Package 'mSigAct'

December 8, 2020

Title mutational Signature Activity analysis ('mSigAct')

Version 2.0.8

Author Steve Rozen, Alvin Wei Tian Ng, Arnoud Boot

Maintainer Steve Rozen <steverozen@gmail.com>

Description Analyze the ``activities'' of mutational signatures in one or more mutational spectra. 'mSigAct' stands for mutational Signature Activity. mSigAct can estimate (conservatively) whether there is evidence that a particular set of mutational signatures is present in a spectrum. It can also determine a *minimal* subset of signatures needed to plausibly reconstruct an observed spectrum. This sparse assign signatures functionality is *deliberately biased* toward using as few signatures as possible.This package does not provide all-purpose estimation for signature attribution.

License GPL-3

URL https://github.com/steverozen/mSigAct

BugReports https://github.com/steverozen/mSigAct/issues

Encoding UTF-8

LazyData true

Language en-US

Remotes github::steverozen/PCAWG7,
github::steverozen/ICAMS@master

Depends R (>= 3.5),

RoxygenNote 7.1.1

VignetteBuilder knitr

biocViews

Imports dplyr,
ICAMS,
lsa,
nloptr,
philentropy,
quadprog,
rlang,
stats,
sets,
tibble,
utils

Suggests BSgenome.Hsapiens.1000genomes.hs37d5,
devtools,
htmlwidgets,
knitr,
PCAWG7,
profvis,
rmarkdown,
testthat (>= 2.1.0),
usethis

R topics documented:

`cossim` *Cosine similarity with useful argument types..*

Description

Calls `cosine`.

Usage

```
cossim(v1, v2)
```

Arguments

v1	A vector or single-column matrix
v2	A vector or single-column matrix

`DefaultManyOpts` *Set default options for many functions, especially* `nloptr`.

Description

Set default options for many functions, especially `nloptr`.

Usage

```
DefaultManyOpts()
```

Value

A list with the following elements

global.opts A sub-list with several options for `nloptr`, q.v., for the global optimization phase, including `eval_f`, the objective function.

local.opts A sub-list with several options for `nloptr`, q.v., for the local optimization phase, including `eval_f`, the objective function and the inequality constraint function `eval_g_ineq`

nbinom.size The dispersion parameter for the negative binomial distribution; smaller is more dispersed. See `NegBinomial`.

trace If > 0 print progress messages.

`ExposureProportions` *Return the proportions of tumors of a given cancer type that have a particular signature*

Description

Return the proportions of tumors of a given cancer type that have a particular signature

Usage

```
ExposureProportions(
  mutation.type,
  cancer.type,
  all.sigs = NULL,
  drop.sigs.no.info = TRUE
)
```

Arguments

mutation.type	A character string, one of "SBS96", "SBS192", "ID", "DBS78".
cancer.type	A character string.
all.sigs	An optional matrix of known signatures, with column names being signatures ids.
drop.sigs.no.info	If TRUE, drop any not present in the column names of all.sigs. There are some signatures that do not have SBS192 versions, including SBS29.

Value

A numerical vector of the proportion of tumors of type cancer.type with each signature for those signatures observed in cancer.type.The names are the signature ids.

---

g_ineq_for_ObjFnBinomMaxLH2

Function to constrain the sum of estimated exposures to the number of mutations in the spectrum.

Description

See nloptr to understand how this function is used.

Usage

```
g_ineq_for_ObjFnBinomMaxLH2(exp, spectrum, sigs, nbinom.size)
```

Arguments

exp	A numeric vector of exposures.
spectrum	The observed spectrum we are trying to reconstruct.
sigs	The signatures with which we are trying to reconstruct the spectrum. (Ignored in this function but used by nloptr.)
nbinom.size	Dispersion parameter. (Ignored in this function but used by nloptr.)

LLHSpectrumNegBinom

Likelihood that 1 observed spectrum was generated from a vector of expected mutation counts.

Description

Likelihood that 1 observed spectrum was generated from a vector of expected mutation counts.

