Package 'expands'

September 28, 2014

Type Package

Title ExPANdS

Version 1.5

Date 2014-09-27

Author Noemi Andor

Maintainer Noemi Andor <noemi.andor@gmail.com>

Description Expanding Ploidy and Allele Frequency on Nested Subpopulations (ExPANdS) characterizes coexisting subpopulations in a tumor using copy number and allele frequencies derived from exome- or whole genome sequencing input data (http://www.ncbi.nlm.nih.gov/pubmed/24177718). The model amplifies the statistical power to detect coexisting genotypes, by exploiting run-specific tradeoffs between depth of coverage and breadth of coverage. ExPANdS predicts the number of clonal expansions, the size of the resulting subpopulations in the tumor bulk, the mutations specific to each subpopulation and tumor purity. The main function runExPANdS provides the complete functionality needed to predict coexisting subpopulations from single nucleotide variations (SNVs) and associated copy numbers. The robustness of the subpopulation predictions by ExPANdS increases with the number of mutations provided. It is recommended that at least 200 mutations are used as an input to obtain stable results. Updates include: (1) Additional optional parameter ``min_CellFreq" provided for function runExPANdS, specifying the minimum cellular prevalence of mutations to be included for subpopulation predictions.(2) Filtered loci with high-level amplifications, according to max_PM setting. This reduces unnecessary processing time, as assignment of mutations within amplified regions to subpopulations is not successful. (3) Additional function ``buildMultiSamplePhylo" available, which integrates the subpopulations predicted in multiple, geographically distinct tumor samples into one common phylogeny.

License GPL-2

Depends R (>= 2.10)

Imports rJava (>= 0.5-0), flexmix (>= 2.3), matlab (>= 0.8.9), mclust (>= 4.2), moments (>= 0.13), ape (>= 3.0), permute (>= 0.8)

SystemRequirements Java (>= 1.5)

NeedsCompilation no

Repository CRAN

Date/Publication 2014-09-28 20:43:46

R topics documented:

`assignMutations` *Mutation Assignment*

Description

Assigns mutations to previously predicted subpopulations.

Usage

```
assignMutations(dm, finalSPs, densities, max_PM=6, min_CellFreq=0.1)
```

Arguments

`dm`	Matrix in which each row corrsponds to a mutation. Has to contain at least the following columnnames: **chr** - the chromosome on which each mutation is located; **startpos** - the genomic position of each mutation; **AF_Tumor** - the allele-frequency of each mutation; **PN_B** - the ploidy of the B-allele in normal (non-tumor) cells (binary variable: 1 if the mutation is a germline variant, 0 if somatic).
`finalSPs`	Matrix in which each row corresponds to a subpopulation, as computed by `clusterCellFrequencies`.

densities	The probability density distribution of cellular frequencies computed for each mutation via computeCellFrequencyDistributions. Has to contain the same number of rows as dm.
max_PM	Upper threshold for the number of amplicons per mutated cell (default: 6). See also cellfrequency_pdf.
min_CellFreq	Lower threshold for the prevalence of a mutated cell (default: 0.1); mutations for which allele frequency x copy number are below this value, are excluded from further computation. Decreasing the value of this variable is not recommended unless extensive depth and breadth of coverage underly the measurements of copy numbers and allele frequencies.

Details

Each mutated locus l is assigned to the subpopulation C, whose size is closest to the maximum likelyhood cellular frequency of l:
$C := argmin_C|argmax_f P_l(f) - f^C|$, where $P_l(f)$ is the probability distribution of cellular frequencies as computed by cellfrequency_pdf and f^C is the size of subpopulation C. The mutated loci assigned to each subpopulation cluster represent the genetic profile of each predicted subpopulation.

Value

A list with two fields:

dm	The input matrix with two additional columns: **SP** - the subpopulation to which the mutation has been asssigned; **%maxP** - confidence of the assignment.
finalSPs	The input matrix of subpopulations with the column **nMutations** updated according to the total number of mutations assigned to each subpopulation.

