial i, Z_{ij} is the treatment indicator for subject j in trial i, μ_S and μ_T are the fixed intercepts for S and T, m_{Si} and m_{Ti} are the corresponding random intercepts, α and β are the fixed treatment effects on S and T, and a_i and b_i are the corresponding random effects. The error terms ε_{Sij} and ε_{Tij} are assumed to be independent.

When a reduced model is requested by the user (by using the argument `Model=c("Reduced")` in the function call), the following univariate models are fitted:

$$S_{ij} = \mu_S + (\alpha + a_i)Z_{ij} + \varepsilon_{Sij}, (2)$$

$$T_{ij} = \mu_T + (\beta + b_i)Z_{ij} + \varepsilon_{Tij}, (2)$$

where μ_S and μ_T are the common intercepts for S and T. The other parameters are the same as defined above, and ε_{Sij} and ε_{Tij} are again assumed to be independent.

When the user requested that a full model approach is used (by using the argument `Model=c("Full")` in the function call, i.e., when models (1) were fitted), the following model is subsequently fitted:

$$\widehat{\beta}_i = \lambda_0 + \lambda_1\widehat{\mu_{Si}} + \lambda_2\widehat{\alpha}_i + \varepsilon_i, (3)$$

where the parameter estimates for β_i, μ_{Si}, and α_i are based on models (1) (see above). When a weighted model is requested (using the argument `Weighted=TRUE` in the function call), model (3) is a weighted regression model (with weights based on the number of observations in trial i). The -2 log likelihood value of the (weighted or unweighted) models (3) (L_1) is subsequently compared to the -2 log likelihood value of an intercept-only model ($\widehat{\beta}_i = \lambda_3$; L_0), and R^2_{ht} is computed based on the Variance Reduction Factor (VFR; for details, see Alonso & Molenberghs, 2007):

$$R^2_{ht} = 1 - exp\left(-\frac{L_1 - L_0}{N}\right),$$

where N is the number of trials.

When a semi-reduced or reduced model is requested (by using the argument `Model=c("SemiReduced")` or `Model=c("Reduced")` in the function call), the following model is fitted:

$$\widehat{\beta}_i = \lambda_0 + \lambda_1\widehat{\alpha}_i + \varepsilon_i,$$

where the parameter estimates for β_i and α_i are based on models (2). The -2 log likelihood value of this (weighted or unweighted) model (L_1) is subsequently compared to the -2 log likelihood value of an intercept-only model ($\widehat{\beta}_i = \lambda_3$; L_0), and R^2_{ht} is computed based on the reduction in the likelihood (as described above).

Value

An object of class `MixedContContIT` with components,

`Data.Analyze` Prior to conducting the surrogacy analysis, data of patients who have a missing value for the surrogate and/or the true endpoint are excluded. In addition, the data of trials (i) in which only one type of the treatment was administered, and (ii) in which either the surrogate or the true endpoint was a constant (i.e., all patients within a trial had the same surrogate and/or true endpoint value) are excluded. In addition, the user can specify the minimum number of patients that a trial should contain in order to include the trial in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded. `Data.Analyze` is the dataset on which the surrogacy analysis was conducted.

`Obs.Per.Trial` A `data.frame` that contains the total number of patients per trial and the number of patients who were administered the control treatment and the experimental treatment in each of the trials (in `Data.Analyze`).

`Trial.Spec.Results` A `data.frame` that contains the trial-specific intercepts and treatment effects for the surrogate and the true endpoints (when a full or semi-reduced model is requested), or the trial-specific treatment effects for the surrogate and the true endpoints (when a reduced model is requested).

`R2ht` A `data.frame` that contains the trial-level surrogacy estimate and its confidence interval.

`R2h.ind` A `data.frame` that contains the individual-level surrogacy estimate and its confidence interval.

`Cor.Endpoints` A `data.frame` that contains the correlations between the surrogate and the true endpoint in the control treatment group (i.e., ρ_{T0S0}) and in the experimental treatment group (i.e., ρ_{T1S1}), their standard errors and their confidence intervals.

`Residuals` A `data.frame` that contains the residuals for the surrogate and true endpoints (ε_{Sij} and ε_{Tij}) that are obtained when models (1) or models (2) are fitted (see the **Details** section above).

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A, & Molenberghs, G. (2007). Surrogate marker evaluation from an information theory perspective. *Biometrics, 63,* 180-186.

See Also

`FixedContContIT`, `plot Information-Theoretic`

Examples

```
# Example 1
# Based on the ARMD data:
data(ARMD)
# Assess surrogacy based on a full mixed-effect model
# in the information-theoretic framework:
Sur <- MixedContContIT(Dataset=ARMD, Surr=Diff24, True=Diff52, Treat=Treat, Trial.ID=Center,
Pat.ID=Id, Model="Full")
# Obtain a summary of the results:
summary(Sur)

## Not run:  # Time consuming (>5sec) code part
# Example 2
# Conduct an analysis based on a simulated dataset with 2000 patients, 200 trials,
# and Rindiv=Rtrial=.8
# Simulate the data:
Sim.Data.MTS(N.Total=2000, N.Trial=200, R.Trial.Target=.8, R.Indiv.Target=.8,
Seed=123, Model="Full")
# Assess surrogacy based on a full mixed-effect model
# in the information-theoretic framework:
Sur2 <- MixedContContIT(Dataset=Data.Observed.MTS, Surr=Surr, True=True, Treat=Treat,
Trial.ID=Trial.ID, Pat.ID=Pat.ID, Model="Full")

# Show a summary of the results:
summary(Sur2)
## End(Not run)
```

Ovarian *The Ovarian dataset*

Description

This dataset combines the data that were collected in four double-blind randomized clinical trials in advanced ovarian cancer (Ovarian Cancer Meta-Analysis Project, 1991). In these trials, the objective was to examine the efficacy of cyclophosphamide plus cisplatin (CP) versus cyclophosphamide plus adriamycin plus cisplatin (CAP) to treat advanced ovarian cancer.

Usage

```
data("Ovarian")
```

Format

A data frame with 1192 observations on the following 7 variables.

`Patient` The ID number of a patient.

`Center` The center in which a patient was treated.

`Treat` The treatment indicator, coded as 0=CP (active control) and 1=CAP (experimental treatment).

`Pfs` Progression-free survival (the candidate surrogate).

`PfsInd` Censoring indicator for progression-free survival.

`Surv` Survival time (the true endpoint).

`SurvInd` Censoring indicator for survival time.

References

Ovarian Cancer Meta-Analysis Project (1991). Cclophosphamide plus cisplatin plus adriamycin versus cyclophosphamide, doxorubicin, and cisplatin chemotherapy of ovarian carcinoma: a meta-analysis. *Classic papers and current comments, 3,* 237-234.

Examples

```
data(Ovarian)
str(Ovarian)
head(Ovarian)
```

---

plot Causal-Inference BinBin

Plots the (Meta-Analytic) Individual Causal Association and related metrics when S and T are binary outcomes

---

Description

This function provides a plot that displays the frequencies, percentages, cumulative percentages or densities of the individual causal association (ICA; R^2_H or R_H), and/or the odds ratios for S and T (θ_S and θ_T).

Usage

```
## S3 method for class 'ICA.BinBin'
plot(x, R2_H=TRUE, R_H=FALSE, Theta_T=FALSE,
Theta_S=FALSE, Type="Density", Labels=FALSE, Xlab.R2_H,
Main.R2_H, Xlab.R_H, Main.R_H, Xlab.Theta_S, Main.Theta_S, Xlab.Theta_T,
Main.Theta_T, Cex.Legend=1, Cex.Position="topright",
col, Par=par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1)), ylim, ...)
```

Arguments

`x`	An object of class `ICA.BinBin`. See `ICA.BinBin`.
`R2_H`	Logical. When `R2_H=TRUE`, a plot of the R^2_H is provided. Default `TRUE`.
`R_H`	Logical. When `R_H=TRUE`, a plot of the R_H is provided. Default `FALSE`.
`Theta_T`	Logical. When `Theta_T=TRUE`, a plot of the θ_T is provided. Default `FALSE`.
`Theta_S`	Logical. When `Theta_S=TRUE`, a plot of the θ_S is provided. Default `FALSE`.

Type	The type of plot that is produced. When Type="Freq" or Type="Percent", the Y-axis shows frequencies or percentages of R_H^2, R_H, θ_T, or θ_S. When Type="CumPerc", the Y-axis shows cumulative percentages. When Type="Density", the density is shown. When the fitted object of class ICA.BinBin was obtained using a general analysis (i.e., using the Monotonicity=c("General") argument in the function call), sperate plots are provided for the different monotonicity scenarios. Default "Density".
Labels	Logical. When Labels=TRUE, the percentage of R_H^2, R_H, θ_T, or θ_S values that are equal to or larger than the midpoint value of each of the bins are displayed (on top of each bin). Default FALSE.
Xlab.R2_H	The legend of the X-axis of the R_H^2 plot.
Main.R2_H	The title of the R_H^2 plot.
Xlab.R_H	The legend of the X-axis of the R_H plot.
Main.R_H	The title of the R_H plot.
Xlab.Theta_S	The legend of the X-axis of the θ_S plot.
Main.Theta_S	The title of the θ_S plot.
Xlab.Theta_T	The legend of the X-axis of the θ_T plot.
Main.Theta_T	The title of the θ_T plot.
Cex.Legend	The size of the legend when Type="All.Densities" is used. Default Cex.Legend=1.
Cex.Position	The position of the legend, Cex.Position="topright" or Cex.Position="topleft". Default Cex.Position="topright".
col	The color of the bins. Default col <- c(8).
Par	Graphical parameters for the plot. Default par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1)).
ylim	The (min, max) values for the Y-axis.
...	Extra graphical parameters to be passed to hist().

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., Molenberghs, G., Buyse, M., & Burzykowski, T. (submitted). A causal-inference approach for the validation of surrogate endpoints based on information theory and sensitivity analysis.

See Also

ICA.BinBin

Examples

```
# Compute R2_H given the marginals,
# assuming monotonicity for S and T and grids
# pi_0111=seq(0, 1, by=.001) and
# pi_1100=seq(0, 1, by=.001)
ICA <- ICA.BinBin.Grid.Sample(pi1_1_=0.261, pi1_0_=0.285,
pi_1_1=0.637, pi_1_0=0.078, pi0_1_=0.134, pi_0_1=0.127,
Monotonicity=c("General"), M=2500, Seed=1)

# Plot the results (density of R2_H):
plot(ICA, Type="Density", R2_H=TRUE, R_H=FALSE,
Theta_T=FALSE, Theta_S=FALSE)
```

---

plot Causal-Inference ContCont

Plots the (Meta-Analytic) Individual Causal Association when S and T are continuous outcomes

---

Description

This function provides a plot that displays the frequencies, percentages, or cumulative percentages of the individual causal association (ICA; ρ_Δ) and/or the meta-analytic individual causal association (MICA; ρ_M) values. These figures are useful to examine the sensitivity of the obtained results with respect to the assumptions regarding the correlations between the counterfactuals (for details, see Alonso et al., submitted; Van der Elst et al., submitted). Optionally, it is also possible to obtain plots that are useful in the examination of the plausibility of finding a good surrogate endpoint when an object of class `ICA.ContCont` is considered.

Usage

```
## S3 method for class 'ICA.ContCont'
plot(x, Xlab.ICA, Main.ICA, Type="Percent",
Labels=FALSE, ICA=TRUE, Good.Surr=FALSE, Main.Good.Surr,
Par=par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1)), col, ...)

## S3 method for class 'MICA.ContCont'
plot(x, ICA=TRUE, MICA=TRUE, Type="Percent",
Labels=FALSE, Xlab.ICA, Main.ICA, Xlab.MICA, Main.MICA,
Par=par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1)), col, ...)
```

Arguments

`x`	An object of class `ICA.ContCont` or `MICA.ContCont`. See `ICA.ContCont` or `MICA.ContCont`.
`ICA`	Logical. When `ICA=TRUE`, a plot of the ICA is provided. Default `TRUE`.
`MICA`	Logical. This argument only has effect when the `plot()` function is applied to an object of class `MICA.ContCont`. When `MICA=TRUE`, a plot of the MICA is provided. Default `TRUE`.

`Type`	The type of plot that is produced. When `Type=Freq` or `Type=Percent`, the Y-axis shows frequencies or percentages of ρ_Δ, ρ_M, and/or δ. When `Type=CumPerc`, the Y-axis shows cumulative percentages of ρ_Δ, ρ_M, and/or δ. Default "Percent".
`Labels`	Logical. When `Labels=TRUE`, the percentage of ρ_Δ, ρ_M, and/or δ values that are equal to or larger than the midpoint value of each of the bins are displayed (on top of each bin). Default `FALSE`.
`Xlab.ICA`	The legend of the X-axis of the ICA plot. Default "ρ_Δ".
`Main.ICA`	The title of the ICA plot. Default "ICA".
`Xlab.MICA`	The legend of the X-axis of the MICA plot. Default "ρ_M".
`Main.MICA`	The title of the MICA plot. Default "MICA".
`Good.Surr`	Logical. When `Good.Surr=TRUE`, a plot of δ is provided. This plot is useful in the context of examinating the plausibility of finding a good surrogate endpoint. Only applies when an object of class `ICA.ContCont` is considered. For details, see Alonso et al. (submitted). Default `FALSE`.
`Main.Good.Surr`	The title of the plot of δ. Only applies when an object of class `ICA.ContCont` is considered. For details, see Alonso et al. (submitted).
`Par`	Graphical parameters for the plot. Default `par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1))`.
`col`	The color of the bins. Default `col <- c(8)`.
`...`	Extra graphical parameters to be passed to `hist()`.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., Molenberghs, G., Buyse, M., & Burzykowski, T. (submitted). On the relationship between the causal inference and meta-analytic paradigms for the validation of surrogate markers.

Van der Elst, W., Alonso, A., & Molenberghs, G. (submitted). An exploration of the relationship between causal inference and meta-analytic measures of surrogacy.

See Also

ICA.ContCont, MICA.ContCont, plot MinSurrContCont

Examples

```
# Plot of ICA

# Generate the vector of ICA values when rho_T0S0=rho_T1S1=.95, and when the
# grid of values {0, .2, ..., 1} is considered for the correlations
# between the counterfactuals:
SurICA <- ICA.ContCont(T0S0=.95, T1S1=.95, T0T1=seq(0, 1, by=.2), T0S1=seq(0, 1, by=.2),
T1S0=seq(0, 1, by=.2), S0S1=seq(0, 1, by=.2))
```

```
# Plot the results:
plot(SurICA)

# Same plot but add the percentages of ICA values that are equal to or larger
# than the midpoint values of the bins
plot(SurICA, Labels=TRUE)

# Plot of both ICA and MICA

# Generate the vector of ICA and MICA values when R_trial=.8, rho_T0S0=rho_T1S1=.8,
# D.aa=5, D.bb=10, and when the grid of values {0, .2, ..., 1} is considered
# for the correlations between the counterfactuals:
SurMICA <- MICA.ContCont(Trial.R=.80, D.aa=5, D.bb=10, T0S0=.8, T1S1=.8,
T0T1=seq(0, 1, by=.2), T0S1=seq(0, 1, by=.2), T1S0=seq(0, 1, by=.2),
S0S1=seq(0, 1, by=.2))

# Plot the vector of generated ICA and MICA values
plot(SurMICA, ICA=TRUE, MICA=TRUE)
```

`plot FixedDiscrDiscrIT`

Provides plots of trial-level surrogacy in the Information-Theoretic framework

---

Description

Produces plots that provide a graphical representation of trial level surrogacy R^2_{ht} based on the Information-Theoretic approach of Alonso & Molenberghs (2007).

Usage

```
 ## S3 method for class 'FixedDiscrDiscrIT'
plot(x, Weighted=TRUE, Xlab.Trial, Ylab.Trial, Main.Trial,
 Par=par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1)), ...)
```

Arguments

`x`	An object of class `FixedDiscrDiscrIT`.
`Weighted`	Logical. This argument only has effect when the user requests a trial-level surrogacy plot (i.e., when `Trial.Level=TRUE` in the function call). If `Weighted=TRUE`, the circles that depict the trial-specific treatment effects on the true endpoint against the surrogate endpoint are proportional to the number of patients in the trial. If `Weighted=FALSE`, all circles have the same size. Default `TRUE`.
`Xlab.Trial`	The legend of the X-axis of the plot that depicts trial-level surrogacy. Default "Treatment effect on the surrogate endpoint (α_i)".
`Ylab.Trial`	The legend of the Y-axis of the plot that depicts trial-level surrogacy. Default "Treatment effect on the true endpoint (β_i)".

Main.Trial	The title of the plot that depicts trial-level surrogacy. Default "Trial-level surrogacy".
Par	Graphical parameters for the plot. Default par(oma=c(0, 0, 0, 0),mar=c(5.1, 4.1, 4.1, 2.1)).
...	Extra graphical parameters to be passed to plot().

Author(s)

Hannah M. Ensor & Christopher J. Weir

References

Alonso, A, & Molenberghs, G. (2007). Surrogate marker evaluation from an information theory perspective. *Biometrics, 63,* 180-186.

See Also

FixedDiscrDiscrIT

Examples

```
## Not run:  # Time consuming (>5sec) code part
# Simulate the data:
Sim.Data.MTS(N.Total=2000, N.Trial=100, R.Trial.Target=.8, R.Indiv.Target=.8,
             Seed=123, Model="Full")

# create a binary true and ordinal surrogate outcome
Data.Observed.MTS$True<-findInterval(Data.Observed.MTS$True,
        c(quantile(Data.Observed.MTS$True,0.5)))
Data.Observed.MTS$Surr<-findInterval(Data.Observed.MTS$Surr,
       c(quantile(Data.Observed.MTS$Surr,0.333),quantile(Data.Observed.MTS$Surr,0.666)))

# Assess surrogacy based on a full fixed-effect model
# in the information-theoretic framework for a binary surrogate and ordinal true outcome:
SurEval <- FixedDiscrDiscrIT(Dataset=Data.Observed.MTS, Surr=Surr, True=True, Treat=Treat,
Trial.ID=Trial.ID, Setting="ordbin")

## Request trial-level surrogacy plot. In the trial-level plot,
## make the size of the circles proportional to the number of patients in a trial:
plot(SurEval, Weighted=FALSE)


## End(Not run)
```

---

plot Information-Theoretic

Provides plots of trial- and individual-level surrogacy in the Information-Theoretic framework

---

Description

Produces plots that provide a graphical representation of trial- and/or individual-level surrogacy (R2_ht and R2_h) based on the Information-Theoretic approach of Alonso & Molenberghs (2007).