Usage

```
LLHSpectrumNegBinom(spectrum, expected.counts, nbinom.size, verbose = FALSE)
```

Arguments

spectrum	An observed spectrum (a numeric vector)
expected.counts	A vector of (integer) expected mutation counts, one expected count for each mutation type. We want to know the likelihood that this model generated the observed spectrum, assuming each mutational types generates counts according to a negative binomial distribution with the given expected.counts (argument mu to NegBinomial) and dispersion parameter nbinom.size.
nbinom.size	The dispersion parameter for the negative binomial distribution; smaller is more dispersed. See NegBinomial.
verbose	If TRUE print messages under some circumstances.

Value

log(likelihood(spectrum | expected.counts)), or, in more detail, the sum of the negative binomial likelihoods that each each element of the spectrum (i.e., the count for each mutation type e.g. ACT > AAT) was generated from the expected count for that mutation type.

MAPAssignActivity1 *Find a Maximum A Posteriori (MAP) assignment of signature exposures that explain one spectrum.*

Description

Find a Maximum A Posteriori (MAP) assignment of signature exposures that explain one spectrum.

Usage

```
MAPAssignActivity1(
  spect,
  sigs,
  sigs.presence.prop,
  max.level = 5,
  p.thresh = 0.05,
  m.opts = DefaultManyOpts(),
  max.mc.cores = min(20, 2^max.level),
  max.subsets = 1000,
  max.presence.proportion = 0.99,
  progress.monitor = NULL
)
```

Arguments

spect	A single spectrum.
sigs	A numerical matrix, possibly an ICAMS catalog.
sigs.presence.prop	The proportions of samples that contain each signature. A numerical vector (values between 0 and 1), with names being a subset of colnames(sigs).
max.level	The maximum number of signatures to try removing.

`p.thresh` If the p value for a better reconstruction with than without a set of signatures is > than `p.thresh`, then we can use exposures without this set.

`m.opts` See `DefaultManyOpts`.

`max.mc.cores` The maximum number of cores to use. On Microsoft Windows machines it is silently changed to 1.

`max.subsets` The maximum number of subsets that can be tested for removal from the set of signatures.

`max.presence.proportion` The maximum value of the proportion of tumors that must have a given signature.

`progress.monitor` Function called at the start of each new level (number of signatures to try excluding). Must take named arguments `value` and `detail`, and no others. Designed for a `AsyncProgress` progress bar function.

Value

A list with the elements

MAP A 2-column `tibble` with the attributions with the highest MAP found. Column 1 contains signature ids; column 2 contains the associated counts.

MAP.row A 1-row `tibble` with various information on the selected exposure.

best.sparse A 2-column `tibble` with the most-sparse attributions with the highest MAP, in the same format as element `MAP`.

best.sparse.row A 1-row `tibble` with various information on the most-sparse exposure with the best MAP.

all.tested A `tibble` of all the search results.

messages Possibly empty character vector with messages.

success `TRUE` is search was successful, `FALSE` otherwise.

time.for.MAP.assign Value from `system.time` for `MAPAssignActivityInternal`.

MAP.recon Reconstruction based on `MAP`.

sparse.MAP.recon Reconstruction based on `best.sparse`.

MAP.distances Various distances and similarities between `spect` and `MAP.recon`.

sparse.MAP.distances Various distances and similarities between `spect` and `sparse.MAP.recon`.

These elements will be `NULL` if `max.subsets` is exceeded.

`ObjFnBinomMaxLHMustRound`

A deprecated negative binomial maximum likelihood objective function.

Description

Use `ObjFnBinomMaxLHRound` instead.

Usage

```
ObjFnBinomMaxLHMustRound(exp, spectrum, sigs, nbinom.size)
```

Arguments

exp	A vector of exposures ("activities").
spectrum	The spectrum to assess.
sigs	The matrix of signatures.
nbinom.size	The dispersion parameter for the negative binomial distribution; smaller is more dispersed. See NegBinomial.

Details

This function will lead to errors in some situations when the rounded reconstructed signature contains 0s for mutations classes for which the target spectrum is > 0.

