Package ‘SimVitD’

April 19, 2020

Type Package

Title Simulation Tools for Planning Vitamin D Studies

Version 0.1.2

Description Simulation tools for planning Vitamin D studies including approximation of power. Individual vitamin D status profiles are simulated, modelling population heterogeneity. Individuals experience random exposures to infectious agents. Development of infection considers a heterogeneous treatment effect. This is incorporated into the power approximation.

License GPL (>= 2)

Maintainer Jason Wyse <`wyseja@tcd.ie`>

Author Rebecca Mangan [aut],
Jason Wyse [aut, cre],
Lina Zgaga [aut]

Depends poisson

Encoding UTF-8

LazyData true

NeedsCompilation no

Repository CRAN

Date/Publication 2020-04-19 18:30:02 UTC

R topics documented:

SimVitD-package 2
exposure.levels 3
infection.count 4
infection.count.plot 5
intensity.function 6
plot.exposure.levels 6
plot.power.calc 7
plot.vitd.curve 8
power.calc 8
print.exposure.levels 10

print.infection.count 10
print.power.calc 11
print.vitd.curve 11
pvalue.calc 12
rr.curve.plot 12
rr.profile.plot 13
summary.exposure.levels 14
summary.infection.count 14
summary.power.calc 15
summary.vitd.curve 15
vitd.curve 16

Index **18**

`SimVitD-package` *Simulation Tools for Vitamin D studies.*

Description

Simulation tools for planning vitamin D studies including power estimation. Population heterogeneity in vitamin D status is modelled by simulating individual vitamin D status profiles. Indviduals are assumed to be exposed randomly to an infectious agent. Exposures are simulated at the individual level. Development of infection from exposure is radomly simulated, with the probability of infection being modulated by the individual's vitamin D status at time of exposure. Power estimates are calculated, accounting for heterogeneous treatment effects within individuals.

Details

Package:	SimVitD
Type:	Package
Version:	1.0.1
Date:	2019-06-14
License:	GPL (>= 2)
LazyLoad:	yes

Power calculations in `SimVitD` proceed by simulating many realisations of a study. Within each study, individuals' vitamin D status trajectories and potential exposures and protections from infections are simulated separately. This is akin to a microsimualation. The core steps of the simulation approach being proposed are:

(i) simulation of an individual's vitamin D status trajectory

(ii) simulation of an individual's exposures to infectious agents

(iii) determination of the probability of developing infection at each exposure time, dependent on an individual's vitamin D status at exposure

(iv) simulation of contracting an infection at conditional on step (iii).

There is functionality to carry out estimation of the power of detecting differences between different types of supplementation schemes. This can be expanded to aid in study planning and design.

The fuctions `vitd.curve` and `exposure.levels` and the accompanying documentation provide a gateway into the simulation tools. Estimation of the power of detecting differences in treatment effects are based on microsimulation; a proposed study is simulated a large number of times and within each of these studies individual vitamin D status profiles are generated to reflect variability in the population. The package considers immunoprotective effects of vitamin D; within each study, exposures to an infectious agent are simulated for each individual. For each exposure, the immune response modulated by the individual's vitamin D status is also simulated. The resulting datasets of cases/non-cases provide the basis for empirical power estimation.

Author(s)

Rebecca Mangan, Jason Wyse, Lina Zgaga

Examples

```
# simulate individual Vitamin D profiles
indprofiles <- vitd.curve( N=10, type="placebo" )

# make a plot of all these profiles
plot( indprofiles )

# intensity function for exposures to infection
intensfun <- intensity.function( summer.rate=0.1, winter.rate=0.9 )

# exposure times
expos <- exposure.levels( indprofiles, rate=2, intensfun, end=2 )

# plot of exposures on top of Vitamin D profiles
plot( expos )

# infection calculation
infect <- infection.count( expos )

# plot infection points on top of exposure points
infection.count.plot( expos, infect )
```

`exposure.levels` *Generate exposures to infectious agent and vitamin D status*

Description

`exposure.levels` creates a list of the exposures to an infectious agent and the status of vitamin D at the time of each exposure for each participant in a group of size N. The exposure times are simulated from a non homogeneous poisson process.

Usage

```
exposure.levels( x, rate, intensity.func = intensity.function(), end = 1 )
```

Arguments

`x`	An object of class `vitd.curve`.
`rate`	numeric, mean number of exposures per week, the rate at which exposures occur in the equivalent homogeneous Poisson process.
`intensity.func`	A function that takes time as sole argument and returns value between 0 and 1, scaling the rate argument. For example, exposures may be higher at certain times for seasonal infections.Input to `nhpp` function see `poisson` package.
`end`	Numeric, when the study ends.