Usage

```
## S3 method for class 'FixedContContIT'
plot(x, Trial.Level=TRUE, Weighted=TRUE, Indiv.Level=TRUE,
Xlab.Indiv, Ylab.Indiv, Xlab.Trial, Ylab.Trial, Main.Trial, Main.Indiv,
Par=par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1)), ...)

## S3 method for class 'MixedContContIT'
plot(x, Trial.Level=TRUE, Weighted=TRUE, Indiv.Level=TRUE,
Xlab.Indiv, Ylab.Indiv, Xlab.Trial, Ylab.Trial, Main.Trial, Main.Indiv,
Par=par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1)), ...)
```

Arguments

x	An object of class MixedContContIT or FixedContContIT.
Trial.Level	Logical. If Trial.Level=TRUE, a plot of the trial-specific treatment effects on the true endpoint against the trial-specific treatment effect on the surrogate endpoints is provided (as a graphical representation of R_{ht}). Default TRUE.
Weighted	Logical. This argument only has effect when the user requests a trial-level surrogacy plot (i.e., when Trial.Level=TRUE in the function call). If Weighted=TRUE, the circles that depict the trial-specific treatment effects on the true endpoint against the surrogate endpoint are proportional to the number of patients in the trial. If Weighted=FALSE, all circles have the same size. Default TRUE.
Indiv.Level	Logical. If Indiv.Level=TRUE, a plot of the trial- and treatment-corrected residuals of the true and surrogate endpoints is provided. This plot provides a graphical representation of R_h. Default TRUE.
Xlab.Indiv	The legend of the X-axis of the plot that depicts individual-level surrogacy. Default "Residuals for the surrogate endpoint (ε_{Sij})".
Ylab.Indiv	The legend of the Y-axis of the plot that depicts individual-level surrogacy. Default "Residuals for the true endpoint (ε_{Tij})".
Xlab.Trial	The legend of the X-axis of the plot that depicts trial-level surrogacy. Default "Treatment effect on the surrogate endpoint (α_i)".
Ylab.Trial	The legend of the Y-axis of the plot that depicts trial-level surrogacy. Default "Treatment effect on the true endpoint (β_i)".
Main.Indiv	The title of the plot that depicts individual-level surrogacy. Default "Individual-level surrogacy".
Main.Trial	The title of the plot that depicts trial-level surrogacy. Default "Trial-level surrogacy".
Par	Graphical parameters for the plot. Default par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1)).
...	Extra graphical parameters to be passed to plot().

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A, & Molenberghs, G. (2007). Surrogate marker evaluation from an information theory perspective. *Biometrics, 63,* 180-186.

See Also

MixedContContIT, FixedContContIT

Examples

```
## Load ARMD dataset
data(ARMD)

## Conduct a surrogacy analysis, using a weighted reduced univariate fixed effect model:
Sur <- MixedContContIT(Dataset=ARMD, Surr=Diff24, True=Diff52, Treat=Treat, Trial.ID=Center,
Pat.ID=Id, Model=c("Full"))

## Request both trial- and individual-level surrogacy plots. In the trial-level plot,
## make the size of the circles proportional to the number of patients in a trial:
plot(Sur, Trial.Level=TRUE, Weighted=TRUE, Indiv.Level=TRUE)

## Make a trial-level surrogacy plot using filled blue circles that
## are transparent (to make sure that the results of overlapping trials remain
## visible), and modify the title and the axes labels of the plot:
plot(Sur, pch=16, col=rgb(.3, .2, 1, 0.3), Indiv.Level=FALSE, Trial.Level=TRUE,
Weighted=TRUE, Main.Trial=c("Trial-level surrogacy (ARMD dataset)"),
Xlab.Trial=c("Difference in vision after 6 months (Surrogate)"),
Ylab.Trial=c("Difference in vision after 12 months (True enpoint)"))

## Add the estimated R2_ht value in the previous plot at position (X=-2.2, Y=0)
## (the previous plot should not have been closed):
R2ht <- format(round(as.numeric(Sur$R2ht[1]), 3))
text(x=-2.2, y=0, cex=1.4, labels=(bquote(paste("R"[ht]^{2}, "="~.(R2ht)))))

## Make an Individual-level surrogacy plot with red squares to depict individuals
## (rather than black circles):
plot(Sur, pch=15, col="red", Indiv.Level=TRUE, Trial.Level=FALSE)
```

---

plot Information-Theoretic BinCombn

Provides plots of trial- and individual-level surrogacy in the Information-Theoretic framework when both S and T are binary, or when S is binary and T is continuous (or vice versa)

Description

Produces plots that provide a graphical representation of trial- and/or individual-level surrogacy (R2_ht and R2_hInd per cluster) based on the Information-Theoretic approach of Alonso & Molenberghs (2007).

Usage

```
## S3 method for class 'FixedBinBinIT'
plot(x, Trial.Level=TRUE, Weighted=TRUE, Indiv.Level.By.Trial=TRUE,
Xlab.Indiv, Ylab.Indiv, Xlab.Trial, Ylab.Trial, Main.Trial, Main.Indiv,
Par=par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1)), ...)

## S3 method for class 'FixedBinContIT'
plot(x, Trial.Level=TRUE, Weighted=TRUE, Indiv.Level.By.Trial=TRUE,
Xlab.Indiv, Ylab.Indiv, Xlab.Trial, Ylab.Trial, Main.Trial, Main.Indiv,
Par=par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1)), ...)

## S3 method for class 'FixedContBinIT'
plot(x, Trial.Level=TRUE, Weighted=TRUE, Indiv.Level.By.Trial=TRUE,
Xlab.Indiv, Ylab.Indiv, Xlab.Trial, Ylab.Trial, Main.Trial, Main.Indiv,
Par=par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1)), ...)
```

Arguments

`x`	An object of class `FixedBinBinIT`, `FixedBinContIT`, or `FixedContBinIT`.
`Trial.Level`	Logical. If `Trial.Level=TRUE`, a plot of the trial-specific treatment effects on the true endpoint against the trial-specific treatment effect on the surrogate endpoints is provided (as a graphical representation of R_{ht}). Default `TRUE`.
`Weighted`	Logical. This argument only has effect when the user requests a trial-level surrogacy plot (i.e., when `Trial.Level=TRUE` in the function call). If `Weighted=TRUE`, the circles that depict the trial-specific treatment effects on the true endpoint against the surrogate endpoint are proportional to the number of patients in the trial. If `Weighted=FALSE`, all circles have the same size. Default `TRUE`.
`Indiv.Level.By.Trial`	Logical. If `Indiv.Level.By.Trial=TRUE`, a plot that shows the estimated $R^2_{h.ind}$ for each trial (and confidence intervals) is provided. Default `TRUE`.
`Xlab.Indiv`	The legend of the X-axis of the plot that depicts the estimated $R^2_{h.ind}$ per trial. Default "$R[h.ind]^2$.
`Ylab.Indiv`	The legend of the Y-axis of the plot that shows the estimated $R^2_{h.ind}$ per trial. Default "Trial".
`Xlab.Trial`	The legend of the X-axis of the plot that depicts trial-level surrogacy. Default "Treatment effect on the surrogate endpoint (α_i)".
`Ylab.Trial`	The legend of the Y-axis of the plot that depicts trial-level surrogacy. Default "Treatment effect on the true endpoint (β_i)".
`Main.Indiv`	The title of the plot that depicts individual-level surrogacy. Default "Individual-level surrogacy".

`Main.Trial`	The title of the plot that depicts trial-level surrogacy. Default "Trial-level surrogacy".
`Par`	Graphical parameters for the plot. Default `par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1))`.
`...`	Extra graphical parameters to be passed to `plot()`.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A, & Molenberghs, G. (2007). Surrogate marker evaluation from an information theory perspective. *Biometrics, 63,* 180-186.

See Also

FixedBinBinIT, FixedBinContIT, FixedContBinIT

Examples

```
## Not run:  # Time consuming (>5sec) code part
# Generate data with continuous Surr and True
Sim.Data.MTS(N.Total=5000, N.Trial=50, R.Trial.Target=.9, R.Indiv.Target=.9,
             Fixed.Effects=c(0, 0, 0, 0), D.aa=10, D.bb=10, Seed=1,
             Model=c("Full"))
# Dichtomize Surr and True
Surr_Bin <- Data.Observed.MTS$Surr
Surr_Bin[Data.Observed.MTS$Surr>.5] <- 1
Surr_Bin[Data.Observed.MTS$Surr<=.5] <- 0
True_Bin <- Data.Observed.MTS$True
True_Bin[Data.Observed.MTS$True>.15] <- 1
True_Bin[Data.Observed.MTS$True<=.15] <- 0
Data.Observed.MTS$Surr <- Surr_Bin
Data.Observed.MTS$True <- True_Bin

# Assess surrogacy using info-theoretic framework
Fit <- FixedBinBinIT(Dataset = Data.Observed.MTS, Surr = Surr,
True = True, Treat = Treat, Trial.ID = Trial.ID,
Pat.ID = Pat.ID, Number.Bootstraps=100)

# Examine results
summary(Fit)
plot(Fit, Trial.Level = FALSE, Indiv.Level.By.Trial=TRUE)
plot(Fit, Trial.Level = TRUE, Indiv.Level.By.Trial=FALSE)

## End(Not run)
```

`plot MaxEntICA BinBin` *Plots the sensitivity-based and maximum entropy based Individual Causal Association when S and T are binary outcomes*

Description

This function provides a plot that displays the frequencies or densities of the individual causal association (ICA; R_H^2) as identified based on the sensitivity- (using the functions `ICA.BinBin`, `ICA.BinBin.Grid.Sample`, or `ICA.BinBin.Grid.Full`) and maximum entropy-based (using the function `MaxEntICABinBin`) approaches.

Usage

```
## S3 method for class 'MaxEntICA.BinBin'
plot(x, ICA.Fit,
Type="Density", Xlab, col, Main, ...)
```

Arguments

`x`	An object of class `MaxEntICABinBin`. See `MaxEntICABinBin`.
`ICA.Fit`	An object of class `ICA.BinBin`. See `ICA.BinBin`.
`Type`	The type of plot that is produced. When `Type="Freq"`, the Y-axis shows frequencies of R_H^2. When `Type="Density"`, the density is shown.
`Xlab`	The legend of the X-axis of the plot.
`col`	The color of the bins (frequeny plot) or line (density plot). Default `col <- c(8)`.
`Main`	The title of the plot.
`...`	Other arguments to be passed to `plot()`

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., & Van der Elst, W. (2015). A maximum-entropy approach for the evluation of surrogate endpoints based on causal inference.

See Also

ICA.BinBin, MaxEntICABinBin

Examples

```
# Sensitivity-based ICA results using ICA.BinBin.Grid.Sample
ICA <- ICA.BinBin.Grid.Sample(pi1_1_=0.341, pi0_1_=0.119, pi1_0_=0.254,
pi_1_1=0.686, pi_1_0=0.088, pi_0_1=0.078, Seed=1,
Monotonicity=c("No"), M=5000)

# Maximum-entropy based ICA
MaxEnt <- MaxEntICABinBin(pi1_1_=0.341, pi0_1_=0.119, pi1_0_=0.254,
pi_1_1=0.686, pi_1_0=0.088, pi_0_1=0.078)

# Plot results
plot(x=MaxEnt, ICA.Fit=ICA)
```

---

plot MaxEntSPF BinBin *Plots the sensitivity-based and maximum entropy based surrogate predictive function (SPF) when S and T are binary outcomes.*

Description

Plots the sensitivity-based (Alonso et al., 2015a) and maximum entropy based (Alonso et al., 2015b) surrogate predictive function (SPF), i.e., $r(i,j) = P(\Delta T = i|\Delta S = j)$, in the setting where both S and T are binary endpoints. For example, $r(-1,1)$ quantifies the probability that the treatment has a negative effect on the true endpoint ($\Delta T = -1$) given that it has a positive effect on the surrogate ($\Delta S = 1$).

Usage

```
## S3 method for class 'MaxEntSPF.BinBin'
plot(x, SPF.Fit, Type="All.Histograms", Col="grey", ...)
```

Arguments

x	A fitted object of class MaxEntSPF.BinBin. See MaxEntSPFBinBin.
SPF.Fit	A fitted object of class SPF.BinBin. See SPF.BinBin.
Type	The type of plot that is requested. Possible choices are: Type="All.Histograms", the histograms of all 9 $r(i,j) = P(\Delta T = i\|\Delta S = j)$ vectors arranged in a 3 by 3 grid; Type="All.Densities", plots of densities of all $r(i,j) = P(\Delta T = i\|\Delta S = j)$ vectors. Default Type="All.Densities".
Col	The color of the bins or lines when histograms or density plots are requested. Default "grey".
...	Other arguments to be passed to the plot() function.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., & Molenberghs, G. (2015a). Assessing a surrogate effect predictive value in a causal inference framework.

Alonso, A., & Van der Elst, W. (2015b). A maximum-entropy approach for the evluation of surrogate endpoints based on causal inference.

See Also

SPF.BinBin

Examples

```
# Sensitivity-based ICA results using ICA.BinBin.Grid.Sample
ICA <- ICA.BinBin.Grid.Sample(pi1_1_=0.341, pi0_1_=0.119, pi1_0_=0.254,
pi_1_1=0.686, pi_1_0=0.088, pi_0_1=0.078, Seed=1,
Monotonicity=c("No"), M=5000)

# Sensitivity-based SPF
SPFSens <- SPF.BinBin(ICA)

# Maximum-entropy based SPF
SPFMaxEnt <- MaxEntSPFBinBin(pi1_1_=0.341, pi0_1_=0.119, pi1_0_=0.254,
pi_1_1=0.686, pi_1_0=0.088, pi_0_1=0.078)

# Plot results
plot(x=SPFMaxEnt, SPF.Fit=SPFSens)
```

---

plot Meta-Analytic *Provides plots of trial- and individual-level surrogacy in the meta-analytic framework*

Description

Produces plots that provide a graphical representation of trial- and/or individual-level surrogacy based on the meta-analytic approach of Buyse & Molenberghs (2000) in the single- and multiple-trial settings.

Usage

```
## S3 method for class 'BifixedContCont'
plot(x, Trial.Level=TRUE, Weighted=TRUE,
Indiv.Level=TRUE, Xlab.Indiv, Ylab.Indiv, Xlab.Trial, Ylab.Trial,
Main.Trial, Main.Indiv, Par=par(oma=c(0, 0, 0, 0),
mar=c(5.1, 4.1, 4.1, 2.1)), ...)

## S3 method for class 'BimixedContCont'
plot(x, Trial.Level=TRUE, Weighted=TRUE,
Indiv.Level=TRUE, Xlab.Indiv, Ylab.Indiv, Xlab.Trial, Ylab.Trial,
```

```
Main.Trial, Main.Indiv, Par=par(oma=c(0, 0, 0, 0),
mar=c(5.1, 4.1, 4.1, 2.1)), ...)

## S3 method for class 'UnifixedContCont'
plot(x, Trial.Level=TRUE, Weighted=TRUE,
Indiv.Level=TRUE, Xlab.Indiv, Ylab.Indiv, Xlab.Trial, Ylab.Trial,
Main.Trial, Main.Indiv, Par=par(oma=c(0, 0, 0, 0),
mar=c(5.1, 4.1, 4.1, 2.1)), ...)

## S3 method for class 'UnimixedContCont'
plot(x, Trial.Level=TRUE, Weighted=TRUE,
Indiv.Level=TRUE, Xlab.Indiv, Ylab.Indiv, Xlab.Trial, Ylab.Trial,
Main.Trial, Main.Indiv, Par=par(oma=c(0, 0, 0, 0),
mar=c(5.1, 4.1, 4.1, 2.1)), ...)
```

Arguments

x	An object of class UnifixedContCont, BifixedContCont, UnimixedContCont, BimixedContCont, or Single.Trial.RE.AA.
Trial.Level	Logical. If Trial.Level=TRUE and an object of class UnifixedContCont, BifixedContCont, UnimixedContCont, or BimixedContCont is considered, a plot of the trial-specific treatment effects on the true endpoint against the trial-specific treatment effect on the surrogate endpoints is provided (as a graphical representation of R_{trial}). If Trial.Level=TRUE and an object of class Single.Trial.RE.AA is considered, a plot of the treatment effect on the true endpoint against the treatment effect on the surrogate endpoint is provided, and a regression line that goes through the origin with slope RE is added to the plot (to depict the constant RE assumption, see Single.Trial.RE.AA for details). If Trial.Level=FALSE, this plot is not provided. Default TRUE.
Weighted	Logical. This argument only has effect when the user requests a trial-level surrogacy plot (i.e., when Trial.Level=TRUE in the function call) and when an object of class UnifixedContCont, BifixedContCont, UnimixedContCont, or BimixedContCont is considered (not when an object of class Single.Trial.RE.AA is considered). If Weighted=TRUE, the circles that depict the trial-specific treatment effects on the true endpoint against the surrogate endpoint are proportional to the number of patients in the trial. If Weighted=FALSE, all circles have the same size. Default TRUE.
Indiv.Level	Logical. If Indiv.Level=TRUE, a plot of the trial- and treatment-corrected residuals of the true and surrogate endpoints is provided (when an object of class UnifixedContCont, BifixedContCont, UnimixedContCont, or BimixedContCont is considered), or a plot of the treatment-corrected residuals (when an object of class Single.Trial.RE.AA is considered). This plot provides a graphical representation of R_{indiv}. If Indiv.Level=FALSE, this plot is not provided. Default TRUE.
Xlab.Indiv	The legend of the X-axis of the plot that depicts individual-level surrogacy. Default "Residuals for the surrogate endpoint (ε_{Sij})" (without the i subscript when an object of class Single.Trial.RE.AA is considered).

Ylab.Indiv	The legend of the Y-axis of the plot that depicts individual-level surrogacy. Default "Residuals for the true endpoint (ε_{Tij})" (without the i subscript when an object of class Single.Trial.RE.AA is considered).
Xlab.Trial	The legend of the X-axis of the plot that depicts trial-level surrogacy. Default "Treatment effect on the surrogate endpoint (α_i)" (without the i subscript when an object of class Single.Trial.RE.AA is considered).
Ylab.Trial	The legend of the Y-axis of the plot that depicts trial-level surrogacy. Default "Treatment effect on the true endpoint (β_i)" (without the i subscript when an object of class Single.Trial.RE.AA is considered).
Main.Indiv	The title of the plot that depicts individual-level surrogacy. Default "Individual-level surrogacy" when an object of class UnifixedContCont, BifixedContCont, UnimixedContCont, or BimixedContCont is considered, and "Adjusted Association (rho_Z) when an object of class Single.Trial.RE.AA is considered.
Main.Trial	The title of the plot that depicts trial-level surrogacy. Default "Trial-level surrogacy" (when an object of class UnifixedContCont, BifixedContCont, UnimixedContCont, or BimixedContCont is considered) or "Relative Effect (RE)" (when an object of class Single.Trial.RE.AA is considered).
Par	Graphical parameters for the plot. Default par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1)).
...	Extra graphical parameters to be passed to plot().

