ICAMS

January 11, 2019

Type Package

Title In-depth Characterization and Analysis of Mutational Signatures

Version 0.0.0.9000

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 GATK, create SNS, DNS, ID catalogs and do different types of plotting.

License GPL-3

Encoding UTF-8

LazyData true

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

Depends R (>= 2.10)

RoxygenNote 6.1.1

Suggests knitr,
rmarkdown,
testthat

VignetteBuilder knitr

Collate 'utility_functions.R'
'VCF_related_functions.R'
'DNS_related_functions.R'
'INDELS_related_functions.R'
'PCAWG7_interaction_functions.R'
'SNS_related_functions.R'
'catalog_related_functions.R'

'plot_DNS_catalog.R'
'plot_INDELS_catalog.R'
'plot_SNS_catalog.R'
'read_write_catalog.R'
'test_functions.R'

R topics documented:

`AddSequence` *Add sequence context to a data frame with mutation records*

Description

Add sequence context to a data frame with mutation records

Usage

```
AddSequence(df, seq = BSgenome.Hsapiens.1000genomes.hs37d5)
```

Arguments

df	An input data frame storing mutation records of a VCF file.
seq	A particular reference genome.

Value

A data frame with a new column added to the input data frame, which contains sequence context information.

AddSequenceID *Add sequence context to a data frame with mutation records*

Description

Add sequence context to a data frame with mutation records

Usage

```
AddSequenceID(df, seq = BSgenome.Hsapiens.1000genomes.hs37d5)
```

Arguments

df	A data frame storing mutation records of a VCF file. IMPORTANT: The representation of indels in df must have been canonicalized, so that context bases (which are added by some indel callers) are placed in a column "Left.context.base" and so that, for deletions, ALT is the empty string, and, for insertions, REF is the empty string.
seq	A particular reference genome.

Value

A data frame with 2 new columns added to the input data frame. One column contains sequence context information and the other column contains the length of the "context" string to the left of the site of the variant.

`AddTranscript` *Add transcript information to a data frame with mutation records*

Description

Add transcript information to a data frame with mutation records

Usage

```
AddTranscript(df, trans.ranges)
```

Arguments

`df`	A data frame storing mutation records of a VCF file.
`trans.ranges`	A data.table with the genomic ranges and strands of transcripts.

Value

A data frame with new columns added to the input data frame, which contain the mutated gene's name, range and strand information.

`AliquotID2SampleID` *AliquotID2SampleID*

Description

AliquotID2SampleID

Usage

```
AliquotID2SampleID(aliquot.id)
```

Arguments

`aliquot.id`	TODO

Value

TODO

Canonicalize1DEL *Canonicalize1DEL*

Description

Canonicalize1DEL

Usage

```
Canonicalize1DEL(ref, alt, context)
```

Arguments

ref	TODO
alt	TODO
context	TODO

Value

TODO

Canonicalize1ID *Canonicalize1ID*

Description

Canonicalize1ID

Usage

```
Canonicalize1ID(ref, alt, context)
```

Arguments

ref	TODO
alt	TODO
context	TODO

Value

TODO

`Canonicalize1INS` *Canonicalize1INS*

Description

Canonicalize1INS

Usage

```
Canonicalize1INS(ref, alt, context)
```

Arguments

`ref`	TODO
`alt`	TODO
`context`	TODO

Value

TODO

`CanonicalizeDNS` *CanonicalizeDNS*

Description

CanonicalizeDNS

Usage

```
CanonicalizeDNS(ref.vec, alt.vec)
```

Arguments

`ref.vec`	TODO
`alt.vec`	TODO

Value

TODO

`CanonicalizeID` *CanonicalizeID*

Description

CanonicalizeID

Usage

```
CanonicalizeID(ref, alt, context)
```

Arguments

`ref`	TODO
`alt`	TODO
`context`	TODO

Value

TODO

`CanonicalizeQUAD` *CanonicalizeQUAD*

Description

CanonicalizeQUAD

Usage

```
CanonicalizeQUAD(quad)
```

Arguments

`quad`	TODO

Value

TODO

`Cat1536ToPdf` *Plot the 1536 mutation catalog of >= 1 samples to a PDF file*

Description

Plot the 1536 mutation catalog of >= 1 samples to a PDF file

Usage

```
Cat1536ToPdf(catalog, name, id = colnames(catalog), abundance)
```

Arguments

`catalog`	A matrix whose rownames indicate the 1536 SNS mutation types while its columns contain the counts of each mutation type from different samples. 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.
`name`	Name of the PDF file to be produced.
`id`	A vector containing the identifier of each sample.
`abundance`	A matrix containing pentanucleotide abundance information.

