Package ‘denvax’

October 13, 2022

Type Package

Title Simple Dengue Test and Vaccinate Cost Thresholds

Version 0.1.2

Description

Provides the mathematical model described by ``Serostatus Testing & Dengue Vaccine Cost-Benefit Thresholds''
in <doi:10.1098/rsif.2019.0234>. Using the functions in the package,
that analysis can be repeated using sample life histories, either synthesized from local seroprevalence data
using other functions in this package (as in the manuscript) or from some other source.
The package provides a vignette which walks through the analysis in the publication, as well as a function
to generate a project skeleton for such an analysis.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 3.5)

Suggests data.table, testthat, usethis, devtools, roxygen2, jsonlite, ggplot2, cowplot, directlabels, rmarkdown, knitr

RoxygenNote 6.1.1

VignetteBuilder knitr

URL `https://gitlab.com/cabp_LSHTM/denvax`

BugReports `https://gitlab.com/cabp_LSHTM/denvax/issues`

NeedsCompilation no

Author Carl A. B. Pearson [aut, cre] (<`https://orcid.org/0000-0003-0701-7860`>),
Kaja M. Abbas [aut] (<`https://orcid.org/0000-0003-0563-1576`>),
Samuel Clifford [aut] (<`https://orcid.org/0000-0002-3774-3882`>),
Stefan Flasche [aut] (<`https://orcid.org/0000-0002-5808-2606`>),
Thomas J. Hladish [aut] (<`https://orcid.org/0000-0003-1819-6235`>)

Maintainer Carl A. B. Pearson <`carl.pearson@lshtm.ac.uk`>

Repository CRAN

Date/Publication 2019-12-01 19:00:02 UTC

R topics documented:

`build.project` *Creates an ROI estimation project.*

Description

Creates an ROI estimation project.

Usage

```
build.project(targetdir, overwrite = FALSE, copy_pub = TRUE)
```

Arguments

`targetdir,`	path to enclosing directory. If this directory does not exist, will attempt to create it (recursively)
`overwrite,`	overwrite existing files corresponding to the project skeleton elements?
`copy_pub,`	copy the '/pub' folder (which contains the analyses from the publication)

Details

This function sets up the skeleton of an analysis to go from seroprevalence data to the ROI estimation surface. That skeleton uses a series of separate scripts for each analytical step (fitting, simulation, analysis, and application), connected via the command line build tool `make`. This approach allows clean substitution for various stages (e.g., using a different model to generate life histories). The following files are created:

Makefile the dependencies for various analysis stages

README.md brief notes about project parts

fit.R script for fitting seroprevalence data

synthesize.R script for generating synthetic populations

digest.R script for converting life histories into probability coefficients for ROI calculation

simple.R a quick example start-to-finish analysis

Value

logical, indicating error free creation of the project skeleton; there may still be other warnings

Examples

```
require(denvax)
tardir <- tempdir() # replace with desired target
build.project(tardir)
list.files(tardir, recursive = TRUE)
```

`denvax` *denvax: Simple Dengue Test and Vaccinate Cost Thresholds*

Description

Provides the mathematical model described by "Serostatus Testing & Dengue Vaccine Cost-Benefit Thresholds" in <doi:10.1098/rsif.2019.0234>. Using the functions in the package, that analysis can be repeated using sample life histories, either synthesized from local seroprevalence data using other functions in this package (as in the manuscript) or from some other source. The package provides a vignette which walks through the analysis in the publication, as well as a function to generate a project skeleton for such an analysis.

`denvax` functions

`serofit`	fits serosurvey data against two-risk model
`synthetic.pop`	using parameter fit data, generate a sample population
`nPxA`	estimate dengue infection outcome probabilities based on a synthetic population
`ROIcoeffs`	using population outcome probabilities, compute ROI equation coefficients
`ROI`	compute ROIs from setting coefficients and cost scenarios
`build.project`	create a template project for estimating ROIs

`lazou2016` *The serosurvey data in L'Azou 2016*

Description

From "Symptomatic Dengue in Children in 10 Asian and Latin American Countries", Table 4.

Usage

```
lazou2016
```

Format

a data.frame (data.table, if installed) with 20 rows and 4 columns:

Country character, common country name (all Peru for this data)

Age character, the bounding ages for the sample; format: lower age '-' upper age

Number integer, the number of samples

Seropositive integer, the number of seropositive samples

Source

`https://doi.org/10.1056/NEJMoa1503877`

Examples

```
require(denvax); require(ggplot2)
data(lazou2016)
ggplot(lazou2016) + aes(Age, Seropositive/Number*100, color = Country) +
  geom_point() + labs(y="Seropositive %", x="Age Group") + lims(y=c(0,100)) +
  theme_minimal()
```

`morrison2010` *The serosurvey data in Morrison 2010*

Description

From "Epidemiology of Dengue Virus in Iquitos, Peru 1999 to 2005: Interepidemic and Epidemic Patterns of Transmission", combining information from Fig. 2 and Fig. 3. The data from Fig. 3 were extracted using `https://automeris.io/WebPlotDigitizer/`

