

# Package ‘SomaticSignatures’

April 10, 2015

**Type** Package

**Title** Somatic Signatures

**Version** 2.2.3

**Author** Julian Gehring (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** S4Vectors, IRanges, GenomeInfoDb, Biostrings, ggplot2, ggbio, reshape2, NMF, pcaMethods, Biobase, methods, proxy

**Depends** R (>= 3.0.2), VariantAnnotation, GenomicRanges

**Suggests** testthat, knitr, parallel, BSgenome.Hsapiens.UCSC.hg19, SomaticCancerAlterations, COSMIC.67, ggdendro, fastICA

**VignetteBuilder** knitr

**ByteCompile** TRUE

**License** GPL-3

**LazyLoad** yes

**biocViews** Sequencing, SomaticMutation, Visualization, Clustering, GenomicVariation, StatisticalMethod

## R topics documented:

|                                    |   |
|------------------------------------|---|
| cluster-spectrum . . . . .         | 2 |
| decomposition-signatures . . . . . | 3 |
| gcContent . . . . .                | 4 |
| GRanges-converters . . . . .       | 4 |
| hs-chrs . . . . .                  | 6 |
| kmerFrequency . . . . .            | 6 |
| kmers-data . . . . .               | 7 |

|                                    |    |
|------------------------------------|----|
| motif-functions . . . . .          | 8  |
| mutation-distribution . . . . .    | 9  |
| mutational-normalization . . . . . | 10 |
| mutational-plots . . . . .         | 10 |
| mutational-signatures . . . . .    | 11 |
| MutationalSignatures . . . . .     | 12 |
| mutationContext . . . . .          | 13 |
| readMutect . . . . .               | 14 |
| sca-data . . . . .                 | 15 |
| scaSNVRanges . . . . .             | 15 |
| signature-plots . . . . .          | 16 |
| signatures21-data . . . . .        | 17 |
| SomaticSignatures . . . . .        | 18 |
| variants-utils . . . . .           | 19 |

|              |           |
|--------------|-----------|
| <b>Index</b> | <b>20</b> |
|--------------|-----------|

---

|                  |                                    |
|------------------|------------------------------------|
| cluster-spectrum | <i>Cluster Mutational Spectrum</i> |
|------------------|------------------------------------|

---

## Description

Cluster the mutational spectrum by sample or motif.

## Usage

```
clusterSpectrum(m, by = c("sample", "motif"), distance = "Cosine", ...)
```

## Arguments

|          |   |
|----------|---|
| m        | Mutational spectrum matrix                |
| by       | Dimension to cluster by.                  |
| distance | Distance function used in the clustering. |
| ...      | Additional arguments passed to 'hclust'.  |

## Details

Hierarchical clustering of the motif matrix aka mutational spectrum.

## Value

An 'hclust' object.

## See Also

hclust  
'proxy' package

---

decomposition-signatures

*Decomposition Functions for Somatic Signatures*

---

## Description

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

## Usage

```
nmfDecomposition(x, r, ..., includeFit = FALSE)
pcaDecomposition(x, r, ..., includeFit = FALSE)
```

## Arguments

|            |  |
|------------|--|
| x          | GRanges object [required]  |
| r          | Number of signatures [integer, required]   |
| ...        | Additional arguments passed to 'NMF::nmf' or 'pcaMethods::pca'.                        |
| includeFit | Include the fit object returned by the low-level decomposition function in the output. |

## Details

The 'nmfDecomposition' and 'pcaDecomposition' functions estimate a set of 'r' somatic signatures using the NMF or PCA, respectively.

In previous versions of the package, these functions were known as 'nmfSignatures' and 'pcaSignatures', respectively. While they are still available, we recommend using the new naming convention.

## Value

The 'signature' functions return a list with the elements:

- wMatrix of the form 'motif x signature'
- hMatrix of the form 'sample x signature'
- vMatrix of the form 'motif x sample', containing the reconstruction of 'm' from 'w' and 'h'.
- mInput matrix 'm'
- rNumber of signatures.
- fitFit object returned by the low-level decomposition function, if 'includeFit' is true.

## See Also

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

---

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**

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

## Arguments

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

## Details

- `grangesExtracts` 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 '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 = granges(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 | <i>Human Chromosome Names</i> |
|---------|-------------------------------|

---

**Description**

List human chromosome names.

**Usage**

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

**Value**

Character vector with chromosome names (NCBI notation).

**Examples**

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

---

|               |                       |
|---------------|-----------------------|
| kmerFrequency | <i>Kmer Frequency</i> |
|---------------|-----------------------|

---

**Description**

Estimate the occurrence 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.

**Value**

Vectors with frequency of k-mers.

**See Also**

kmerFrequency

**Examples**

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

---

motif-functions      *Group somatic motifs*

---

### Description

Tabulate somatic motifs by a grouping variable.

