Package: SEEPS (via r-universe)

February 22, 2025

Title Sequence evolution and epidemiological process simulator

Version 0.2.0

Description A modular, modern simulation suite and toolkit for simulating transmission networks, phylogenies, and evolutionary pairwise distance matrices under different models and assumptions for viral/sequence evolution. While intially developed for HIV, SEEPS offers modular utilities for custom workflows for extension beyond HIV.

License file LICENSE

maintainer Michael D. Kupperman <mkupperman@lanl.gov>

Encoding UTF-8

LazyData true

Suggests testthat (>= 2.1.0), rlang (>= 1.0.0), knitr, rmarkdown

Imports ape (>= 5.0.0), phyclust (>= 0.1), rngtools (>= 1.0.0), stringr (>= 1.3.1), stats

RoxygenNote 7.3.2

Depends R (>= 3.5.0)

VignetteBuilder knitr

Config/pak/sysreqs libicu-dev

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

RemoteUrl https://github.com/MolEvolEpid/SEEPS

RemoteRef HEAD

RemoteSha 919ad50f20ccff2d71a42acf57913fb1e87846cf

Contents

add_root_to_newick *Add a root node to a newick tree string with distance 0 from the root*

Description

A utility function for placing an explicit "root" into a newick tree string. at the implicit root. This function performs: ‘(tree); -> (tree:0, root:0);‘.

Usage

```
add_root_to_newick(tree, root_name = "root")
```

Arguments

tree	A newick tree string to add a root node to.
root_name	The name of the root node to add. Default is "root".

See Also

phylogeny_to_newick

biphasic_HIV_rate *Simple biphasic rate function used in [Kupperman et al.]*

Description

Simple biphasic rate function used in [Kupperman et al.]

Usage

```
biphasic_HIV_rate(current_step, birth_step, params)
```

Arguments

current_step	Current time step
birth_step	When the infection was generated
params	A list with one element ‘R0‘ needed to evaluate the rate function.

build_distance_matrix_from_df

Build a distance matrix from a dataframe of sequences using ape

Description

Use ape to compute a distance matrix using the specified distance model. Default is the "TN93" model.

Usage

```
build_distance_matrix_from_df(df, model = "TN93", keep_root = FALSE)
```

check_rate_model

Check that a rate model is mathematically valid

Description

A helper function to check that a rate model paramterization is mathematically valid.

Usage

```
check_rate_model(rate_model)
```

Arguments

rate_model	A list of rate model parameters.

Details

Check that all rates are positive, that the fraction of bases add up to 1, the fraction of invariant sites is in [0, 1), and the discrete gamma model is defined for ‘alpha‘ and ‘ncat‘.

See Also

generate_rate_model generate_sequences

clean_up

Clean up the simulation state and return the relevant data.

Description

Clean up the simulation state and return the relevant data.

Usage

```
clean_up(state)
```

`contact_traced_uniform_ids` *Obtain a sample using a iterative contact tracing with a uniform discovery rate*

Description

Perform iterative contact tracing on a simulated contact network. Randomness within contact tracing comes from the probability of discovering a new infection. This function assumes the infection rate is uniform across all possible contacts and individuals.

Usage

```
contact_traced_uniform_ids(
  active,
  parents,
  minimum_sample_size,
  p,
  max_attempts = 1
)
```

Arguments

`active`	A vector of active individuals
`parents`	A matrix encoding the transmission history
`minimum_sample_size`	The minimum number of individuals to form a sample
`p`	The probability of discovering each contact during tracing
`max_attempts`	Maximum number of attempts to perform to obtain a sample. If no sample is found after this number of attempts, 'FALSE' is returned

Details

Contact tracing is terminated when 2 * min_sample_size active nodes are discovered, or when there are no more detections to trace. At least 'min_sample_size' individuals will always be returned.

To determine the initial detection, we loop over a the list of active nodes until we complete a successful contact tracing.

All contact tracing algorithms store both the ids of all discovered individuals and the ids of discovered active individuals.

Value

A list with three fields: "status", "samples", and "found"

`contact_traced_uniform_restarts_ids` *Obtain a sample using a iterative contact tracing with a uniform discovery rate If a minimum number is not found, the algorithm is restarted with a new initial detection*

Description

Perform iterative contact tracing on a simulated contact network. Randomness within contact tracing comes from the probability of discovering a new infection. This function assumes the infection rate is uniform across all possible contacts and individuals.

