Package ‘SimRVPedigree’

October 12, 2022

Type Package
Title Simulate Pedigrees Ascertained for a Rare Disease
Version 0.4.4
Depends R (>= 3.4.0)
Imports kinship2 (>= 1.6.4), dplyr (>= 0.7.4), stats (>= 3.4.0), methods (>= 3.4.0)
Suggests doParallel (>= 1.0.11), doRNG (>= 1.6.6), graphics (>= 3.4.0), knitr (>= 1.17), rmarkdown (>= 1.8), roxygen2 (>= 6.0.1), testthat (>= 2.0.0)
License GNU General Public License
LazyData TRUE
RoxygenNote 6.1.1
VignetteBuilder knitr
Encoding UTF-8
NeedsCompilation no
Author Christina Nieuwoudt [aut, cre], Jinko Graham [aut]
Maintainer Christina Nieuwoudt <cnieuwou@sfu.ca>
Repository CRAN
Date/Publication 2020-02-09 21:00:06 UTC

R topics documented:

AgeSpecific_Hazards .......................................................... 2
censor_ped ................................................................. 3
EgPeds ................................................................. 4
AgeSpecific_Hazards

Description
A dataset that contains age-specific hazard rates to roughly mimic: (1) the age-specific hazard rates for lymphoid cancer onset in the United States, (2) the age-specific hazard rates for death in the United States, and (3) the age-specific hazard rates for death for individuals, living in the United States, who have been diagnosed with a lymphoid cancer.

Usage
data(AgeSpecific_Hazards)

Format
A data frame with 100 rows and 3 variables:

- **pop_onset_hazard** The age-specific population hazard rate for lymphoid cancer
- **unaffected_death_hazard** The age-specific hazard rate for death in the unaffected population
- **affected_death_hazard** The age-specific hazard rate for death in the affected population

Details
The AgeSpecific_Hazards dataset contains age-specific hazard rates which roughly mimic: (1) the age-specific hazard rates for lymphoid cancer onset in the United States, (2) the age-specific hazard rates for death in the United States, and (3) the age-specific hazard rates for death for individuals, living in the United States, who have been diagnosed with a lymphoid cancer. The age-specific hazard rates of lymphoid cancer onset and death in the affected population may be estimated by a program such as the Surveillance, Epidemiology, and End Results Program (SEER), and the age-specific hazard rates of death in the United States may be estimated from actuarial life tables provided by the Social Security Administration.
The three columns in the AgeSpecific_Hazards dataset provide age-specific hazard rates, in yearly increments, beginning at age 0 and ending with age 100. That is, the values in the first row describe the hazard rates for an individual whose age is contained in the interval \([0, 1)\), while the values in the second row describe the hazard rates for an individual whose age is contained in the interval \([1, 2)\), and so on.

References

The Surveillance, Epidemiology, and End Results (SEER) Program. [https://seer.cancer.gov/](https://seer.cancer.gov/)


censor_ped

**Description**

censor_ped censors a pedigree of any information that occurs after a specified year.

**Usage**

censor_ped(ped_file, censor_year = NULL)

**Arguments**

- **ped_file**  
  An object of class ped. A pedigree generated by `sim_ped` or `sim_RVped`, or an object created by the function `new.ped`. See details.

- **censor_year**  
  Numeric. The censor year. If not supplied, defaults to the year the pedigree was ascertainment, i.e. the proband’s onset year. See details.

**Details**

Upon supplying a pedigree and a censor year the `censor_ped` function will remove all individuals born after censor_year and censor all disease onset and death events after the censor_year.

Users who wish to use `censor_ped` for pedigrees not generated by `sim_ped` or `sim_RVped` must use `new.ped` to create an object of class ped. When creating the ped object please provide as much relevant date information as possible, i.e. years of birth, onset, and death. When present please specify a proband as described in `new.ped`.

By default, censor_year is set to the year that the pedigree is ascertainment, i.e. the year the proband experienced disease onset. However, if ped_file does not contain the proband identification variable the user must supply a value for censor_year.

**Value**

The censored pedigree.
See Also
   new.ped

Examples

#Read in age-specific harard data and create hazard object.
data(AgeSpecific_Hazards)
haz_obj <- hazard(hazardDF = AgeSpecific_Hazards)

#Simulate a pedigree ascertained for multiple affecteds
set.seed(3)
RVped2015 <- sim_RVped(hazard_rates = haz_obj,
    num_affected = 2,
    ascertain_span = c(1900, 2015),
    GRR = 30, carrier_prob = 0.002,
    RVfounder = TRUE,
    stop_year = 2015,
    recall_probs = c(1),
    founder_byears = c(1900, 1905),
    FamID = 1)[[2]]

# Plot the 2015 pedigree
plot(RVped2015)
mtext(side = 3, line = 2, "Reference Year: 2015")

# Censor RVped2015 after 1960
RVped1960 <- censor_ped(ped_file = RVped2015, censor_year = 1960)

# Plot the 1960 pedigree
plot(RVped1960)
mtext(side = 3, line = 2, "Reference Year: 1960")

---

**EgPeds**

**Example pedigrees**

Description

A dataset containing five example pedigrees.

Usage

data(EgPeds)

Format

A data frame with 65 rows and 14 variables:

- **FamID**  Family identification number
**find_mrca**

Find the most recent common ancestor of two pedigree members

**Description**

Find the most recent common ancestor of two pedigree members

**Usage**

`find_mrca(ped, ID1, ID2)`

**Arguments**

- `ped` : A ped object
- `ID1` : The ID of the first relative
- `ID2` : The ID of the second relative

**Value**

The ID of the common ancestor
Examples

library(SimRVPedigree)
data(AgeSpecific_Hazards)

set.seed(5)
ex_ped <- sim_ped(hazard_rates = hazard(hazardDF = AgeSpecific_Hazards),
                  GRR = 10, FamID = 1,
                  founder_byears = c(1800, 1900),
                  stop_year = 2020)

plot(ex_ped)

# Find most recent common ancestor of individuals with IDs 19 and 21
find_mrca(ped = ex_ped, ID1 = 19, ID2 = 21)

# Note that someone can be their own most recent common ancestor.
# In the following example, since the individual with ID 8 is the grandmother
# of the individual with ID 21, the find_mrca function returns 8.
find_mrca(ped = ex_ped, ID1 = 8, ID2 = 21)

# For unrelated individuals, the find_mrca function returns NA
find_mrca(ped = ex_ped, ID1 = 8, ID2 = 15)
find_mrca(ped = ex_ped, ID1 = 5, ID2 = 4)

hazard

Create an object of class hazard.