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Buyse, M., Molenberghs, G., Burzykowski, T., Renard, D., & Geys, H. (2000). The validation of surrogate endpoints in meta-analysis of randomized experiments. *Biostatistics, 1,* 49-67.

See Also

UnifixedContCont, BifixedContCont, UnifixedContCont, BimixedContCont, Single.Trial.RE.AA

Examples

```
##### Multiple-trial setting

## Load ARMD dataset
data(ARMD)

## Conduct a surrogacy analysis, using a weighted reduced univariate fixed effect model:
Sur <- UnifixedContCont(Dataset=ARMD, Surr=Diff24, True=Diff52, Treat=Treat, Trial.ID=Center,
Pat.ID=Id, Number.Bootstraps=100, Model=c("Reduced"), Weighted=TRUE)

## Request both trial- and individual-level surrogacy plots. In the trial-level plot,
## make the size of the circles proportional to the number of patients in a trial:
plot(Sur, Trial.Level=TRUE, Weighted=TRUE, Indiv.Level=TRUE)

## Make a trial-level surrogacy plot using filled blue circles that
```

```
## are transparent (to make sure that the results of overlapping trials remain
## visible), and modify the title and the axes labels of the plot:
plot(Sur, pch=16, col=rgb(.3, .2, 1, 0.3), Indiv.Level=FALSE, Trial.Level=TRUE,
Weighted=TRUE, Main.Trial=c("Trial-level surrogacy (ARMD dataset)"),
Xlab.Trial=c("Difference in vision after 6 months (Surrogate)"),
Ylab.Trial=c("Difference in vision after 12 months (True enpoint)"))

## Add the estimated R2_trial value in the previous plot at position (X=-7, Y=11)
## (the previous plot should not have been closed):
R2trial <- format(round(as.numeric(Sur$Trial.R2[1]), 3))
text(x=-7, y=11, cex=1.4, labels=(bquote(paste("R"[trial]^{2}, "="~.(R2trial)))))

## Make an Individual-level surrogacy plot with red squares to depict individuals
## (rather than black circles):
plot(Sur, pch=15, col="red", Indiv.Level=TRUE, Trial.Level=FALSE)

## Same plot as before, but now with smaller squares, a y-axis with range [-40; 40],
## and the estimated R2_indiv value in the title of the plot:
R2ind <- format(round(as.numeric(Sur$Indiv.R2[1]), 3))
plot(Sur, pch=15, col="red", Indiv.Level=TRUE, Trial.Level=FALSE, cex=.5,
ylim=c(-40, 40), Main.Indiv=bquote(paste("R"[indiv]^{2}, "="~.(R2ind))))


##### Single-trial setting

## Conduct a surrogacy analysis in the single-trial meta-analytic setting:
SurSTS <- Single.Trial.RE.AA(Dataset=ARMD, Surr=Diff24, True=Diff52, Treat=Treat, Pat.ID=Id)

# Request a plot of individual-level surrogacy and a plot that depicts the Relative effect
# and the constant RE assumption:
plot(SurSTS, Trial.Level=TRUE, Indiv.Level=TRUE)
```

`plot MinSurrContCont` *Graphically illustrates the theoretical plausibility of finding a good surrogate endpoint in the continuous-continuous case*

Description

This function provides a plot that displays the frequencies, percentages, or cumulative percentages of ρ^2_{min} for a fixed value of δ (given the observed variances of the true endpoint in the control and experimental treatment conditions and a specified grid of values for the unidentified parameter $\rho_{T_0T_1}$; see `MinSurrContCont`). For details, see the online appendix of Alonso et al., submitted.

Usage

```
## S3 method for class 'MinSurrContCont'
plot(x, main, col, Type="Percent", Labels=FALSE,
Par=par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1)), ...)
```

Arguments

x	An object of class MinSurrContCont. See MinSurrContCont.
main	The title of the plot.
col	The color of the bins.
Type	The type of plot that is produced. When Type=Freq or Type=Percent, the Y-axis shows frequencies or percentages of ρ^2_{min}. When Type=CumPerc, the Y-axis shows cumulative percentages of ρ^2_{min}. Default "Percent".
Labels	Logical. When Labels=TRUE, the percentage of ρ^2_{min} values that are equal to or larger than the midpoint value of each of the bins are displayed (on top of each bin). Only applies when Type=Freq or Type=Percent. Default FALSE.
Par	Graphical parameters for the plot. Default par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1)).
...	Extra graphical parameters to be passed to hist().

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., Molenberghs, G., Buyse, M., & Burzykowski, T. (submitted). On the relationship between the causal inference and meta-analytic paradigms for the validation of surrogate markers.

See Also

MinSurrContCont

Examples

```
# compute rho^2_min in the setting where the variances of T in the control
# and experimental treatments equal 100 and 120, delta is fixed at 50,
# and the grid G={0, .01, ..., 1} is considered for the counterfactual
# correlation rho_T0T1:
MinSurr <- MinSurrContCont(T0T0 = 100, T1T1 = 120, Delta = 50,
T0T1 = seq(0, 1, by = 0.01))

# Plot the results (use percentages on Y-axis)
plot(MinSurr, Type="Percent")

# Same plot, but add the percentages of ICA values that are equal to or
# larger than the midpoint values of the bins
plot(MinSurr, Labels=TRUE)
```

plot PredTrialTContCont

Plots the expected treatment effect on the true endpoint in a new trial (when both S and T are normally distributed continuous endpoints)

Description

The key motivation to evaluate a surrogate endpoint is to be able to predict the treatment effect on the true endpoint T based on the treatment effect on S in a new trial $i = 0$. The function `Pred.TrialT.ContCont` allows for making such predictions. The present plot function shows the results graphically.

Usage

```
## S3 method for class 'PredTrialTContCont'
plot(x, Size.New.Trial=5, CI.Segment=1, ...)
```

Arguments

`x`	A fitted object of class `Pred.TrialT.ContCont`, for details see `Pred.TrialT.ContCont`.
`Size.New.Trial`	The expected treatment effect on T is drawn as a black circle with size specified by `Size.New.Trial`. Default `Size.New.Trial=5`.
`CI.Segment`	The confidence interval around the expected treatment effect on T is depicted by a dashed horizontal line. By default, the width of the horizontal line of the horizontal section of the confidence interval indicator is 2 times the values specified by `CI.Segment`. Default $CI.Segment = 1$.
`...`	Extra graphical parameters to be passed to `plot()`.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

See Also

Pred.TrialT.ContCont

Examples

```
# Generate dataset
Sim.Data.MTS(N.Total=2000, N.Trial=15, R.Trial.Target=.95,
R.Indiv.Target=.8, D.aa=10, D.bb=50,
Fixed.Effects=c(1, 2, 30, 90), Seed=1)

# Evaluate surrogacy using a reduced bivariate mixed-effects model
BimixedFit <- BimixedContCont(Dataset = Data.Observed.MTS,
Surr = Surr, True = True, Treat = Treat, Trial.ID = Trial.ID,
Pat.ID = Pat.ID, Model="Reduced")
```

```
# Suppose that in a new trial, it was estimated alpha_0 = 30
# predict beta_0 in this trial
Pred_Beta <- Pred.TrialT.ContCont(Object = BimixedFit,
alpha_0 = 30)

# Examine the results
summary(Pred_Beta)

# Plot the results
plot(Pred_Beta)
```

`plot SPF BinBin` *Plots the surrogate predictive function (SPF).*

Description

Plots the surrogate predictive function (SPF), i.e., $r(i,j) = P(\Delta T = i|\Delta S = j)$, in the setting where both S and T are binary endpoints. For example, $r(-1,1)$ quantifies the probability that the treatment has a negative effect on the true endpoint ($\Delta T = -1$) given that it has a positive effect on the surrogate ($\Delta S = 1$).

Usage

```
## S3 method for class 'SPF.BinBin'
plot(x, Type="All.Histograms", Specific.Pi="r_0_0", Col="grey",
Box.Plot.Outliers=FALSE, Legend.Pos="topleft", Legend.Cex=1, ...)
```

Arguments

`x`	A fitted object of class `SPF.BinBin`. See `ICA.BinBin`.
`Type`	The type of plot that is requested. Possible choices are: `Type="All.Histograms"`, the histograms of all 9 $r(i,j) = P(\Delta T = i|\Delta S = j)$ vectors arranged in a 3 by 3 grid; `Type="All.Densities"`, plots of densities of all $r(i,j) = P(\Delta T = i|\Delta S = j)$ vectors; `Type="Histogram"`, the histogram of a particular $r(i,j) = P(\Delta T = i|\Delta S = j)$ vector (the `Specific.Pi=` argument has to be used to specify the desired $r(i,j)$); `Type="Density"`, the density of a particular $r(i,j) = P(\Delta T = i|\Delta S = j)$ vector (the `Specific.Pi=` argument has to be used to specify the desired $r(i,j)$); `Type="Box.Plot"`, a box plot of all $r(i,j) = P(\Delta T = i|\Delta S = j)$ vectors; `Type="Lines.Mean"`, a line plot the depicts the means of all $r(i,j) = P(\Delta T = i|\Delta S = j)$ vectors; `Type="Lines.Median"`, a line plot the depicts the medians of all $r(i,j) = P(\Delta T = i|\Delta S = j)$ vectors; `Type="Lines.Mode"`, a line plot the depicts the modes of all $r(i,j) = P(\Delta T = i|\Delta S = j)$ vectors; `Type="3D.Mean"`, a 3D bar plot the depicts the means of all $r(i,j) = P(\Delta T = i|\Delta S = j)$ vectors; `Type="3D.Median"`, a 3D bar plot the depicts the medians of all $r(i,j) = P(\Delta T = i|\Delta S = j)$ vectors; `Type="3D.Mode"`, a 3D bar plot the depicts the modes of all $r(i,j) = P(\Delta T = i|\Delta S = j)$ vectors; `Type="3D.Spinning.Mean"`, a spinning 3D plot

that depicts the means of all $r(i,j) = P(\Delta T = i|\Delta S = j)$ vectors that can be rotated; `Type="3D.Spinning.Median"`, a spinning 3D plot that depicts the medians of all $r(i,j) = P(\Delta T = i|\Delta S = j)$ vectors that can be rotated; `Type="3D.Spinning.Mode"`, a spinning 3D plot that depicts the modes of all $r(i,j) = P(\Delta T = i|\Delta S = j)$ vectors that can be rotated.

`Specific.Pi` When `Type="Histogram"` or `Type="Density"` , the histogram/density of a particular $r(i,j) = P(\Delta T = i|\Delta S = j)$ vector is shown. The `Specific.Pi=` argument is used to specify the desired $r(i,j)$). Default `r_0_0`.

`Col` The color of the bins or lines when histograms or density plots are requested. Default `"grey"`.

`Box.Plot.Outliers`
Logical. Should outliers be depicted in the box plots?. Default `FALSE`.

`Legend.Pos` Position of the legend when a `type="Box.Plot"`, `type="Lines.Mean"`, `type="Lines.Median"`, or `type="Lines.Mode"` is requested. Default `"topleft"`.

`Legend.Cex` Size of the legend when a `type="Box.Plot"`, `type="Lines.Mean"`, `type="Lines.Median"`, or `type="Lines.Mode"` is requested. Default `1`.

`...` Arguments to be passed to the plot, histogram, ... functions.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., & Molenberghs, G. (2015). Assessing a surrogate effect predictive value in a causal inference framework.

See Also

SPF.BinBin

Examples

```
# Generate plausible values for Pi
ICA <- ICA.BinBin.Grid.Sample(pi1_1_=0.341, pi0_1_=0.119,
pi1_0_=0.254, pi_1_1=0.686, pi_1_0=0.088, pi_0_1=0.078, Seed=1,
Monotonicity=c("General"), M=2500)

# Compute the surrogate predictive function (SPF)
SPF <- SPF.BinBin(ICA)

# Explore the results
summary(SPF)

# Examples of plots
plot(SPF, Type="All.Histograms")
plot(SPF, Type="All.Densities")
plot(SPF, Type="Histogram", Specific.Pi="r_0_0")
plot(SPF, Type="Box.Plot", Legend.Pos="topleft", Legend.Cex=.7)
```

```
plot(SPF, Type="Lines.Mean")
plot(SPF, Type="Lines.Median")
plot(SPF, Type="3D.Mean")
plot(SPF, Type="3D.Median")
plot(SPF, Type="3D.Spinning.Mean")
plot(SPF, Type="3D.Spinning.Median")
```

`plot TrialLevelIT` *Provides a plots of trial-level surrogacy in the information-theoretic framework based on the output of the* `TrialLevelIT()` *function*

Description

Produces a plot that provides a graphical representation of trial-level surrogacy based on the output of the `TrialLevelIT()` function (information-theoretic framework).

Usage

```
## S3 method for class 'TrialLevelIT'
plot(x, Xlab.Trial,
Ylab.Trial, Main.Trial, Par=par(oma=c(0, 0, 0, 0),
mar=c(5.1, 4.1, 4.1, 2.1)), ...)
```

Arguments

`x`	An object of class `TrialLevelIT`.
`Xlab.Trial`	The legend of the X-axis of the plot that depicts trial-level surrogacy. Default "Treatment effect on the surrogate endpoint (α_i)".
`Ylab.Trial`	The legend of the Y-axis of the plot that depicts trial-level surrogacy. Default "Treatment effect on the true endpoint (β_i)".
`Main.Trial`	The title of the plot that depicts trial-level surrogacy. Default "Trial-level surrogacy".
`Par`	Graphical parameters for the plot. Default `par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1))`.
`...`	Extra graphical parameters to be passed to `plot()`.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Buyse, M., Molenberghs, G., Burzykowski, T., Renard, D., & Geys, H. (2000). The validation of surrogate endpoints in meta-analysis of randomized experiments. *Biostatistics, 1,* 49-67.

See Also

UnifixedContCont, BifixedContCont, UnifixedContCont, BimixedContCont, TrialLevelIT

Examples

```
# Generate vector treatment effects on S
set.seed(seed = 1)
Alpha.Vector <- seq(from = 5, to = 10, by=.1) + runif(min = -.5, max = .5, n = 51)

# Generate vector treatment effects on T
set.seed(seed=2)
Beta.Vector <- (Alpha.Vector * 3) + runif(min = -5, max = 5, n = 51)

# Apply the function to estimate R^2_{h.t}
Fit <- TrialLevelIT(Alpha.Vector=Alpha.Vector,
Beta.Vector=Beta.Vector, N.Trial=50, Model="Reduced")

# Plot the results
plot(Fit)
```

`plot TrialLevelMA` *Provides a plots of trial-level surrogacy in the meta-analytic framework based on the output of the* `TrialLevelMA()` *function*

Description

Produces a plot that provides a graphical representation of trial-level surrogacy based on the output of the `TrialLevel()` function (meta-analytic framework).

Usage

```
## S3 method for class 'TrialLevelMA'
plot(x, Weighted=TRUE, Xlab.Trial,
Ylab.Trial, Main.Trial, Par=par(oma=c(0, 0, 0, 0),
mar=c(5.1, 4.1, 4.1, 2.1)), ...)
```

Arguments

`x`	An object of class `TrialLevelMA`.
`Weighted`	Logical. If `Weighted=TRUE`, the circles that depict the trial-specific treatment effects on the true endpoint against the surrogate endpoint are proportional to the number of patients in the trial. If `Weighted=FALSE`, all circles have the same size. Default `TRUE`.
`Xlab.Trial`	The legend of the X-axis of the plot that depicts trial-level surrogacy. Default "Treatment effect on the surrogate endpoint (α_i)".
`Ylab.Trial`	The legend of the Y-axis of the plot that depicts trial-level surrogacy. Default "Treatment effect on the true endpoint (β_i)".
`Main.Trial`	The title of the plot that depicts trial-level surrogacy. Default "Trial-level surrogacy".
`Par`	Graphical parameters for the plot. Default `par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1))`.
`...`	Extra graphical parameters to be passed to `plot()`.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Buyse, M., Molenberghs, G., Burzykowski, T., Renard, D., & Geys, H. (2000). The validation of surrogate endpoints in meta-analysis of randomized experiments. *Biostatistics, 1,* 49-67.

See Also

UnifixedContCont, BifixedContCont, UnifixedContCont, BimixedContCont, TrialLevelMA

Examples

```
# Generate vector treatment effects on S
set.seed(seed = 1)
Alpha.Vector <- seq(from = 5, to = 10, by=.1) + runif(min = -.5, max = .5, n = 51)
# Generate vector treatment effects on T
set.seed(seed=2)
Beta.Vector <- (Alpha.Vector * 3) + runif(min = -5, max = 5, n = 51)
# Vector of sample sizes of the trials (here, all n_i=10)
N.Vector <- rep(10, times=51)

# Apply the function to estimate R^2_{trial}
Fit <- TrialLevelMA(Alpha.Vector=Alpha.Vector,
Beta.Vector=Beta.Vector, N.Vector=N.Vector)

# Plot the results and obtain summary
plot(Fit)
summary(Fit)
```

---

plot TwoStageSurvSurv *Plots trial-level surrogacy in the meta-analytic framework when two survival endpoints are considered.*

Description

Produces a plot that graphically depicts trial-level surrogacy when the surrogate and true endpoints are survival endpoints.

Usage

```
## S3 method for class 'TwoStageSurvSurv'
plot(x, Weighted=TRUE, xlab, ylab, main,
Par=par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1)), ...)
```

Arguments

x	An object of class TwoStageContCont.
Weighted	Logical. If Weighted=TRUE, the circles that depict the trial-specific treatment effects on the true endpoint against the surrogate endpoint are proportional to the number of patients in the trial. If Weighted=FALSE, all circles have the same size. Default TRUE.
xlab	The legend of the X-axis, default "Treatment effect on the surrogate endpoint (α_i)".
ylab	The legend of the Y-axis, default "Treatment effect on the true endpoint (β_i)".
main	The title of the plot, default "Trial-level surrogacy".
Par	Graphical parameters for the plot. Default par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1)).
...	Extra graphical parameters to be passed to plot().

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

See Also

TwoStageSurvSurv

Examples

```
# Open Ovarian dataset
data(Ovarian)
# Conduct analysis
Results <- TwoStageSurvSurv(Dataset = Ovarian, Surr = Pfs, SurrCens = PfsInd,
True = Surv, TrueCens = SurvInd, Treat = Treat, Trial.ID = Center)
# Examine results of analysis
summary(Results)
plot(Results)
```

---

plot.SurvSurv — *Provides plots of trial- and individual-level surrogacy in the Information-Theoretic framework when both S and T are time-to-event endpoints*

Description

Produces plots that provide a graphical representation of trial- and/or individual-level surrogacy (R2_ht and R2_hInd per cluster) based on the Information-Theoretic approach of Alonso & Molenberghs (2007).

