Package 'iMRMC'

December 12, 2017

Type Package

Title Multi-Reader, Multi-Case Analysis Methods (ROC, Agreement, and Other Metrics)

Version 1.1.0

Author Brandon Gallas

Maintainer Brandon Gallas <Brandon.Gallas@fda.hhs.gov>

Description Do Multi-Reader, Multi-Case (MRMC) analyses of data from imaging studies where clinicians (readers) evaluate patient images (cases). What does this mean? ... Many imaging studies are designed so that every reader reads every case in all modalities, a fully-crossed study. In this case, the data is cross-correlated, and we consider the readers and cases to be cross-correlated random effects.
An MRMC analysis accounts for the variability and correlations from the readers and cases when estimating variances, confidence intervals, and p-values. The functions in this package can treat arbitrary study designs and studies with missing data, not just fully-crossed study designs.
The initial package analyzes the reader-average area under the receiver operating characteristic (ROC) curve with U-statistics according to Gallas, Bandos, Samuelson, and Wagner 2009 <doi:10.1080/03610920802610084>.
Additional functions analyze other endpoints with U-statistics (binary performance and score differences) following the work by Gallas, Pennello, and Myers 2007 <doi:10.1364/JOSAA.24.000B70>.
Package development and documentation is at <https://github.com/DIDSR/iMRMC/tree/master>.

License CC0

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Suggests testthat

NeedsCompilation no

Repository CRAN

Date/Publication 2017-12-12 15:18:25 UTC

R topics documented:

`convertDFtoDesignMatrix` *Convert an MRMC data frame to a design matrix*

Description

Convert an MRMC data frame to a design matrix, dropping readers or cases with no observations

Usage

```
convertDFtoDesignMatrix(dfMRMC)
```

Arguments

`dfMRMC`	An MRMC data frame

Value

A matrix [nCases, nReaders] indicating which scores were reported for each reader and case

convertDFtoScoreMatrix

Convert an MRMC data frame to a score matrix

Description

Convert an MRMC data frame to a score matrix, dropping readers or cases with no observations

Usage

```
convertDFtoScoreMatrix(dfMRMC)
```

Arguments

dfMRMC	An MRMC data frame

Value

A matrix [nCases, nReaders] of the scores each reader reported for each case

createIMRMCdf

Convert a data frame with all needed factors to doIMRMC formatted data frame

Description

Convert a data frame with all needed factors to doIMRMC formatted data frame

Usage

```
createIMRMCdf(dFrame, keyColumns = list(readerID = "readerID", caseID =
  "caseID", modalityID = "modalityID", score = "score", truth = "truth"),
  truePositiveFactor = "cancer")
```

Arguments

dFrame	This data frame includes columns for readerID, caseID, modalityID, score, and truth. These columns are not expected to be named as such and other columns may exist.
keyColumns	This list identifies the column names of the data frame to be used for the analysis. list(readerID = "***", caseID = "***", modalityID = "***", score = "***", truth="***")
truePositiveFactor	The true positive label, such as "cancer" or "1"

Value

output a doIMRMC formatted data frame: rows for truth and rows for data

---

`doIMRMC` *MRMC analysis of the area under the ROC curve*

Description

doIMRMC takes ROC data as a data frame and runs a multi-reader multi-case analysis based on U-statistics as described in the following papers Gallas2006_Acad-Radiol_v13p353 (single-modality), Gallas2008_Neural-Networks_v21p387 (multiple modalities, arbitrary study designs), Gallas2009_Commun-Stat-A-Theor_v38p2586 (framework paper).

