Package 'seqCNA'

October 10, 2021

Type Package

Title Copy number analysis of high-throughput sequencing cancer data

Version 1.38.0

Date 2015-01-09

Author David Mosen-Ansorena

Maintainer David Mosen-Ansorena <dmosen.gn@cicbiogune.es>

Import GLAD, doSNOW, adehabitatLT, seqCNA.annot

Depends R (>= 3.0), GLAD (>= 2.14), doSNOW (>= 1.0.5), adehabitatLT (>= 0.3.4), seqCNA.annot (>= 0.99), methods

Description Copy number analysis of high-throughput sequencing cancer data with fast summarization, extensive filtering and improved normalization

License GPL-3

SystemRequirements samtools

biocViews CopyNumberVariation, Genetics, Sequencing

git_url https://git.bioconductor.org/packages/seqCNA

git_branch RELEASE_3_13

git_last_commit 9827ffc

git_last_commit_date 2021-05-19

Date/Publication 2021-10-10

R topics documented:

`seqCNA-package` *Copy number analysis of deep sequencing cancer data*

Description

Fast summarization, extensive filtering and improved normalization

Details

Package:	seqCNA
Type:	Package
Version:	1.13.0
Date:	2014-01-09
License:	GPL-3

applyFilters applyThresholds plotCNProfile readSeqsumm runGLAD runSeqnorm runSeqsumm writeCNProfile

Author(s)

David Mosen-Ansorena

Maintainer: David Mosen-Ansorena <dmosen.gn@cicbiogune.es>

`applyFilters` *The function applies a set of filters to the raw profile.*

Description

The available filters are: PEM-based (only for paired-end data), trimming of extreme values, mappability-based and common CNVs.

Usage

```
applyFilters(rco, pem.filter=0, trim.filter=0, mapp.filter=0, mapq.filter=0, cnv.filter=FALSE, plots=
```

Arguments

rco	A SeqCNAInfo-class object, with read count (RC) and genomic information, normally the output of the readSeqsumm function.
pem.filter	seqsumm summarizes reads not only by window, but also by read type (i.e. SAM read flags). The ratio between proper and improper reads is calculated for each window, where higher ratios correlate with windows in centromeres, structural polymorphisms,... and in general regions advisable to be filtered. The value indicates the quantile of windows to filter based on the improper read ratio.
trim.filter	A numeric value or a vector with two numeric values. Set to a value between 0 and 1 to filter such top and bottom quantiles (e.g. 0.01). If two numeric values are provided, the first one is used for the upper quantile and the second for the lower quantile. If the normal sample is not provided, set to 1 for automatic upper threshold selection (lower thresohold can be set with a second value or take the same value as the upper threshold). Pre-normalization is performed over the paired-normal read count, or the tumoural one if the paired-normal is not available, against the GC content. This keeps the trimming from being biased by the influence of GC content on the read count.
mapp.filter	Only available if the annotation package has information on the sample genome and build (currently hg18 and hg19). Mappability ranges from 0 to 1. The higher the parameter value, the less windows will be filtered.
mapq.filter	This filter discards genomic windows with low mean mapping quality in the reads of the tumoural sample. The mean mapping quality in a window may range from 0 to a few tens, with an upper limit of 255. Therefore, the higher the parameter value, the more windows will be filtered.
cnv.filter	A boolean indicating whether to apply the CNV-based filter. If applied, windows with a common CNV (Altshuler et al., 2010) spanning at least 95 per cent of the window are filtered. This filter is available for completion purposes, but, given that the frequency of common CNVs can be as low as 0.01, very few polymorphisms are captured in comparison to the amount of expected false positives.
plots	A boolean indicating whether to output plots to the folder folder.
folder	If the plots parameter is set to TRUE, path to the folder where the plots with filtering information are to be generated. If no folder is indicated or does not exist, plots will be displayed within R.
nproc	A value indicating how many processing cores to use for the processes of building genome information and automatic trimming, if applicable. Greater values speed up the processsing using more CPU cores and RAM memory, but you should not use values greater than the number of cores in your machine. If unsure, the safest value is 1, but most computers nowadays are multi-core, so you could probably go up to 2, 4 or 8.

