Package 'ICAMS'

February 28, 2019

Type Package

Title In-depth Characterization and Analysis of Mutational Signatures

Version 0.0.0.9005

Author Steve Rozen, Nanhai Jiang, Arnoud Boot

Maintainer Steve Rozen <steverozen@gmail.com>

Description This package has functions to read in VCF files from Strelka and Mutect (in the Broad GATK package), create, read, and write single nucleotide substitutions (SNS), double nucleotide substitutions (DNS), insertions and deletions (ID) catalogs and do different types of plotting.
This alpha version only works with VCFs for human GRCh37, but will work for arbitrary human catalogs (assuming no major change in ``opportunities'' between GRCh37 and GRCh38).

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

VignetteBuilder knitr

Collate 'ICAMS.R'
'INDELS_related_functions.R'
'utility_functions.R'
'VCF_to_catalog_functions.R'
'data.R'
'plot.R'
'read_write_catalog.R'
'test_functions.R'

R topics documented:

Abundance *Nucleotide abundance*

Description

Nucleotide abundance information for a particular organism

Usage

```
abundance.2bp.exome.GRCh37

abundance.2bp.genome.GRCh37

abundance.3bp.exome.GRCh37

abundance.3bp.genome.GRCh37

abundance.4bp.exome.GRCh37

abundance.4bp.genome.GRCh37

abundance.5bp.exome.GRCh37

abundance.5bp.genome.GRCh37

abundance.2bp.exome.GRCh38

abundance.2bp.genome.GRCh38

abundance.3bp.exome.GRCh38

abundance.3bp.genome.GRCh38

abundance.4bp.exome.GRCh38

abundance.4bp.genome.GRCh38

abundance.5bp.exome.GRCh38

abundance.5bp.genome.GRCh38

abundance.2bp.exome.GRCm38

abundance.2bp.genome.GRCm38

abundance.3bp.exome.GRCm38

abundance.3bp.genome.GRCm38

abundance.4bp.exome.GRCm38
```

abundance.4bp.genome.GRCm38

abundance.5bp.exome.GRCm38

abundance.5bp.genome.GRCm38

Format

A single-column matrix containing the counts of particular sequences in a genome or part of a genome. This include 2-mers, 3-mers, 4-mers, 5-mers, stranded or strand-agnostic, and genome-wide, in-transcript, or in-exome, for different reference genome versions and for different organisms. The names should be self explanatory.

Details

abundance.2bp.genome.GRCh37, abundance.2bp.exome.GRCh37 A matrix containing dinucleotide abundance information for **Human** GRCh37. Its row names indicate 10 different types of 2 base pairs combinations while its column contains the occurrences of each type. It can be used in plotting function PlotCatDNS78ToPdf.

abundance.2bp.genome.GRCh38, abundance.2bp.exome.GRCh38 A matrix containing dinucleotide abundance information for **Human** GRCh38. Its row names indicate 10 different types of 2 base pairs combinations while its column contains the occurrences of each type. It can be used in plotting function PlotCatDNS78ToPdf.

abundance.2bp.genome.GRCm38, abundance.2bp.exome.GRCm38 A matrix containing dinucleotide abundance information for **Mouse** GRCm38. Its row names indicate 10 different types of 2 base pairs combinations while its column contains the occurrences of each type. It can be used in plotting function PlotCatDNS78ToPdf.

abundance.3bp.genome.GRCh37, abundance.3bp.exome.GRCh37 A matrix containing trinucleotide abundance information for **Human** GRCh37. Its row names indicate 32 different types of 3 base pairs combinations while its column contains the occurrences of each type. It can be used in plotting function PlotCatSNS96ToPdf.

abundance.3bp.genome.GRCh38, abundance.3bp.exome.GRCh38 A matrix containing trinucleotide abundance information for **Human** GRCh38. Its row names indicate 32 different types of 3 base pairs combinations while its column contains the occurrences of each type. It can be used in plotting function PlotCatSNS96ToPdf.

