Package 'ICAMS'

March 15, 2019

Type Package

Title In-depth Characterization and Analysis of Mutational Signatures

Version 0.0.0.9009

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 and plot the mutational spectra.

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,
RCurl,
stats,
stringr,
utils

Depends R (>= 3.5),

RoxygenNote 6.1.1

Suggests knitr,
rmarkdown,
testthat

R topics documented:

`CatalogRowOrder` *Canonical order of row names in a catalog*

Description

Canonical order of row names in a catalog

Usage

```
catalog.row.order
```

Format

A list which contains string of characters indicating the canonical order of row names in a catalog.

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+.

CollapseCatalog *Collapse catalog functions*

Description

Collapse a catalog matrix

Usage

```
Collapse192To96(catalog)

Collapse1536To96(catalog)

Collapse144To78(catalog)
```

Arguments

catalog	A catalog as defined in ICAMS

Details

`Collapse192To96` Collapse a SNS 192 catalog matrix to a SNS 96 catalog matrix.

`Collapse1536To96` Collapse a SNS 1536 catalog matrix to a SNS 96 catalog matrix.

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

Value

A catalog as defined in ICAMS

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 so that the logic behind it will be documented for users.

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
```

The deletion caller 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

```
TODO(Steve): add check in code
GACTAG----TT GACTAGTT GACTAG[CTAG]TT
                        **** ----

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

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

In the implementation, the function finds:

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

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),
2. Matrix of mutation densities, i.e. mutations per occurrences of source sequences (one column per sample), or
3. Mutational signatures (proportions of different mutations, summing to 1, one column per signature).

ICAMS can read in variant call files (VCFs) generated by Strelka or Mutect, and collate the mutations into "catalogs" of mutational spectra. ICAMS can plot the catalogs of mutational spectra and signatures.

ICAMS can build and plot catalogs of mutational spectra for single nucleotide substitutions (SNS), double nucleotide substitutions (DNS), and small insertions and deletions (ID). It can also read and write these catalogs.

Creating catalogs from variant call files (VCF files)

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

Plotting catalogs

Functions for plotting catalogs of mutational spectra or of mutational signatures to a PDF file. Mutational *signatures* are similar to spectra, but where spectra consist of counts 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). `PlotCatalogToPdf`

Writing catalogs

Functions for writing a catalog of mutational spectra or of mutational signatures to a file on disk. `WriteCatalog`

Reading catalogs

Functions for reading files that contain catalogs of mutational spectra or of signatures in standardized format. ReadCatalog

Transforming catalogs

There is a function to transform 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. 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 opposed to count-based catalogs, which contain the number of ACC-to-ATC mutations, the number of ACG-to-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). TransformCatalog

Collapsing catalogs

Functions for collapsing a mutational spectrum or signature catalog based on a fined-grained set of features (for example, single-nucleotide substitutions in the context of the preceding and following 2 bases) 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). CollapseCatalog

Exported data

1. CatalogRowOrder Canonical order of row names in a catalog.
2. TranscriptRanges Transcript ranges and strand information for a particular organism.

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)
```

Arguments

`vector.of.file.paths`
: A vector containing the paths of the Mutect VCF files.

genome	A particular reference genome(without quotation marks). Use available.genomes to get the list of "BSgenome data packages" currently available. There are 2 types of predefined reference genome which are incorporated in this function. User can invoke a predefined human GRCh38/hg38 BSgenome data package by typing genome = "GRCh38" or genome = "hg38". User can invoke a predefined human GRCh37/hg19 BSgenome data package by typing genome = "GRCh37" or genome = "hg19".
trans.ranges	A data.table which contains transcript range and strand information.

Details

This function calls VCFsToSNSCatalogs, VCFsToDNSCatalogs and VCFsToIDCatalogs

Value

A list of 3 SNS catalogs (one each for 96, 192, and 1536) , 3 DNS catalogs (one each for 78, 136, and 144) and ID catalog.

Note

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

PlotCatalog *Plot catalog functions*

Description

Plot the catalog of one sample which has mutations

Usage

