

Package ‘SATS’

September 15, 2026

Title Signature Analyzer for Targeted Sequencing (SATS)

Version 1.0.11

Date 2026-09-15

Description Performs mutational signature analysis for tumors profiled by targeted sequencing. Unlike the canonical analysis of mutational signatures, SATS factorizes the mutation count matrix into a panel-context matrix (measuring the number of mutation opportunities per million base pairs for each tumor), a signature profile matrix, and a signature activity matrix. SATS also calculates the expected number of mutations attributed to each signature, namely the signature burden, for each tumor profiled by targeted sequencing. For more details see Lee et al. (2024) <[doi:10.1101/2023.05.18.23290188](https://doi.org/10.1101/2023.05.18.23290188)>.

Imports stats, glmnet, GenomicRanges, IRanges, Biostrings, dplyr

Depends R (>= 4.1.0)

Suggests testthat, BSgenome.Hsapiens.UCSC.hg19,
BSgenome.Hsapiens.UCSC.hg38

License GPL-2

NeedsCompilation yes

Author DongHyuk Lee [aut],
Bin Zhu [aut],
Bill Wheeler [cre]

Maintainer Bill Wheeler <wheelerb@imsweb.com>

Repository CRAN

Date/Publication 2026-09-15 21:40:08 UTC

Contents

SATS-package	2
CalculateSignatureBurdens	3
EstimateSigActivity	4
GenerateLMatrix	5
GeneratePanelSize	7
GenerateVMatrix	8
MappingSignature	9

ReadBEDAsPanelInfo	10
ReadVCFAsMutationRecord	12
RefTMB	13
SimData	13
ValidateSATSInputs	14
Index	16

SATS-package	<i>SATS (Signature Analyzer for Targeted Sequencing)</i>
--------------	--

Description

This package performs mutational signature analysis for tumors profiled by targeted sequencing. Unlike the canonical analysis of mutational signatures, SATS factorizes the mutation counts matrix into a panel context matrix (measuring mutation opportunities per million base pairs for each tumor), a signature profile matrix, and a signature activity matrix. SATS also calculates the expected number of mutations attributed by a signature, namely signature burden, for each tumor profiled by targeted sequencing.

Details

This package includes a novel algorithm, SATS, to perform mutational signature analysis for targeted sequencing data. The algorithm first applies the signeR algorithm to extract profiles of de novo mutational signatures by appropriately adjusting for various panel sizes. Next, the profiles of identified de novo mutational signatures are mapped to the profiles of catalog signatures of tumor mutation burden (TMB), measured as the number of mutations per million base pairs, using penalized non-negative least squares. Then, given the panel sizes and profiles of mapped TMB catalog signatures, signature activities are estimated for all samples simultaneously through the Expectation-Maximization (EM) algorithm. Finally, the expected number of mutations attributed by a signature, namely signature burden, is calculated for each tumor profiled by targeted sequencing.

The main functions in this package are [GenerateVMatrix](#), [GenerateLMatrix](#), [ValidateSATSInputs](#), [EstimateSigActivity](#), [CalculateSignatureBurdens](#), and [MappingSignature](#).

The accompanying manuscript applies SATS to 111,711 tumors profiled by targeted sequencing from AACR Project GENIE to construct a real-world, panel-calibrated pan-cancer catalogue of mutational signatures.

Author(s)

Donghyuk Lee <dhyuklee@pusan.ac.kr> and Bin Zhu <bin.zhu@nih.gov>

References

Lee, D., Hua, M., Wang, D., Song, L., Yu, K., Yang, X., Shi, J., Landi, M., Zhu, B. A real-world pan-cancer catalogue of mutational signatures from 111,711 tumors. Submitted.

CalculateSignatureBurdens
Calculate signature burdens

Description

Estimation of the expected number of mutations attributed by TMB-based catalog signatures (signature burden) given the sample-level panel-context matrix, the catalog signature-profile matrix and the signature activities matrix.

