

Package ‘SynSigEval’

October 6, 2020

Type Package

Title Evaluate Mutational Signature Analysis Software
Using Synthetic Spectra Created by Package SynSigGen

Version 0.1.2

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 (>= 0.0.2),
stats,
devtools,
utils,
rlang,
graphics,
grDevices,
ggplot2,
ggbeeswarm,
gtools

Remotes github::steverozen/ICAMSxtra@*release

Depends R (>= 3.5)

RoxygenNote 7.1.1

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

R topics documented:

CopyBestSignatureAnalyzerResult	2
CreateEMuOutput	3
CreatehelmsmanOutput	4
CreateMultiModalMuSigOutput	4
helmsmanCatalog2ICAMS	5
ICAMSCatalog2EMu	6
ICAMSCatalog2helmsman	6
ICAMSCatalog2MM	7
MMCatalog2ICAMS	7
PlotCatCOMPOSITE	8
ReadAndAnalyzeExposures	8
ReadAndAnalyzeSigs	9
ReadEMuCatalog	10
ReadEMuExposureFile	11
ReadExposureMM	11
ReadhelmsmanExposure	12
ReadSigProfilerExposure	12
ReadSigProfilerSigDBS78	13
ReadSigProfilerSigSBS96	13
SignatureAnalyzerSummarizeSBS1SBS5	14
SignatureAnalyzerSummarizeTopLevel	14
SplitCatCOMPOSITE	15
SummarizeMultiRuns	15
SummarizeMultiToolsMultiDatasets	16
SummarizeMultiToolsOneDataset	17
SummarizeOneToolMultiDatasets	18
SummarizeSigOneAttrSubdir	19
SummarizeSigOneExtrAttrSubdir	20
SummarizeSigOnehelmsmanSubdir	20
SummarizeSigOneSigProExtractorSubdir	21
SummarizeSigOneSigProSSSubdir	22
SummarizeSigProExtractor	22
SynSigEval	23
Index	25

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	<i>Prepare input file for EMu from a EMu formatted catalog file.</i>
-----------------	--

Description

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

Usage

```
CreateEMuOutput(
  catalog,
  out.dir = paste0(dirname(catalog), "/ExtraAttr/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 be the input when running EMu program later on compiled binary.

Value

invisible(catalog), original catalog in EMu format

CreatehelmsmanOutput	<i>Prepare input file for helmsman from a helmsman formatted catalog file.</i>
----------------------	--

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 be the input when running helmsman program later on Python platform.

Value

`invisible(catMatrix)`, original catalog in helmsman format

CreateMultiModalMuSigOutput	<i>Prepare input file for MultiModalMuSig from a MultiModalMuSig formatted catalog file.</i>
-----------------------------	--

Description

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

Usage

```
CreateMultiModalMuSigOutput(
  catalog,
  out.dir = paste0(dirname(catalog), "/ExtraAttr/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 be 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	<i>Convert Catalogs from ICAMS format to MM format</i>
-----------------	--

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	<i>Convert Catalogs (File or Matrix) from MM format to ICAMS format</i>
-----------------	---

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	<i>Plot the a SignatureAnalyzer COMPOSITE signature or catalog into separate pdfs</i>
------------------	---

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	<i>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.</i>
--------------------	---

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	<i>Read Exposure files in EMu format.</i>
---------------------	---

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	<i>Read Catalog files in MM format</i>
----------------	--

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**

<code>file</code>	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**

<code>file</code>	The name of the file to read.
-------------------	-------------------------------

Value

The corresponding signature matrix in standard internal representation.