Details

The function outputs two plots to the folder folder. We suggest reviewing these plots in order to adjust the filters to adequate values. For the Venn diagrams, functions by Thomas Girke are used (see http://faculty.ucr.edu/~tgirke/Documents/R_BioCond/R_BioCondManual.html).

Value

A SeqCNAInfo-class object, with additional information on which windows to filter and the profiles needed for subsequent normalization in runSeqnorm.

Author(s)

David Mosen-Ansorena

References

Integrating common and rare genetic variation in diverse human populations. Altshuler DM, Gibbs RA, Brooks LD, McEwen JE. Nature. 2010 Sep 2; 467:52-8

Examples

```
data(seqsumm_HCC1143)
rco = readSeqsumm(tumour.data=seqsumm_HCC1143)

### FILTERING ###

pem.filter = 0
trim.filter = 1
cnv.filter = FALSE
mapp.filter = 0
mapq.filter = 2

rco = applyFilters(rco, pem.filter, trim.filter, mapp.filter, mapq.filter, cnv.filter, plots=FALSE)
```

---

applyThresholds *Apply thresholds on the segmented profile in order to call copy numbers.*

Description

Given a set of thresholds and the copy number of the lower resulting range, copy numbers are called.

Usage

```
applyThresholds(rco, thresholds, min.CN)
```

Arguments

rco	A SeqCNAInfo-class object, with read count (RC) and genomic information, normally the output of the runGLAD function.
thresholds	A vector with values that establish the ranges for each copy number.
min.CN	The copy number of the lowest range defined by the thresholds.

Value

A SeqCNAInfo-class object, with additional information on the copy numbers called for each genomic window.

Author(s)

David Mosen-Ansorena

Examples

```
data(seqsumm_HCC1143)
rco = readSeqsumm(tumour.data=seqsumm_HCC1143)
rco = applyFilters(rco, 0, 1, 0, 2, FALSE, plots=FALSE)
rco = runSeqnorm(rco, plots=FALSE)
rco = runGLAD(rco)

### CALLING ###

thresholds = seq(-0.9,4,by=0.9)
min.CN = 1

rco = applyThresholds(rco, thresholds, min.CN)
```

---

plotCNProfile *Plots the profile of a SeqCNAInfo-class object.*

Description

The plotted elements depend on the processing applied on the SeqCNAInfo-class object.

Usage

```
plotCNProfile(rco, folder = NULL)
```

Arguments

rco	A SeqCNAInfo-class object.
folder	Path to the folder where the plots with the output from the SeqCNAInfo-class object are to be generated. If no folder is indicated or does not exist, plots will be displayed within R.

Value

Nothing is returned from this function. Check the folder folder for a file called seqnorm_out.jpg.

Author(s)

David Mosen-Ansorena

Examples

```
data(seqsumm_HCC1143)
rco = readSeqsumm(tumour.data=seqsumm_HCC1143)
rco = applyFilters(rco, 0, 1, 0, 2, FALSE, plots=FALSE)
rco = runSeqnorm(rco, plots=FALSE)
rco = runGLAD(rco)
rco = applyThresholds(rco, seq(-0.9,4,by=0.9), 1)

plotCNProfile(rco)
```

`readSeqsumm` *Function that reads seqsumm summarized files.*

Description

The function reads the `seqsumm_out.txt` file(s) from the indicated directories and builds a `SeqCNAInfo-class` object, with read count (RC) and genomic information.

Usage

```
readSeqsumm(build="", tumour.data=NULL, normal.data=NULL, folder=NULL, normal.folder=NULL, resample.w
```