Usage

```
## S3 method for class 'SurvSurv'
plot(x, Trial.Level=TRUE, Weighted=TRUE,
Indiv.Level.By.Trial=TRUE, Xlab.Indiv, Ylab.Indiv, Xlab.Trial,
Ylab.Trial, Main.Trial, Main.Indiv,
Par=par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1)), ...)
```

Arguments

x	An object of class FixedBinBinIT.
Trial.Level	Logical. If Trial.Level=TRUE, a plot of the trial-specific treatment effects on the true endpoint against the trial-specific treatment effect on the surrogate endpoints is provided (as a graphical representation of R_{ht}). Default TRUE.
Weighted	Logical. This argument only has effect when the user requests a trial-level surrogacy plot (i.e., when Trial.Level=TRUE in the function call). If Weighted=TRUE, the circles that depict the trial-specific treatment effects on the true endpoint against the surrogate endpoint are proportional to the number of patients in the trial. If Weighted=FALSE, all circles have the same size. Default TRUE.
Indiv.Level.By.Trial	Logical. If Indiv.Level.By.Trial=TRUE, a plot that shows the estimated $R^2_{h.ind}$ for each trial (and confidence intervals) is provided. Default TRUE.
Xlab.Indiv	The legend of the X-axis of the plot that depicts the estimated $R^2_{h.ind}$ per trial. Default "$R[h.ind]^2$.
Ylab.Indiv	The legend of the Y-axis of the plot that shows the estimated $R^2_{h.ind}$ per trial. Default "Trial".
Xlab.Trial	The legend of the X-axis of the plot that depicts trial-level surrogacy. Default "Treatment effect on the surrogate endpoint (α_i)".
Ylab.Trial	The legend of the Y-axis of the plot that depicts trial-level surrogacy. Default "Treatment effect on the true endpoint (β_i)".
Main.Indiv	The title of the plot that depicts individual-level surrogacy. Default "Individual-level surrogacy".
Main.Trial	The title of the plot that depicts trial-level surrogacy. Default "Trial-level surrogacy".
Par	Graphical parameters for the plot. Default par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1)).
...	Extra graphical parameters to be passed to plot().

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A, & Molenberghs, G. (2007). Surrogate marker evaluation from an information theory perspective. *Biometrics, 63,* 180-186.

See Also

SurvSurv

Examples

```
# Open Ovarian dataset
data(Ovarian)

# Conduct analysis
Fit <- SurvSurv(Dataset = Ovarian, Surr = Pfs, SurrCens = PfsInd,
True = Surv, TrueCens = SurvInd, Treat = Treat,
Trial.ID = Center, Alpha=.05)

# Examine results
summary(Fit)
plot(Fit, Trial.Level = FALSE, Indiv.Level.By.Trial=TRUE)
plot(Fit, Trial.Level = TRUE, Indiv.Level.By.Trial=FALSE)
```

Pos.Def.Matrices *Generate 4 by 4 correlation matrices and flag the positive definite ones*

Description

Based on vectors (or scalars) for the six off-diagonal correlations of a 4 by 4 matrix, the function `Pos.Def.Matrices` constructs all possible matrices that can be formed by combining the specified values, computes the minimum eigenvalues for each of these matrices, and flags the positive definite ones (i.e., valid correlation matrices).

Usage

```
Pos.Def.Matrices(T0T1=seq(0, 1, by=.2), T0S0=seq(0, 1, by=.2), T0S1=seq(0, 1,
by=.2), T1S0=seq(0, 1, by=.2), T1S1=seq(0, 1, by=.2), S0S1=seq(0, 1, by=.2))
```

Arguments

`T0T1`	A vector or scalar that specifies the correlation(s) between T0 and T1 that should be considered to construct all possible 4 by 4 matrices. Default `seq(0, 1, by=.2)`, i.e., the values $0, 0.20, \ldots, 1$.
`T0S0`	A vector or scalar that specifies the correlation(s) between T0 and S0 that should be considered to construct all possible 4 by 4 matrices. Default `seq(0, 1, by=.2)`.
`T0S1`	A vector or scalar that specifies the correlation(s) between T0 and S1 that should be considered to construct all possible 4 by 4 matrices. Default `seq(0, 1, by=.2)`.
`T1S0`	A vector or scalar that specifies the correlation(s) between T1 and S0 that should be considered to construct all possible 4 by 4 matrices. Default `seq(0, 1, by=.2)`.
`T1S1`	A vector or scalar that specifies the correlation(s) between T1 and S1 that should be considered to construct all possible 4 by 4 matrices. Default `seq(0, 1, by=.2)`.
`S0S1`	A vector or scalar that specifies the correlation(s) between S0 and S1 that should be considered to construct all possible 4 by 4 matrices. Default `seq(0, 1, by=.2)`.

Details

The generated object `Generated.Matrices` (of class `data.frame`) is placed in the workspace (for easy access).

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

See Also

`Sim.Data.Counterfactuals`

Examples

```
## Generate all 4x4 matrices that can be formed using rho(T0,S0)=rho(T1,S1)=.5
## and the grid of values 0, .2, ..., 1 for the other off-diagonal correlations:
Pos.Def.Matrices(T0T1=seq(0, 1, by=.2), T0S0=.5, T0S1=seq(0, 1, by=.2),
T1S0=seq(0, 1, by=.2), T1S1=.5, S0S1=seq(0, 1, by=.2))

## Examine the first 10 rows of the the object Generated.Matrices:
Generated.Matrices[1:10,]

## Check how many of the generated matrices are positive definite
## (counts and percentages):
table(Generated.Matrices$Pos.Def.Status)
table(Generated.Matrices$Pos.Def.Status)/nrow(Generated.Matrices)

## Make an object PosDef which contains the positive definite matrices:
PosDef <- Generated.Matrices[Generated.Matrices$Pos.Def.Status==1,]

## Shows the 10 first matrices that are positive definite:
PosDef[1:10,]
```

Pred.TrialT.ContCont *Compute the expected treatment effect on the true endpoint in a new trial (when both S and T are normally distributed continuous endpoints)*

Description

The key motivation to evaluate a surrogate endpoint is to be able to predict the treatment effect on the true endpoint T based on the treatment effect on S in a new trial $i = 0$. The function `Pred.TrialT.ContCont` allows for making such predictions based on fitted models of class `BimixedContCont`, `BifixedContCont`, `UnimixedContCont` and `UnifixedContCont`.

Usage

```
Pred.TrialT.ContCont(Object, mu_S0, alpha_0, alpha.CI=0.05)
```

Arguments

`Object`	A fitted object of class `BimixedContCont`, `BifixedContCont`, `UnimixedContCont` and `UnifixedContCont`. Some of the components in these fitted objects are needed to estimate $E(\beta + b_0)$ and its variance.
`mu_S0`	The intercept of a regression model in the new trial $i = 0$ where the surrogate endpoint is regressed on the true endpoint, i.e., $S_{[}0j] = \mu_{S0} + \alpha_0 Z_{0j} + \varepsilon_{S0j}$, where S is the surrogate endpoint, j is the patient indicator, and Z is the treatment. This argument only needs to be specified when a full model was used to examine surroacy.
`alpha_0`	The regression weight of the treatment in the regression model specified under argument `mu_S0`.
`alpha.CI`	The α-level to be used to determine the confidence interval around $E(\beta + b_0)$. Default `alpha.CI=0.05`.

Details

The key motivation to evaluate a surrogate endpoint is to be able to predict the treatment effect on the true endpoint T based on the treatment effect on S in a new trial $i = 0$.

When a so-called full (fixed or mixed) bi- or univariate model was fitted in the surrogate evaluation phase (for details, see `BimixedContCont`, `BifixedContCont`, `UnimixedContCont` and `UnifixedContCont`), this prediction is made as:

$$E(\beta + b_0 | m_{S0}, a_0) = \beta + \begin{pmatrix} d_{Sb} \\ d_{ab} \end{pmatrix}^T \begin{pmatrix} d_{SS} & D_{Sa} \\ d_{Sa} & d_{aa} \end{pmatrix}^{-1} \begin{pmatrix} \mu_{S0} - \mu_S \\ \alpha_0 - \alpha \end{pmatrix}$$

$$Var(\beta + b_0 | m_{S0}, a_0) = d_{bb} + \begin{pmatrix} d_{Sb} \\ d_{ab} \end{pmatrix}^T \begin{pmatrix} d_{SS} & D_{Sa} \\ d_{Sa} & d_{aa} \end{pmatrix}^{-1} \begin{pmatrix} d_{Sb} \\ d_{ab} \end{pmatrix},$$

where all components are defined as in `BimixedContCont`. When the univariate mixed-effects models are used or the (univariate or bivariate) fixed effects models, the fitted components contained in `D.Equiv` are used instead of those in `D`.

When a reduced-model approach was used in the surrogate evaluation phase, the prediction is made as:

$$E(\beta + b_0 | a_0) = \beta + \frac{}{d}_{ab} d_{aa} + (\alpha_0 - \alpha),$$

$$Var(\beta + b_0 | a_0) = d_{bb} - \frac{d_{ab}^2}{d_{aa}},$$

where all components are defined as in `BimixedContCont`. When the univariate mixed-effects models are used or the (univariate or bivariate) fixed effects models, the fitted components contained in `D.Equiv` are used instead of those in `D`.

Value

`Beta_0`	The predicted β_0.
`Variance`	The variance of the prediction.
`Lower`	The lower bound of the confidence interval around the expected β_0.
`Upper`	The upper bound of the confidence interval around the expected β_0.
`alpha.CI`	The α-level used to establish the confidence interval.
`Surr.Model`	The model that was used to compute β_0
.	
`alpha_0`	The slope of the regression model specified in the `Arguments` section.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Burzykowski, T., Molenberghs, G., & Buyse, M. (2005). *The evaluation of surrogate endpoints*. New York: Springer-Verlag.

See Also

`UnifixedContCont`, `BifixedContCont`, `UnimixedContCont`

Examples

```
# Generate dataset
Sim.Data.MTS(N.Total=2000, N.Trial=15, R.Trial.Target=.8,
R.Indiv.Target=.8, D.aa=10, D.bb=50, Fixed.Effects=c(1, 2, 30, 90),
Seed=1)

# Evaluate surrogacy using a reduced bivariate mixed-effects model
BimixedFit <- BimixedContCont(Dataset = Data.Observed.MTS, Surr = Surr,
True = True, Treat = Treat, Trial.ID = Trial.ID, Pat.ID = Pat.ID,
Model="Reduced")

# Suppose that in a new trial, it was estimated alpha_0 = 30
# predict beta_0 in this trial
Pred_Beta <- Pred.TrialT.ContCont(Object = BimixedFit,
alpha_0 = 30)

# Examine the results
summary(Pred_Beta)

# Plot the results
plot(Pred_Beta)
```

`Prentice` *Evaluates surrogacy based on the Prentice criteria for continuous endpoints (single-trial setting)*

Description

The function `Prentice` evaluates the validity of a potential surrogate based on the Prentice criteria (Prentice, 1989) in the setting where the candidate surrogate and the true endpoint are normally distributed endpoints.

Warning The Prentice approach is included in the *Surrogate* package for illustrative purposes (as it was the first formal approach to assess surrogacy), but this method has some severe problems that renders its use problematic (see **Details** below). It is recommended to replace the Prentice approach by a more statistically-sound approach to evaluate a surrogate (e.g., the meta-analytic methods; see the functions `UnifixedContCont`, `BifixedContCont`, `UnimixedContCont`, `BimixedContCont`).

Usage

```
Prentice(Dataset, Surr, True, Treat, Pat.ID, Alpha=.05)
```

Arguments

`Dataset`	A `data.frame` that should consist of one line per patient. Each line contains (at least) a surrogate value, a true endpoint value, a treatment indicator, a patient ID, and a trial ID.
`Surr`	The name of the variable in `Dataset` that contains the surrogate values.
`True`	The name of the variable in `Dataset` that contains the true endpoint values.
`Treat`	The name of the variable in `Dataset` that contains the treatment indicators. The treatment indicator should either be coded as 1 for the experimental group and -1 for the control group, or as 1 for the experimental group and 0 for the control group.
`Pat.ID`	The name of the variable in `Dataset` that contains the patient's ID.
`Alpha`	The α-level that is used to examine whether the Prentice criteria are fulfilled. Default 0.05.

Details

The Prentice criteria are examined by fitting the following regression models (when the surrogate and true endpoints are continuous variables):

$$S_j = \mu_S + \alpha Z_j + \varepsilon_{Sj}, (1)$$

$$T_j = \mu_T + \beta Z_j + \varepsilon_{Tj}, (2)$$

$$T_j = \mu + \gamma Z_j + \varepsilon_j, (3)$$

$$T_j = \tilde{\mu}_T + \beta_S Z_j + \gamma_Z S_j + \tilde{\varepsilon}_{Tj}, (4)$$

where the error terms of (1) and (2) have a joint zero-mean normal distribution with variance-covariance matrix

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_{SS} & \\ \sigma_{ST} & \sigma_{TT} \end{pmatrix}$$

,

and where j is the subject indicator, S_j and T_j are the surrogate and true endpoint values of subject j, and Z_j is the treatment indicator for subject j.

To be in line with the Prentice criteria, Z should have a significant effect on S in model 1 (Prentice criterion 1), Z should have a significant effect on T in model 2 (Prentice criterion 2), S should have a significant effect on T in model 3 (Prentice criterion criterion 3), and the effect of Z on T should be fully captured by S in model 4 (Prentice criterion 4).

The Prentice approach to assess surrogavy has some fundamental limitations. For example, the fourth Prentice criterion requires that the statistical test for the β_S in model 4 is non-significant. This criterion is useful to reject a poor surrogate, but it is not suitable to validate a good surrogate (i.e., a non-significant result may always be attributable to a lack of statistical power). Even when lack of power would not be an issue, the result of the statistical test to evaluate the fourth Prentice criterion cannot prove that the effect of the treatment on the true endpoint is fully captured by the surrogate.

The use of the Prentice approach to evaluate a surrogate is not recommended. Instead, consider using the single-trial meta-anlytic method (if no multiple clinical trials are available or if there is no other clustering unit in the data; see function `Single.Trial.RE.AA`) or the multiple-trial meta-analytic methods (see `UnifixedContCont`, `BifixedContCont`, `UnimixedContCont`, and `BimixedContCont`).

Value

`Prentice.Model.1`
: An object of class `lm` that contains the fitted model 1 (using the Prentice approach).

`Prentice.Model.2`
: An object of class `lm` that contains the fitted model 2 (using the Prentice approach).

`Prentice.Model.3`
: An object of class `lm` that contains the fitted model 3 (using the Prentice approach).

`Prentice.Model.4`
: An object of class `lm` that contains the fitted model 4 (using the Prentice approach).

`Prentice.Passed`
: `Logical`. If all four Prentice criteria are fulfilled, `Prentice.Passed=TRUE`. If at least one criterion is not fulfilled, `Prentice.Passed=FALSE`.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Burzykowski, T., Molenberghs, G., & Buyse, M. (2005). *The evaluation of surrogate endpoints*. New York: Springer-Verlag.

Prentice, R. L. (1989). Surrogate endpoints in clinical trials: definitions and operational criteria. *Statistics in Medicine, 8,* 431-440.

Examples

```
## Load the ARMD dataset
data(ARMD)

## Evaluate the Prentice criteria in the ARMD dataset
Prent <- Prentice(Dataset=ARMD, Surr=Diff24, True=Diff52, Treat=Treat, Pat.ID=Id)

# Summary of results
summary(Prent)
```

RandVec *Generate random vectors with a fixed sum*

Description

This function generates an n by m array x, each of whose m columns contains n random values lying in the interval [a,b], subject to the condition that their sum be equal to s. The distribution of values is uniform in the sense that it has the conditional probability distribution of a uniform distribution over the whole n-cube, given that the sum of the x's is s. The function uses the `randfixedsum` algorithm, written by Roger Stafford and implemented in MatLab. For details, see http://www.mathworks.com/matlabcentral/fileexchange/9700-random-vectors-with-fixed-sum/content/randfixedsum.m

Usage

```
RandVec(a=0, b=1, s=1, n=9, m=1, Seed=sample(1:1000, size = 1))
```

Arguments

a	The function `RandVec` generates an `n` by `m` matrix `x`. Each of the `m` columns contain `n` random values lying in the interval [a,b]. The argument `a` specifies the lower limit of the interval. Default `0`.
b	The argument `b` specifies the upper limit of the interval. Default `1`.
s	The argument `s` specifies the value to which each of the `m` generated columns should sum to. Default `1`.
n	The number of requested elements per column. Default `9`.
m	The number of requested columns. Default `1`.
Seed	The seed that is used. Default `sample(1:1000, size = 1)`.

Value

An object of class `RandVec` with components,

`RandVecOutput` The randomly generated vectors.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

The function is an R adaptation of a matlab program written by Roger Stafford. For details on the original Matlab algorithm, see: http://www.mathworks.com/matlabcentral/fileexchange/9700-random-vectors-with-fixed-sum/content/randfixedsum.m

Examples

```
# generate two vectors with 10 values ranging between 0 and 1
# where each vector sums to 1
# (uniform distribution over the whole n-cube)
Vectors <- RandVec(a=0, b=1, s=1, n=10, m=2)
sum(Vectors$RandVecOutput[,1])
sum(Vectors$RandVecOutput[,2])
```

`Restrictions.BinBin` *Examine restrictions in π_f under different montonicity assumptions for binary S and T*

Description

The function `Restrictions.BinBin` gives an overview of the restrictions in π_f under different assumptions regarding montonicity when both S and T are binary.

Usage

```
Restrictions.BinBin(pi1_1_, pi1_0_, pi_1_1, pi_1_0, pi0_1_, pi_0_1)
```

Arguments

`pi1_1_`	A scalar that contains $P(T = 1, S = 1 \mid Z = 0)$, i.e., the proability that $S = T = 1$ when under treatment $Z = 0$.
`pi1_0_`	A scalar that contains $P(T = 1, S = 0 \mid Z = 0)$.
`pi_1_1`	A scalar that contains $P(T = 1, S = 1 \mid Z = 1)$.
`pi_1_0`	A scalar that contains $P(T = 1, S = 0 \mid Z = 1)$.
`pi0_1_`	A scalar that contains $P(T = 0, S = 1 \mid Z = 0)$.
`pi_0_1`	A scalar that contains $P(T = 0, S = 1 \mid Z = 1)$.

Value

An overview of the restrictions for the freely varying parameters imposed by the data is provided

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., & Molenberghs, G. (2014). Validation of surrogate endpoints: the binary-binary setting from a causal inference perspective.

