Package ‘DCLEAR’

July 15, 2021

Version 1.0.0
Date 2021-07-13
Title Distance Based Cell Lineage Reconstruction
Author Il-Youp Kwak [aut, cre],
       Wuming Gong [aut]
Maintainer Il-Youp Kwak <ikwak2@cau.ac.kr>
License GPL-3
VignetteBuilder knitr
LinkingTo Rcpp, RcppArmadillo
RoxygenNote 7.1.1
Depends R (>= 3.6.0)
biocViews
Encoding UTF-8
Imports BiocParallel, dplyr, Matrix, matrixStats, ape, phangorn, Rcpp,
       igraph, methods, purrr, stringr, tidyr, tidyverse,
       rBayesianOptimization, rlang
Suggests knitr, rmarkdown, markdown
Description R codes for distance based cell lineage reconstruction. Our methods won both sub-

URL https://github.com/ikwak2/DCLEAR
NeedsCompilation yes
Repository CRAN
Date/Publication 2021-07-15 07:30:02 UTC

R topics documented:

   as_phylo ................................................................. 2
   as_phylo.igraph-method .............................................. 3
as_phylo

Generic function for as_phylo

Description

Generic function for as_phylo

Usage

as_phylo(x, ...)

Arguments

x a graph object

... additional parameters

Value

a phylo object
as_phylo,igraph-method

Description
Convert an igraph object to a phylo object

Usage
## S4 method for signature 'igraph'
as_phylo(x)

Arguments
x an igraph object

Value
a phylo object

DCLEAR

DCLEAR: A package for DCLEAR: Distance based Cell LineAge Reconstruction

Description
Distance based methods for inferring lineage trees from single cell data

dist_kmer_replacement_inference

Core function of computing kmer replacement distance

Description
Compute the sequence distance matrix using inferred kmer replacement matrix

Usage
dist_kmer_replacement_inference(x, kmer_summary, k = 2)
Arguments

- `x`: input data in phyDat format
- `kmer_summary`: a kmer_summary object
- `k`: k-mers (default k=2)

Value

- a dist object

Author(s)

- Wuming Gong (gongx030@umn.edu)

Description

Generic function for dist_replacement

Usage

- `dist_replacement(x, kmer_summary, k, ...)`

Arguments

- `x`: a sequence object
- `kmer_summary`: a kmer_summary object
- `k`: k-mer length
- `...`: additional parameters

Value

- a dist object
**dist_replacement,phyDat,kmer_summary,integer-method**

*Compute the kmer replacement distance*

**Description**

Compute the kmer replacement distance between sequences

**Usage**

```r
## S4 method for signature 'phyDat,kmer_summary,integer'
dist_replacement(x, kmer_summary, k = 2, ...)
```

**Arguments**

- **x**: input data in phyDat format
- **kmer_summary**: a kmer_summary object
- **k**: k-mer length
- **...**: other arguments passed to substr_kmer

**Value**

a dist object

**Author(s)**

Wuming Gong (gongx030@umn.edu)
Arguments

x  input data in phyDat format
kmer_summary  a kmer_summary object
k  k-mer length
...  other arguments passed to substr_kmer

Value

a dist object

Author(s)

Wuming Gong (gongx030@umn.edu)

dist_weighted_hamming  Generic function for dist_weighted_hamming

Description

Generic function for dist_weighted_hamming

Usage

dist_weighted_hamming(x, wVec, ...)

Arguments

x  a sequence object
wVec  weight vector
...  additional parameters

Value

Calculated distance matrix of input sequences. The result is a 'dist' class object.
Description

implementation of weighted hamming algorithm

Usage

## S4 method for signature 'phyDat,numeric'

```r
dist_weighted_hamming(x, wVec, dropout = FALSE)
```

Arguments

- **x**: Sequence object of 'phyDat' type.
- **wVec**: Weight vector for the calculation of weighted hamming distance
- **dropout**: Different weighting strategy is taken to consider interval dropout with dropout = 'TRUE'. Default is, dropout = 'FALSE'.

Value

Calculated distance matrix of input sequences. The result is a 'dist' class object.

