Package 'SITH'

October 12, 2022

Type Package

Title A Spatial Model of Intra-Tumor Heterogeneity

Version 1.1.0

Date 2021-01-03

Author Phillip B. Nicol

Maintainer Phillip B. Nicol <philnicol740@gmail.com>

Description Implements a three-dimensional stochastic model of cancer growth and mutation similar to the one described in Waclaw et al. (2015) <doi:10.1038/nature14971>. Allows for interactive 3D visualizations of the simulated tumor. Provides a comprehensive summary of the spatial distribution of mutants within the tumor. Contains functions which create synthetic sequencing datasets from the generated tumor.

License GPL (>= 2)

Depends R (>= 3.6.0)

Imports Rcpp (>= 1.0.4), scatterplot3d, stats, graphics, grDevices

Suggests rgl, igraph, knitr, rmarkdown, testthat

LinkingTo Rcpp

RoxygenNote 7.1.1

VignetteBuilder knitr

Encoding UTF-8

URL https://github.com/phillipnicol/SITH

BugReports https://github.com/phillipnicol/SITH/issues

NeedsCompilation yes

Repository CRAN

Date/Publication 2021-01-05 15:10:02 UTC

R topics documented:

`SITH-package` *Visualize and analyze intratumor heterogeneity using a spatial model of tumor growth*

Description

The **SITH** (spatial model of intratumor heterogeneity) package implements a lattice based spatial model of tumor growth and mutation. Interactive 3D visualization of the tumor are possible using **rgl**. Additional functions for visualization and investigating the spatial distribution of mutants are provided. **SITH** also provides functions to simulate single cell sequencing and bulk sampling data sets from the simulated tumor.

Background

On-lattice models of tumor growth and mutation are computationally efficient and provide a simple setting to study how spatially constrained growth impacts intratumor heterogeneity. While this model has been studied extensively in literature (see Waclaw (2015), Chkhaidze (2019), Opasic (2019)), existing software is either not publicly available or inconvenient to use with R.

The motivation for creating the **SITH** package was to provide a spatial simulator that is both easy to use and can be used entirely with R. The core function in the package is `simulateTumor()`, which wraps a C++ implementation of the model into R using the **Rcpp** package. Once the results of the simulation are saved as an R object, **SITH** provides several other useful functions for studying this model.

See the package vignette for more information on the model and the algorithm used.

Author(s)

Phillip B. Nicol

References

B. Waclaw, I. Bozic, M. Pittman, R. Hruban, B. Vogelstein and M. Nowak. A spatial model predicts that dispersal and cell turnover limit intratumor heterogeneity. *Nature*, pages 261-264, 2015. https://doi.org/10.1038/nature14971.

K. Chkhaidze, T. Heide, B. Werner, M. Williams, W. Huang, G. Caravagna, T. Graham, and A. Sottoriva. Spatially constrained tumour growth affects the patterns of clonal selection and neutral drift in cancer genomic data. *PLOS Computational Biology*, 2019. https://doi.org/10.1371/journal.pcbi.1007243.

L. Opasic, D. Zhou, B. Wener, D. Dingli and A. Traulsen. How many samples are needed to infer truly clonal mutations from heterogeneous tumours? *BMC Cancer*, https://doi.org/10.1186/s12885-019-5597-1.

Examples

```
#Simulate tumor growth
out <- simulateTumor()

#3d interactive visualization using rgl
visualizeTumor(out)
#or see regions with lots of mutants
visualizeTumor(out, plot.type = "heat")

#get a summary on the spatial dist. of mutants
sp <- spatialDistribution(out)

#simulate single cell sequencing
Scs <- randomSingleCells(tumor = out, ncells = 5, fnr = 0.1)

#simulate bulk sampling
Bulks <- randomBulkSamples(tumor = out, nsamples = 5)
```

`bulkSample` *Simulate bulk sampling*

Description

Simulate bulk sequencing data by taking a local sample from the tumor and computing the variant allele frequencies of the various mutations.

Usage

```
bulkSample(tumor, pos, cube.length = 5, threshold = 0.05, coverage = 0)
```

Arguments

tumor	A list which is the output of simulateTumor().
pos	The center point of the sample.
cube.length	The side length of the cube of cells to be sampled.
threshold	Only mutations with an allele frequency greater than the threshold will be included in the sample.
coverage	If nonzero then deep sequencing with specified coverage is performed.

Details

A local region of the tumor is sampled by constructing a cube with side length cube.length around the center point pos. Each cell within the cube is sampled, and the reported quantity is variant (or mutation) allele frequency. Lattice sites without cells are assumed to be normal tissue, and thus the reported MAF may be less than 1.0 even if the mutation is present in all cancerous cells.