Value

invisible(TRUE)

`Cat192StrandToPdf` *Plot the transcription strand bias graph of 6 SNS mutation types ("C>A", "C>G", "C>T", "T>A", "T>C", "T>G") of different samples to a PDF file.*

Description

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

Usage

```
Cat192StrandToPdf(catalog, name, id = colnames(catalog),
  type = "counts", cex = 1, abundance = NULL)
```

Arguments

catalog	A matrix whose rownames indicate the 192 SNS mutation types while its columns contain the counts of each mutation type from different samples.
name	The name of the PDF file to be produced.
id	The ID information of the sample which has mutations.
type	A vector of values indicating the type of graph for each sample. If type = "counts", the graph will plot the occurrences of the 192 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 trinucleotides for each mutation type. The default value for type is "counts".
cex	A numerical value giving the amount by which mutation class labels, y axis labels, sample name and legend should be magnified relative to the default.
abundance	A matrix containing trinucleotide abundance and strand information, to be used only when type = "density".

Value

invisible(TRUE)

Cat192ToPdf *Plot the SNS 192 mutation catalog of different samples to a PDF file*

Description

Plot the SNS 192 mutation catalog of different samples to a PDF file

Usage

```
Cat192ToPdf(catalog, name, id = colnames(catalog), type = "counts",
  cex = 0.8, abundance = NULL)
```

Arguments

catalog	A matrix whose rownames indicate the 192 SNS mutation types while its columns contain the counts of each mutation type from different samples.
name	The name of the PDF file to be produced.
id	The ID information of the sample which has mutations.
type	A vector of values indicating the type of graph for each sample. If type = "counts", the graph will plot the occurrences of the 192 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 trinucleotides for each mutation type. The default value for type is "counts".
cex	A numerical value giving the amount by which mutation class labels on top of graph, y axis labels and sample name should be magnified relative to the default.
abundance	A matrix containing trinucleotide abundance and strand information, to be used only when type = "density".

Value

invisible(TRUE)

`Cat96ToPdf` *Plot the SNS 96 mutation catalog of different samples to a PDF file*

Description

Plot the SNS 96 mutation catalog of different samples to a PDF file

Usage

```
Cat96ToPdf(catalog, name, id = colnames(catalog), type = "density",
  abundance = NULL)
```

Arguments

`catalog`	A matrix whose rownames indicate the 96 SNS mutation types while its columns contain the counts of each mutation type from different samples.
`name`	The name of the PDF file to be produced.
`id`	A vector containing the ID information of different samples.
`type`	A vector of values indicating the type of plot for each sample. If type = "density", the graph will plot the rates of mutations per million trinucleotides for each mutation type. If type = "counts", the graph will plot the occurrences of the 96 mutation types in the sample. If type = "signature", the graph will plot mutation signatures of the sample. The default value for type is "density".
`abundance`	A matrix containing trinucleotide abundance information. To be used only when type = "density".

Value

invisible(TRUE)

`CatDNS144ToPdf` *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 different samples to a PDF file.*

Description

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 different samples to a PDF file.

Usage

```
CatDNS144ToPdf(catalog, name, id = colnames(catalog), type = "counts",
  cex = 1, abundance = NULL)
```

Arguments

catalog	A matrix whose rownames indicate the 144 DNS mutation types while its columns contain the counts of each mutation type from different samples.
name	The name of the PDF file to be produced.
id	The ID information of the sample which has mutations.
type	A vector of values indicating the type of graph for each sample. If type = "counts", the graph will plot the occurrences of the 10 major DNS mutation types in the sample. If type = "signature", the graph will plot mutation signatures of the 10 major DNS mutation types in the sample. If type = "density", the graph will plot the rates of mutations per million dinucleotides for each of the 10 major DNS mutation types. The default value for type is "counts".
cex	A numerical value giving the amount by which mutation class labels, y axis labels, sample name and legend should be magnified relative to the default.
abundance	A matrix containing dinucleotide abundance and strand information, to be used only when type = "density".

Value

invisible(TRUE)

CatDNS78ToPdf *Plot the DNS 78 mutation catalog of different samples to a PDF file*

Description

Plot the DNS 78 mutation catalog of different samples to a PDF file

Usage

```
CatDNS78ToPdf(catalog, name, id = colnames(catalog), type = "density",
  abundance = NULL)
```

Arguments

catalog	A matrix whose rownames indicate the 78 DNS mutation types while its columns contain the counts of each mutation type from different samples.
name	The name of the PDF file to be produced.
id	A vector containing the ID information of different samples.
type	A vector of values indicating the type of plot for each sample. If type = "density", the graph will plot the rates of mutations per million nucleotides for each mutation type. If type = "counts", the graph will plot the occurrences of the 78 mutation types in the sample. If type = "signature", the graph will plot mutation signatures of the sample. The default value for type is "density".
abundance	A matrix containing dinucleotide abundance information, to be used only when type = "density".

Value

invisible(TRUE)

`CatIDToPdf` *Plot the insertion and deletion catalog of different samples to a PDF file*

Description

Please take note that the deletions Repeat Size ranges from 0 to 5+ in the catalog, but for plotting and end user documentation it ranges from 1 to 6+.