Usage

```
contact_traced_uniform_restarts_ids(active, parents, minimum_sample_size, p)
```

Arguments

`active`	A vector of active individuals
`parents`	A matrix encoding the transmission history
`minimum_sample_size`	The minimum number of individuals to form a sample
`p`	The probability of discovering each contact during tracing

Details

Contact tracing is terminated when 2 * min_sample_size active nodes are discovered, or when there are no more detections to trace. At least ‘min_sample_size‘ individuals will always be returned.

To determine the initial detection, we loop over a the list of active nodes until we complete a successful contact tracing.

All contact tracing algorithms store both the ids of all discovered individuals and the ids of discovered active individuals.

Value

A list with four fields: "status", "samples", "success", and "found" if the algorithm fails to find a sample, FALSE is returned instead

`contact_tracing_core` *Core contact tracing algorithm*

Description

Given a set of active nodes and the transmision history (parents), sample a set of nodes that are contact traced using the ‘discovery_function‘ parameter to determine the transmission rate.

Usage

```
contact_tracing_core(
  detected_id,
  active,
  parents,
  discovery_function,
  termination_function
)
```

Arguments

`detected_id`	The id of the node used to start the tracing
`active`	A vector of active individuals
`parents`	A matrix of transmission history
`discovery_function`	A function that takes a list of nodes and relative transmission times and determines which will be included

`contact_tracing_engine` *Perform contact tracing on a simulated contact network*

Description

The engine for simulating contact tracing. Provide network information (‘active‘, ‘parents‘), a list or vector of individuals to begin tracing with (‘detected_id‘), functions to control the discovery and termination of tracing (‘discovery_function‘, ‘termination_function‘), and parameters about iteration (‘max_attempts‘) and fallback termination conditions (‘minimum_size‘).

Usage

```
contact_tracing_engine(
  detected_id,
  active,
  parents,
  discovery_function,
  max_attempts,
  termination_function,
  minimum_size = 3
)
```

Arguments

detected_id	A vector of individuals to begin tracing with
active	A vector of active individuals
parents	A matrix encoding the transmission history
discovery_function	A function to determine which individuals to trace
termination_function	A function to determine when to terminate tracing a contact tracing.
minimum_size	The minimum number of individuals to form a sample

Value

A list with three fields: "status", "samples", and "found"

---

fasta_string_to_dataframe

Convert a fasta string output by seq-gen to a dataframe

Description

A utility function to build a dataframe from a fasta string.

Usage

```
fasta_string_to_dataframe(fasta_string, trim = TRUE, include_root = FALSE)
```

Details

@param fasta_string A string containing the fasta output from seq-gen

Value

A dataframe with the "seq" and "name" columns

geneology_to_distance_matrix

Reduce a geneology to a pairwise distance matrix.

Description

Reduce a geneology to a pairwise distance matrix.

Usage

```
geneology_to_distance_matrix(geneology, mode = "mu", spike_root = FALSE)
```

geneology_to_distance_matrix_classic

Reduce a geneology to a pairwise distance matrix.

Description

Provided for backwards compatability with [Kupperman et al 2022]. New code should use '[geneology_to_distance_matrix]'.

Usage

```
geneology_to_distance_matrix_classic(geneology, spike_root = FALSE)
```

See Also

geneology_to_distance_matrix

geneology_to_phylogeny_bpb

Generate a phylogeny from a transmission history using BioPhyBreak's coalescent simulator

Description

Run a backwards simulation using the transmission history to generate a phylogeny. See 'geneology_to_phylogeny_bpb' for details on how the input should be structured. This function assumes that all coalescent events occur at or after the initial infection, that there is a single introduction and ancestral sequence.

Usage

```
geneology_to_phylogeny_bpb(
  transmission_history,
  infection_times,
  leaf_sample_ids,
  sample_times,
  a = 5,
  b = 5,
  make_plot = FALSE
)
```

Arguments

transmission_history	A transmission history matrix
infection_times	A vector of infection times

See Also

reduce_transmission_history_bpb

generate_sequences

---

generate_rate_model *Generate a rate model from provided paramters*

Description

Accept a set of parameters for GTR+I+G model. See below for details of the parameterization. To disable I, set the parameter i to 0. To disable G, set alpha=1 and ncat=1 for an exponential distribution.

