Package: ringbp (via r-universe)

May 10, 2026

Type Package

Title Simulate and evaluate contact tracing scenarios

Version 0.1.2

Description A package to simulate transmission of an infectious disease and associated efforts to trace conacts and reduce onwards transmission.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Depends R (>= 3.5.0)

Imports data.table, future.apply, purrr, sn

Suggests future, testthat

URL `https://epiforecasts.io/ringbp`, `https://github.com/epiforecasts/ringbp`

BugReports `https://github.com/epiforecasts/ringbp/issues`

Repository https://epiforecasts.r-universe.dev

Date/Publication 2024-01-18 11:54:14 UTC

RemoteUrl https://github.com/epiforecasts/ringbp

RemoteRef v0.1.2

RemoteSha c64bd673927f5993a4e2ee41d4e01f8ae796e9f5

Contents

`detect_extinct` *Calculate whether outbreaks went extinct or not*

Description

Calculate whether outbreaks went extinct or not

Usage

```
detect_extinct(outbreak_df_week = NULL, cap_cases = NULL, week_range = 12:16)
```

Arguments

`outbreak_df_week`	data.table weekly cases produced by the outbreak model
`cap_cases`	integer number of cumulative cases at which the branching process was terminated
`week_range`	integer vector giving the (zero indexed) week range to test for whether an extinction occurred.

Author(s)

Joel Hellewell

`dist_setup` *Create partial function to sample from gamma distributions*

Description

Create partial function to sample from gamma distributions

Usage

```
dist_setup(dist_shape = NULL, dist_scale = NULL)
```

Arguments

`dist_shape`	numeric shape parameter of Weibull distribution
`dist_scale`	numeric scale parameter of Weibull distribution

Value

partial function that takes a numeric argument for number of samples

Author(s)

Joel Hellewell

`extinct_prob` *Calculate proportion of runs that have controlled outbreak*

Description

Calculate proportion of runs that have controlled outbreak

Usage

```
extinct_prob(outbreak_df_week = NULL, cap_cases = NULL, week_range = 12:16)
```

Arguments

`outbreak_df_week`	data.table weekly cases produced by the outbreak model
`cap_cases`	integer number of cumulative cases at which the branching process was terminated
`week_range`	integer vector giving the (zero indexed) week range to test for whether an extinction occurred.

Author(s)

Joel Hellewell

`inf_fn` *Samples the serial interval for given incubation period samples*

Description

Samples the serial interval for given incubation period samples

Usage

```
inf_fn(inc_samp = NULL, k = NULL)
```

Arguments

`inc_samp`	vector of samples from the incubation period distribution
`k`	numeric skew parameter for sampling the serial interval from the incubation period

---

`outbreak_model` *Run a single instance of the branching process model*

Description

Run a single instance of the branching process model

Usage

```
outbreak_model(
  num.initial.cases = NULL,
  prop.ascertain = NULL,
  cap_max_days = NULL,
  cap_cases = NULL,
  r0isolated = NULL,
  r0community = NULL,
  r0subclin = NULL,
  disp.iso = NULL,
  disp.com = NULL,
  disp.subclin = NULL,
  k,
  delay_shape = NULL,
  delay_scale = NULL,
  prop.asym = NULL,
  quarantine = NULL
)
```

Arguments

`num.initial.cases`	The number of initial or starting cases which are all assumed to be missed.
`prop.ascertain`	numeric proportion of infectious contacts ascertained by contact tracing (must be 0<=x<=1)
`cap_max_days`	Stop the simulation when this many days is reached.
`cap_cases`	Stop the simulation when this many cases is reached.
`r0isolated`	numeric reproduction number for isolated cases (must be >0)
`r0community`	numeric reproduction number for non-isolated cases (must be >0)
`r0subclin`	numeric reproduction number for sub-clinical non-isolated cases (must be >0)
`disp.iso`	numeric dispersion parameter for isolated cases (must be >0)
`disp.com`	numeric dispersion parameter for non-isolated cases (must be >0)
`disp.subclin`	numeric dispersion parameter for sub-clincial non-isolated cases (must be >0)
`k`	numeric skew parameter for sampling the serial interval from the incubation period
`delay_shape`	numeric shape parameter of delay distribution

`delay_scale`	numeric scale parameter of delay distribution
`prop.asym`	proportion of cases that are completely asymptomatic.
`quarantine`	logical whether quarantine is in effect, if TRUE then traced contacts are isolated before symptom onset

Value

data.table of cases by week, cumulative cases, and the effective reproduction number of the outreak

Author(s)