Usage

```
doIMRMC(data = NULL, fileName = NULL, workDir = NULL,
  iMRMCjarFullPath = NULL, stdout = FALSE, stderr = FALSE,
  stripDatesForTests = FALSE)
```

Arguments

`data`	This data.frame contains the following variables: readerID [Factor] w/ nR levels "reader1", "reader2", ... caseID [Factor] w/ nC levels "case1", "case2", ... modalityID [Factor] w/ 1 level simRoeMetz.config$modalityID score [num] reader score each row of this data frame corresponds to an observation for every caseID, there must be a row corresponding to the truth observation the readerID for a truth observation is "truth" the modalityID for a truth observation is "truth" the score for a truth observation must be either 0 (signal-absent) or 1 (signal-present)
`fileName`	This character string identifies the location of an iMRMC input file. The input file is identical to data except there is a free text section to start, then a line with "BEGIN DATA:", then the data frame info.
`workDir`	This character string determines the directory where intermediate results are written. If this parameter is not set, the program writes the intermediate results to the directory specified by tempdir() and then deletes them.
`iMRMCjarFullPath`	This character string identifies the location of the iMRMC.jar file this jar file can be downloaded from https://github.com/DIDSR/iMRMC/releases this R program supports version iMRMC-v3p2.jar
`stdout`	where output to 'stdout' or 'stderr' should be sent. Possible values are "", to the R console (the default), NULL or FALSE (discard output), TRUE (capture the output in character vector) or a character string naming a file.
`stderr`	where output to 'stdout' or 'stderr' should be sent. Possible values are "", to the R console (the default), NULL or FALSE (discard output), TRUE (capture the output in a character vector) or a character string naming a file.

`stripDatesForTests`
: Since results include a date and time stamp, these need to be stripped out when doing the package tests. This parameter flags whether or not the dates should be stripped out.

Details

In detail, this procedure reads the name of an input file from the local file system, or takes a data frame and writes it to the local file system formatted for the iMRMC program (found at https://github.com/DIDSR/iMRMC/releases), it executes the iMRMC program which writes the results to the local files system, it reads the analysis results from the local file system, packs the analysis results into a list object, deletes the data and analysis results from the local file system, and returns the list object.

Value

iMRMCoutput [list] the objects of this list are described in detail in the iMRMC documentation which can be found at <http://didsr.github.io/iMRMC/000_iMRMC/userManualHTML/index.htm>

Here is a quick summary: perReader [data.frame] this data frame contains the performance results for each reader. Key variables of this data frame are AUCA, AUCB, AUCAminusAUCB and the corresponding variances, confidence intervals, degrees of freedom and p-values. Ustat [data.frame] this data frame contains the reader-average performance results. The analysis results are based on U-statistics and the papers listed above. Key variables of this data frame are AUCA, AUCB, AUCAminusAUCB and the corresponding variances, confidence intervals, degrees of freedom and p-values. MLEstat [data.frame] this data frame contains the reader-average performance results. The analysis results are based on V-statistics, which approximates the true distribution with the empirical distribution. The empirical distribution equals the nonparametric MLE estimate of the true distribution, which is also equivalent to the ideal bootstrap estimate. Please refer to the papers listed above. Key variables of this data frame are AUCA, AUCB, AUCAminusAUCB and the corresponding variances, confidence intervals, degrees of freedom and p-values. ROC [list] each object of this list is an object containing an ROC curve. There is an ROC curve for every combination of reader and modality. For every modality, there are also four average ROC curves. These are discussed in Chen2014_Br-J-Radiol_v87p20140016 The diagonal average averages the reader-specific ROC curves along y = -x + b for b in (0,1) The horizontal average averages the reader specific ROC curves along y = b for b in (0,1) The vertical average averages the reader specific ROC curves along x = b for b in (0,1) The pooled average ignores readerID and pools all the scores together to create one ROC curve. varDecomp [list] the objects of this list are different decompositions of the total variance Please refer to Gallas2009_Commun-Stat-A-Theor_v38p2586 (framework paper). The different decompositions are BCK, BDG, DBM, MS, OR.