Arguments

`build`	String indicating the genome and build used to generate and annotate the output `SeqCNAInfo-class` object. Currently, the annotation package supports `hg18` and `hg19`. This means that common CNV and mappability filters are only available for these builds, and that GC content is estimated from the tumoural sample - or the normal sample if available.
`tumour.data`	A dataframe with the seqsumm information for the tumoural sample.
`normal.data`	If applicable, a dataframe with the seqsumm information for the normal sample. Otherwise, disregard this parameter.
`folder`	Path to the folder where the tumoural sample `seqsumm_out.txt` file is located. Only used if no data is passed through the `{tumour.data}` parameter.
`normal.folder`	If applicable, path to the folder where the paired normal sample `seqsumm_out.txt` file is located. Otherwise, disregard this parameter. Only used if no data is passed through the `{normal.data}` parameter.
`resample.win`	An integer that allows to specify a new bigger summarization window. If used, it must be an exact multiple of the window in the data read.
`sex`	A boolean indicating whether to read sex chromosomes into the output `SeqCNAInfo-class` object.
`nproc`	A value indicating how many processing cores to use for the process of resampling, if applicable. Greater values speed up resampling using more CPU cores and RAM memory, but you should not use values greater than the number of cores in your machine. If unsure, the safest value is 1, but most computers nowadays are multi-core, so you could probably go up to 2, 4 or 8.

Details

See seqsumm_HCC1143 for an example table read by the function.

Value

A SeqCNAInfo-class object, with information on read count (RC), genome build, and summarization window size and position. If applicable, it also contains paired normal RC. If paired-end mapping (PEM) was used in the alignment, RCs are broken down by read type.

Author(s)

David Mosen-Ansorena

Examples

```
data(seqsumm_HCC1143)
rco = readSeqsumm(tumour.data=seqsumm_HCC1143)
```

---

runGLAD *Wrapper function that runs the GLAD segmentation algorithm on a SeqCNAInfo-class object.*

Description

The function calls the corresponding function in the GLAD library with appropriate parameters.

Usage

```
runGLAD(rco, lambdabreak=8, nproc=2)
```

Arguments

rco	A SeqCNAInfo-class object, with read count (RC) and genomic information, normally the output of the runSeqnorm function.
lambdabreak	Penalty term used in GLAD during the optimization of the number of breakpoints step. Higher values result in lower amount of segments.
nproc	A value indicating how many processing cores to use in the segmentation. Greater values speed up segmentation using more CPU cores and RAM memory, but you should not use values greater than the number of cores in your machine. If unsure, the safest value is 1, but most computers nowadays are multi-core, so you could probably go up to 2, 4 or 8.

Value

A SeqCNAInfo-class object, with additional information on the segmented (i.e. smoothed) profile.

Author(s)

David Mosen-Ansorena

References

Analysis of array CGH data: from signal ratio to gain and loss of DNA regions. Hupe P, Stransky N, Thiery JP, Radvanyi F, Barillot E. Bioinformatics. 2004 Dec 12; 20(18):3413-22.

Examples

```
data(seqsumm_HCC1143)
rco = readSeqsumm(tumour.data=seqsumm_HCC1143)
rco = applyFilters(rco, 0, 1, 0, 2, FALSE, plots=FALSE)
rco = runSeqnorm(rco, rco@win, "quadratic", nproc=2, plots=FALSE)

### SEGMENTATION ###

rco = runGLAD(rco)
```

`runSeqnorm` *This function calls seqnorm, which normalizes the tumoural profile with respect to either a paired sample or GC content.*

Description

The seqnorm method performs several regressions on homogeneous regions of the tumoural genomic profile.

Usage

```
runSeqnorm(rco, norm.win = NULL, method = "quadratic", lambdabreak=8, minSeg=7, maxSeg=35, nproc = 2, pl
```

Arguments

`rco`	A `SeqCNAInfo-class` object, with read count (RC) and genomic information, normally the output of the `readSeqsumm` function.
`norm.win`	The genomic window size, given in kilobases, in which to normalize the read count profile. Leave it to the default NULL value for automatic selection. The window size should be the same as or a multiple of the summarization window in `runSeqsumm`. If not a multiple, the window size will be adjusted to the nearest one. For efficiency, we recommend to use the same value as the summarization window size, unless it is very small (e.g. 2Kbp).
`method`	A string being `"loess"`, `"cubic"` or `"quadratic"`, indicating which form of regression to use in the normalization. LOESS is more sensitive to noise and slower than the other two, which apply polynomial regression. In principle, we recommend using a quadratic polynomial regression, but there might be ocassions in which the others yield better fits.