abundance.3bp.genome.GRCm37, abundance.3bp.exome.GRCm37 A matrix containing trinucleotide abundance information for **Mouse** GRCm37. Its row names indicate 32 different types of 3 base pairs combinations while its column contains the occurrences of each type. It can be used in plotting function PlotCatSNS96ToPdf.

abundance.4bp.genome.GRCh37, abundance.4bp.exome.GRCh37 A matrix containing tetranucleotide abundance information for **Human** GRCh37. Its row names indicate 136 different types of 4 base pairs combinations while its column contains the occurrences of each type. It can be used in plotting function PlotCatDNS136ToPdf.

abundance.4bp.genome.GRCh38, abundance.4bp.exome.GRCh38 A matrix containing tetranucleotide abundance information for **Human** GRCh38. Its row names indicate 136 different types of 4 base pairs combinations while its column contains the occurrences of each type. It can be used in plotting function PlotCatDNS136ToPdf.

abundance.4bp.genome.GRCm37, abundance.4bp.exome.GRCm37 A matrix containing tetranucleotide abundance information for **Mouse** GRCm37. Its row names indicate 136 different types of 4 base pairs combinations while its column contains the occurrences of each type. It can be used in plotting function PlotCatDNS136ToPdf.

abundance.5bp.genome.GRCh37, abundance.5bp.exome.GRCh37 A matrix containing pentanucleotide abundance information for **Human** GRCh37. Its row names indicate 512 different types of 5 base pairs combinations while its column contains the occurrences of each type. It can be used in plotting function PlotCatSNS1536ToPdf.

abundance.5bp.genome.GRCh38, abundance.5bp.exome.GRCh38 A matrix containing pentanucleotide abundance information for **Human** GRCh38. Its row names indicate 512 different types of 5 base pairs combinations while its column contains the occurrences of each type. It can be used in plotting function PlotCatSNS1536ToPdf.

abundance.5bp.genome.GRCm37, abundance.5bp.exome.GRCm37 A matrix containing pentanucleotide abundance information for **Mouse** GRCm37. Its row names indicate 512 different types of 5 base pairs combinations while its column contains the occurrences of each type. It can be used in plotting function PlotCatSNS1536ToPdf.

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+.

CatalogRowHeaders *Row headers information for writing a catalog to disk in standardized format*

Description

Row headers information for writing a catalog to disk in standardized format

Usage

```
catalog.row.headers.SNS.96

catalog.row.headers.SNS.192

catalog.row.headers.SNS.1536

catalog.row.headers.DNS.78

catalog.row.headers.DNS.144

catalog.row.headers.DNS.136

catalog.row.headers.ID
```

Format

A data frame which contains the row headers information for writing a catalog to disk in standardized format.

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+.

CatalogRowOrder *Canonical order of row names in a catalog*

Description

Canonical order of row names in a catalog

Usage

```
catalog.row.order.SNS.96

catalog.row.order.SNS.192

catalog.row.order.SNS.1536

catalog.row.order.DNS.78

catalog.row.order.DNS.144

catalog.row.order.DNS.136

catalog.row.order.ID
```

Format

A 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 matrix to be collapsed whose row names indicate the mutation types while its columns show the occurrences of each mutation type of different samples.

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 canonical catalog matrix whose row names indicate the mutation types while its columns show the occurrences of each mutation type of different samples.

`CreateDinucAbundance` *Create dinucleotide abundance file*

Description

Create dinucleotide abundance file

Usage

```
CreateDinucAbundance(path)
```

Arguments

path	Path to the file with the nucleotide abundance information with 4 base pairs.

Value

A matrix whose row names indicate 10 different types of 2 base pairs combinations while its column contains the occurrences of each type.

`CreatePentanucAbundance` *Create pentanucleotide abundance file*

Description

Create pentanucleotide abundance file

Usage

```
CreatePentanucAbundance(path)
```

Arguments

path	Path to the file with the nucleotide abundance information with 5 base pairs.

Value

A matrix whose row names indicate 512 different types of 5 base pairs combinations while its column contains the occurrences of each type.

CreateTetranucAbundance *Create tetranucleotide abundance file*

Description

Create tetranucleotide abundance file

Usage

```
CreateTetranucAbundance(path)
```

Arguments

path	Path to the file with the nucleotide abundance information with 4 base pairs.