Examples

```
# Create a sample configuration file
config <- sim.gRoeMetz.config()
# Simulate an MRMC ROC data set
dFrame.imrmc <- sim.gRoeMetz(config)
# Analyze the MRMC ROC data
result <- doIMRMC(dFrame.imrmc)
```

`init.lecuyerRNG` *Initialize the l'Ecuyer random number generator*

Description

See the documentation for the parallel package

Usage

```
init.lecuyerRNG(seed = 1, stream = 2)
```

Arguments

`seed`	This determines the position in each stream
`stream`	This determines the stream

Value

Nothing

`roc2binary` *Convert ROC data formatted for doIMRMC to TPF and FPF data formatted for doIMRMC*

Description

Convert ROC data formatted for doIMRMC to TPF and FPF data formatted for doIMRMC

Usage

```
roc2binary(df.auc, threshold)
```

Arguments

`df.auc`	data frame of roc scores formatted for doIMRMC
`threshold`	The threshold for determining binary decisions

Value

a list of two data frames (df.tpf and df.fpf) both formatted for doIMRMC

`sim.gRoeMetz` *Simulate an MRMC data set of an ROC experiment comparing two modalities*

Description

This procedure simulates an MRMC data set of an ROC experiment comparing two modalities. It is based on Gallas2014_J-Med-Img_v1p031006, which generalizes of the model in Roe1997_Acad-Radiol_v4p298 and Roe1997_Acad-Radiol_v4p587. Specifically, it allows the variance components to depend on the truth and the modality. For the simpler Roe and Metz model, you can enter the smaller set of parameters into sim.gRoeMetz.config and get back the larger set of parameters and then used with this function.

Usage

```
sim.gRoeMetz(config)
```

Arguments

`config` [list] of simulation parameters:

- Experiment labels and size
  - modalityID.A: [chr] label modality A
  - modalityID.B: [chr] label modality B
  - nR: [num] number of readers
  - nC.neg: [num] number of signal-absent cases
  - nC.pos: [num] number of signal-present cases
- There are six fixed effects:
  - mu.neg: [num] signal-absent (neg, global mean)
  - mu.pos: [num] signal-present (pos, global mean)
  - mu.Aneg: [num] modality A signal-absent (Aneg, modality effect)
  - mu.Bneg: [num] modality B signal-absent (Bneg, modality effect)
  - mu.Apos: [num] modality A signal-present (Apos, modality effect)
  - mu.Bpos: [num] modality B signal-present (Bpos, modality effect)
- There are six random effects that are independent of modality
  - var_r.neg: [num] variance of random reader effect
  - var_c.neg: [num] variance of random case effect
  - var_rc.neg: [num] variance of random reader by case effect
  - var_r.pos: [num] variance of random reader effect
  - var_c.pos: [num] variance of random case effect
  - var_rc.pos: [num] variance of random reader by case effect
- There are six random effects that are specific to modality A
  - var_r.Aneg: [num] variance of random reader effect
  - var_c.Aneg: [num] variance of random case effect

  - var_rc.Aneg: [num] variance of random reader by case effect
  - var_r.Apos: [num] variance of random reader effect
  - var_c.Apos: [num] variance of random case effect
  - var_rc.Apos: [num] variance of randome reader by case effect
- There are six random effects that are specific to modality B
  - var_r.Bneg: [num] variance of random reader effect
  - var_c.Bneg: [num] variance of random case effect
  - var_rc.Bneg: [num] variance of random reader by case effect
  - var_r.Bpos: [num] variance of random reader effect
  - var_c.Bpos: [num] variance of random case effect
  - var_rc.Bpos: [num] variance of randome reader by case effect

Details

The simulation is a linear model with six fixed effects related to modality and truth and 18 normally distributed independent random effects for readers, cases, and the interaction between the two. Here is the linear model:

L.mrct = mu.t + mu.mt
+ reader.rt + case.ct + readerXcase.rct
+ modalityXreader.mrt + modalityXcase.mct + modalityXreaderXcase.mrct

- m=modality (levels: A and b)
- t=truth (levels: neg and Pos)
- mu.t is the global mean for t=neg and t=pos cases
- mu.mt is the modality specific fixed effects for t=neg and t=pos cases
- the remaining terms are the random effects: all independent normal random variables

Value

dFrame.imrmc [data.frame] with (nC.neg + nC.pos)*(nR+1) rows including

- readerID: [Factor] w/ nR levels "reader1", "reader2", ...
- caseID: [Factor] w/ nC levels "case1", "case2", ...
- modalityID: [Factor] w/ 1 level config$modalityID
- score: [num] reader score

Note that the first nC.neg + nC.pos rows specify the truth labels for each case. For these rows, the readerID must be "truth" or "-1" and the score must be 0 for negative cases and 1 for postive cases.