Usage

```
generate_rate_model(
  a2c,
  a2g,
  a2t,
  c2g,
  c2t,
  g2t,
  fa,
  fc,
  fg,
  ft,
  i,
```

```
  alpha,
  ncat
)
```

Arguments

a2c	The rate of adenosine to cytosine transversions.
a2g	The rate of adenosine to cytosine transitions.
a2t	The rate of adenosine to thymine transversions.
c2g	The rate of cytosine to guanine transversions.
c2t	The rate of cytosine to thymine transitions.
g2t	The rate of guanine to thymine transversions.
fa	The fraction of adenosine at equilibrum.
fc	The fraction of cytosine at equilibrum.
fg	The fraction of guanine at equilibrum.
ft	The fraction of thymine at equilibrum.
i	The proportion of invariant sites. Traditionally estimated by the proportion of sites with no observed mutations.
alpha	The shape parameter for the discrete gamma distribution.
ncat	The number of categories in the discrete gamma distribution.

Details

For an overview of GTR+I+G and substitition models, see [here](https://www.ccg.unam.mx/~vinuesa/Model_fitting_in_phyl
For a more detailed construction, see [Yang 1994](https://doi.org/10.1007/BF00178256), and [Yang 1996](https://doi.org/10.1093/oxfordjournals.molbev.a025625).

Value

A list of rate model parameters.

See Also

generate_sequences get_V3_rate_model get_p17_rate_model check_rate_model

Examples

```
rate_model <- generate_rate_model(
    a2c = 1, a2g = 1, a2t = 1,
    c2g = 1, c2t = 1, g2t = 2,
    fa = 0.25, fc = 0.25,
    fg = 0.25, ft = 0.25,
    i = 0.1, alpha = 0.25, ncat = 8
)
```

`generate_sequences` *Generate sequences from a phylogeny using Seq-Gen through phyclust*

Description

Given a phylogeny, generate possible sequences using Seq-Gen [Rambaut & Grassley, 1997]. A root sequence for the simulation is required. The root sequence is placed at the MRCA of the phylogeny unless 'spike_root = TRUE' is specified when the phylogeny is constructed.

Usage

```
generate_sequences(
  phylogeny,
  branch_rate,
  root_sequence,
  rng_seed = -1,
  rate_model,
  rate_per_nt = FALSE
)
```

Arguments

`phylogeny`	A phylogeny object
`root_sequence`	A root sequence
`rate_model`	A list of GTR+I+G model parameters. Expects a list of 13 parameters: (6 rate parameters) 'a2c', 'a2g', 'a2t', 'c2g', 'c2t', 'g2t', (nucleotide frequencies:) 'fa, fc, fg, ft', (proportion of sites with no variation:) 'i', (site specific heterogeneity shape parameter) 'alpha', and number of categories for discretized gamma heterogeneity ('ncat').
`rate_per_nt`	Flag to indicate whether the mutation rate is per nucleotide (nt) or per sequence. If per-sequence ('rate_per_nt=FALSE'), the mutation rate is normalized by the length of the root sequence.

Details

If the root sequence is a character vector, it is flattened into a single string.

See Also

geneology_to_phylogeny_bpb

gen_const_phase *Simulate a fixed number of steps*

Description

Usually, we want to simulate a fixed number of steps after the exponential growth phase. This function performs that step.

Usage

```
gen_const_phase(state, parameters, num_steps, verbose = FALSE)
```

gen_exp_phase *Simulate an exponential growth.*

Description

Simulate an exponential growth.

Usage

```
gen_exp_phase(state, parameters)
```

gen_transmission_history_balanced_tree *Generate a transmission history for a given number of individuals under a optimally balanced tree*

Description

Simulate a balanced transmission tree with a given number of individuals. If the ‘population_size‘ is a power of 2, then the tree will be unique up to relabeling the tips. If the ‘population_size‘ is not a power of 2, then the tree that minimizes Sackin's index is not unique. To constrain the solution, we build the tree programmatically. We convert each leaf node to a cherry in the initial layer, to create a new layer of leaves. We repeat until we have a ‘population_size‘ number of leaves. As a result, we will always find the shortest tree with the prescribed number of leaves that is perfectly balanced.

