Package 'liayson'

December 21, 2018

Type Package

Title Linking Singe-Cell Transcriptomes Atween Contemporary Subpopulation Genomes

Version 1.0.1

Date 2018-12-10

Author Noemi Andor

Maintainer Noemi Andor <liayson.scRNA.R@gmail.com>

Description

Given an RNA-seq derived cell-by-gene matrix and an DNA-seq derived copy number segmentation, LIAYSON predicts the number of clones present in a tumor, their size, the copy number profile of each clone and the clone membership of each single cell.

License GPL-2

URL https://github.com/noemiandor/liayson, https://groups.google.com/d/forum/liayson

Depends R (>= 3.0)

Imports phangorn, RColorBrewer, ape, parallel, plyr, matlab, biomaRt, distances, arules, e1071, proxy, gplots, methods

Suggests mclust, fpc, NbClust, modeest, pastecs, vegan

NeedsCompilation no

Repository CRAN

Date/Publication 2018-12-21 15:30:03 UTC

RoxygenNote 6.1.0

R topics documented:

`aggregateSegmentExpression` *Aggregating genes across copy number segments.*

Description

Calculates average expression of genes grouped by common segment membership.

Usage

```
aggregateSegmentExpression(epg, segments, mingps = 20, GRCh=37)
```

Arguments

`epg`	Gene-by-cell matrix of expression. Recommendation is to cap extreme UMI counts (e.g. at the 99% quantile) and to include only cells expressing at least 1,000 genes.
`segments`	Matrix in which each row corresponds to a copy number segment as calculated by a circular binary segmentation algorithm. Has to contain at least the following column names: **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.
`mingps`	Minimum number of expressed genes a segment needs to contain in order to be included in output.
`GRCh`	Human reference genome version to be used for annotating gene coordinates.

Details

Let S := { $S_1, S_2, ...S_n$ } be the set of n genomic segments that have been obtained from DNA-sequencing a given sample (e.g. from bulk exome-sequencing, scDNA-sequencing, etc.). Genes are mapped to their genomic coordinates using the biomaRt package and assigned to a segment based on their coordinates. Genes are grouped by their segment membership, to obtain the average number of UMIs and the number of expressed genes per segment S_j per cell i.

Value

List with fields:

eps	Segment-by-cell matrix of expression values.
gps	Segment-by-cell matrix of the number of expressed genes.

Author(s)

Noemi Andor

Examples

```
data(epg)
data(segments)
#X=aggregateSegmentExpression(epg, segments, mingps=20, GRCh=38)
```

---

assignCellsToClusters *Assigns cells to previously defined clones.*

Description

Cells that have not been used to define clones (such as cycling or apoptotic cells) can retrospectively be assigned a clone membership.

Usage

```
assignCellsToClusters(outc, xps, similarity=T)
```

Arguments

outc	List containing segment-by-cell matrix and clone membership of each cell. See clusterCells).
xps	Segment-by-cell matrix of expression- or copy number states. Columns represent new cells to be assigned to existing clones.
similarity	Whether to use similarity (similarity=T) or distance (similarity=F), when comparing cells to existing clones. Default similarity metric is "correlation. Default distance metric is "Euclidean".

Details

Let S := { $S_1, S_2, ...S_n$ } be the set of n genomic segments obtained from bulk DNA-sequencing. Further, let $S_I \in S$ be the subset of segments for which cells within a clone have a well defined copy number state. Pearson Correlation Coefficients are calculated as similarity metric between each new cell and the consensus profile of each clone, based on segments $s \in S_I$. Each cell is assigned to the clone to which it is most similar.
Alternatively, if similarity is set to false, the Euclidean distance metric is used instead of the Pearson Correlation.

Value

List with same components as input:

cnps	Segment-by-cell matrix of copy number states, with new cells added as columns.
sps	The clone membership of each cell (that is columns in cnps).

Author(s)

Noemi Andor

`clusterCells` *Grouping cells into clones.*

Description

Clusters cells according to their copy number profile.

Usage

```
clusterCells(cnps, k=NA, h=NA, weights=NULL, minSegLength=1E6,
                        chrOrder=NULL,  HFUN="ward.D2",...)
```

Arguments

cnps	Segment-by-cell matrix of copy number states (output of `segmentExpression2CopyNumber`).
k	Desired number of clusters (see also `cutree`).
h	Threshold used to define clones from hierarchical clustering result. A subtree is defined as a clone if the maximum distance between its cell members is less than $100*h\%$ of the genome.
weights	Vector of weights assigning differential importance to segments (typically calculated based on segment lengths).
minSegLength	Minimum number of base pairs below which a segment is to be excluded when defining clones.
chrOrder	Specifies order in which chromosomes should be plotted.
HFUN	Agglomeration method used to compute the hierarchical clustering (see also `hclust`).
...	additional arguments passed on to `heatmap.2`

Details

Let CNF be the matrix of copy number states per non-private segment per cell, with entries (i, j) pointing to the copy number state of cell j at locus i. Pairwise distances between cells are calculated in Hamming space of their segmental copy number profiles (rows in CNF), weighted by segment length. Hierarchical clustering is used to build a tree of the cells from the distance matrix. A subtree is defined as a clone if the maximum distance between its cell members is less than a user-defined fraction of the genome (h).
Alternatively, if k is set, the tree is cut to obtain k clones.
If neither h nor k are set, Akaike information criterion is used to decide on anywhere between 1 and 30 clones.