Description

Create a hazard object, required input for sim_RVped, sim_ped, and sim_life functions.

Usage

hazard(hazardDF, partition = NULL, subtype_ID = NULL)

Arguments

hazardDF  Data.frame. A data.frame containing the age-specific hazard rate(s) of disease in the population of interest, the age-specific hazard rate for death in the unaffected population, and the age-specific hazard rate for death in the affected population. See details.

partition  Numeric vector. The partition of ages, in years, over which to apply the age-specific hazard rates in hazardDF. If not supplied, defaults to a partition that starts at 0 and increases in yearly increments. See details.

subtype_ID  List. If specifying a disease with multiple subtypes, a list of character subtype IDs. By default, subtype_ID = NULL, i.e. no subtypes to simulate.
Details

Users are permitted to specify hazard objects for two scenarios: (1) for a disease without subtypes or (2) for a disease with multiple subtypes.

When simulating a disease without subtypes, hazardDF must contain 3 columns that meet the following criteria:

**column 1:** age-specific hazard rates of *disease* for the population of interest

**column 2:** age-specific hazard rates of *death* for the *unaffected* population. If the disease of interest is sufficiently rare, so that death by the disease is rare, the user may choose to use the population, age-specific, hazard rates of death instead.

**column 3:** age-specific hazard rates of *death* for the *affected* population.

When simulating a disease with *n* disease subtypes, hazardDF must contain *n* + 2 columns that meet the following criteria:

**column 1:** age-specific hazard rates of *disease* for the first subtype of interest

**column 2:** age-specific hazard rates of *disease* for the second subtype of interest

... 

**column n:** age-specific hazard rates of *disease* for the *nth* subtype of interest

**column n + 1:** age-specific hazard rates of *death* for the *unaffected* population. If the disease of interest is sufficiently rare, so that death by the disease is rare, the user may choose to use the population, age-specific, hazard rates of death instead.

**column n + 2:** age-specific hazard rates of *death* for an individual affected by any of the *n* subtypes.

Users must provide partition in years; e.g. a hazard rate for a baby between 6 months and 1 year of age should have lower bound 0.5 years and an upper bound 1 year. Additionally, partition must apply to all of the age-specific hazard rates in hazardDF.

Value

An object of class hazard.

Examples

```r
# Specifying the hazard rates for a disease
# with only one subtype (or grouped subtypes).
data(AgeSpecific_Hazards)

head(AgeSpecific_Hazards)
nrow(AgeSpecific_Hazards)

my_HR <- hazard(hazardDF = AgeSpecific_Hazards)
class(my_HR)
head(my_HR[[1]])

#NOTE: since partition was not supplied, the partition has been assumed to
# start at 0 and increase in yearly increments.
my_HR[[2]]
```
new.ped

Create an object of class ped.

Description

Create an object of class ped, from a data.frame, required input for reassign_gen, censor_ped, and trim_ped functions.

Usage

new.ped(ped_file)

Arguments

ped_file Data.frame. A pedigree, see details.

Details

The data frame supplied to new.ped, ped_file, must contain the following variables:

<table>
<thead>
<tr>
<th>name</th>
<th>type</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FamID</td>
<td>numeric</td>
<td>family identification number</td>
</tr>
<tr>
<td>ID</td>
<td>numeric</td>
<td>individual identification number</td>
</tr>
<tr>
<td>dadID</td>
<td>numeric</td>
<td>identification number of father</td>
</tr>
<tr>
<td>momID</td>
<td>numeric</td>
<td>identification number of mother</td>
</tr>
<tr>
<td>sex</td>
<td>numeric</td>
<td>gender identification; if male sex = 0, if female sex = 1</td>
</tr>
<tr>
<td>affected</td>
<td>logical</td>
<td>disease-affection status:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>affected = TRUE if affected by disease, and FALSE otherwise,</td>
</tr>
</tbody>
</table>
 Optionally, `ped_file` may contain any of the following variables:

<table>
<thead>
<tr>
<th>name</th>
<th>type</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>available</td>
<td>logical</td>
<td>availability status; available = TRUE if available, and FALSE otherwise.</td>
</tr>
<tr>
<td>DA1</td>
<td>numeric</td>
<td>paternally inherited allele at the assumed disease locus: DA1 = 1 if rare variant is present, and 0 otherwise</td>
</tr>
<tr>
<td>DA2</td>
<td>numeric</td>
<td>maternally inherited allele at the assumed disease locus: DA2 = 1 if rare variant is present, and 0 otherwise</td>
</tr>
<tr>
<td>birthYr</td>
<td>numeric</td>
<td>the individual's birth year</td>
</tr>
<tr>
<td>onsetYr</td>
<td>numeric</td>
<td>the individual’s year of disease onset, when applicable, otherwise NA</td>
</tr>
<tr>
<td>deathYr</td>
<td>numeric</td>
<td>the individual’s year of death, when applicable, otherwise NA</td>
</tr>
<tr>
<td>RR</td>
<td>numeric</td>
<td>the individual’s relative-risk of disease</td>
</tr>
<tr>
<td>Gen</td>
<td>numeric</td>
<td>the individual’s generation number relative to the eldest founder. For the eldest founder Gen = 1, for his or her offspring Gen = 2, etc.</td>
</tr>
<tr>
<td>proband</td>
<td>logical</td>
<td>proband identifier: proband = TRUE if individual is the proband, and FALSE otherwise.</td>
</tr>
<tr>
<td>subtype</td>
<td>character</td>
<td>the individual’s disease subtype, when applicable, otherwise NA</td>
</tr>
</tbody>
</table>

We note that some of the optional fields above may be required for various `ped` functions

Value

An object of class ped.