SignatureAnalyzerSummarizeSBS1SBS5

Summarize all sub-directories of SignatureAnalyzer results on the correlated 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, overwrite = FALSE)
```

Arguments

`top.level.dir` Path to top level directory.
`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, 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 SignatureAnalyzer run, names `sa.run.<n>`, where `<n>` is an integer (string of digits).

`overwrite` If TRUE overwrite existing summary files.

SplitCatCOMPOSITE	<i>Split COMPOSITE (SNS1536+DBS78+ID83) catalogs in ICAMS format into 3 individual catalogs.</i>
-------------------	--

Description

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

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	<i>Assess/evaluate multiple summarized runs for one dataset from one computational approach.</i>
--------------------	--

Description

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

Usage

```
SummarizeMultiRuns(datasetName, toolName, tool.dir, run.names)
```

Arguments

datasetName	Name of the dataset. (e.g. "S.O.1.Rsq.0.1"). Usually, it is has the same name as <code>basename(top.dir)</code> .
toolName	Name of computational approach. (e.g. "SigProExtractor")
tool.dir	Fourth level path from the <code>top.dir</code> . Expected to have multiple runs with different names (e.g. "seed.1") That is, <code>top.dir/sp.sp/ExtrAttr/sa.results/</code> or <code>top.dir/sa.sa.96/Attr/deconstructSigs.results/</code> Here, <code>top.dir</code> refers to a top-level directory which contains the full information of a synthetic dataset. (e.g. <code>syn.2.7a.7b.abst.v8</code>) This code depends on a conventional directory structure documented elsewhere. However there should be a directory within the <code>tool.dir</code> which stores the software output.
run.names	A character vector records the list of <code>run.dir</code> , or fifth level directories from the dataset top-level folder. E.g., <code>c("seed.1","seed.691")</code>

Details

Also writes multiple files into folder `tool.dir`:

Value

A list contain `c(mean,sd)` of multiple runs: Cosine similarity True Positives(TP): Ground-truth signatures which are active in the spectra, and extracted. False Negatives(FN): Ground-truth signatures not extracted. False Positives(FP): Signatures wrongly extracted, not resembling any ground-truth signatures. True positive rate (TPR, Sensitivity): $TP / (TP + FN)$ False discovery rate (FDR): $FP / (FP + TP)$

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 `third.level.dir/multiTools.Rda` exists.

Usage

```
SummarizeMultiToolsMultiDatasets(
  dataset.dirs,
  second.third.level.dirname,
  out.dir,
  overwrite = FALSE
)
```

Arguments

`dataset.dirs` Paths of top-level dataset directories trees you want to investigate. E.g. `"/S.0.1.Rsq.0.1"`
`second.third.level.dirname`

Name of the `second.level.dir` (e.g. `"sp.sp"`) and the `third.level.dir` (e.g. `"ExtrAttr"`) to be investigated.

Examples are: `"sp.sp/ExtrAttr"`, `"sa.sa.96/Attr"`

Note: `multiTools.RDa` are expected to be exist under `dataset.dirs/second.third.level.dirname`

`out.dir` Path of the output directory.

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

Details

`multiTools.Rda` is generated by [SummarizeMultiToolsOneDataset](#)).

SummarizeMultiToolsOneDataset

Combine results for a single dataset, from different computational approaches.

Description

Summarize results from each computational approach in `third.level.dir/tool.dirnames` (generated by [SummarizeMultiRuns](#)), combine them into `third.level.dir`.

Usage

