Package 'ICAMS'

March 22, 2019

Type Package

Title In-depth Characterization and Analysis of Mutational Signatures

Version 1.0.2

Author Steve Rozen, Nanhai Jiang, Arnoud Boot

Maintainer Steve Rozen <steverozen@gmail.com>

Description A toolkit for analysis and visualization of experimentally elucidated mutational signatures -- the kind of analysis and visualization presented in Boot et al., ``In-depth characterization of the cisplatin mutational signature in human cell lines and in esophageal and liver tumors", 2018, https://genome.cshlp.org/content/28/5/654.short. This package has functions to read in variant call files and to collate the corresponding catalog of mutational spectra and to plot catalogs of mutational spectra or signatures.

License GPL-3

Encoding UTF-8

LazyData true

Language en-US

biocViews

Imports Biostrings,
BSgenome,
BSgenome.Hsapiens.1000genomes.hs37d5,
BSgenome.Hsapiens.UCSC.hg38,
data.table,
dplyr,
GenomicRanges,
graphics,
grDevices,
methods,
RColorBrewer,
stats,
stringr,
utils

Depends R (>= 3.5),

RoxygenNote 6.1.1

Suggests testthat

R topics documented:

CatalogRowOrder . . . 2
CollapseCatalog . . . 3
CreateCatalogAttribute . . . 3
FindDelMH . . . 4
GetVAF . . . 6
ICAMS . . . 6
MutectVCFFilesToCatalog . . . 9
PlotCatalog . . . 9
PlotCatalogToPdf . . . 10
ReadAndSplitMutectVCFs . . . 10
ReadAndSplitStrelkaSNSVCFs . . . 11
ReadCatalog . . . 12
ReadStrelkaIDVCFs . . . 12
revc . . . 13
StrelkaIDVCFFilesToCatalog . . . 13
StrelkaSNSVCFFilesToCatalog . . . 14
TranscriptRanges . . . 15
TransformCatalog . . . 15
VCFsToDNSCatalogs . . . 17
VCFsToIDCatalogs . . . 17
VCFsToSNSCatalogs . . . 18
WriteCatalog . . . 18

Index **20**

`CatalogRowOrder` *Standard order of row names in a catalog.*

Description

This data is designed for those who need to create their own catalogs from formats not supported by this package. The rownames denote the mutation types. For example, for SNS96 catalogs, the rowname AGAT represents a mutation from AGA > ATA.

Usage

```
catalog.row.order

catalog.row.order
```

Format

A list of character vectors indicating the standard orders of row names in catalogs.

Note

In the ID (insertion and deletion) catalog, deletion repeat size is in the range from 0 to 5+, but for plotting and end user documentation it ranges from 1 to 6+.

`CollapseCatalog` *"Collapse" a catalog.*

Description

"Collapse" a catalog. Do not use this function for signature catalogs.

Usage

```
Collapse192To96(catalog)

Collapse1536To96(catalog)

Collapse144To78(catalog)
```

Arguments

`catalog`	A catalog as defined in ICAMS.

Details

`Collapse192To96` Collapse an SNS 192 catalog to an SNS 96 catalog.

`Collapse1536To96` Collapse an SNS 1536 catalog to an SNS 96 catalog.

`Collapse144To78` Collapse a DNS 144 catalog to a DNS 78 catalog.

Value

A catalog as defined in ICAMS.

`CreateCatalogAttribute`
Create an S3 object of class "catalog"

Description

Create an S3 object of class "catalog"

Usage

```
CreateCatalogAttribute(catalog, ref.genome, region, type)
```

Arguments

`catalog`	A catalog as defined in ICAMS.
`ref.genome`	A character string acting as a genome identifier, one of "GRCh37", "hg19", "GRCh38", "hg38".
`region`	A character string acting as a region identifier, one of "genome", "exome".
`type`	A character string acting as a catalog type identifier, one of "counts", "density", "signature".

Value

An S3 object of class "catalog".

`FindDelMH` *Return the length of microhomology at a deletion.*

Description

Return the length of microhomology at a deletion.