```
PlotCatSNS96(catalog, type, id = colnames(catalog), cex = 0.8,
  grid = TRUE, upper = TRUE, xlabels = TRUE)

PlotCatSNS192(catalog, type, id = colnames(catalog), cex = 0.8)

PlotSNSClassStrandBias(catalog, type, id = colnames(catalog), cex = 1)

PlotCatSNS1536(catalog, type, id = colnames(catalog))

PlotCatDNS78(catalog, type, id = colnames(catalog))

PlotDNSClassStrandBias(catalog, type, id = colnames(catalog), cex = 1)

PlotCatDNS136(catalog, type, id = colnames(catalog))

PlotCatID(catalog, type, id = colnames(catalog))
```

Arguments

catalog	A catalog as described in ICAMS. The input catalog must be in **matrix** format, you may use data.matrix to convert a data frame to a numeric matrix. This catalog matrix must have rownames to facilitate sorting in the plotting functions. You many use CatalogRowOrder to give row names to your catalog matrix.
type	A character specifying type of the input catalog, one of "counts", "signature" or "density". If type = "counts", the graph will plot the occurrences of the mutation types in the sample. If type = "signature", the graph will plot mutation signatures of the sample. If type = "density", the graph will plot the rates of mutations per million nucleotides for each mutation type. (Please take note there is no "signature" type for PlotCatDNS136 function, no "density" type for PlotCatID function and the option of type = "density" is not implemented for function PlotCatSNS192, PlotSNSClassStrandBias and PlotDNSClassStrandBias at the current stage.)
id	The identifier of the sample which has mutations.
cex	A numerical value giving the amount by which mutation class labels, mutation counts(if it exists), y axis and its labels, x axis labels and its annotations(if it exists) sample name and legend(if it exists) should be magnified relative to the default.
grid	If TRUE, draw grid lines in the graph.
upper	If TRUE, draw horizontal lines and the names of major mutation class on top of graph.
xlabels	If TRUE, draw x axis labels.

Details

PlotCatSNS96 Plot the SNS 96 mutation catalog of one sample.

PlotCatSNS192 Plot the SNS 192 mutation catalog of one sample.

PlotSNSClassStrandBias Plot the transcription strand bias graph of 6 SNS mutation types ("C>A", "C>G", "C>T", "T>A", "T>C", "T>G") in one sample.

PlotCatSNS1536 Plot the pentanucleotide sequence contexts for one sample, normalized by pentanucleotide occurrence in the genome. The mutation types are in six-letters like CATTAT, first 2-letters CA refers to (-2, -1) position, third letter T refers to the base which has mutation, next second 2-letters TA refers to (+1, +2) position, last letter T refers to the base after mutation.

PlotCatDNS78 Plot the DNS 78 mutation catalog of one sample.

PlotDNSClassStrandBias Plot the transcription strand bias graph of 10 major DNS mutation types ("AC>NN", "AT>NN", "CC>NN", "CG>NN", "CT>NN", "GC>NN", "TA>NN", "TC>NN", "TG>NN", "TT>NN") in one sample.

PlotCatDNS136 Plot the tetranucleotide sequence context of 10 major DNS mutation types ("AC>NN", "AT>NN", "CC>NN", "CG>NN", "CT>NN", "GC>NN", "TA>NN", "TC>NN", "TG>NN", "TT>NN") for one sample.

PlotCatID Plot the insertion and deletion catalog of one sample. (Please take note that deletion repeat size ranges from 0 to 5+ in the catalog, but for plotting and end user documentation it ranges from 1 to 6+.)

Value

invisible(TRUE)

PlotCatalogToPdf *Plot catalog to pdf functions*

Description

Plot mutation catalogs of various samples to a PDF file

Usage

```
PlotCatSNS96ToPdf(catalog, name, type, id = colnames(catalog),
  grid = TRUE, upper = TRUE, xlabels = TRUE)

