# Package 'Omixer'

November 12, 2026

**Type** Package

**Title** Omixer: multivariate and reproducible sample randomization to proactively counter batch effects in omics studies

**Version** 1.22.0

**Description** Omixer - an Bioconductor package for multivariate and reproducible sample randomization, which ensures optimal sample distribution across batches with well-documented methods. It outputs lab-friendly sample layouts, reducing the risk of sample mixups when manually pipetting randomized samples.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Imports** dplyr, ggplot2, forcats, tibble, gridExtra, magrittr, readr, tidyselect, grid, stats, stringr

**Depends** R (>= 4.0.0)

**RoxygenNote** 7.1.1

**Suggests** knitr, rmarkdown, BiocStyle, magick, testthat

**VignetteBuilder** knitr

**biocViews** DataRepresentation, ExperimentalDesign, QualityControl, Software, Visualization

**BugReports** https://github.com/molepi/Omixer/issues

**git_url** https://git.bioconductor.org/packages/Omixer

**git_branch** RELEASE_3_23

**git_last_commit** 4552d73

**git_last_commit_date** 2026-04-28

**Repository** Bioconductor 3.23

**Date/Publication** 2026-11-11

**Author** Lucy Sinke [cre, aut]

**Maintainer** Lucy Sinke <l.j.sinke@lumc.nl>

# Contents

Omixer . . . . . 2
omixerCorr . . . . . 2
omixerRand . . . . . 3
omixerSheet . . . . . 5
omixerSpecific . . . . . 6

**Index** **8**

---

`Omixer` *Multivariate Randomization*

---

## Description

Omixer provides functions to perform optimal randomization of sample lists prior to omic profiling. This minimizes the correlation between biological factors and technical covariates, whilst ensuring there is insufficient evidence for any of these associations.

## Details

Following this multivariate randomization, Omixer can also be used to smoothly bridge the gap between dry and wet labs, by creating visually intuitive sample sheets.

With clear documentation and the possibility to reproduce any output sample list, Omixer sets the standard for transparency and reproducibility in this often vague methodological step.

## Author(s)

Lucy Sinke <l.j.sinke@lumc.nl>

---

`omixerCorr` *Correlation Tests*

---

## Description

This function uses appropriate tests of correlation between two variables and stores the estimate and p-value in a list.

## Usage

```
omixerCorr(x, y)
```

## Arguments

| | |
|---|---|
| `x` | Randomization variable (e.g. age) |
| `y` | Technical covariate (e.g. plate number) |

## Details

For two categorical variables, the Cramer's V estimate is stored alongside chi-square p-value. For all other combinations of variables, Pearson's correlation coefficient and p-value are stored.

Please note: variables will be converted to numeric class within this function.

## Value

List of correlation estimate and p-value

## Examples

```
library(tibble)
library(forcats)
library(stringr)

sampleList <- tibble(sampleId=str_pad(1:48, 4, pad="0"),
sex=as_factor(sample(c("m", "f"), 48, replace=TRUE)),
age=round(rnorm(48, mean=30, sd=8), 0),
smoke=as_factor(sample(c("yes", "ex", "never"), 48, replace=TRUE)),
date=sample(seq(as.Date('2008/01/01'), as.Date('2016/01/01'),
    by="day"), 48))

omixerCorr(sampleList$age, sampleList$sex)
```

---

`omixerRand` *Multivariate Randomization*

---

## Description

As the main function of the Omixer package, this function outputs a randomized sample list that minimizes correlations between biological factors and technical covariates.