See Also

MarginalProbs

Examples

```
Restrictions.BinBin(pi1_1_=0.262, pi0_1_=0.135, pi1_0_=0.286,
pi_1_1=0.637, pi_1_0=0.078, pi_0_1=0.127)
```

Schizo *Data of five clinical trials in schizophrenia*

Description

These are the data of five clinical trials in schizophrenia. A total of 2128 patients were treated by 198 investiagators (psychiatrists). Patients' schizophrenic symptoms were measured using the PANSS, BPRS, and CGI. There were two treatment conditions (risperidone and control).

Usage

```
data(Schizo)
```

Format

A `data.frame` with 2128 observations on 9 variables.

`Id` The patient ID.

`InvestID` The ID of the investigator (psychiatrist) who treated the patient.

`Treat` The treatment indicator, coded as -1 = control and 1 = Risperidone.

`CGI` The change in the CGI score (= score at the start of the treatment - score at the end of the treatment).

`PANSS` The change in the PANSS score (= score at the start of the treatment - score at the end of the treatment).

`BPRS` The change in the PANSS score (= score at the start of the treatment - score at the end of the treatment).

PANSS_Bin The dichotomized PANSS change score, coded as 1 = a reduction of 20% or more in the PANSS score (score at the end of the treatment relative to score at the beginning of the treatment), 0 = otherwise.

BPRS_Bin The dichotomized BPRS change score, coded as 1 = a reduction of 20% or more in the BPRS score (score at the end of the treatment relative to score at the beginning of the treatment), 0 = otherwise.

CGI_Bin The sichtomized change in the CGI score, coded as 1 = a change of more than 3 points on the original scale (score at the end of the treatment relative to score at the beginning of the treatment), 0 = otherwise.

Schizo_Bin *Data of a clinical trial in Schizophrenia (with binary outcomes).*

Description

These are the data of a clinical trial in Schizophrenia (a subset of the dataset Schizo_Bin, study 1 where the patients were administered 10 mg. of haloperidol or 8 mg. of risperidone). A total of 454 patients were treated by 117 investigators (psychiatrists). Patients' schizophrenia symptoms at baseline and at the end of the study (after 8 weeks) were measured using the PANSS and BPRS. The variables BPRS_Bin and PANSS_Bin are binary outcomes that indicate whether clinically meaningful change had occurred (1 = a reduction of 20% or higher in the PANSS/BPRS scores at the last measurement compared to baseline; 0 = no such reduction; Leucht et al., 2005; Kay et al., 1988).

Usage

```
data(Schizo_Bin)
```

Format

A data.frame with 454 observations on 5 variables.

Id The patient ID.

InvestI The ID of the investigator (psychiatrist) who treated the patient.

Treat The treatment indicator, coded as −1 = control treatment (10 mg. haloperidol) and 1 = experimental treatment (8 mg. risperidone).

PANSS_Bin The dichotomized change in the PANSS score (1 = a reduction of 20% or more in the PANSS score, 0=otherwise)

BPRS_Bin The dichotomized change in the BPRS score (1 = a reduction of 20% or more in the BPRS score, 0=otherwise)

CGI_Bin The sichtomized change in the CGI score, coded as 1 = a change of more than 3 points on the original scale (score at the end of the treatment relative to score at the beginning of the treatment), 0 = otherwise.

References

Kay, S.R., Opler, L.A., & Lindenmayer, J.P. (1988). Reliability and validity of the Positive and Negative Syndrome Scale for schizophrenics. Psychiatric Research, 23, 99-110.

Leucht, S., et al. (2005). Clinical implications of Brief Psychiatric Rating Scale scores. The British Journal of Psychiarty, 187, 366-371.

`Schizo_PANSS` *Longitudinal PANSS data of five clinical trials in schizophrenia*

Description

These are the longitudinal PANSS data of five clinical trial in schizophrenia. A total of 2151 patients were treated by 198 investiagators (psychiatrists). There were two treatment conditions (risperidone and control). Patients' schizophrenic symptoms were measured using the PANSS at different time moments following start of the treatment. The variables Week1-Week8 express the change scores over time using the raw (semi-continuous) PANSS scores. The variables Week1_bin - Week8_bin are binary indicators of a 20% or higher reduction in PANSS score versus baseline. The latter corresponds to a commonly accepted criterion for defining a clinically meaningful response (Kay et al., 1988).

Usage

```
data(Schizo_PANSS)
```

Format

A `data.frame` with 2151 observations on 6 variables.

`Id` The patient ID.

`InvestID` The ID of the investigator (psychiatrist) who treated the patient.

`Treat` The treatment indicator, coded as -1 = placebo and 1 = Risperidone.

`Week1` The change in the PANSS score 1 week after starting the treatment (= score at the start of the treatment - score at 1 week after starting the treatment).

`Week2` The change in the PANSS score 2 weeks after starting the treatment.

`Week4` The change in the PANSS score 4 weeks after starting the treatment.

`Week6` The change in the PANSS score 6 weeks after starting the treatment.

`Week8` The change in the PANSS score 8 weeks after starting the treatment.

`Week1_bin` The dichotomized change in the PANSS score 1 week after starting the treatment (1=a 20% or higher reduction in PANSS score versus baseline, 0=otherwise).

`Week2_bin` The dichotomized change in the PANSS score 2 weeks after starting the treatment.

`Week4_bin` The dichotomized change in the PANSS score 4 weeks after starting the treatment.

`Week6_bin` The dichotomized change in the PANSS score 6 weeks after starting the treatment.

`Week8_bin` The dichotomized change in the PANSS score 8 weeks after starting the treatment.

References

Kay, S.R., Opler, L.A., & Lindenmayer, J.P. (1988). Reliability and validity of the Positive and Negative Syndrome Scale for schizophrenics. Psychiatric Research, 23, 99-110.

`Sim.Data.Counterfactuals`

Simulate a dataset that contains counterfactuals

Description

The function `Sim.Data.Counterfactuals` simulates a dataset that contains four (continuous) counterfactuals (i.e., potential outcomes) and a (binary) treatment indicator. The counterfactuals T_0 and T_1 denote the true endpoints of a patient under the control and the experimental treatments, respectively, and the counterfactuals S_0 and S_1 denote the surrogate endpoints of the patient under the control and the experimental treatments, respectively. The user can specify the number of patients, the desired mean values for the counterfactuals (i.e., $\boldsymbol{\mu}_c$), and the desired correlations between the counterfactuals (i.e., the off-diagonal values in the standardized $\boldsymbol{\Sigma}_c$ matrix). For details, see the papers of Alonso et al. (submitted) and Van der Elst et al. (submitted).

Usage

```
Sim.Data.Counterfactuals(N.Total=2000,
mu_c=c(0, 0, 0, 0), T0S0=0, T1S1=0, T0T1=0, T0S1=0,
T1S0=0, S0S1=0, Seed=sample(1:1000, size=1))
```

Arguments

`N.Total`	The total number of patients in the simulated dataset. Default 2000.
`mu_c`	A vector that specifies the desired means for the counterfactuals S_0, S_1, T_0, and T_1, respectively. Default `c(0, 0, 0, 0)`.
`T0S0`	A scalar that specifies the desired correlation between the counterfactuals T0 and S0 that should be used in the generation of the data. Default 0.
`T1S1`	A scalar that specifies the desired correlation between the counterfactuals T1 and S1 that should be used in the generation of the data. Default 0.
`T0T1`	A scalar that specifies the desired correlation between the counterfactuals T0 and T1 that should be used in the generation of the data. Default 0.
`T0S1`	A scalar that specifies the desired correlation between the counterfactuals T0 and S1 that should be used in the generation of the data. Default 0.
`T1S0`	A scalar that specifies the desired correlation between the counterfactuals T1 and S0 that should be used in the generation of the data. Default 0.
`S0S1`	A scalar that specifies the desired correlation between the counterfactuals T0 and T1 that should be used in the generation of the data. Default 0.
`Seed`	A seed that is used to generate the dataset. Default `sample(x=1:1000, size=1)`, i.e., a random number between 1 and 1000.

Details

The generated object `Data.Counterfactuals` (of class `data.frame`) is placed in the workspace.

The specified values for T0S0, T1S1, T0T1, T0S1, T1S0, and S0S1 in the function call should form a matrix that is positive definite (i.e., they should form a valid correlation matrix). When the user specifies values that form a matrix that is not positive definite, an error message is given and the object `Data.Counterfactuals` is not generated. The function `Pos.Def.Matrices` can be used to examine beforehand whether a 4 by 4 matrix is positive definite.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., Molenberghs, G., Buyse, M., & Burzykowski, T. (submitted). On the relationship between the causal inference and meta-analytic paradigms for the validation of surrogate markers.

Van der Elst, W., Alonso, A., & Molenberghs, G. (submitted). An exploration of the relationship between causal inference and meta-analytic measures of surrogacy.

See Also

`Sim.Data.MTS`, `Sim.Data.STS`

Examples

```
## Generate a dataset with 2000 patients, cor(S0,T0)=cor(S1,T1)=.5,
## cor(T0,T1)=cor(T0,S1)=cor(T1,S0)=cor(S0,S1)=0, with means
## 5, 9, 12, and 15 for S0, S1, T0, and T1, respectively:
Sim.Data.Counterfactuals(N=2000, T0S0=.5, T1S1=.5, T0T1=0, T0S1=0, T1S0=0, S0S1=0,
mu_c=c(5, 9, 12, 15), Seed=1)
```

`Sim.Data.CounterfactualsBinBin`

Simulate a dataset that contains counterfactuals for binary endpoints

---

Description

The function `Sim.Data.CounterfactualsBinBin` simulates a dataset that contains four (binary) counterfactuals (i.e., potential outcomes) and a (binary) treatment indicator. The counterfactuals T_0 and T_1 denote the true endpoints of a patient under the control and the experimental treatments, respectively, and the counterfactuals S_0 and S_1 denote the surrogate endpoints of the patient under the control and the experimental treatments, respectively. The user can specify the number of patients and the desired probabilities of the vector of potential outcomes (i.e., $\boldsymbol{Y'}_c$=(T_0, T_1, S_0, S_1)).

Usage

```
Sim.Data.CounterfactualsBinBin(Pi_s=rep(1/16, 16),
N.Total=2000, Seed=sample(1:1000, size=1))
```

Arguments

`Pi_s`	The vector of probabilities of the potential outcomes, i.e., pi_{0000}, pi_{0100}, pi_{0010}, pi_{0001}, pi_{0101}, pi_{1000}, pi_{1010}, pi_{1001}, pi_{1110}, pi_{1101}, pi_{1011}, pi_{1111}, pi_{0110}, pi_{0011}, pi_{0111}, pi_{1100}. Default `rep(1/16, 16)`.
`N.Total`	The desired number of patients in the simulated dataset. Default 2000.
`Seed`	A seed that is used to generate the dataset. Default `sample(x=1:1000, size=1)`, i.e., a random number between 1 and 1000.

Details

The generated object `Data.STSBinBin.Counter` (which contains the counterfactuals) and `Data.STSBinBin.Obs` (the "observable data") (of class `data.frame`) is placed in the workspace.

Value

An object of class `Sim.Data.CounterfactualsBinBin` with components,

`Data.STSBinBin.Obs`	The generated dataset that contains the "observed" surrogate endrpoint, true endpoint, and assigned treatment.
`Data.STSBinBin.Counter`	The generated dataset that contains the counterfactuals.
`Vector_Pi`	The vector of probabilities of the potential outcomes, i.e., pi_{0000}, pi_{0100}, pi_{0010}, pi_{0001}, pi_{0101}, pi_{1000}, pi_{1010}, pi_{1001}, pi_{1110}, pi_{1101}, pi_{1011}, pi_{1111}, pi_{0110}, pi_{0011}, pi_{0111}, pi_{1100}.
`Pi_Marginals`	The vector of marginal probabilities $\pi_{1\cdot1\cdot}$, $\pi_{0\cdot1\cdot}$, $\pi_{1\cdot0\cdot}$, $\pi_{0\cdot0\cdot}$, $\pi_{\cdot1\cdot1}$, $\pi_{\cdot1\cdot0}$, $\pi_{\cdot0\cdot1}$, $\pi_{\cdot0\cdot0}$.
`True.R2_H`	The true R^2_H value.
`True.Theta_T`	The true odds ratio for T.
`True.Theta_S`	The true odds ratio for S.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

Examples

```
## Generate a dataset with 2000 patients, and values 1/16
## for all proabilities between the counterfactuals:
Sim.Data.CounterfactualsBinBin(N.Total=2000)
```

`Sim.Data.MTS` *Simulates a dataset that can be used to assess surrogacy in the multiple-trial setting*

Description

The function `Sim.Data.MTS` simulates a dataset that contains the variables Treat, Trial.ID, Surr, True, and Pat.ID. The user can specify the number of patients and the number of trials that should be included in the simulated dataset, the desired R_{trial} and R_{indiv} values, the desired variability of the trial-specific treatment effects for the surrogate and the true endpoints (i.e., d_{aa} and d_{bb}, respectively), and the desired fixed-effect parameters of the intercepts and treatment effects for the surrogate and the true endpoints.

Usage

```
Sim.Data.MTS(N.Total=2000, N.Trial=50, R.Trial.Target=.8, R.Indiv.Target=.8,
Fixed.Effects=c(0, 0, 0, 0), D.aa=10, D.bb=10, Seed=sample(1:1000, size=1),
Model=c("Full"))
```

Arguments

`N.Total`	The total number of patients in the simulated dataset. Default 2000.
`N.Trial`	The number of trials. Default 50.
`R.Trial.Target`	The desired R_{trial} value in the sumilated dataset. Default 0.80
`R.Indiv.Target`	The desired R_{indiv} value in the simulated dataset. Default 0.80.
`Fixed.Effects`	A vector that specifies the desired fixed-effect intercept for the surrogate, fixed-effect intercept for the true endpoint, fixed treatment effect for the surrogate, and fixed treatment effect for the true endpoint, respectively. Default `c(0, 0, 0, 0)`.
`D.aa`	The desired variability of the trial-specific treatment effects on the surrogate endpoint. Default 10.
`D.bb`	The desired variability of the trial-specific treatment effects on the true endpoint. Default 10.
`Model`	The type of model that will be fitted on the data when surrogacy is assessed, i.e., a full, semireduced, or reduced model (for details, see `UnifixedContCont`, `UnimixedContCont`, `BifixedContCont`, `BimixedContCont`).
`Seed`	The seed that is used to generate the dataset. Default `sample(x=1:1000, size=1)`, i.e., a random number between 1 and 1000.

Details

The generated object `Data.Observed.MTS` (of class `data.frame`) is placed in the workspace (for easy access).

The number of patients per trial in the simulated dataset is identical in each trial, and equals the requested total number of patients divided by the requested number of trials (`=N.Total/N.Trial`).

If this is not a whole number, a warning is given and the number of patients per trial is automatically rounded up to the nearest whole number. See **Examples** below.

Treatment allocation is balanced when the number of patients per trial is an odd number. If this is not the case, treatment allocation is balanced up to one patient (the remaining patient is randomly allocated to the exprimental or the control treatment groups in each of the trials).

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

See Also

UnifixedContCont, BifixedContCont, UnimixedContCont, BimixedContCont, Sim.Data.STS

Examples

```
# Simulate a dataset with 2000 patients, 50 trials, Rindiv=Rtrial=.8, D.aa=10,
# D.bb=50, and fixed effect values 1, 2, 30, and 90:
Sim.Data.MTS(N.Total=2000, N.Trial=50, R.Trial.Target=.8, R.Indiv.Target=.8, D.aa=10,
D.bb=50, Fixed.Effects=c(1, 2, 30, 90), Seed=1)

# Sample output, the first 10 rows of Data.Observed.MTS:
Data.Observed.MTS[1:10,]

# Note: When the following code is used to generate a dataset:
Sim.Data.MTS(N.Total=2000, N.Trial=99, R.Trial.Target=.5, R.Indiv.Target=.8,
D.aa=10, D.bb=50, Fixed.Effects=c(1, 2, 30, 90), Seed=1)

# R gives the following warning:

# > NOTE: The number of patients per trial requested in the function call
# > equals 20.20202 (=N.Total/N.Trial), which is not a whole number.
# > To obtain a dataset where the number of patients per trial is balanced for
# > all trials, the number of patients per trial was rounded to 21 to generate
# > the dataset. Data.Observed.MTS thus contains a total of 2079 patients rather
# > than the requested 2000 in the function call.
```

Sim.Data.STS *Simulates a dataset that can be used to assess surrogacy in the single-trial setting*

Description

The function Sim.Data.STS simulates a dataset that contains the variables Treat, Surr, True, and Pat.ID. The user can specify the total number of patients, the desired R_{indiv} value (also referred to as the adjusted association (γ) in the single-trial meta-analytic setting), and the desired means of the surrogate and the true endpoints in the experimental and control treatment groups.

Usage

```
Sim.Data.STS(N.Total=2000, R.Indiv.Target=.8, Means=c(0, 0, 0, 0), Seed=
sample(1:1000, size=1))
```

Arguments

N.Total	The total number of patients in the simulated dataset. Default 2000.
R.Indiv.Target	The desired R_{indiv} (or γ) value in the simulated dataset. Default 0.80.
Means	A vector that specifies the desired mean for the surrogate in the control treatment group, mean for the surrogate in the experimental treatment group, mean for the true endpoint in the control treatment group, and mean for the true endpoint in the experimental treatment group, respectively. Default `c(0, 0, 0, 0)`.
Seed	The seed that is used to generate the dataset. Default `sample(x=1:1000, size=1)`, i.e., a random number between 1 and 1000.

Details

The generated object `Data.Observed.STS` (of class `data.frame`) is placed in the workspace (for easy access).

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

See Also

Sim.Data.MTS, Single.Trial.RE.AA

Examples

```
# Simulate a dataset:
Sim.Data.STS(N.Total=2000, R.Indiv.Target=.8, Means=c(1, 5, 20, 37), Seed=1)
```

Sim.Data.STSBinBin — *Simulates a dataset that can be used to assess surrogacy in the single trial setting when S and T are binary endpoints*

Description

The function `Sim.Data.STSBinBin` simulates a dataset that contains four (binary) counterfactuals (i.e., potential outcomes) and a (binary) treatment indicator. The counterfactuals T_0 and T_1 denote the true endpoints of a patient under the control and the experimental treatments, respectively, and the counterfactuals S_0 and S_1 denote the surrogate endpoints of the patient under the control and the experimental treatments, respectively. In addition, the function provides the "observable" data based on the dataset of the counterfactuals, i.e., the S and T endpoints given the treatment that was allocated to a patient. The user can specify the assumption regarding monotonicity that should be made to generate the data (no monotonicity, monotonicity for S alone, monotonicity for T alone, or monotonicity for both S and T).