Usage

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

Arguments

`catalog`	A matrix whose rownames indicate the insertion and deletion mutation types while its column contains the counts of each mutation type from different samples.
`name`	The name of the PDF file to be produced.
`id`	A vector containing the ID information of different samples.
`type`	A vector of values indicating the type of plot for each sample. If type = "counts", the graph will plot the occurrences of the insertion and deletion mutation types in the sample. If type = "signature", the graph will plot mutation signatures of the sample. The default value for type is "counts".

Value

invisible(TRUE)

`CaTypeAliquot2CaTypeSampleID` *CaTypeAliquot2CaTypeSampleID*

Description

CaTypeAliquot2CaTypeSampleID

Usage

```
CaTypeAliquot2CaTypeSampleID(ca.type.and.aliquot.id)
```

Arguments

`ca.type.and.aliquot.id`	TODO

Value

TODO

CheckSeqContextInVCF *Check that the sequence context information is consistent with the value of the column REF.*

Description

Check that the sequence context information is consistent with the value of the column REF.

Usage

```
CheckSeqContextInVCF(vcf, column.to.use)
```

Arguments

vcf	In-memory VCF as a data.frame; must be an SNS or DNS VCF.
column.to.use	The column name as a string of the column in the VCF with the context information

Value

Throws error with location information if the value of REF is inconsistent with the value of seq.21context. Assumes the first base of the reference allele is at position (size(<context string>)-1)/2, and generates error if this is not an integer. Indices are 1-based.

Collapse144to78 *Collapse a DNS 144 catalog matrix to a DNS 78 catalog matrix*

Description

Collapse a DNS 144 catalog matrix to a DNS 78 catalog matrix

Usage

```
Collapse144to78(catDNS144)
```

Arguments

catDNS144	A DNS 144 catalog matrix whose row names indicate the 192 mutation types while its columns show the occurrences of each mutation type of different samples.

Value

A DNS 78 catalog matrix whose row names indicate the 96 mutation types while its columns show the occurrences of each mutation type of different samples.

Collapse1536to96 *Collapse a SNS 1536 catalog matrix to a 96 catalog matrix*

Description

Collapse a SNS 1536 catalog matrix to a 96 catalog matrix

Usage

```
Collapse1536to96(cat1536)
```

Arguments

cat1536	A SNS 1536 catalog matrix whose row names indicate the 1536 mutation types while its columns show the occurrences of each mutation type of different samples.

Value

A SNS 96 catalog matrix whose row names indicate the 96 mutation types while its columns show the occurrences of each mutation type of different samples.

Collapse192to96 *Collapse a SNS 192 catalog matrix to a 96 catalog matrix*

Description

Collapse a SNS 192 catalog matrix to a 96 catalog matrix

Usage

```
Collapse192to96(cat192)
```

Arguments

cat192	A SNS 192 catalog matrix whose row names indicate the 192 mutation types while its columns show the occurrences of each mutation type of different samples.

Value

A SNS 96 catalog matrix whose row names indicate the 96

CreateOneColDNSCatalog

*Create double nucleotide catalog for *one* sample from a Variant Call Format (VCF) file*

Description

Create double nucleotide catalog for *one* sample from a Variant Call Format (VCF) file

Usage

```
CreateOneColDNSCatalog(vcf, sample.id = "count")
```

Arguments

vcf	An in-memory VCF file annotated by the AddSequence and AddTranscript functions. It must *not* contain indels and must *not* contain SNS (single nucleotide substituions), or triplet base substituions etc.
sample.id	Usually the sample id, but defaults to "count".

Value

A list of three matrices containing the DNS catalog: catDNS78, catDNS144, catQUAD136 respectively.

CreateOneColIDCatalog

*Create mutation catalog for *one* sample from a Variant Call Format (VCF) file*

Description

Create mutation catalog for *one* sample from a Variant Call Format (VCF) file

Usage

```
CreateOneColIDCatalog(ID.vcf, SBS.vcf)
```

Arguments

ID.vcf	An in-memory VCF as a data.frame annotated by the AddSequence and AddTranscript functions. It must only contain indels and must *not* contain SBS (single base substituions), DBS or triplet base substituions etc. * Sequence must already have been added to ID.vcf One design decision for variant callers is the representation of "complex indels", e.g. mutations e.g. CAT > GC. Some callers represent this as C>G, A>C, and T>_. Others might represent it as CAT > CG. Multiple issues can arise. In PCAWG, overlapping indel/SBS calls from different callers were included in the indel VCFs.

SBS.vcf	An in-memory VCF as a data frame. Because we have to work with some PCAWG data, we will look for neigboring indels and indels adjoining SBS. That means this functions takes an SBS VCF and an ID VCF from the same sample.

Value

Returns a list with two elemsents: ID.cat: A matrix containing the mutation catalog information. Problems: Locations of neighboring indels or indels neighboring SBS. In the future we might handle these depending on what we find in the indel calls from different variant callers.

CreateOneColSNSCatalog

*Create single nucleotide mutation catalog for *one* sample from a Variant Call Format (VCF) file.*

Description

Create single nucleotide mutation catalog for *one* sample from a Variant Call Format (VCF) file.

