Package 'SomaticSignatures'

October 8, 2014

Type Package

Title Somatic Signatures

Version 1.0.1

Date 2014-05-11

Author Julian Gehring, with contribution of Bernd Fischer (EMBL Heidelberg)

Maintainer Julian Gehring <julian.gehring@embl.de>

Description The SomaticSignatures package identifies mutational signatures of single nucleotide variants (SNVs).

URL http://bioconductor.org/packages/release/bioc/html/SomaticSignatures.html,https://github.com/julian-gehring/SomaticSignatures

Imports GenomeInfoDb, GenomicRanges, IRanges, VariantAnnotation,Biostrings, ggplot2, ggbio, stringr, reshape2, NMF, methods,pcaMethods, gtools

Depends R (>= 3.0.2)

Suggests testthat, knitr, BiocStyle, parallel,BSgenome.Hsapiens.UCSC.hg19, SomaticCancerAlterations, h5vc,h5vcData, fastICA

VignetteBuilder knitr

ByteCompile TRUE

License GPL-3

LazyLoad yes

biocViews
Sequencing, SomaticMutation, Visualization, Clustering,HighThroughputSequencingData, Cancer, PrinicpalComponent,GenomicVariation, StatisticalMethod

R topics documented:

`gcContent` *GC Content*

Description

Compute the GC content for regions of a reference sequence.

Usage

```
gcContent(regions, ref)
```

Arguments

`regions`	GRanges object with the regions for which the GC content should be computed.
`ref`	Reference sequence object, as a 'BSgenome' or 'FaFile' object.

Value

A numeric vector with the GC content [0,1] for each region.

See Also

Inspired by the 'getGCcontent' function of the 'exomeCopy' package.

Examples

```
library(BSgenome.Hsapiens.UCSC.hg19)

regs = GRanges(c("chr1", "chr2"), IRanges(1e7, width = 100))

gc = gcContent(regs, BSgenome.Hsapiens.UCSC.hg19)
```

GRanges-converters *GRanges converter functions*

Description

A set of utilities functions to convert and extract data in 'GRanges' objects.

Usage

```
grangesPlain(x)
ncbi(x)
ucsc(x)
seqchar(x)
```

Arguments

x	A 'GRanges' object or one inheriting from the 'GRanges' class [required].

Details

- grangesPlainExtracts only the 'GRanges' information by dropping the metadata columns of the object. The 'seqinfo' slot is kept.
- ncbi, ucscShorthand for converting the seqnames notation to 'UCSC' (e.g. 'chr1', 'chrM') or 'NCBI' (e.g. '1', 'MT") notation, respectively. This also sets the 'genome' slot in the 'seqinfo' field to 'NA'.
- seqcharExtracts the 'seqnames' as a character vector.

Value

For 'grangesPlain': A GRanges object without metadata.

For 'ncbi', 'ucsc': An object of the same class as the input.

For 'seqchar': A character vector with 'seqnames'.

See Also

'GenomicRanges' package: 'seqnames', 'mcols'

'GenomeInfoDb' package: 'seqlevelsStyle'

Examples

```
mutect_path = system.file("examples", "mutect.tsv", package = "SomaticSignatures")
vr1 = readMutect(mutect_path, strip = TRUE)

## extract the GRanges
gr = grangesPlain(vr1)
```

```
## convert back and forth
gr_ncbi = ncbi(gr)
gr_ucsc = ucsc(gr_ncbi)

identical(gr, gr_ucsc)

## extract the seqnames as a character vector
seq_chars = seqchar(gr)
```

---

hs-chrs *Human Chromosome Names*

Description

List human chromosome names.

Usage

```
hsToplevel()
hsAutosomes()
hsAllosomes()
hsLinear()
```

Value

Character vector with chromosome names (NCBI notation).

Examples

```
hsToplevel()

hsAutosomes()

hsAllosomes()

hsLinear()
```

---

kmerFrequency *Kmer Frequency*

Description

Estimate the occurance frequency of k-mers in a reference sequence.