Author(s)

Noemi Andor

See Also

clusterCellFrequencies

---

assignQuantityToMutation

Quantity assignment (copy number) to mutations

Description

Assigns a quantity to each mutated locus. Currently, the only assignable quantity is the average ploidy (among all cells) of the locus in which the mutation is embedded.

Usage

```
assignQuantityToMutation(dm, cbs, quantityColumnLabel="CN_Estimate")
```

Arguments

dm	Matrix in which each row corrsponds to a mutation. Has to contain at least the following column names: **chr** - the chromosome on which each mutation is located; **startpos** - the genomic position of each mutation.
cbs	Matrix in which each row corrsponds to a copy number fragment as computed by a circular binary segmentation algorithm. Has to contain at least the following columnnames: **chr** - chromosome; **startpos** - the first genomic position of a copy number segment; **endpos** - the last genomic position of a copy number segment; **CN_Estimate** - the copy number estimated for each segment.
quantityColumnLabel	The name of the new column. Valid options are: `FPKM, CN_Estimate.`

Value

dm	The input matrix with three additional columns: **quantityID** - the ID of the assigned quantity; **quantityColumnLabel** - the quantity; **segmentLength** - the length of the segment from which the quantity has been assigned.

Author(s)

Noemi Andor

Examples

```
data(cbs)
data(snv)
dm=assignQuantityToMutation(snv,cbs,quantityColumnLabel="CN_Estimate")
```

---

assignQuantityToSP *Quantity assignment (ploidy) to subpopulations*

Description

Assigns quantities to predicted SPs. Currently, the only assignable quantity are SP specific ploidies for the input genome segments (obtained by CBS).

Usage

```
assignQuantityToSP(cbs, dm, colName = "PM", keepAmbigSeg=FALSE)
```

Arguments

dm	Matrix in which each row corrsponds to a mutation. Has to contain at least the following columnnames: **chr** - the chromosome on which each mutation is located; **startpos** - the genomic position of each mutation; **SP** - the SP to which the mutation has been assigned; **PM** - the ploidy of the SP for the segment in which the mutation is embedded.
cbs	Matrix in which each row corrsponds to a copy number fragment as computed by a circular binary segmentation algorithm. Has to contain at least the following columnnames: **chr** - chromosome; **startpos** - the first genomic position of a copy number segment; **endpos** - the last genomic position of a copy number segment; **CN_Estimate** - the copy number estimated for each segment.
colName	The SP specific value assigned to each copy number fragment. Possible values: PM, PM_B. Default: PM.
keepAmbigSeg	Whether to assign ploidy to a subpopulation, SP_i, for a segment containing multiple SP_i specific SNVs, at least two of which have distinct ploidies. If set to TRUE, the median SNV ploidy is assigned as segment ploidy. Setting this parameter to TRUE is not recommended as the output will include segment-assignements where subpopulation specific ploidy is ambiguous.Recommend repeating circular binary segmentation with less stringent parameters instead, to reduce segment length and thus the prevalence of ambiguous assignements. Default: FALSE.

Value

cbs	The input matrix with one additional column for each predicted SP: **SP_size** - the ploidy of each segment in the corresponding SP; Value <NA> indicates that no ploidy could be inferred for the segment in the corresponding SP (either because the SP had zero mutations within the segment, or because the SP had multiple, ambiguous mutation-ploidies within the segment).

Author(s)

Noemi Andor

---

buildMultiSamplePhylo *Relations between inter- and intra-sample subpopulations*

Description

Predicts phylogenetic relations between subpopulations from subpopulation specific copy number and point mutation profiles, while including information about sample origin of each subpopulation. This function differs from buildPhylo in that it integrates the subpoulations predicted in multiple, geographically distinct tumor-samples into one common phylogeny and in that it includes point mutations in addition to copy number variations for phylogeny reconstruction.