### Usage

```
motifMatrix(vr, group = "sampleNames", normalize = TRUE)
```

### Arguments

|           |  |
|-----------|--|
| vr        | GRanges object [required]                                  |
| group     | Grouping variable name [character, default: 'sampleNames'] |
| normalize | Normalize to frequency                                     |

### Details

The 'motifMatrix' function transforms the metadata columns of a 'VRanges' object, as returned by the 'mutationContext' function, to a matrix of the form 'motifs x groups'. This constitutes the bases for the estimation of the signatures. By default (with 'normalize' set to TRUE), the counts are transformed to frequencies, such that the sum of frequencies of each group equal 1. Otherwise (with 'normalize' set to FALSE), the counts for each motifs in a group is returned.

### Value

Occurance matrix with motifs in rows and samples in columns.

### See Also

'mutationContext', 'mutationContextMutect'

### Examples

```
## Not run:  
motifMatrix(sca_motifs, group = "study")  
  
## End(Not run)
```



---

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 'motifMatrix' [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

motifMatrix

---

```
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**

```
identifySignatures(m, nSigs, decomposition = nmfDecomposition, ...)
```

**Arguments**

|                            |   |
|----------------------------|---|
| <code>m</code>             | Motif matrix, as returned by <code>'motifMatrix'</code> [required].   |
| <code>nSigs</code>         | Number of signatures [integer, required].   |
| <code>decomposition</code> | Function to apply for the matrix decompositon. The methods NMF and PCA are already implemented in the functions <code>'nmfDecomposition'</code> and <code>'pcaDecomposition'</code> , respectively. |
| <code>...</code>           | Additional arguments passed to the <code>'decomposition'</code> function.   |

**Details**

`'identifySignatures'` estimate a set of `'r'` somatic signatures, based on a matrix decompositon method (such as NMF, PCA).

**Value**

An object of class `'MutationalSignatures'`.

**See Also**

The predefined decomposition functions: `'nmfDecomposition'` and `'pcaDecomposition'`  
`'mutationContext'`, `'mutationContextMutect'`  
`'motifMatrix'`  
class `'MutationalSignatures'`

**Examples**

```
data("sca_mm", package = "SomaticSignatures")  
  
sigs = identifySignatures(sca_mm, 5)
```

---

MutationalSignatures *'MutationalSignatures' class and methods*

---

**Description**

Object representing of somatic signatures.

**Usage**

```
## S4 method for signature MutationalSignatures  
signatures(object)  
  
## S4 method for signature MutationalSignatures  
samples(object)  
  
## S4 method for signature MutationalSignatures  
observed(object)  
  
## S4 method for signature MutationalSignatures  
fitted(object)  
  
## S4 method for signature MutationalSignatures  
show(object)
```

**Arguments**

object            *'MutationalSignatures'* object

**Value**

```
help("MutationalSignatures")
```

**See Also**

identifySignatures

---

|                              |                                  |
|------------------------------|----------------------------------|
| <code>mutationContext</code> | <i>mutationContext functions</i> |
|------------------------------|----------------------------------|

---

### 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

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

---

|            |                   |
|------------|-------------------|
| readMutect | <i>readMutect</i> |
|------------|-------------------|

---

**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](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)
```

---

sca-data

*SomaticCancerAlterations Results*

---

## Description

Motif matrix and 5 estimated signatures (NMF) from the 'SomaticCancerAlterations' dataset. For details, see the vignette of the 'SomaticSignatures' package.

## See Also

'SomaticCancerAlterations' package

## Examples

```
data(sca_mm, package = "SomaticSignatures")
data(sca_sigs, package = "SomaticSignatures")
```

---

scaSNVRanges

*SNV VRanges from SCA dataset*

---

## Description

Create VRanges for somatic SNV calls in the SomaticCancerAlterations dataset.

## Usage

```
scaSNVRanges(chrs = hsAutosomes())
```

**Arguments**

chrs                      Chromosomes to include in the results. Defaults to human autosomes.

**Value**

A 'VRanges' object with somatic SNV calls.

**See Also**

'SomaticCancerAlterations' package

---

signature-plots                      *Plot Mutational Signatures*

---

**Description**

Visualize estimated signatures, sample contribution, and mutational spectra.