Examples

```r
data(EgPeds)
head(EgPeds)

ped1 = new.ped(EgPeds[EgPeds$FamID == 1, ])
head(ped1, n = 3)
class(ped1)
summary(ped1)

AllPeds = new.ped(EgPeds)
head(AllPeds)
class(AllPeds)
summary(AllPeds)
```

Description

Create a kinship2 pedigree structure from an object of class ped
Usage

`ped2pedigree(x)`

Arguments

`x`  
A ped object.

Value

A pedigree object. See `pedigree` for details.

References


---

### plot.ped

Plot pedigree

#### Description

Plot pedigree

#### Usage

```r
## S3 method for class 'ped'
plot(x, ref_year = NULL, gen_lab = FALSE, plot_legend = TRUE, location = "topleft", radius = 0.2, density = c(-1, 35, 55), angle = c(90, 65, 40), gen_stretch = 2, cex = 1, adj = 1, line = 2, mar = c(5.1, 4.1, 4.1, 2.1), ...)
```

Arguments

- `x`  
An object of class ped.
- `ref_year`  
When provided, the reference year for age labels. Users may supply a (numeric) year which will create age labels at the specified year. Alternatively, users may set `ref_year = "ascYR"`, which will create age lables for the year the pedigree was ascertained, when ascertained. By default, `ref_year = NULL` and no age labels are created.
- `gen_lab`  
Logical. Should generation labels be printed in the margin. By default, FALSE.
- `plot_legend`  
Logical. Should legend for symbol shading be plotted. By default, TRUE.
- `location`  
The location for the pedigree legend, as in `pedigree.legend`. Options include: "topleft", "topright", "bottomright", or "bottomleft". By default, location = "topleft".
- `radius`  
The radius size for the pedigree legend, as in `pedigree.legend`. By default, radius = 0.2.
- `...`  
Optional arguments.
density  The density of shading in plotted symbols, as in plot.pedigree. By default, density = c(-1, 35, 55).

angle  The angle of shading in plotted symbols, as in plot.pedigree. By default, angle = c(90, 65, 40).

gen_stretch Numeric. Used to stretch the spacing between generation labels. By default, gen_stretch = 2. Increase for more space between labels, decrease for less space.

cex  The text size. By default, cex = 1.

adj  When ref_year is supplied, used to adjust position of reference year, as in mtext. By default, adj = 1.

line  When ref_year is supplied, used to adjust position of reference year, as in mtext. By default, line = 2.

mar  The sizes for plot margins, as in par.

...  Extra options that feed to plot.pedigree, or plot.

References


See Also

plot.pedigree, pedigree.legend, plot, par

Examples

#Read in age-specific harard data and create hazard object.
data(AgeSpecific_Hazards)
haz_obj <- hazard(hazardDF = AgeSpecific_Hazards)

#Simulate a pedigree ascertained for multiple affecteds
set.seed(2)
RVped2015 <- sim_RVped(hazard_rates = haz_obj,
                        num_affected = 2,
                        ascertain_span = c(1900, 2015),
                        GRR = 30, carrier_prob = 0.002,
                        RVfounder = TRUE,
                        stop_year = 2015,
                        recall_probs = c(1),
                        founder_byears = c(1900, 1905),
                        FamID = 1)[[2]]

summary(RVped2015)

#plot pedigree without age labels
plot(RVped2015)

#plot pedigree with age labels, set the
#reference year to be the ascertainment year
The `reassign_gen` function assigns generation numbers among affected family members so that generation 1 represents the generation of the most recent common ancestor of all disease-affected relatives. We note that the individual in generation 1 could themselves be disease-affected, i.e. an individual can be considered their own ancestor.

For example, consider a family with 2 affected members. If the disease-affected relatives are a parent and a child, the affected parent would be assigned generation 1, and the affected child generation
2. However, if the disease-affected relatives are a pair of siblings, each is be assigned generation 2 since a common parent of the two is assumed to be a carrier of a latent susceptibility variant. Similarly, if the disease-affected relatives are a pair of cousins, is assigned generation 3, since a common grandparent is the most recent common ancestor from whom they could have inherited a shared variant associated with the disease.

Users who wish to assign generation number based on affection status in pedigrees that have not been simulated with the SimRVpedigree package must create a ped object using `new.ped`.

Value

A ped object containing only affected members, obligate carriers, and founders with generation numbers reassigned among disease-affected relatives based on their most recent common ancestor, as described in details.

References


See Also

`new.ped`

Examples

```r
# Read in example pedigrees
data(EgPeds)
class(EgPeds)

# Create ped object
Bpeds <- new.ped(EgPeds)
summary(Bpeds)

# Reassign generation numbers in the first four pedigrees in EgPeds
Apeds <- lapply(seq_len(5), function(x){
    reassign_gen(Bpeds[Bpeds$FamID == x, ]))
Apeds <- do.call(rbind, Apeds)

# Compare pedigrees before and after reassigning
gen number based on affected status
par(mfrow = c(1, 2))
for (k in 1:5) {
    plot(subset(Bpeds, FamID == k), gen_lab = TRUE, plot_legend = FALSE)
    mtext(paste0("Ped", k, ": before generation reassignment", sep = ""),
          side = 3, line = 1.5)

    plot(subset(Apeds, FamID == k), gen_lab = TRUE, plot_legend = FALSE)
    mtext(paste0("Ped", k, ": after generation reassignment", sep = ""),
```
SimRVPedigree

Simulate pedigrees ascertained for disease status

Description

The SimRVPedigree package provides methods to simulate and manipulate pedigrees ascertained to contain multiple relatives affected by a rare disease.

Details

Family-based studies to identify genetic susceptibility factors associated with rare diseases are regaining traction. The resurgence in popularity is due to the fact that family-based studies have more power to detect rare variants, require smaller sample sizes, and can more accurately detect sequencing errors than case-control studies, Wijsman (2012). However, identifying a suitable number of families for analysis can require years of continued collaboration between researchers and clinicians. As a result, collecting new data to replicate findings or evaluate methodology is impractical. The SimRVPedigree package aims to address this problem by providing a platform to randomly simulate families ascertained to contain multiple relatives affected by a rare disease. The distinguishing feature of the SimRVPedigree package is that it aims to mimic the process of family development, while allowing users to incorporate multiple facets of family ascertainment.