## Usage

```
omixerRand(
  df,
  sampleId = "sampleId",
  block = "block",
  iterNum = 1000,
  wells,
  div = "none",
  positional = FALSE,
  plateNum = 1,
  layout,
  mask = 0,
  techVars,
  randVars
)
```

## Arguments

| | |
|---|---|
| df | Sample list |
| sampleId | String specifying sample ID variable |
| block | Paired sample identifier |
| iterNum | Number of layouts to generate |
| wells | Number of wells on a plate |
| div | Plate subdivisions |
| positional | Logical indicator of positional batch effects |
| plateNum | Number of plates |
| layout | Custom plate layout as data frame |
| mask | Wells to be left empty |
| techVars | Technical covariates |
| randVars | Randomization variables |

## Value

Selected randomized sample list as a data frame

Randomization environment of optimal list generation

## Examples

```
library(tibble)
library(forcats)
library(stringr)

sampleList <- tibble(sampleId=str_pad(1:48, 4, pad="0"),
sex=as_factor(sample(c("m", "f"), 48, replace=TRUE)),
age=round(rnorm(48, mean=30, sd=8), 0),
smoke=as_factor(sample(c("yes", "ex", "never"), 48, replace=TRUE)),
date=sample(seq(as.Date('2008/01/01'), as.Date('2016/01/01'),
               by="day"), 48))

randVars <- c("sex", "age", "smoke", "date")

omixerLayout <- omixerRand(sampleList, sampleId="sampleId",
block="block", iterNum=10, wells=48, div="row",
plateNum=1, randVars=randVars)
```

omixerSheet *Sample Sheet Generation*

## Description

This function will generate visually intuitive plate layouts for the wet lab, with the option to colour code different types of samples (e.g. for studies investigating multiple tissues).

## Usage

```
omixerSheet(omixerLayout = omixerLayout, group)
```

## Arguments

| | |
|---|---|
| omixerLayout | Randomized sample list |
| group | Colour-coding indicator |

## Value

PDF of sample layout in working directory

## Examples

```
library(tibble)
library(forcats)
library(stringr)

sampleList <- tibble(sampleId=str_pad(1:48, 4, pad="0"),
sex=as_factor(sample(c("m", "f"), 48, replace=TRUE)),
age=round(rnorm(48, mean=30, sd=8), 0),
smoke=as_factor(sample(c("yes", "ex", "never"), 48, replace=TRUE)),
date=sample(seq(as.Date('2008/01/01'), as.Date('2016/01/01'),
               by="day"), 48))

randVars <- c("sex", "age", "smoke", "date")

omixerLayout <- omixerRand(sampleList, sampleId="sampleId",
block="block", iterNum=10, wells=48, div="row",
plateNum=1, randVars=randVars)

omixerSheet(omixerLayout)
```

`omixerSpecific` *Sample List Regeneration*

## Description

Regenerate an Omixer-produced randomized sample list quickly, after setting up the random environement from omixerRand

## Usage

```
omixerSpecific(
  df,
  sampleId = "sampleId",
  block = "block",
  wells,
  div = "none",
  positional = FALSE,
  plateNum = 1,
  layout,
  mask = 0,
  techVars,
  randVars
)
```

## Arguments

| | |
|---|---|
| `df` | Sample list |
| `sampleId` | String specifying sample ID variable |
| `block` | Paired sample identifier |
| `wells` | Number of wells on a plate |
| `div` | Plate subdivisions |
| `positional` | Logical indicator of positional batch effects |
| `plateNum` | Number of plates |
| `layout` | Custom plate layout as data frame |
| `mask` | Wells to be left empty |
| `techVars` | Technical covariates |
| `randVars` | Randomization variables |

## Value

Chosen layout as a data frame

## Examples

```
library(tibble)
library(forcats)
library(stringr)

sampleList <- tibble(sampleId=str_pad(1:48, 4, pad="0"),
sex=as_factor(sample(c("m", "f"), 48, replace=TRUE)),
age=round(rnorm(48, mean=30, sd=8), 0),
smoke=as_factor(sample(c("yes", "ex", "never"), 48, replace=TRUE)),
date=sample(seq(as.Date('2008/01/01'), as.Date('2016/01/01'),
               by="day"), 48))

randVars <- c("sex", "age", "smoke", "date")

omixerLayout <- omixerSpecific(sampleList, sampleId="sampleId",
block="block", wells=48, div="row",
plateNum=1, randVars=randVars)
```

# Index

Omixer, 2
omixerCorr, 2
omixerRand, 3
omixerSheet, 5
omixerSpecific, 6