**Usage**

```
plotObservedSpectrum(s, colorby = c("sample", "alteration"))
plotFittedSpectrum(s, colorby = c("sample", "alteration"))

plotMutationSpectrum(vr, group, colorby = c("sample", "alteration"))

plotSignatureMap(s)
plotSignatures(s, normalize = TRUE, percent = FALSE)

plotSampleMap(s)
plotSamples(s, normalize = FALSE, percent = FALSE)
```

**Arguments**

s                              MutationalSignatures object [required]

vr                             VRanges object

colorby                      Which variable to use for the coloring in the spectra representation.

normalize                    Plot relative contributions (TRUE) instead of absolute (TRUE) ones.

percent                      Display the results as fraction (FALSE) or percent (TRUE).

group                        Grouping variable

**Details**

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



**Value**

A ggplot object

**See Also**

'ggplot2' package

**Examples**

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

```
plotSamples(sigs_nmf)
```

```
plotSignatures(sigs_nmf)
```

---

|                   |                      |
|-------------------|----------------------|
| signatures21-data | <i>21 Signatures</i> |
|-------------------|----------------------|

---

**Description**

Published signatures, taken from <ftp://ftp.sanger.ac.uk/pub/cancer/AlexandrovEtAl/signatures.txt>

**References**

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): 415-21. doi:10.1038/nature12477.

**Examples**

```
data(signatures21, package = "SomaticSignatures")
```

```
head(signatures21)
```

---

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 occurrence 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, Bernd Fischer, Michael Lawrence, Wolfgang Huber: SomaticSignatures: Inferring Mutational Signatures from Single Nucleotide Variants. 2014, bioRxiv preprint, <http://dx.doi.org/10.1101/010686>

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

- \*Topic **IO**
  - readMutect, [14](#)
- \*Topic **datasets**
  - kmers-data, [7](#)
  - sca-data, [15](#)
  - signatures21-data, [17](#)
- \*Topic **manip**
  - GRanges-converters, [4](#)
  - mutationContext, [13](#)
- \*Topic **package**
  - SomaticSignatures, [18](#)
- \*Topic **utilities**
  - GRanges-converters, [4](#)
- cluster-spectrum, [2](#)
- clusterSpectrum (cluster-spectrum), [2](#)
- decomposition-signatures, [3](#)
- dfConvertColumns (variants-utils), [19](#)
- findSignatures (mutational-signatures), [11](#)
- fitted (MutationalSignatures), [12](#)
- fitted, MutationalSignatures-method (MutationalSignatures), [12](#)
- gcContent, [4](#)
- GRanges-converters, [4](#)
- hs-chrs, [6](#)
- hsAllosomes (hs-chrs), [6](#)
- hsAutosomes (hs-chrs), [6](#)
- hsLinear (hs-chrs), [6](#)
- hsToplevel (hs-chrs), [6](#)
- identifySignatures (mutational-signatures), [11](#)
- k3we (kmers-data), [7](#)
- k3wg (kmers-data), [7](#)
- kmerFrequency, [6](#)
- kmers (kmers-data), [7](#)
- kmers-data, [7](#)
- motif-functions, [8](#)
- motifMatrix (motif-functions), [8](#)
- mutation-distribution, [9](#)
- mutational-normalization, [10](#)
- mutational-plots, [10](#)
- mutational-signatures, [11](#)
- MutationalSignatures, [12](#)
- MutationalSignatures-class (MutationalSignatures), [12](#)
- mutationContext, [13](#)
- mutationContextH5vc (mutationContext), [13](#)
- mutationContextMutect (mutationContext), [13](#)
- mutationDistance (mutation-distribution), [9](#)
- ncbi (GRanges-converters), [4](#)
- nmfDecomposition (decomposition-signatures), [3](#)
- nmfSignatures (decomposition-signatures), [3](#)
- normalizeMotifs (mutational-normalization), [10](#)
- observed (MutationalSignatures), [12](#)
- observed, MutationalSignatures-method (MutationalSignatures), [12](#)
- pcaDecomposition (decomposition-signatures), [3](#)
- pcaSignatures (decomposition-signatures), [3](#)
- plotFittedSpectrum (signature-plots), [16](#)
- plotMutationSpectrum (signature-plots), [16](#)
- plotObservedSpectrum (signature-plots), [16](#)

plotRainfall (mutation-distribution), 9  
plotSampleMap (signature-plots), 16  
plotSamples (signature-plots), 16  
plotSignatureMap (signature-plots), 16  
plotSignatures (signature-plots), 16  
plotVariantAbundance  
    (mutational-plots), 10

readMutect, 14

samples (MutationalSignatures), 12  
samples, MutationalSignatures-method  
    (MutationalSignatures), 12

sca-data, 15  
sca\_mm (sca-data), 15  
sca\_sigs (sca-data), 15  
scaSNVRanges, 15

seqchar (GRanges-converters), 4  
show (MutationalSignatures), 12  
show, MutationalSignatures-method  
    (MutationalSignatures), 12

signature-plots, 16  
signatures (MutationalSignatures), 12  
signatures, MutationalSignatures-method  
    (MutationalSignatures), 12

signatures21 (signatures21-data), 17  
signatures21-data, 17  
sigs\_nmf (sca-data), 15  
sigs\_pca (sca-data), 15

SomaticSignatures, 18  
SomaticSignatures-package  
    (SomaticSignatures), 18

ucsc (GRanges-converters), 4

variants-utils, 19