Usage

```
kmerFrequency(ref, n = 1e4, k = 1, ranges = as(seqinfo(ref), "GRanges"))
```

Arguments

ref	A 'BSgenome' or 'FaFile' object matching the respective reference sequence [required].
n	The number of samples to draw [integer, default: 1e4].
k	The 'k'-mer size of the context, including the variant position [integer, default: 3].
ranges	Ranges in respect to the reference sequence to sample from [GRanges, default: take from the 'seqinfo' slot].

Details

The k-mer frequency is estimated by random sampling of 'n' locations across the specified 'ranges' of the reference sequence.

Value

A named vector, with names corresponding the the k-mer and value to the frequency.

Examples

```
library(BSgenome.Hsapiens.UCSC.hg19)

kmer_freq = kmerFrequency(BSgenome.Hsapiens.UCSC.hg19, 1e2, 3)
```

---

kmers-data *Kmer datasets*

Description

3mer base frequencies of human whole-genome and whole-exome sampling, based on the hg19/GRCh37 reference sequence.

For details, see the 'inst/scripts/kmers-data.R' script.

See Also

kmerFrequency

Examples

```
data(kmers, package = "SomaticSignatures")
```

`mutation-distribution` *Distributions of mutational locations.*

Description

Summary and plotting function for characterizing the distributions of mutations along the genome.

Usage

```
mutationDistance(x)

plotRainfall(x, group, size = 2, alpha = 0.5, space.skip = 0, ...)
```

Arguments

`x`	A 'GRanges' or 'VRanges' object [required].
`group`	The variable name for color groups [optional].
`size`	Point size [default: 2]
`alpha`	Alpha value for points [default: 0.5]
`space.skip`	Space between chromosomes, as defined by 'plotGrandLinear' [default: 0]
`...`	Additional arguments passed to 'plotGrandLinear'

Value

- mutationDensityThe position-sorted GRanges 'x' with the additional column 'distance', specifying the distance from the previous mutation (or the beginning of the chromosome if it happens to be the first mutation on the chromosome.)
- plotRainfallObject of class 'ggbio', as returned by 'plotGrandLinear'.

See Also

'ggbio::plotGrandLinear'

Examples

```
library(GenomicRanges)
library(IRanges)

set.seed(1)
chr_len = 100
gr = GRanges(rep(1:3, each = 10),
  IRanges(start = sample.int(chr_len, 30, replace = FALSE), width = 1),
  mutation = sample(c("A", "C", "G", "T"), 30, replace = TRUE))
seqlengths(gr) = rep(chr_len, 3)

p = plotRainfall(gr)
```

```
print(p)
```

`mutational-normalization` *Normalize Somatic Motifs*

Description

Normalize somatic motifs, to correct for biases between samples.

Usage

```
normalizeMotifs(x, norms)
```

Arguments

`x`	Matrix, as returned by 'mutationContextMatrix' [required]
`norms`	Vector with normalization factors [required]. The names must match the base sequence names in 'x'.

Value

A matrix as 'x' with normalized counts.

See Also

mutationContextMatrix

`mutational-plots` *Mutational Plots*

Description

Plots for variant analysis

Usage

```
plotVariantAbundance(x, group = NULL, alpha = 0.5, size = 2)
```

Arguments

`x`	A VRanges object [required].
`group`	Grouping variable, refers to a column name in 'x'. By default, no grouping is performed.
`alpha`	Alpha value for data points.
`size`	Size value for data points.

Details

The 'plotVariantAbundance' shows the variant frequency in relation to the total coverage at each variant position. This can be useful for examining the support of variant calls.

Value

A 'ggplot' object.

`mutational-signatures` *Estimate Somatic Signatures*

Description

Estimate somatic signatures from sequence motifs with a selection of statistical methods.