References


Ellen M. Wijsman (2012). The role of large pedigrees in an era of high-throughput sequencing. Human Genetetics, 131, 1555-1563

description

Primarily intended as an internal function, sim_life simulates all life events for an individual starting at birth, age 0, and ending with death or the end of the study.

Usage

sim_life(hazard_rates, GRR, carrier_prob, RV_status, YOB, stop_year, NB_params = c(2, 4/7), fert = 1)
Arguments

- **hazard_rates**: An object of class `hazard`, created by `hazard`. The genetic relative-risk of disease, i.e. the relative-risk of disease for individuals who carry at least one copy of the causal variant. Note: When simulating diseases with multiple subtypes `GRR` must contain one entry for each simulated subtype. See details.

- **GRR**: Numeric. The carrier probability for all causal variants with relative-risk of disease `GRR`. By default, `carrier_prob = 0.002`

- **carrier_prob**: Numeric. The carrier probability for all causal variants with relative-risk of disease `GRR`. By default, `carrier_prob = 0.002`

- **RV_status**: Numeric. `RV_status = TRUE` if the individual is a carrier of a rare variant that increases disease susceptibility, and `FALSE` otherwise.

- **YOB**: A positive number. The individual’s year of birth.

- **stop_year**: Numeric. The last year of study. If not supplied, defaults to the current year.

- **NB_params**: Numeric vector of length 2. The size and probability parameters of the negative binomial distribution used to model the number of children per household. By default, `NB_params = c(2, 4/7)`, due to the investigation of Kojima and Kelleher (1962).

- **fert**: Numeric. A constant used to rescale the fertility rate after disease-onset. By default, `fert = 1`.

Details

Starting at birth, age 0, `sim_life` generates waiting times to reproduction, onset, and death. The event with the shortest waiting time is chosen as the next life event, and the individual’s age is updated by the waiting time of the winning event. Conditioned on the individual’s new age, this process is applied recursively, until death or until the end of the study is reached.

We make the following assumptions regarding the simulation of waiting times:

1. We assume that, given an individual’s current age, their time to disease onset is the waiting time in a non-homogeneous Poisson process with an age-specific hazard rate that follows a proportional hazards model. In this model, individuals who have NOT inherited the rare variant experience disease onset according to the baseline (or population) hazard rate of disease. On the other hand, individuals who have inherited the rare variant are assumed to have an increased risk of disease onset relative to those who have inherited it. The user is expected to supply the baseline hazard rate of disease, as well as the relative-risk of disease for genetic cases. Additionally, we impose the restriction that individuals may only experience disease onset once, and remain affected from that point on.

2. We assume that, given an individual’s current age, their time to death is the waiting time in a non-homogeneous Poisson process with age-specific hazard rate determined by their affection status. We assume that disease-affected individuals experience death according to the age-specific hazard rate for death in the affected population. On the other hand, we assume that unaffected individuals experience death according to the age-specific hazard rate for death in the unaffected population. If the disease of interest is sufficiently rare, the user may choose to substitute the population age-specific hazard rate for death for the aforementioned age-specific hazard rate for death in the unaffected population. The user is expected to supply age-specific hazard rates of death for both the affected and unaffected populations.
3. We assume that, given an individual’s current age, their time to reproduction is the waiting time in a homogeneous Poisson process. That is, we assume that individuals reproduce at uniform rate during their reproductive years. For example, one’s reproductive years may span from age 20 to age 35 years. To mimic observed age-specific fertility data, the birth range for an individual is simulated as follows: first we sample the lower bound uniformly from ages 16 to 27, next we sample the range of the birth span uniformly from 10 to 18 years and add this value to the lower bound to determine the upper bound of the birth range. We do not allow for offspring to be produced outside of an individual’s simulated reproductive birth span.

The events simulated by sim_life are labelled as follows:

- "Start" the individual’s year of birth.
- "Child" a reproductive event, i.e. creation of offspring
- "Onset" disease onset event,
- "Death" death event

Value

an object of class event. An object of class event is a list that contains the following items.

life_events A named numeric vector of life events, see details.
repro_events A vector of reproduction years, that is the year(s) that the individual produces offspring. When the individual does not reproduce repro_events = NULL.
onset_event Numeric. When applicable the year of disease-onset. When onset does not occur onset_event = NA
death_event Numeric. When applicable the year of death. When death is censored death_event = NA
subtype Character. When applicable, the disease subtype.
censor_year Numeric. When applicable the last year that data was observed. Note: after death censor_year = NA

References


Examples

data(AgeSpecific_Hazards)
my_HR <- hazard(hazardDF = AgeSpecific_Hazards)

# The following commands simulate all life events for an individual who
# has NOT inherited a causal variant, born in 1900. From the output, this
# individual has two children, one in 1921 and another in 1923, and then
# dies in 1987.
set.seed(135)
sim_life(hazard_rates = my_HR, GRR = 10,
carrier_prob = 0.002,
RV_status = FALSE,
YOB = 1900, stop_year = 2000)

# Using the same random seed, notice how life events can vary for
# someone who has inherited the causal variant, which carries a
# relative-risk of 10. From the output, this individual also has
# two children, but then experiences disease onset in 1974,
# and dies in 1976.
set.seed(135)
sim_life(hazard_rates = my_HR, GRR = 10,
carrier_prob = 0.002,
RV_status = TRUE,
YOB = 1900, stop_year = 2000)

# The following commands simulate life events for an individual who
# has inherited a causal rare variant, with relative risk 100 for
# the subtype "HL" and no increased relative risk for the subtype "NHL".
set.seed(1)
sim_life(hazard_rates = hazard(SubtypeHazards,
    subtype_ID = c("HL", "NHL")),
    GRR = c(100, 1),
carrier_prob = 0.002,
RV_status = TRUE,
YOB = 1900, stop_year = 2000)

---

**sim_ped**  
*Simulate a pedigree*

**Description**

Please note the distinction between `sim_ped` and `sim_RVped`. Pedigrees simulated using `sim_ped` do not account for study design. To simulate a pedigree ascertained to contain multiple family members affected by a disease please use `sim_RVped`.

