Package 'SynSigEval'

December 20, 2021

Type Package

Title Evaluate Results of Mutational Signature Analysis Software
on Synthetic Spectra Created by Package SynSigGen

Version 0.3.1

Author Steven G. Rozen, Yang Wu

Maintainer Steven G. Rozen <steverozen@gmail.com>

Description
Examine and evaluate the output of mutational signature analysis computational approaches.

License GPL-3

Language en-US

Encoding UTF-8

LazyData true

Imports data.table,
ICAMS,
ICAMSxtra (>= 0.0.8),
stats,
devtools,
utils,
rlang,
fs,
graphics,
grDevices,
ggplot2,
ggpubr,
ggbeeswarm,
gtools

Remotes github::steverozen/ICAMSxtra

Depends R (>= 3.5)

RoxygenNote 7.1.2

Suggests testthat,
knitr,
lsa,
rmarkdown,
DelayedArray,
BiocManager

R **topics documented:**

CopyBestSignatureAnalyzerResult

> *Find the SignatureAnalyzer results directory with the best results and*
> *make a copy of it as* sa.results.dir/best.run/

Description

Find the SignatureAnalyzer results directory with the best results and make a copy of it as sa.results.dir/best.run/

Usage

```
CopyBestSignatureAnalyzerResult(
  sa.results.dir,
  verbose = FALSE,
  overwrite = FALSE
)
```

Arguments

sa.results.dir	See BestSignatureAnalyzerResult
verbose	See BestSignatureAnalyzerResult
overwrite	If TRUE overwrite existing "best.run"

Value

The path of the best directory that was copied as a string, with the list directories examined as the attribute `run.directories`.

CreateEMuOutput *Prepare input file for EMu from a EMu formatted catalog file.*

Description

Prepare input file for EMu from a EMu formatted catalog file.

Usage

```
CreateEMuOutput(
  catalog,
  out.dir = paste0(dirname(catalog), "/ExtrAttr/EMu.results"),
  overwrite = FALSE
)
```

Arguments

catalog	a catalog in ICAMS format. It can be a .csv file, or a matrix or data.frame. Usually, it refers to `"ground.truth.syn.catalog.csv"`.
out.dir	Directory that will be created for the output; abort if it already exists. Usually, the `out.dir` will be a `EMu.results` folder directly under the folder storing `catalog`.
overwrite	If TRUE, overwrite existing output

Details

Creates folder named `EMu.results` containing catalogs in EMu-formatted catalogs: Rows are signatures; the first column is the name of the mutation type, while the remaining columns are samples (tumors). These EMu-formatted catalogs will the input when running EMu program later on compiled binary.

Value

`invisible(catalog)`, original catalog in EMu format

CreatehelmsmanOutput   *Prepare input file for helmsman from a helmsman formatted catalog file.*

Description

Prepare input file for helmsman from a helmsman formatted catalog file.

Usage

```
CreatehelmsmanOutput(
  catalog,
  out.dir = paste0(dirname(catalog), "/ExtrAttr/helmsman.results"),
  overwrite = FALSE
)
```

Arguments

catalog	a catalog in ICAMS format. It can be a .csv file, or a matrix or data.frame. Usually, it refers to "ground.truth.syn.catalog.csv".
out.dir	Directory that will be created for the output; abort if it already exists. Usually, the out.dir will be a helmsman.results folder directly under the folder storing catalog.
overwrite	If TRUE, overwrite existing output

Details

Creates folder named helmsman.results containing catalogs in helmsman-formatted catalogs: Rows are signatures; the first column is the name of the mutation type, while the remaining columns are samples (tumors). These helmsman-formatted catalogs will the input when running helmsman program later on Python platform.

Value

invisible(catMatrix), original catalog in helmsman format

CreateMultiModalMuSigOutput

                   Prepare input file for MultiModalMuSig from a MultiModalMuSig formatted catalog file.

Description

Prepare input file for MultiModalMuSig from a MultiModalMuSig formatted catalog file.

Usage

```
CreateMultiModalMuSigOutput(
  catalog,
  out.dir = paste0(dirname(catalog), "/ExtrAttr/MultiModalMuSig.results"),
  overwrite = FALSE
)
```

Arguments

catalog	a catalog in ICAMS format. It can be a .csv file, or a matrix or data.frame. Usually, it refers to "ground.truth.syn.catalog.csv".
out.dir	Directory that will be created for the output; abort if it already exists. Usually, the out.dir will be a MultiModalMuSig.results folder directly under the folder storing catalog.
overwrite	If TRUE, overwrite existing output

Details

Creates folder named MultiModalMuSig.results containing catalogs in MultiModalMuSig-formatted catalogs: Rows are signatures; the first column is the name of the mutation type, while the remaining columns are samples (tumors). These MM-formatted catalogs will the input when running MultiModalMuSig program later on Julia platform.