Usage

```
CreateOneColSNSCatalog(vcf, sample.id = "count")
```

Arguments

vcf	An in-memory VCF file annotated by the AddSequence and AddTranscript functions. It must *not* contain indels and must *not* contain DNS (double nucleotide substituions), or triplet base substituions etc., even if encoded as neighboring SNS.
sample.id	Usually the sample id, but defaults to "count".

Value

A list of three matrices containing the SNS mutation catalog: 96, 192, 1536 catalog respectively.

CreateSampleId *CreateSampleId*

Description

Get catalog sorting function from msigtools

Usage

```
CreateSampleId(filename, aliquot.id)
```

Arguments

filename	TODO
aliquot.id	TODO

Value

TODO

`CreateTransRange` *Create a Transcript Range file from the raw GFF3 File*

Description

Create a Transcript Range file from the raw GFF3 File

Usage

```
CreateTransRange(path)
```

Arguments

`path`	The name/path of the raw GFF3 File, or a complete URL.

Value

A data frame which contains chromosome name, start, end position, strand information and gene name. Only the following four gene types are kept to facilitate transcriptional strand bias analysis: protein_coding, retained_intron, processed_transcript and nonsense_mediated_decay.

`DirectoryOfSBSSimple2Catalog`
DirectoryOfSBSSimple2Catalog

Description

DirectoryOfSBSSimple2Catalog

Usage

```
DirectoryOfSBSSimple2Catalog(path.to.dir, cts = empty.cats,
  verbose = 3)
```

Arguments

`path.to.dir`	TODO
`cts`	TODO
`verbose`	TODO

Value

TODO

`DirectoryOfSBSSimple2ListOfVCF` *DirectoryOfSBSSimple2ListOfVCF*

Description

DirectoryOfSBSSimple2ListOfVCF

Usage

```
DirectoryOfSBSSimple2ListOfVCF(path.to.dir, cts = empty.cats,
  verbose = 3)
```

Arguments

`path.to.dir`	TODO
`cts`	TODO
`verbose`	TODO

Value

TODO

`DNSVCFsToCatalogs` *Create a list of 3 catalogs (one each for DNS78, DNS144 and QUAD136) out of the contents of the VCFs in list.of.vcfs*

Description

Create a list of 3 catalogs (one each for DNS78, DNS144 and QUAD136) out of the contents of the VCFs in list.of.vcfs

Usage

```
DNSVCFsToCatalogs(list.of.vcfs, genome, trans.ranges)
```

Arguments

`list.of.vcfs`	List vector of in-memory VCFs. The list names will be the sample ids in the output catalog.
`genome`	Name of a particular reference genome (without quotations marks).
`trans.ranges`	A data frame containing transcript ranges.

Value

A list of 3 catalogs, one each for DNS78, DNS144, QUAD136: catDNS78 catDNS144 catQUAD136

`FindDelMH` *FindDelMH*

Description

Microhomology can be alligned in multiple equivalent ways. Example:

Usage

```
FindDelMH(context, q, pos)
```

Arguments

`context`	TODO
`q`	TODO
`pos`	TODO

Details

GGCTAGTT aligned to

GGCTAGAACTAGTT GG——CTAGTT GGCTAGTT GG[CTAGAA]CTAGTT —- —- GGC——TAGTT GGCTAGTT GGC[TAGAAC]TAGTT * — * — GGCT——AGTT GGCTAGTT GGCTA——GTT GGCTAGTT GGCTAG——TT GGCTAGTT

All the same pairs of sequence, aligned 5 different ways. 4 bp of microhomology.

Need to find:

(1) The maxium match of undeleted sequence on left that is identical to the right end of deleted sequence, and

(2) The maxium match of undeleted sequence on right that is identical to the left end of deleted sequence.

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

Value

TODO

`FindMaxRepeatDel` *FindMaxRepeatDel*

Description

q is a substring of context at pos to pos + len(q) - 1

Usage

```
FindMaxRepeatDel(context, q, pos)
```

Arguments

context	TODO
q	TODO
pos	TODO

Details

e.g. q = ac pos = 3 context = xyaczt pos ^

for deletion, if substr(context, pos, pos + len(q) - 1) != q there is an error.