Usage

```
buildMultiSamplePhylo(samGr,out, treeAlgorithm = "bionjs", keepAmbigSeg = FALSE, plotF=1)
```

Arguments

samGr	List with three fields: **cbs** - Matrix in which each row corresponds to a copy number segment. CBS is typically the output of a circular binary segmentation algorithm. Columns in CBS must be labeled and must include chr, startpos, endpos and CN_Estimate. **sps** - Matrix in which each row corresponds to a somatic mutations. Columns must include: chr - the chromosome on which each mutation is located; startpos - the genomic position of each mutation; SP - the subpopulation to which the mutation has been asssigned; PM - the total ploidy of all alleles at the mutated genomic locus, in the assigned subpopulation; PM_B - the ploidy of the B-allele at the mutated genomic locus, in the assigned subpopulation. **label** - Label denoting sample origin of each subpopulation. Entry mandatory for each geographical sample.
out	Prefix of file to which multi-sample phylogeny will be saved.
treeAlgorithm	Neighbor joining algorithm used for phylogeny reconstruction (from library ape). Options: bionjs (default), njs.
keepAmbigSeg	Input parameter for called function: assignQuantityToSP.
plotF	Option for displaying a visual representation of the phylogenetic tree (0 - no display; 1 - display). Default: 1.

Details

Reconstructs phylogenetic relationships between subpopulations using the BIONJ algorithm of Gascuel. Pairwise distances between subpopulations are calculated as:
$C := (cnv_{i=j} + snv_{i=j})/(cnv_{ij} + snv_{ij})$, where $cnv_{i=j}$ is the number of copy number segments for which subpopulations i and j have the same copy number; $snv_{i=j}$ is the number of point mutations for which subpopulations i and j have the same mutation status and cnv_{ij}, snv_{ij} are the total number of copy number segments and mutations respectively, for which both subpopulations have available information. Subpopulations with insufficient ploidy and point mutations information are excluded from phylogeny.

Value

An object of class "phylo" (library ape).

Author(s)

Noemi Andor

See Also

buildPhylo

buildPhylo *Relations between subpopulations*

Description

Predicts phylogenetic relations between subpopulations from subpopulation specific copy number profiles.

Usage

```
buildPhylo(ploidy, outF, treeAlgorithm="bionjs")
```

Arguments

ploidy	Ploidy-matrix in which each row corrsponds to a copy number segment. Has to contain at least one column for each predicted subpopulation, with columnname labeled SP_xx, where xx is the size of the corresponding SP. Ploidy-matrix can be obtained by calling assignQuantityToSP.
outF	Prefix of file to which phylogeny will be saved.
treeAlgorithm	Neighbor joining algorithm used for phylogeny reconstruction (from library ape). Options: bionjs (default), njs.

Details

Reconstructs phylogenetic relationships between subpopulations using the BIONJ algorithm of Gascuel. Pairwise distances between subpopulations are calculated as the number of copy number segments for which both subpopulations have the same copy number, divided by the total number of copy number segments for which both subpopulations have available copy number information. Subpopulations with insufficient ploidy information are excluded from phylogeny.

Value

An object of class "phylo" (library ape).

Author(s)

Noemi Andor

See Also

assignQuantityToSP

cbs *Matrix of copy number fragments*

Description

Copy number segments as obtained by circular binary segmentation. Data is derived from a Glioblastoma tumor (TCGA-06-0152-01).

Usage

```
data(cbs)
```

Format

Numeric matrix with 120 rows (one per copy-number segment) and 4 columns:
chr - the chromosome
startpos - genomic position at which copy-number segment starts.
endpos - genomic position at which copy-number segment ends.
CN_Estimate - average copy-number of the segment among all cells.

Source

Data derived from The Cancer Genome Atlas (TCGA).

cellfrequency_pdf *Computes the probability distribution of cellular frequencies for a single mutation.*

Description

Calculates P - the probability density distribution of cellular frequencies for one single mutation. For each f, the value of $P(f)$ reflects the probability that the mutation is present in a fraction f of cells.