Value

invisible(catMatrix), original catalog in MultiModalMuSig format

---

helmsmanCatalog2ICAMS *Read Catalog files or matrices in helmsman format.*

Description

Read Catalog files or matrices in helmsman format.

Usage

```
helmsmanCatalog2ICAMS(
  cat,
  region = "unknown",
  catalog.type = "counts.signature"
)
```

Arguments

cat	Input catalog, can be a tab-delimited text file in helmsman format, or a matrix/data.frame object.
region	Catalog region. Can be a specific genomic or exomic region, or "unknown". Default: "unknown"
catalog.type	Is the catalog a signature catalog, or a spectrum catalog? Default: "counts.signature"

Value

a catalog matrix in ICAMS format.

ICAMSCatalog2EMu   *Convert Catalogs from ICAMS format to EMu format*

Description

Convert Catalogs from ICAMS format to EMu format

Usage

```
ICAMSCatalog2EMu(catalog)
```

Arguments

catalog   A catalog matrix in ICAMS format. (SNS only!)

Value

a matrix without any dimnames, but the values are the transposition of the values in catalog.

ICAMSCatalog2helmsman *Convert Catalogs from ICAMS format to helmsman format*

Description

Convert Catalogs from ICAMS format to helmsman format

Usage

```
ICAMSCatalog2helmsman(catalog, type = "spectra")
```

Arguments

catalog   A catalog matrix in ICAMS format. (SNS only!)

type   Whether it is a spectra catalog ("spectra") or a signature catalog ("signature").

Value

a catalog matrix in helmsman format.

ICAMSCatalog2MM	*Convert Catalogs from ICAMS format to MM format*

Description

Convert Catalogs from ICAMS format to MM format

Usage

```
ICAMSCatalog2MM(catalog)
```

Arguments

catalog	A catalog matrix in ICAMS format. (SNS/DNS/ID)

Value

a catalog matrix in MultiModalMuSig format.

MMCatalog2ICAMS	*Convert Catalogs (File or Matrix) from MM format to ICAMS format*

Description

Convert Catalogs (File or Matrix) from MM format to ICAMS format

Usage

```
MMCatalog2ICAMS(cat, region = "unknown", catalog.type = "counts.signature")
```

Arguments

cat	Input catalog, can be a tab-delimited file or matrix in MultiModalMuSig format.
region	Catalog region. Can be a specific genomic or exomic region, or "unknown". Default: "unknown"
catalog.type	Is the catalog a signature catalog, or a spectrum catalog? Default: "counts.signature"

Value

a catalog matrix in ICAMS format.

PlotCatCOMPOSITE   *Plot the a SignatureAnalyzer COMPOSITE signature or catalog into separate pdfs*

Description

Plot the a SignatureAnalyzer COMPOSITE signature or catalog into separate pdfs

Usage

```
PlotCatCOMPOSITE(catalog, filename.header, type, id = colnames(catalog))
```

Arguments

catalog   Catalog or signature matrix

filename.header

                Contain path and the beginning part of the file name. The name of the pdf files will be: filename.header.SNS.96.pdf filename.header.SNS.1536.pdf filename.header.DNS.78.pdf filename.header.ID.83.pdf

type   See PlotCatalogToPdf.

id   A vector containing the identifiers of the samples or signatures in catalog.

ReadAndAnalyzeExposures

                Assess how well inferred exposures match input exposures We assume that in many cases attribution programs will be run outside of R on file inputs and will generate fill outputs.

Description

Assess how well inferred exposures match input exposures

We assume that in many cases attribution programs will be run outside of R on file inputs and will generate fill outputs.

Usage

```
ReadAndAnalyzeExposures(
  extracted.sigs,
  ground.truth.sigs,
  inferred.exp.path,
  ground.truth.exposures
)
```

Arguments

extracted.sigs Path to file containing the extracted signature profiles.

ground.truth.sigs

  File containing signature profiles from which the synthetic data were generated.

inferred.exp.path

  File containing mutation counts (exposures) of synthetic tumors which are inferred to extracted or input signatures.

ground.truth.exposures

  File containing the exposures from which the synthetic catalogs were generated. This file is used to restrict assessment of signature exposures to only those signatures in ground.truth.sigs that were actually represented in the exposures.