Examples

```
# Create a sample configuration object
config <- sim.gRoeMetz.config()
# Simulate an MRMC ROC data set
dFrame.imrmc <- sim.gRoeMetz(config)
# Analyze the MRMC ROC data
result <- doIMRMC(dFrame.imrmc)
```

`sim.gRoeMetz.config` *Create a configuration object for the sim.gRoeMetz program*

Description

This function creates a configuration object for the Roe & Metz simulation model to be used as input for the sim.gRoeMetz program. The default model returned when there are no arguments given to the function is the "HH" model from Roe1987_Acad-Radiol_v4p298. Following that paper, The user can specify three parameters related to experiment size (nR, nC.neg, nC.pos) and five parameters parameters specifying a linear model that does not depend on modality or truth (mu.neg, mu.pos, var_r, var_c, var_rc).

Usage

```
sim.gRoeMetz.config(nR = 5, nC.neg = 40, nC.pos = 40, mu.neg = 0,
  mu.pos = 1, var_r = 0.03, var_c = 0.3, var_rc = 0.2)
```

Arguments

`nR`	Number of readers (default = 5)
`nC.neg`	Number of signal-absent cases (default = 25)
`nC.pos`	Number of signal-present cases (default = 25)
`mu.neg`	Mean fixed effect of signal-absent distribution (default = 0.0) Modality specific parameters are set to zero: mu.Aneg = mu.Bneg = 0
`mu.pos`	Mean fixed effect of signal-present distribution (default = 1.0) Modality specific parameters are set to zero: mu.Apos = mu.Bpos = 0
`var_r`	Variance of reader random effect (default = 0.03) var_r.neg = var_r.pos = var_r.Aneg = var_r.Apos = var_r.Bneg = var_r.Bpos = var_r
`var_c`	Variance of case random effect (default = 0.30) var_c.neg = var_c.pos = var_c.Aneg = var_c.Apos = var_c.Bneg = var_c.Bpos = var_c
`var_rc`	Variance of reader.by.case random effect (default = 0.20) var_rc.neg = var_rc.pos = var_rc.Aneg = var_rc.Apos = var_rc.Bneg = var_rc.Bpos = var_rc

Details

If no arguments, this function returns a default simulation configuration for sim.gRoeMetz

Value

config [list] Refer to the sim.gRoeMetz input variable

Examples

```
# Create a sample configuration object
config <- sim.gRoeMetz.config()
# Simulate an MRMC ROC data set
dFrame.imrmc <- sim.gRoeMetz(config)
# Analyze the MRMC ROC data
result <- doIMRMC(dFrame.imrmc)
```

`simMRMC` *Simulate an MRMC data set*

Description

This program simulates observations from one set of readers scoring one set of cases. It produces one modality and one truth state of ROC data following Roe1997_Acad-Radiol_v4p298 and Roe1997_Acad-Radiol_v4p587. In order to produce an entire ROC data set, please use sim.gRoeMetz.

Usage

```
simMRMC(simMRMC.config)
```

Arguments

`simMRMC.config` [list] of simulation parameters:

- modalityID [chr] label modalityID
- readerIDs [factor] the ID of each reader
- caseIDs [factor] the ID of each case
- mu [num] mean
- var_r [num] variance of random reader effect
- var_c [num] variance of random case effect
- var_rc [num] variance of random reader by case effect

Details

The simulation is a linear model with one fixed effect and three normally distributed independent random effects corresponding to readers, cases, and an interaction between the two.