PlotCatSNS192ToPdf(catalog, name, id = colnames(catalog),
  type = "counts", cex = 0.8)

PlotSNSClassStrandBiasToPdf(catalog, name, type, id = colnames(catalog),
  cex = 1)

PlotCatSNS1536ToPdf(catalog, name, type, id = colnames(catalog))

PlotCatDNS78ToPdf(catalog, name, type, id = colnames(catalog))

PlotDNSClassStrandBiasToPdf(catalog, name, type, id = colnames(catalog),
  cex = 1)

PlotCatDNS136ToPdf(catalog, name, type, id = colnames(catalog))

PlotCatIDToPdf(catalog, name, type, id = colnames(catalog))
```

Arguments

catalog	A catalog as described in ICAMS. The input catalog must be in **matrix** format, you may use data.matrix to convert a data frame to a numeric matrix. This catalog matrix must have rownames to facilitate sorting in the plotting functions. You many use CatalogRowOrder to give row names to your catalog matrix.
name	The name of the PDF file to be produced.
type	A character specifying type of the input catalog, one of "counts", "signature" or "density". If type = "counts", the graph will plot the occurrences of the mutation types in the sample. If type = "signature", the graph will plot mutation signatures of the sample. If type = "density", the graph will plot the rates of mutations per million nucleotides for each mutation type. (Please take note there is no "signature" type for PlotCatDNS136ToPdf function, no "density" type for PlotCatIDtoPdf function and the option of type = "density" is not implemented for function PlotCatSNS192ToPdf, PlotSNSClassStrandBiasToPdf and PlotDNSClassStrandBiasToPdf at the current stage.)
id	A vector containing the identifiers of the samples in catalog.
grid	If TRUE, draw grid lines in the graph.
upper	If TRUE, draw horizontal lines and the names of major mutation class on top of graph.

xlabels	If TRUE, draw x axis labels.
cex	A numerical value giving the amount by which mutation class labels, y axis labels, sample name and legend (if it exists) should be magnified relative to the default.

Details

PlotCatSNS96ToPdf Plot the SNS 96 mutation catalog of various samples to a PDF file.

PlotCatSNS192ToPdf Plot the SNS 192 mutation catalog of various samples to a PDF file.

PlotSNSClassStrandBiasToPdf Plot the transcription strand bias graph of 6 SNS mutation types ("C>A", "C>G", "C>T", "T>A", "T>C", "T>G") of various samples to a PDF file.

PlotCatSNS1536ToPdf Plot the 1536 mutation catalog of >= 1 samples to a PDF file. The mutation types are in six-letters like CATTAT, first 2-letters CA refers to (-2, -1) position, third letter T refers to the base which has mutation, next second 2-letters TA refers to (+1, +2) position, last letter T refers to the base after mutation.

PlotCatDNS78ToPdf Plot the DNS 78 mutation catalog of various samples to a PDF file.

PlotDNSClassStrandBiasToPdf Plot the transcription strand bias graph of 10 major DNS mutation types ("AC>NN", "AT>NN", "CC>NN", "CG>NN", "CT>NN", "GC>NN", "TA>NN", "TC>NN", "TG>NN", "TT>NN") of various samples to a PDF file.

PlotCatDNS136ToPdf Plot the tetranucleotide sequence contexts of 10 major DNS mutation types ("AC>NN", "AT>NN", "CC>NN", "CG>NN", "CT>NN", "GC>NN", "TA>NN", "TC>NN", "TG>NN", "TT>NN") of various samples to a PDF file.

PlotCatIDToPdf Plot the insertion and deletion catalog of various samples to a PDF file. (Please take note that deletion repeat size ranges from 0 to 5+ in the catalog, but for plotting and end user documentation it ranges from 1 to 6+.)

Value

invisible(TRUE)

ReadAndSplitMutectVCFs

Read and split Mutect VCF files from paths

Description

Read and split Mutect VCF files from paths

Usage

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

Arguments

vector.of.file.paths	A vector containing the paths of the 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 from paths*

Description

Read and split Strelka SNS VCF files from paths

Usage

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

Arguments

`vector.of.file.paths`
: A vector containing the paths of the VCF files.

Value

A list of 3 in-memory objects with the elements: SNS.vcfs: List of Data frames of pure SNS mutations – no DNS or 3+BS mutations DNS.vcfs: List of Data frames of pure DNS mutations – no SNS or 3+BS mutations ThreePlus: List of Data tables with the key CHROM, LOW.POS, HIGH.POS and additional information (reference sequence, alternative sequence, context, etc.) Additional information not fully implemented at this point because of limited immediate biological interest.

See Also

`StrelkaSNSVCFFilesToCatalog`

ReadCatalog *Read Catalog Functions*

Description

Read a catalog in standardized format from path

Usage