Value

TODO

`FindMaxRepeatIns` *FindMaxRepeatIns*

Description

If q is an insertion into context between pos and pos+1 if q is repeated in context it might start at pos+1:

Usage

```
FindMaxRepeatIns(context, q, pos)
```

Arguments

context	TODO
q	TODO
pos	TODO

Details

e.g. q = ac pos = 4 context = abxyac pos ^ start ^

or q might start at pos + 1 - len(q)

e.g. q = ac pos = 4 context = xyaczz pos ^ start ^

Value

TODO

`GetStrelkaVAF` *Extract the VAFs (variant allele frequencies) from a VAF created by Strelka version 1*

Description

Extract the VAFs (variant allele frequencies) from a VAF created by Strelka version 1

Usage

```
GetStrelkaVAF(strelka.vcf)
```

Arguments

`strelka.vcf`	said VCF as a data.frame

Value

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

`InitAliquotSample` *InitAliquotSample*

Description

In InitAliquotSample, note the hack to add one more mapping. One aliquot ID is not in PCAWG as per ZHANG Junjun's email https://mail.google.com/mail/u/0/#search/Junjun.Zhang [1] "CMDI-UK::f92cf0a2-0172-a1cc-e040-11ac0c486b1a" > setdiff(StripCaType(colnames(ofcat96)), StripCaType(ycol)) [1] "CMDI-UK::SP116883" was the associated sample id. See https://mail.google.com/mail/u/0/#search/SP116883/FMfc

Usage

```
InitAliquotSample()
```

Value

TODO

`MakeVCFDNSdf` *Take DNS ranges and the original VCF and generate a VCF with dinucleotide REF and ALT alleles. The output VCF has minimal columns: just CHROM, POS, ID, REF, ALT.*

Description

Take DNS ranges and the original VCF and generate a VCF with dinucleotide REF and ALT alleles. The output VCF has minimal columns: just CHROM, POS, ID, REF, ALT.

Usage

```
MakeVCFDNSdf(DNS.range.df, SNS.vcf.dt)
```

Arguments

`DNS.range.df`	Data frame with columns CHROM, LOW, HIGH
`SNS.vcf.dt`	TODO

Value

TODO

`OneSBSVCFAllCatalogs` *OneSBSVCFAllCatalogs*

Description

TODO(steve) merge this in where needed

Usage

```
OneSBSVCFAllCatalogs(vcf, genome, sample.id = "count", verbose = 0)
```

Arguments

`vcf`	TODO
`genome`	TODO
`sample.id`	TODO
`verbose`	TODO

Value

TODO

PlotCat1536 *Plot the pentanucleotide sequence contexts for one sample, normalized by pentanucleotide occurrence in the genome.*

Description

Plot the pentanucleotide sequence contexts for one sample, normalized by pentanucleotide occurrence in the genome.

Usage

```
PlotCat1536(catalog, id, scale = TRUE, abundance)
```

Arguments

catalog	A matrix whose rownames indicate the 1536 SNS mutation types while its column contains the counts of each mutation type. 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.
id	The id of the sample to be displayed on top of the graph.
scale	A logical value indicating whether to do color scaling for all mutation types.
abundance	A matrix containing pentanucleotide abundance information.

Value

invisible(TRUE)

PlotCat192 *Plot the SNS 192 mutation catalog of one sample*

Description

Plot the SNS 192 mutation catalog of one sample

Usage

```
PlotCat192(catalog, id, type = "counts", cex = 0.8, abundance = NULL)
```

Arguments

catalog	A matrix whose rownames indicate the 192 SNS mutation types while its column contains the counts of each mutation type.
id	The ID information of the sample which has mutations.
type	A value indicating the type of the graph. If type = "counts", the graph will plot the occurrences of the 192 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 trinucleotides for each mutation type. The default value for type is "counts".

cex	A numerical value giving the amount by which mutation class labels on top of graph, y axis labels and sample name should be magnified relative to the default.
abundance	A matrix containing trinucleotide abundance and strand information, to be used only when type = "density".

Value

invisible(TRUE)

`PlotCat192Strand` *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*

Description

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

Usage

```
PlotCat192Strand(catalog, id, type = "counts", cex = 1,
  abundance = NULL)
```

Arguments

catalog	A matrix whose rownames indicate the 192 SNS mutation types while its column contains the counts of each mutation type.
id	The ID information of the sample which has mutations.
type	A value indicating the type of the graph. If type = "counts", the graph will plot the occurrences of the 6 SNS mutation types in the sample. If type = "signature", the graph will plot mutation signatures of the 6 SNS mutation types in the sample. If type = "density", the graph will plot the rates of mutations per million trinucleotides for each of the 6 SNS mutation types. The default value for type is "counts".
cex	A numerical value giving the amount by which mutation class labels, y axis labels, sample name and legend should be magnified relative to the default.
abundance	A matrix containing trinucleotide abundance and strand information, to be used only when type = "density".

Value

invisible(TRUE)

PlotCat96 *Plot the SNS 96 mutation catalog of one sample*

Description

Plot the SNS 96 mutation catalog of one sample

Usage

```
PlotCat96(catalog, id, type = "density", abundance = NULL)
```

Arguments

catalog	A matrix whose rownames indicate the 96 SNS mutation types while its columns contain the counts of each mutation type.
id	The ID information of the sample which has mutations.
type	A value indicating the type of the graph. If type = "density", the graph will plot the rates of mutations per million trinucleotides for each mutation type. If type = "counts", the graph will plot the occurrences of the 96 mutation types in the sample. If type = "signature", the graph will plot mutation signatures of the sample. The default value for type is "density".
abundance	A matrix containing trinucleotide abundance information. To be used only when type = "density".