Value

List with three fields:

`cnps`	Segment-by-cell matrix of copy number states.
`sps`	The clone membership of each cell (that is, columns in cnps).
`tree`	An object of class hclust.

Author(s)

Noemi Andor

References

Andor, N.*, Lau, B.*, Catalanotti, C., Kumar, V., Sathe, A., Belhocine, K., Wheeler, T., et al. (2018) Joint single cell DNA-Seq and RNA-Seq of gastric cancer reveals subclonal signatures of genomic instability and gene expression. doi: https://doi.org/10.1101/445932

Examples

```
data(cnps)
set.seed(2)
rcells = sample(colnames(cnps), 120)
outc = clusterCells(cnps[apply(cnps, 1, var)>0, rcells])
```

---

cnps *Segment-by-cell matrix of copy number states from NCI-N87 cell line.*

Description

Matrix of segments (rows) x 200 cells (columns) with entries denoting inferred copy numbers.

Usage

```
data(cnps)
```

Source

Data obtained from Ji lab at Stanford.

---

epg *Gene-by-cell matrix of expression from NCI-N87 cell line.*

Description

Matrix of genes (rows) x 200 cells (columns) with entries denoting UMI counts.

Usage

```
data(epg)
```

Source

Data obtained from Ji lab at Stanford.

---

eps *Segment-by-cell matrix of expression from NCI-N87 cell line.*

Description

Matrix of segments (rows) x 200 cells (columns) with entries denoting average expression values.

Usage

```
data(eps)
```

Source

Data obtained from Ji lab at Stanford.

`getNumRes` *Clone size resolution.*

Description

Informs user about resolution at which clone sizes are stored.

Usage

```
getNumRes()
```

Details

For internal and external use.

Author(s)

Noemi Andor

`runLIAYSON` *Main Function.*

Description

Given an RNA-seq derived cell-by-gene matrix and an DNA-seq derived copy number segmentation, LIAYSON predicts the number of clones present in a tumor, their size, the copy number profile of each clone and the clone membership of each single cell.

Usage

```
runLIAYSON(X, S, sName, mingps = 20, GRCh = 37, h = 0.2, minSegLength=1E6, outD = NULL)
```

Arguments

`X`	Gene-by-cell matrix of expression. Recommendation is to cap extreme UMI counts (e.g. at the 99% quantile) and to include only cells expressing at least 1,000 genes.
`S`	Matrix in which each row corresponds to a copy number segment as calculated by a circular binary segmentation algorithm. Has to contain at least the following column names: **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.
`sName`	Sample name.

mingps	Minimum number of expressed genes a segment needs to contain in order to be included in output.
GRCh	Human reference genome version to be used for annotating gene coordinates.
h	Height at which the tree should be cut (see also cutree).
minSegLength	Minimum number of base pairs below which a segment is to be excluded when defining clones.
outD	The output directory.

Author(s)

Noemi Andor

See Also

clusterCells segmentExpression2CopyNumber

Examples

```
data(epg)
data(segments)
#out = runLIAYSON(epg, segments, sName="NCI-N87", GRCh = 38, h=0.05)
```

---

saveClusteredCells *Saving clones to user-defined output.*

Description

Writes clone statistics, dendrogram and clone-specific mutation profiles.

Usage

```
saveClusteredCells(outc, outD, sName)
```

Arguments

outc	Output of clusterCells or assignCellsToClusters: list containing segment-by-cell matrix, clone membership of each cell and the underlying dendrogram.
outD	The output directory.
sName	Prefix for the output files (typically the sample name).

Details

Writes each of the following aspects of a sample's clonal composition into an output file:
1. The clone membership of each cell (*.spstats)
2. The segment-by-cell matrix of copy number states (*.sc.cbs)
3. The consensus copy number profile of each detected clone, calculated as the average profile of cells that are members of the respective clone (*.sps.cbs)
4. The cell dendrogram (*.tree).

Author(s)

Noemi Andor

`segmentExpression2CopyNumber` *Calling CNVs.*

Description

Maps single cell expression profiles to copy number profiles.