```
ReadCatSNS96(path, strict = TRUE)

ReadCatSNS192(path, strict = TRUE)

ReadCatSNS1536(path, strict = TRUE)

ReadCatDNS78(path, strict = TRUE)

ReadCatDNS144(path, strict = TRUE)

ReadCatDNS136(path, strict = TRUE)

ReadCatID(path, strict = TRUE)
```

Arguments

path	Path to a catalog on disk in the standardized format.
strict	If TRUE, do additional checks on the input, and stop if the checks fail.

Details

ReadCatSNS96 Read a 96 SNS catalog from path

ReadCatSNS192 Read a 192 SNS catalog from path

ReadCatSNS1536 Read a 1536 SNS catalog from path

ReadCatDNS78 Read a 78 DNS catalog from path

ReadCatDNS144 Read a 144 DNS catalog from path

ReadCatDNS136 Read a 136 DNS catalog from path

ReadCatID Read a ID (insertion/deletion) catalog from path. (Please take note that deletion repeat size ranges from 0 to 5+ in the catalog, but for plotting and end user documentation it ranges from 1 to 6+.)

See also WriteCatalog

Value

A catalog in canonical in-memory format.

ReadStrelkaIDVCFs *Read Strelka ID (insertion and deletion) VCF files from paths*

Description

Read Strelka ID (insertion and deletion) VCF files from paths

Usage

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

Arguments

vector.of.file.paths	A vector containing the paths of 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)
```

Arguments

`vector.of.file.paths`	A vector containing the paths of the Strelka ID VCF files.
`genome`	A particular reference genome(without quotation marks). Use available.genomes to get the list of "BSgenome data packages" currently available. There are 2 types of predefined reference genome which are incorporated in this function. User can invoke a predefined human GRCh38/hg38 BSgenome data package by typing `genome = "GRCh38"` or `genome = "hg38"`. User can invoke a predefined human GRCh37/hg19 BSgenome data package by typing `genome = "GRCh37"` or `genome = "hg19"`.

Details

This function calls `VCFsToIDCatalogs`

Value

An ID (indel) catalog

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)
```

Arguments

vector.of.file.paths	A vector containing the paths of the Strelka SNS VCF files.
genome	A particular reference genome(without quotation marks). Use available.genomes to get the list of "BSgenome data packages" currently available. There are 2 types of predefined reference genome which are incorporated in this function. User can invoke a predefined human GRCh38/hg38 BSgenome data package by typing genome = "GRCh38" or genome = "hg38". User can invoke a predefined human GRCh37/hg19 BSgenome data package by typing genome = "GRCh37" or genome = "hg19".
trans.ranges	A data.table which contains transcript range and strand information.

Details

This function calls VCFsToSNSCatalogs and VCFsToDNSCatalogs

Value

A list of 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 organism

Usage