Value

`exposure.levels` returns an object of class `exposure.levels` that is a list. The list has the following slots.

`exposures`	The exposure times to infection for each participant in the group of size N.
`levels`	The vitamin D staus levels for each participant at the time of exposure to infection.

`infection.count` *Generate whether a participant contracted a infection at exposure times.*

Description

Generates probabilities of developing infection, the relative risk and whether a participant becomes infected or not at exposure times.

Usage

```
infection.count( expos, baseline = 0.03, RR = 3, holding.time = 2, lohi.vit = c(10,70) )
```

Arguments

`expos`	An object of class `exposure.levels`.
`baseline`	Numeric, baseline prevalence of developing infection at any exposure time. Probability of developing infection when a participant has maximum (fully replete) vitamin D levels.
`RR`	Numeric, the relative risk of the study. The difference between becoming infected at highest and lowest vitamin D levels.
`holding.time`	Numeric, the mean number of weeks for after becoming infected when the participant can not become re-infected. Times are simulated independently from an exponential distribution with this mean.
`lohi.vit`	A vector of length 2 giving the level at which a participant is considered to have insufficient and sufficient vitamin D status levels.

Value

`infection.count` returns an object of class `infection.count` that is a list. The list has the following slots.

`baseline`	The baseline prevalence of disease.
`RR`	The relative risk of the study.
`inflection`	The inflection points of the relative risk curve.
`probs`	Matrix, the probability of developing infection at each exposure time for each participant.
`relativerisk`	Matrix, the relative risk of infection at each exposure time for each participant.
`infection`	Matrix, 1 if participant was infected at the corresponding exposure time, 0 if participant was not infected at exposure time.
`count`	Number of infections for each participant over the course of the study.
`mean`	The mean number of infections for the group over the course of the study.

`infection.count.plot` *Plotting the times of disease on the vitamin D curves.*

Description

Plot whether a participant was disease or not as points on top of the vitamin D curves plotted from `plot.vitd.curve`.

Usage

```
infection.count.plot( expos, infect, pch = 20, cex = 1.5, col = "blue" )
```

Arguments

`expos`	An object of class `exposure.levels`.
`infect`	An object of class `infection.count`.
`pch`	Integer, specifying symbol for the points.
`cex`	Numeric, value giving the amount by which plotting the points should be magnified.
`col`	Character, value for the colour of the points.

`intensity.function` *Generate an intensity function*

Description

Generates intensity function, function with time as sole argument, used as input to `nhhp.event.times` function from `poisson` package

Usage

```
intensity.function( summer.rate = 0, winter.rate = 1, flu = TRUE )
```

Arguments

`summer.rate`	Numeric value bwteeen 0 and 1, the rate at which exposures occur in summer months.
`winter.rate`	Numeric value between 0 and 1, the rate at which exposures occur in winter months.
`flu`	Logical argument. TRUE: to define winter months as flu season (Sep-May in Northern Hemisphere, Mar-Nov in Southern Hemisphere), FALSE: to have constant exposure rate all year.

Value

`intensity.function` returns a function that takes time as sole argument and returns value between 0 and 1, the returned function acts as `intensity.func` for input to `exposure.levels`.

`plot.exposure.levels` *Plotting the exposures to infection on the vitamin D curves.*

Description

Plot the exposures to infection as points on top of the vitamin D curves plotted from `plot.vitd.curve`.

Usage

```
    ## S3 method for class 'exposure.levels'
plot( x, pch = 1, cex = 1.3, col = "red", ... )
```

Arguments

`x`	An object of class `exposure.levels`.
`pch`	Integer, specifying symbol for the points.
`cex`	Numeric, value giving the amount by which plotting the points should be magnified.
`col`	Character, value for the colour of the points.
`...`	Additional arguments to be passed to the `plot` function.

`plot.power.calc` *Plotting a power.calc object*

Description

Plot the power between two groups as the number of participants and the relative risk vary.