Usage

```
Sim.Data.STSBinBin(Monotonicity=c("No"), N.Total=2000, Seed)
```

Arguments

`Monotonicity`	The assumption regarding monotonicity that should be made when the data are generated, i.e., `Monotonicity="No"` (no monotonicity assumed), `Monotonicity="True.Endp"` (monotonicity assumed for the true endpoint alone), `Monotonicity="Surr.Endp"` (monotonicity assumed for the surrogate endpoint alone), and `Monotonicity="Surr.True.Endp"` (monotonicity assumed for both endpoints). Default `Monotonicity="No"`.
`N.Total`	The desired number of patients in the simulated dataset. Default 2000.
`Seed`	A seed that is used to generate the dataset. Default `sample(x=1:1000, size=1)`, i.e., a random number between 1 and 1000.

Details

The generated objects `Data.STSBinBin_Counterfactuals` (which contains the counterfactuals) and `Data.STSBinBin_Obs` (which contains the observable data) of class `data.frame` are placed in the workspace. Other relevant output can be accessed based on the fitted object (see $Value$ below)

Value

An object of class `Sim.Data.STSBinBin` with components,

`Data.STSBinBin.Obs`	The generated dataset that contains the "observed" surrogate endrpoint, true endpoint, and assigned treatment.
`Data.STSBinBin.Counter`	The generated dataset that contains the counterfactuals.
`Vector_Pi`	The vector of probabilities of the potential outcomes, i.e., pi_{0000}, pi_{0100}, pi_{0010}, pi_{0001}, pi_{0101}, pi_{1000}, pi_{1010}, pi_{1001}, pi_{1110}, pi_{1101}, pi_{1011}, pi_{1111}, pi_{0110}, pi_{0011}, pi_{0111}, pi_{1100}.
`Pi_Marginals`	The vector of marginal probabilities $\pi_{1\cdot1\cdot}$, $\pi_{0\cdot1\cdot}$, $\pi_{1\cdot0\cdot}$, $\pi_{0\cdot0\cdot}$, $\pi_{\cdot1\cdot1}$, $\pi_{\cdot1\cdot0}$, $\pi_{\cdot0\cdot1}$, $\pi_{\cdot0\cdot0}$.
`True.R2_H`	The true R^2_H value.
`True.Theta_T`	The true odds ratio for T.
`True.Theta_S`	The true odds ratio for S.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

Examples

```
## Generate a dataset with 2000 patients,
## assuming no monotonicity:
Sim.Data.STSBinBin(Monotonicity=c("No"), N.Total=200)
```

`Single.Trial.RE.AA` *Conducts a surrogacy analysis based on the single-trial meta-analytic framework*

Description

The function `Single.Trial.RE.AA` conducts a surrogacy analysis based on the single-trial meta-analytic framework of Buyse & Molenberghs (1998). See **Details** below.

Usage

```
Single.Trial.RE.AA(Dataset, Surr, True, Treat, Pat.ID, Alpha=.05,
Number.Bootstraps=500, Seed=sample(1:1000, size=1))
```

Arguments

`Dataset`	A `data.frame` that should consist of one line per patient. Each line contains (at least) a surrogate value, a true endpoint value, a treatment indicator, and a patient ID.
`Surr`	The name of the variable in `Dataset` that contains the surrogate values.
`True`	The name of the variable in `Dataset` that contains the true endpoint values.
`Treat`	The name of the variable in `Dataset` that contains the treatment indicators. The treatment indicator should either be coded as 1 for the experimental group and -1 for the control group, or as 1 for the experimental group and 0 for the control group. The $-1/1$ coding is recommended.
`Pat.ID`	The name of the variable in `Dataset` that contains the patient's ID.
`Alpha`	The α-level that is used to determine the confidence intervals around Alpha (which is a parameter estimate of a model where the surrogate is regressed on the treatment indicator, see **Details** below), Beta, RE, and γ. Default 0.05.
`Number.Bootstraps`	The number of bootstrap samples that are used to obtain the bootstrapp-based confidence intervals for RE and the adjusted association (γ). Default 500.
`Seed`	The seed that is used to generate the bootstrap samples. Default `sample(x=1:1000, size=1)`, i.e., a random number between 1 and 1000.

Details

The Relative Effect (RE) and the adjusted association (γ) are based on the following bivariate regression model (when the surrogate and the true endpoints are continuous variables):

$$S_j = \mu_S + \alpha Z_j + \varepsilon_{Sj},$$

$$T_j = \mu_T + \beta Z_j + \varepsilon_{Tj},$$

where the error terms have a joint zero-mean normal distribution with variance-covariance matrix:

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_{SS} & \\ \sigma_{ST} & \sigma_{TT} \end{pmatrix},$$

and where j is the subject indicator, S_j and T_j are the surrogate and true endpoint values of patient j, and Z_j is the treatment indicator for patient j.

The parameter estimates of the fitted regression model and the variance-covariance matrix of the residuals are used to compute RE and the adjusted association (γ), respectively:

$$RE = \frac{\beta}{\alpha},$$

$$\gamma = \frac{\sigma_{ST}}{\sqrt{\sigma_{SS}\sigma_{TT}}}.$$

Note

The single-trial meta-analytic framework is hampered by a number of issues (Burzykowski et al., 2005). For example, a key motivation to validate a surrogate endpoint is to be able to predict the effect of Z on T as based on the effect of Z on S in a new clinical trial where T is not (yet) observed. The RE allows for such a prediction, but this requires the assumption that the relation between α and β can be described by a linear regression model that goes through the origin. In other words, it has to be assumed that the RE remains constant across clinical trials. The constant RE assumption is unverifiable in a single-trial setting, but a way out of this problem is to combine the information of multiple clinical trials and generalize the RE concept to a multiple-trial setting (as is done in the multiple-trial meta-analytic approach, see `UnifixedContCont`, `BifixedContCont`, `UnimixedContCont`, and `BimixedContCont`).

Value

An object of class `Single.Trial.RE.AA` with components,

`Data.Analyze`	Prior to conducting the surrogacy analysis, data of patients who have a missing value for the surrogate and/or the true endpoint are excluded. `Data.Analyze` is the dataset on which the surrogacy analysis was conducted.
`Alpha`	An object of class `data.frame` that contains the parameter estimate for α, its standard error, and its confidence interval. Note that `Alpha` is not to be confused with the `Alpha` argument in the function call, which specifies the α-level of the confidence intervals of the parameters.
`Beta`	An object of class `data.frame` that contains the parameter estimate for β, its standard error, and its confidence interval.
`RE.Delta`	An object of class `data.frame` that contains the estimated RE, its standard error, and its confidence interval (based on the Delta method).
`RE.Fieller`	An object of class `data.frame` that contains the estimated RE, its standard error, and its confidence interval (based on Fieller's theorem).
`RE.Boot`	An object of class `data.frame` that contains the estimated RE, its standard error, and its confidence interval (based on bootstrapping). Note that the occurence of outliers in the sample of bootstrapped RE values may lead to standard errors

and/or confidence intervals that are not trustworthy. Such problems mainly occur when the parameter estimate for α is close to 0 (taking its standard error into account). To detect possible outliers, studentized deleted residuals are computed (by fitting an intercept-only model with the bootstrapped RE values as the outcome variable). Bootstrapped RE values with an absolute studentized residual larger than $t(1-\alpha/2n; n-2)$ are marked as outliers (where n = the number of bootstrapped RE values; Kutner et al., 2005). A warning is given when outliers are found, and the position of the outlier(s) in the bootstrap sample is identified. Inspection of the vector of bootstrapped RE values (see `RE.Boot.Samples` below) is recommended in this situation, and/or the use of the confidence intervals that are based on the Delta method or Fieller's theorem (rather than the bootstrap-based confidence interval).

`AA` An object of class `data.frame` that contains the adjusted association (i.e., γ), its standard error, and its confidence interval (based on the Fisher-Z transformation procedure).

`AA.Boot` An object of class `data.frame` that contains the adjusted association (i.e., γ), its standard error, and its confidence interval (based on a bootstrap procedure).

`RE.Boot.Samples`
A vector that contains the RE values that were generated during the bootstrap procedure.

`AA.Boot.Samples`
A vector that contains the adjusted association (i.e., γ) values that were generated during the bootstrap procedure.

`Cor.Endpoints` A `data.frame` that contains the correlations between the surrogate and the true endpoint in the control treatment group (i.e., ρ_{T0T1}) and in the experimental treatment group (i.e., ρ_{T1S1}), their standard errors and their confidence intervals.

`Residuals` A `data.frame` that contains the residuals for the surrogate and true endpoints that are obtained when the surrogate and the true endpoint are regressed on the treatment indicator.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Burzykowski, T., Molenberghs, G., & Buyse, M. (2005). *The evaluation of surrogate endpoints*. New York: Springer-Verlag.

Buyse, M., & Molenberghs, G. (1998). The validation of surrogate endpoints in randomized experiments. *Biometrics, 54,* 1014-1029.

Buyse, M., Molenberghs, G., Burzykowski, T., Renard, D., & Geys, H. (2000). The validation of surrogate endpoints in meta-analysis of randomized experiments. *Biostatistics, 1,* 49-67.

Kutner, M. H., Nachtsheim, C. J., Neter, J., & Li, W. (2005). *Applied linear statistical models (5th ed.)*. New York: McGraw Hill.

See Also

UnifixedContCont, BifixedContCont, UnimixedContCont, BimixedContCont, ICA.ContCont

Examples

```
# Example 1, based on the ARMD data:
data(ARMD)

# Assess surrogacy based on the single-trial meta-analytic approach:
Sur <- Single.Trial.RE.AA(Dataset=ARMD, Surr=Diff24, True=Diff52, Treat=Treat, Pat.ID=Id)

# Obtain a summary and plot of the results
summary(Sur)
plot(Sur)


# Example 2
# Conduct an analysis based on a simulated dataset with 2000 patients
# and Rindiv=.8
# Simulate the data:
Sim.Data.STS(N.Total=2000, R.Indiv.Target=.8, Seed=123)

# Assess surrogacy:
Sur2 <- Single.Trial.RE.AA(Dataset=Data.Observed.STS, Surr=Surr, True=True, Treat=Treat,
Pat.ID=Pat.ID)

# Show a summary and plots of results
summary(Sur2)
plot(Sur2)
```

SPF.BinBin *Evaluate the surrogate predictive function (SPF) in the binary-binary setting (sensitivity-analysis based approach)*

Description

Computes the surrogate predictive function (SPF) based on sensitivity-analyis, i.e., $r(i,j) = P(\Delta T = i|\Delta S = j)$, in the setting where both S and T are binary endpoints. For example, $r(-1,1)$ quantifies the probability that the treatment has a negative effect on the true endpoint ($\Delta T = -1$) given that it has a positive effect on the surrogate ($\Delta S = 1$). All quantities of interest are derived from the vectors of 'plausible values' for π (i.e., vectors π that are compatible with the observable data at hand). See **Details** below.

Usage

```
SPF.BinBin(x)
```

Arguments

x	A fitted object of class ICA.BinBin, ICA.BinBin.Grid.Full, or ICA.BinBin.Grid.Sample.

Details

All $r(i,j) = P(\Delta T = i|\Delta S = j)$ are derived from π (vector of potential outcomes). Denote by $\boldsymbol{Y}' = (T_0, T_1, S_0, S_1)$ the vector of potential outcomes. The vector $\boldsymbol{Y}$ can take 16 values and the set of parameters $\pi_{ijpq} = P(T_0 = i, T_1 = j, S_0 = p, S_1 = q)$ (with $i, j, p, q = 0/1$) fully characterizes its distribution.

Based on the data and assuming SUTVA, the marginal probabilites $\pi_{1\cdot1\cdot}$, $\pi_{1\cdot0\cdot}$, $\pi_{\cdot1\cdot1}$, $\pi_{\cdot1\cdot0}$, $\pi_{0\cdot1\cdot}$, and $\pi_{\cdot0\cdot1}$ can be computed (by hand or using the function `MarginalProbs`). Define the vector

$$\boldsymbol{b}' = (1, \pi_{1\cdot1\cdot}, \pi_{1\cdot0\cdot}, \pi_{\cdot1\cdot1}, \pi_{\cdot1\cdot0}, \pi_{0\cdot1\cdot}, \pi_{\cdot0\cdot1})$$

and $\boldsymbol{A}$ is a contrast matrix such that the identified restrictions can be written as a system of linear equation

$$\boldsymbol{A\pi} = \boldsymbol{b}.$$

The matrix $\boldsymbol{A}$ has rank 7 and can be partitioned as $\boldsymbol{A} = (\boldsymbol{A_r}|\boldsymbol{A_f})$, and similarly the vector $\boldsymbol{\pi}$ can be partitioned as $\boldsymbol{\pi}' = (\boldsymbol{\pi_r'}|\boldsymbol{\pi_f'})$ (where f refers to the submatrix/vector given by the 9 last columns/components of $\boldsymbol{A}/\boldsymbol{\pi}$). Using these partitions the previous system of linear equations can be rewritten as

$$\boldsymbol{A_r\pi_r} + \boldsymbol{A_f\pi_f} = \boldsymbol{b}.$$

The functions `ICA.BinBin`, `ICA.BinBin.Grid.Sample`, and `ICA.BinBin.Grid.Full` contain algorithms that generate plausible distributions for $\boldsymbol{Y}$ (for details, see the documentation of these functions). Based on the output of these functions, `SPF.BinBin` computes the surrogate predictive function.

Value

`r_1_1`	The vector of values for $r(1,1)$, i.e., $P(\Delta T = 1|\Delta S = 1)$.
`r_min1_1`	The vector of values for $r(-1,1)$.
`r_0_1`	The vector of values for $r(0,1)$.
`r_1_0`	The vector of values for $r(1,0)$.
`r_min1_0`	The vector of values for $r(-1,0)$.
`r_0_0`	The vector of values for $r(0,0)$.
`r_1_min1`	The vector of values for $r(1,-1)$.
`r_min1_min1`	The vector of values for $r(-1,-1)$.
`r_0_min1`	The vector of values for $r(0,-1)$.
`Monotonicity`	The assumption regarding monotonicity under which the result was obtained.

Author(s)

Wim Van der Elst, Paul Meyvisch, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., & Molenberghs, G. (2015). Assessing a surrogate effect predictive value in a causal inference framework.

See Also

ICA.BinBin, ICA.BinBin.Grid.Sample, ICA.BinBin.Grid.Full, plot.SPF.BinBin

Examples

```
# Use ICA.BinBin.Grid.Sample to obtain plausible values for pi
ICA_BINBIN_Grid_Sample <- ICA.BinBin.Grid.Sample(pi1_1_=0.341, pi0_1_=0.119,
pi1_0_=0.254, pi_1_1=0.686, pi_1_0=0.088, pi_0_1=0.078, Seed=1,
Monotonicity=c("General"), M=2500)

# Obtain SPF
SPF <- SPF.BinBin(ICA_BINBIN_Grid_Sample)

# examine results
summary(SPF)
plot(SPF)
```

SurvSurv *Assess surrogacy for two survival endpoints based on information theory and a two-stage approach*

Description

The function SurvSurv implements the information-theoretic approach to estimate individual-level surrogacy (i.e., $R^2_{h.ind}$) and the two-stage approach to estimate trial-level surrogacy (R^2_{trial}, R^2_{ht}) when both endpoints are time-to-event variables (Alonso & Molenberghs, 2008). See the **Details** section below.

Usage

```
SurvSurv(Dataset, Surr, SurrCens, True, TrueCens, Treat,
Trial.ID, Weighted=TRUE, Alpha=.05)
```

Arguments

Dataset	A data.frame that should consist of one line per patient. Each line contains (at least) a surrogate value and censoring indicator, a true endpoint value and censoring indicator, a treatment indicator, and a trial ID.
Surr	The name of the variable in Dataset that contains the surrogate endpoint values.
SurrCens	The name of the variable in Dataset that contains the censoring indicator for the surrogate endpoint values (1 = event, 0 = censored).
True	The name of the variable in Dataset that contains the true endpoint values.
TrueCens	The name of the variable in Dataset that contains the censoring indicator for the true endpoint values (1 = event, 0 = censored).
Treat	The name of the variable in Dataset that contains the treatment indicators.

`Trial.ID` The name of the variable in `Dataset` that contains the trial ID to which the patient belongs.

`Weighted` Logical. If `TRUE`, then a weighted regression analysis is conducted at stage 2 of the two-stage approach. If `FALSE`, then an unweighted regression analysis is conducted at stage 2 of the two-stage approach. See the **Details** section below. Default `TRUE`.

`Alpha` The α-level that is used to determine the confidence intervals around R^2_{trial} and R_{trial}. Default 0.05.

Details

Individual-level surrogacy

Alonso & Molenbergs (2008) proposed to redefine the surrogate endpoint S as a time-dependent covariate $S(t)$, taking value 0 until the surrogate endpoint occurs and 1 thereafter. Furthermore, these author considered the models

$$\lambda[t \mid x_{ij}, \beta] = K_{ij}(t)\lambda_{0i}(t)exp(\beta x_{ij}),$$

$$\lambda[t \mid x_{ij}, s_{ij}, \beta, \phi] = K_{ij}(t)\lambda_{0i}(t)exp(\beta x_{ij} + \phi S_{ij}),$$

where $K_{ij}(t)$ is the risk function for patient j in trial i, x_{ij} is a p-dimensional vector of (possibly) time-dependent covariates, β is a p-dimensional vector of unknown coefficients, $\lambda_{0i}(t)$ is a trial-specific baseline hazard function, S_{ij} is a time-dependent covariate version of the surrogate endpoint, and ϕ its associated effect.

The mutual information between S and T is estimated as $I(T, S) = 1G^2$, where n is the number of patients and G^2 is the log likelihood test comparing the previous two models. Individual-level surrogacy can then be estimated as

$$R^2_{h.ind} = 1 - exp\left(-\frac{1}{n}G^2\right).$$

O'Quigley and Flandre (2006) pointed out that the previous estimator depends upon the censoring mechanism, even when the censoring mechanism is non-informative. For low levels of censoring this may not be an issue of much concern but for high levels it could lead to biased results. To properly cope with the censoring mechanism in time-to-event outcomes, these authors proposed to estimate the mutual information as $I(T, S) = \frac{1}{k}G^2$, where k is the total number of events experienced. Individual-level surrogacy is then estimated as

$$R^2_{h.ind} = 1 - exp\left(-\frac{1}{k}G^2\right).$$

Trial-level surrogacy

A two-stage approach is used to estimate trial-level surrogacy, following a procedure proposed by Buyse et al. (2011). In stage 1, the following trial-specific Cox proportional hazard models are fitted:

$$S_{ij}(t) = S_{i0}(t)exp(\alpha_i Z_{ij}),$$

$$T_{ij}(t) = T_{i0}(t)exp(\beta_i Z_{ij}),$$

where $S_{i0}(t)$ and $T_{i0}(t)$ are the trial-specific baseline hazard functions, Z_{ij} is the treatment indicator for subject j in trial i, and α_i, β_i are the trial-specific treatment effects on S and T, respectively.