L.rc = mu + readerEffect.r + caseEffect.c + readerXcaseEffect.rc

Value

L [data.frame] with nC*nR rows of 4 variables

- L$modalityID [factor] determined by input modalityID
- L$readerID [factor] determined by input readerIDs
- L$caseID [factor] determined by input caseIDs
- L$score [num] R.r + C.c + RC.rc

- r = 1,2,...,nR
- c = 1,2,...,nC
- R.r ~ N(0,var_r)
- C.c ~ N(0,var_c)
- RC.rc ~ N(0,var_rc)

Examples

```
# Create a sample configuration object
config <- sim.gRoeMetz.config()
# Simulate an MRMC ROC data set
dFrame.imrmc <- sim.gRoeMetz(config)
# Analyze the MRMC ROC data
result <- doIMRMC(dFrame.imrmc)
```

---

simRoeMetz.example *Simulates a sample MRMC ROC experiment*

Description

Simulates a sample MRMC ROC experiment

Usage

```
simRoeMetz.example()
```

Value

dFrame.imrmc [data.frame] Please refer to the description of the simRoeMetz return variable

Examples

```
# Simulate a sample MRMC ROC data set
dFrame.imrmc <- simRoeMetz.example()
# Analyze the MRMC ROC data
result <- doIMRMC(dFrame.imrmc)
```

successDFtoROCdf — *Convert an MRMC data frame of successes to one formatted for doIMRMC*

Description

Convert an MRMC data frame of successes to one formatted for doIMRMC

Usage

```
successDFtoROCdf(df)
```

Arguments

df	Each row contains a success observation for one reader evaluating one case

Value

data frame ready for doIMRMC

undoIMRMCdf — *Convert a doIMRMC formatted data frame to a standard data frame with all factors.*

Description

Convert a doIMRMC formatted data frame to a standard data frame with all factors.

Usage

```
undoIMRMCdf(df.MRMC)
```

Arguments

df.MRMC	This data frame includes columns for readerID, caseID, modalityID, score. Each row is a reader x case x modality observation from the study In addition to observations from the study, this data frame requires rows specifying the truth for each caseID. For truth specifications, the readerID needs to equal "truth" or "-1", modalityID can be anything ("truth" is a good choice), and score should be 0 for signal-absent normal case, 1 for signal-present disease case.

Details

Delete rows specifying truth and put the truth information on every row.

Value

output a data frame with columns readerID, caseID, modalityID, score, truth

uStat11.diff *Create the kernel and design matrices for uStat11*

Description

The kernel is the difference kernel

Usage

```
uStat11.diff(df.input, keyColumns = c("readerID", "caseID", "modalityID",
  "score"), modalitiesToCompare = c("modalityA", "modalityB", "modalityA",
  "modalityB"))
```

Arguments

df.input	Data frame of observations, one per row. Columns also identify random and fixed effects.
keyColumns	The required columns
modalitiesToCompare	The factors identifying the modalities to compare

uStat11.identity *Create the kernel and design matrices for uStat11*

Description

The kernel is the identity kernel

Usage

```
uStat11.identity(df.input, keyColumns = c("readerID", "caseID", "modalityID",
  "score"), modalitiesToCompare = c("modalityA", "modalityB"))
```

Arguments

df.input	Data frame of observations, one per row. Columns also identify random and fixed effects.
keyColumns	The required columns
modalitiesToCompare	The factors identifying the modalities to compare

`uStat11.jointD` *Analysis of U-statistics degree 1,1*

Description

These two functions calculate the mean and variance of a user-specified U-statistic kernel, which is a function of cross-correlated scores.

The motivation for this analysis is data collected in imaging studies where multiple readers read multiple cases in different modes or modalities. The goal is to evaluate the variance of a reader- and case-averaged endpoint, accounting for cross-correlated data arising from two random effects: the random reader skill and the random case difficulty. This analysis is sometimes referred to as an MRMC analysis. Of course, the random effects can be from sources other than readers and cases.

Usage

```
uStat11.jointD(df.input, keyColumns = c("readerID", "caseID", "modalityID",
  "score"), modalitiesToCompare = c("modalityA", "modalityB"),
  kernelFlag = 1)

uStat11.conditionalD(df.input, keyColumns = c("readerID", "caseID",
  "modalityID", "score"), modalitiesToCompare = c("modalityA", "modalityB"),
  kernelFlag = 1)
```

Arguments

`df.input` Data frame of observations, one per row. Columns identify random effects, fixed effects, and the observation. Namely,

- `readerID`: The factor corresponding to the different readers in the study. The readerID is treated as a random effect.
- `caseID`: The factor corresponding to the different cases in the study. The caseID is treated as a random effect.
- `modalityID`: The factor corresponding to the different modalities in the study. The modalityID is treated as a fixed effect.
- `score`: The score given by the reader to the case for the modality indicated.