ObjFnBinomMaxLHNoRoundOK

A deprecated negative binomial maximum likelihood objective function.

Description

Use ObjFnBinomMaxLHRound instead.

Usage

```
ObjFnBinomMaxLHNoRoundOK(exp, spectrum, sigs, nbinom.size)
```

Arguments

exp	A vector of exposures ("activities").
spectrum	The spectrum to assess.
sigs	The matrix of signatures.
nbinom.size	The dispersion parameter for the negative binomial distribution; smaller is more dispersed. See NegBinomial.

Details

This function rounds sometimes, which leads to minor differences in log likelihoods of reconstructed spectra (LLHSpectrumNegBinom) compared to the value returned by this function.

`ObjFnBinomMaxLHRound` *The preferred negative binomial maximum likelihood objective function.*

Description

Can be used as the objective function for `SparseAssignActivity`, `SparseAssignActivity1`, and `SignaturePresenceTest1`. (Internally used by by `nloptr`.)

Usage

```
ObjFnBinomMaxLHRound(exp, spectrum, sigs, nbinom.size)
```

Arguments

`exp`	A vector of exposures ("activities").
`spectrum`	The spectrum to assess.
`sigs`	The matrix of signatures.
`nbinom.size`	The dispersion parameter for the negative binomial distribution; smaller is more dispersed. See `NegBinomial`.

Value

-1 * log(likelihood(spectrum | reconstruction))

`nloptr` minimizes the objective function, so the lower the objective function, the better.

`OneMAPAssignTest` *Run one test of* `MAPAssignActivity1`.

Description

Run one test of `MAPAssignActivity1`.

Usage

```
OneMAPAssignTest(
  spect,
  reference.exp,
  cancer.type,
  mutation.type,
  exposure.mutation.type,
  max.subsets = 1000,
  max.level = 5,
  max.mc.cores = 100,
  m.opts = DefaultManyOpts(),
  out.dir = NULL,
  p.thresh,
```

```
  max.presence.proportion,
  sigs.prop = NULL,
  sigs = NULL
)
```

Arguments

spect	A single spectrum.
reference.exp	Compare the inferred exposures to this.
cancer.type	Character string from a fixed set indicating different cancer types, used to look up the set of signatures known in that cancer type and the proportion of cancers of that type that have the signature. TODO: provide information on how to find the allowed cancer types.
mutation.type	One of "SBS96", "SBS192", "ID", "DBS78".
exposure.mutation.type	One of "SBS96", "ID", "DBS78".
max.subsets	The maximum number of subsets that can be tested for removal from the set of signatures.
max.level	The maximum number of signatures to try removing.
max.mc.cores	The maximum number of cores to use. On Microsoft Windows machines it is silently changed to 1.
m.opts	See DefaultManyOpts.
out.dir	If non-NULL create this directory if necessary and put results there.
p.thresh	If the p value for a better reconstruction with than without a set of signatures is > than p.thresh, then we can use exposures without this set.
max.presence.proportion	The maximum value of the proportion of tumors that must have a given signature. Used so that it is possible to exclude a signature from a spectrum, e.g. perhaps all examples of tumor types have SBS5, but we want to allow a small chance that SBS5 is not present.
sigs.prop	The proportions of samples that contain each signature. A numerical vector (values between 0 and 1), with names being signature identifiers. Can be the return value from ExposureProportions.
sigs	Matrix of signatures.

OptimizeExposure *Optimize the reconstruction of a spectrum from a set of signatures.*

Description

Optimize the reconstruction of a spectrum from a set of signatures.

Usage

```
OptimizeExposure(spectrum, sigs, m.opts, ...)
```

Arguments

spectrum	The spectrum to be reconstructed.
sigs	The available signatures.
m.opts	Options that govern the numerical optimization. For documentation see DefaultManyOpts.
...	Additional arguments for eval_f.