Usage

```
    ## S3 method for class 'power.calc'
plot( x, col = "hotpink", lwd = 1.5, lty = 1,
    ylab = "power", x.legend = NULL, y.legend = NULL,
    main.legend = "", legend.size = 1, ... )
```

Arguments

`x`	An object of class `power.calc`.
`col`	Character, value for the colour of the line.
`lwd`	Numeric, a value for the line width.
`lty`	Numeric, a value for the line type.
`ylab`	A title for the y-axis.
`x.legend`	Numeric, a value for the x position of the legend. If not given, tries to default to a sensible value.
`y.legend`	Numeric, a value for the y position of the legend. If not given, tries to default to a sensible value.
`main.legend`	Character, a title for the legend.
`legend.size`	Numeric, a value for the size of the legend.
`...`	Additional arguments to be passed to the `plot` function.

`plot.vitd.curve` *Plotting a vitd.curve object*

Description

Plot the vitamin D curves for a group of size N.

Usage

```
  ## S3 method for class 'vitd.curve'
plot( x, main = " ", xlab = " ", ylab = "25 Hydroxy Vitamin D", ... )
```

Arguments

`x`	An object of class `vitd.curve`.
`main`	Main title for the plot.
`xlab`	A title for the x-axis.
`ylab`	A title for the y-axis.
`...`	Additional arguments to the `plot` function.

`power.calc` *Generate the power to detect the difference between two groups*

Description

generates a value for the power between two groups

Usage

```
power.calc( num.participants, num.sims, test.type, sig.level = 0.05,
                      vitdcurves.placebo, vitdcurves.treatment,
                      baseline = 0.03, RR = 3,
                      rate = 1, intensity.func = intensity.extreme,
                      holding.time = 2, mc.error = 1, lohi.vit = c(10,70) )
```

Arguments

`num.participants`	Numeric, the number of participants in each group to calculate the power for.
`num.sims`	Numeric, the number of simulations of the study to run.
`test.type`	Character, one of 'proportions', 'count', 'crossover'. The type of test you would like to calculate the power for. If 'crossover' then `placebo.group` is ignored and `treatment.group` should contain curves constructed using one of the crossover options.

`sig.level`	Numeric between 0 and 1, the significance level used to test for a statistically significant difference between the groups.
`vitdcurves.placebo`	An object of class `vitd.curve`.
`vitdcurves.treatment`	An object of class `vitd.curve`.
`baseline`	Numeric, baseline prevalence of getting diseased at any exposure time. Probability of getting diseased when a participant has maximum vitamin D levels.
`RR`	Numeric, the relative risk of the study. The difference between getting diseased at highest and lowest vitamin D levels.
`rate`	Numeric, mean number of exposures per week, the rate at which exposures occur in the equivalent homogeneous Poisson process.
`intensity.func`	A function that takes time as sole argument and returns value between 0 and 1, input to nhpp function see poisson package.
`holding.time`	Numeric, the mean number of weeks for the holding time.
`mc.error`	Number of times to repeat the experiment at each `num.participants` value to explore Monte Carlo error.
`lohi.vit`	Inflection points of the relative risk curve used in `infection.count`.

Details

If `test.type` is equal to `crossover` then `vitdcurves.placebo` is ignored and `vitdcurves.treatment` should contain curves constructed using one of the crossover options. If `test.type` is equal to `proportions` and `num.participants` is small, warning messages from `prop.test` may appear.

Value

`power.calc` returns an object of class `power.calc` that is a list. The list has the following slots.

`test.type`	The type of study the power has been calculated on.
`baseline`	The baseline prevalence for disease.
`RR`	The relative risk of the study.
`Npergroup`	The number of participants per group in the study.
`MC.rep`	Number of repetitions of experiment to approximate Monte Carlo error.
`Power`	A `length(RR) x mc.error x length(Npergroup)` array of estimated power values.

Examples

```
# simulate placebo group
placebo <- vitd.curve( N = 10, type = "placebo" )

# simulate treatment group
treatment <- vitd.curve( N = 10, type = "dynamic-dose" )

# intensity function for exposures to infection
```

```
intensfun <- intensity.function( summer.rate = 0, winter.rate = 1 )

# calculate power: example only- run for much larger num.sims
pow <- power.calc( num.participants = c(10,20,30),
        num.sims = 10,  test.type = 'count',
        vitdcurves.placebo = placebo, vitdcurves.treatment = treatment,
        baseline = 0.03, RR = c(2,3,4), rate = 1, intensity.func = intensfun )

### NOT RUN ###
# approximate the Monte Carlo error in estimation of the power-- takes longer to run
#pow <- power.calc( num.participants = c(10,20,30),
#         snum.sims = 100, test.type = 'count',
#         vitdcurves.placebo = placebo, vitdcurves.treatment = treatment,
#         baseline = 0.03, RR = c(2,3,4), rate = 1,
#         intensity.func = intensfun, mc.error = 10 )

# plot power curves
plot( pow, xlab = "n", x.legend = 10, y.legend = 1, main.legend = "Relative Risk" )
```

`print.exposure.levels` *Print summary of an exposure.levels object.*

Description

Print a summary of a exposure.levels object.

