Package 'expands'

November 2, 2015

Type Package

Title Expanding Ploidy and Allele-Frequency on Nested Subpopulations

Version 1.6

Date 2015-10-28

Author Noemi Andor

Maintainer Noemi Andor <expands.r@gmail.com>

Description ExPANdS characterizes coexisting subpopulations in a single tumor sample 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 detects coexisting genotypes by leveraging 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 input to obtain stable results. Updates in EXPANDS version 1.6 include: (1) So far mutations had been assigned to maximal one subpopulation. However mutations may not be exclusive to the assigned subpopulation but may also be present in smaller, descending subpopulations. This version of expands decides whether or not this is the case leveraging the predicted phylogenetic structure of the subpopulation composition. (2) Included homozygous deletion as potential scenario when modeling (SNV,CNV) pairs with overlapping genomic location, that are propagated during distinct clonal expansions. (3) Optimized solution to improve sensitivity at cell-frequency distribution margins. Need for improvement was because subpopulation detection sensitivity correlates to centrality of subpopulation size during clustering. Tolerance of copy number and allele frequency measurement errors must be higher for marginal cell-frequencies than for central cell-frequencies, in order to counteract the reduced cluster detection sensitivity at the cell-frequency distribution margins. This is only relevant during subpopulation detection (SNV clustering), cell-frequency independent error tolerance still applies during SNV assignment. (4) Fixed a bug where incorrect data matrix conversion could occur when handing non-numerical matrix as parameter to function runExPANdS. Further documentation and FAQ around ExPANdS is available at http://dna-discovery.stanford.edu/software/expands.

License GPL-2

URL http://dna-discovery.stanford.edu/software/expands

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)

Suggests phylobase (>= 0.6.8)

SystemRequirements Java (>= 1.5)

NeedsCompilation no

Repository CRAN

Date/Publication 2015-11-02 11:33:43

R topics documented:

`assignMutations` *Mutation Assignment*

Description

Assigns mutations to previously predicted subpopulations.

Usage

```
assignMutations(dm, finalSPs, max_PM=6)
```

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

Details

Each mutated locus l is assigned to the subpopulation C, whose size f_C can best explain the allele frequency (AF) and copy number (CN) observed at l.Three alternative cell frequency probabilities, $P_x(f_C)$, are calculated for the SNV at locus l, with x denoting three alternative evolutionary scenarios (see also cellfrequency_pdf):
1. $x := s$ –> Separate fit of SNV and CNV. CNV does not influence ploidy of the SNV, either because CNV occurs before SNV or because SNV and CNV occur independently from each other (i.e. they are never co-propagated during the same clonal expansion)
2. $x := p$ –> Partial dependency of SNV ploidy on CNV. The SNV is propagated during the expansion of C. Subsequently, the CNV is propagated during a clonal expansion of a cell-member of C.
3. $x := j$ –> Joint fit of SNV and CNV, assuming they co-occur together in the same cell and are propagated during the exact same clonal expansion.

The SNV is then assigned to subpopulation $C := argmax_C(P_s(f_C), P_p(f_C), P_j(f_C))$.

The mutated loci assigned to each subpopulation cluster represent the genetic profile of each predicted subpopulation.
The assignment between subpopulation C and locus l only implies that the SNV at l has been first propagated during the clonal expansion that gave rise to C. So SNVs present in C may not be exclusive to C but may also be present in subpopulations smaller than C. Whether or not this is the case depends on the phylogenetic structure of the subpopulation composition. See also buildPhylo.

Value

A list with two fields:

dm	The input matrix with seven additional columns: **SP** - the subpopulation to which the point mutation has been asssigned; **PM_B** - the ploidy of the B-allele at the mutated genomic locus, in the assigned subpopulation (SP). **PM** - the total ploidy of all alleles, in the assigned subpopulation (SP). **SP_cnv** - if the point mutation lies within an amplified or deleted region: the subpopulation to which the copy number variation has been asssigned. This

entry has the same value as SP if and only if: i) the SNV and the CNV were propagated during the same clonal expansion or ii) the SNV lies within a copy neutral region.
PM_cnv - the total ploidy of all alleles, in the CNV harboring subpopulation (SP_cnv).
%maxP - confidence of the point mutation assignment to SP.
scenario - the evolutionary scenario under which the subpopulation configurations for this genomic locus have been solved (see also parameter "snv_cnv_flag" in `cellfrequency_pdf`).

`finalSPs` The input matrix of subpopulations with 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 subpopulations. Currently, the only assignable quantity are subpopulation specific ploidies for the input genome segments (obtained for example by circular binary segmentation).