Usage

```
gen_transmission_history_balanced_tree(population_size, spike_root = FALSE)
```

Arguments

n	number of individuals to have in the final layer

gen_transmission_history_exponential_constant *Generate transmission history for full population*

Description

Forward stochastic simulation to generate transmission history for a complete population with an exponential growth then constant population structure.

Usage

```
gen_transmission_history_exponential_constant(
  minimum_population,
  offspring_rate_fn,
  maximum_population_target,
  total_steps,
  spike_root = FALSE
)
```

Details

This simulation is intended for HIV, but may be broadly applicable if the lifespan distribution is adjusted to other distributions/parameters.

get_biphasic_HIV_rate *Biphasic rate function factory*

Description

Biphasic rate function factory

Usage

```
get_biphasic_HIV_rate(params)
```

Arguments

params	list of parameters used.

`get_biphasic_HIV_rate_function` *Factory function to generate biphasic rate functions*

Description

Factory function to generate biphasic rate functions

Usage

```
get_biphasic_HIV_rate_function(
  front_density_factor,
  front_cutoff,
  target_length
)
```

Arguments

`front_density_factor`	How much relative significance to place in the first phase of the rate function.
`front_cutoff`	Length of first phase. An integer 1 or greater.
`target_length`	Expected length of an infection. Used for normalization to ensure an average of $R0$ secondary infections.

Value

Callable, rate function

`get_Kphasic_hiv_rate_function` *Kphasic HIV rate function factory Provide a list of relative importance (multiples over a "base" rate) for*

Description

Kphasic HIV rate function factory Provide a list of relative importance (multiples over a "base" rate) for

Usage

```
get_Kphasic_hiv_rate_function(rate_list, target_length, params)
```

Arguments

rate_list	# List of relative rates for each phase.
target_length	List of lengths of each phase.
params	list with element R0

Value

Callable, rate function

---

get_p17_rate_model *Rate model for gag p17 estimated from the Swedish transmission chain*

Description

Estimates for GTR+I+G rate model parameter from [Leitner et al. 1997] on the Swedish transmission chain on p17. Rate estimates for p17 follow from a secondary analysis.

Usage

```
get_p17_rate_model()
```

Value

A list of rate model parameters.

See Also

generate_sequences generate_rate_model

Examples

```
rate_model <- get_p17_rate_model()
```

---

get_pol_rate_model *Rate model for pol estimated from the Swedish transmission chain*

Description

Estimates for GTR+I+G rate model parameter on the Swedish transmission chain on pol region. Follow up analysis by Lundgren et al. [2022] on the Swedish transmission chain on pol region.

Usage

```
get_pol_rate_model(nonzero_I = FALSE)
```

Arguments

nonzero_I	Boolean indicating whether to use the I parameter (invariant sites) or not. Default is 'TRUE'(0.255). If 'FALSE', the I parameter is set to 0.

Value

A list of rate model parameters.

See Also

generate_sequences generate_rate_model

---

get_V3_rate_model *Rate model for env V3 estimated from the Swedish transmission chain*

Description

Estimates for GTR+I+G rate model parameter from [Leitner et al. 1997] on the Swedish transmission chain on envelope V3 region.

Usage

```
get_V3_rate_model(nonzero_I = TRUE)
```

Arguments

nonzero_I	Boolean indicating whether to use the I parameter (invariant sites) or not. Default is 'TRUE'(0.68). If 'FALSE', the I parameter is set to 0.

Value

A list of rate model parameters.

See Also

generate_sequences generate_rate_model

Examples

```
rate_model <- get_V3_rate_model()
```

initialize *Utility function for initializing the simulation*

Description

Utility function for initializing the simulation

Usage

```
initialize(parameters)
```

keep_samples *Keep a set of samples in the simulation*

Description

Remove all other samples from the simulation. Intended to be used to represent a masking event, Where only a subset of the population is propagated forward in time.

Usage

```
keep_samples(state, samples)
```

Details

Provide a state object and a vector of samples to keep. A new state object is returned with only the samples kept.

lookup_sequence_by_index *Obtain a sequence from the builtin reference sequences using coordinates*

Description

Provide an interval and a reference sequence name. Currently supported are: "HIV1" (HXB2 reference) and "toy" (a poly A/C/G/T example sequence for testing).

