Package 'CopyNumber450k'

October 9, 2015

Title R package for calling CNV from Illumina 450k methylation microarrays

Version 1.4.0

Date 2014-02-04

Author Simon Papillon-Cavanagh, Jean-Philippe Fortin, Nicolas De Jay

Maintainer Simon Papillon-Cavanagh

<simon.papillon-cavanagh@mail.mcgill.ca>

Description This package contains a set of functions that allow CNV calling from Illumina 450k methylation microarrays.

License Artistic-2.0

Imports methods

Depends Biobase, minfi, DNAcopy, preprocessCore, BiocGenerics

Suggests CopyNumber450kData, minfiData

LinkingTo

LazyData yes

Collate AllClasses.R AllGenerics.R CNV450kSet-accessors.R CNV450kSet-actions.R CNV450kSet-comparison.R CNV450kSet-initialize.R CNV450kSet-plot.R extract.R normalization.functional.R normalization.quantile.R

biocViews DNAMethylation, Microarray, Preprocessing, QualityControl, CopyNumberVariation

NeedsCompilation no

R topics documented:

CNV450kSet-class *CNV450kSet instances*

Description

This class holds Illumina 450k methylation microarray data and annotations for CNV calling.

Usage

```
## Constructor
CNV450kSet(RGChannelSet)
```

Arguments

RGChannelSet	A class defined in the minfi package that represents raw (unprocessed) data from a two color microarray; specifically an Illumina methylation array.

Details

This class inherits from eSet. The class is a representation of an intensity matrix that is computed from the MethylSet class produced by RGChannelSet in the minfi package.

Constructor

Instances are constructed using the CNV450kSet function with the arguments outlined above.

Accessors

This class has a few accessors defined on top of those provided in eSet. In the following code, object is a CNV450kSet.

getManifest(object) Gets the manifest associated with the object.

getSegments(object) Gets list of segments produced by the segmentize method.

getSummary(object) Gets statistics extracted from raw data (RGChannelSet) and preprocessed methylation data (MethylSet) that are used internally.

Methods

dropSNPprobes(object, maf_threshold = 0) Removes probes mapping to or targeting known SNPs from consideration. Returns a new CNV450kSet object in which SNP-containing probes are discarded.

normalize(object, type = c("functional", "quantile")) Normalizes the intensity matrix and returns normalized object. Refer to the vignette for details on the types of normalization.

segmentize(object, verbose = TRUE, p.adjust.method = "bonferroni") Arranges probes into segments related by signal intensity and proximity by circular binary segmentation. Certain class methods require that segments be created. Refer to the vignette for more details on the algorithm.

`computeSignificance(object, p.value.threshold = 0.01, num.mark.threshold = 10)` Computes the significance for each segment in each sample. Requires that `segmentize` be called on the object beforehand.

Comparison Methods

The following methods require that `segmentize` be called on the object before their calls.

`findCNV(object, gene_names, type = c("gain", "loss", "both"))` Returns a matrix in which `i, j` denotes the presence or absence of a CNV event for gene `i` and sample `j`. `gene_names` is a vector containing the gene symbols of interest.

`intersectCNV(object, sample_indices, type = c("gain", "loss", "both"))` Returns a vector of gene symbols corresponding to gain or loss of genes within a group, sorted by CNV abundance; `sample_indices` is a vector containing the indices of the samples belonging to the group.

`subgroupDifference(object, group1_indices, group2_indices)` Returns two vectors (gains and losses) containing Fisher's exact test p-values on gene-based CNV counts between two sample groups (or conditions); `group1_indices` and `group2_indices` are vectors containing the indices of the samples belonging to each respective group.

Plotting Methods

`plotSample(object, index, chr, start, end)` Produces a plot of the genomic segments and relative values for sample at `index`. `chr, start ,end` are optional parameters to be used to zoom in a specific genomic location.

`plotDensity(object, color.by = c("array.row", "array.col", "sample.group", "slide", "origin"),\ col` Plots the density distribution of the intensity matrix of the object.

`plotPCA(object, color.by = c("array.row", "array.col", "sample.group", "slide", "origin"),\ color.f` Plots the PCA scatter plot of the intensity matrix of the object.

`write.csv(object, ...)` Writes the segment output for each sample in csv format.

Author(s)

Simon Papillon-Cavanagh, Jean-Philippe Fortin, Nicolas De Jay

See Also

`eSet` for the basic class structure. Objects of this class are typically created from a `RGChannelSet` using the constructor.

Examples

```
library(CopyNumber450kData)
library(minfiData)

# Load the CopyNumber450kData control set
data(RGcontrolSetEx)

# Load example data (n=6) from minfiData
```

```
data(RGsetEx)
# In order to reduce example time, let's use only one sample
# and 30 controls (instead of 52). In real life situations, it is advised to
# use all the available controls
RGsetEx <- RGsetEx[, 5]
RGcontrolSetEx <- RGcontrolSetEx[, sample(1:ncol(RGcontrolSetEx), 30)]

# Combine both RGsets in a single RGset
RGset <- combine(RGcontrolSetEx, RGsetEx)

# Create the object
mcds <- CNV450kSet(RGset)

# In order to speed up example computation, we will randomly subset the
# probes used by CopyNumber450k. THIS SHOULD NEVER BE DONE AS IT SERVES
# ONLY FOR SPEEDING UP THE EXAMPLE.
mcds <- mcds[sample(1:nrow(mcds), 10000), ]

# Drop SNP probes
mcds <- dropSNPprobes(mcds, maf_threshold=0.01)

# Normalization
mcds <- normalize(mcds, "quantile")

# Some plots
plotDensity(mcds, main="Density plot of functional normalized data")
plotPCA(mcds, main="PCA plot of functional normalized data")

# Segmentation
mcds <- segmentize(mcds)

# Plotting the results
plotSample(mcds, 1, main="Genomic view of Sample 1")
plotSample(mcds, 1, chr="chr1", ylim=c(-.25,.25))

# Saving the results in csv format
write.csv(mcds, file="segments.csv")
```

Index