Joel Hellewell

Examples

```
## Not run:
incfn <- dist_setup(dist_shape = 2.322737,dist_scale = 6.492272)
# delay distribution sampling function
delayfn <- dist_setup(2, 4)
# generate initial cases
case_data <- outbreak_setup(num.initial.cases = 5,
                            incfn=incfn,
                            delayfn = delayfn,
                            k=1.95,
                            prop.asym=0)
# generate next generation of cases
case_data <- outbreak_step(case_data = case_data,
                           disp.iso = 1,
                           disp.com = 0.16,
                           disp.subclin = 0.16,
                           r0isolated = 0,
                           r0community = 2.5,
                           r0subclin = 1.25,
                           prop.asym = 0,
                           incfn = incfn,
                           delayfn = delayfn,
                           prop.ascertain = 0,
                           k = 1.95,
                           quarantine = FALSE)

## End(Not run)
```

`outbreak_setup` *Set up initial cases for branching process*

Description

Set up initial cases for branching process

Usage

```
outbreak_setup(num.initial.cases, incfn, delayfn, k, prop.asym)
```

Arguments

`num.initial.cases`	Integer number of initial cases
`incfn`	function that samples from incubation period Weibull distribution; generated using dist_setup
`delayfn`	function that samples from the onset-to-hospitalisation delay Weibull distribution; generated using dist_setup
`k`	Numeric skew parameter for sampling the serial interval from the incubation period
`prop.asym`	Numeric proportion of cases that are sublinical (between 0 and 1)

Value

data.table of cases in outbreak so far

Author(s)

Joel Hellewell

Examples

```
## Not run:
# incubation period sampling function
incfn <- dist_setup(dist_shape = 2.322737,dist_scale = 6.492272)
# delay distribution sampling function
delayfn <- dist_setup(delay_shape, delay_scale)
outbreak_setup(num.initial.cases = 5,incfn,delayfn,k=1.95,prop.asym=0)

## End(Not run)
```

`outbreak_step` *Move forward one generation in the branching process*

Description

Move forward one generation in the branching process

Usage

```
outbreak_step(
  case_data = NULL,
  disp.iso = NULL,
  disp.com = NULL,
  r0isolated = NULL,
  r0community = NULL,
  prop.asym = NULL,
  incfn = NULL,
  delayfn = NULL,
  prop.ascertain = NULL,
  k = NULL,
  quarantine = NULL,
  r0subclin = NULL,
  disp.subclin = NULL
)
```

Arguments

`case_data`	data.table of cases in outbreak so far; initially generated by outbreak_setup
`disp.iso`	numeric dispersion parameter for isolated cases (must be >0)
`disp.com`	numeric dispersion parameter for non-isolated cases (must be >0)
`r0isolated`	numeric reproduction number for isolated cases (must be >0)
`r0community`	numeric reproduction number for non-isolated cases (must be >0)
`prop.asym`	proportion of cases that are completely asymptomatic.
`incfn`	function samples from incubation period; generated by dist_setup
`delayfn`	function samples from the onset-to-hospitalisation delay; generated by dist_setup
`prop.ascertain`	numeric proportion of infectious contacts ascertained by contact tracing (must be 0<=x<=1)
`k`	numeric skew parameter for sampling the serial interval from the incubation period
`quarantine`	logical whether quarantine is in effect, if TRUE then traced contacts are isolated before symptom onset
`r0subclin`	numeric reproduction number for sub-clinical non-isolated cases (must be >0)
`disp.subclin`	numeric dispersion parameter for sub-clincial non-isolated cases (must be >0)

Author(s)

Joel Hellewell

Examples

```
## Not run:
# incubation period sampling function
incfn <- dist_setup(dist_shape = 2.322737,dist_scale = 6.492272)
```

```
# delay distribution sampling function
delayfn <- dist_setup(1.651524, 4.287786)
# generate initial cases
case_data <- outbreak_setup(num.initial.cases = 5,incfn,delayfn,k=1.95,prop.asym=0)
# generate next generation of cases
case_data <- outbreak_step(case_data = case_data,
disp.iso = 1,
disp.com = 0.16,
r0isolated = 0,
r0subclin = 1.25,
disp.subclin = 0.16,
r0community = 2.5,
prop.asym = 0,
incfn = incfn,
delayfn = delayfn,
prop.ascertain = 0,
k = 1.95,
quarantine = FALSE)[[1]]

## End(Not run)
```

`parameter_sweep` *Sweep across parameters*

Description

Explore scenarios using gridding with sampling for parameters not in the grid. Parameters that are included in the grid are currently hard coded. Use the `future` package to control parallisation outside of the function.