Usage

```
lookup_sequence_by_index(organism_name, start, stop)
```

Arguments

start	The start coordinate of the sequence
end	The end coordinate of the sequence

`lookup_sequence_by_name` *Obtain a sequence from the builtin reference sequences using an name and annotated region*

Description

Return a portion of a reference genome using a standard annotation name.

Usage

```
lookup_sequence_by_name(organism_name, region_name)
```

Details

Currently supported are: "HIV1" (HXB2 reference) and "toy" (a poly A/C/G/T example sequence for testing). Supported HIV1 regions include the short annotated list, see (here)[https://www.hiv.lanl.gov/components/sequenc for the full list. Clinical regions (p17, V3) are supported as "p17-clinical" and "v3-clinical".

For a more detailed lookup proceedure with the reference sequences using user-provided coordinates, see 'lookup_sequence_by_index'.

See Also

lookup_sequence_by_index

`never_terminate_early_factory` *Factory function to never terminate contact tracing until all known nodes have been traced*

Description

When contact tracing, we don't want to stop until we have traced all known. This is computationally more expensive, but better describes reality.

Usage

```
never_terminate_early_factory()
```

phylogeny_to_newick *Convert a phylogeny array to a newick tree*

Description

Build a newick tree from a phylogeny or geneology array. Nodes are named "#_". Specify the mode argument to select the branch length mode. Mode "mu" denotes a (sampled) mutation count, while mode "mean" denotes expected distances.

Usage

```
phylogeny_to_newick(phylogeny, mode = "mu", label_mode = "local")
```

Arguments

phylogeny	A phylogeny or geneology array.
mode	String to determine reconstruction mode. Default is "mu", for mutation count. Alternative "mean" for expected distance.
label_mode	String to determine how to label nodes. Default is "local", for node index. Alternative "abs" for absolute index.

random_fixed_size_ids *Sample a fixed number of of individuals randomly from the population.*

Description

A helper function for sampling a fixed number of individuals from the population. Select a fixed number of individuals from the active set. Does not use any information about the transmission history or the active population size to select the sample.

Usage

```
random_fixed_size_ids(active, minimum_size, spike_root = FALSE)
```

Arguments

active	A vector of active individuals
minimum_size	The minimum number of individuals to sample.
spike_root	Whether to include the root in the sample.

See Also

random_prop_ids

`random_prop_ids` *Sample a proportion of the active individuals*

Description

A helper function for sampling a proportion of the active individuals. Select a proportion of individuals from the active set. Does not use any information about the transmission history to select the sample.

Usage

```
random_prop_ids(active, proportion, minimum_size, spike_root = FALSE)
```

Arguments

`active`	A vector of active individuals
`proportion`	A float between 0 and 1. The proportion of active individuals to sample.
`minimum_size`	The minimum number of individuals to sample. If the proportional sample is smaller than the minimum size, the proprotional size is used.
`spike_root`	Whether to include the root in the sample.

See Also

random_fixed_size_ids

`reduce_large_matrix` *Reduce a large matrix by randomly sampling a cluster of closely related individuals*

Description

An obtained sample (through contact tracing or random sampling) may be larger than needed. This function extracts a subset of closely related individuals with a randomly selected "center". This respects the ‘spike_root‘ option, and if ‘spike_root = TRUE‘, will return the root individual in the last column.

Usage

```
reduce_large_matrix(
  oversampled_matrix,
  subsample_size,
  spike_root = FALSE,
  root_position = 0,
  index_id = -1,
  sort_order = NULL
)
```

`reduce_transmission_history` *Reduce simulation output to transmission history for a subset.*

Description

Reduce a large simulation output to a smaller transmission history for a subset by tracing back the ancestory of each individual in the sample. If ‘spike_root‘ is ‘TRUE‘, then the root of the tree is included in the geneology. We call this a geneology, rather than a phylogeny as it assumes that the transmission history is the phylogeny and no within-host diversity occurs.

Usage

```
reduce_transmission_history(samples, parents, current_step, spike_root = FALSE)
```

Arguments

`samples`	A vector of individuals (integers) to include in the sample.
`parents`	A matrix of parental individuals that encodes the transmission history and sample times.
`current_step`	The current (absolute) time step in the simulation.
`spike_root`	A boolean indicating whether the geneology should include the root of the outbreak or not. Default is ‘FALSE‘. This should be specified even if the founding infection is sampled, as the root of the outbreak will have evolved since the founding event.