Details

Generates output files by calling [MatchSigsAndRelabel](#)

Value

A [data.frame](#) recording:

Ground.truth.exposure: sum of ground truth exposures of all tumors to all ground-truth signatures.

Inferred.exposure: sum of inferred exposures of all tumors to all ground-truth signatures. Here, inferred exposure of a tumor to a ground-truth signature equals to the sum of the exposures of this tumor to all extracted signatures which are most similar to a ground-truth signature. If there is no extracted signature resembling an ground-truth signature, the inferred exposure of this ground-truth signature will be 0.

Absolute.difference: sum of absolute difference between ground-truth exposure and inferred exposure of all tumors to all ground-truth signatures.

ReadAndAnalyzeSigs   *Assess how well extracted signatures match input signatures We assume that in many cases extraction programs will be run outside of R on file inputs and will generate fill outputs.*

Description

Assess how well extracted signatures match input signatures

We assume that in many cases extraction programs will be run outside of R on file inputs and will generate fill outputs.

Usage

```
ReadAndAnalyzeSigs(extracted.sigs, ground.truth.sigs, ground.truth.exposures)
```

Arguments

extracted.sigs   Path to file containing the extracted signature profiles.

ground.truth.sigs

        File containing signature profiles from which the synthetic data were generated.

ground.truth.exposures

        File containing the exposures from which the synthetic catalogs were generated. This file is used to restrict assessment to only those signatures in ground.truth.sigs that were actually represented in the exposures.

Details

Generates output files by calling [MatchSigsAndRelabel](#)

Value

See [MatchSigsAndRelabel](#)

ReadEMuCatalog   *Read Catalog files in EMu format.*

Description

Read Catalog files in EMu format.

Usage

```
ReadEMuCatalog(
  cat,
  mutTypes,
  sigOrSampleNames,
  region = "unknown",
  catalog.type = "counts.signature"
)
```

Arguments

cat   A tab-delimited catalog text file in EMu format; or a EMu formatted matrix or data.frame.

mutTypes   Types of mutations. They are usually from an ICAMS:::catalog.row.header object.

sigOrSampleNames

        If input file is a counts signature file (catalog.type == "counts.signature"), signature names should be provided.

        If input file is a counts spectra file (catalog.type == "counts"), names of samples should be provided.

region   Catalog region. Can be a specific genomic or exomic region, or "unknown". Default: "unknown"

catalog.type   Is the catalog a signature catalog, or a spectrum catalog? Default: "counts"

Value

a catalog matrix in ICAMS format.

ReadEMuExposureFile   *Read Exposure files in EMu format.*

Description

Read Exposure files in EMu format.

Usage

```
ReadEMuExposureFile(exposureFile, sigNames, sampleNames)
```

Arguments

exposureFile   Exposure file generated by EMu. Usually, it is called "W_components.txt".

sigNames   Names of signatures. These will be served as the rownames of the exposure matrix.

sampleNames   Names of samples in exposure file.

Return ICAMS/SynSigEval formatted exposure matrix.

ReadExposureMM   *Read Catalog files in MM format*

Description

Read Catalog files in MM format

Usage

```
ReadExposureMM(exposureFile)
```

Arguments

exposureFile   Input exposure file, can be a tab-delimited text file in MultiModalMuSig format.

Value

a exposure matrix in ICAMS format.

ReadhelmsmanExposure   *Read Exposure files in helmsman format.*

Description

Read Exposure files in helmsman format.

Usage

```
ReadhelmsmanExposure(exposure, check.names = TRUE)
```

Arguments

exposure   Exposure file generated by helmsman. Usually, it is called "W_components.txt".

check.names   logical. If TRUE then the names of the variables in the data frame are checked
                to ensure that they are syntactically valid variable names. If necessary they are
                adjusted (by make.names) so that they are, and also to ensure that there are no
                duplicates.

                Return ICAMS/SynSigEval formatted exposure matrix.

ReadSigProfilerExposure
                Read a file containing exposures attributed by SigProfiler/Python

Description

Read a file containing exposures attributed by SigProfiler/Python

Usage

```
ReadSigProfilerExposure(file)
```

Arguments

file   The name of the file to read.

Value

The corresponding signature matrix in standard internal representation.