**Usage**

```r
sim_ped(hazard_rates, GRR, FamID, founder_byears, stop_year = NULL, 
carrier_prob = 0.002, RVfounder = FALSE, NB_params = c(2, 4/7), 
fert = 1)
```

**Arguments**

- `hazard_rates` An object of class hazard, created by `hazard`.  

**GRR**

Numeric. The genetic relative-risk of disease, i.e. the relative-risk of disease for individuals who carry at least one copy of the causal variant. Note: When simulating diseases with multiple subtypes GRR must contain one entry for each simulated subtype. See details.

**FamID**

Numeric. The family ID to assign to the simulated pedigree.

**founder_byears**

Numeric vector of length 2. The span of years from which to simulate, uniformly, the birth year for the founder who introduced the rare variant to the pedigree.

**stop_year**

Numeric. The last year of study. If not supplied, defaults to the current year.

**carrier_prob**

Numeric. The carrier probability for all causal variants with relative-risk of disease GRR. By default, carrier_prob = 0.002

**RVfounder**

Logical. Indicates if all pedigrees segregate the rare, causal variant. By default, RVfounder = FALSE See details.

**NB_params**

Numeric vector of length 2. The size and probability parameters of the negative binomial distribution used to model the number of children per household. By default, NB_params = c(2, 4/7), due to the investigation of Kojima and Kellerher (1962).

**fert**

Numeric. A constant used to rescale the fertility rate after disease-onset. By default, fert = 1.

**Details**

To introduce the rare variant to the pedigree, we allow users to choose from one of the following two assumptions:

1. Assume that the variant is rare enough that a single copy has been introduced by one founder, and begin the simulation of the pedigree with this founder, as in Bureau (2014).

2. Simulate the starting founder’s rare-variant status with probability equal to the carrier probability of the rare variant in the population. We note that under this setting pedigrees may not segregate the rare variant.

The sim_ped function starts simulating the pedigree by generating the birth year for the starting founder, uniformly between the years specified by founder_byears. Next, all life events are simulated for the founder via sim_life. Possible life events include: reproduction, disease onset, and death. We only allow disease onset to occur once, i.e. no remission. Computationally, this implies that after disease onset, the waiting time to death is always simulated using the age-specific mortality rates for the affected population. Life events for individuals who have inherited the rare variant are simulated such that their relative-risk of disease is GRR, according to a proportional hazards model. The relative-risk of disease onset for individuals who have not inherited the causal variant is assumed to be 1. Any life events that occur after stop_year are censored.

When segregating in the pedigree, the rare variant is transmitted from parent to offspring according to Mendel’s laws. The process of simulating life events is repeated for any offspring that are produced before stop_year.

**Value**

The simulated pedigree.
See Also

sim_RVped, sim_life

References


Alexandre Bureau, Samuel G. Younkin, Margaret M. Parker, Joan E. Bailey-Wilson, Mary L. Marazita, Jeffrey C. Murray, Elisabeth Mangold, Hasan Albacha-Hejazi, Terri H. Beaty, and Ingo Ruczinski (2014). Inferring rare disease risk variants based on exact probabilities of sharing by multiple affected relatives. Bioinformatics; Vol. 30, No. 15, pp. 2189-2196.

Examples

data(AgeSpecific_Hazards)

# Simulate a random pedigree
set.seed(5)
ex_ped <- sim_ped(hazard_rates = hazard(hazardDF = AgeSpecific_Hazards),
GRR = 10,
FamID = 1,
founder_byears = c(1900, 1910),
stop_year = 2015)

# View the simulated pedigree
ex_ped

# Plot the pedigree
plot(ex_ped, location = "topleft")

# Plot the pedigree, this time with age labels for # all descendents of the starting founder (ID 1)
plot(ex_ped, ref_year = 2015,
cex= 0.75, symbolsize = 1.25,
location = "topleft")

# Simulate a random pedigree. This time set RVfounder to TRUE so that # the eldest introduces a causal rare variant with probability 1.
set.seed(5)
ex_ped <- sim_ped(hazard_rates = hazard(hazardDF = AgeSpecific_Hazards),
RVfounder = TRUE,
GRR = 10,
FamID = 1,
founder_byears = c(1900, 1910),
stop_year = 2015)
# Plot the pedigree with age labels
plot(ex_ped, ref_year = 2015,
     cex= 0.75, symbolsize = 1.25,
     location = "topleft")

sim_RVped  
Simulate a pedigree ascertained to contain multiple disease-affected relatives

Description

sim_RVped simulates a pedigree ascertained to contain multiple affected members, selects a proband, and trims the pedigree to contain only those individuals that are recalled by the proband.

Usage

sim_RVped(hazard_rates, GRR, num_affected, ascertain_span, FamID,
          founder_byears, stop_year = NULL, recall_probs = NULL,
          carrier_prob = 0.002, RVfounder = FALSE, NB_params = c(2, 4/7),
          fert = 1, first_diagnosis = NULL, sub_criteria = NULL)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hazard_rates</td>
<td>An object of class hazard, created by hazard.</td>
</tr>
<tr>
<td>GRR</td>
<td>Numeric. The genetic relative-risk of disease, i.e. the relative-risk of disease for individuals who carry at least one copy of the causal variant. Note: When simulating diseases with multiple subtypes GRR must contain one entry for each simulated subtype. See details.</td>
</tr>
<tr>
<td>num_affected</td>
<td>Numeric vector. The minimum number of disease-affected relatives required for ascertainment.</td>
</tr>
<tr>
<td>ascertain_span</td>
<td>Numeric vector of length 2. The year span of the ascertainment period. This period represents the range of years during which the proband developed disease and the family would have been ascertained for multiple affected relatives.</td>
</tr>
<tr>
<td>FamID</td>
<td>Numeric. The family ID to assign to the simulated pedigree.</td>
</tr>
<tr>
<td>founder_byears</td>
<td>Numeric vector of length 2. The span of years from which to simulate, uniformly, the birth year for the founder who introduced the rare variant to the pedigree.</td>
</tr>
<tr>
<td>stop_year</td>
<td>Numeric. The last year of study. If not supplied, defaults to the current year.</td>
</tr>
<tr>
<td>recall_probs</td>
<td>Numeric. The proband’s recall probabilities for relatives, see details. If not supplied, the default value of four times kinship coefficient between the proband and the relative is used.</td>
</tr>
<tr>
<td>carrier_prob</td>
<td>Numeric. The carrier probability for all causal variants with relative-risk of disease GRR. By default, carrier_prob = 0.002</td>
</tr>
</tbody>
</table>
**Details**

When RV_founder = TRUE, all simulated pedigrees will segregate a genetic susceptibility variant. In this scenario, we assume that the variant is rare enough that it has been introduced by one founder, and we begin the simulation of the pedigree with this founder. Alternatively, when RV_founder = FALSE we simulate the starting founder's causal variant status with probability carrier_prob. When RV_founder = FALSE pedigrees may not segregate the genetic susceptibility variant. The default selection is RV_founder = FALSE. Additionally, we note that sim_RVpedigree is intended for rare causal variants; users will receive a warning if carrier_prob > 0.002.