Author(s)

Il-Youp Kwak

Examples

```r
library(DCLEAR)
library(phangorn)
library(ape)

set.seed(1)
mu_d1 = c(30, 20, 10, 5, 5, 1, 0.01, 0.001)
mu_d1 = mu_d1/sum(mu_d1)
simn = 10 # number of cell samples
m = 10 # number of targets
sD = sim_seqdata(sim_n = simn, m = m, mu_d = 0.03,
                 d = 12, n_s = length(mu_d1), outcome_prob = mu_d1, p_d = 0.005 )
## RF score with hamming distance
D_hm = dist.hamming(sD$seqs)
tree_hm = NJ(D_hm)
RF.dist(tree_hm, sD$tree, normalize = TRUE)
## RF score with weighted hamming
```
get_distance_prior

Description

prior distribution of distance

Usage

get_distance_prior(x)

Arguments

x  
a kmer_summary object

Value

a probabilistic vector of the distribution of nodal distances

Author(s)

Wuming Gong (gongx030@umn.edu)
get_node_names

Description
Convenient function for get node names

Usage
get_node_names(x)

Arguments
x node id

Value	node names

Author(s)
Wuming Gong (gongx030@umn.edu)

get_replacement_probability

Description
Compute p(A,B|d), the conditional probability of seeing a replacement of from kmer A to B or vice versa

Usage
get_replacement_probability(x)

Arguments
x a kmer_summary object

Value
an 3D probabilistic array (kmers by kmers by distances)

Author(s)
Wuming Gong (gongx030@umn.edu)
get_transition_probability

Description
Compute \( p(A,X|B,Y,d) \), the conditional probability of seeing a replacement from \( A \) to \( B \) given the previous replacement \( B \) from \( Y \) at nodal distance \( d \).

Usage
get_transition_probability(x)

Arguments
x a kmer_summary object

Value
an 3D probabilistic array (kmer by kmer by distances)

Author(s)
Wuming Gong (gongx030@umn.edu)

lineages

Description
Lineage data

Usage
data(lineages)

Format
An object of class list of length 100.

Examples
data(lineages)
process_sequence

Generic function for process_sequence

Description

Generic function for process_sequence

Usage

process_sequence(x, ...)

Arguments

x

a sequence object

... additional parameters

Value

a 'lineage_tree_config' object

process_sequence,phyDat-method

Process sequences

Description

Process sequences

Usage

## S4 method for signature 'phyDat'
process_sequence(x, division = 16L)

Arguments

x

input data in phyDat format
division

cell divisions (default: 16L)

Value

a 'lineage_tree_config' object

Author(s)

Wuming Gong (gongx030@umn.edu)
### random_tree

**Description**

Simulate a random lineage tree

**Usage**

```r
random_tree(n_samples, division = 16L)
```

**Arguments**

- `n_samples`: number of samples to simulate
- `division`: number of cell division

**Value**

a data frame

**Author(s)**

Wuming Gong (gongx030@umn.edu)

---

### sample_outcome_prob

**Description**

Sampling outcome probability based on a gamma distribution

**Usage**

```r
sample_outcome_prob(config, num_states = 20L, shape = 0.1, scale = 2)
```

**Arguments**

- `config`: a lineage_tree_config object
- `num_states`: number of states used in simulation.
- `shape`: shape parameter in gamma distribution
- `scale`: scale parameter in gamma distribution

**Value**

a probability vector for each alphabet
simulate

Generic function for simulate

Description

Generic function for simulate

Usage

simulate(config, ...)

Arguments

config a lineage_tree_config object

... additional parameters

Value

a lineage_tree object

simulate, lineage_tree_config-method

simulate

Description

Simulate a cell lineage tree Adopted from https://github.com/elifesciences-publications/CRISPR_recorders_sims/blob/master/MATLAB_sims/GESTALT_30hr_1x_simulation.m

Usage

## S4 method for signature 'lineage_tree_config'
simulate(config, n_samples = 200)

Arguments

config simulation configuration; a lineage_tree_config object

n_samples number of samples to simulate

Value

a lineage_tree object

Author(s)