Usage

```
parameter_sweep(scenarios = NULL, samples = 1, sim_fn = NULL)
```

Arguments

`scenarios`	A dataframe containing all gridded scenarios - see the examples for the required structure. Defaults to NULL.
`samples`	Numeric, defaults to 1. The number of samples to take.
`sim_fn`	Function, defaults to NULL. The vectorised model simulation function - see the examples for usage.

Value

A nested tibble containing the parameters for each scenario and a nested list of output from `wuhan_sim`.

Examples

```
## Not run:
library(data.table)

scenarios <- data.table(expand.grid(
delay_group = list(data.table(
 delay = c("SARS","Wuhan"),
 delay_shape = c(1.651524,2.305172),
 delay_scale = c(4.287786,9.483875)
)),
k_group = list(data.table(
 theta = c("<1%","15%","30%"),
 k = c(1,0.88,0.47)
)),
index_R0 = c(1.5,2.5,3.5),
prop.asym = c(0,  0.1),
control_effectiveness = seq(0,1,0.2),
num.initial.cases = c(5,20,40))

list_cols <- grep("_group", colnames(scenarios), value = TRUE)
non_list_cols <- setdiff(colnames(scenarios), list_cols)

expanded_groups <- scenarios[, rbindlist(delay_group), by = c(non_list_cols)]
expanded_k <- scenarios[, rbindlist(k_group), by = c(non_list_cols)]

scenarios <- merge(
  expanded_groups, expanded_k, by = non_list_cols, allow.cartesian = TRUE
)
scenarios[, scenario :=  1:.N]

## Parameterise fixed paramters
sim_with_params <- purrr::partial(ringbp::scenario_sim,
                                 cap_max_days = 365,
                                 cap_cases = 5000,
                                 r0isolated = 0,
                                 disp.iso= 1,
                                 disp.subclin = 0.16,
                                 disp.com = 0.16,
                                 quarantine = FALSE)


## Default is to run sequntially on a single core
future::plan("sequential")
## Set up multicore if using see ?future::plan for details
## Use the workers argument to control the number of cores used.
future::plan("multiprocess")


## Run paramter sweep
sweep_results <- ringbp::parameter_sweep(scenarios, sim_fn = sim_with_params, samples = 1)
```

```
sweep_results

## End(Not run)
```

`scenario_sim` *Run a specified number of simulations with identical parameters*

Description

Run a specified number of simulations with identical parameters

Usage

```
scenario_sim(
  n.sim,
  prop.ascertain,
  cap_max_days,
  cap_cases,
  r0isolated,
  r0community,
  disp.iso,
  disp.com,
  k,
  delay_shape,
  delay_scale,
  num.initial.cases,
  prop.asym,
  quarantine,
  r0subclin = NULL,
  disp.subclin = NULL
)
```

Arguments

`n.sim`	number of simulations to run
`prop.ascertain`	numeric proportion of infectious contacts ascertained by contact tracing (must be 0<=x<=1)
`cap_max_days`	Stop the simulation when this many days is reached.
`cap_cases`	Stop the simulation when this many cases is reached.
`r0isolated`	numeric reproduction number for isolated cases (must be >0)
`r0community`	numeric reproduction number for non-isolated cases (must be >0)
`disp.iso`	numeric dispersion parameter for isolated cases (must be >0)
`disp.com`	numeric dispersion parameter for non-isolated cases (must be >0)

`k`	numeric skew parameter for sampling the serial interval from the incubation period
`delay_shape`	numeric shape parameter of delay distribution
`delay_scale`	numeric scale parameter of delay distribution
`num.initial.cases`	The number of initial or starting cases which are all assumed to be missed.
`prop.asym`	proportion of cases that are completely asymptomatic.
`quarantine`	logical whether quarantine is in effect, if TRUE then traced contacts are isolated before symptom onset
`r0subclin`	numeric reproduction number for sub-clinical non-isolated cases (must be >0)
`disp.subclin`	numeric dispersion parameter for sub-clincial non-isolated cases (must be >0)

Value

A data.table object returning the results for multiple simulations using the same set of parameters. The table has columns

- week: The week in the simulation.
- weekly_cases: The number of new cases that week.
- cumulative: The cumulative cases.
- effective_r0: The effective reproduction rate for the whole simulation
- cases_per_gen: A list column with the cases per generation. This is repeated each row.
- sim: Index column for which simulation.

Author(s)

Joel Hellewell

Examples

```
## Not run:
res <- scenario_sim(n.sim = 5,
num.initial.cases = 5,
cap_max_days = 365,
cap_cases = 2000,
r0isolated = 0,
r0community = 2.5,
disp.iso = 1,
disp.com = 0.16,
k = 0.7,
delay_shape = 2.5,
delay_scale = 5,
prop.asym = 0,
prop.ascertain = 0)

## End(Not run)
```

Index