`ReadSigProfilerSigDBS78`

Read a file containing DBS78 signatures extracted by SigProfiler/Python

---

Description

Read a file containing DBS78 signatures extracted by SigProfiler/Python

Usage

```
ReadSigProfilerSigDBS78(file)
```

Arguments

file   The name of the file to read.

Value

The corresponding signature matrix in standard internal representation.

`ReadSigProfilerSigID83`

Read a file containing ID83 signatures extracted by SigProfiler/Python

---

Description

Read a file containing ID83 signatures extracted by SigProfiler/Python

Usage

```
ReadSigProfilerSigID83(file)
```

Arguments

file   The name of the file to read.

Value

The corresponding signature matrix in standard internal representation.

```
ReadSigProfilerSigSBS96
```
Read a file containing SBS96 signatures extracted by SigProfiler/Python

Description

Read a file containing SBS96 signatures extracted by SigProfiler/Python

Usage

```
ReadSigProfilerSigSBS96(file)
```

Arguments

file   The name of the file to read.

Value

The corresponding signature matrix in standard internal representation.

```
RelabelExSigs
```
Append most similar ground-truth sig and pairwise cosine similarity to the name of each extracted sig in matrix of extracted sigs.

Description

Append most similar ground-truth sig and pairwise cosine similarity to the name of each extracted sig in matrix of extracted sigs.

Usage

```
RelabelExSigs(sigAnalysis)
```

Arguments

sigAnalysis   A list returned by function ReadAndAnalyzeSigs, at least including:

                1. `ex.sigs`: matrix of extracted sigs
                2. `sim.matrix`: full matrix between extracted sigs and ground-truth sigs
                3. `table`: matrix include pairs of true positive ground-truth sigs and true positive extracted sigs.

Value

Matrix of extracted sigs, yet the names of signatures changed. New name: <old_name>(<name_of_most_similar_ground cosine_similarity) e.g., Sig.A –> Sig.A (SBS1 0.998)

SignatureAnalyzerSummarizeSBS1SBS5

Summarize all sub-directories of SignatureAnalyzer results on the cor-related SBS1 / SBS5.

Description

This is special-purpose function to summarize results from one in-silico experiment that examines how well signatures can be extracted from synthetic tumors with correlated SBS1 and SBS5.

Usage

```
SignatureAnalyzerSummarizeSBS1SBS5(
  top.level.dir,
  summarize.exp = TRUE,
  overwrite = FALSE
)
```

Arguments

top.level.dir	Path to top level directory.
summarize.exp	Whether to summarize exposures when the file specified by inferred.exp.path exists.
overwrite	If TRUE overwrite existing directories and files.

SignatureAnalyzerSummarizeTopLevel

Summarize all subdirectories of SignatureAnalyzer results on a major dataset.

Description

This function depends on a particular directory structure: see argument top.level.dir. This function finds the best of multiple SignatureAnalyzer extraction runs and summarizes the comparison of the best run with the ground truth.

Usage

```
SignatureAnalyzerSummarizeTopLevel(
  top.level.dir,
  summarize.exp = TRUE,
  overwrite = FALSE
)
```

Arguments

top.level.dir   Path to top level directory, which must contain the following subdirectories:

- sa.sa.96/sa.results/
- sp.sp/sa.results/
- sa.sa.COMPOSITE/sa.results/
- sp.sa.COMPOSITE/sa.results/

Each of the directories must contain additional subdirectories, one for each Sig-natureAnalyzer run, names sa.run.<n>, where <n> is an integer (string of dig-its).

summarize.exp   Whether to summarize exposures when the <run.dir>/<which.run>/sa.output.exp.csv exists.

overwrite   If TRUE overwrite and files in existing run.dir/summary folder.

SplitCatCOMPOSITE   *Split COMPOSITE (SNS1536+DBS78+ID83) catalogs in ICAMS for-*
                           mat into 3 individual catalogs.

Description

Split COMPOSITE (SNS1536+DBS78+ID83) catalogs in ICAMS format into 3 individual cata-logs.

Usage

```
SplitCatCOMPOSITE(catalog)
```

Arguments

catalog   Input catalog, can be a .csv file or matrix in ICAMS COMPOSITE format.

Value

a list, containing 3 catalog matrices in MultiModalMuSig format. Each matrix contains SNS1536, DBS78 and ID83 information, respectively.

SummarizeMultiRuns   *Assess/evaluate multiple summarized runs on one dataset by one com-*
                           putational approach.