Wuming Gong (gongx030@umn.edu)
Description

Generate single cell barcode data set with tree shaped lineage information

Usage

sim_seqdata(
  sim_n = 200,
  m = 200,
  mu_d = 0.03,
  d = 15,
  n_s = 23,
  outcome_prob = NULL,
  p_d = 0.003
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sim_n</td>
<td>Number of cell samples to simulate.</td>
</tr>
<tr>
<td>m</td>
<td>Number of targets.</td>
</tr>
<tr>
<td>mu_d</td>
<td>Mutation rate. (a scalar or a vector)</td>
</tr>
<tr>
<td>d</td>
<td>Number of cell divisions.</td>
</tr>
<tr>
<td>n_s</td>
<td>Number of possible outcome states</td>
</tr>
<tr>
<td>outcome_prob</td>
<td>Outcome probability vector (default is NULL)</td>
</tr>
<tr>
<td>p_d</td>
<td>Dropout probability</td>
</tr>
</tbody>
</table>

Value

The result is a list containing two objects, 'seqs' and 'tree'. The 'seqs' is a 'phyDat' object of 'sim_n' number of simulated barcodes corresponding to each cell, and The 'tree' is a 'phylo' object, a ground truth tree structure for the simulated data.

Author(s)

Il-Youp Kwak

Examples

library(DCLEAR)
library(phangorn)
library(ape)
set.seed(1)
mu_d1 = c( 30, 20, 10, 5, 5, 1, 0.01, 0.001)
mu_d1 = mu_d1/sum(mu_d1)
simn = 10 # number of cell samples
m = 10  ## number of targets
sD = sim_seqdata(sim_n = simn, m = m, mu_d = 0.03,
    d = 12, n_s = length(mu_d1), outcome_prob = mu_d1, p_d = 0.005)
## RF score with hamming distance
D_hm = dist.hamming(sD$seqs)
tree_hm = NJ(D_hm)
RF.dist(tree_hm, sD$tree, normalize = TRUE)

## RF score with weighted hamming
InfoW = -log(mu_d1)
InfoW[1:2] = 1
D_wh = dist_weighted_hamming(sD$seqs, InfoW, dropout=FALSE)
tree_wh = NJ(D_wh)
RF.dist(tree_wh, sD$tree, normalize = TRUE)

## RF score with weighted hamming, considering dropout situation
nfoW = -log(mu_d1)
InfoW[1] = 1
InfoW[2] = 12
InfoW[3:7] = 3
D_wh2 = dist_weighted_hamming(sD$seqs, InfoW, dropout = TRUE)
tree_wh2= NJ(D_wh2)
RF.dist(tree_wh2, sD$tree, normalize = TRUE)

---

**substr_kmer**

*Generic function for substr_kmer*

**Description**

Generic function for substrate_kmer

**Usage**

`substr_kmer(x, ...)`

**Arguments**

- `x` a kmer object
- `...` additional parameters

**Value**

a new kmer_summary object
substr_kmer, kmer_summary-method

Subsetting a kmer_summary object

Description

Summarize the short k-mer summary from the long k-mer summary

Usage

## S4 method for signature 'kmer_summary'
substr_kmer(x, k = 2)

Arguments

x a kmer_summary object
k k-mer length (default: 2)

Value

da new kmer_summary object

Author(s)

Wuming Gong (gongx030@umn.edu)

summarize_kmer

Generic function for summarize_kmer

Description

Generic function for summarize_kmer

Usage

summarize_kmer(x, ...)

Arguments

x a sequence object
... additional parameters

Value

da kmer_summary object
Description

Summarize kmer distributions with input sequences

Usage

```r
## S4 method for signature 'phyDat'
summarize_kmer(
  x,
  division = 16L,
  k = 2,
  reps = 20L,
  n_samples = 200L,
  n_nodes = 100L,
  n_targets
)
```

Arguments

- `x`: input data as a phyDat object
- `division`: number of cell division
- `k`: k-mer (default = 2)
- `reps`: number of simulated trees
- `n_samples`: number of samples to simulate
- `n_nodes`: number of nodes to sample (including both leaves and internal nodes)
- `n_targets`: sequence length. If this argument is missing, the length of the input sequences will be used.

