Package: permRand (via r-universe)

June 1, 2026

Title Permutation Randomization

Version 1.0.0

Description The package provides randomization using permutation for applications. To provide a Quality Control check, QC samples can be randomized within strata. A second function allows for the ability to "switch" samples to meet set requirements and perform a certain amount of minimization on these switches. The functions are flexible for users by specifying strata size and number of QC samples per strata. The randomization meets the following requirements • QC sample requirements: QC samples not adjacent, QC samples from same mother must follow certain patterns. • Matched sample sets must be within a single strata, and next to each other.

URL https://github.com/m-mellers/permRand

BugReports michelle.mellers.ctr@usuhs.edu

License use_gpl3_license()

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Suggests knitr, rmarkdown

VignetteBuilder knitr

Imports tidyverse

Config/pak/sysreqs
cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev make libharfbuzz-dev libicu-dev libjpeg-dev libpng-dev libtiff-dev libuv1-dev libwebp-dev libxml2-dev libssl-dev libx11-dev zlib1g-dev

Repository https://m-mellers.r-universe.dev

Date/Publication 2025-08-22 18:02:33 UTC

RemoteUrl https://github.com/m-mellers/permrand

RemoteRef HEAD

RemoteSha 1d6d45e23b923e80dbb7260c693262b9aab030e4

Contents

`allRand` *Data Randomization*

Description

Randomizes aliquots to batches.

Usage

```
allRand(dataR, batchTot, numQC, withinN, numMatch, chkRep)
```

Arguments

`dataR`	Data for randomization.
`batchTot`	c(batchTot1, batchTot2) sizes of plates, just use one plate per batch, batch size inclusive of QC samples.
`numQC`	Number of QC samples per batch.
`withinN`	Number of samples away that the QC samples must be from each other.
`numMatch`	Number of QC samples from a single mother within a batch.
`chkRep`	Check if there is a repeat of the groups within the batches.

Value

A dataset with serum order randomized.

`batchCount` *Tests the number in each batch.*

Description

Counts the number of samples that are in each of the batches.

Usage

```
batchCount(dataS, batchSizeT)
```

Arguments

`dataS`	Test dataset.
`batchSizeT`	Batch size to test for.

Value

The ID of any batch that does not contain the specified number of samples.

`countQC` *Number of QC in each batch.*

Description

Count how many QC samples are in each of the batches, and if it doesn't match the number specified.

Usage

```
countQC(dataS, QCN)
```

Arguments

`dataS`	Test dataset.
`QCN`	Number of QC samples per batch.

Value

The output includes any batches that does not contain the number of QC samples specified.

`formatRand` *Formats data*

Description

The function `formatRand` formats the dataset for the randomization function. This function inputs serum data for both the study subjects and QC.

Usage

```
formatRand(QCdata, serumIDR, serumPack)
```

Arguments

`serumIDR`	Serum data with serumIDs.
`serumPack`	Serum data with packing lists.
`QCData`	QC data.

Value

A dataset that is formatted and ready for the randomization file.

`orderCases` *Number of single group in sequence.*

Description

Tests if a large number of cases or controls are next to each other.

Usage

```
orderCases(dataI, betW)
```

Arguments

`dataI`	Dataset to be tested.
`betW`	Number of cases or controls to check if they are next to each other.

Value

The output stores if there are any cases or controls together beyond a certain specified value.

`outputLab` *Output labels.*

Description

Creates output labels for the randomized groups.

Usage

```
outputLab(dataOut, blind, origP, maxRows, maxCols, newPack)
```

Arguments

`dataOut`	Dataset to be formatted for packing list.
`origP`	Indicator, 0/1, inclusion of the original packing location (1) or deletion of the packing location (0).
`maxRows`	Maximum row for the output dataset.
`maxCols`	Maximum column for the output dataset.
`newPack`	0/1 indicator to generate new packing locations.
`Blind`	Indicator 0/1 select if a blinded (1) or unblinded(0) packing list is to be generated.

Value

A dataset to be used for packing lists.

Examples

```
blind <- outputLab(dataOut=serumRand,blind=1,origP=0,maxRows=9,maxCols=9,newPack=1)
unBlindSw <- outputLab(serumSwitchP,blind=0,origP=.,maxRows=.,maxCols=.,newPack=0)
```

`randTest` *QC Identifiers*

Description

We first assign IDs linking mother/child and events using the function `randTest`.

Usage

```
randTest(dataMom, dataChild, maxAliq, nEvent)
```

Arguments

`dataMom`	The mother dataset.
`dataChild`	Child dataset.
`maxAliq`	Number of aliquots per mother aliquot.
`nEvent`	Number of aliquots per each of event or lab.

Value

The output of the function is a dataset with the ID links.

`switchR` *Switching Generating Function*

Description

Minimizes switches without completely re-randomizing the locations.

Usage

```
switchR(dataIn, numqc, numqcM, batchS)
```

Arguments

`dataIn`	Randomized dataset.
`numqc`	Number of QC samples per set.
`numqcM`	Numberof QC matching samples.
`batchS`	New batch size.

Value

A dataset with switches indicated.

Examples

```
serumSwitch <- switchR(dataIn=serumRand,numqc=2,numqcM=2,batchS=43)
```

`testCCAcross` *Sample groups within batches*

Description

Tests and finds sample groups that are across batches.

Usage

```
testCCAcross(dataS)
```

Arguments

`dataS`	The test dataset.

Value

The output lists all batches with not enough QC sample sets or the QC samples do not come from the same mother.

`testPair` *Ensures complete sets.*

Description

Tests if sets are next to each other. Any sets that are not next to each other are flagged.

Usage

```
testPair(dataS)
```

Arguments

`dataS`	Test dataset.

Value

The output reports any sets that are separated in the "loc".

`testQCmatch` *Tests QC matches*

Description

Measures that every batch has at least the specified number of matching QC sample sets in a batch.

Usage

```
testQCmatch(dataS, numQCs, numMatch)
```

Arguments

`dataS`	Randomized data.
`numQCs`	Number of QCs specified per dataset.
`numMatch`	Number of QC samples form a single mother within a batch.

Value

The output lists all batches with not enough QC sample sets or the QC samples do not come from the same mother.

`testRand` *Test Dataset*

Description

Generates test data for the randomization functions.

Usage

```
testRand(rowSize, colSize, studySize, expNS, numCC, QCpct, child)
```

Arguments

`rowSize`	Max row size.
`colSize`	Max column size.
`studySize`	Number of cases.
`expNS`	Number of aliquots per case/control.
`numCC`	Number of controls per case.
`QCpct`	Percent of QCs for number of samples.
`child`	Number of children per "mother' aliquot.

Value

A practice dataset.

Examples

```
testR <- testRand(rowSize=20,colSize=15,studySize=1000,expNS=7000,numCC=2,QCpct=0.05,child=4)
```

`uniqueID` *Unique IDs*

Description

Test for unique IDs.

Usage

```
uniqueID(testD, IDN)
```

Arguments

`testD`	Test dataset.
`IDN`	ID to test.

Value

Any IDs that are not unique.

Index