Value

invisible(TRUE)

PlotCatDNS144 *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.*

Description

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.

Usage

```
PlotCatDNS144(catalog, id, type = "counts", cex = 1,
  abundance = NULL)
```

Arguments

catalog	A matrix whose rownames indicate the 144 DNS mutation types while its column contains the counts of each mutation type.
id	The ID information of the sample which has mutations.
type	A value indicating the type of the graph. If type = "counts", the graph will plot the occurrences of the 10 major DNS mutation types in the sample. If type = "signature", the graph will plot mutation signatures of the 10 major DNS mutation types in the sample. If type = "density", the graph will plot the rates of mutations per million dinucleotides for each of the 10 major DNS mutation types. The default value for type is "counts".
cex	A numerical value giving the amount by which mutation class labels, y axis labels, sample name and legend should be magnified relative to the default.
abundance	A matrix containing dinucleotide abundance and strand information, to be used only when type = "density".

Value

invisible(TRUE)

PlotCatDNS78 *Plot the DNS 78 mutation catalog of one sample*

Description

Plot the DNS 78 mutation catalog of one sample

Usage

```
PlotCatDNS78(catalog, id, type = "density", abundance = NULL)
```

Arguments

catalog	A matrix whose rownames indicate the 78 DNS mutation types while its columns contain the counts of each mutation type from different samples.
id	The ID information of the sample which has mutations.
type	A value indicating the type of the graph. If type = "density", the graph will plot the rates of mutations per million nucleotides for each mutation type. If type = "counts", the graph will plot the occurrences of the 78 mutation types in the sample. If type = "signature", the graph will plot mutation signatures of the sample. The default value for type is "density".
abundance	A matrix containing dinucleotide abundance information, to be used only when type = "density".

Value

invisible(TRUE)

PlotCatID *Plot the insertion and deletion catalog of one sample.*

Description

Please take note that the deletions Repeat Size ranges from 0 to 5+ in the catalog, but for plotting and end user documentation it ranges from 1 to 6+.

Usage

```
PlotCatID(catalog, id, type = "counts")
```

Arguments

catalog	A matrix whose rownames indicate the insertion and deletion mutation types while its column contains the counts of each mutation type.
id	The ID information of the sample which has mutations.
type	A value indicating the type of the graph. If type = "counts", the graph will plot the occurrences of the insertion and deletion mutation types in the sample. If type = "signature", the graph will plot mutation signatures of the sample. The default value for type is "counts".

Value

invisible(TRUE)

PyrPenta *PyrPenta*

Description

PyrPenta

Usage

```
PyrPenta(mutstring)
```

Arguments

mutstring	a mutation string

Value

a mutation string

PyrTri *PyrTri*

Description

PyrTri

Usage

```
PyrTri(mutstring)
```

Arguments

mutstring	a mutation string

Value

a mutation string

ReadAbundance3Bp *Read data from a nucleotide abundance file with 3 base pairs*

Description

Read data from a nucleotide abundance file with 3 base pairs

Usage

```
ReadAbundance3Bp(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.

ReadAbundance4Bp *Read data from a nucleotide abundance file with 4 base pairs*

Description

Read data from a nucleotide abundance file with 4 base pairs

Usage

```
ReadAbundance4Bp(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.

ReadAbundance5Bp *Read data from a nucleotide abundance file with 5 base pairs*

Description

Read data from a nucleotide abundance file with 5 base pairs

Usage

```
ReadAbundance5Bp(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.

`ReadBedTranscriptRanges` *Read transcript ranges and strands from a bed format file. Mostly for testing.*

Description

Read transcript ranges and strands from a bed format file. Mostly for testing.

Usage

```
ReadBedTranscriptRanges(path)
```

Arguments

`path`	Path to the file with the transcript information (in bed format).

Value

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

`ReadCat1536` *Read a 1536 SNS catalog from path*

Description

Read a 1536 SNS catalog from path

Usage

```
ReadCat1536(path, strict = TRUE)
```

Arguments

`path`	Path to a catalog on disk in the "PCAWG7" format.
`strict`	If TRUE, do additional checks on the input, and stop if the checks fail.

Value

A catalog in canonical in-memory format.

ReadCat192 *Read a 192 SNS catalog from path*

Description

Read a 192 SNS catalog from path

Usage

```
ReadCat192(path, strict = TRUE)
```

Arguments

path	Path to a catalog on disk in the "PCAWG7" format.
strict	If TRUE, do additional checks on the input, and stop if the checks fail.

Value

A catalog in canonical in-memory format.

ReadCat96 *Read 96-channel spectrum or signatures in PCAWG7 format*

Description

Read 96-channel spectrum or signatures in PCAWG7 format

Usage

```
ReadCat96(path, strict = TRUE)
```

Arguments

path	Path to a catalog on disk in the "PCAWG7" format.
strict	If TRUE, do additional checks on the input, and stop if the checks fail.

Value

A catalog in canonical in-memory format.

ReadCatDNS144 *Read a 144 DNS catalog from path*

Description

Read a 144 DNS catalog from path

Usage