Description

Summarize results from each computational approach in resultPath/run.names (generated by running a computational approach), combine them into resultPath.

Usage

```
SummarizeMultiRuns(datasetName, toolName, resultPath, run.names)
```

Arguments

datasetName   Name of the dataset. (e.g. "S.0.1.Rsq.0.1"). Usually, it is has the same name as `basename(top.dir)`.

toolName   Name of computational approach. (e.g. "SigProExtractor")

resultPath   Path expected to have multiple result folders each named as `run.names` (e.g. "seed.1"). The example `resultPath` is S.0.1.Rsq.0.1/sp.sp/ExtrAttr/hdp.results/ in old folder structure, or 3a.Original_output_K_unspecified/hdp/S.0.1.Rsq.0.1 in new folder structure.

run.names   A character vector records the list of directories which are under `resultPath` and contain results of computational approach, and a `summary` folder generated by [SummarizeSigOneExtrAttrSubdir](#).

Details

Also writes multiple files into folder `resultPath`.

Value

A list contain values of measures measures in multiple runs:

- $averCosSim Average cosine similarity. Only similarities between TP sigs and extracted sigs most similar to them.
- $truePos True Positives(TP): Ground-truth signatures which are active in the spectra, and extracted.
- $falseNeg False Negatives(FN): Ground-truth signatures not extracted.
- $falsePos False Positives(FP): Signatures wrongly extracted, not resembling any ground-truth signatures.
- $TPR True positive rate (TPR, Sensitivity): TP / (TP + FN)
- $PPV Positive predictive value (PPV, Precision): TP / (FP + TP)
- $cosSim Cosine similarity between each of the ground-truth signatures, and its most similar extracted signature.
- $AggManhattanDist (if exposures of signatures were inferred) Scaled Manhattan distance between ground-truth and inferred exposures to each of the ground-truth signatures.

This list also contains mean and sd, and other statistics of these measures in

- $fivenum - summary generated by [fivenum](#) - columns of this table refer to Tukey's five number summary for each extraction measure across all runs:
  - `min` - minimum
  - `lower-hinge` - first quartile. Serve as the lower-hinge of the box-whisker plot.
  - `median` - median of measure across all runs.
  - `upper-hinge` - third quartile. Serve as the upper-hinge of the box-whisker plot.
  - `max` - maximum
- $fivenumMD - Tukey's five number summary for aggregately-scaled Manhattan distance.
- $meanSD - mean and standard deviation for extraction measures.
- $meanSDMD - mean and standard deviation for aggregately-scaled Manhattan distance.

SummarizeMultiToolsMultiDatasets
                              Summarize results for multiple datasets, by different computational
                              approaches.

Description

Summarize results of mutational signature extraction and exposure inference by multiple computational approaches on multiple datasets. Before running this function, make sure the summary file for each single data set `toolSummaryPaths/OneToolSummary.Rda` exists.

Usage

```
SummarizeMultiToolsMultiDatasets(
  toolSummaryPaths,
  out.dir,
  display.datasetName = FALSE,
  sort.by.composite.extraction.measure = "descending",
  overwrite = FALSE
)
```

Arguments

toolSummaryPaths

> Paths of top-level dataset directories trees you want to investigate. E.g. "./S.0.1.Rsq.0.1"

> Note: `OneToolSummary.RDa` are expected to be exist under `toolSummaryPaths`.

out.dir   Path of the output directory.

display.datasetName

> Whether to put the name of spectra datasets inside of the csv outputs of summary tables.

sort.by.composite.extraction.measure

> Whether to re-order the computational approaches on violin plots, based on the mean of composite measure.

> `"descending"`: Put the computational approach with the highest mean composite measure to the left, and arrange approaches in descending order. `"ascending"`: Put the computational approach with the lowest mean composite measure to the left, and arrange approaches in ascending order. Anything else: Keep the computational approaches in a smart alphabetical order embedded with numbers, defined by mixedsort.

overwrite   Whether to overwrite the contents in out.dir if it already exists. (Default: FALSE)

Details

`OneToolSummary.Rda` is generated by SummarizeOneToolMultiDatasets).

SummarizeOneToolMultiDatasets

Combine results for multiple datasets, from one computational approaches.

Description

Summarize results from each computational approach in `toolPath/datasetNames` and combine them into `out.dir`.