Usage

```
FindDelMH(context, deleted.seq, pos, trace = 0)
```

Arguments

`context`	The deleted sequence plus ample surrounding sequence on each side (at least as long as `del.sequence`).
`deleted.seq`	The deleted sequence in `context`.
`pos`	The position of `del.sequence` in `context`.
`trace`	If > 0, cat various messages.

Details

This function is primarily for internal use, but we export it to document the underlying logic.

Example:

`GGCTAGTT` aligned to `GGCTAGAACTAGTT` with a deletion represented as:

```
GGCTAGAACTAGTT
GG------CTAGTT GGCTAGTT GG[CTAGAA]CTAGTT
                            ----   ----
```

Presumed repair mechanism leading to this:

```
  ....
GGCTAGAACTAGTT
CCGATCTTGATCAA

=>

  ....
GGCTAG      TT
CC      GATCAA
        ....

=>

GGCTAGTT
CCGATCAA
```

Variant-caller software can represent the same deletion in several different, but completely equivalent, ways.

```
GGC------TAGTT GGCTAGTT GGC[TAGAAC]TAGTT
                          * ---  * ---

GGCT------AGTT GGCTAGTT GGCT[AGAACT]AGTT
                          ** --  ** --

GGCTA------GTT GGCTAGTT GGCTA[GAACTA]GTT
                          *** -  *** -

GGCTAG------TT GGCTAGTT GGCTAG[AACTAG]TT
                          ****   ****
```

A deletion in a *repeat* can also be represented in several different ways. A deletion in a repeat is abstractly equivalent to microhomology that spans the entire deleted sequence. For example;

```
GACTAGCTAGTT
GACTA----GTT GACTAGTT GACTA[GCTA]GTT
                        *** -*** -
```

is really a repeat

```
GACTAG----TT GACTAGTT GACTAG[CTAG]TT
                        **** ----

GACT----AGTT GACTAGTT GACT[AGCT]AGTT
                        ** --** --
```

This function only flags this case with a -1 return; it does not figure out the repeat extent.

This function finds:

1. The maximum match of undeleted sequence to the left of the deletion that is identical to the right end of the deleted sequence, and
2. The maximum match of undeleted sequence to the right of the deletion that is identical to the left end of the deleted sequence.

The microhomology sequence is the concatenation of items (1) and (2).

Value

The length of the maximum microhomology of `del.sequence` in `context`.

GetVAF *Extract the VAFs (variant allele frequencies) from a VCF file.*

Description

Extract the VAFs (variant allele frequencies) from a VCF file.

Usage

```
GetStrelkaVAF(vcf)

GetMutectVAF(vcf)
```

Arguments

vcf	said VCF as a data.frame.

Value

A vector of VAFs, one for each row of vcf.

ICAMS *ICAMS: In-depth Characterization and Analysis of Mutational Signatures*

Description

A toolkit for analysis and visualization of experimentally elucidated mutational signatures – the kind of analysis and visualization presented in Boot et al., "In-depth characterization of the cisplatin mutational signature in human cell lines and in esophageal and liver tumors", *Genome Research*, 2018, https://genome.cshlp.org/content/28/5/654.short.

Details

ICAMS can read in variant call files (VCFs) generated by Strelka or Mutect, and collate the mutations into "catalogs" of mutational spectra. ICAMS can create and plot catalogs of mutational spectra or signatures for single nucleotide substitutions (SNS), double nucleotide substitutions (DNS), and small insertions and deletions (ID). It can also read and write these catalogs.

Catalogs and signatures

A key data type in ICAMS is a "catalog" of mutation counts, of mutation densities, or of mutational signatures.

A catalog is one of the following:

1. Matrix of mutation counts (one column per sample), representing (count-based) mutational spectra.
2. Matrix of mutation densities, i.e. mutations per occurrences of source sequences (one column per sample), representing (density-based) mutational spectra.

3. Matrix of mutational signatures, which are similar to spectra. However where spectra consist of counts or densities of mutations in each mutation class (e.g. ACA > AAA, ACA > AGA, ACA > ATA, ACC > AAC, ...), signatures consist of the proportions of mutations in each class (with all the proportions summing to 1).#' A mutational signature can be based on either:
   (a) mutation counts (a "count-based mutational signature"), or
   (b) mutation densities (a "density-based mutational signature").