Usage

```
cellfrequency_pdf(af, cnv, pnb, freq, max_PM=6)
```

Arguments

af	The allelic frequency at which the muation has been observed.
cnv	The ploidy of the locus in which the mutation is embedded.
pnb	The ploidy of the B-allele in normal cells (binary variable: 1 if the mutation is a germline variant, 0 if somatic).
freq	Array of cellular frequencies at which the probabilities will be calculated.

max_PM — Upper threshold for the number of amplicons per mutated cell (default: 6). max_PM is the maximum number of amplicons above which solutions are rejected in the cell-frequency estimation step described below, i.e. $PM <= max\_PM$. The choice of max_PMshould depend on genomic depth of coverage and on the fraction of the genome sequenced: the higher the quality and abundance of data, the higher max_PM.

Details

We consider two types of molecular mechanisms that convert a locus into its mutated state: copy number variation (CNV) inducing events and single nucleotide variation (SNV) inducing events. We assume that a normal state is defined by a total ploidy of two and B allele ploidy below two, whereas a mutated state has an increased fraction of B alleles. The conditions defining these states for each locus l are as follows: i) $PM^B, PN^B, PM, PN \in N$; ii) $PM^B \geq 1; PN^B \leq 1; PN = 2$; iii) $\frac{PM^B}{PM} \geq \frac{PN^B}{PN}$.

PM^B and PN^B denote the ploidy of the B allele in each cell type: mutated cells and normal cells, respectively. The value of PN^B is one if l has a germline variant, zero otherwise. PM, PN are the total ploidy of mutated cells and normal cells. PM is required to be between one and max_PM, that is, we exclude solutions for which the maximum number of amplicons per cell exceeds the user defined constant max_PM.
The function returns the probability distribution, $P_l(f)$, that the mutation at locus l is present in a fraction f of cells, where $f \in [min_C ellFreq, 1.1]$. At default settings the interval starts at 0.1 because cellular frequencies below 0.1 are typically detected at very low allele-frequencies (<0.05), which in turn are often artifacts at moderate sequencing coverage.The interval ends at 1.1 because a local maxima at $f \approx 1.0$ implies monotonically decreasing funtion-values before and after 1.0. So the extended threshold of 1.1 hat the purpose of allowing for this monotonically decreasing phase to the right of a peak close to 1.0.

Value

List with three components:

p — The probability that the mutation is present in a fraction f of cells, for each input frequency f.

bestF — The cellular frequency that best explains the observed allele frequency and ploidities.

errors — Errors encountered during the density estimation step.

Author(s)

Noemi Andor

References

Noemi Andor, Julie Harness, Sabine Mueller, Hans Werner Mewes and Claudia Petritsch. (2013) ExPANdS: Expanding Ploidy and Allele Frequency on Nested Subpopulations. Bioinformatics.

Examples

```
freq=seq(0.1,1.1,by=0.01);
cfd=cellfrequency_pdf(af=0.26,cnv=2.13,pnb=0,freq=freq, max_PM=6)
plot(freq,cfd$p,type="l",xlab="f",ylab="P(f)");
```

`clusterCellFrequencies` *Clustering of cellular frequency probability distributions*

Description

Calculates overrepresented cell frequencies using a two-step clustering procedure. Based on the assumption that passenger mutations occur within a cell prior to the driver event that initiates the expansion, each clonal expansion should be marked by multiple mutations. Thus mutations and copy number variations that took place in a cell prior to a clonal expansion should be present in a similar fraction of cells and leave a similar trace in the subsequent clonal expansion.