Returns a list with elements

loglh The log likelihood of the best solution (set of exposures) found.

exposure The vector of exposures that generated loglh, i.e. the number of mutations ascribed to each signature.

objective The final value of the objective function.

solution The optimum exposures.

warnings A character vector of warnings.

global.search.diagnostics Diagnostics from nloptr.

local.search.diagnostics Diagnostics from nloptr.

OptimizeExposureQP *Quadratic programming optimization of signature activities*

Description

Quadratic programming optimization of signature activities

Usage

```
OptimizeExposureQP(spectrum, signatures)
```

Arguments

spectrum	Mutational signature spectrum as a numeric vector or single column data frame or matrix.
signatures	Matrix or data frame of signatures from which reconstruct spectrum. Rows are mutation types and columns are signatures. Should have column names for interpretable results. Cannot be a vector because the column names are needed.

Value

A vector of exposures with names being the colnames from signatures. Code adapted from SignatureEstimation::decomposeQP.

`OptimizeExposureQPBootstrap` *Bootstrap* `OptimizeExposureQP` *and filter exposures by confidence intervals*

Description

Bootstrap `OptimizeExposureQP` and filter exposures by confidence intervals

Usage

```
OptimizeExposureQPBootstrap(
  spectrum,
  signatures,
  num.replicates = 10000,
  conf.int = 0.95,
  mc.cores = 10
)
```

Arguments

`spectrum`	Mutational signature spectrum as a numeric vector or single column data frame or matrix.
`signatures`	Matrix or data frame of signatures from which reconstruct `spectrum`. Rows are mutation types and columns are signatures. Should have column names for interpretable results. Cannot be a vector because the column names are needed.
`num.replicates`	Number of bootstrap replicates.
`mc.cores`	The maximum number of cores to use. On MS Windows machines it defaults to 1.
`conf.interval`	Discard signatures with `conf.int` that overlaps 0.

`PCAWGMAPTest` *Run* `MAPAssignActivity1` *on one sample from the PCAWG platinum data set.*

Description

Run `MAPAssignActivity1` on one sample from the PCAWG platinum data set.

Usage

```
PCAWGMAPTest(
  cancer.type,
  sample.index,
  mutation.type,
  max.level = 5,
  max.mc.cores,
```

```
  m.opts = DefaultManyOpts(),
  out.dir = NULL,
  p.thresh = 0.01,
  max.presence.proportion = 0.99,
  sigs.prop = NULL
)
```

Arguments

cancer.type	A cancer type from the PCAWG exposures matrix.
sample.index	The index of the sample within the exposures matrix.
mutation.type	One of "SBS96", "SBS192", "ID", "DBS78"
max.level	The maximum number of signatures to try removing.
max.mc.cores	The maximum number of cores to use. On Microsoft Windows machines it is silently changed to 1.
m.opts	See DefaultManyOpts.
out.dir	If non-NULL create this directory if necessary and put results there.
p.thresh	If the p value for a better reconstruction with than without a set of signatures is > than p.thresh, then we can use exposures without this set.
max.presence.proportion	The maximum value of the proportion of tumors that must have a given signature. Used so that it is possible to exclude a signature from a spectrum, e.g. perhaps all examples of tumor types have SBS5, but we want to allow a small chance that SBS5 is not present.
sigs.prop	The proportions of samples that contain each signature. A numerical vector (values between 0 and 1), with names being signature identifiers. Can be the return value from ExposureProportions.

Value

See OneMAPAssignTest.

PossibleArtifacts *Return a character vector of the IDs of possible SBS96 signature artifacts.*

Description

Return a character vector of the IDs of possible SBS96 signature artifacts.

Usage

```
PossibleArtifacts()
```

RareSignatures *Return a character vector of the IDs of rare SBS96 signatures.*

Description

Return a character vector of the IDs of rare SBS96 signatures.