If you need to create a catalog from a source other than this package (i.e. other than with `ReadCatalog` or `StrelkaSNSVCFFilesToCatalog`, `MutectVCFFilesToCatalog`, etc.), then you must ensure that the rows are in the expected order and have the expected rownames. See `CatalogRowOrder` for the expected rownames and order.

Creating catalogs from variant call files (VCF files)

1. `StrelkaSNSVCFFilesToCatalog` creates 3 SNS catalogs (96, 192, 1536) and 3 DNS catalogs (78, 136, 144) from the Strelka SNS VCFs.
2. `StrelkaIDVCFFilesToCatalog` creates ID (indel) catalog from the Strelka ID VCFs.
3. `MutectVCFFilesToCatalog` creates 3 SNS catalogs (96, 192, 1536), 3 DNS catalogs (78, 136, 144) and ID (indel) catalog from the Mutect VCFs.

The `genome` argument

Many functions take the argument `genome`. This can be either

1. A variable from the Bioconductor BSgenome package that contains a particular reference genome, for example `BSgenome.Hsapiens.1000genomes.hs37d5`. `BSgenome::available.genomes()` returns the available genomes.
2. The strings `"hg38"` or `"GRCh38"` are shorthand for `BSgenome.Hsapiens.UCSC.hg38`, and the strings `"hg19"` or `"GRCh37"` are shorthand for `BSgenome.Hsapiens.1000genomes.hs37d5`.

The Bioconductor `BSgenome` package

This package will be installed automatically if `ICAMS` is installed with `devtools::install_local` or with `devtools::install_github`. Otherwise you must manually install `BSgenome` and the necessary genomes, e.g.

```
BSgenome.Hsapiens.1000genomes.hs37d5.
```

See instructions at

```
https://bioconductor.org/packages/release/bioc/html/BSgenome.html.
```

Genomes other than the two human genomes mentioned above must be installed manually.

Use `available.genomes` to get the list of available genomes.

Plotting catalogs

The `PlotCatalog` functions plot mutational spectra for one sample or plot one mutational signature.

The `PlotCatalogToPdf` functions plot catalogs of mutational spectra or of mutational signatures to a PDF file.

Writing catalogs

The `WriteCatalog` functions write a catalog of mutational spectra or of mutational signatures to a file.

Reading catalogs

The `ReadCatalog` functions read a file that contains a catalog of mutational spectra or of signatures in standardized format.

Transforming catalogs

The `TransformCatalog` function transforms catalogs of mutational spectra or signatures to account for differing abundances of the source sequence of the mutations in the genome.

For example, mutations from ACG are much rarer in the human genome than mutations from ACC simply because CG dinucleotides are rare in the genome. Consequently, there are two possible representations of mutational spectra or signatures. One representation is based on mutation counts as observed in a given genome, and this approach is widely used, as, for example, at https://cancer.sanger.ac.uk/cosmic/signatures, which presents signatures based on observed mutation counts in the human genome. We call these "count-based spectra" or "count-based signatures".

Alternatively, mutational spectra or signatures can be represented as mutations per source sequence, for example the number of ACT > AGT mutations occurring at all ACT 3-mers in a genome. We call these "density-based spectra" or "density-based signatures".

This function can also transform spectra based on observed genome-wide counts to "density"-based catalogs. In density-based catalogs mutations are expressed as mutations per source sequences. For example, a density-based catalog represents the proportion of ACCs mutated to ATCs, the proportion of ACGs mutated to ATGs, etc. This is different from count-based catalogs, which contain the number of ACC > ATC mutations, the number of ACG > ATG mutations, etc.

This function can also transform observed-count based spectra or signatures from genome to exome based counts, or between different species (since the abundances of source sequences vary between genome and exome and between species).

Collapsing catalogs

The `CollapseCatalog` functions

1. take a mutational spectrum or signature catalog that is based on a fined-grained set of features (for example, single-nucleotide substitutions in the context of the preceding and following 2 bases), and
2. collapse it to a catalog based on a coarser-grained set of features (for example, single-nucleotide substitutions in the context of the immediately preceding and following bases).