Next, the second stage of the analysis is conducted:

$$\widehat{\beta_i} = \lambda_0 + \lambda_1 \widehat{\alpha_i} + \varepsilon_i,$$

where the parameter estimates for β_i and α_i are based on the full model that was fitted in stage 1.

When the argument `Weighted=FALSE` is used in the function call, the model that is fitted in stage 2 is an unweighted linear regression model. When a weighted model is requested (using the argument `Weighted=TRUE` in the function call), the information that is obtained in stage 1 is weighted according to the number of patients in a trial.

The classical coefficient of determination of the fitted stage 2 model provides an estimate of R^2_{trial}.

Value

An object of class `SurvSurv` with components,

`Results.Stage.1`
: The results of stage 1 of the two-stage model fitting approach: a `data.frame` that contains the trial-specific log hazard ratio estimates of the treatment effects for the surrogate and the true endpoints.

`Results.Stage.2`
: An object of class `lm` (linear model) that contains the parameter estimates of the regression model that is fitted in stage 2 of the analysis.

`R2.ht`
: A `data.frame` that contains the trial-level coefficient of determination (R^2_{ht}), its standard error and confidence interval.

`R2.hind`
: A `data.frame` that contains the individual-level coefficient of determination (R^2_{hind}), its standard error and confidence interval.

`R2h.ind.QF`
: A `data.frame` that contains the individual-level coefficient of determination using the correction proposed by O'Quigley and Flandre (2006), its standard error and confidence interval.

`R2.hInd.By.Trial.QF`
: A `data.frame` that contains individual-level surrogacy estimates using the correction proposed by O'Quigley and Flandre (2006), (cluster-based estimates) and their confidence interval for each of the trials seperately.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A. A., & Molenberghs, G. (2008). Evaluating time-to-cancer recurrence as a surrogate marker for survival from an information theory perspective. *Statistical Methods in Medical Research, 17,* 497-504.

Buyse, M., Michiels, S., Squifflet, P., Lucchesi, K. J., Hellstrand, K., Brune, M. L., Castaigne, S., Rowe, J. M. (2011). Leukemia-free survival as a surrogate end point for overall survival in the evaluation of maintenance therapy for patients with acute myeloid leukemia in complete remission. *Haematologica, 96,* 1106-1112.

O'Quigly, J., & Flandre, P. (2006). Quantification of the Prentice criteria for surrogate endpoints. *Biometrics, 62,* 297-300.

See Also

plot.SurvSurv

Examples

```
# Open Ovarian dataset
data(Ovarian)

# Conduct analysis
Fit <- SurvSurv(Dataset = Ovarian, Surr = Pfs, SurrCens = PfsInd,
True = Surv, TrueCens = SurvInd, Treat = Treat,
Trial.ID = Center)

# Examine results
plot(Fit)
summary(Fit)
```

Test.Mono *Test whether the data are compatible with monotonicity for S and/or T (binary endpoints)*

Description

For some situations, the observable marginal probabilities contain sufficient information to exclude a particular monotonicity scenario. For example, under monotonicity for S and T, one of the restrictions that the data impose is $\pi_{0111} < min(\pi_{0\cdot 1\cdot}, \pi_{\cdot 1\cdot 1})$. If the latter condition does not hold in the dataset at hand, monotonicity for S and T can be excluded.

Usage

```
Test.Mono(pi1_1_, pi0_1_, pi1_0_, pi_1_1, pi_1_0, pi_0_1)
```

Arguments

pi1_1_	A scalar that contains $P(T = 1, S = 1 \mid Z = 0)$.
pi0_1_	A scalar that contains $P(T = 0, S = 1 \mid Z = 0)$.
pi1_0_	A scalar that contains $P(T = 1, S = 0 \mid Z = 0)$.
pi_1_1	A scalar that contains $P(T = 1, S = 1 \mid Z = 1)$.
pi_1_0	A scalar that contains $P(T = 1, S = 0 \mid Z = 1)$.
pi_0_1	A scalar that contains $P(T = 0, S = 1 \mid Z = 1)$.

Author(s)

Wim Van der Elst, Ariel Alonso, Marc Buyse, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., & Molenberghs, G. (2015). Validation of surrogate endpoints: the binary-binary setting from a causal inference perspective.

Examples

```
Test.Mono(pi1_1_=0.2619048, pi1_0_=0.2857143, pi_1_1=0.6372549,
pi_1_0=0.07843137, pi0_1_=0.1349206, pi_0_1=0.127451)
```

TrialLevelIT *Estimates trial-level surrogacy in the information-theoretic framework*

Description

The function `TrialLevelIT` estimates trial-level surrogacy based on the vectors of treatment effects on S (i.e., α_i), intercepts on S (i.e., μ_i) and T (i.e., β_i) in the different trials. See the **Details** section below.

Usage

```
TrialLevelIT(Alpha.Vector, Mu_S.Vector=NULL,
Beta.Vector, N.Trial, Model="Reduced", Alpha=.05)
```

Arguments

`Alpha.Vector`	The vector of treatment effects on S in the different trials, i.e., α_i.
`Mu_S.Vector`	The vector of intercepts for S in the different trials, i.e., μ_{Si}. Only required when a full model is requested.
`Beta.Vector`	The vector of treatment effects on T in the different trials, i.e., β_i.
`N.Trial`	The total number of available trials.
`Model`	The type of model that should be fitted, i.e., `Model=c("Full")` or `Model=c("Reduced")`. See the **Details** section below. Default `Model=c("Reduced")`.
`Alpha`	The α-level that is used to determine the confidence intervals around R^2_{trial} and R_{trial}. Default 0.05.

Details

When a full model is requested (by using the argument `Model=c("Full")` in the function call), trial-level surrogacy is assessed by fitting the following univariate model:

$$\beta_i = \lambda_0 + \lambda_1 \mu_{Si} + \lambda_2 \alpha_i + \varepsilon_i, (1)$$

where β_i = the trial-specific treatment effects on T, μ_{Si} = the trial-specific intercepts for S, and α_i = the trial-specific treatment effects on S. The -2 log likelihood value of model (1) (L_1) is subsequently compared to the -2 log likelihood value of an intercept-only model ($\beta_i = \lambda_3$; L_0), and R^2_{ht} is computed based based on the Variance Reduction Factor (for details, see Alonso & Molenberghs, 2007):

$$R^2_{ht} = 1 - exp\left(-\frac{L_1 - L_0}{N}\right),$$

where N is the number of trials.

When a reduced model is requested (by using the argument `Model=c("Reduced")` in the function call), the following model is fitted:

$$\beta_i = \lambda_0 + \lambda_1 \alpha_i + \varepsilon_i.$$

The -2 log likelihood value of this model (L_1 for the reduced model) is subsequently compared to the -2 log likelihood value of an intercept-only model ($\beta_i = \lambda_3$; L_0), and R^2_{ht} is computed based on the reduction in the likelihood (as described above).

Value

An object of class `TrialLevelIT` with components,

`Alpha.Vector`	The vector of treatment effects on S in the different trials.
`Beta.Vector`	The vector of treatment effects on T in the different trials.
`N.Trial`	The total number of trials.
`R2.ht`	A `data.frame` that contains the trial-level coefficient of determination (R^2_{ht}), its standard error and confidence interval.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Burzykowski, T., Molenberghs, G., & Buyse, M. (2005). *The evaluation of surrogate endpoints*. New York: Springer-Verlag.

Buyse, M., Molenberghs, G., Burzykowski, T., Renard, D., & Geys, H. (2000). The validation of surrogate endpoints in meta-analysis of randomized experiments. *Biostatistics, 1,* 49-67.

See Also

UnimixedContCont, UnifixedContCont, BifixedContCont, BimixedContCont, plot.TrialLevelIT

Examples

```
# Generate vector treatment effects on S
set.seed(seed = 1)
Alpha.Vector <- seq(from = 5, to = 10, by=.1) + runif(min = -.5, max = .5, n = 51)

# Generate vector treatment effects on T
set.seed(seed=2)
Beta.Vector <- (Alpha.Vector * 3) + runif(min = -5, max = 5, n = 51)

# Apply the function to estimate R^2_{h.t}
Fit <- TrialLevelIT(Alpha.Vector=Alpha.Vector,
Beta.Vector=Beta.Vector, N.Trial=50, Model="Reduced")

summary(Fit)
plot(Fit)
```

TrialLevelMA *Estimates trial-level surrogacy in the meta-analytic framework*

Description

The function TrialLevelMA estimates trial-level surrogacy based on the vectors of treatment effects on S (i.e., α_i) and T (i.e., β_i) in the different trials. In particular, β_i is regressed on α_i and the classical coefficient of determination of the fitted model provides an estimate of R^2_{trial}. In addition, the standard error and CI are provided.

Usage

```
TrialLevelMA(Alpha.Vector, Beta.Vector,
N.Vector, Weighted=TRUE, Alpha=.05)
```

Arguments

Alpha.Vector	The vector of treatment effects on S in the different trials, i.e., α_i.
Beta.Vector	The vector of treatment effects on T in the different trials, i.e., β_i.
N.Vector	The vector of trial sizes N_i.
Weighted	Logical. If TRUE, then a weighted regression analysis is conducted. If FALSE, then an unweighted regression analysis is conducted. Default TRUE.
Alpha	The α-level that is used to determine the confidence intervals around R^2_{trial} and R_{trial}. Default 0.05.

Value

An object of class `TrialLevelMA` with components,

`Alpha.Vector`	The vector of treatment effects on S in the different trials.
`Beta.Vector`	The vector of treatment effects on T in the different trials.
`N.Vector`	The vector of trial sizes N_i.
`Trial.R2`	A `data.frame` that contains the trial-level coefficient of determination (R^2_{trial}), its standard error and confidence interval.
`Trial.R`	A `data.frame` that contains the trial-level correlation coefficient (R_{trial}), its standard error and confidence interval.
`Model.2.Fit`	The fitted stage 2 model.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Burzykowski, T., Molenberghs, G., & Buyse, M. (2005). *The evaluation of surrogate endpoints*. New York: Springer-Verlag.

Buyse, M., Molenberghs, G., Burzykowski, T., Renard, D., & Geys, H. (2000). The validation of surrogate endpoints in meta-analysis of randomized experiments. *Biostatistics, 1,* 49-67.

See Also

`UnimixedContCont`, `UnifixedContCont`, `BifixedContCont`, `BimixedContCont`, `plot Meta-Analytic`

Examples

```
# Generate vector treatment effects on S
set.seed(seed = 1)
Alpha.Vector <- seq(from = 5, to = 10, by=.1) + runif(min = -.5, max = .5, n = 51)
# Generate vector treatment effects on T
set.seed(seed=2)
Beta.Vector <- (Alpha.Vector * 3) + runif(min = -5, max = 5, n = 51)
# Vector of sample sizes of the trials (here, all n_i=10)
N.Vector <- rep(10, times=51)

# Apply the function to estimate R^2_{trial}
Fit <- TrialLevelMA(Alpha.Vector=Alpha.Vector,
Beta.Vector=Beta.Vector, N.Vector=N.Vector)

# Plot the results and obtain summary
plot(Fit)
summary(Fit)
```

`TwoStageSurvSurv` *Assess trial-level surrogacy for two survival endpoints using a two-stage approach*

Description

The function `TwoStageSurvSurv` uses a two-stage approach to estimate R^2_{trial}. In stage 1, trial-specific Cox proportional hazard models are fitted and in stage 2 the trial-specific estimated treatment effects on T are regressed on the trial-specific estimated treatment effects on S (measured on the log hazard ratio scale). The user can specify whether a weighted or unweighted model should be fitted at stage 2. See the **Details** section below.

Usage

```
TwoStageSurvSurv(Dataset, Surr, SurrCens, True, TrueCens, Treat,
Trial.ID, Weighted=TRUE, Alpha=.05)
```

Arguments

`Dataset`	A `data.frame` that should consist of one line per patient. Each line contains (at least) a surrogate value and censoring indicator, a true endpoint value and censoring indicator, a treatment indicator, and a trial ID.
`Surr`	The name of the variable in `Dataset` that contains the surrogate endpoint values.
`SurrCens`	The name of the variable in `Dataset` that contains the censoring indicator for the surrogate endpoint values (1 = event, 0 = censored).
`True`	The name of the variable in `Dataset` that contains the true endpoint values.
`TrueCens`	The name of the variable in `Dataset` that contains the censoring indicator for the true endpoint values (1 = event, 0 = censored).
`Treat`	The name of the variable in `Dataset` that contains the treatment indicators.
`Trial.ID`	The name of the variable in `Dataset` that contains the trial ID to which the patient belongs.
`Weighted`	Logical. If `TRUE`, then a weighted regression analysis is conducted at stage 2 of the two-stage approach. If `FALSE`, then an unweighted regression analysis is conducted at stage 2 of the two-stage approach. See the **Details** section below. Default `TRUE`.
`Alpha`	The α-level that is used to determine the confidence intervals around R^2_{trial} and R_{trial}. Default 0.05.

Details

A two-stage approach is used to estimate trial-level surrogacy, following a procedure proposed by Buyse et al. (2011). In stage 1, the following trial-specific Cox proportional hazard models are fitted:

$$S_{ij}(t) = S_{i0}(t)exp(\alpha_i Z_{ij}),$$

$$T_{ij}(t) = T_{i0}(t)exp(\beta_i Z_{ij}),$$

where $S_{i0}(t)$ and $T_{i0}(t)$ are the trial-specific baseline hazard functions, Z_{ij} is the treatment indicator for subject j in trial i, μ_{Si}, and α_i and β_i are the trial-specific treatment effects on S and T, respectively.

Next, the second stage of the analysis is conducted:

$$\widehat{\beta}_i = \lambda_0 + \lambda_1 \widehat{\alpha}_i + \varepsilon_i,$$

where the parameter estimates for β_i, μ_{Si}, and α_i are based on the full model that was fitted in stage 1.

When the argument `Weighted=FALSE` is used in the function call, the model that is fitted in stage 2 is an unweighted linear regression model. When a weighted model is requested (using the argument `Weighted=TRUE` in the function call), the information that is obtained in stage 1 is weighted according to the number of patients in a trial.

The classical coefficient of determination of the fitted stage 2 model provides an estimate of R^2_{trial}.

Value

An object of class `TwoStageSurvSurv` with components,

`Data.Analyze` Prior to conducting the surrogacy analysis, data of trials that do not have at least three patients per treatment arm are excluded due to estimation constraints (Burzykowski et al., 2001). `Data.Analyze` is the dataset on which the surrogacy analysis was conducted.

`Results.Stage.1`
The results of stage 1 of the two-stage model fitting approach: a `data.frame` that contains the trial-specific log hazard ratio estimates of the treatment effects for the surrogate and the true endpoints.

`Results.Stage.2`
An object of class `lm` (linear model) that contains the parameter estimates of the regression model that is fitted in stage 2 of the analysis.

`Trial.R2` A `data.frame` that contains the trial-level coefficient of determination (R^2_{trial}), its standard error and confidence interval.

`Trial.R` A `data.frame` that contains the trial-level correlation coefficient (R_{trial}), its standard error and confidence interval.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Burzykowski, T., Molenberghs, G., Buyse, M., Geys, H., & Renard, D. (2001). Validation of surrogate endpoints in multiple randomized clinical trials with failure-time endpoints. *Applied Statistics, 50,* 405-422.

Buyse, M., Michiels, S., Squifflet, P., Lucchesi, K. J., Hellstrand, K., Brune, M. L., Castaigne, S., Rowe, J. M. (2011). Leukemia-free survival as a surrogate end point for overall survival in the evaluation of maintenance therapy for patients with acute myeloid leukemia in complete remission. *Haematologica, 96,* 1106-1112.

See Also

plot.TwoStageSurvSurv

Examples

```
# Open Ovarian dataset
data(Ovarian)

# Conduct analysis
Results <- TwoStageSurvSurv(Dataset = Ovarian, Surr = Pfs, SurrCens = PfsInd,
True = Surv, TrueCens = SurvInd, Treat = Treat, Trial.ID = Center)

# Examine results of analysis
summary(Results)
plot(Results)
```

UnifixedContCont *Fits univariate fixed-effect models to assess surrogacy in the meta-analytic multiple-trial setting (continuous-continuous case)*

Description

The function `UnifixedContCont` uses the univariate fixed-effects approach to estimate trial- and individual-level surrogacy when the data of multiple clinical trials are available. The user can specify whether a (weighted or unweighted) full, semi-reduced, or reduced model should be fitted. See the **Details** section below.

Usage

```
UnifixedContCont(Dataset, Surr, True, Treat, Trial.ID, Pat.ID, Model=c("Full"),
Weighted=TRUE, Min.Trial.Size=2, Alpha=.05, Number.Bootstraps=500,
Seed=sample(1:1000, size=1))
```

Arguments

`Dataset`	A `data.frame` that should consist of one line per patient. Each line contains (at least) a surrogate value, a true endpoint value, a treatment indicator, a patient ID, and a trial ID.
`Surr`	The name of the variable in `Dataset` that contains the surrogate endpoint values.
`True`	The name of the variable in `Dataset` that contains the true endpoint values.

`Treat` The name of the variable in `Dataset` that contains the treatment indicators. The treatment indicator should either be coded as 1 for the experimental group and -1 for the control group, or as 1 for the experimental group and 0 for the control group.

`Trial.ID` The name of the variable in `Dataset` that contains the trial ID to which the patient belongs.

`Pat.ID` The name of the variable in `Dataset` that contains the patient's ID.

`Model` The type of model that should be fitted, i.e., `Model=c("Full")`, `Model=c("Reduced")`, or `Model=c("SemiReduced")`. See the **Details** section below. Default `Model=c("Full")`.

`Weighted` Logical. If `TRUE`, then a weighted regression analysis is conducted at stage 2 of the two-stage approach. If `FALSE`, then an unweighted regression analysis is conducted at stage 2 of the two-stage approach. See the **Details** section below. Default `TRUE`.

`Min.Trial.Size` The minimum number of patients that a trial should contain to be included in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded from the analysis. Default 2.

`Alpha` The α-level that is used to determine the confidence intervals around R^2_{trial}, R_{trial}, R^2_{indiv}, and R_{indiv}. Default 0.05.

`Number.Bootstraps`
The standard errors and confidence intervals for R^2_{indiv} and R_{indiv} are determined as based on a bootstrap procedure. `Number.Bootstraps` specifies the number of bootstrap samples that are used. Default 500.

`Seed` The seed to be used in the bootstrap procedure. Default $sample(1:1000, size = 1)$.