Usage

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

Arguments

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 (average value across all subpopulations in the sample).

dm — Matrix in which each row corrsponds to a mutation. Has to contain at least the following columnnames:
chr - the chromosome on which each point mutation is located;
startpos - the genomic position of each mutation;
SP - the subpopulation to which the point mutation has been assigned;
SP_cnv - the subpopulation with copy number variation (CNV) in the same genomic segment in which SP has a point mutation;

PM - ploidy of the subpopulation with the point mutation (SP);
PM_cnv - ploidy of genomic segment (overlapping the point mutation), in the subpopulation with CNV (SP_cnv).

`colName` The subpopulation specific value assigned to each copy number fragment. Possible values: PM_cnv, PM, PM_B. Default: PM_cnv.

`keepAmbigSeg` Whether or not to assign ploidy to a subpopulation, SP_i, for a segment containing multiple SP_i specific ploidities, at least two of which have distinct magnitudes. If set to TRUE, the median 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 subpopulation:
SP_xx - where xx is the size of the corresponding subpopulation (SP). Column entries contain the ploidy of each segment in SP; Value <NA> indicates that no ploidy could be inferred for the segment in this subpopulation (either because SP had no point mutations/CNVs within the segment, or because SP had multiple, ambiguous ploidy assignments 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 to infer inter-sample phylogenetic relations.

Usage

```
buildMultiSamplePhylo(samGr,out,treeAlgorithm="bionjs",ambigSg=F,plotF=1,spRes=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 (see cbs). **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 matrix. Entry is 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.
ambigSg	Input parameter "keepAmbigSeg" for called function: assignQuantityToSP.
plotF	Option for displaying a visual representation of the phylogenetic tree (0 - no display; 1 - display). Default: 1.
spRes	Option on whether or not to ignore the subpopulations calculated for each sample and instead treat every geographical tumor-sample as one single tumor-metapopulation (Default value: 1 - subpopulation resolution; 0 - metapopulation resolution).

Details

Reconstructs phylogenetic relationships between subpopulations using neighbor-joining algorithms provided by R-package 'ape'. Pairwise distances between subpopulations i and j are calculated as: $d_{ij} := (cnv_{i=j} + snv_{i=j})/(cnv_{ij} + snv_{ij})$, 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", dm=NA)
```

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. Subpopulation columnnames must be labeled SP_xx, where xx is the size of the corresponding subpopulation. 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.
dm	Optional matrix in which each row corresponds to a mutation. Only mutations located on autosomes should be included. Columns in dm must be labeled and must include: **SP** - the subpopulation to which the point mutation has been asssigned. **SP_cnv** - the subpopulation to which the CNV (overlapping with the point mutation) has been asssigned (if an CNV is present). **chr** - the chromosome on which each point mutation is located; **startpos** - the genomic position of each point mutation; **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. If dm is available, an attempt will be made to: i) assign total ploidy (PM) to subpopulations with point mutations and ii) assign every mutation to >1 subpopulation according to the inferred phylogenetic relations between subpopulations. Default: NA.

Details

Reconstructs phylogenetic relationships between subpopulations using neighbor-joining algorithms provided by R-package 'ape'. 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

List with two fields:

`tree`	An object of class "phylo" (library ape).
`dm`	The input matrix with each row representing a point mutation and additional columns: **SP_xx** - where xx is the size of the corresponding subpopulation. Column entries contain a binary indicator of whether or not the point mutation in this row is present in SP_xx.