If coverage is non-zero then deep sequencing can be simulated. For a chosen coverage C, it is known that the number of times the base is read follows a $Pois(C)$ distribution (see Illumina's website). Let d be the true coverage sampled from this distribution. Then the estimated VAF is drawn from a $Bin(d, p)/d$ distribution.

Note that cube.length is required to be an odd integer (in order to have a well-defined center point).

Value

A data frame with 1 row and columns corresponding to the mutations. The entries are the mutation allele frequency.

Author(s)

Phillip B. Nicol

References

K. Chkhaidze, T. Heide, B. Werner, M. Williams, W. Huang, G. Caravagna, T. Graham, and A. Sottoriva. Spatially con- strained tumour growth affects the patterns of clonal selection and neutral drift in cancer genomic data. PLOS Computational Biology, 2019. https://doi.org/10.1371/journal.pcbi.1007243.
Lander ES, Waterman MS.(1988) Genomic mapping by fingerprinting random clones: a mathematical analysis, Genomics 2(3): 231-239.

Examples

```
set.seed(116776544, kind = "Mersenne-Twister", normal.kind = "Inversion")
out <- simulateTumor(max_pop = 1000)
df <- bulkSample(tumor = out, pos = c(0,0,0))
```

plotSlice *2D cross section of the simulated tumor*

Description

2D cross section of the simulated tumor.

Usage

```
plotSlice(tumor, slice.dim = "x", level = 0, plot.type = "normal")
```

Arguments

tumor	A list which is the output of simulateTumor().
slice.dim	One of "x", "y" or "z", which denotes the dimension which will be fixed to obtain a 2D cross section.
level	Which value will the dimension given in slice.dim be fixed at?
plot.type	Which type of plot to draw. "Normal" assigns a random rgb value to each genotype while "heat" colors cells with more mutations red and cells with fewer mutations blue. This is exactly the same as plot.type in visualizeTumor.

Value

None.

Author(s)

Phillip B. Nicol

progressionChain *Create a linear chain graph to describe the order of mutations*

Description

A helper function for simulateTumor() which returns to the user the edge list for a linear chain.

Usage

```
progressionChain(n)
```

Arguments

n	Number of vertices in the chain

Value

A matrix with 4 columns and n-1 rows which will be accepted as input to `simulateTumor()`.

Author(s)

Phillip B. Nicol <philnicol740@gmail.com>

Examples

```
G <- progressionChain(3)
```

`progressionDAG_from_igraph`
Define the progression of mutations from an `igraph` *object*

Description

A helper function for `simulateTumor()` which returns to the user the edge list for a DAG which is defined as an `igraph` object.

Usage

```
progressionDAG_from_igraph(iG)
```

Arguments

`iG`	An igraph object for a directed acyclic graph.

Value

A matrix with 4 columns which contains the edges of the graph as well as the rate of crossing each edge and the selective advantage/disadvantage obtained by crossing each edge.

Author(s)

Phillip B. Nicol <philnicol740@gmail.com>

`randomBulkSamples` *Simulate multi-region bulk sampling*

Description

Simulate bulk sequencing data by taking a local sample from the tumor and computing the variant allele frequencies of the various mutations.

Usage

```
randomBulkSamples(
  tumor,
  nsamples,
  cube.length = 5,
  threshold = 0.05,
  coverage = 0
)
```

Arguments

`tumor`	A list which is the output of `simulateTumor()`.
`nsamples`	The number of bulk samples to take.
`cube.length`	The side length of the cube of cells to be sampled.
`threshold`	Only mutations with an allele frequency greater than the threshold will be included in the sample.
`coverage`	If nonzero then deep sequencing with specified coverage is performed.

Details

This is the same as `bulkSample()`, except multiple samples are taken with random center points.

Value

A data frame with `nsamples` rows and columns corresponding to the mutations. The entries are the mutation allele frequency.

Author(s)

Phillip B. Nicol

Examples

```
out <- simulateTumor(max_pop = 1000)
df <- randomBulkSamples(tumor = out, nsamples = 5, cube.length = 5, threshold = 0.05)
```

`randomNeedles` *Simulate fine needle aspiration*

Description

Simulate a sampling procedure which takes a fine needle through the simulated tumor and reports the mutation allele frequency of the sampled cells.

Usage

```
randomNeedles(tumor, nsamples, threshold = 0.05, coverage = 0)
```

Arguments

`tumor`	A list which is the output of `simulateTumor()`.
`nsamples`	The number of samples to take.
`threshold`	Only mutations with an allele frequency greater than the threshold will be included in the sample.
`coverage`	If nonzero then deep sequencing with specified coverage is performed.