Value

- a kmer_summary object

Author(s)

Wuming Gong (gongx030@umn.edu)
summarize_kmer_core  summarize_kmer_core

Description

Summarize kmer distributions (core function)

Usage

```r
summarize_kmer_core(
  k = 2,
  reps = 20L,
  n_samples = 200L,
  n_nodes = 100L,
  config = NULL
)
```

Arguments

- `k`  
  k-mer (default = 2)
- `reps`  
  number of simulated trees
- `n_samples`  
  number of samples to simulate
- `n_nodes`  
  number of nodes to sample (including both leaves and internal nodes)
- `config`  
  lineage tree configuration (a lineage_tree_config object)

Value

a kmer_summary object

Author(s)

Wuming Gong (gongx030@umn.edu)

---

WH  WH

Description

implementation of weighted hamming algorithm

Usage

```r
WH(x, InfoW, dropout = FALSE)
```
Arguments

- **x**: Sequence object of 'phyDat' type.
- **InfoW**: Weight vector for the calculation of weighted hamming distance
- **dropout**: Different weighting strategy is taken to consider interval dropout with dropout = 'TRUE'. Default is, dropout = 'FALSE'.

Value

Calculated distance matrix of input sequences. The result is a 'dist' class object.

Author(s)

Il-Youp Kwak

Examples

```r
set.seed(1)
library(phangorn)
mu_d1 = c(30, 20, 10, 5, 1, 0.01, 0.001)
mu_d1 = mu_d1/sum(mu_d1)
simn = 10 # number of cell samples
m = 10 ## number of targets
sD = sim_seqdata(sim_n = simn, m = m, mu_d = 0.03,
                 d = 12, n_s = length(mu_d1), outcome_prob = mu_d1, p_d = 0.005 )

## RF score with hamming distance
D_h = dist.hamming(sD$seqs)
tree_h = NJ(D_h)
RF.dist(tree_h, sD$tree, normalize = TRUE)

## RF score with weighted hamming
InfoW = -log(mu_d1)
InfoW[1:2] = 1

D_wh = WH(sD$seqs, InfoW)
tree_wh = NJ(D_wh)
RF.dist(tree_wh, sD$tree, normalize = TRUE)

## RF score with weighted hamming, considering dropout situation
InfoW = -log(mu_d1)
InfoW[1] = 1
InfoW[2] = 12
InfoW[3:7] = 3

D_wh2 = WH(sD$seqs, InfoW, dropout=TRUE)
tree_wh2 = NJ(D_wh2)
RF.dist(tree_wh2, sD$tree, normalize = TRUE)
```
**WH_train**  
*Train weights for WH*

**Description**

Train weights for WH and output weight vector

**Usage**

```r
WH_train(X)
```

**Arguments**

- `X`: a list of k number of input data, X[[1]] ... X[[k]]. The ith data have sequence information as phyDat format in X[[i]][[1]], and tree information in X[[i]][[2]] as phylo format.

**Value**

a weight vector

**Author(s)**

Il-Youp Kwak (ikwak2@cau.ac.kr)

---

**WH_train_fit**  
*Train weights for WH, and output distance object*

**Description**

Train weights for WH using the given data, and fit the distance matrix for a input sequence.

**Usage**

```r
WH_train_fit(x, X)
```

**Arguments**

- `x`: input data in phyDat format
- `X`: a list of k number of input data, X[[1]] ... X[[k]]. The ith data have sequence information as phyDat format in X[[i]][[1]], and tree information in X[[i]][[2]] as phylo format.

**Value**

a dist object
Author(s)

Il-Youp Kwak (ikwak2@cau.ac.kr)
Index

* datasets
  lineages, 10

as_phylo, 2
as_phylo, igraph-method, 3

DCLEAR, 3
dist_kmer_replacement_inference, 3
dist_replacement, 4
dist_replacement, phyDat, kmer_summary, integer-method,
  5
dist_replacement, phyDat, missing, integer-method,
  5
dist_weighted_hamming, 6
dist_weighted_hamming, phyDat, numeric-method,
  7

get_distance_prior, 8
get_node_names, 9
get_replacement_probability, 9
get_transition_probability, 10

lineages, 10

process_sequence, 11
process_sequence, phyDat-method, 11

random_tree, 12

sample_outcome_prob, 12
sim_seqdata, 14
simulate, 13
simulate, lineage_tree_config-method,
  13
substr_kmer, 15
substr_kmer, kmer_summary-method, 16
summarize_kmer, 16
summarize_kmer, phyDat-method, 17
summarize_kmer_core, 18

WH, 18
WH_train, 20
WH_train_fit, 20