Usage

```
SummarizeOneToolMultiDatasets(
  datasetNames = SynSigGen::SBS1SBS5datasetNames,
  datasetGroup,
  datasetGroupName,
  datasetSubGroup = NULL,
  datasetSubGroupName = NULL,
  toolName,
  toolPath,
  out.dir,
  display.datasetName = FALSE,
  overwrite = FALSE
)
```

Arguments

datasetNames   Names of datasets which are also folder names under `toolPath`. These folders contain results of `toolName` on such datasets. E.g. `SynSigGen::SBS1SBS5datasetNames`

datasetGroup   Numeric or character vector differentiating datasets within each group. E.g. For SBS1-SBS5 correlated datasets, we can consider the value of SBS1-SBS5 exposure ratio as the value for `datsetgroup`: `rep(c(0.1,0.5,1,2,5,10),each = 4)`

       The value is set to `Default` if unspecified.

datasetGroupName

       Meaning of all datasetGroup. E.g. For SBS1-SBS5 correlated datasets, we can consider `"SBS1-SBS5 exposure ratio"` as what `datasetGroup` is referring to.

datasetSubGroup

       Numeric or character vector differentiating datasets within each sub-group. E.g. For SBS1-SBS5 correlated datasets, we can consider the value of SBS1-SBS5 correlation as the value of subgroup: `rep(c(0.1,0.2,0.3,0.6),times = 5)`

datasetSubGroupName

       Meaning of all `datasetSubGroup`. E.g. For SBS1-SBS5 correlated datasets, we can consider `"SBS1-SBS5 correlation"` as what `datasetSubGroup` is referring to.

toolName   Name of computational approach to be investigated (e.g. "SigProExtractor")

toolPath   The path of the results of the computational approach to be investigated. May
             include top-level directory (e.g. `3a.Original_output_K_unspecified`) and
             second-level directory containing outputs and summaries of one computational
             approach to be investigated (e.g. `SigProExtractor` or `SigProExtractor.results`).

             One example: `3a.Original_output_K_unspecified/SigProExtractor`

             Note: this function expects file `multiRun.RDa` generated by `SummarizeMultiRuns`
             under `toolPath/datasetNames`

out.dir   Path of the output directory.

display.datasetName
             Whether to put the name of spectra datasets inside of the csv outputs of summary
             tables.

overwrite   Whether to overwrite the contents in out.dir if it already exists. (Default: FALSE)

SummarizeSigOneAttrSubdir

*Assess/evaluate results from packages which can ONLY do exposure
attribution.*

Description

Packages including but not limited to: deconstructSigs, YAPSA.

Usage

```
SummarizeSigOneAttrSubdir(
  run.dir,
  ground.truth.exposure.dir = paste0(run.dir, "/../../../"),
  overwrite = FALSE,
  summary.folder.name = "summary",
  export.Manhattan.each.spectrum = FALSE
)
```

Arguments

run.dir   Lowest level path to results, e.g. `<top.dir>/sa.sa.96/Attr/YAPSA.results/seed.1/`
             Here, `<top.dir>` refers to a top-level directory which contains the full informa-
             tion of a synthetic dataset. (e.g. `syn.2.7a.7b.abst.v8`) This code depends on
             a conventional directory structure documented elsewhere. For packages which
             can do both extraction and attribution, we expect two files, `ground.truth.signatures.csv`
             and `inferred.exposures.csv` are in the folder.

ground.truth.exposure.dir
             Folder which stores ground-truth exposures. It defaults to be sub.dir, i.e.
             run.dir/../../

overwrite   If TRUE overwrite existing directories and files.

summary.folder.name
             The name of the folder containing summary results. Usually, it equals to "sum-
             mary".

export.Manhattan.each.spectrum
             Whether to export csv files for Manhattan distance of each mutational spectrum.

Details

Here, we excluded SignatureEstimation. Although it is also a package with only attribution, but it has two attribution algorithms. Therefore the naming of the results are slightly different from the other two packages.

SummarizeSigOneExtrAttrSubdir

Assess/evaluate results from packages which can do BOTH extraction and attribution, excluding SigProfiler-Python and SignatureAnalyzer.

Description

Packages including but not limited to: hdp, MutationalPatterns, sigfit, signeR, SomaticSignatures.