```
ReadCatDNS144(path, strict = TRUE)
```

Arguments

path	Path to a catalog on disk in the "PCAWG7" format.
strict	If TRUE, do additional checks on the input, and stop if the checks fail.

Value

A catalog in canonical in-memory format.

ReadCatDNS78 *Read a 78 DNS catalog from path*

Description

Read a 78 DNS catalog from path

Usage

```
ReadCatDNS78(path, strict = TRUE)
```

Arguments

path	Path to a catalog on disk in the "PCAWG7" format.
strict	If TRUE, do additional checks on the input, and stop if the checks fail.

Value

A catalog in canonical in-memory format.

ReadCatID — *Read a ID (insertion/deletion) catalog from path*

Description

Please take note that the deletions Repeat Size ranges from 0 to 5+ in the catalog, but for plotting and end user documentation it ranges from 1 to 6+.

Usage

```
ReadCatID(path, strict = TRUE)
```

Arguments

path	Path to a catalog on disk in the "PCAWG7" format.
strict	If TRUE, do additional checks on the input, and stop if the checks fail.

Value

A catalog in canonical in-memory format.

ReadCatQUAD136 — *Read a 136 QUAD catalog from path*

Description

Read a 136 QUAD catalog from path

Usage

```
ReadCatQUAD136(path, strict = TRUE)
```

Arguments

path	Path to a catalog on disk in the "PCAWG7" format.
strict	If TRUE, do additional checks on the input, and stop if the checks fail.

Value

A catalog in canonical in-memory format.

ReadIDSimpleAsVCF *ReadIDSimpleAsVCF*

Description

ReadIDSimpleAsVCF

Usage

```
ReadIDSimpleAsVCF(path, verbose = 0)
```

Arguments

path	TODO
verbose	TODO

Value

TODO

ReadListOfVCFs *Read a list of VCF files from path*

Description

Read a list of VCF files from path

Usage

```
ReadListOfVCFs(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.

`ReadSimpleAsVCF` *ReadSimpleAsVCF*

Description

ReadSimpleAsVCF

Usage

```
ReadSimpleAsVCF(path, verbose = 0)
```

Arguments

`path`	TODO
`verbose`	TODO

Value

TODO

`ReadStrelkaVCF` *Read in the data lines of a Variant Call Format (VCF) file*

Description

Read in the data lines of a Variant Call Format (VCF) file

Usage

```
ReadStrelkaVCF(path)
```

Arguments

`path`	The name/path of the VCF file, or a complete URL.

Value

A data frame storing mutation records of a VCF file.

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

Description

Read transcript ranges and strands from a bed 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 of every string in string.vec.

RevcDNS144 *RevcDNS144*

Description

RevcDNS144

Usage

```
RevcDNS144(mutstring)
```

Arguments

mutstring	TODO

Value

TODO

RevcSNS96 *RevcSNS96*

Description

RevcSNS96

Usage

```
RevcSNS96(mutstring)
```

Arguments

mutstring	a mutation string

Value

a mutation string

SampleID2AliquotID *SampleID2AliquotID*

Description

SampleID2AliquotID

Usage

```
SampleID2AliquotID(sample.id)
```

Arguments

sample.id	TODO

Value

TODO

SNSVCFsToCatalogs *Create a list of 3 catalogs (one each for 96, 192, 1536) out of the contents of the VCFs in list.of.vcfs*

Description

Create a list of 3 catalogs (one each for 96, 192, 1536) out of the contents of the VCFs in list.of.vcfs

Usage

```
SNSVCFsToCatalogs(list.of.vcfs, genome, trans.ranges)
```

Arguments

list.of.vcfs	List vector of in-memory VCFs. The list names will be the sample ids in the output catalog.
genome	Name of a particular reference genome (without quotations marks).
trans.ranges	A data frame containing transcript ranges.

Value

A list of 3 catalogs, one each for 96, 192, 1536: cat96 cat192 cat1536

SplitCaTypeAndID *SplitCaTypeAndID*

Description

SplitCaTypeAndID

Usage

```
SplitCaTypeAndID(x)
```

Arguments

x	TODO

Value

TODO

SplitSNSVCF *Split an in-memory VCF into SNS, DNS, and variants involving > 2 consecutive bases*

Description

SNSs are single nucleotide substitutions, eg C>T, A<G,.... DNSs are double nucleotide substitutions, eg CC>TT, AT>GG, ... Variants involving > 2 consecutive bases are rare, so this function just records them. These would be variants such ATG>CCT, AGAT > TCTA, ...

Usage

```
SplitSNSVCF(vcf.df, max.vaf.diff = 0.02)
```

Arguments

vcf.df	An in-memory data frame containing a VCF file contents.
max.vaf.diff	The maximum difference of VAF, default value is 0.02.