Data

1. `CatalogRowOrder` Standard order of rownames in a catalog. The rownames of encode the type of each mutation. The rownames denote the mutation types. For example, for SNS96 catalogs, the rowname AGAT represents a mutation from AGA > ATA.
2. `TranscriptRanges` Transcript ranges and strand information for a particular reference genome.

`MutectVCFFilesToCatalog` *Create SNS and DNS catalogs from Mutect VCF files*

Description

Create 3 SNS catalogs (96, 192, 1536) and 3 DNS catalogs (78, 136, 144) from the Mutect VCFs specified by vector.of.file.paths

Usage

```
MutectVCFFilesToCatalog(vector.of.file.paths, genome, trans.ranges, region)
```

Arguments

`vector.of.file.paths`	Character vector of file paths to the Mutect VCF files.
`genome`	A genome argument as described in `ICAMS`.
`trans.ranges`	A data.table which contains transcript range and strand information.
`region`	A character string acting as a region identifier, one of "genome", "exome".

Details

This function calls `VCFsToSNSCatalogs`, `VCFsToDNSCatalogs` and `VCFsToIDCatalogs`

Value

A list of S3 objects with class "catalog". See `CreateCatalogAttribute` for more details. There are 3 SNS catalogs (one each for 96, 192, and 1536), 3 DNS catalogs (one each for 78, 136, and 144) and an ID (indel) catalog.

Note

SNS 192 and DNS 144 catalogs include only mutations in transcribed regions.

`PlotCatalog` *Plot* **one** *spectrum or signature.*

Description

Plot the spectrum of **one** sample or plot **one** signature.

Usage

```
PlotCatalog(catalog, strandbias = FALSE, ...)
```

Arguments

catalog	An S3 object with class "catalog". See CreateCatalogAttribute for more details.
strandbias	If TRUE, plot strand bias graph for SNS192 or DNS144 catalog. Leave out this parameter if you don't intend to plot strand bias graph.
...	Arguments to be passed to methods.

Value

invisible(TRUE)

PlotCatalogToPdf *Plot catalogs to a PDF file.*

Description

Plot catalogs to a PDF file.

Usage

```
PlotCatalogToPdf(catalog, filename, strandbias = FALSE, ...)
```

Arguments

catalog	An S3 object with class "catalog". See CreateCatalogAttribute for more details.
filename	The name of the PDF file to be produced.
strandbias	If TRUE, plot strand bias graph for SNS192 or DNS144 catalog. Leave out this parameter if you don't intend to plot strand bias graph.
...	Arguments to be passed to methods.

Value

invisible(TRUE)

ReadAndSplitMutectVCFs *Read and split Mutect VCF files.*

Description

Read and split Mutect VCF files.

Usage

```
ReadAndSplitMutectVCFs(vector.of.file.paths)
```

Arguments

`vector.of.file.paths`
: Character vector of file paths to the Mutect VCF files.

Value

A list with 3 in-memory VCFs and two left-over VCF-like data frames with rows that were not incorporated into the first 3 VCFs, as follows:

1. `SNS` VCF with only single nucleotide substitutions.
2. `DNS` VCF with only doublet nucleotide substitutions as called by Mutect.
3. `ID` VCF with only small insertions and deletions.
4. `other.subs` VCF like data.frame with rows for coordinate substitutions involving 3 or more nucleotides, e.g. ACT > TGA or AACT > GGTA.
5. `multiple.alternative.alleles` VCF like data.frame with rows for variants with multiple alternative alleles, for example ACT mutated to both AGT and ACT at the same position.

See Also

`MutectVCFFilesToCatalog`

`ReadAndSplitStrelkaSNSVCFs` *Read and split Strelka SNS VCF files.*

Description

Read and split Strelka SNS VCF files.

Usage

```
ReadAndSplitStrelkaSNSVCFs(vector.of.file.paths)
```

Arguments

`vector.of.file.paths`
: Character vector of file paths to the Strelka SNS VCF files.