Usage

```
SummarizeSigOneExtrAttrSubdir(
  run.dir,
  ground.truth.exposure.dir = paste0(run.dir, "/../../../"),
  summarize.exp = TRUE,
  overwrite = FALSE,
  summary.folder.name = "summary",
  export.Manhattan.each.spectrum = FALSE
)
```

Arguments

run.dir   A directory which contains output of computational approach in one run on a
                    specific dataset, possibly with a specified seed. E.g. 2b.Full_output_K_as_2/hdp.results/S.0.1

                    This code depends on a conventional directory structure documented in NEWS.md.

ground.truth.exposure.dir
                    Folder which stores ground-truth exposures. Should contain a file named ground.truth.syn.exposu
                    In PCAWG paper: run.dir/../../ In SBS1-SBS5 paper: 0.Input_datasets/S.0.1.Rsq.0.1/

summarize.exp   Whether to summarize exposures when the file specified by inferred.exp.path
                    exists.

overwrite   If TRUE overwrite and files in existing run.dir/summary folder.

summary.folder.name
                    The name of the folder containing summary results. Usually, it equals to "sum-
                    mary".

export.Manhattan.each.spectrum
                    Whether to export csv files for Manhattan distance of each mutational spectrum.

SummarizeSigOnehelmsmanSubdir

Assess/evaluate results from helmsman.NMF

Description

Assess/evaluate results from helmsman.NMF

Usage

```
SummarizeSigOnehelmsmanSubdir(
  run.dir,
  ground.truth.exposure.dir = paste0(run.dir, "/../../../"),
  summarize.exp = TRUE,
  overwrite = FALSE
)
```

Arguments

run.dir   A directory which contains output of helmsman.NMF in one run on a specific
                    dataset, possibly with a specified seed. E.g. 2b.Full_output_K_as_2/helmsman.NMF.results/S.0
                    This code depends on a conventional directory structure documented in NEWS.md.

ground.truth.exposure.dir
                    Folder which stores ground-truth exposures. Should contain a file named ground.truth.syn.exposu
                    In PCAWG paper: run.dir/../../ In SBS1-SBS5 paper: 0.Input_datasets/S.0.1.Rsq.0.1/

summarize.exp   Whether to summarize exposures when the file specified by inferred.exp.path
                    exists.

overwrite   If TRUE overwrite and files in existing run.dir/summary folder.

SummarizeSigOneSigProExtractorSubdir

Assess/evaluate results from SigProExtractor (v0.0.5.45+)

Description

SigProfiler-python de novo extraction and attribution package. Assessment is restricted to v0.0.5.43
~ v0.0.5.77, because different version has different folder structure.

Usage

```
SummarizeSigOneSigProExtractorSubdir(
  run.dir,
  ground.truth.exposure.dir = paste0(run.dir, "/../../../"),
  summarize.exp = TRUE,
  overwrite = FALSE,
  hierarchy = FALSE,
  summary.folder.name = "summary",
  export.Manhattan.each.spectrum = FALSE
)
```

Arguments

run.dir	A directory which contains output of SigProExtractor in one run on a specific dataset, possibly with a specified seed. E.g. 2b.Full_output_K_as_2/SigProExtractor.results/
	This code depends on a conventional directory structure documented in NEWS.md.
ground.truth.exposure.dir	
	Folder which stores ground-truth exposures. Should contain a file named ground.truth.syn.exposu
	In PCAWG paper: run.dir/../../ In SBS1-SBS5 paper: 0.Input_datasets/S.0.1.Rsq.0.1/
summarize.exp	Whether to summarize exposures when the file specified by inferred.exp.path exists.
overwrite	If TRUE overwrite and files in existing run.dir/summary folder.
hierarchy	Whether the user have enabled hierarchy = True when running SigProExtractor. specifying True or False into SigProExtractor will cause the program to generate different folder structure.
summary.folder.name	
	The name of the folder containing summary results. Usually, it equals to "summary".
export.Manhattan.each.spectrum	
	Whether to export csv files for Manhattan distance of each mutational spectrum.

Details

This function cannot be used on new SigProfilerExtractor (v1+) as the folder structure has been changed markedly

---

SummarizeSigOneSigProSSSubdir

Assess/evaluate results from sigproSS (a.k.a. SigProfiler Python attribution package)

Description

Assess/evaluate results from sigproSS (a.k.a. SigProfiler Python attribution package)

Usage

```
SummarizeSigOneSigProSSSubdir(
  run.dir,
  ground.truth.exposure.dir = paste0(run.dir, "/../../../"),
  overwrite = FALSE,
  summary.folder.name = "summary",
  export.Manhattan.each.spectrum = FALSE
)
```