```
SummarizeMultiToolsOneDataset(
  third.level.dir,
  toolNames,
  tool.dirnames,
  datasetGroup,
  datasetGroupName,
  datasetSubGroup,
  datasetSubGroupName
)
```

Arguments

`third.level.dir`

Third level path distinguishing de novo extraction + attribution packages from attribution-only packages. Examples: `top.dir/sp.sp/ExtrAttr/` `top.dir/sa.sa/Attr/`

`toolNames` Names of computational approach. (e.g. "SigProExtractor")

`tool.dirnames` Third level path from the `top.dir`. Expected to have summarized results generated by [SummarizeMultiRuns](#). (multiRun.RDa, ManhattanDist.csv, meanSD.csv, meanSD.Manhattan.dist.csv) Examples: "signeR.results" (Under `third.level.dir` "ExtrAttr") "deconstructSigs.results" (Under `third.level.dir` "Attr")

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 within the `tool.names` which stores the software output.

`datasetGroup` Numeric or character vector specifying the group each dataset belong to. E.g. For SBS1-SBS5 correlated datasets, we can consider slope as the group: `c("slope=0.1", "slope=0.5", "slope=0.9")` Default: "Default"

`datasetGroupName`

Meaning or label of all `datasetGroup`. E.g. For SBS1-SBS5 correlated datasets, we can consider "SBS1:SBS5 mutation count ratio" as the label of the `datasetGroup` slope.

`datasetSubGroup`

Numeric or character vector differentiating datasets within each group. E.g. For SBS1-SBS5 correlated datasets, we can consider Pearson's R^2 as the subgroup: `c("Rsqr=0.1", "Rsqr=0.2", "Rsqr=0.3", "Rsqr=0.6")` Default: Names of datasets, which are `basename(dataset.dirs)`

datasetSubGroupName

Meaning or label of all datasetSubGroup. E.g. For SBS1-SBS5 correlated datasets, we can consider "Pearson's R squared" as the label of the datasetSubGroup Pearson's R^2 .

Details

This function generates multiTools.RDa under third.level.dir

Value

A list contain c(mean,sd) of multiple runs: Cosine similarity True Positives(TP): Ground-truth signatures which are active in the spectra, and extracted. False Negatives(FN): Ground-truth signatures not extracted. False Positives(FP): Signatures wrongly extracted, not resembling any ground-truth signatures. True positive rate (TPR, Sensitivity): $TP / (TP + FN)$ False discovery rate (FDR): $FP / (FP + TP)$

SummarizeOneToolMultiDatasets

Combine results for multiple datasets, from one computational approaches.

Description

Summarize results from each computational approach in third.level.dir/tool.dirnames (generated by [SummarizeMultiRuns](#)), combine them into third.level.dir.

Usage

```
SummarizeOneToolMultiDatasets(
  dataset.dirs,
  datasetGroup = NULL,
  datasetGroupName,
  datasetSubGroup = NULL,
  datasetSubGroupName,
  toolName,
  tool.dirname,
  out.dir,
  overwrite = FALSE
)
```

Arguments

dataset.dirs	Paths of top-level dataset directories trees you want to investigate. E.g. <code>"/S.0.1.Rsq.0.1"</code>
datasetGroup	Numeric or character vector specifying the group each dataset belong to. E.g. For SBS1-SBS5 correlated datasets, we can consider slope (SBS1:SBS5 count ratio) as the group: <code>c(0.1, 0.5, 1, 2, 5, 10)</code> Default: <code>"Default"</code>
datasetGroupName	Meaning or label of all datasetGroup. E.g. For SBS1-SBS5 correlated datasets, we can consider <code>"SBS1:SBS5 mutation count ratio"</code> as the label of the datasetGroup slope.

datasetSubGroup	Numeric or character vector differentiating datasets within each group. E.g. For SBS1-SBS5 correlated datasets, we can consider Pearson's R^2 as the subgroup: <code>c(0.1,0.2,0.3,0.6)</code> Default: Names of datasets, which are <code>basename(dataset.dirs)</code>
datasetSubGroupName	Meaning or label of all datasetSubGroup. E.g. For SBS1-SBS5 correlated datasets, we can consider "Pearson's R^2 " as the label of the datasetSubGroup
toolName	Name of computational approach to be investigated (e.g. "SigProExtractor")
tool.dirname	Name of the second.level.dir (e.g. "sp.sp"), third.level.dir (e.g. "ExtrAttr") and tool.dir (e.g. "SigProExtractor.results") to be investigated. One example: "sp.sp/ExtrAttr/SigProExtractor.results" Note: this function expects the summary generated by SummarizeSigOneSubdir under <code>dataset.dirs/tool.dirname</code>
out.dir	Path of the output directory.
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
)
```

Arguments

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

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

Description

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

Usage

```
SummarizeSigOneExtrAttrSubdir(  
  run.dir,  
  ground.truth.exposure.dir = paste0(run.dir, "../.../"),  
  overwrite = FALSE  
)
```

Arguments

run.dir	Lowest level path to result of a run. E.g. <top.dir>/sa.sa.96/ExtrAttr/SomaticSignatures.results 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. For packages which can do both extraction and attribution, we expect two files, extracted.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.