Value

A list of 3 in-memory objects with the elements:

StandardChromName *Standardize the Chromosome name annotations for a data frame*

Description

Standardize the Chromosome name annotations for a data frame

Usage

```
StandardChromName(df)
```

Arguments

df	A data frame whose first column contains the Chromosome name.

Value

A data frame whose Chromosome names are only in the form of 1:22, "X" and "Y".

StripCaType *StripCaType*

Description

StripCaType

Usage

```
StripCaType(x)
```

Arguments

x	TODO

Value

TODO

`TestDNSCatalog` *This function is to test whether the predefined functions are working correctly to produce the desired DNS catalogs*

Description

This function is to test whether the predefined functions are working correctly to produce the desired DNS catalogs

Usage

```
TestDNSCatalog(vcf.df)
```

Arguments

`vcf.df`	An in-memory data frame containing a VCF file contents.

`TestMakeCatalogFromSNSVCFs` *This function is to make catalogs from the sample VCF files to compare with the expected catalog information*

Description

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

Usage

```
TestMakeCatalogFromSNSVCFs()
```

`TestSNSandDNSCat` *This function is to test whether the predefined functions are working correctly to produce the desired SNS and DNS catalogs*

Description

This function is to test whether the predefined functions are working correctly to produce the desired SNS and DNS catalogs

Usage

```
TestSNSandDNSCat()
```

`TestSNSCatalog` *This function is to test whether the predefined functions are working correctly to produce the desired SNS catalogs*

Description

This function is to test whether the predefined functions are working correctly to produce the desired SNS catalogs

Usage

```
TestSNSCatalog(vcf.df)
```

Arguments

`vcf.df`	An in-memory data frame containing a VCF file contents.

`VCFFiles2Catalog` *Create 3 SNS catalogs (96, 192, 1536) and 3 DNS catalogs (78, 136, 144) in the VCFs specified by vector.of.file.paths*

Description

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

Usage

```
VCFFiles2Catalog(vector.of.file.paths, genome, trans.ranges)
```

Arguments

`vector.of.file.paths`	A vector containing the paths of the VCF files.
`genome`	Name of a particular reference genome (without quotations marks).
`trans.ranges`	A data.table which contains transcript range and strand information.

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)

WriteCat — *Write a matrix of mutation catalog to a file on disk*

Description

Write a matrix of mutation catalog to a file on disk

Usage

```
WriteCat(ct, path, num.row, row.order, row.header, strict)
```

Arguments

ct	A matrix of mutation catalog.
path	The path of the file to be written on disk.
num.row	The number of rows in the file to be written.
row.order	The row order to be used for writing the file.
row.header	The row header to be used for writing the file.
strict	If TRUE, do additional checks on the input, and stop if the checks fail.

WriteCat1536 — *Write a SNS 1536 mutation catalog to a file on disk*

Description

Write a SNS 1536 mutation catalog to a file on disk

Usage

```
WriteCat1536(ct, path, strict = TRUE)
```

Arguments

ct	A matrix of SNS 1536 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.

WriteCat192 *Write a SNS 192 mutation catalog to a file on disk*

Description

Write a SNS 192 mutation catalog to a file on disk

Usage

```
WriteCat192(ct, path, strict = TRUE)
```

Arguments

ct	A matrix of SNS 192 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.

WriteCat96 *Write a SNS 96 mutation catalog to a file on disk*

Description

Write a SNS 96 mutation catalog to a file on disk

Usage

```
WriteCat96(ct, path, strict = TRUE)
```

Arguments

ct	A matrix of SNS 96 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.

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

Description

Write a DNS 144 mutation catalog to a file on disk

Usage

```
WriteCatDNS144(ct, path, strict = TRUE)
```

Arguments

ct	A matrix of DNS 144 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.

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

Description

Write a DNS 78 mutation catalog to a file on disk

Usage

```
WriteCatDNS78(ct, path, strict = TRUE)
```

Arguments

ct	A matrix of DNS 78 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.

WriteCatID *Write a ID (insertion/deletion) catalog to a file on disk*

Description

Please take note that the deletions Repeat Size ranges from 0 to 5+ in the catalog, but for plotting and end user documentation it ranges from 1 to 6+.

Usage

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

Arguments

ct	A matrix of ID (insertion/deletion) 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.

WriteCatQUAD136 *Write a QUAD 136 catalog to a file on disk*

Description

Write a QUAD 136 catalog to a file on disk

Usage

```
WriteCatQUAD136(ct, path, strict = TRUE)
```

Arguments

ct	A matrix of QUAD 136 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.

WriteMinimalVCFList *WriteMinimalVCFList*

Description

Each VCF in list.of.VCFs needs to have columns CHROM, POS, REF, ALT, VAF, STRAND ??????. They will be extracted and re-ordered if necessary to be consistent. The header line will begin with #CHROM

Usage

```
WriteMinimalVCFList(list.of.VCFs, path)
```

Arguments

list.of.VCFs	TODO
path	TODO

Value

TODO

Index