Usage

```
CalculateSignatureBurdens(L, W, H)
```

Arguments

L	Sample-level panel-context matrix or data frame with samples in columns, see GenerateLMatrix
W	Catalog signature-profile matrix or data frame with signatures in columns
H	Activity matrix or data frame with samples in columns, see EstimateSigActivity

Details

The sample-level panel-context matrix L is of size P (the mutation context) by N (the sample size). The catalog signature-profile matrix W has dimension P by K (the number of signatures) and the activity matrix H is of size K by N. For single base substitutions (SBS), P is 96. If K is the number of signatures and N is the number of samples, then H must be of dimension K X N, $\text{ncol}(L) = N$, and $\text{ncol}(W) = K$. The input matrices must contain numeric, finite and non-negative values. When row names and column names are present, CalculateSignatureBurdens() checks that L, W and H contain matched mutation contexts, signature names and sample IDs. If the same names are present in a different order, L and W are reordered to match H. If one object has names on an axis used for alignment and the paired object does not, or if the named entries differ between matrices, the function stops with an informative error. Users can run [ValidateSATSInputs](#) directly to check and align inputs before calling this function. For the catalog signature profile matrix W, reference SBS TMB signature profiles in data(SimData) can be used.

Value

A matrix of dimension K X N, where K is the number of signatures and N is the number of samples.

Author(s)

Donghyuk Lee <dhyuklee@pusan.ac.kr> and Bin Zhu <bin.zhu@nih.gov>

See Also

[EstimateSigActivity](#), [ValidateSATSInputs](#)

Examples

```
data(SimData, package="SATS")

CalculateSignatureBurdens(SimData$L, SimData$TrueW_TMB, SimData$TrueH)
# For more detailed usage, please refer to README and the user manual
# in https://github.com/binzhulab/SATS/tree/main.
```

EstimateSigActivity	<i>Estimate signature activity</i>
---------------------	------------------------------------

Description

Estimate signature activities from a mutation-count matrix, a sample-level panel-context matrix, and a catalog signature-profile matrix.

Usage

```
EstimateSigActivity(V, L, W, n.start=50, iter.max=5000, eps=1e-5)
```

Arguments

V	Mutation-count matrix or data frame with samples in columns
L	Sample-level panel-context matrix or data frame with samples in columns, see GenerateLMatrix
W	Catalog signature-profile matrix or data frame with signatures in columns
n.start	Number of initializations. The default is 50.
iter.max	Maximum number of iterations in the EM algorithm. The default is 5000.
eps	Stopping tolerance in the EM algorithm. The default is 1e-5.

Details

The sample-level panel-context matrix L and mutation-count matrix V are of size P (the mutation context) by N (the sample size). The catalog signature-profile matrix W has dimension P by K (the number of signatures). For single base substitutions (SBS), P is 96. For the objects V, L, and W, we must have $\dim(V) = \dim(L)$ and $\text{ncol}(W) = K$, where K is the number of signatures. The input matrices must contain numeric, finite and non-negative values, and V must contain integer-like mutation counts. When row names and column names are present, `EstimateSigActivity()` checks that V, L and W contain matched mutation contexts and matched sample IDs. If the same names are present in a different order, L and W are reordered to match V. If one object has names on an axis used for alignment and the paired object does not, or if the named entries differ between matrices, the function stops with an informative error. Users can run [ValidateSATSInputs](#) directly to check and align inputs before calling this function. `EstimateSigActivity()` uses an EM algorithm to estimate the signature activity matrix. The arguments `n.start`, `iter.max` and `eps` control the number of random starts, maximum number of EM iterations and stopping tolerance, respectively. Because convergence to a local optimum can occur, the default uses multiple initial values (`n.start = 50`). For each initial value, the default maximum number of EM iterations is `iter.max = 5000`, and the stopping tolerance is `eps = 1e-5`. For the catalog signature profile matrix W, reference SBS TMB signature profiles in `data(SimData)` can be used.

Value

A list containing the estimated activity matrix H, the log-likelihood loglike, and the logical value converged.

Author(s)

Donghyuk Lee <dhyuklee@pusan.ac.kr> and Bin Zhu <bin.zhu@nih.gov>

See Also

[CalculateSignatureBurdens](#), [ValidateSATsInputs](#)

Examples

```
data(SimData, package="SATS")

EstimateSigActivity(SimData$V, SimData$L, SimData$TrueW_TMB)

# For more detailed usage, please refer to README and the user manual
# in https://github.com/binzhulab/SATS/tree/main.
```

GenerateLMatrix	<i>Generate a sample-level panel-context matrix</i>
-----------------	---

Description

Generate the sample-level L matrix from panel-context or panel-coordinate information and sample-panel annotations.