lambdabreak	The lambda prime parameter in GLAD, indicating the penalty term used during the optimization of the number of breakpoints.
minSeg	The minimum number of segments in which the second pass regression is based. Too few would result in noisy estimates of the regression curve. We recommend to leave it to the default.
maxSeg	The minimum number of segments in which the second pass regression is based. Too many would include segments whose regression curve does not approximate the overall curve. Additionally, they would add additional execution time. We recommend to leave it to the default.
nproc	A value indicating how many processing cores to use in the segmentation prior to regression and in the segment regressions. Greater values speed up normalization using more CPU cores and RAM memory, but you should not use values greater than the number of cores in your machine. If unsure, the safest value is 1, but most computers nowadays are multi-core, so you could probably go up to 2, 4 or 8.
plots	A boolean value indicating whether to output normalization plots to the folder. These are useful for checking the correctness of the normalization and the improvements over the standard normalization.
folder	If the plots parameter is set to TRUE, path to the folder where the plots with normalization information are to be generated. If no folder is indicated or does not exist, plots will be displayed within R.

Value

A SeqCNAInfo-class object, with additional information on the normalized profile.

Author(s)

David Mosen-Ansorena

References

Analysis of array CGH data: from signal ratio to gain and loss of DNA regions. Hupe P, Stransky N, Thiery JP, Radvanyi F, Barillot E. Bioinformatics. 2004 Dec 12; 20(18):3413-22.

Examples

```
data(seqsumm_HCC1143)
rco = readSeqsumm(tumour.data=seqsumm_HCC1143)
rco = applyFilters(rco, 0, 1, 0, 2, FALSE, plots=FALSE)

### NORMALIZATION ###

rco = runSeqnorm(rco, plots=FALSE)
```

runSeqsumm *Function that generates seqsumm summarized files.*

Description

The function calls the seqsumm program to summarize aligned reads into genomic windows. The results are output to a file called `seqsumm_out.txt`.

Usage

```
runSeqsumm(summ.win=50, file=NULL, folder=NULL, output.file="seqsumm_out.txt", samtools.path="samtool
```

Arguments

`summ.win`	The genomic window size, expressed in kilobases as a whole positive number, in which to summarize the reads.
`file`	Full path to the SAM or BAM file. It is recommended to place it alone in a folder, where analysis output files will be generated.
`folder`	Only used if folder is not indicated. Path to the folder where the sample SAM or BAM is located. The function first searches for the SAM file and, if not existent, for the BAM file. If several SAM (or BAM) files exist in the folder, only the first one will be used. The summarization results will be written to this same folder.
`output.file`	The name of the file where the output will be stored (in the same folder as the SAM/BAM file). Be aware, if you choose an alterative name, that function `readSeqsumm` will look for `seqsumm_out.txt`.
`samtools.path`	The program SAMtools is required if your alignment file is in BAM format. In case SAMtools is not in your path variable, you should indicate the path to the executable.

Details

Beware that this is a computationally intensive task, so it might take from a few minutes to several hours depending on the number of reads in the aligned file.

Value

A file called `seqsumm_out.txt` (or the indicated name), which will be read by the function `readSeqsumm`, is generated in the indicated folder. Therefore, you only need to call this function once, unless you want to generate a new file with other input/parameters. The output file has chromosome, window, mean GC and mean mapping quality columns, and either a count column for single-end alignments or five count columns for paired-end alignments: - Type 1. Mapped with correct orientation and within insert size. - Type 2. Mapped with correct orientation and uniquely, but wrong insert size. - Type 3. Mapped within the insert size but wrong orientation. - Type 4. Mapped uniquely, but wrong orientation and insert size. - Type 5. One of the mates in unmapped. See `seqsumm_HCC1143` for an example table generated by the function. The function does not return anything.

Author(s)

David Mosen-Ansorena

Examples

```
file.copy(file.path(system.file(package="seqCNA"), "extdata/test.sam"), tempdir())
tempdir()

window.size = 50

runSeqsumm(window.size, folder=tempdir())
```

SeqCNAInfo-class *Class* "SeqCNAInfo"

Description

A structure that contains information on depth of coverage (DOC), genome build, and summarization window size and position. If applicable, it also contains paired normal DOC. If paired-end mapping (PEM) was used in the alignment, DOCs are broken down by read type. Further processing with seqCNA functions of these objects, attach filtering information, and normalized, segmented and called profiles.