Usage

```
mutationContextMatrix(x, group = "sample", normalize = TRUE)

findSignatures(x, r, method = c("nmf", "pca", "kmeans"), ...)
nmfSignatures(x, r, seed = "ica", ...)
kmeansSignatures(x, r, ...)
pcaSignatures(x, r, ...)

plotSamplesObserved(s, group = "study")
plotSignatureMap(s)
plotSignatures(s)
plotSampleMap(s)
plotSamples(s)
```

Arguments

`x`	GRanges object [required]
`group`	Grouping variable name [character, default: 'sample']
`normalize`	Normalize to frequency
`r`	Number of signatures [integer, required]
`method`	Method to apply (currently: 'nmf' or 'kmeans')
`seed`	seed for NMF, default: "ica"
`...`	Additional arguments passed to 'NMF::nmf'
`s`	results signature object

Details

The 'mutationContextMatrix' function transforms the metadata columns of a 'VRanges' object, as returned by the 'mutationContext' function, to a matrix of the form 'motifs x samples'. This constitutes the bases for the estimation of the signatures.

The 'nmfSignatures', 'pcaSignatures', and 'kmeansSignatures' functions estimate a set of 'r' somatic signatures using the NMF, PCA, or k-means clustering, respectively.

With the plotting function, the obtained signatures and their occurrance in the samples can be visualized either as a heatmap ('plotSignatureMap', 'plotSampleMap') or a barchart ('plotSignature', 'plotSamples').

See Also

'mutationContext', 'mutationContextMutect'

'NMF' package, 'pcaMethods' package, 'prcomp', 'kmeans'

`mutationContext` *mutationContext functions*

Description

Extract the sequence context surrounding a SNV from a genomic reference.

Usage

```
mutationContext(vr, ref, k = 3, strand = FALSE, unify = TRUE, check = TRUE)
mutationContextMutect(vr, k = 3, unify = TRUE)
mutationContextH5vc(vc, ms, unify = TRUE)
```

Arguments

`vr`	'VRanges' object, with 'ref' and 'alt' columns filled [required]. For 'mutationContextMutect', an object as returned by the 'readMutect' function.
`ref`	A 'BSgenome' or 'FaFile' object representing the reference sequence [required]. More generally, any object with a defined 'getSeq' method can be used.
`k`	The 'k'-mer size of the context, including the variant position [integer, default: 3]. The variant will be located at the middle of the k-mer which requires 'k' to be odd.
`strand`	Should all variants be converted to the 'plus' strand? [logical, default: FALSE].
`unify`	Should the alterations be converted to have a C/T base pair as a reference alleles? [logical, default: TRUE]
`check`	Should the reference base of 'vr' be checked against 'ref' [logical, default: TRUE]? In case the two references do not match, a warning will be printed.
`vc`	A 'DataFrame' object as returned from a variant calling analysis by 'h5vc::h5dapply'. See the 'details' section for more information.
`ms`	A 'DataFrame' object as returned by 'h5vc::mutationSpectrum'. See the 'details' section for more information.

Details

The somatic motifs of a SNV, composed out of (a) the base change and (b) the sequence context surrounding the variant, is extracted from a reference sequence with the 'mutationContext' function.

For mutect variant calls, all relevant information is already contained in the results and somatic motifs can constructed by using the 'mutationContextMutect' function, without the need for the reference sequence.

For h5vc variant calls, the information is merged from the outputs of the 'h5dapply' and 'mutationSpectrum' functions of the 'h5vc' package. A detailed example is shown in the vignette of the package.

Value

The original 'VRanges' object 'vr', with the additional columns

`alteration`	DNAStringSet with 'reflalt'.
`context`	DNAStringSet with '..N..' of length 'k', where N denotes the variant position.

See Also

'readMutect' for 'mutationContextMutect'

'mutationSpectrum' from the 'h5vc' package for 'mutationContextH5vc'

Examples

```
mutect_path = system.file("examples", "mutect.tsv", package = "SomaticSignatures")
vr1 = readMutect(mutect_path)
ct1 = mutationContextMutect(vr1)


library(h5vc)
library(h5vcData)

tally_file = system.file("extdata", "example.tally.hfs5", package = "h5vcData")
data("example.variants", package = "h5vcData")
ms = mutationSpectrum(variantCalls, tally_file, "/ExampleStudy", context = 1)
vc = variantCalls

vr2 = mutationContextH5vc(vc, ms)
vr2$sample = sampleNames(vr2)
ct2 = mutationContextMatrix(vr2)
```

readMutect *readMutect*

Description

Import 'mutect' calls.