Value

A matrix whose row names indicate 136 different types of 4 base pairs combinations while its column contains the occurrences of each type.

CreateTrinucAbundance *Create trinucleotide abundance file*

Description

Create trinucleotide abundance file

Usage

```
CreateTrinucAbundance(path)
```

Arguments

path	Path to the file with the nucleotide abundance information with 3 base pairs.

Value

A matrix whose row names indicate 32 different types of 3 base pairs combinations while its column contains the occurrences of each type.

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

This package has functions to read in VCF files from Strelka and Mutect (in the Broad GATK package), create, read, and write single nucleotide substitutions (SNS), double nucleotide substitutions (DNS), insertions and deletions (ID) catalogs and do different types of plotting.

Details

This alpha version only works with VCFs for human GRCh37, but will work for arbitrary **human** catalogs (assuming no major change in "opportunities" between GRCh37 and GRCh38).

Reading and splitting VCF files

1. ReadAndSplitStrelkaSNSVCFs Read and split Strelka single nucleotide substitution (SNS) VCFs (not Strelka indel VCFS).
2. ReadStrelkaIDVCFs Read Strelka indel (ID) VCFs (not Strelka SNS VCFS).
3. ReadAndSplitMutectVCFs Read and split Mutect VCFs, which contain indels and double nucleotide substitutions (DNSs) as well and SNSs.

Creating catalogs from 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 (indels) catalog from the Strelka ID VCFs.
3. MutectVCFFilesToCatalog, which creates 3 SNS catalogs (96, 192, 1536), 3 DNS catalogs (78, 136, 144) and ID (indels) catalog from the Mutect VCFs.

Reading catalogs

Functions for reading files that contain mutational spectrum catalogs in standardized format. These also work for reading mutational signature profiles. `ReadCatalog`

Writing catalogs

Functions for writing a mutational spectrum catalog to a file on disk. These also work for writing mutational signature profiles. `WriteCatalog`

Transforming catalogs

Functions for transforming count spectra from a particular organism region to an inferred count spectra based on the target nucleotide abundance. `TransformSpectra`

Collapsing catalogs

Functions for collapsing a mutation catalog. `CollapseCatalog`

Plotting catalogs

Functions for plotting mutation spectrum catalogs to a PDF file. These also work for plotting mutational signature profiles. `PlotCatalogToPdf`

Exported data

1. `CatalogRowOrder` Canonical order of row names in a catalog.
2. `CatalogRowHeaders` Row headers information for writing a catalog to disk in standardized format.
3. `Abundance` Nucleotide abundance information for a particular organism.
4. `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`	Name of a particular reference genome (without quotations marks).
`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.

PlotCatalogToPdf *Plot catalog to pdf functions*

Description

Plot mutation catalogs of various samples to a PDF file

Usage

```
PlotCatSNS96ToPdf(catalog, name, id = colnames(catalog),
  type = "density", grid = FALSE, upper = TRUE, xlabels = TRUE,
  abundance = NULL)

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

PlotCatSNS192StrandToPdf(catalog, name, id = colnames(catalog),
  type = "counts", cex = 1, abundance = NULL)

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

PlotCatDNS78ToPdf(catalog, name, id = colnames(catalog),
  type = "density", abundance = NULL)

PlotCatDNS144ToPdf(catalog, name, id = colnames(catalog),
  type = "counts", cex = 1, abundance = NULL)

PlotCatDNS136ToPdf(catalog, name, id = colnames(catalog),
  type = "density", abundance = NULL)

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

Arguments

catalog	A matrix of mutation counts. Rownames indicate the mutation types. Each column contains the mutation counts for one sample.
name	The name of the PDF file to be produced.
id	A vector containing the identifiers of the samples in catalog.
type	A vector of values indicating the type of plot for each sample. 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 "density" type for PlotCatID-toPdf function and the option of type = "density" is not implemented for function PlotCatSNS192ToPdf, PlotCatSNS192StrandToPdf and PlotCatDNS144ToPdf at the current stage.)
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.
abundance	A single column matrix, see Abundance, used only when `type = "density"`.
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.

`PlotCatSNS192StrandToPdf` 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.

`PlotCatDNS144ToPdf` 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.

`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.

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+.

ReadTranscriptRanges *Read transcript ranges and strands from a gff3 format file. Use this one for the new, cut down gff3 file (2018 11 24)*

Description

Read transcript ranges and strands from a gff3 format file. Use this one for the new, cut down gff3 file (2018 11 24)

Usage