```
trans.ranges.GRCh37

trans.ranges.GRCh38
```

Format

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

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. The input catalog can **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	Either NULL or a numeric vector with one element for each source sequence for the mutation types in catalog or a string specifying such a vector, one of "GRCh37.genome", "GRCh37.exome", "GRCh38.genome", or "GRCh38.exome". This is the abundance upon which the counts, densities, or proportions in catalog are based. For example, for SNS in trinucleotide context, e.g. ACT > AGT or TAC > TTC, the source sequences are ACT and TAC.
target.abundance	Same possibilities as source.abundance.

Details

Only certain transformations are legal.

1. The type "density" must always be associated with a NULL abundance.
2. The type "signature" can be associated with a NULL abundance.
3. The type "counts" must **not** be associated with the NULL abundance.
4. Otherwise, the following are legal:

(a) `counts -> counts`
(b) `counts -> density`
(c) `counts -> signature`
(d) `density -> counts` (in which case the semantics are to infer the genome-wide or exome wide counts based on the densities.)
(e) `density -> signature`
(f) `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)
```

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 particular reference genome(without quotation marks). Use available.genomes to get the list of "BSgenome data packages" currently available. There are 2 types of predefined reference genome which are incorporated in this function. User can invoke a predefined human GRCh38/hg38 BSgenome data package by typing `genome = "GRCh38"` or `genome = "hg38"`. User can invoke a predefined human GRCh37/hg19 BSgenome data package by typing `genome = "GRCh37"` or `genome = "hg19"`.
`trans.ranges`	A data frame containing transcript ranges.

Value

A list of 3 DNS catalogs, one each for 78, 144, 136: catDNS78 catDNS144 catDNS136

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)
```

Arguments

list.of.vcfs	List of in-memory VCFs. The list names will be the sample ids in the output catalog.
genome	A particular reference genome(without quotation marks). Use available.genomes to get the list of "BSgenome data packages" currently available. There are 2 types of predefined reference genome which are incorporated in this function. User can invoke a predefined human GRCh38/hg38 BSgenome data package by typing genome = "GRCh38" or genome = "hg38". User can invoke a predefined human GRCh37/hg19 BSgenome data package by typing genome = "GRCh37" or genome = "hg19".

Value

An ID (indel) catalog

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)
```

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 particular reference genome(without quotation marks). Use available.genomes to get the list of "BSgenome data packages" currently available. There are 2 types of predefined reference genome which are incorporated in this function. User can invoke a predefined human GRCh38/hg38 BSgenome data package by typing genome = "GRCh38" or genome = "hg38". User can invoke a predefined human GRCh37/hg19 BSgenome data package by typing genome = "GRCh37" or genome = "hg19".
trans.ranges	A data frame containing transcript ranges.

Value

A list of 3 SNS catalogs, one each for 96, 192, 1536: catSNS96 catSNS192 catSNS1536

Note

SNS 192 catalog only contains mutations in transcribed regions.

`WriteCatalog` *Write Catalog Functions*

Description

Write a mutation catalog to a file on disk

Usage

```
WriteCatSNS96(ct, path, strict = TRUE)

WriteCatSNS192(ct, path, strict = TRUE)

WriteCatSNS1536(ct, path, strict = TRUE)

WriteCatDNS78(ct, path, strict = TRUE)

WriteCatDNS144(ct, path, strict = TRUE)

WriteCatDNS136(ct, path, strict = TRUE)

WriteCatID(ct, path, strict = TRUE)
```

Arguments

`ct`	A catalog as defined in `ICAMS`.
`path`	The path of the file to be written on disk.
`strict`	If TRUE, do additional checks on the input, and stop if the checks fail.

Details

`WriteCatSNS96` Write a SNS 96 mutation catalog to a file on disk

`WriteCatSNS192` Write a SNS 192 mutation catalog to a file on disk

`WriteCatSNS1536` Write a SNS 1536 mutation catalog to a file on disk

`WriteCatDNS78` Write a DNS 78 mutation catalog to a file on disk

`WriteCatDNS144` Write a DNS 144 mutation catalog to a file on disk

`WriteCatDNS136` Write a 136 DNS catalog from path

`WriteCatID` Write a ID (insertion/deletion) catalog to a file on disk. (Please take note that deletion repeat size ranges from 0 to 5+ in the catalog, but for plotting and end user documentation it ranges from 1 to 6+.)

See also `ReadCatalog`

Index