Usage

```
morrison2010
```

Format

a data.frame (data.table, if installed) with 13 rows and 4 columns:

Country character, common country name (all Peru for this data)

Age integer, the age category

Number integer, the number of samples

Seropositive integer, the number of seropositive samples

Source

`https://doi.org/10.1371/journal.pntd.0000670`

Examples

```
require(denvax)
data(morrison2010)
with(morrison2010, plot(Age, Seropositive/Number*100, ylab="% Seropositive", ylim=c(0,100)))
```

`nPxA` *Compute the nPx(A), C(A) proportions from a population of life histories*

Description

Compute the nPx(A), C(A) proportions from a population of life histories

Usage

```
nPxA(lifehistory)
```

Arguments

`lifehistory,`	a matrix with rows (sample individuals) and columns (outcome in year of life); see synthetic.pop return value

Details

computes the relevant nPx(A) and C(A): the probabilities of the various life trajectories, by age. See <doi:10.1098/rsif.2019.0234>, SI section II.A (Cost Benefit Equations: Definitions)

Value

a `data.frame` (`data.table`, if available) with columns

A integer; the reference year of life, from 1 to `dim(lifehistory)[2]`

p_0 numeric; probability of 0 lifetime infections

p_1p numeric; probability of 1 or more lifetime infections

p_2p numeric; probability of 2 or more lifetime infections

p0_1 numeric; probability of 0 infections at age A, and 1 lifetime infection

p0_1p numeric; probability of 0 infections at age A, and 1 or more lifetime infections

p0_2p numeric; probability of 0 infections at age A, and 2 or more lifetime infections

p1_A numeric; probability of 1 infection at age A, and 1 or more lifetime infections

p1_2p numeric; probability of 1 infection at age A, and 2 or more lifetime infections

p1p_A numeric; probability of 1 or more infections at age A, and 1 or more lifetime infections

p2p_A numeric; probability of 2 or more infections at age A, and 2 or more lifetime infections

CA numeric; probability of converting from seronegative to seropositive between age A and A+1

Examples

```
require(denvax);
data(morrison2010) # has counts by age
fit <- with(morrison2010, serofit(sero=Seropositive, N=Number, age.min=Age))
m2010pop <- synthetic.pop(fit, runs = 10, popsize = 10) # small sample size for example run time
m2010lh <- nPxA(m2010pop)
m2010lh
with(m2010lh,
  plot(A, p0_2p*100, type="l",
    xlab="Age", ylab="%", ylim = c(0, 100),
    main="Individuals w/ No Infections,\nbut that will have 2"
  )
)
```

ROI *Compute the ROI surfaces given test and vaccine cost fractions.*

Description

Compute the ROI surfaces given test and vaccine cost fractions.

Usage

```
ROI(rcoeffs, nus = seq(0.3, 0.9, by = 0.05), taus = 10^seq(-2, -0.5, by
  = 0.05))
```

Arguments

`rcoeffs,`	a data.frame with the ROI surface coefficients from ROIcoeffs
`nus,`	the series of normalized vaccine costs to use for ROI calcs
`taus,`	the series of normalized test costs to use for ROI calcs

Details

tabulates ROI

Value

a ‘data.frame‘ (‘data.table‘, if available) with columns:

nu numeric, the normalized vaccine cost used

tau numeric, the normalized test cost used

mechanism character, either "ordinal" or "binary" corresponding to the type of test

A integer; the age when routine test-then-vaccinate strategy starts (from `As`)

L integer; the maximum number of tests for routine test-then-vaccinate strategy (from `Ls`)

cost numeric; the intervention cost (as a fraction of second infection cost)

benefit numeric; the difference in health outcome cost (as a fraction of second infection cost) minus ‘cost‘; positive values indicate positive net benefit

roi numeric; return on investment: ‘benefit‘ over ‘cost‘

Examples

```
require(denvax);
data(morrison2010) # has counts by age
fit <- with(morrison2010, serofit(sero=Seropositive, N=Number, age.min=Age))
m2010pop <- synthetic.pop(fit, runs = 10, popsize = 10) # small sample size for example run time
m2010lh <- nPxA(m2010pop)
L <- 5
rc <- ROIcoeffs(m2010lh, As=5:10, Ls=L)
rois <- ROI(rc, nus = 0.5, taus = 0.01)
srois <- subset(rois, mechanism == "binary")
mrois <- matrix(srois$roi, nrow = L)
contour(x=unique(srois$L), y=unique(srois$A), z=mrois,
  xlab = "Max # of Tests", ylab = "Initial Age", main="ROI Contour"
)
```

`ROIcoeffs` *Compute the Return on Investment (ROI) surface coefficients from population probabilities*

Description

Compute the Return on Investment (ROI) surface coefficients from population probabilities

Usage

```
ROIcoeffs(probabilities, As = 5:20, Ls = (diff(range(As)) + 1):1)
```

Arguments

`probabilities,`	a `data.frame` (or `data.table`) with the probabilities resulting from nPxA. Rows must correspond to ages, starting with age 1
`As,`	the starting age(s) to consider
`Ls,`	the maximum number of tests for each age; should either be an integer per age or a single integer for all ages. The default behavior computes the number of tests (for each age) that makes the maximum of ‘As‘ the maximum testing age Note: results will also be provided for shorter testing intervals, as the intermediate coefficients are calculated as part of computing the value at the maximum `L`