Usage

```
GenerateLMatrix(Panel_context, Patient_Info, Class = c("SBS", "DBS"),
  SBS_order = c("COSMIC", "signer"), ref.genome = "hg19")
```

Arguments

Panel_context	Either a panel-context matrix returned from GeneratePanelSize or a panel-coordinate data frame containing Chromosome, Start_Position, End_Position and SEQ_ASSAY_ID.
Patient_Info	A sample-panel annotation data frame containing SEQ_ASSAY_ID and either PATIENT_ID or SAMPLE_ID.
Class	A character string specifying the mutation class. Used when Panel_context is a panel-coordinate data frame.
SBS_order	Mutation-type order, either "COSMIC" or "signer". Used for SBS analyses when Panel_context is a panel-coordinate data frame.
ref.genome	The reference genome, either "hg19" or "hg38". Used when Panel_context is a panel-coordinate data frame.

Details

`GenerateLMatrix()` links each sample to its specific sequencing assay, identified by `SEQ_ASSAY_ID`. For the standard preprocessing workflow, provide panel-coordinate information as the first argument and sample-panel annotation as the second argument. The function then calculates panel-level mutation-context opportunity counts internally and expands them to a sample-level L matrix.

For backward compatibility, the first argument can also be a panel-context matrix returned by [GeneratePanelSize](#). In that lower-level workflow, `GenerateLMatrix()` only expands the provided panel-context matrix to the sample level.

BED target-region files can first be converted into SATS-compatible panel-coordinate tables with [ReadBEDAsPanelInfo](#).

Value

A data frame (L matrix) of P by N, where P is the number of mutation channels (96 for SBS or 78 for DBS) and N is the number of tumors.

Author(s)

Donghyuk Lee <dhyuklee@pusan.ac.kr> and Bin Zhu <bin.zhu@nih.gov>

See Also

[GeneratePanelSize](#), [GenerateVMatrix](#), [ReadBEDAsPanelInfo](#)

Examples

```
data(SimData, package="SATS")

keep <- match(SimData$PatientInfo$SEQ_ASSAY_ID,
             unique(SimData$PanelEx$SEQ_ASSAY_ID), nomatch=0) > 0
PatientInfo <- SimData$PatientInfo[keep, ]

L_mat <- GenerateLMatrix(SimData$PanelEx, PatientInfo, Class="SBS",
                       SBS_order="COSMIC", ref.genome="hg19")

Panel_context <- GeneratePanelSize(genomic_information=SimData$PanelEx,
                                   Class="SBS", SBS_order="COSMIC")
L_mat2 <- GenerateLMatrix(Panel_context, PatientInfo)

stopifnot(identical(colnames(L_mat), colnames(L_mat2)))
stopifnot(identical(rownames(L_mat), rownames(L_mat2)))
# For more detailed usage, please refer to README and the user manual
# in https://github.com/binzhulab/SATS/tree/main.
```

GeneratePanelSize	<i>Generate a panel-context matrix</i>
-------------------	--

Description

Generate a panel-context matrix from panel-coordinate information.

Usage

```
GeneratePanelSize(genomic_information, Class = c("SBS", "DBS"),
  SBS_order = c("COSMIC", "signer"), ref.genome="hg19")
```

Arguments

genomic_information	Data frame of panel information (see details).
Class	A character string specifying the mutation class. Must be either "SBS" (for single base substitutions) or "DBS" (for double base substitutions). If "DBS" is selected, the SBS_order argument is ignored.
SBS_order	Mutation-type order, either "COSMIC" or "signer" (see details).
ref.genome	The reference genome, either "hg19" or "hg38".

Details

The first argument `genomic_information` must contain the columns `Chromosome`, `Start_Position`, `End_Position` and `SEQ_ASSAY_ID`. The `Chromosome` column identifies the chromosome, `Start_Position` and `End_Position` define the targeted interval, and `SEQ_ASSAY_ID` identifies the sequencing assay or panel. These column names must be provided exactly. BED target-region files can first be converted into this format with [ReadBEDAsPanelInfo](#).