Arguments

run.dir	Lowest level path to results, e.g. <top.dir>/sa.sa.96/ExtrAttr/SigProExtractor.results/see
	Here, <top.dir> refers to a top-level directory which contains the full information of a synthetic dataset. (e.g. syn.2.7a.7b.abst.v8) This code depends on a conventional directory structure documented elsewhere. However there should be a directory <run.dir>/SBS96 which stores SigProfiler results.

ground.truth.exposure.dir

> TODO(Wu Yang): Fix this File name which stores ground-truth exposures; defaults to `"ground.truth.syn.exposures.csv"`. This file can be found in the `sub.dir`, i.e. `<run.dir>/../../../`

overwrite   If TRUE overwrite existing directories and files.

summary.folder.name

> The name of the folder containing summary results. Usually, it equals to "summary".

export.Manhattan.each.spectrum

> Whether to export csv files for Manhattan distance of each mutational spectrum.

SummarizeSigProExtractor

Summarize SigProfiler results in the sa.sa.96 and/or sp.sp subdirectories.

Description

Summarize SigProfiler results in the sa.sa.96 and/or sp.sp subdirectories.

Usage

```
SummarizeSigProExtractor(
  top.dir,
  sub.dir = c("sa.sa.96", "sp.sp"),
  overwrite = FALSE
)
```

Arguments

top.dir   The top directory of a conventional data structure containing at least one of the subdirectories: sa.sa.96/sp.results and sp.sp/sp.results; see further documentation elsewhere.

sub.dir   The subdirectory under `top.dir`, and containing a folder named sp.results. By default, it contains both `c("sa.sa","sp.sp")`. But you should specify `sub.dir = "sp.sp"` for `top.dir` with only the `sp.sp` subdirectory (as is the case for the correlated SBS1-and-SBS5-containing data sets).

overwrite   If TRUE overwrite and files in existing `run.dir/summary` folder.

Details

Results are put in standardized subdirectories of `top.dir`.

SynSigEval	*SynSigEval*

Description

Assess the performance of two steps in mutational signature analysis:

- signature extraction
- exposure inference (a.k.a. signature attribution)

by computational approaches, using catalogs of synthetic mutational spectra created by package SynSigGen.

Input

SynSigEval requires the input data listed below:

1. E, matrix of synthetic exposures (signatures x samples)
2. S, mutational signature profiles (mutation type x signature)
3. synthetic.spectra, synthetic mutational spectra with known ground-truth mutational signature profiles (S) and exposures (synthetic.exposures). It can be created from SynSigGen.
4. T, signatures extracted by SignatureAnalyzer, SigProfiler, or other computational approaches on synthetic.spectra. For attribution-only approaches, T=S.
5. F, exposures inferred by computational approaches on synthetic.spectra.

Folder structure for SynSigEval v0.2

Summary function will fit to the new 5-level folder structure:

First Level - top.level.dir: dataset folder (e.g. "S.0.1.Rsq.0.1", "syn.pancreas"). All spectra datasets under any top.level.dir have the same exposure.

Second Level - ground.truth.exposure.dir: spectra folder: (e.g. "sp.sp", "sa.sa.96"). All spectra datasets under any second.level.dir have the same signature and the same exposure counts.

Third Level - third.level.dir: It can be ("Attr") for storing results of packages which can only do exposure attribution of known signatures ("Attr"); it can also be ("ExtrAttr"), folder to store results of software packages which can do de-novo extraction and following attribution.

Fourth Level - tool.dir: The results of a software package (e.g. "SigProExtractor.results","SignatureEstimation.QP.resu Under this level, tool.dir may contain multiple run.dir, each is a run of the software package using a specific number of seed.

Fifth level - run.dir: contains results from a run of the software package using a specific number of seed. (e.g. "seed.1")

Summarize results

1. Summarize results in fifth-level run.dir:
   Relevant functions are:
   - SummarizeSigProExtractor
   - SignatureAnalyzerSummarizeTopLevel
   - SignatureAnalyzerSummarizeSBS1SBS5

- `SummarizeSigOneExtrAttrSubdir`
- `SummarizeSigOneAttrSubdir`
- `SummarizeSigOnehelmsmanSubdir`
- `SummarizeSigOneSigProSSSubdir`

2. Summarize results of multiple runs by a computational approach on one spectra data set: `SummarizeMultiRuns`

3. Summarize results of multiple computational approaches on one spectra data set: `SummarizeMultiToolsOneDataset`

4. Summarize results of multiple computational approaches on multiple spectra data sets: `SummarizeMultiToolsMultiDatasets`

Index

27