Details

computes the coefficients for the economic calculations

Value

a `data.frame` (`data.table`, if available) with columns:

A integer; the age when routine test-then-vaccinate strategy starts (from `As`)

L integer; the maximum number of tests for routine test-then-vaccinate strategy (from `Ls`)

vacfrac numeric; the fraction of individuals participating in this strategy that get vaccinated

pri.offset numeric; the (additive) reduction in `vacfrac` if using the ordinal test

Sfrac numeric; the proportion experiencing second infection costs

Fresp numeric; the F/S cost fraction term, when comparing vaccination with and without testing

Sgain numeric; the S term, when comparing vaccination with and without testing

Examples

```
require(denvax);
data(morrison2010) # has counts by age
fit <- with(morrison2010, serofit(sero=Seropositive, N=Number, age.min=Age))
m2010pop <- synthetic.pop(fit, runs = 10, popsize = 10) # small sample size for example run time
m2010lh <- nPxA(m2010pop)
rc <- ROIcoeffs(m2010lh, As=5:10, Ls=5)
pp <- par()
par(mfrow=c(1, 2))
rcs <- subset(rc, A==10 & L < 11)
with(rcs, plot(
  L, aveTests, type="l",
  xlab="Max # of Tests Allowed",
  ylab="Ave # of Tests Administered",
  main="Starting @ Age 10",
  ylim=c(1, 3)
))
rcs <- subset(rc, A==5 & L < 11)
with(rcs, plot(
  L, aveTests, type="l",
  xlab="Max # of Tests Allowed",
  ylab="",
  main="Starting @ Age 5",
  ylim=c(1, 3)
))
par(pp)
```

`serofit` *Model fitting for serological data*

Description

Model fitting for serological data

Usage

```
serofit(seropositive, N, age.min = use.default(1L:length(seropositive),
  "age.min"), age.max = use.default(age.min, "age.max"))
```

Arguments

`seropositive,`	the number of seropositive samples for each age group; length(seropositive) must be at least 3
`N,`	the total number of samples for each age group; length(N) must equal length(seropositive)
`age.min,`	the low age in age groups; defaults to ‘1:length(seropositive)‘, i.e. assumes the seropositive data corresponds to yearly cohorts starting at age 1.
`age.max,`	the upper age in age groups; defaults to ‘age.min‘, i.e. assumes each category corresponds to a single year

Details

Fits a constant force of infection, two-risk category model using seroprevalence survey data. i.e.:

$$ P_+(A) = p_H * (1-(1-f_H)^A) + (1-p_H) * (1-(1-f_L)^A) $$

This probability is fit to the seroprevalence by age category data, using maximum likelihood and `optim`.

Value

a list of best-fit parameters, all numeric values:

f_H force of infection, for the high risk group

f_L force of infection, for the low risk group

p_H the proportion of the population at high risk

Examples

```
require(denvax);
data(morrison2010) # has counts by age
fit <- with(morrison2010, serofit(sero=Seropositive, N=Number, age.min=Age))
if (requireNamespace("data.table", quietly = TRUE)) {
data(lazou2016) # has counts by age range, instead of counts for every year
# this example uses `data.table`` functions to simplify processing
# several groups at once
```

```
  lazou2016[,{
    agerange <- data.table::tstrsplit(Age, "-")
    serofit(
      sero     = Seropositive,
      N        = Number,
      age.min  = as.integer(agerange[[1]]),
      age.max  = as.integer(agerange[[2]])
    )
  }, by = Country]
}
```

`synthetic.pop` *Compute synthetic population trajectories from model parameters*

Description

Compute synthetic population trajectories from model parameters

Usage

```
synthetic.pop(pars, runs = 100, popsize = 1000, maxAge = 70,
  rngseed = NULL)
```

Arguments

`pars,`	a list with elements ‘f_H‘, ‘f_L‘, and ‘p_H‘; see serofit return value
`runs,`	the number of different serotype timelines to simulate
`popsize,`	the number of individuals to sample for each run
`maxAge,`	the length of each lifetime
`rngseed,`	an optional seed for the random number generator

Details

Using fitted parameters for a two-risk, constant force of infection model, simulate a dengue annual exposures model for the requested number of serotype series (‘runs‘) and individuals (‘popsize‘). The resulting matrix is a collection of integers, 0-4. 0 indicates no infection, 1-4 infection by the corresponding serotype.

Value

a matrix of integers 0-4, rows ‘runs*popsize‘ x columns ‘maxAge‘

Examples

```
require(denvax);
data(morrison2010) # has counts by age
fit <- with(morrison2010, serofit(sero=Seropositive, N=Number, age.min=Age))
m2010pop <- synthetic.pop(fit, runs = 10, popsize = 10) # small sample size for example run time
head(m2010pop)
```

Index