For SBS analyses, `SBS_order = "COSMIC"` returns the COSMIC-style SBS96 mutation-channel order and `SBS_order = "signer"` returns the order used by the `signer` package. This argument controls the row order of the returned mutation-opportunity matrix only; the COSMIC reference-signature version used for mapping is controlled separately by [MappingSignature](#).

Note: The result of `GeneratePanelSize()` may not be an L matrix. For the standard preprocessing workflow, users can call [GenerateLMatrix](#) directly with panel-coordinate and sample-panel annotation data. `GeneratePanelSize()` is retained as a lower-level helper for users who want to inspect or reuse panel-level context counts. The resulting panel-level matrix can be used as the opportunity matrix for `signer()` or expanded to a sample-level L matrix with [GenerateLMatrix](#). The mutation-channel row order must match the row order of the mutation-count matrix `V`.

Value

A data frame of `P` by `S`, where `P` is the number of mutation channels (96 for SBS or 78 for DBS) and `S` is the number of sequencing panels (`SEQ_ASSAY_ID`). Entries denote mutation opportunities per million base pairs for each panel.

Author(s)

Donghyuk Lee <dhyuklee@pusan.ac.kr> and Bin Zhu <bin.zhu@nih.gov>

See Also

[GenerateLMatrix](#), [GenerateVMatrix](#), [ReadBEDAsPanelInfo](#), [EstimateSigActivity](#)

Examples

```
data(SimData, package="SATS")
keep <- match(SimData$PatientInfo$SEQ_ASSAY_ID,
              unique(SimData$PanelEx$SEQ_ASSAY_ID), nomatch=0) > 0
PatientInfo <- SimData$PatientInfo[keep, ]

## For the single base substitution (SBS) case:
Panel_context1 <- GeneratePanelSize(genomic_information = SimData$PanelEx, Class = "SBS",
                                   SBS_order = "COSMIC", ref.genome="hg19")
L_mat1 <- GenerateLMatrix(Panel_context1, PatientInfo)
Panel_context2 <- GeneratePanelSize(genomic_information = SimData$PanelEx, Class = "SBS",
                                   SBS_order = "signeR", ref.genome="hg19")
L_mat2 <- GenerateLMatrix(Panel_context2, PatientInfo)

## For the double base substitution (DBS) case:
Panel_context_DBS <- GeneratePanelSize(genomic_information = SimData$PanelEx, Class = "DBS",
                                       ref.genome = "hg19")
L_mat_DBS <- GenerateLMatrix(Panel_context_DBS, PatientInfo)

# For more detailed usage, please refer to README and the user manual
# in https://github.com/binzhulab/SATS/tree/main
```

GenerateVMatrix

Generate a Mutation Count Matrix from Mutation Records

Description

Generate an SBS96 or DBS78 mutation count matrix from a MAF-like mutation record table.

Usage

```
GenerateVMatrix(mutation_record, Class = c("SBS", "DBS"),
               ref.genome = "hg19", mutation_order = NULL)
```

Arguments

mutation_record	A MAF-like data frame containing mutation records.
Class	A character string specifying the mutation class. Must be either "SBS" for single base substitutions or "DBS" for double base substitutions.

ref.genome The reference genome, either "hg19" or "hg38".

mutation_order Optional character vector specifying the row order of the returned mutation count matrix. If NULL, SATS uses the row order of RefTMB\$TMB_SBS_v3.4 for SBS or RefTMB\$TMB_DBS_v3.4 for DBS.

Details

The mutation_record data frame must contain the columns Chromosome, Start_Position, End_Position, Variant_Type, Reference_Allele, Tumor_Seq_Allele2 and Tumor_Sample_Barcode. SBS analyses use records with Variant_Type == "SNP" and generate COSMIC-style 96-channel mutation contexts. DBS analyses use records with Variant_Type == "DNP" and generate 78 double-base substitution contexts.

This function is intended for MAF-like tabular mutation records. Simple single-sample VCF files can first be converted into this tabular format with [ReadVCFAsMutationRecord](#).

Value

A numeric matrix with mutation contexts in rows and tumor samples in columns.