Value

A list of 3 in-memory objects as follows:

1. `SNS.vcfs` List of data.frames of pure SNS mutations – no DNS or 3+BS mutations.
2. `DNS.vcfs` List of data.frames of pure DNS mutations – no SNS or 3+BS mutations.
3. `ThreePlus` List of data.tables with the key CHROM, LOW.POS, HIGH.POS. containing rows that that in the input that did not represent SNSs or DNSs.

See Also

`StrelkaSNSVCFFilesToCatalog`

`ReadCatalog` *Read catalog.*

Description

Read a catalog in standardized format from path.

Usage

```
ReadCatalog(path, ref.genome, region, type, strict = TRUE)
```

Arguments

`path`	Path to a catalog on disk in the standardized format.
`ref.genome`	A character string acting as a genome identifier, one of "GRCh37", "hg19", "GRCh38", "hg38".
`region`	A character string acting as a region identifier, one of "genome", "exome".
`type`	A character string acting as a catalog type identifier, one of "counts", "density", "signature".
`strict`	If TRUE, then stop if additional checks on the input fail.

Details

See also `WriteCatalog`

Value

An S3 object with class "catalog". See `CreateCatalogAttribute` for more details.

Note

In the ID (insertion and deletion) catalog, deletion repeat size ranges from 0 to 5+, but for plotting and end user documentation it ranges from 1 to 6+.

`ReadStrelkaIDVCFs` *Read Strelka ID (insertion and deletion) VCF files.*

Description

Read Strelka ID (insertion and deletion) VCF files.

Usage

```
ReadStrelkaIDVCFs(vector.of.file.paths)
```

Arguments

`vector.of.file.paths`	Character vector of file paths to the VCF files.

Value

A list of vcfs from vector.of.file.paths.

Note

In the ID (insertion and deletion) catalog, deletion repeat size ranges from 0 to 5+, but for plotting and end user documentation it ranges from 1 to 6+.

See Also

StrelkaIDVCFFilesToCatalog

---

revc *Reverse complement every string in* string.vec.

Description

Reverse complement every string in string.vec.

Usage

```
revc(string.vec)
```

Arguments

string.vec	a vector of type character.

Value

A vector of type characters with the reverse complement of every string in string.vec.

StrelkaIDVCFFilesToCatalog *Create ID (indel) catalog from Strelka ID VCF files*

Description

Create ID (indel) catalog from the Strelka ID VCFs specified by vector.of.file.paths

Usage

```
StrelkaIDVCFFilesToCatalog(vector.of.file.paths, genome, region)
```

Arguments

vector.of.file.paths	Character vector of file paths to the Strelka ID VCF files.
genome	A genome argument as described in ICAMS.
region	A character string acting as a region identifier, one of "genome", "exome".

Details

This function calls VCFsToIDCatalogs

Value

An S3 object containing an ID (indel) catalog with class "catalog". See CreateCatalogAttribute for more details.

Note

In the ID (insertion and deletion) catalog, deletion repeat size ranges from 0 to 5+, but for plotting and end user documentation it ranges from 1 to 6+.

StrelkaSNSVCFFilesToCatalog

Create SNS and DNS catalogs from Strelka SNS VCF files.

Description

Create 3 SNS catalogs (96, 192, 1536) and 3 DNS catalogs (78, 136, 144) from the Strelka SNS VCFs specified by vector.of.file.paths

Usage

```
StrelkaSNSVCFFilesToCatalog(vector.of.file.paths, genome, trans.ranges,
  region)
```

Arguments

vector.of.file.paths	Character vector of file paths to the Strelka SNS VCF files.
genome	A reference genome as described in ICAMS.
trans.ranges	A data.table which contains transcript range and strand information.
region	A character string acting as a region identifier, one of "genome", "exome".

Details

This function calls VCFsToSNSCatalogs and VCFsToDNSCatalogs.

Value

A list of S3 objects with class "catalog". See CreateCatalogAttribute for more details. There are 3 SNS catalogs (one each for 96, 192, and 1536) and 3 DNS catalogs (one each for 78, 136, and 144)

Note

SNS 192 and DNS 144 catalog only contains mutations in transcribed regions.