Objects from the Class

Objects can be created by calls to `readSeqsumm` and further modified by `applyFilters`, `runSeqnorm`, `runGLAD` and `applyThresholds`.

Slots

`tumour`: A list with either 1 or 5 numeric vectors, depending on whether the reads are single- or paired-end. Each vector represents the read count for a specific group of read types in a window of size `win`. The first vector corresponds to the proper reads.

`normal`: A list analogue to `tumour`, but for the normal sample. If there is no information on a normal sample, the list has length zero.

`seq`: A vector indicating the chromosome of each window.

`pos`: A vector indicating the index of each window within the corresponding chromosome.

`build`: Genome and build of the sample. Available builds in the package (hg18, hg19) enable some filters and better GC correction.

`win`: An integer indicating the window size in kilobases.

`x`: The profile against which to perform normalization. Either GC content (estimated or from annotation) or paired-normal read count profile.

`y`: The profile to be normalized, this is, the read count profile of the tumoural sample.

`skip`: A vector with the indexes of the windows that are to be discarded prior to normalization.

`output`: A dataframe with a minimum of 3 columns: chromosome, window start position and normalized profile values. With further processing of the object, columns for the segmented and called profiles are appended.

`thr`: A vector with values that establish the ranges for each copy number.

`minCN`: The copy number of the lowest range defined by the thresholds.

`gc`: Mean GC content of the reads within each genomic window. The means from the tumoural sample are used unless there is a paired-normal sample.

`mapq`: Mean mapping quality of the reads within each genomic window. The means from the tumoural sample are used unless there is a paired-normal sample.

Methods

summary `signature(object = "SeqCNAInfo")`: Prints on screen a summary of the object and the applied processing.

Author(s)

David Mosen-Ansorena

Examples

```
showClass("SeqCNAInfo")
```

---

seqsumm_HCC1143 *Example of a seqsumm output, namely for a sequenced sample of the HCC1143 cancer cell-line.*

Description

The `runSeqsumm` function was run over a SAM file at a summarization window of 200Kbps and only chromosomes 1 to 5 were kept.

Usage

```
data(seqsumm_HCC1143)
```

Format

A data frame with 5314 genomic windows on the following 5 variables.

`chrom` A numeric vector with the chromosome of the corresponding window.

`win.start` A numeric vector with the first zero-based base pair index of the corresponding window.

`reads.gc` A numeric vector with the mean GC content for the reads within the corresponding window.

`reads.mapq` A numeric vector with the mean mapping quality for the reads within the corresponding window.

`counts` A numeric vector with the read count within the corresponding window.

References

High-resolution mapping of copy-number alterations with massively parallel sequencing. Chiang DY, Getz G, Jaffe DB, O'Kelly MJ, Zhao X, Carter SL, Russ C, Nusbaum C, Meyerson M, Lander ES. Nature. 2010 Jan; 6(1):99-103

Examples

```
data(seqsumm_HCC1143)
head(seqsumm_HCC1143)
```

`writeCNProfile` *Outputs the processed profile of a* `SeqCNAInfo-class` *object to the specified folder.*

Description

The output columns depend on the processing applied on the `SeqCNAInfo-class` object.

Usage

```
writeCNProfile(rco, folder)
```

Arguments

`rco`	A `SeqCNAInfo-class` object.
`folder`	Path to the folder where the table with the output from the `SeqCNAInfo-class` object is to be generated.

Value

Nothing is returned from this function. Check the `folder` folder for a text file called `seqCNA_out.txt`.

Author(s)

David Mosen-Ansorena

Examples

```
data(seqsumm_HCC1143)
rco = readSeqsumm(tumour.data=seqsumm_HCC1143)
rco = applyFilters(rco, 0, 1, 0, 2, FALSE, plots=FALSE)
rco = runSeqnorm(rco, plots=FALSE)
rco = runGLAD(rco)
rco = applyThresholds(rco, seq(-0.9,4,by=0.9), 1)

writeCNProfile(rco, tempdir())
```

Index