Author(s)

Donghyuk Lee <dhyuklee@pusan.ac.kr> and Bin Zhu <bin.zhu@nih.gov>

See Also

[GenerateLMatrix](#), [GeneratePanelSize](#), [ReadVCFAsMutationRecord](#)

Examples

```
dir <- system.file("extdata", "refitting_examples", package = "SATS")
sbs_file <- file.path(dir, "SBS_MAF_two_samples.txt")
sbs_mut <- read.table(sbs_file, header = TRUE, sep = "\t", quote = "",
                     stringsAsFactors = FALSE)

V_sbs <- GenerateVMatrix(sbs_mut, Class = "SBS", ref.genome = "hg19")
dim(V_sbs)
```

MappingSignature

Find a subset of TMB-based catalog SBS signatures

Description

This function finds a subset of TMB-based catalog SBS signatures whose linear combinations approximate de novo SBS signatures detected by `signeR`.

Usage

```
MappingSignature(W_hat, W_ref=NULL, niter=100, cutoff.I2=0.1, min.repeats=80,
                 COSMICv="v3.4")
```

Arguments

<code>W_hat</code>	Matrix or data frame of de novo signatures from <code>signeR</code>
<code>W_ref</code>	NULL or a matrix or data frame of TMB-based catalog signatures. If NULL, then it will default to <code>RefTMB\$TMB_SBS_v3.2</code> or <code>RefTMB\$TMB_SBS_v3.4</code> depending on the value of <code>COSMICv</code> below (see RefTMB).
<code>niter</code>	Number of iterations. The default is 100.
<code>cutoff.I2</code>	Coefficient cutoff used to select reference signatures. The default is 0.1, requiring a retained reference signature to contribute more than 10 percent to a de novo profile in a repeated fit.
<code>min.repeats</code>	Minimum number of repeated fits in which a reference signature must satisfy <code>coeff > cutoff.I2</code> . The default is 80.
<code>COSMICv</code>	Version of the TMB-based COSMIC signatures ("v3.2" or "v3.4"). This option is ignored if <code>W_ref</code> is not NULL. The default is "v3.4".

Details

`MappingSignature()` applies penalized non-negative least squares (pNNLS) for selecting the TMB-based catalog signatures. Specifically, it repeats pNNLS 100 times (`niter`) to reduce the randomness of cross-validation involved in pNNLS. Then TMB-based catalog signatures are selected with a coefficient greater than `cutoff.I2` in at least `min.repeats` repeats. With the default settings, a retained signature must have a coefficient greater than 0.1 in at least 80 of 100 repeated fits.

Value

A data frame with one row per retained reference signature. The `Reference` column contains the corresponding column names of `W_ref` (COSMIC SBS names when the default COSMIC reference matrix is used), and `freq` gives the number of repeated fits in which the signature coefficient exceeded `cutoff.I2`.

Author(s)

Donghyuk Lee <dhyuklee@pusan.ac.kr> and Bin Zhu <bin.zhu@nih.gov>

ReadBEDAsPanelInfo *Convert BED Target Regions to SATS Panel Information*

Description

Convert a Browser Extensible Data (BED) target-region file into the SATS-compatible panel-coordinate table used by [GeneratePanelSize](#) and [GenerateLMatrix](#).

Usage

```
ReadBEDAsPanelInfo(bed_file, seq_assay_id = NULL, has_header = FALSE,
                   chromosome_col = 1, start_col = 2, end_col = 3,
                   seq_assay_col = NULL, name_col = NULL)
```

Arguments

<code>bed_file</code>	Path to a BED file. Files ending in <code>.gz</code> are read with <code>gzfile</code> .
<code>seq_assay_id</code>	Sequencing assay identifier to assign to every interval when <code>seq_assay_col</code> is not supplied.
<code>has_header</code>	Logical value indicating whether the BED file has a header row.
<code>chromosome_col</code>	Column index or name for chromosome.
<code>start_col</code>	Column index or name for the BED start coordinate.
<code>end_col</code>	Column index or name for the BED end coordinate.
<code>seq_assay_col</code>	Optional column index or name containing <code>SEQ_ASSAY_ID</code> .
<code>name_col</code>	Optional column index or name to preserve as <code>Hugo_Symbol</code> .

Details

BED coordinates are 0-based and half-open. SATS panel-coordinate tables use 1-based inclusive coordinates. Therefore this function returns `Start_Position = BED_start + 1` and `End_Position = BED_end`.