TranscriptRanges *Transcript ranges data*

Description

Transcript ranges and strand information for a particular reference genome.

Usage

```
trans.ranges.GRCh37

trans.ranges.GRCh38
```

Format

A data.table which contains transcript range and strand information for a particular reference genome.

Details

`trans.ranges.GRCh37` A data.table which contains transcript range and strand information for **Human** GRCh37. It is derived from a raw **GFF3** format file, from which only the following four gene types are kept to facilitate transcriptional strand bias analysis: protein_coding, retained_intron, processed_transcript and nonsense_mediated_decay. It contains chromosome name, start, end position, strand information and gene name and is keyed by chrom, chromStart, and chromEnd. It can be used in function `StrelkaSNSVCFFilesToCatalog`.

`trans.ranges.GRCh38` A data.table which contains transcript range and strand information for **Human** GRCh38. It is derived from a raw **GFF3** format file, from which only the following four gene types are kept to facilitate transcriptional strand bias analysis: protein_coding, retained_intron, processed_transcript and nonsense_mediated_decay. It contains chromosome name, start, end position, strand information and gene name and is keyed by chrom, chromStart, and chromEnd. It can be used in function `StrelkaSNSVCFFilesToCatalog`.

TransformCatalog *Transform between count and density catalogs and signatures and between different source-sequence abundances.*

Description

Transform between count and density catalogs and signatures and between different source-sequence abundances.

Usage

```
TransformCatalog(catalog, which.n, source.type,
  target.type = source.type, source.abundance = NULL,
  target.abundance = NULL)
```

Arguments

`catalog` An SNS or DNS catalog as described in ICAMS; must **not** be an ID (indel) catalog.

`which.n` The length of the source sequences, one of 2:5.

`source.type` A character specifying type of the input catalog, one of `"counts"`, `"signature"` or `"density"`.

`target.type` A character specifying type of the output catalog, with the same possible values as `source.type`.

`source.abundance`
One of

1. `NULL`, which indicates that the source type is a catalog of density-based spectra or signatures.
2. A numeric vector with one element for each source sequence for the mutation types in `catalog`, where by source we mean, for example, ACT as the source sequence for ACT > AGT mutations.
3. A string specifying such a vector, one of `"GRCh37.genome"`, `"GRCh37.exome"`, `"GRCh38.genome"`, or `"GRCh38.exome"`.

For the last two options, the numerical vector contains the abundance upon which the counts or proportions in `catalog` are based.

`target.abundance`
Same possibilities as `source.abundance`.

Details

Only certain parings of type and abundance are legal, as follows:

1. The type `"density"` must always be associated with a `NULL` abundance.
2. The type `"signature"` is allowed to be associated with a `NULL` abundance. A `NULL` abundance indicates that the signature is a "density-based" signature (see ICAMS).
3. The type `"counts"` must **not** be associated with the `NULL` abundance.

Only the following transformations are legal:

1. `counts -> counts`
2. `counts -> density`
3. `counts -> signature`
4. `density -> counts` (in which case the semantics are to infer the genome-wide or exome wide counts based on the densities.)
5. `density -> signature`
6. `signature -> signature`

Value

A catalog as defined in ICAMS

VCFsToDNSCatalogs *Create DNS catalogs from VCFs*

Description

Create a list of 3 catalogs (one each for DNS78, DNS144 and DNS136) out of the contents in list.of.DNS.vcfs. The VCFs must not contain any type of mutation other then DNSs.

Usage

```
VCFsToDNSCatalogs(list.of.DNS.vcfs, genome, trans.ranges, region)
```

Arguments

list.of.DNS.vcfs	List of in-memory data frames of pure DNS mutations – no SNS or 3+BS mutations. The list names will be the sample ids in the output catalog.
genome	A genome argument as described in ICAMS.
trans.ranges	A data frame containing transcript ranges.
region	A character string acting as a region identifier, one of "genome", "exome".

Value

A list of S3 objects with class "catalog", one each for DNS 78, 144, 136: catDNS78, catDNS144, catDNS136. See CreateCatalogAttribute for more details.