Usage

```
readMutect(file, columns, strip = FALSE)
```

Arguments

file	Location of the mutect tsv files [character, required]
columns	Names of columns to import from the file [character vector, optional, default: missing]. If missing, all columns will be imported.
strip	Should additional columns be imported? [logical, default: FALSE]. If TRUE, return only the bare 'VRanges' object.

Details

The 'readMutect' functions imports the mutational calls of a '*.tsv' file returned by the 'mutect' caller to a 'VRanges' object. For a description of the information of the columns, please refer to the mutect documentation.

Value

A 'VRanges' object, with each row corresponding to one variant in the original file.

References

Cibulskis, Kristian, Michael S. Lawrence, Scott L. Carter, Andrey Sivachenko, David Jaffe, Carrie Sougnez, Stacey Gabriel, Matthew Meyerson, Eric S. Lander, and Gad Getz. "Sensitive Detection of Somatic Point Mutations in Impure and Heterogeneous Cancer Samples." Nature Biotechnology advance online publication (February 10, 2013). doi:10.1038/nbt.2514.

http://www.broadinstitute.org/cancer/cga/mutect_run

Examples

```
mutect_path = system.file("examples", "mutect.tsv", package = "SomaticSignatures")
vr1 = readMutect(mutect_path)
vr2 = readMutect(mutect_path, strip = TRUE)
```

SomaticSignatures *SomaticSignatures package*

Description

Identifying somatic signatures of single nucleotide variants.

Details

The 'SomaticSignatures' package offers the framework for identifying mutational signatures of single nucleotide variants (SNVs) from high-throughput experiments. In the concept of mutational signatures, a base change resulting from an SNV is regarded in term of motifs which embeds the variant in the context of the surrounding genomic sequence. Based on the frequency of such motifs across samples, mutational signatures and their occurrance in the samples can be estimated. An introduction into the methodology and a use case are illustrated in the vignette of this package.

Author(s)

Julian Gehring, with contributions from Bernd Fischer and Wolfgang Huber.

Maintainer: Julian Gehring, EMBL Heidelberg <julian.gehring@embl.de>

References

Nik-Zainal, Serena, Ludmil B. Alexandrov, David C. Wedge, Peter Van Loo, Christopher D. Greenman, Keiran Raine, David Jones, et al. "Mutational Processes Molding the Genomes of 21 Breast Cancers." Cell 149, no. 5 (May 25, 2012): 979-993. doi:10.1016/j.cell.2012.04.024.

Alexandrov, Ludmil B., Serena Nik-Zainal, David C. Wedge, Samuel A. J. R. Aparicio, Sam Behjati, Andrew V. Biankin, Graham R. Bignell, et al. "Signatures of Mutational Processes in Human Cancer." Nature 500, no. 7463 (August 22, 2013): 415-421. doi:10.1038/nature12477.

Gaujoux, Renaud, and Cathal Seoighe. "A Flexible R Package for Nonnegative Matrix Factorization." BMC Bioinformatics 11, no. 1 (July 2, 2010): 367. doi:10.1186/1471-2105-11-367.

Stacklies, Wolfram, Henning Redestig, Matthias Scholz, Dirk Walther, and Joachim Selbig. "pcaMethods - A Bioconductor Package Providing PCA Methods for Incomplete Data." Bioinformatics 23, no. 9 (May 1, 2007): 1164-1167. doi:10.1093/bioinformatics/btm069.

Examples

```
vignette(package = "SomaticSignatures")
```

`variants-utils` *Utility functions*

Description

Utility functions

Usage

```
dfConvertColumns(x, from = "character", to = "factor")
```

Arguments

`x`	A 'data.frame' to convert [required].
`from`	The class of the columns to be converted [default: 'character'].
`to`	The class of the columns to be converted to [default: 'factor'].

Details

The 'dfConvertColumns' converts all columns of a data frame with class 'from' to the class 'to'.

Value

A 'data.frame' object.

Index