Details

When the full bivariate mixed-effects model is fitted to assess surrogacy in the meta-analytic framework (for details, Buyse & Molenberghs, 2000), computational issues often occur. In that situation, the use of simplified model-fitting strategies may be warranted (for details, see see Burzykowski et al., 2005; Tibaldi et al., 2003).

The function `UnifixedContCont` implements one such strategy, i.e., it uses a two-stage univariate fixed-effects modelling approach to assess surrogacy. In the first stage of the analysis, two univariate linear regression models are fitted to the data of each of the i trials. When a full or semi-reduced model is requested (by using the argument `Model=c("Full")` or `Model=c("SemiReduced")` in the function call), the following univariate models are fitted:

$$S_{ij} = \mu_{Si} + \alpha_i Z_{ij} + \varepsilon_{Sij},$$

$$T_{ij} = \mu_{Ti} + \beta_i Z_{ij} + \varepsilon_{Tij},$$

where i and j are the trial and subject indicators, S_{ij} and T_{ij} are the surrogate and true endpoint values of subject j in trial i, Z_{ij} is the treatment indicator for subject j in trial i, μ_{Si} and μ_{Ti} are the fixed trial-specific intercepts for S and T, and α_i and β_i are the fixed trial-specific treatment effects on S and T, respectively. The error terms ε_{Sij} and ε_{Tij} are assumed to be independent.

When a reduced model is requested by the user (by using the argument `Model=c("Reduced")` in the function call), the following univariate models are fitted:

$$S_{ij} = \mu_S + \alpha_i Z_{ij} + \varepsilon_{Sij},$$

$$T_{ij} = \mu_T + \beta_i Z_{ij} + \varepsilon_{Tij},$$

where μ_S and μ_T are the common intercepts for S and T (i.e., it is assumed that the intercepts for the surrogate and the true endpoints are identical in each of the trials). The other parameters are the same as defined above, and ε_{Sij} and ε_{Tij} are again assumed to be independent.

An estimate of R^2_{indiv} is provided by $r(\varepsilon_{Sij}, \varepsilon_{Tij})^2$.

Next, the second stage of the analysis is conducted. When a full model is requested (by using the argument `Model=c("Full")` in the function call), the following model is fitted:

$$\widehat{\beta}_i = \lambda_0 + \lambda_1 \widehat{\mu_{Si}} + \lambda_2 \widehat{\alpha}_i + \varepsilon_i,$$

where the parameter estimates for β_i, μ_{Si}, and α_i are based on the full models that were fitted in stage 1.

When a semi-reduced or reduced model is requested (by using the argument `Model=c("SemiReduced")` or `Model=c("Reduced")` in the function call), the following model is fitted:

$$\widehat{\beta}_i = \lambda_0 + \lambda_1 \widehat{\alpha}_i + \varepsilon_i.$$

where the parameter estimates for β_i and α_i are based on the semi-reduced or reduced models that were fitted in stage 1.

When the argument `Weighted=FALSE` is used in the function call, the model that is fitted in stage 2 is an unweighted linear regression model. When a weighted model is requested (using the argument `Weighted=TRUE` in the function call), the information that is obtained in stage 1 is weighted according to the number of patients in a trial.

The classical coefficient of determination of the fitted stage 2 model provides an estimate of R^2_{trial}.

Value

An object of class `UnifixedContCont` with components,

`Data.Analyze` Prior to conducting the surrogacy analysis, data of patients who have a missing value for the surrogate and/or the true endpoint are excluded. In addition, the data of trials (i) in which only one type of the treatment was administered, and (ii) in which either the surrogate or the true endpoint was a constant (i.e., all patients within a trial had the same surrogate and/or true endpoint value) are excluded. In addition, the user can specify the minimum number of patients that a trial should contain in order to include the trial in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded. `Data.Analyze` is the dataset on which the surrogacy analysis was conducted.

`Obs.Per.Trial` A `data.frame` that contains the total number of patients per trial and the number of patients who were administered the control treatment and the experimental treatment in each of the trials (in `Data.Analyze`).

`Results.Stage.1`
: The results of stage 1 of the two-stage model fitting approach: a `data.frame` that contains the trial-specific intercepts and treatment effects for the surrogate and the true endpoints (when a full or semi-reduced model is requested), or the trial-specific treatment effects for the surrogate and the true endpoints (when a reduced model is requested).

`Residuals.Stage.1`
: A `data.frame` that contains the residuals for the surrogate and true endpoints that are obtained in stage 1 of the analysis (ε_{Sij} and ε_{Tij}).

`Results.Stage.2`
: An object of class `lm` (linear model) that contains the parameter estimates of the regression model that is fitted in stage 2 of the analysis.

`Trial.R2`
: A `data.frame` that contains the trial-level coefficient of determination (R^2_{trial}), its standard error and confidence interval.

`Indiv.R2`
: A `data.frame` that contains the individual-level coefficient of determination (R^2_{indiv}), its standard error and confidence interval.

`Trial.R`
: A `data.frame` that contains the trial-level correlation coefficient (R_{trial}), its standard error and confidence interval.

`Indiv.R`
: A `data.frame` that contains the individual-level correlation coefficient (R_{indiv}), its standard error and confidence interval.

`Cor.Endpoints`
: A `data.frame` that contains the correlations between the surrogate and the true endpoint in the control treatment group (i.e., ρ_{T0S0}) and in the experimental treatment group (i.e., ρ_{T1S1}), their standard errors and their confidence intervals.

`D.Equiv`
: The variance-covariance matrix of the trial-specific intercept and treatment effects for the surrogate and true endpoints (when a full or semi-reduced model is fitted, i.e., when `Model=c("Full")` or `Model=c("SemiReduced")` is used in the function call), or the variance-covariance matrix of the trial-specific treatment effects for the surrogate and true endpoints (when a reduced model is fitted, i.e., when `Model=c("Reduced")` is used in the function call). The variance-covariance matrix `D.Equiv` is equivalent to the $\boldsymbol{D}$ matrix that would be obtained when a (full or reduced) bivariate mixed-effect approach is used; see function `BimixedContCont`).

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Burzykowski, T., Molenberghs, G., & Buyse, M. (2005). *The evaluation of surrogate endpoints*. New York: Springer-Verlag.

Buyse, M., Molenberghs, G., Burzykowski, T., Renard, D., & Geys, H. (2000). The validation of surrogate endpoints in meta-analysis of randomized experiments. *Biostatistics, 1,* 49-67.

Tibaldi, F., Abrahantes, J. C., Molenberghs, G., Renard, D., Burzykowski, T., Buyse, M., Parmar, M., et al., (2003). Simplified hierarchical linear models for the evaluation of surrogate endpoints. *Journal of Statistical Computation and Simulation, 73,* 643-658.

See Also

UnimixedContCont, BifixedContCont, BimixedContCont, plot Meta-Analytic

Examples

```
## Not run:  #Time consuming (>5 sec) code parts
# Example 1, based on the ARMD data
data(ARMD)

# Fit a full univariate fixed-effects model with weighting according to the
# number of patients in stage 2 of the two stage approach to assess surrogacy:
Sur <- UnifixedContCont(Dataset=ARMD, Surr=Diff24, True=Diff52, Treat=Treat, Trial.ID=Center,
Pat.ID=Id, Model="Full", Weighted=TRUE)

# Obtain a summary and plot of the results
summary(Sur)
plot(Sur)

# Example 2
# Conduct an analysis based on a simulated dataset with 2000 patients, 100 trials,
# and Rindiv=Rtrial=.8
# Simulate the data:
Sim.Data.MTS(N.Total=2000, N.Trial=100, R.Trial.Target=.8, R.Indiv.Target=.8,
Seed=123, Model="Reduced")

# Fit a reduced univariate fixed-effects model without weighting to assess
# surrogacy:
Sur2 <- UnifixedContCont(Dataset=Data.Observed.MTS, Surr=Surr, True=True, Treat=Treat,
Trial.ID=Trial.ID, Pat.ID=Pat.ID, Model="Reduced", Weighted=FALSE)

# Show a summary and plots of results:
summary(Sur2)
plot(Sur2, Weighted=FALSE)
## End(Not run)
```

`UnimixedContCont` *Fits univariate mixed-effect models to assess surrogacy in the meta-analytic multiple-trial setting (continuous-continuous case)*

Description

The function `UnimixedContCont` uses the univariate mixed-effects approach to estimate trial- and individual-level surrogacy when the data of multiple clinical trials are available. The user can specify whether a (weighted or unweighted) full, semi-reduced, or reduced model should be fitted. See the **Details** section below.

Usage

```
UnimixedContCont(Dataset, Surr, True, Treat, Trial.ID, Pat.ID, Model=c("Full"),
Weighted=TRUE, Min.Trial.Size=2, Alpha=.05, Number.Bootstraps=500,
Seed=sample(1:1000, size=1), ...)
```

Arguments

`Dataset`	A `data.frame` that should consist of one line per patient. Each line contains (at least) a surrogate value, a true endpoint value, a treatment indicator, a patient ID, and a trial ID.
`Surr`	The name of the variable in `Dataset` that contains the surrogate endpoint values.
`True`	The name of the variable in `Dataset` that contains the true endpoint values.
`Treat`	The name of the variable in `Dataset` that contains the treatment indicators. The treatment indicator should either be coded as 1 for the experimental group and -1 for the control group, or as 1 for the experimental group and 0 for the control group.
`Trial.ID`	The name of the variable in `Dataset` that contains the trial ID to which the patient belongs.
`Pat.ID`	The name of the variable in `Dataset` that contains the patient's ID.
`Model`	The type of model that should be fitted, i.e., `Model=c("Full")`, `Model=c("Reduced")`, or `Model=c("SemiReduced")`. See the **Details** section below. Default `Model=c("Full")`.
`Weighted`	Logical. If `TRUE`, then a weighted regression analysis is conducted at stage 2 of the two-stage approach. If `FALSE`, then an unweighted regression analysis is conducted at stage 2 of the two-stage approach. See the **Details** section below. Default `TRUE`.
`Min.Trial.Size`	The minimum number of patients that a trial should contain to be included in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded from the analysis. Default 2.
`Alpha`	The α-level that is used to determine the confidence intervals around R^2_{trial}, R_{trial}, R^2_{indiv}, and R_{indiv}. Default 0.05.
`Number.Bootstraps`	The confidence intervals for R^2_{indiv} and R_{indiv} are determined as based on a bootstrap procedure. `Number.Bootstraps` specifies the number of bootstrap samples that are to be used. Default 500.
`Seed`	The seed to be used in the bootstrap procedure. Default $sample(1:1000, size = 1)$.
`...`	Other arguments to be passed to the function `lmer` (of the R package `lme4`) that is used to fit the geralized linear mixed-effect models in the function `BimixedContCont`. For details, see the lme4 manual.

Details

When the full bivariate mixed-effects model is fitted to assess surrogacy in the meta-analytic framework (for details, Buyse & Molenberghs, 2000), computational issues often occur. In that situation,

the use of simplified model-fitting strategies may be warranted (for details, see Burzykowski et al., 2005; Tibaldi et al., 2003).

The function `UnimixedContCont` implements one such strategy, i.e., it uses a two-stage univariate mixed-effects modelling approach to assess surrogacy. In the first stage of the analysis, two univariate mixed-effects models are fitted to the data. When a full or semi-reduced model is requested (by using the argument `Model=c("Full")` or `Model=c("SemiReduced")` in the function call), the following univariate models are fitted:

$$S_{ij} = \mu_S + m_{Si} + (\alpha + a_i)Z_{ij} + \varepsilon_{Sij},$$

$$T_{ij} = \mu_T + m_{Ti} + (\beta + b_i)Z_{ij} + \varepsilon_{Tij},$$

where i and j are the trial and subject indicators, S_{ij} and T_{ij} are the surrogate and true endpoint values of subject j in trial i, Z_{ij} is the treatment indicator for subject j in trial i, μ_S and μ_T are the fixed intercepts for S and T, m_{Si} and m_{Ti} are the corresponding random intercepts, α and β are the fixed treatment effects for S and T, and a_i and b_i are the corresponding random treatment effects, respectively. The error terms ε_{Sij} and ε_{Tij} are assumed to be independent.

When a reduced model is requested (by using the argument `Model=c("Reduced")` in the function call), the following two univariate models are fitted:

$$S_{ij} = \mu_S + (\alpha + a_i)Z_{ij} + \varepsilon_{Sij},$$

$$T_{ij} = \mu_T + (\beta + b_i)Z_{ij} + \varepsilon_{Tij},$$

where μ_S and μ_T are the common intercepts for S and T (i.e., it is assumed that the intercepts for the surrogate and the true endpoints are identical in each of the trials). The other parameters are the same as defined above, and ε_{Sij} and ε_{Tij} are again assumed to be independent.

An estimate of R^2_{indiv} is computed as $r(\varepsilon_{Sij}, \varepsilon_{Tij})^2$.

Next, the second stage of the analysis is conducted. When a full model is requested by the user (by using the argument `Model=c("Full")` in the function call), the following model is fitted:

$$\widehat{\beta}_i = \lambda_0 + \lambda_1 \widehat{\mu_{Si}} + \lambda_2 \widehat{\alpha}_i + \varepsilon_i,$$

where the parameter estimates for β_i, μ_{Si}, and α_i are based on the models that were fitted in stage 1, i.e., $\beta_i = \beta + b_i$, $\mu_{Si} = \mu_S + m_{Si}$, and $\alpha_i = \alpha + a_i$.

When a reduced or semi-reduced model is requested by the user (by using the arguments `Model=c("SemiReduced")` or `Model=c("Reduced")` in the function call), the following model is fitted:

$$\widehat{\beta}_i = \lambda_0 + \lambda_1 \widehat{\alpha}_i + \varepsilon_i,$$

where the parameters are the same as defined above.

When the argument `Weighted=FALSE` is used in the function call, the model that is fitted in stage 2 is an unweighted linear regression model. When a weighted model is requested (using the argument `Weighted=TRUE` in the function call), the information that is obtained in stage 1 is weighted according to the number of patients in a trial.

The classical coefficient of determination of the fitted stage 2 model provides an estimate of R^2_{trial}.

Value

An object of class `UnimixedContCont` with components,

`Data.Analyze` Prior to conducting the surrogacy analysis, data of patients who have a missing value for the surrogate and/or the true endpoint are excluded. In addition, the data of trials (i) in which only one type of the treatment was administered, and (ii) in which either the surrogate or the true endpoint was a constant (i.e., all patients within a trial had the same surrogate and/or true endpoint value) are excluded. In addition, the user can specify the minimum number of patients that a trial should contain in order to include the trial in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded. `Data.Analyze` is the dataset on which the surrogacy analysis was conducted.

`Obs.Per.Trial` A `data.frame` that contains the total number of patients per trial and the number of patients who were administered the control treatment and the experimental treatment in each of the trials (in `Data.Analyze`).

`Results.Stage.1`
The results of stage 1 of the two-stage model fitting approach: a `data.frame` that contains the trial-specific intercepts and treatment effects for the surrogate and the true endpoints (when a full or semi-reduced model is requested), or the trial-specific treatment effects for the surrogate and the true endpoints (when a reduced model is requested).

`Residuals.Stage.1`
A `data.frame` that contains the residuals for the surrogate and true endpoints that are obtained in stage 1 of the analysis (ε_{Sij} and ε_{Tij}).

`Fixed.Effect.Pars`
A `data.frame` that contains the fixed intercept and treatment effects for S and T (i.e., μ_S, μ_T, α, and β) when a full, semi-reduced, or reduced model is fitted in stage 1.

`Random.Effect.Pars`
A `data.frame` that contains the random intercept and treatment effects for S and T (i.e., m_{Si}, m_{Ti}, a_i and b_i) when a full or semi-reduced model is fitted in stage 1, or that contains the random treatment effects for S and T (i.e., a_i, and b_i) when a reduced model is fitted in stage 1.

`Results.Stage.2`
An object of class `lm` (linear model) that contains the parameter estimates of the regression model that is fitted in stage 2 of the analysis.

`Trial.R2` A `data.frame` that contains the trial-level coefficient of determination (R^2_{trial}), its standard error and confidence interval.

`Indiv.R2` A `data.frame` that contains the individual-level coefficient of determination (R^2_{indiv}), its standard error and confidence interval.

`Trial.R` A `data.frame` that contains the trial-level correlation coefficient (R_{trial}), its standard error and confidence interval.

`Indiv.R` A `data.frame` that contains the individual-level correlation coefficient (R_{indiv}), its standard error and confidence interval.

`Cor.Endpoints` A `data.frame` that contains the correlations between the surrogate and the true endpoint in the control treatment group (i.e., ρ_{T0S0}) and in the experimental treatment group (i.e., ρ_{T1S1}), their standard errors and their confidence intervals.

`D.Equiv` The variance-covariance matrix of the trial-specific intercept and treatment effects for the surrogate and true endpoints (when a full or semi-reduced model is fitted, i.e., when `Model=c("Full")` or `Model=c("SemiReduced")` is used in the function call), or the variance-covariance matrix of the trial-specific treatment effects for the surrogate and true endpoints (when a reduced model is fitted, i.e., when `Model=c("Reduced")` is used in the function call). The variance-covariance matrix `D.Equiv` is equivalent to the $\boldsymbol{D}$ matrix that would be obtained when a (full or reduced) bivariate mixed-effects approach is used; see function `BimixedContCont`).

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Burzykowski, T., Molenberghs, G., & Buyse, M. (2005). *The evaluation of surrogate endpoints*. New York: Springer-Verlag.

Buyse, M., Molenberghs, G., Burzykowski, T., Renard, D., & Geys, H. (2000). The validation of surrogate endpoints in meta-analysis of randomized experiments. *Biostatistics, 1,* 49-67.

Tibaldi, F., Abrahantes, J. C., Molenberghs, G., Renard, D., Burzykowski, T., Buyse, M., Parmar, M., et al., (2003). Simplified hierarchical linear models for the evaluation of surrogate endpoints. *Journal of Statistical Computation and Simulation, 73,* 643-658.

See Also

`UnifixedContCont`, `BifixedContCont`, `BimixedContCont`, `plot Meta-Analytic`

Examples

```
## Not run:  #Time consuming code part
# Conduct an analysis based on a simulated dataset with 2000 patients, 100 trials,
# and Rindiv=Rtrial=.8
# Simulate the data:
Sim.Data.MTS(N.Total=2000, N.Trial=100, R.Trial.Target=.8, R.Indiv.Target=.8,
Seed=123, Model="Reduced")

# Fit a reduced univariate mixed-effects model without weighting to assess surrogacy:
Sur <- UnimixedContCont(Dataset=Data.Observed.MTS, Surr=Surr, True=True, Treat=Treat,
Trial.ID=Trial.ID, Pat.ID=Pat.ID, Model="Reduced", Weighted=FALSE)

# Show a summary and plots of the results:
summary(Sur)
plot(Sur, Weighted=FALSE)
## End(Not run)
```

Index