Note

DNS 144 catalog only contains mutations in transcribed regions.

VCFsToIDCatalogs *Create ID (insertion and deletion) catalog from ID VCFs*

Description

Create ID (insertion and deletion) catalog from ID VCFs

Usage

```
VCFsToIDCatalogs(list.of.vcfs, genome, region)
```

Arguments

list.of.vcfs	List of in-memory VCFs. The list names will be the sample ids in the output catalog.
genome	A genome argument as described in ICAMS.
region	A character string acting as a region identifier, one of "genome", "exome".

Value

An S3 object containing an ID (indel) catalog with class "catalog". See CreateCatalogAttribute for more details.

`VCFsToSNSCatalogs` *Create SNS catalogs from SNS VCFs*

Description

Create a list of 3 catalogs (one each for 96, 192, 1536) out of the contents in list.of.SNS.vcfs. The SNS VCFs must not contain DNSs, indels, or other types of mutations.

Usage

```
VCFsToSNSCatalogs(list.of.SNS.vcfs, genome, trans.ranges, region)
```

Arguments

`list.of.SNS.vcfs`	List of in-memory data frames of pure SNS mutations – no DNS or 3+BS mutations. The list names will be the sample ids in the output catalog.
`genome`	A genome argument as described in ICAMS.
`trans.ranges`	A data frame containing transcript ranges.
`region`	A character string acting as a region identifier, one of "genome", "exome".

Value

A list of S3 objects with class "catalog", one each for SNS 96, 192, 1536: catSNS96 catSNS192 catSNS1536. See CreateCatalogAttribute for more details.

Note

SNS 192 catalog only contains mutations in transcribed regions.

`WriteCatalog` *Write catalog*

Description

Write a catalog to a file on disk.

Usage

```
WriteCatalog(catalog, path, strict = TRUE)
```

Arguments

catalog	An S3 object with class "catalog". See CreateCatalogAttribute for more details.
path	The path of the file to be written on disk.
strict	If TRUE, then fail if additional checks on the input fail.

Details

See also ReadCatalog

Note

In the ID (insertion and deletion) catalog, deletion repeat size ranges from 0 to 5+, but for plotting and end user documentation it ranges from 1 to 6+.

Index

∗Topic **datasets**
CatalogRowOrder, 2
TranscriptRanges, 15

available.genomes, *7*

BSgenome, *7*

catalog.row.order *(*CatalogRowOrder*)*, 2
CatalogRowOrder, 2, *7, 8*
Collapse144To78 *(*CollapseCatalog*)*, 3
Collapse1536To96 *(*CollapseCatalog*)*, 3
Collapse192To96 *(*CollapseCatalog*)*, 3
CollapseCatalog, 3, *8*
CreateCatalogAttribute, 3, *9, 10*, *12*, *14*, *17–19*

FindDelMH, 4

GetMutectVAF *(*GetVAF*)*, 6
GetStrelkaVAF *(*GetVAF*)*, 6
GetVAF, 6

ICAMS, *3*, 6, *7*, *9*, *13, 14*, *16–18*
ICAMS-package *(*ICAMS*)*, 6

MutectVCFFilesToCatalog, *7*, 9, *11*

PlotCatalog, *7*, 9
PlotCatalogToPdf, *7*, 10

ReadAndSplitMutectVCFs, 10
ReadAndSplitStrelkaSNSVCFs, 11
ReadCatalog, *7, 8*, 12, *19*
ReadStrelkaIDVCFs, 12
revc, 13

StrelkaIDVCFFilesToCatalog, *7*, *13*, 13
StrelkaSNSVCFFilesToCatalog, *7*, *11*, 14, *15*

trans.ranges.GRCh37 *(*TranscriptRanges*)*, 15
trans.ranges.GRCh38 *(*TranscriptRanges*)*, 15
TranscriptRanges, *8*, 15
TransformCatalog, *8*, 15

VCFsToDNSCatalogs, *9*, *14*, 17
VCFsToIDCatalogs, *9*, *14*, 17
VCFsToSNSCatalogs, *9*, *14*, 18

WriteCatalog, *8*, *12*, 18