SummarizeSigOnehelmsmanSubdir	<i>Assess/evaluate results from SigProfiler-python (a.k.a. SigProExtractor) Assessment is restricted to v0.0.5.43, because different version has different folder structure.</i>
-------------------------------	--

Description

Assess/evaluate results from SigProfiler-python (a.k.a. SigProExtractor) Assessment is restricted to v0.0.5.43, because different version has different folder structure.

Usage

```
SummarizeSigOnehelmsmanSubdir(
    run.dir,
    ground.truth.exposure.dir = paste0(run.dir, "../.../"),
    overwrite = FALSE,
    hierarchy = 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	Folder which stores ground-truth exposures. Usually, it refers to sub.dir, i.e. run.dir/.../
overwrite	If TRUE overwrite existing directories and files.
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. (Default: FALSE)

SummarizeSigOneSigProExtractorSubdir

Assess/evaluate results from SigProExtractor SigProfiler-python de novo extraction and attribution package. Assessment is restricted to v0.0.5.43+, because different version has different folder structure.

Description

Assess/evaluate results from SigProExtractor SigProfiler-python de novo extraction and attribution package. Assessment is restricted to v0.0.5.43+, because different version has different folder structure.

Usage

```
SummarizeSigOneSigProExtractorSubdir(
    run.dir,
    ground.truth.exposure.dir = paste0(run.dir, "../.../"),
    overwrite = FALSE,
    hierarchy = 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.
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. (Default: FALSE)

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
)
```

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.

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	whether to overwrite the existing run.dir/summary folder? If chosen to be FALSE and there is an existing summary folder, an error will be raised.

Details

Results are put in standardized subdirectories of top.dir.

SynSigEval	<i>SynSigEval</i>
------------	-------------------

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.results"). 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

BestSignatureAnalyzerResult, [3](#)

CopyBestSignatureAnalyzerResult, [2](#)

CreateEMuOutput, [3](#)

CreatehelmsmanOutput, [4](#)

CreateMultiModalMuSigOutput, [4](#)

data.frame, [9](#)

helmsmanCatalog2ICAMS, [5](#)

ICAMSCatalog2EMu, [6](#)

ICAMSCatalog2helmsman, [6](#)

ICAMSCatalog2MM, [7](#)

make.names, [12](#)

MatchSigsAndRelabel, [9](#), [10](#)

MMCatalog2ICAMS, [7](#)

PlotCatalogToPdf, [8](#)

PlotCatCOMPOSITE, [8](#)

ReadAndAnalyzeExposures, [8](#)

ReadAndAnalyzeSigs, [9](#)

ReadEMuCatalog, [10](#)

ReadEMuExposureFile, [11](#)

ReadExposureMM, [11](#)

ReadhelmsmanExposure, [12](#)

ReadSigProfilerExposure, [12](#)

ReadSigProfilerSigDBS78, [13](#)

ReadSigProfilerSigSBS96, [13](#)

SignatureAnalyzerSummarizeSBS1SBS5, [14](#),
[24](#)

SignatureAnalyzerSummarizeTopLevel, [14](#),
[24](#)

SplitCatCOMPOSITE, [15](#)

SummarizeMultiRuns, [15](#), [17](#), [18](#)

SummarizeMultiToolsMultiDatasets, [16](#)

SummarizeMultiToolsOneDataset, [16](#), [17](#)

SummarizeOneToolMultiDatasets, [18](#)

SummarizeSigOneAttrSubdir, [19](#), [24](#)

SummarizeSigOneExtrAttrSubdir, [20](#), [24](#)

SummarizeSigOnehelmsmanSubdir, [20](#), [24](#)

SummarizeSigOneSigProExtractorSubdir,
[21](#)

SummarizeSigOneSigProSSSubdir, [22](#), [24](#)

SummarizeSigProExtractor, [22](#), [24](#)

SynSigEval, [23](#)