Details

To include within host diversity, use ‘reduce_transmission_history_bpb‘ to extract a subst of the transmission history that we need for the phylogeny, and see ‘geneology_to_phylogeny_bpb‘ to simulate within-host diversity and recover a true phylogeny.

Value

A list with 1 element: "geneology" a matrix of transmission history that encodes an evolutionary tree.

See Also

reduce_transmission_history_bpb

`reduce_transmission_history_bpb` *Reduce simulation output to transmission history for a subset to include within host diversity.*

Description

For a detailed explenation of inputs, see 'reduce_transmission_history', which is intended to reconstruct back only until the most recent common ancestor of the sample, and return a tree.

Usage

```
reduce_transmission_history_bpb(samples, parents, current_step)
```

Arguments

`samples`	A vector of individuals (integers) to include in the sample.
`parents`	A matrix of parental individuals that encodes the transmission history and sample times.
`current_step`	The current (absolute) time step in the simulation.
`spike_root`	A boolean indicating whether the geneology should include the root of the outbreak or not. Default is 'FALSE'. This should be specified even if the founding infection is sampled, as the root of the outbreak will have evolved since the founding event.

Details

To include within host diversity, use 'reduce_transmission_history_bpb' to extract a subset of the transmission history that we need for the phylogeny, and see 'geneology_to_phylogeny_bpb' to simulate within-host diversity and recover a true phylogeny.

Value

A list with 4 elements: 'parents' A vector of parents of each infection in the sample until the root 'times' A vector of times of sampling times in the tree. Sample times for internal nodes are after the last offspring generation time needed to reconstruct the sample. 'transmission_times' A vector of transmission times of each infection in the tree. 'samples_available' A boolean vector (mask) of which samples are leaves in the tree. Used by the coalescent simulation to know which individuals should be assigned detected sequences.

See Also

reduce_transmission_history_bpb

`reduce_transmission_history_mt` *Reduce simulation output to transmission history for a subset.*

Description

Reduce a large simulation output to a smaller transmission history for a subset by tracing back the ancestory of each individual in the sample. If ‘spike_root‘ is ‘TRUE‘, then the root of the tree is included in the geneology. We call this a geneology, rather than a phylogeny as it assumes that the transmission history is the phylogeny and no within-host diversity occurs.

Usage

```
reduce_transmission_history_mt(
  samples,
  parents,
  current_step,
  spike_root = FALSE
)
```

Arguments

`samples`	A vector of individuals (integers) to include in the sample.
`parents`	A matrix of parental individuals that encodes the transmission history and sample times.
`current_step`	The current (absolute) time step in the simulation.
`spike_root`	A boolean indicating whether the geneology should include the root of the outbreak or not. Default is ‘FALSE‘. This should be specified even if the founding infection is sampled, as the root of the outbreak will have evolved since the founding event.

Details

To include within host diversity, use ‘reduce_transmission_history_bpb‘ to extract a subst of the transmission history that we need for the phylogeny, and see ‘geneology_to_phylogeny_bpb‘ to simulate within-host diversity and recover a true phylogeny.

Value

A list with 1 element: "geneology" a matrix of transmission history that encodes an evolutionary tree.

See Also

reduce_transmission_history_bpb

`remove_samples` *Remove samples from the simulation*

Description

Remove a set of samples from the simulation. Intended to be used with contact tracing, we expect individuals identified through contact tracing to be removed from the population of active (uncontrolled or undiagnosed) cases.

Usage

```
remove_samples(state, samples)
```

Details

Provide a state object and a vector of samples to remove. A new state object is returned with the samples removed.

`show_available_regions` *Show the list of available regions for a provided reference sequence*

Description

Show the list of available regions for a provided reference sequence

Usage

```
show_available_regions(organism_name, chart = TRUE)
```

Arguments

`organism_name`	The name of the reference sequence
`chart`	bool (default TRUE). If TRUE, a chart of the available regions is printed to the screen.

Value

A list of available regions for the named reference sequence. If the organism_name is not found, an NULL is printed.

`show_available_sequences`
Show the list of available reference sequences

Description

A list of all available organism/sequence names is printed to the screen.