We note that when GRR = 1, pedigrees do not segregate the causal variant regardless of the setting selected for RVfounder. When the causal variant is introduced to the pedigree we transmit it from parent to offspring according to Mendel’s laws.

When simulating diseases with multiple subtypes GRR is a numeric list indicating the genetic-relative risk for each subtype specified in the hazard object supplied to hazard_rates. For example, for a disease with two disease subtypes, if we set GRR = c(20, 1) individuals who inherit the causal variant are 20 times more likely than non-carriers to develop the first subtype and as likely as non-carriers to develop the second subtype.

We begin simulating the pedigree by generating the year of birth, uniformly, between the years specified in founder_byears for the starting founder. Next, we simulate this founder’s life events using the sim_life function, and censor any events that occur after the study stop_year. Possible life events include: reproduction, disease onset, and death. We continue simulating life events for any offspring, censoring events which occur after the study stop_year, until the simulation process terminates. We do not simulate life events for marry-ins, i.e. individuals who mate with either the starting founder or offspring of the starting founder.

We do not model disease remission. Rather, we impose the restriction that individuals may only experience disease onset once, and remain affected from that point on. If disease onset occurs then we apply the hazard rate for death in the affected population.

sim_RVped will only return ascertained pedigrees with at least num_affected affected individuals. That is, if a simulated pedigree does not contain at least num_affected affected individuals
sim_RVped will discard the pedigree and simulate another until the condition is met. We note that even for \( \text{num}_\text{affected} = 2 \), sim_RVped can be computationally expensive. To simulate a pedigree with no proband, and without a minimum number of affected members use \text{sim_ped} instead of sim_RVped.

When simulating diseases with multiple subtypes, users may wish to apply additional ascertainment criteria using the \text{sub_criteria} argument. When supplied, this argument allows users to impose numeric subtype-specific ascertainment criteria. For example, if \( \text{sub_criteria} = \text{list}("\text{HL}", 1) \) then at least 1 of the \( \text{num}_\text{affected} \) disease-affected relatives must be affected by subtype "HL" for the pedigree to be ascertained. We note that the first entry of \text{sub_criteria}, i.e. the subtype label, must match the one of subtype labels in the hazards object supplied to \text{hazard_rates}. See examples.

Upon simulating a pedigree with \( \text{num}_\text{affected} \) individuals, sim_RVped chooses a proband from the set of available candidates. Candidates for proband selection must have the following qualities:

1. experienced disease onset between the years specified by \text{ascertain_span},
2. if less than \( \text{num}_\text{affected} - 1 \) individuals experienced disease onset prior to the lower bound of \text{ascertain_span}, a proband is chosen from the affected individuals, such that there were at least \( \text{num}_\text{affected} \) affected individuals when the pedigree was ascertained through the proband.

We allow users to specify the first year that reliable diagnoses can be made using the argument \text{first_diagnosis}. All subjects who experience disease onset prior to this year are not considered when ascertaining the pedigree for a specific number of disease-affected relatives. By default, \text{first_diagnosis} = \text{NULL} so that all affected relatives, recalled by the proband, are considered when ascertaining the pedigree.

After the proband is selected, the pedigree is trimmed based on the proband’s recall probability of his or her relatives. This option is included to model the possibility that a proband either cannot provide a complete family history or that they explicitly request that certain family members not be contacted. If \text{recall_probs} is missing, the default values of four times the kinship coefficient, as defined by Thompson, between the proband and his or her relatives are assumed. This has the effect of retaining all first degree relatives with probability 1, retaining all second degree relatives with probability 0.5, retaining all third degree relatives with probability 0.25, etc. Alternatively, the user may specify a list of length \( l \), such that the first \( l - 1 \) items represent the respective recall probabilities for relatives of degree 1, 2, ..., \( l - 1 \) and the \( l^{th} \) item represents the recall probability of a relative of degree \( l \) or greater. For example, if \( \text{recall_probs} = \text{c}(1, 0.75, 0.5) \), then all first degree relatives (i.e. parents, siblings, and offspring) are retained with probability 1, all second degree relatives (i.e. grandparents, grandchildren, aunts, uncles, nieces and nephews) are retained with probability 0.75, and all other relatives are retained with probability 0.5. To simulate fully ascertained pedigrees, simply specify \( \text{recall_probs} = \text{c}(1) \).

In the event that a trimmed pedigree fails the \( \text{num}_\text{affected} \) condition, sim_RVped will discard that pedigree and simulate another until the condition is met. For this reason, the values specified for \text{recall_probs} affect computation time.

**Value**

A list containing the following data frames:

- \text{full_ped} The full pedigree, prior to proband selection and trimming.
ascertained_ped

The ascertained pedigree, with proband selected and trimmed according to proband recall probability. See details.