The returned data frame contains at least `Chromosome`, `Start_Position`, `End_Position` and `SEQ_ASSAY_ID`; these columns can be passed directly to `GeneratePanelSize` or `GenerateLMatrix`.

Value

A data frame of panel target intervals in SATS coordinate format.

Author(s)

Donghyuk Lee <dhyuklee@pusan.ac.kr> and Bin Zhu <bin.zhu@nih.gov>

See Also

[GeneratePanelSize](#), [GenerateLMatrix](#), [ReadVCFAsMutationRecord](#)

Examples

```
dir <- system.file("extdata", "refitting_examples", package = "SATS")
bed_file <- file.path(dir, "SATS_example_panel.bed")

panel_info <- ReadBEDAsPanelInfo(
  bed_file = bed_file,
  seq_assay_id = "SATS_EXAMPLE_PANEL",
  name_col = 4
)
head(panel_info)
```

ReadVCFAsMutationRecord

Convert a Simple VCF File to a SATS Mutation Record Table

Description

Convert a simple single-sample Variant Call Format (VCF) file into the MAF-like mutation record table used by [GenerateVMatrix](#).

Usage

```
ReadVCFAsMutationRecord(vcf_file, sample_id = NULL,
                        keep_filter = c("PASS", "."))
```

Arguments

vcf_file	Path to a VCF file. Files ending in .gz are read with gzfile.
sample_id	Optional sample identifier for the returned Tumor_Sample_Barcode column. If NULL, SATS uses the single VCF sample column when exactly one is present, or derives a sample identifier from the file name when the VCF has no sample column. Multi-sample VCF genotype parsing is not supported by this lightweight converter.
keep_filter	Character vector of FILTER values to retain. Use NULL to retain all records.

Details

The returned data frame contains the columns required by [GenerateVMatrix](#): Chromosome, Start_Position, End_Position, Variant_Type, Reference_Allele, Tumor_Seq_Allele2 and Tumor_Sample_Barcode.

This function is a lightweight input-preparation utility for standard single-sample targeted-panel VCF files. It does not replace a full clinical variant-normalization pipeline. Multiallelic records are split by ALT allele, but complex normalization, multi-sample genotype parsing, tumor-normal interpretation, phasing and decomposition of complex events should be performed upstream when needed. [GenerateVMatrix](#) uses Variant_Type == "SNP" for SBS analyses and Variant_Type == "DNP" for DBS analyses.

Value

A MAF-like data frame suitable for [GenerateVMatrix](#).

Author(s)

Donghyuk Lee <dhyuklee@pusan.ac.kr> and Bin Zhu <bin.zhu@nih.gov>

See Also

[GenerateVMatrix](#), [ReadBEDAsPanelInfo](#)

Examples

```

dir <- system.file("extdata", "refitting_examples", package = "SATS")
vcf_file <- file.path(dir, "SBS_two_variants.vcf")

mutation_record <- ReadVCFAsMutationRecord(vcf_file)
V <- GenerateVMatrix(mutation_record, Class = "SBS", ref.genome = "hg19")
dim(V)

```

RefTMB

*Example Data***Description**

The pan-cancer repertoire of reference signatures and the reference TMB (Tumor mutation burden) signature profiles of SBS (Single base substitutions) and DBS (Double base substitutions).

Details

This file consists of the list RefTMB with the following objects:

- TMB_SBS_v3.2 : The version 3.2 reference SBS TMB signature profiles of size 96 by 76.
- TMB_DBS_v3.2 : The version 3.2 reference DBS TMB signature profiles of size 78 by 11.
- TMB_SBS_v3.4 : The version 3.4 reference SBS TMB signature profiles of size 96 by 85.
- TMB_DBS_v3.4 : The version 3.4 reference DBS TMB signature profiles of size 78 by 20.
- SBS_refSigs : The list of cancer specific SBS signature names.
- DBS_refSigs : The list of cancer specific DBS signature names.

SimData

*Simulated example data***Description**

Simulated data used in SATS examples and tests.