Usage

```
clusterCellFrequencies(densities, precision, nrep=30, min_CellFreq=0.1)
```

Arguments

`densities`	Matrix as obtained by `computeCellFrequencyDistributions`. Each row corresponds to a mutation and each column corresponds to a cellular frequency. Each value $densities[i, j]$ represents the probability that mutation i is present in a fraction f of cells, where f is given by: $colnames(densities[, j])$.
`precision`	Precision with which subpopulation size is predicted, a small value reflects a high resolution and can trigger a higher number of predicted subpopulations (recommended: 0.1/log(n/7), where n = # mutations).
`nrep`	Positive integer indicating the number of algorithm repetitions (default: 30).
`min_CellFreq`	Lower threshold for the prevalence of a mutated cell (default: 0.1); mutations for which allele frequency x copy number are below this value, are excluded from further computation. Decreasing the value of this variable is not recommended unless extensive depth and breadth of coverage underly the measurements of copy numbers and allele frequencies.

Details

In the first step, mutations with similar cellular frequencies are grouped together by hierarchical cluster analysis of the probability distributions using the Kullback-Leibler divergence as a distance measure. The cell frequency at each cluster-maxima denotes the size of the subpopulation that harbors the clustered mutations. In the second step, each cluster is extended by members with similar distributions in an interval around the cluster-maxima.

Value

SPs — Matrix of predicted subpopulations. Each row corresponds to a subpopulation and each column contains information about that subpopulation, such as the size in the sequenced tumor bulk (column **Mean Weighted**) and the confidence with which the subpopulation has been detected (column **score**).

Author(s)

Noemi Andor

References

Noemi Andor, Julie Harness, Sabine Mueller, Hans Werner Mewes and Claudia Petritsch. (2013) ExPANdS: Expanding Ploidy and Allele Frequency on Nested Subpopulations. Bioinformatics.

---

computeCellFrequencyDistributions

Gathering of cell frequency probability distributions

Description

Computes the probability distributions of cell frequencies, by calling `cellfrequency_pdf` for each mutation separately.

Usage

```
computeCellFrequencyDistributions(dm, max_PM=6, precision, min_CellFreq=0.1)
```

Arguments

dm — Matrix in which each row corrsponds to a mutation. Has to contain at least the following columnnames:
chr - the chromosome on which each mutation is located;
startpos - the position of each mutation;
AF_Tumor - the allele-frequency of each mutation;
PN_B - the ploidy of the B-allele in normal cells (binary variable: 1 if the mutation is a germline variant, 0 if somatic).

max_PM — Upper threshold for the number of amplicons per mutated cell (default: 6). See also `cellfrequency_pdf`.

precision — Precision with which subpopulation size is predicted, a small value reflects a high resolution and can trigger a higher number of predicted subpopulations (recommended: `0.1/log(n/7)`, where `n=#` mutations).

min_CellFreq — Lower threshold for the prevalence of a mutated cell (default: 0.1); mutations for which allele frequency x copy number are below this value, are excluded from further computation. Decreasing the value of this variable is not recommended unless extensive depth and breadth of coverage underly the measurements of copy numbers and allele frequencies.

Value

List with three fields:

`freq`	The cellular frequencies for which probabilities are computed.
`densities`	Matrix in which each row corresponds to a mutation and each column corresponds to a cellular frequency. Each value $densities[i, j]$ represents the probability that mutation i is present in a fraction f of cells, where f is $freq[j]$.
`dm`	The input matrix with column f updated according to the cellular frequency that best explains the observed allele frequency and ploidities (PM, PN, PN_B).

Author(s)

Noemi Andor

`plotSPs` *Subpopulation Visualization*

Description

Plots coexistent subpopulations determined by ExPANdS.

Usage

```
plotSPs(dm, sampleID=NA, cex=0.5)
```

Arguments

`dm`	Matrix in which each row corrsponds to a mutation (for example, the matrix output by `assignMutations`). Has to contain at least the following columnnames: **chr** - the chromosome on which each mutation is located; **startpos** - the genomic position of each mutation; **AF_Tumor** - the allele-frequency of each mutation; **PN_B** - the ploidy of the B-allele in normal cells (binary variable: 1 if the mutation is a germline variant, 0 if somatic); **SP** - the subpopulation to which each mutation has been assigned (as fraction of cells in the tumor bulk); **%maxP** - the confidence with which the mutation has been assigned to the correspoding subpopulation.
`sampleID`	The name of the sample in which the mutations have been detected.
`cex`	The amount by which plotting text and symbols should be magnified relative to the default. See also `help(par)`.