See Also

sim_ped, trim_ped, sim_life

References


Examples

#Read in age-specific hazards
data(AgeSpecific_Hazards)

#Simulate pedigree ascertained for multiple affected individuals
set.seed(2)
ex_RVped <- sim_RVped(hazard_rates = hazard(hazardDF = AgeSpecific_Hazards),
GRR = 20,
RVfounder = TRUE,
FamID = 1,
founder_byears = c(1900, 1905),
ascertain_span = c(1995, 2015),
num_affected = 2,
stop_year = 2017,
recall_probs = c(1, 1, 0))

# Observe: ex_RVped is a list containing two ped objects
summary(ex_RVped)

# The first is the original pedigree prior
# to proband selection and trimming
plot(ex_RVped[[1]])

# The second is the ascertained pedigree which
# has been trimmed based on proband recall
plot(ex_RVped[[2]])
summary(ex_RVped[[2]])

# NOTE: by default, RVfounder = FALSE.
# Under this setting pedigrees segregate a causal
# variant with probability equal to carrier_prob.
# Simulate Pedigrees with Multiple Disease Subtypes #
# Simulating pedigrees with multiple subtypes
# Import subtype-specific hazards rates for Hodgkin's lymphoma and non-Hodgkin's lymphoma
# data(SubtypeHazards)
head(SubtypeHazards)

my_hazards <- hazard(SubtypeHazards,
  subtype_ID = c("HL", "NHL"))

# Simulate pedigree ascertained for at least two individuals
# affected by either Hodgkin's lymphoma or non-Hodgkin's lymphoma.
# Set GRR = c(20, 1) so that individuals who carry a causal variant
# are 20 times more likely than non-carriers to develop "HL" but have
# same risk as non-carriers to develop "NHL".
set.seed(45)
ex_RVped <- sim_RVped(hazard_rates = my_hazards,
  GRR = c(20, 1),
  RVfounder = TRUE,
  FamID = 1,
  founder_byears = c(1900, 1905),
  ascertain_span = c(1995, 2015),
  num_affected = 2,
  stop_year = 2017,
  recall_probs = c(1, 1, 0))

plot(ex_RVped[[2]], cex = 0.6)

# Note that we can modify the ascertainment criteria so that
# at least 1 of the two disease-affected relatives are affected by
# the "HL" subtype by supplying c("HL", 1) to the sub_criteria
# argument.
set.seed(69)
ex_RVped <- sim_RVped(hazard_rates = my_hazards,
  GRR = c(20, 1),
  RVfounder = TRUE,
  FamID = 1,
  founder_byears = c(1900, 1905),
  ascertain_span = c(1995, 2015),
  num_affected = 2,
  stop_year = 2017,
  recall_probs = c(1, 1, 0),
  sub_criteria = list("HL", 1))

plot(ex_RVped[[2]], cex = 0.6)
**Description**

A dataset that contains the following age-specific hazard rates: (1) the age-specific hazard rates for Hodgkin Lymphoma and Non-Hodgkin Lymphoma in the United States, (2) the age-specific hazard rates for death in the United States, and (3) the age-specific hazard rates for death for individuals, living in the United States, who have been diagnosed with either Hodgkin Lymphoma and Non-Hodgkin Lymphoma.

**Usage**

```r
data(SubtypeHazards)
```

**Format**

A data frame with 100 rows and 4 variables:

- `pop_HL_hazard` The population, age-specific hazard rate for Hodgkin Lymphoma
- `pop_NHL_hazard` The population, age-specific hazard rate for Non-Hodgkin Lymphoma
- `unaffected_death_hazard` The age-specific hazard rate for death in the unaffected population
- `affected_death_hazard` The age-specific hazard rate for death in the affected population

**Details**

The `SubtypeHazards` dataset contains the following age-specific hazard rates which roughly mimic: (1) the age-specific hazard rates for Hodgkin Lymphoma and Non-Hodgkin Lymphoma in the United States, (2) the age-specific hazard rates for death in the United States, and (3) the age-specific hazard rates for death for individuals, living in the United States, who have been diagnosed with either Hodgkin Lymphoma and Non-Hodgkin Lymphoma. The age-specific hazard rates of disease onset and death in the affected populations were estimated by the Surveillance, Epidemiology, and End Results Program (SEER) SEER*Stat Software, and the age-specific hazard rates of death in the United States were estimated from actuarial life tables provided by the Social Security Administration.

The four columns in the `SubtypeHazards` dataset provide age-specific hazard rates, in yearly increments, beginning at age 0 and ending with age 100. That is, the values in the first row describe the hazard rates for an individual whose age is contained in the interval [0, 1), while the values in the second row describe the hazard rates for an individual whose age is contained in the interval [1, 2), and so on.

**References**

- The Surveillance, Epidemiology, and End Results (SEER) Program. [https://seer.cancer.gov/](https://seer.cancer.gov/)
summary.ped

Summarize a sample of pedigrees

Description

Summarize a sample of pedigrees

Usage

## S3 method for class 'ped'
summary(object, ...)

Arguments

object An object of class ped.
...
additional arguments passed to other methods.

Details

The summary.ped function returns two data frames. The first is called family_info, and contains the following fields for each family supplied.

<table>
<thead>
<tr>
<th>variable</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FamID</td>
<td>family identification number</td>
</tr>
<tr>
<td>total Relatives</td>
<td>total number of relatives</td>
</tr>
<tr>
<td>numAffected</td>
<td>total number of disease-affected individuals</td>
</tr>
<tr>
<td>aveOnsetAge</td>
<td>average onset age among the disease-affected relatives</td>
</tr>
<tr>
<td>aveIBD</td>
<td>average of the pairwise IBD probabilities among the disease-affected relatives</td>
</tr>
<tr>
<td>ascertainYear</td>
<td>the year the pedigree was ascertained</td>
</tr>
<tr>
<td>segRV</td>
<td>logical Indicates whether or not pedigree segregates a causal variant.</td>
</tr>
<tr>
<td>p_subtypeLabel</td>
<td>NOTE: this is only listed when pedigrees contain relatives affected by multiple subtypes.</td>
</tr>
</tbody>
</table>

The second item returned by summary.ped is called affected_info, and contains the following fields for each disease-affected relative supplied.

<table>
<thead>
<tr>
<th>variable</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FamID</td>
<td>family identification number</td>
</tr>
<tr>
<td>ID</td>
<td>individual identification number</td>
</tr>
<tr>
<td>birthYr</td>
<td>the individual’s birth year, when applicable, otherwise NA</td>
</tr>
<tr>
<td>onsetYr</td>
<td>the individual’s year of disease onset, when applicable, otherwise NA</td>
</tr>
<tr>
<td>deathYr</td>
<td>the individual’s year of death, when applicable, otherwise NA</td>
</tr>
<tr>
<td>proband</td>
<td>a proband identifier: proband = TRUE if the individual is the proband, and FALSE otherwise.</td>
</tr>
</tbody>
</table>
RVstatus the individual’s causal RV status; set to 1 if individual is a carrier, and 0 otherwise.

subtype NOTE: this is only listed when pedigree was simulated for diseases with multiple subtypes. The individual’s disease subtype.

