Package ‘CloneData’

July 20, 2022

Title Data to Support CloneSeeker Algorithm

Version 1.0.5

Date 2022-06-30

Author Kevin R. Coombes, Mark Zucker

Description This is a data package to provide simulated example data for the CloneSeeker package, which implements an algorithm to determine clonal architecture from SNP-array copy number data, sequencing mutation data, or both. See Zucker and colleagues (2019) <doi:10.1093/bioinformatics/btz057>.

Maintainer Kevin R. Coombes <krc@silicovore.com>

Depends R (>= 3.0)

Imports CloneSeeker, stats, utils

License Apache License (== 2.0)

LazyData yes

URL http://oompa.r-forge.r-project.org/

NeedsCompilation no

R topics documented:

Generating data from artificial mixtures

Generating sets of artificially mixed and altered heterogeneous data

Description

Generating and saving a ’simulated’ tumor data set by artificially mixing and altering real SNP array data that can be used in clonal heterogeneity analysis to assess accuracy of algorithms.

Usage

```
generateMixtures(dataPath, mixPath, nPerK, segmentedData, ID_pool, pos)
```

Arguments

`dataPath` path to which simulated tumors will be saved.

`mixPath` path to which artificially mixed and altered SNP array data will be saved.

`nPerK` a vector of integers denoting the number of tumors to generate for each possible number of clones, where the nth entry dictates how many n-clone tumors will be generated.

`segmentedData` segmented SNP array data from which mixtures will be generated; must contain following columns: 'loc.start' (segment start locus), 'loc.end' (segment end locus), 'seg.median' (median Log R ratio), 'SamID' (sample ID), 'chrom' (chromosome number), 'AvgBAF' (average B allele fraction for segment), 'num.mark' (number of markers per segment).

`ID_pool` a list of sample IDs from segmentedData from which samples will be drawn to generate artificial mixtures.

`pos` a data frame with two columns, `Chr` and `Position`, defining the chromsoomal locations of the simulated SNPs.

Details

A set of artificial mixtures (with CNVs artificially added) can be generated from real SNP array data. The number of artificial mixtures to generate - and how many mixtures for each possible number of clones to generate - can be set with the input parameters.

Value

The `generateMixtures` function generates and saves two lists for each mixture: a 'tumor' (consisingt of artificially altered real data making up the 'clones' of the mixture, saved in the path 'simpath'), with objects: `psi`, a vector of clonal fractions, `clones`, which is a list of tumor clones, each of which in turn consists of a data frame `cn` and a data frame `seq`, a list `altered` (a list of segments artificially altered), and a list `change` (the copy number change introduced to the altered segments); and a simulated data object (saved in the path 'datapath'), with objects: `cn.data` and `se .data`. Each component is itself a data frame. Note that in some cases, one of these data frames may have zero rows or may be returned as an `NA`.

Each list in the `cn` component contains seven columns:

`chr` the chromosome number;

`start` the starting locus of each genomic segment;

`end` the ending locus of each genomic segment;

`A` the first allelic copy number at each genomic segment;

`B` the second allelic copy number at each genomic segment;

`seg` the segment number; and

`parent.index` the index of the clone from which this clone is descended (equals 0 if the clone is an original tumor clone).

Each list in the `seq` component contains seven columns:

`chr` the chromosome number;

`start` the locus of the simulated SNVs;

`seg` the segment on which each SNV occurs;

`mut.id` the id unique id number for each simulated SNV;

`mutated.copies` the number of copies of the mutated allele at each SNV;

`alllele` which allele (A or B) is mutated at each SNV; and

`normal.copies` the number of copies of the unmutated allele at each SNV.

The `cn.data` component contains seven columns:

`chr` the chromosome number;

`seq` a unique segment identifier;

`LRR` simulated segment-wise log ratios;

`BAF` simulated segment-wise B allele frequencies;

`X` **and** `Y` simulated intensities for two separate alleles/haplotypes per segment; and

`markers` the simulated number of SNPS per segment.

The `seq.data` component contains eight columns:

`chr` the chromosome number;

`seq` a unique "segment" identifier;

`mut.id` a unique mutation identifier;

`refCounts` **and** `varCounts` the simulated numbers of reference and variant counts per mutation;

`VAF` the simulated variant allele frequency;

`totalCounts` the simulated total number of read counts; and

`status` a character (that should probably be a factor) indicating whether a variant should be viewed as somatic or germline.