Usage

```
RareSignatures()
```

ReconstructSpectrum *Given signatures (sigs) and exposures (exp), return a spectrum or spectra*

Description

Given signatures (sigs) and exposures (exp), return a spectrum or spectra

Usage

```
ReconstructSpectrum(sigs, exp, use.sig.names = FALSE)
```

Arguments

sigs	Signature as a matrix or or data frame, with each row one mutation type (g.e. CCT > CAT or CC > TT) and each column a signature.
exp	The exposures for one or more samples as a matrix or data.frame, with each row a signature and each column a sample.
use.sig.names	If TRUE check that rownames(exp) is a subset of colnames(sigs), and use only the columns in sigs that are present in exp.

Details

Does not care or check if colSums(sigs) == 1. Error checking is minimal since this function is called often.

SignaturePresenceTest

Test whether a given signature is plausibly present in a catalog.

Description

Test whether a given signature is plausibly present in a catalog.

Usage

```
SignaturePresenceTest(
  spectra,
  sigs,
  target.sig.index,
  m.opts = NULL,
  mc.cores = 10
)
```

Arguments

spectra	The catalog (matrix) to analyze. This could be an ICAMS catalog or a numerical matrix.
sigs	A catalog of signatures from which to choose. This could be and ICAMS catalog or a numerical matrix.
target.sig.index	The index of the signature the presence of which we want to test.
m.opts	If NULL use the return from calling DefaultManyOpts. For documentation see DefaultManyOpts.
mc.cores	Number of cores to use. Always silently changed to 1 on Microsoft Windows.

SignaturePresenceTest1

Test whether a given signature is plausibly present in a spectrum.

Description

For backward compatibility. See also AnySigSubsetPresent.

Usage

```
SignaturePresenceTest1(spectrum, sigs, target.sig.index, m.opts)
```

Arguments

spectrum	The spectrum to analyze.
sigs	A catalog of signatures from which to choose.
target.sig.index	The index of the signature the presence of which we want to test.
m.opts	For documentation see DefaultManyOpts.

`SparseAssignActivity` *Find known signatures that can most sparsely reconstruct each spectrum in a catalog.*

Description

Find known signatures that can most sparsely reconstruct each spectrum in a catalog.

Usage

```
SparseAssignActivity(
  spectra,
  sigs,
  max.level = 5,
  p.thresh = 0.05,
  m.opts = NULL,
  num.parallel.samples = 5,
  mc.cores.per.sample = min(20, 2^max.level)
)
```

Arguments

`spectra`	The spectra (multiple spectra) to be reconstructed.
`sigs`	The known signatures to use in reconstruction.
`max.level`	The largest number of signatures to consider discarding in the reconstruction.
`p.thresh`	The maximum p value based on which it is decided to retain a signature in a reconstruction.
`m.opts`	For documentation see `DefaultManyOpts`.
`num.parallel.samples`	The (maximum) number of samples to run in parallel; each sample in turn can require multiple cores, as governed by `mc.cores.per.sample`.
`mc.cores.per.sample`	The maximum number of cores to use for each sample. On Microsoft Windows machines it is silently changed to 1.

Value

A list with the inferred exposure matrix as element `exposure`.

XPCAWGMAPTest *Run* MAPAssignActivity1 *on one sample from the PCAWG platinum data set with and with pre-filtering by bootstrapped quadratic programming.*

Description

Run MAPAssignActivity1 on one sample from the PCAWG platinum data set with and with pre-filtering by bootstrapped quadratic programming.

Usage

```
XPCAWGMAPTest(
  cancer.type,
  sample.index,
  mutation.type,
  max.level = 5,
  max.mc.cores,
  out.dir = NULL,
  p.thresh = 0.01,
  m.opts = DefaultManyOpts(),
  max.presence.proportion = 0.99,
  sigs.prop = NULL
)
```

Arguments

cancer.type	A cancer type from the PCAWG exposures matrix.
sample.index	The index of the sample within the exposures matrix.
mutation.type	One of "SBS96", "SBS192", "ID", "DBS78"
max.level	The maximum number of signatures to try removing.
max.mc.cores	The maximum number of cores to use. On Microsoft Windows machines it is silently changed to 1.
out.dir	If non-NULL create this directory if necessary and put results there.
p.thresh	If the p value for a better reconstruction with than without a set of signatures is > than p.thresh, then we can use exposures without this set.
m.opts	See DefaultManyOpts.
max.presence.proportion	The maximum value of the proportion of tumors that must have a given signature. Used so that it is possible to exclude a signature from a spectrum, e.g. perhaps all examples of tumor types have SBS5, but we want to allow a small chance that SBS5 is not present.
sigs.prop	The proportions of samples that contain each signature. A numerical vector (values between 0 and 1), with names being signature identifiers. Can be the return value from ExposureProportions.