Value

family_info A data frame containing family specific variables for each pedigree supplied. See details.

affected_info A data frame containing information for the affected individuals in each pedigree supplied. See details.

Examples

# Read in age-specific hazard data and create hazard object.
data(AgeSpecific_Hazards)
haz_obj <- hazard(hazardDF = AgeSpecific_Hazards)

# Simulate a pedigree ascertained for multiple affecteds
set.seed(6)
RVped2015 <- sim_RVped(hazard_rates = haz_obj,
num_affected = 2,
ascertain_span = c(1900, 2015),
GRR = 30, carrier_prob = 0.002,
RVfounder = TRUE,
stop_year = 2015,
recall_probs = c(1),
founder_byears = c(1900, 1925),
FamID = 1)[[2]]

# Plot the pedigree with age labels at the year 2015
plot(RVped2015, ref_year = 2015)

# View summary information for the pedigree
summary(RVped2015)

# Import the EgPeds dataset and create ped object
data(EgPeds)
study_peds <- new.ped(EgPeds)

# View summary information for study_peds
summary(study_peds)

trim_ped Trim pedigree based on proband recall
trim_ped

Description

Primarily intended as an internal function, trim_ped chooses a proband and trims relatives based on the proband’s probability of recalling his or her relatives.

Usage

trim_ped(ped_file, recall_probs = NULL)

Arguments

- ped_file: An object of class ped. A pedigree generated by sim_ped or sim_RVped, or an object created by the function new.ped. See details.
- recall_probs: Numeric. The proband’s recall probabilities for relatives, see details. If not supplied, the default value of four times kinship coefficient between the proband and the relative is used.

Details

By default recall_probs is four times the kinship coefficient, as defined by Thompson (see references), between the proband and the probands relative, which results in a recall probability of $2^{-(n-1)}$ for a relative of degree $n$. Alternatively, the user may specify a list of recall probabilities of length $l > 0$, in which case the first $l-1$ items in recall_probs are the respective proband recall probabilities for relatives of degree 1, 2, ..., $l-1$, and the $l$th item in recall_probs is the proband’s recall probability for all relatives of degree $l$ or greater. For example if recall_probs = c(1) all relatives will be recalled by the proband with probability 1.

Occasionally, a trimmed family member must be retained to ensure that the pedigree can be plotted. When this occurs, family members who share a non-zero kinship coefficient with the proband are censored of all pertinent information, and will always have the following qualities:

1. availability status = 0
2. affected status = NA
3. birth year = NA
4. onset year = NA
5. death year = NA
6. RR = NA

Users who wish to use trim_ped for pedigrees not generated by sim_ped or sim_RVped must use new.ped to create an object of class ped. The ped object must contain the following variables for each pedigree member:

<table>
<thead>
<tr>
<th>name</th>
<th>type</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FamID</td>
<td>numeric</td>
<td>family identification number</td>
</tr>
<tr>
<td>ID</td>
<td>numeric</td>
<td>individual identification number</td>
</tr>
<tr>
<td>dadID</td>
<td>numeric</td>
<td>identification number of father</td>
</tr>
<tr>
<td>momID</td>
<td>numeric</td>
<td>identification number of mother</td>
</tr>
<tr>
<td>sex</td>
<td>numeric</td>
<td>gender identification; if male sex = 0, if female sex = 1</td>
</tr>
<tr>
<td>affected</td>
<td>logical</td>
<td>disease-affection status:</td>
</tr>
</tbody>
</table>
trim_ped

proband logical a proband identifier: proband = TRUE if the individual is the proband, and FALSE otherwise.
affected = TRUE if affected by disease, and FALSE otherwise,
birthYr numeric the individual’s birth year.
onsetYr numeric the individual’s disease onset year, when applicable.
deathYr numeric the individual’s death year, when applicable.
RR numeric the individual’s relative risk of disease.
available logical availability status; available = TRUE if available, and FALSE otherwise.

Value
ped_trim The trimmed pedigree.

References

See Also
sim_RVped, sim_ped, new_ped

Examples
#Read in example pedigree to trim
data(EgPeds)
egPeds <- new.ped(EgPeds)

#plot example_ped using kinship2
plot(subset(egPeds, FamID == 1), location = "topright", cex = 0.85)
mtext("Original Pedigree", side = 3, line = 2)

## Trim pedigree examples
# Illustrate the effect of various settings for recall_probs
Recall_Probabilities <- list(c(1),
c(1, 0.5),
c(1, 0.25, 0.1))

for (k in 1:length(Recall_Probabilities)) {
  set.seed(2)
  #trim pedigree
  TrimPed <- trim_ped(ped_file = subset(egPeds, FamID == 1),
                     recall_probs = Recall_Probabilities[[k]])
  plot(TrimPed, location = "topright", cex = 0.85)
mtext(paste0("recall_probs = (", sep = "",
   paste(Recall_Probabilities[[k]], collapse = ", ", ", ") ),
   side = 3, line = 2 )
}
Index

* datasets

  AgeSpecific_Hazards, 2
  EgPeds, 4
  SubtypeHazards, 25

AgeSpecific_Hazards, 2

censor_ped, 3, 8

EgPeds, 4

find_mrca, 5

hazard, 6, 15, 17, 20, 21, 26

mtext, 11

new.ped, 3, 4, 8, 12, 13, 28, 29

par, 11

ped2pedigree, 9

pedigree, 10

pedigree.legend, 10, 11

plot, 11

plot.ped, 10

plot.pedigree, 11

reassign_gen, 8, 12

sim_life, 6, 14, 18, 19, 21, 23

sim_ped, 3, 6, 17, 22, 23, 29

sim_RVped, 3, 6, 17, 19, 20, 29

SimRVPedigree, 14

SimRVPedigree-package (SimRVPedigree), 14

SubtypeHazards, 25

summary.ped, 26

trim_ped, 8, 23, 27