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 point mutation or CNV. For each cell-frequency 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, snv_cnv_flag=3, SP_cnv = NA, PM_cnv = NA)
```

Arguments

`af`	The allelic frequency at which the point muation has been observed.
`cnv`	The average copy number 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). B-alleles with normal cell ploidy>1 are not modeled.
`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.
`snv_cnv_flag`	Flag indicating the evolutionary scenario under which frequency should be estimated: 1 - cellular frequency of SNV only; 2 - cellular frequency of CNV only (parameters AF and pnb are ignored); 3 - cellular frequency of SNV and CNV simultaneously, under the assumption that they are identical (co-occurrence assumption of SNV and CNV in same subpopulation); 4 - cellular frequency of SNV, under the assumption that it occured in the ancestor of the subpopulation with the CNV (this is the only scenario under which the SNV can have distinct ploidies in distinct subpopulations). Default: 3.
`SP_cnv`	Size of the subpopulation that harbors a copy number variation (CNV) at this locus. This variable is only relevant if an CNV and an SNV have overlapping genomic location, yet have been propagated during distinct clonal expansions (snv_cnv_flag=4).
`PM_cnv`	Total ploidy in subpopulation which harbors a copy number variation (CNV) at this locus. This variable is only relevant if an CNV and an SNV have overlapping genomic location, yet have been propagated during distinct clonal expansions (snv_cnv_flag=4).

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, $P(f)$, that the mutation at locus l is present in a fraction f of cells, where $f \in [0, 1]$. Four alternative cell frequency probability distributions, $P(f)$, can be obtained for each allele-frequency + copy number pair (AF, CN).

1. $P_s(f_{cnv})$ separately modeling the size f_{cnv} of the subpopulation propagating an CNV: $PM * f_{cnv} + PN * (1 - f_{cnv}) = CN$
2. $P_s(f_{snv})$ and $P_p(f_{snv})$ modeling the size f_{snv} of the subpopulation propagating an SNV:
2a) $P_s(f_{snv})$: $PM^B * f_{snv} + PN^B * (1 - f_{snv}) = AF * CN$, where $PM^B \leq max(2, PM)$;
Here f_{snv} is calcualted separately of f_{cnv}, under the assumption that i) SNV and CNV occur independently from each other (i.e. they are never co-propagated during the same clonal expansion) or ii) SNV occured in a denscendant of the subpopulation with the CNV.
2b) $P_p(f_{snv})$: $PM^B * (f_{snv} - f_{cnv}) + pm^B * f_{cnv} + PN^B * (1 - f_{snv}) = AF * CN$, where $pm^B \neq PM^B$ and $pm^B \neq 2$.
Here f_{snv} is calcualted partially dependent on f_{cnv}, under the assumption that the SNV occured in an ancestor of the subpopulation with the CNV.
3. $P_j(f)$ jointly modeling the size f of the subpopulation propagating both the SNV and the CNV simulataneously: enforcing both equations, 1) and 2a), with additional constraints: $f := f_{snv} = f_{cnv}$ and $PM^B \leq PM$

In 1) and 2) the SNV is present in a subpopulation different of the CNV harboring subpopulation. In 3) both the SNV and an CNV at l were propagated during the same clonal expansion.

Value

List with four components:

`p`	The probability that the point mutation/CNV is present in a fraction f of cells, for each input frequency f in parameter $freq$.
`bestF`	The cellular frequency that best explains the observed allele frequency and/or copy number.
`fit`	Matrix with each row containing one alternative solution, (PM, PM_B, f), as well as an assesment of how well the solution fits above equations (Column "dev").

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(0.1,1.0,by=0.01);
cfd=cellfrequency_pdf(af=0.26,cnv=1.95,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 approach. 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 "frequency-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 lead to a higher number of predicted subpopulations.
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).

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 noise score at which the subpopulation has been detected (column **score**: lower values ~ higher subpopulation detection confidence).

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 lead to a higher number of predicted subpopulations.
min_CellFreq	Lower boundary for the prevalence of a mutated cell (default: 0.1).

Value

List with three fields:

freq	The cellular frequencies for which probabilities are computed.
densities	Matrix in which each row corresponds to a point 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 $freq[j]$ of cells.
dm	The input matrix with column f updated according to the cellular frequency that best explains the observed allele frequency and copy number.

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 point mutation (for example, the matrix output of 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; **PM** - the total ploidy of all alleles at the mutated genomic locus, in subpopulation SP.

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 point 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 point 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 point mutations to obtain stable results.

Usage

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

Arguments

SNV	Matrix in which each row corresponds to a point 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 noise score 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 boundary for the cellular prevalence interval of a mutated cell. In default settings the interval starts at 0.1 because cellular frequencies below 0.1 typically correspond to low allele-frequencies (often <0.05), which in turn are often artifacts at moderate sequencing coverage. Mutations for which allele frequency * copy number are below $min_{CellFreq}$, 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 lead to a higher number of predicted subpopulations.

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

`snvF` Prefix of file to which predicted subpopulation composition will be saved. Default: the name of the file from which mutations have been read or "out.expands" if input mutations are not handed over as file path.

`maxN` Upper limit for number of point mutations used during clustering (default: 8000; increasing value of this parameter not recommended). If number of user supplied point mutations exceeds $maxN$, the clustering of cellular frequency distributions will be restricted to point mutations found within $region$.

`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. Alternative user supplied regions should also be coding regions, as the seletive pressure is higher as 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 noise score at which the subpopulation has been detected (column **score**).

`dm` Matrix containing the input mutations with at least five additional columns:
SP - the subpopulation to which the point mutation has been asssigned;
SP_cnv - the subpopulation to which the CNV has been asssigned (if an CNV exists at this locus);
%maxP - the confidence of point mutation assignment.
f - Deprecated. The maximum likelyhood cellular prevalence of this point mutation, before it has been assigned to SP. This value is based on the copy number and allele frequency of the mutation exclusively and is independent of other point 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 subpopulation harboring the point mutation (SP).
PM_B - the ploidy of the B-allele at the mutated genomic locus, in the subpopulation harboring the point mutation (SP).
PM_cnv - the total ploidy of all alleles at the mutated genomic locus, in the subpopulation harboring an CNV (SP_cnv).

`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, e.g. as obtained from a circular binary segmentation algorithm. Includes one additional column for each predicted subpopulation, containing the ploidy of each segment in the corresponding subpopulation.

`tree` An object of class "phylo" (library ape) as obtained by `buildPhylo`. Contains 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