Usage

```
  ## S3 method for class 'exposure.levels'
print( x, ... )
```

Arguments

`x`	An object of class `exposure.levels`.
`...`	Optional arguments to lower level functions.

`print.infection.count` *Print a summary of a infection.count object*

Description

Print a summary of a infection.count object.

Usage

```
  ## S3 method for class 'infection.count'
print( x, ... )
```

Arguments

`x`	An object of class `infection.count`.
`...`	Optional arguments to lower level functions.

`print.power.calc` *Summary of a power.calc object*

Description

Print a summary of a power.calc object.

Usage

```
  ## S3 method for class 'power.calc'
print( x, ... )
```

Arguments

`x`	An object of class `power.calc`.
`...`	Optional arguments to lower level functions.

`print.vitd.curve` *Print summary of a vitd.curve object*

Description

Print a summary of a vitd.curve object.

Usage

```
  ## S3 method for class 'vitd.curve'
print( x, ... )
```

Arguments

`x`	An object of class `vitd.curve`.
`...`	Optional arguments to lower level functions.

`pvalue.calc` *Generate a pvalue for a test of difference between two groups*

Description

Generates a pvalue to test for a statistically significant difference between two groups.

Usage

```
pvalue.calc( placebo.infection, treatment.infection, test.type )
```

Arguments

`placebo.infection`	An object of class `infection.count`.
`treatment.infection`	An object of class `infection.count`.
`test.type`	Character, one of 'proportions', 'crossover', 'count'. The type of test you would like to calculate a pvalue for.

Value

`pvalue.calc` returns a list with the summary from the test and the pvalue.

`rr.curve.plot` *Plotting the relative risk curve for vitamin D status.*

Description

Plot the relative risk curve for vitamin D with exposure points and whether a participant developed infection at that exposure time.

Usage

```
rr.curve.plot( expos, infect, main = NULL, xlab = "25 Hydroxy Vitamin D",
               ylab = "RR", col = "red", pch = 1, cex = 1 )
```

Arguments

`expos`	An object of class `exposure.levels`.
`infect`	An object of class `infection.count`.
`main`	Main title for the plot.
`xlab`	A title for the x-axis.
`ylab`	A title for the y-axis.
`col`	Character, value for the colour of the points.
`pch`	Integer, specifying symbol for the points.
`cex`	Numeric, value giving the amount by which plotting the points should be magnified.

`rr.profile.plot` *Plotting a vitamin D profile and relative risk curve for vitamin D.*

Description

Plot a vitamin D status profile for a single participant and the relative risk curve for vitamin D (with exposure times and whether a participant was infected at that exposure time) side by side.

Usage

```
rr.profile.plot( vitdcurves, expos, infect )
```

Arguments

`vitdcurves`	An object of class `vitd.curve`.
`expos`	An object of class `exposure.levels`.
`infect`	An object of class `infection.count`.

Examples

```
# simulate vitamin D profiles
profiles <- vitd.curve( N = 10, type = "placebo" )

# intensity function for exposures to infection
intensfun <- intensity.function( summer.rate=0.1, winter.rate=0.9 )

# simulate exposure times
expos <- exposure.levels( profiles, rate=2, intensfun, end=2 )

# simulate disease
infection <- infection.count( expos )

# plot
rr.profile.plot( profiles, expos, infection )
```

---

summary.exposure.levels

Summary of a vitd.curve object

Description

Print a summary of a exposure.levels object.

Usage

```
  ## S3 method for class 'exposure.levels'
summary( object, ... )
```

Arguments

`object`	An object of class `exposure.levels`.
`...`	Optional arguments to lower level functions.

---

summary.infection.count

Summary of a infection.count object

Description

Print a summary of a infection.count object.

Usage

```
  ## S3 method for class 'infection.count'
summary( object, ... )
```

Arguments

`object`	An object of class `infection.count`.
`...`	Optional arguments to lower level functions.

`summary.power.calc` *Summary of a power.calc object*

Description

Print a summary of a power.calc object.