Value

For each mutation (x-axis) the function displays:
- the size of the subpopulation to which the mutation has been assigned (squares). Each square is colored based on the confidence with which the mutation has been assigned to the corresponding subpopulation (black - highest, white - lowest).
- the total ploidy of all alleles at the mutated genomic locus in that subpopulation (dots).
- the allele frequency of the mutation. Allele frequencies and ploidities are colored based on the chromosome on which the mutation is located (stars - somatic mutations, triangles - loss of heterozygosity).

Author(s)

Noemi Andor

`roi` *Regions of interest*

Description

For internal use only. Default regional boundary for mutations included during clustering, comprising ca. 468 MB centered on the human exome. Relevant if number of input mutations exceeds user defined threshold (often applies to whole genome sequencing data). A saved image of this object is in sysdata.rda.

Format

Numeric matrix in which each row corresponds to a genomic segment.
Columns:
chr - the chromosome of the segment ;
start - the first genomic position of the segment;
end - the last genomic position of the segment.

Source

Data derived from human SureSelectExome_hg19 50MB library kit annotation

See Also

`runExPANdS`

runExPANdS *Main Function*

Description

Given a set of mutations, ExPANdS predicts the number of clonal expansions in a tumor, the size of the resulting subpopulations in the tumor bulk and which mutations accumulate in a cell prior to its clonal expansion. Input-parameters SNV and CBS hold the paths to tabdelimited files containing the mutations and the copy numbers respectively. Alternatively SNV and CBS can be read into the workspace and passed to runExPANdS as numeric matrices. The robustness of the subpopulation predictions by ExPANdS increases with the number of mutations provided. It is recommended that SNV contains at least 200 mutations to obtain stable results.

Usage

```
runExPANdS(SNV, CBS, maxScore=2.5, max_PM=6, min_CellFreq=0.1, precision=NA,
 plotF=2,snvF="out.expands",maxN=8000,region=NA)
```

Arguments

SNV	Matrix in which each row corresponds to a mutation. Only mutations located on autosomes should be included. Columns in SNV must be labeled and must include: **chr** - the chromosome on which each mutation is located; **startpos** - the genomic position of each mutation; **AF_Tumor** - the allele-frequency of each mutation; **PN_B** - ploidy of B-allele in normal cells. A value of 0 indicates that the mutation has only been detected in the tumor sample (i.e. somatic mutation). A value of 1 indicates that the mutation is also present in the normal (control) sample, albeit at reduced allele frequency (i.e. mutation is consequence of LOH). Mutations, for which the allele frequency in the tumor sample is lower than the corresponding allele frequency in the normal sample, should not be included.
CBS	Matrix in which each row corresponds to a copy number segment. CBS is typically the output of a circular binary segmentation algorithm. Columns in CBS must be labeled and must include: **chr** - chromosome; **startpos** - the first genomic position of a copy number segment; **endpos** - the last genomic position of a copy number segment; **CN_Estimate** - the absolute copy number estimated for each segment.
maxScore	Upper threshold for the confidence of subpopulation detection. Only subpopulations identified at a score below $maxScore$ (default 2.5) are kept.
max_PM	Upper threshold for the number of amplicons per mutated cell (default: 6). Increasing the value of this variable is not recommended unless extensive depth and breadth of coverage underly the measurements of copy numbers and allele frequencies. See also cellfrequency_pdf.

`min_CellFreq` Lower threshold for the prevalence of a mutated cell (default: 0.1); mutations for which allele frequency x copy number are below this value, are excluded from further computation. Decreasing the value of this variable is not recommended unless extensive depth and breadth of coverage underly the measurements of copy numbers and allele frequencies.

`precision` Precision with which subpopulation size is predicted, a small value reflects a high resolution and can trigger a higher number of predicted subpopulations (default 0.1/log(n/7), where n = # mutations).