Usage

```
segmentExpression2CopyNumber(eps, gpc, cn, seed=0, outF=NULL, maxPloidy=8,
                                        nCores=2, stdOUT="log.applyAR2seg")
```

Arguments

`eps`	Segment-by-cell matrix of expression.
`gpc`	Number of genes expressed per cell.
`cn`	Average copy number across cells for each segment (i.e. row in eps).
`seed`	The fraction of entries in a-priori segment-by-cell copy number matrix to be used as seed for association rule mining.
`outF`	Output file prefix in which to print intermediary heatmaps and histograms, or NULL (default) if no print.
`maxPloidy`	The maximum ploidy to accept as solution.
`nCores`	The numbers of threads used.
`stdOUT`	Log-file to which standard output is redirected during parallel processing.

Details

Let S := { $S_1, S_2, ...S_n$ } be the set of n genomic segments obtained from bulk DNA-sequencing. Let E_{ij} and G_{ij} be the average number of UMIs and the number of expressed genes per segment i per cell j. The segment-by-cell expression matrix is first normalized by gene coverage. For each $x \in S$, the linear regression model:

$$E_{x*} \sim \sum_{i \in S} G_{i*}$$

, fits the average segment expression per cell onto the cell's overall gene coverage. The model's residuals R_{ij} reflect inter-cell differences in expression per segment that cannot be explained by differential gene coverage per cell. A first approximation of the segment-by-cell copy number matrix CN is given by:

$$CN_{ij} := R_{ij} * (cn_i / \bar{R_{i*}})$$

, where cn_i is the population-average copy number of segment i derived from DNA-seq. Above transformation of E_{ij} into CN_{ij} is in essence a numerical optimization, shifting the distribution of each segment to the average value expected from bulk DNA-seq.

Let $x' \in CN$ be the measured copy number of a given segment-cell pair, and x its corresponding true copy number state. The probability of assigning copy number x to a cell j at locus i depends on:
A. Cell j's read count at locus i, calculated conditional on the measurement x'. Using a Gaussian smoothing kernel, we compute the kernel density estimate of the read counts at locus i across cells to identify the major (M) and the minor (m) copy number states of i as the highest and second highest peak of the fit respectively. Then we calculate the proportion of cells expected at state m as $f = \frac{cn_i - M}{m - M}$. The probability of assigning copy number x to a cell j at locus i is calculated as:
$P_A(x|x') \sim$
: $0, if x \notin m, M$
: $P_{ij}(x'|N(m, sd = f)), if x == m$
: $P_{ij}(x'|N(M, sd = 1 - f)), if x == M$

B. Cell j's read count at other loci, i.e. how similar the cell is to other cells that have copy number x at locus i. We use Apriori - an algorithm for association rule mining - to find groups of loci that tend to have correlated copy number states across cells. Let $V_{i,K \to x}$ be the set of rules concluding copy number x for locus i, where $k \in K$ are copy number profiles of up to $n = 4$ loci in the form $\{ S_1 = x_1, S_2 = x_2, ...S_n = x_n \}$. Further let C_r be the confidence of a rule $r \in V_{i,K \to x}$. For each cell $j \in J$ matching any of the copy number profiles in K, we calculate:
$P_B(x) \sim \sum_{r \in V_{i,K \to x}} C_r$
, the cumulative confidence of the rules in support of x at i.

We first obtain a seed of cell-segment pairs by assigning a-priori copy number states only when $argmax_{x \in [1,8]} P_A(x|x') > t$. We use this seed as input to **B**. Finally, a-posteriori copy number for segment i in cell j is calculated as:

$$argmax_{x \in [1,8]} P_A(x|x') + P_B(x)$$

Value

Segment-by-cell matrix of copy number states.

Author(s)

Noemi Andor

References

Andor, N.*, Lau, B.*, Catalanotti, C., Kumar, V., Sathe, A., Belhocine, K., Wheeler, T., et al. (2018) Joint single cell DNA-Seq and RNA-Seq of gastric cancer reveals subclonal signatures of genomic instability and gene expression. doi: https://doi.org/10.1101/445932

Borgelt C & Kruse R. (2002) Induction of Association Rules: Apriori Implementation.

See Also

apriori

Examples

```
##Calculate number of genes expressed per each cell:
data(epg)
gpc = apply(epg>0, 2, sum)

##Call function:
data(eps)
data(segments)
cn=segments[rownames(eps),"CN_Estimate"]
#cnps = segmentExpression2CopyNumber(eps, gpc, cn, seed=0.5, nCores=2, stdOUT="log")
#head(eps[,1:3]); ##Expression of first three cells
#head(cnps[,1:3]); ##Copy number of first three cells
```

---

segments *Bulk copy number profile of NCI-N87 cell line.*

Description

Copy number segmentation matrix obtained as average among G0G1 cells.

Usage

```
data(segments)
```

Format

Matrix in which each row corresponds to a copy number segment as calculated by a circular binary segmentation algorithm. Has to contain at least the following column names:
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.

Source

Data obtained from Ji lab at Stanford.

Index