`keyColumns` Identify the factors corresponding to the readerID, caseID, modalityID, and score (or alternative random and fixed effects).

`modalitiesToCompare` The factors identifying the modalities to compare.

`kernelFlag` This determines the kernel function

- `kernelFlag` = 1 == identity kernel: requires two modalities: A,B.
- `kernelFlag` = 2 == kernel of the difference in modalities: requires four modalities: A,B,C,D.

Details

uStat11.conditionalD is identical to uStat11.jointD when the study is fully-crossed: when every reader readers all the cases in both modalities. For arbitrary study designs the two functions differ according to how the components of variance are estimated.

- uStat11.conditionalD follows Gallas2007_J-Opt-Soc-Am-A_v24pB70 <doi:10.1364/JOSAA.24.000B70> and estimates the components of variance (which isolate combinations of different random effects) with nested conditional means.
- uStat11.jointD is analogous to the method in Gallas2008_Neural-Networks_v21p387 <doi:10.1016/j.neunet.2007.12 and estimates the components of variance (which isolate combinations of different random effects) with a joint distribution over all the observations giving equal weight to each one.

Both functions yield unbiased variance estimates. Our simulations find that uStat11.conditionalD is statistically more efficient than uStat11.jointD (its variance estimate is more precise), but it is slower.

Please refer to the tests/testthat folder of the package for examples using these functions.

Value

This function calculates the mean and variance of the indicated U-statistic kernel, which is a function of the scores. For the identity kernel, we simply return the mean and variance of the scores.

The function returns a list of outputs. Many of these outputs have three elements.

- If kernelFlag = 1 == identity kernel, the first element corresponds to the mean score of modality A, the second corresponds to mean score of modality B, and the third corresponds to the mean of the difference in scores from modality A and B.
- If kernelFlag = 2 == difference kernel, the first element corresponds to the mean difference in scores from modalities A and B, the second element corresponds to the mean difference in scores from modalities C and D, and the third elements corresponds to the difference of the just-mentioned differences.

There are 16 outputs:

- mean: See description above.
- var: The variance of the mean.
- var.1obs: The variance of one reader-case-modality observation.
- meanPerR The reader-specific means.
- nR The number of readers in the study.
- nC The number of cases in the study.
- nCperR The number of cases evaluated by each reader for each modality.
- moments The second order moments of the problem.
- coeff The coefficients corresponding to the second-order moments such that the scalar product between the moments and coefficients yields the variance.
- kernel.A A matrix showing the kernel evaluated for each combination of each reader and case for modality A (or AB).

- `design.A` A matrix showing the what data exists for each combination of each reader and case for modality A (or AB).
- `kernel.B` A matrix showing the kernel evaluated for each combination of each reader and case for modality B (or CD).
- `design.B` A matrix showing the what data exists for each combination of each reader and case for modality B (or CD).

Examples

```
# Create an MRMC data frame
# Refer to Gallas2014_J-Med-Img_v1p031006
simRoeMetz.config <- sim.gRoeMetz.config()

# Simulate data
df.MRMC <- sim.gRoeMetz(simRoeMetz.config)

# Reformat data
df <- undoIMRMCdf(df.MRMC)

# Grab part of the data
df <- droplevels(df[grepl("pos", df$caseID), ])

#### uStat11.jointD.identity ####
# Calculate the reader- and case-averaged difference in scores from testA and testB
# (kernelFlag = 1 specifies the U-statistics kernel to be the identity)
result.jointD.identity <- uStat11.jointD(
  df,
  kernelFlag = 1,
  keyColumns = c("readerID", "caseID", "modalityID", "score"),
  modalitiesToCompare = c("testA", "testB"))

cat("\n")
cat("uStat11.jointD.identity \n")
print(result.jointD.identity[1:2])
```

Index