Usage

```
show_available_sequences()
```

`simulate_all_paradigms_HIV_V3`
Sample distance matrices for all three major simulation paradigms: the transmission history, the phylogeny, and derived from simulated sequences (TN93)

Description

This simulation returns three distance matrices for the same set of individuals. The first matrix (matrix_trans) is derived from the transmission history. The second matrix (matrix_phylo) is derived from the phylogeny by simulating a coalescent process on the transmission history. The third matrix (matrix_seq) is derived from the sequences by computing the pairwise distances using the TN93 model.

Usage

```
simulate_all_paradigms_HIV_V3(params)
```

Value

list with keys "matrix" and "input_params"

Examples

```
parameters <- list("rate_function_parameters" = list("R0"=5), "a"=5, "b"=5,
                   "mutation_rate" = 0.0067,  # units are per nt
                   "contact_tracing_discovery_probability" = 0.9,
                   "minimum_population"=15, "maximum_population_target"=500)
```

simulate_classic_HIV *Classic HIV simulator for population level simulations.*

Description

A simple agent-based simulation for HIV. Returns pairwise distance matrix samples obtained from a simulated outbreak. This simulation models population level diversity, but does not explicitly consider within-host diversity. A mutation clock rate, R0, and several epidemiological parameters are required to describe the duration of infections.

Usage

```
simulate_classic_HIV(params)
```

Value

list with key "matrix"

Examples

```
parameters <- list("rate_function_parameters" <- list("R0"=5),
                   "mutation_rate" = 0.0067 / 12 * 300,
                   "minimum_population"=50, "maximum_population_target"=100)
```

simulate_modern_HIV *Modern HIV simulator*

Description

A modern agent-based simulation for HIV that explicitly models within-host simulations using biophybreak extensions.

Usage

```
simulate_modern_HIV(params)
```

Value

list with keys "matrix" and "input_params"

Examples

```
parameters <- list("rate_function_parameters" = list("R0"=5), "a"=5, "b"=5,
                   "minimum_population"=15, "maximum_population_target"=500)
```

simulate_sequences_HIV_V3 *Modern HIV epidemic sequences simulator*

Description

A modern agent-based simulation for HIV that explicitly models within-host simulations using biophybreak extensions. This pre-build simulation returns a collection of sequences, rather than a matrix.

Usage

```
simulate_sequences_HIV_V3(params)
```

Value

list with keys "matrix" and "input_params"

Examples

```
parameters <- list("rate_function_parameters" = list("R0"=5), "a"=5, "b"=5,
                   "mutation_rate" = 0.0067,  # units are per nt
                   "contact_tracing_discovery_probability" = 0.9,
                   "minimum_population"=15, "maximum_population_target"=500)
```

step *Perform a single step of the simulation*

Description

Perform a single step of the simulation

Usage

```
step(state, parameters)
```

`stochastify_transmission_history` *Sample evolutionary distances to an integer number of mutations per branch*

Description

Convert temporal distances for a geneology to an integer number of mutations. This step is extracted so a single geneology can be re-sampled and create multiple possible matrices.

Usage

```
stochastify_transmission_history(transmission_history, rate)
```

Arguments

`transmission_history`	A matrix output by ‘reduce transmission history‘.
`rate`	Clock rate for evolutionary process with units of substitions per sequence per year.

`sufficient_data_data_factory` *Factory function for a detection to check termination condition based on A sufficient amount of individuals Terminate when we have found a ‘minimum_size‘ number of active individuals*

Description

Factory function for a detection to check termination condition based on A sufficient amount of individuals Terminate when we have found a ‘minimum_size‘ number of active individuals

Usage

```
sufficient_data_data_factory(minimum_size)
```

Arguments

`minimum_size`	The minimum number of discovered active individuals to terminate

uniform_discovery_factory *A factory function to discover connections with uniform probability*

Description

Obtain a discovery function that determines which adjacent nodes in the contact network will be revealed with uniform probability.

Usage

```
uniform_discovery_factory(p)
```

Arguments

p	The discovery probability. A float between 0 and 1.

See Also

contact_traced_uniform_ids

validate_state *Utility function to check that a state is (computationally) valid*

Description

Check that the state object is a valid and has the correct list elements.

Usage

```
validate_state(state)
```

Index