Details

This sampling procedure is inspired by Chkhaidze et. al. (2019) and simulates fine needle aspiration. A random one-dimensional cross-section of the tumor is chosen, and the cells within this cross section are sampled, reporting mutation allele frequency.

Author(s)

Phillip B. Nicol

References

K. Chkhaidze, T. Heide, B. Werner, M. Williams, W. Huang, G. Caravagna, T. Graham, and A. Sottoriva. Spatially con- strained tumour growth affects the patterns of clonal selection and neutral drift in cancer genomic data. PLOS Computational Biology, 2019. https://doi.org/10.1371/journal.pcbi.1007243.

Examples

```
out <- simulateTumor(max_pop = 1000)
df <- randomNeedles(tumor = out, nsamples = 5)
```

`randomSingleCells` *Simulate single cell sequencing data*

Description

Simulate single cell sequencing data by random selecting cells from the tumor.

Usage

```
randomSingleCells(tumor, ncells, fpr = 0, fnr = 0)
```

Arguments

`tumor`	A list which is the output of `simulateTumor()`.
`ncells`	The number of cells to sample.
`fpr`	The false positive rate
`fnr`	The false negative rate

Details

The procedure is exactly the same as `singleCell()` except that it allows multiple cells to be sequenced at once (chosen randomly throughout the entire tumor).

Value

A data frame with sample names on the row and mutation ID on the column. A 1 indicates that the mutation is present in the cell and a 0 indicates the mutation is not present.

Author(s)

Phillip B. Nicol <philnicol740@gmail.com>

Examples

```
out <- simulateTumor(max_pop = 1000)
df <- randomSingleCells(tumor = out, ncells = 5, fnr = 0.1)
```

`simulateTumor` *Spatial simulation of tumor growth*

Description

Simulate the spatial growth of a tumor with a multi-type branching process on the three-dimensional integer lattice.

Usage

```
simulateTumor(
  max_pop = 250000,
  div_rate = 0.25,
  death_rate = 0.18,
  mut_rate = 0.01,
  driver_prob = 0.003,
  selective_adv = 1.05,
  disease_model = NULL,
  verbose = TRUE
)
```

Arguments

`max_pop`	Number of cells in the tumor.
`div_rate`	Cell division rate.
`death_rate`	Cell death rate.
`mut_rate`	Mutation rate. When a cell divides, both daughter cell acquire $Pois(u)$ genetic alterations
`driver_prob`	The probability that a genetic alteration is a driver mutation.
`selective_adv`	The selective advantage conferred to a driver mutation. A cell with k driver mutations is given birth rate bs^k.
`disease_model`	Edge list for a directed acyclic graph describing possible transitions between states. See `progressionChain()` for an example of a valid input matrix.
`verbose`	Whether or not to print simulation details to the R console.

Details

The model is based upon Waclaw et. al. (2015), although the simulation algorithm used is different. A growth of a cancerous tumor is modeled using an exponential birth-death process on the three-dimensional integer lattice. Each cell is given a birth rate b and a death rate d such that the time until cell division or cell death is exponentially distributed with parameters b and d, respectively. A cell can replicate if at least one of the six sites adjacent to it is unoccupied. Each time cell replication occurs, both daughter cells receive $Pois(u)$ genetic alterations. Each alteration is a driver mutation with some probability du. A cell with k driver mutations is given birth rate bs^k. The simulation begins with a single cell at the origin at time $t = 0$.

The model is simulated using a Gillespie algorithm. See the package vignette for details on how the algorithm is implemented.

Value

A list with components

- `cell_ids` - A data frame containing the information for the simulated cells. (x,y,z) position, allele ID number (note that 0 is the wild-type allele), number of genetic alterations, and Euclidean distance from origin are included.
- `muts` - A data frame consisting of the mutation ID number, the count of the mutation within the population, and the mutation allele frequency (which is the count divided by N).
- `phylo_tree` - A data frame giving all of the information necessary to determine the order of mutations. The parent of a mutation is defined to be the most recent mutation that precedes it. Since the ID 0 corresponds to the initial mutation, 0 does not have any parents and is thus the root of the tree.
- `genotypes` - A data frame containing the information about the mutations that make up each allele. The i-th row of this data frame corresponds to the allele ID $i-1$. The positive numbers in each row correspond to the IDs of the mutations present in that allele, while a -1 is simply a placeholder and indicates no mutation. The count column gives the number of cells which have the specific allele.
- `color_scheme` - A vector containing an assignment of a color to each allele.
- `drivers` - A vector containing the ID numbers for the driver mutations.
- `time` - The simulated time (in days).
- `params` - The parameters used for the simulation.