Details

This file consists of the list SimData with the following objects:

- V : Simulated mutation-count matrix of size 96 by 10027 generated from a Poisson distribution with mean corresponding to each element of $L \ast (WH)$, where $\ast$ denotes elementwise multiplication, L is the sample-level panel-context matrix below, W is the tumor mutation burden-normalized signature-profile matrix (TrueW_TMB below), and H is the signature-activity matrix (TrueH below).
- L : Sample-level panel-context matrix of size 96 by 10027.

- TrueH : Simulated signature-activity matrix of size 6 by 10027 used to generate V.
- TrueW_TMB : Tumor mutation burden-normalized signature-profile matrix of size 96 by 6 for 6 SBS signatures (SBS1, SBS2/13, SBS4, SBS5, SBS40, SBS89) used to generate V.
- SingleTumorEx : A single simulated tumor (column singleV) and its sequencing context (column singleL).
- PanelEx: An example sequencing panel information with Chromosome, Start_Position, End_Position, SEQ_ASSAY_ID.
- PatientInfo: An example patient data set containing patient IDs associated with SEQ_ASSAY_ID.

ValidateSATSInputs	<i>Validate SATS input matrices</i>
--------------------	-------------------------------------

Description

Validate and align SATS mutation-count, panel-context, signature-profile and signature-activity matrices before running refitting or burden calculation.

Usage

```
ValidateSATSInputs(V=NULL, L=NULL, W=NULL, H=NULL)
```

Arguments

V	Optional mutation-count matrix or data frame with mutation contexts in rows and samples in columns. Values must be non-negative, finite and integer-like.
L	Optional panel-context matrix or data frame with mutation contexts in rows and samples in columns. Values must be non-negative and finite.
W	Optional catalog signature-profile matrix or data frame with mutation contexts in rows and signatures in columns. Values must be non-negative and finite.
H	Optional signature-activity matrix or data frame with signatures in rows and samples in columns. Values must be non-negative and finite.

Details

ValidateSATSInputs() performs the same matrix-level checks used internally by [EstimateSigActivity](#) and [CalculateSignatureBurdens](#). It checks object type, matrix dimensions, numeric values, finite values, non-negative values, duplicated or missing row and column identifiers, mutation-context alignment, sample-ID alignment and signature-name alignment.

When compatible names are present in a different order, the returned matrices are reordered to a common order. For example, L and W are reordered to match V for activity estimation, and L and W are reordered to match H for burden calculation. If one object has names on an axis used for alignment and the paired object does not, or if the named entries differ, the function stops with an informative error. If paired axes are both unnamed, validation proceeds by dimension.

Value

A named list containing the supplied matrices after compatible named axes have been aligned. Objects that were not supplied are omitted from the returned list.

Author(s)

Donghyuk Lee <dhyuklee@pusan.ac.kr> and Bin Zhu <bin.zhu@nih.gov>

See Also

[EstimateSigActivity](#), [CalculateSignatureBurdens](#)

Examples

```
data(SimData, package="SATS")

obj <- ValidateSATSEInputs(
  V = SimData$V[, 1:3],
  L = SimData$L[, c(3, 1, 2)],
  W = SimData$TrueW_TMB
)
identical(colnames(obj$L), colnames(obj$V))
```

Index

- * **data**

- RefTMB, [13](#)

- SimData, [13](#)

- * **mutational signatures**

- CalculateSignatureBurdens, [3](#)

- EstimateSigActivity, [4](#)

- ValidateSATSInputs, [14](#)

- * **package**

- SATS-package, [2](#)

CalculateSignatureBurdens, [2](#), [3](#), [5](#), [14](#), [15](#)

EstimateSigActivity, [2](#), [3](#), [4](#), [8](#), [14](#), [15](#)

GenerateLMatrix, [2–4](#), [5](#), [7–11](#)

GeneratePanelSize, [5](#), [6](#), [7](#), [9–11](#)

GenerateVMatrix, [2](#), [6](#), [8](#), [8](#), [12](#)

MappingSignature, [2](#), [7](#), [9](#)

ReadBEDAsPanelInfo, [6–8](#), [10](#), [12](#)

ReadVCFAsMutationRecord, [9](#), [11](#), [12](#)

RefTMB, [10](#), [13](#)

SATS (SATS-package), [2](#)

SATS-package, [2](#)

SimData, [13](#)

ValidateSATSInputs, [2–5](#), [14](#)