Value

A list with two elements, each the result for one call to OneMAPAssignTest.

YPCAWGMAPTest *Run* MAPAssignActivity1 *on one sample from the PCAWG platinum data set with global opts maxeval 10000 and 1000 and compare the results*

Description

Run MAPAssignActivity1 on one sample from the PCAWG platinum data set with global opts maxeval 10000 and 1000 and compare the results

Usage

```
YPCAWGMAPTest(
  cancer.type,
  sample.index,
  mutation.type,
  max.level = 5,
  max.mc.cores,
  out.dir = NULL,
  p.thresh = 0.01,
  m.opts = DefaultManyOpts(),
  max.presence.proportion = 0.99,
  sigs.prop = NULL
)
```

Arguments

cancer.type	A cancer type from the PCAWG exposures matrix.
sample.index	The index of the sample within the exposures matrix.
mutation.type	One of "SBS96", "SBS192", "ID", "DBS78"
max.level	The maximum number of signatures to try removing.
max.mc.cores	The maximum number of cores to use. On Microsoft Windows machines it is silently changed to 1.
out.dir	If non-NULL create this directory if necessary and put results there.
p.thresh	If the p value for a better reconstruction with than without a set of signatures is > than p.thresh, then we can use exposures without this set.
m.opts	See DefaultManyOpts.
max.presence.proportion	The maximum value of the proportion of tumors that must have a given signature. Used so that it is possible to exclude a signature from a spectrum, e.g. perhaps all examples of tumor types have SBS5, but we want to allow a small chance that SBS5 is not present.
sigs.prop	The proportions of samples that contain each signature. A numerical vector (values between 0 and 1), with names being signature identifiers. Can be the return value from ExposureProportions.

Value

A list with two elements, each the result for one call to OneMAPAssignTest.

ZPCAWGMAPTest *Run* MAPAssignActivity1 *on one sample from the PCAWG platinum data set with and with pre-filtering by bootstrapped quadratic programming.*

Description

Run MAPAssignActivity1 on one sample from the PCAWG platinum data set with and with pre-filtering by bootstrapped quadratic programming.

Usage

```
ZPCAWGMAPTest(
  cancer.type,
  sample.index,
  mutation.type,
  max.level = 5,
  max.mc.cores,
  out.dir = NULL,
  p.thresh = 0.01,
  m.opts = DefaultManyOpts(),
  max.presence.proportion = 0.99,
  sigs.prop = NULL
)
```

Arguments

cancer.type	A cancer type from the PCAWG exposures matrix.
sample.index	The index of the sample within the exposures matrix.
mutation.type	One of "SBS96", "SBS192", "ID", "DBS78"
max.level	The maximum number of signatures to try removing.
max.mc.cores	The maximum number of cores to use. On Microsoft Windows machines it is silently changed to 1.
out.dir	If non-NULL create this directory if necessary and put results there.
p.thresh	If the p value for a better reconstruction with than without a set of signatures is > than p.thresh, then we can use exposures without this set.
m.opts	See DefaultManyOpts.
max.presence.proportion	The maximum value of the proportion of tumors that must have a given signature. Used so that it is possible to exclude a signature from a spectrum, e.g. perhaps all examples of tumor types have SBS5, but we want to allow a small chance that SBS5 is not present.
sigs.prop	The proportions of samples that contain each signature. A numerical vector (values between 0 and 1), with names being signature identifiers. Can be the return value from ExposureProportions.

Value

A list with two elements, each the result for one call to OneMAPAssignTest.

Index