`plotF` Option for displaying a visual representation of the identified SPs (0 - no display; 1 - display subpopulation size; 2 - display subpopulation size and phylogeny; default: 2).

`snvF` The name of the file from which mutations have been read.

`maxN` Upper limit for # SNVs during clustering. If number of user supplied SNVs exceeds <maxN>, the clustering of cellular frequency distributions will be restricted to SNVs found within <region> (default: 8000; increasing value of this parameter not recommended as complexity at least O(n^2)).

`region` Regional boundary for mutations included during clustering. Matrix in which each row corresponds to a genomic segment. Columns must include:
chr - the chromosome of the segment ;
start - the first genomic position of the segment;
end - the last genomic position of the segment.
Default: SureSelectExome_hg19, comprising ca. 468 MB centered on the human exome. User supplied regions should also be coding regions, as the seletive pressure is higher and biological noise affecting cellular frequencies is reduced compared to non-coding regions.

Value

List with fields:

`finalSPs` Matrix of predicted subpopulations. Each row corresponds to a subpopulation and each column contains information about that subpopulation, such as the size in the sequenced tumor bulk (column **Mean Weighted**) and the confidence with which the subpopulation has been detected (column **score**).

`dm` Matrix containing the input mutations with at least four additional columns:
SP - the subpopulation to which the mutation has been asssigned;
%maxP - the confidence of assignment. **f** - Deprecated. The maximum likelyhood cellular prevalence of this mutation, before it has been assigned to a SP. This value is based on the copy number and allele frequency of the mutation exclusively and is independent of other mutations. Column SP is less sensitive to noise and considered the more accurate estimation of cellular mutation prevalence. **PM** - the total ploidy of all alleles at the mutated genomic locus, in the assigned subpopulation. **PM_B** - the ploidy of the B-allele at the mutated genomic locus, in the assigned subpopulation.

`densities` Matrix as obtained by `computeCellFrequencyDistributions`. Each row corresponds to a mutation and each column corresponds to a cellular frequency. Each value $densities[i, j]$ represents the probability that mutation i is present in a fraction f of cells, where f is given by: $colnames(densities[, j])$.

ploidy	Matrix as obtained by assignQuantityToSP. Each row corresponds to a copy number segment as obtained by CBS. Includes one additional column for each predicted SP, holding the ploidy of each segment in the corresponding SP.
tree	An object of class "phylo" (library ape) as obtained by buildPhylo. Holds the inferred phylogenetic relationships between subpopulations.

Author(s)

Noemi Andor

References

Noemi Andor, Julie Harness, Sabine Mueller, Hans Werner Mewes and Claudia Petritsch. (2013) ExPANdS: Expanding Ploidy and Allele Frequency on Nested Subpopulations. Bioinformatics.

Examples

```
data(snv);
data(cbs);
maxScore=2.5;
set.seed(4); idx=sample(1:nrow(snv), 60, replace=FALSE);
#out= runExPANdS(snv[idx,], cbs, maxScore);
```

---

snv *Single Nucleotide Variations*

Description

Somatic mutations and Loss of Heterozygosity (LOH) of a Glioblastoma tumor (TCGA-06-0152-01)

Usage

```
data(snv)
```

Format

Numeric matrix with 773 rows (one per mutation) and 7 columns:
chr - the chromosome
startpos - genomic position
endpos - same as above
REF - ASCII code of the reference nucleotide (in hg18/hg19)
ALT - ASCII code of the B-allele nucleotide
AF_Tumor - allele frequeny of B-allele
PN_B - ploidy of B-allele in normal cells. A value of 0 indicates that the mutation has only been detected in the tumor sample (i.e. somatic mutations). A value of 1 indicates that the mutation is also present in the normal (control) sample, albeit at reduced allele frequency (i.e. mutation is consequence of LOH). Other mutations should not be included.

Source

Data derived from The Cancer Genome Atlas (TCGA)

Index