```
ReadTranscriptRanges(path)
```

Arguments

path	Path to the file with the transcript information with 1-based start end positions of genomic ranges.

Value

A data.table keyed by chrom, chromStart, and chromEnd.

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	Name of a particular reference genome (without quotations marks).

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`	Name of a particular reference genome (without quotations marks).
`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)

`TestMakeCatalogFromMutectVCFs` *This function is to make catalogs from the sample Mutect VCF file to compare with the expected catalog information.*

Description

This function is to make catalogs from the sample Mutect VCF file to compare with the expected catalog information.

Usage

```
TestMakeCatalogFromMutectVCFs()
```

`TestMakeCatalogFromStrelkaIDVCFs` *This function is to make catalogs from the sample Strelka ID VCF files to compare with the expected catalog information.*

Description

This function is to make catalogs from the sample Strelka ID VCF files to compare with the expected catalog information.

Usage

```
TestMakeCatalogFromStrelkaIDVCFs()
```

`TestMakeCatalogFromStrelkaSNSVCFs` *This function is to make catalogs from the sample Strelka SNS VCF files to compare with the expected catalog information.*

Description

This function is to make catalogs from the sample Strelka SNS VCF files to compare with the expected catalog information.

Usage

```
TestMakeCatalogFromStrelkaSNSVCFs()
```

`TranscriptRanges` *Transcript ranges data*

Description

Transcript ranges and strand information for a particular organism

Usage

```
trans.ranges.GRCh37

old.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`.

`old.trans.ranges.GRCh37` A data.table which contains transcript range and strand information for **Human** GRCh37, which is derived from a raw **BED** format file and is keyed by chrom, chromStart, and chromEnd. This is mostly for testing purpose, may be removed in the future.

TransformSpectra *Transform nucleotide spectra functions*

Description

Transform count spectra from a particular organism region to an inferred count spectra based on the target nucleotide abundance.

Usage

```
TransDinucSpectra(catalog, source.abundance, target.abundance)

TransTrinucSpectra(catalog, source.abundance, target.abundance)

TransTetranucSpectra(catalog, source.abundance, target.abundance)

TransPentanucSpectra(catalog, source.abundance, target.abundance)
```

Arguments

`catalog`	A matrix of mutation counts. Rownames indicate the mutation types. Each column contains the mutation counts for one sample.
`source.abundance`	An abundance matrix specified by the user, which can be created using functions `CreateDinucAbundance`, `CreateTrinucAbundance`, `CreateTetranucAbundance`, `CreatePentanucAbundance`. There are 6 types of predefined abundance matrix which are incorporated in this function ("GRCh37.genome", "GRCh37.exome", "GRCh38.genome", "GRCh38.exome", "GRCm38.genome", "GRCm38.exome"). User can invoke a specific predefined abundance matrix by typing its name, e.g. source.abundance = "GRCh37.genome".

target.abundance
: An abundance matrix specified by the user, which can be created using functions CreateDinucAbundance, CreateTrinucAbundance, CreateTetranucAbundance, CreatePentanucAbundance. There are 6 types of predefined abundance matrix which are incorporated in this function ("GRCh37.genome", "GRCh37.exome", "GRCh38.genome", "GRCh38.exome", "GRCm38.genome", "GRCm38.exome"). User can invoke a specific predefined abundance matrix by typing its name, e.g. target.abundance = "GRCm38.genome".

Value

A matrix of inferred mutation counts. Rownames indicate the mutation types which are the same as those in catalog. Each column contains the inferred mutation counts for one sample based on target.abundance.

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 matrix of mutation catalog.
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