Usage

```
  ## S3 method for class 'power.calc'
summary( object, ... )
```

Arguments

`object`	An object of class `power.calc`.
`...`	Optional arguments to lower level functions.

`summary.vitd.curve` *Summary of a vitd.curve object*

Description

Print a summary of a vitd.curve object.

Usage

```
  ## S3 method for class 'vitd.curve'
summary( object, ... )
```

Arguments

`object`	An object of class `vitd.curve`.
`...`	Optional arguments to lower level functions.

`vitd.curve` *Generate Vitamin D curves*

Description

Generates a vitamin D status profile curve for each individual in a group

Usage

```
vitd.curve( N, type = c("placebo","fixed-dose","dynamic-dose",
            "cross-placebo-fixed-dose", "cross-placebo-dynamic-dose"),
            start = 0, end = 2, cross = 0.5*(start + end), Min.Height = 10,
            Max.Height = 80, Flat.Height = 50, Spread.Min = 1,
       Spread.Max = 1, Spread.FH = 1, supp.dose = 20, north.hemi = TRUE, res = 40 )
```

Arguments

`N`	Integer, number of participants in the group.
`type`	Character, type of group, one of 'placebo', 'fixed-dose', 'dynamic-dose', 'cross-placebo-fixed-dose', 'cross-placebo-dynamic-dose'.
`start`	Integer, when the study started.
`end`	Integer, when the study ended.
`cross`	For crossover design, the fraction of the study before crossing over from `[start,end]`
`Min.Height`	Numeric, mean minimum height of curves.
`Max.Height`	Numeric, mean maximum height of curves.
`Flat.Height`	Numeric, threshold height of curves when treatment group.
`Spread.Min`	Numeric, how the minimum heights of participants are spread around the mean. 1 corresponds to narrowest spread, 100 corresponds to most scattered spread.
`Spread.Max`	Numeric, how the maximum heights of participants are spread around the mean. 1 corresponds to narrowest spread, 100 corresponds to most scattered spread.
`Spread.FH`	Numeric, how the flat heights of participants are spread around the mean. 1 corresponds to narrowest spread, 100 corresponds to most scattered spread.
`supp.dose`	Numeric, fixed dosage administered in the traditional supplementation scheme.
`north.hemi`	Logical argument. TRUE: to define summer/winter months as in the Northern Hemisphere. FALSE: to define summer/winter months as in the Southern Hemisphere.
`res`	Resolution parameter for plotting of vitamin D curves.

Value

`vitd.curve` returns an object of class `vitd.curve` that is a list. The list has the following slots.

`time`	The values of time in the study passed to `plot.vitd.curve` and `exposure.levels`.
`curve`	A list for input to `plot.vitd.curve` and `exposure.levels` including the following slots.
`outp`	Points to plot each individual's vitamin D curve.
`min.heights`	The minimum height for each individual's curve.
`max.heights`	The maximum height for each individual's curve.
`flatheights`	The threshold level for each individual. Only returned if Treatment type.
`min.height`	The mean minimum height for the group. Used as input to `power.calc`.
`max.height`	The mean amplitude for the group. Used as input to `power.calc`.
`flatheight`	The mean threshold level for the group. Used as input to `power.calc`.
`spread.min`	The group's spread around the minimum height. Used as input to `power.calc`.
`spread.max`	The group's spread around the maximum height. Used as input to `power.calc`.
`spread.fh`	The group's spread around the threshold level. Used as input to `power.calc`.
`type`	The type of group. One of 'Placebo', 'Traditional', 'Treatment'. Used as input to `power.calc`.
`supp.dose`	The fixed supplementation dose when traditional supplementation group.
`north.hemi`	Flag for which hemisphere the study is taking place in.

Index

`exposure.levels`, *3*, 3, *4*, *6*

`infection.count`, 4, *5*
`infection.count.plot`, 5
`intensity.function`, *6*, 6

`plot.exposure.levels`, 6
`plot.power.calc`, 7
`plot.vitd.curve`, 8
`power.calc`, 8, *9*
`print.exposure.levels`, 10
`print.infection.count`, 10
`print.power.calc`, 11
`print.vitd.curve`, 11
`pvalue.calc`, *12*, 12

`rr.curve.plot`, 12
`rr.profile.plot`, 13

`SimVitD-package`, 2
`summary.exposure.levels`, 14
`summary.infection.count`, 14
`summary.power.calc`, 15
`summary.vitd.curve`, 15

`vitd.curve`, *3*, 16, *17*