Author(s)

Kevin R. Coombes <krc@silicovore.com>, Mark Zucker <zucker.64@buckeyemail.osu.edu>

References

Zucker MR, Abruzzo LV, Herling CD, Barron LL, Keating MJ, Abrams ZB, Heerema N, Coombes KR. Inferring Clonal Heterogeneity in Cancer using SNP Arrays and Whole Genome Sequencing. Bioinformatics. To appear. doi: 10.1093/bioinformatics/btz057.

Examples

```
# Set of 300 simulated 'tumors' generated by artificially mixing and
# altering real data; 60 samples with one #clone, 60 with 2 clones,
# ..., 60 with 5 clones.
data("hapmapSegments", package = "CloneData")
data("snpPositions", package = "CloneData")
IDset <- c('NA07019', 'NA12234', 'NA12249', 'NA12753', 'NA12761',
           'NA18545', 'NA18975', 'NA18999', 'NA18517')
# Generating the data set:
## Not run:
generateMixtures(dataPath = 'mixdat', mixPath = 'mixsim',
                 nPerK = rep(60,5),  segmentedData = hapmapSegments,
                 ID_pool = IDset, pos = snpPositions)

## End(Not run)
```

`Generating simulated tumor and data sets`

Generating sets of simulated tumors with SNP array and SNV data

Description

Generating and saving a set of simulated tumors and data that can be used in clonal heterogeneity analysis to assess accuracy of algorithms.

Usage

```
generateSimulationSet(simPath, dataPath, nPerK, rounds=400, nu=0,
                      pcnv=1, norm.contam=FALSE, dataPars=NULL)
```

Arguments

`simPath`	path to which simulated tumors will be saved.
`dataPath`	path to which simulated SNP array and/or SNV data will be saved.
`nPerK`	a vector of integers denoting the number of tumors to generate for each possible number of clones, where the nth entry dictates how many n-clone tumors will be generated.
`rounds`	integer; the number of branches or total 'historical' clones generated in the tumor simulation.
`nu`	an integer; the average number of mutations occuring per clonal branching event.
`pcnv`	a real number between 0 to 1; the probability of a CNV occurring at each clonal branching event.
`norm.contam`	a logical value; determines whether to include normal contamination in simulated tumor.
`dataPars`	a list of parameters for data generation; see Details.

Details

A set of simulation can be generated including both the simulated clonally heterogeneous tumors and the data generated therefrom. The size and general characteristics of the tumor set, as well as the types of data to be created from it (SNP array data and/or SNV data), are determined by the input parameter s. The script included generates three simulated data sets, each with 300 simulations, one with only copy number alterations (and only SNP array data), one with only single nucleotide variants (SNVs) and SNV data, and one with both.

Value

The `generateSimulationSet` function generates and saves two lists for each simulation:

1. a simulated tumor (saved in the path `simpath`), with objects: `psi`, a vector of clonal fractions, and `clones`, which is a list of tumor clones, each of which in turn consists of a data frame `cn` and a data frame `seq`; and
2. a simulated data object (saved in the path `datapath`), with objects: `cn.data` and `se .data`. Each component is itself a data frame. Note that in some cases, one of these data frames may have zero rows or may be returned as an `NA`.

Each list in the cn component contains seven columns:

chr the chromosome number;

start the starting locus of each genomic segment;

end the ending locus of each genomic segment;

A the first allelic copy number at each genomic segment;

B the second allelic copy number at each genomic segment;

seg the segment number; and

parent.index the index of the clone from which this clone is descended (equals 0 if the clone is an original tumor clone).

Each list in the seq component contains seven columns:

chr the chromosome number;

start the locus of the simulated SNVs;

seg the segment on which each SNV occurs;

mut.id the id unique id number for each simulated SNV;

mutated.copies the number of copies of the mutated allele at each SNV;

alllele which allele (A or B) is mutated at each SNV; and

normal.copies the number of copies of the unmutated allele at each SNV.