Author(s)

Phillip B. Nicol <philnicol740@gmail.com>

References

B. Waclaw, I. Bozic, M. Pittman, R. Hruban, B. Vogelstein and M. Nowak. A spatial model predicts that dispersal and cell turnover limit intratumor heterogeneity. *Nature*, pages 261-264, 2015.

D. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, volume 81, pages 2340-2361, 1970.

Examples

```
out <- simulateTumor(max_pop = 1000)
#Take a look at mutants in order of decreasing MAF
sig_muts <- out$muts[order(out$muts$MAF, decreasing = TRUE),]

#Specify the disease model
out <- simulateTumor(max_pop = 1000, disease_model = progressionChain(3))
```

`singleCell` *Simulate single cell sequencing data*

Description

Simulate single cell sequencing data by selecting a cell at a specified position

Usage

```
singleCell(tumor, pos, noise = 0)
```

Arguments

`tumor`	A list which is the output of `simulateTumor()`.
`pos`	A vector of length 3 giving the (x,y,z) coordinates of the cell to sample.
`noise`	The false negative rate.

Details

This function selects the cell at `pos` (error if no cell at specified position exists) and returns the list of mutations present in the cell. Due to technological artifacts, the false negative rate can be quite higher (10-20 percent). To account for this, the `noise` parameter introduces false negatives into the data set at the specified rate.

Value

A data frame with 1 row and columns corresponding to the mutations present in the cell. A 1 indicates that the mutation is detected while a 0 indicates the mutation is not detected.

Author(s)

Phillip B. Nicol <philnicol740@gmail.com>

References

K. Jahn, J. Kupiers and N. Beerenwinkel. Tree inference for single-cell data. Genome Biology, volume 17, 2016. https://doi.org/10.1186/s13059-016-0936-x.

Examples

```
set.seed(1126490984)
out <- simulateTumor(max_pop = 1000)
df <- singleCell(tumor = out, pos = c(0,0,0), noise = 0.1)
```

spatialDistribution *Quantify the spatial distribution of mutants*

Description

Provides a summary the spatial distribution of mutants within the simulated tumor.

Usage

```
spatialDistribution(tumor, N = 500, cutoff = 0.01, make.plot = TRUE)
```

Arguments

tumor	A list which is the output of simulateTumor().
N	The number of pairs to sample.
cutoff	For a plot of clone sizes, all mutations with a MAF below cutoff are ignored.
make.plot	Whether or not to make plots.

Details

The genotype of a cell can be interpreted as a binary vector where the i-th component is 1 if mutation i is present in the cell and is 0 otherwise. Then a natural comparison of the similarity between two cells is the Jaccard index $J(A, B) = |I(A, B)|/|U(A, B)|$, where $I(A, B)$ is the intersection of A and B and $U(A, B)$ is the union. This function estimates the Jaccard index as a function of Euclidean distance between the cells by randomly sampling N pairs of cells.

Value

A list with the following components

- mean_mutant - A data frame with 2 columns giving the mean number of mutants as a function of Euclidean distance from the lattice origin (Euclid. distance rounded to nearest integer).
- mean_driver - The same as mean_mutant except for driver mutations only. Will be NULL if no drivers are present in the simulated tumor.
- jaccard A data frame with two columns giving mean jaccard index as a function of Euclidean distance between pairs of cells (rounded to nearest integer).

Author(s)

Phillip B. Nicol

Examples

```
set.seed(1126490984)
out <- simulateTumor(max_pop = 1000, driver_prob = 0.1)
sp <- spatialDistribution(tumor = out, make.plot = FALSE)
```

`visualizeTumor` *Interactive visualization of the simulated tumor*

Description

Interactive visualization of the simulated tumor using the `rgl` package (if available).

Usage

```
visualizeTumor(tumor, plot.type = "normal", background = "black", axes = FALSE)
```

Arguments

`tumor`	A list which is the output of `simulateTumor()`.
`plot.type`	Which type of plot to draw. "Normal" assigns a random rgb value to each genotype while "heat" colors cells with more mutations red and cells with fewer mutations blue.
`background`	If rgl is installed, this will set the color of the background
`axes`	Will include axes (rgl only).

Details

If **rgl** is installed, then the plots will be interactive. If **rgl** is unavailable, static plots will be created with **scatterplot3d**. Since plotting performance with **scatterplot3d** is reduced, it is strongly recommended that **rgl** is installed for optimal use of this function.

Value

None.

Author(s)

Phillip B. Nicol

Index