The cn.data component contains seven columns:

chr the chromosome number;

seq a unique segment identifier;

LRR simulated segment-wise log ratios;

BAF simulated segment-wise B allele frequencies;

X **and** Y simulated intensities for two separate alleles/haplotypes per segment; and

markers the simulated number of SNPS per segment.

The seq.data component contains eight columns:

chr the chromosome number;

seq a unique "segment" identifier;

mut.id a unique mutation identifier;

refCounts **and** varCounts the simulated numbers of reference and variant counts per mutation;

VAF the simulated variant allele frequency;

totalCounts the simulated total number of read counts; and

status a character (that should probably be a factor) indicating whether a variant should be viewed as somatic or germline.

Author(s)

Kevin R. Coombes <krc@silicovore.com>, Mark Zucker <zucker.64@buckeyemail.osu.edu>

References

Zucker MR, Abruzzo LV, Herling CD, Barron LL, Keating MJ, Abrams ZB, Heerema N, Coombes KR. Inferring Clonal Heterogeneity in Cancer using SNP Arrays and Whole Genome Sequencing. Bioinformatics. To appear. doi: 10.1093/bioinformatics/btz057.

Examples

```
# Simulation set with just CNVs, 300 simulations in total, 60 with 1
#clone, 60 with 2 clones... 60 with 5 clones.
## Not run:
generateSimulationSet(simPath = 'sims-cnv', dataPath = 'data-cnv',
    nPerK = rep(60,5), rounds = 400, nu = 0, pcnv = 1, norm.contam = FALSE)

## End(Not run)
```

hapmapSegments *Segmented HapMap Copy Number Data*

Description

Data from 225 HapMap control samples that has been analyzed with DNAcopy to identify segments and record information about copy number variaiton.

Usage

```
data(hapmapSegments)
```

Format

A data frame (`hapmapSegments`), with seven columns and 61,163 rows. This object contains the results of performing a segmentation copy number analysis (using `DNAcopy`). The seven columns are:

`loc.start` The starting base position of the segment.

`loc.end` The ending base postion of the segment.

`seg.median` The median log R ratio across the segment.

`SamID` The HapMap sample ID.

`chrom` The chromosome on which the segment is located, stored as a single character in {1, 2, .., 22, X, Y}.

`AvgBAF` The average B allele frequency across the segment.

`num.mark` The number of markers (i.e., measured SNPs) located in the segment.

Source

BeadChip readings derived from 225 HapMap controls assessed on Human610-Quadv1 BeadChips were downloaded from the Gene Expression Omnibus (http://www.ncbi.nlm.nih.gov/geo; accession number GSE17205, 73 CEU samples; accession number GSE17206, 75 CH + JP; accession number GSE17207, 77 YRI). Raw BeadChip data from 168 patients with CLL and 225 HapMap controls were preprocessed to decode SNP/probe positions and generate genotype calls, log R ratio, and B-allele frequency (BAF) estimates using Illumina GenomeStudio, version 2010.2 (Illumina Inc.). Further processing to produce the segmentation results is described in the paper by Schweighofer et al. [1]

References

[1] Schweighofer CD, Coombes KR, Majewski T, Barron LL, Lerner S, Sargent RL, O'Brien S, Ferrajoli A, Wierda WG, Czerniak BA, Medeiros LJ, Keating MJ, Abruzzo LV. Genomic variation by whole-genome SNP mapping arrays predicts time-to-event outcome in patients with chronic lymphocytic leukemia: a comparison of CLL and HapMap genotypes. J Mol Diagn. 2013 Mar;15(2):196-209.

[2] International HapMap Consortium. The International HapMap Project. Nature. 2003 Dec 18;426(6968):789-96.

`snpPositions` *SNP Genomic Coordinates*

Description

Chromosome base positions of human SNPS, initially derived from the SNPS included on Illumina microarrays.

Usage

```
data(snpPositions)
```

Format

A data frame (`snpPositions`), with two columns and 600,470 rows. This object contains the postions of SNPs on the Illumina 600K chip. It can be used to simulate SNP-chip data.

Source

Produced by Mark Zucker from Illumina data.

References

Zucker MR, Abruzzo LV, Herling CD, Barron LL, Keating MJ, Abrams ZB, Coombes KR. Inferring Clonal Heterogeneity in Cancer using SNP Arrays and Whole Genome Sequencing. Submitted.

Index