Package ‘mSigAct’

May 15, 2020

Title mutational Signature Activity analysis ('mSigAct')

Version 2.0.0

Author Steve Rozen, Alvin Wei Tian Ng, Arnoud Boot

Maintainer Steve Rozen <steverozen@gmail.com>

Description Analyze the 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 mutational signature is present in a spectrum and can determine a minimal subset of signatures needed to plausibly reconstruct an observed spectrum.

License GPL-3

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

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

Encoding UTF-8

LazyData true

Language en-US

Depends R (>= 3.5),

RoxygenNote 6.1.1

Remotes github::steverozen/PCAWG7

biocViews

Imports ICAMS,
nloptr,
stats,
lsa,
sets

Suggests testthat (>= 2.1.0),
usethis,
utils,
PCAWG7

R topics documented:

`background.info` *Specifications of background signatures*

Description

Specifications of background signatures

Usage

```
HepG2.background.info
```

Format

A list with the elements

1. `background.sig` The background signature profile.
2. `mean.sig` Deprecated – the background signature profile computed as the mean of the background spectra; this is essentially the same as `background.sig`.
3. `sig.nbinom.size` The `size` argument for `dnbinom` for sampling error around the components of `background.sig`.
4. `count.nbinom.mu` The `mu` argument for `dnbinom` for the distribution of total counts due to `background.sig` across replicate exposed clones.
5. `count.nbinom.size` The `size` argument for `dnbinom` for the distribution of total counts due to `background.sig` across replicate exposed clones.

Examples

```
# MORE MORE MORE
```

`cisplatin.exposed.HepG2.96` *Mutational spectra of cisplatin exposed HepG2 cells.*

Description

Mutational spectra of cisplatin exposed HepG2 cells.

Usage

```
cisplatin.exposed.HepG2.96
```

Format

An `ICAMS` `counts` catalog.

`DefaultManyOpts` *Set default options for many functions, especially* `link[noptr]{nloptr}`.

Description

Set default options for many functions, especially `link[noptr]{nloptr}`.

Usage

```
DefaultManyOpts()
```

Value

A list with the following elements

global.opts Options for `nloptr`, q.v., for the global optimization phase.

local.opts Options for `nloptr`, q.v., for the local optimization phase.

nbinom.size The `size` parameter used by `dnbinom`.

trace If > 0 print progress messages.

EstimateSignatureFromSpectraLH

Build a signature for background extraction from a matrix of spectra.

Description

This function not only produces a signature, but also an estimate of the number of mutations usually generated by the signature and an indication of variability around that estimate.

Usage

```
EstimateSignatureFromSpectraLH(spectra, algorithm = "NLOPT_LN_COBYLA",
  maxeval = 1000, print_level = 0, xtol_rel = 0.001,
  xtol_abs = 1e-04)
```

Arguments

spectra	An ICAMS catalog with catalog.type = "counts".
algorithm	See link[nloptr]{nloptr}.
maxeval	See link[nloptr]{nloptr}.
print_level	See link[nloptr]{nloptr}.
xtol_rel	See link[nloptr]{nloptr}.
xtol_abs	See link[nloptr]{nloptr}.

Details

Only works on SBS 96 signatures.

Value

A list with the elements

1. signature An ICAMS catalog with catalog.type == "counts.signature".
2. log10.counts Mean log base 10 of the total counts in spectra
3. sd.log10.counts.per.base Standard deviation of log10.counts.per.base.

HepG2.background.spectra

Background spectra for HepG2.

Description

Background spectra for HepG2.

Usage

```
HepG2.background.spectra
```

Format

An ICAMS counts catalog.

`HepG2.bg.tests.no.noise` *Tests of HepG2 background subtraction*

Description

Tests of HepG2 background subtraction

Usage

```
HepG2.bg.tests.no.noise
```

Format

A data table

Examples

```
HepG2.bg.tests.no.noise[1:2, 1:10]
```

`kucab.control.dist` *Resampled mutations counts for combinations of 2:4* `kucab.controls`*.*

Description

Resampled mutations counts for combinations of 2:4 `kucab.controls`.

Usage

```
kucab.control.dist
```

Format

A list with elements 2:4, each of which is a vector of 10,000 elements, each of which is the mean of total mutation counts of 2 to 4 control spectra.

`kucab.controls` *Control spectra from Kucab et al., 2019*

Description

Control spectra from Kucab et al., 2019

Usage

```
kucab.controls
```

Format

An `ICAMS` `counts` catalog with 35 samples.

`LLHOfSignatureGivenSpectrum` *The log likelihood of a* **single** *observed spectrum given the signature.*

Description

Assumes that all mutations `spectrum` were generated by `signature`.

Usage

```
LLHOfSignatureGivenSpectrum(spectrum, signature, nbinom.size)
```

Arguments

`spectrum`	A single background spectrum.
`signature`	A signature as a numeric vector.
`nbinom.size`	The `size` argument for `dnbinom`.

`LLHSpectrumNegBinom` *Likelihood that* **1** *observed spectrum was generated from a vector of expected counts. Returns 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 expected count for that mutation type.*

Description

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

Returns 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 expected count for that mutation type.

Usage

```
LLHSpectrumNegBinom(spectrum, expected.counts, nbinom.size)
```

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 `dnbinom`) and dispersion parameter `nbinom.size`.
`nbinom.size`	The `size` parameter that governs dispersion. See `dnbinom`. Smaller values correspond to larger dispersion.

Value

`log(likelihood(spectrum | expected.counts))`

`ObjFn1` *Objective function for* `FindSignatureMinusBackground`.

Description

Objective function for `FindSignatureMinusBackground`.

Usage

```
ObjFn1(est.target.sig.and.b, obs.spectra, bg.sig.info)
```

Arguments

`est.target.sig.and.b`
A numeric vector of which elements 1:96 are the signature (as a vector) and the remaining elements are the coefficients for each input spectrum in `obs.spectra`.

`obs.spectra` The observed spectra, which may be the sum of the activities of the background signature and the signature of an experimental treatment.

`bg.sig.info` Information on the background signature. See for example `HepG2.background.info`.

`SignaturePresenceTest1`
Test whether a given signature is plausibly present in a spectrum.

Description

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

Usage

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

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` See `DefaultManyOpts`.

`eval_f` See `link[nloptr]{nloptr}`.

`simple.100000.NLOPT_LN_COBYLA`
Another temp test output

Description

Another temp test output

Usage

```
simple.100000.NLOPT_LN_COBYLA
```

Format

An object of class `list` of length 12.

`simple.200000.NLOPT_GN_DIRECT_L`
Another temp test output

Description

Another temp test output

Usage

```
simple.200000.NLOPT_GN_DIRECT_L
```

Format

An object of class `list` of length 12.

`simple.40000.HepG2.tests`
Temporary test data

Description

Temporary test data

Usage

```
simple.40000.HepG2.tests
```

Format

Complicated

Examples

```
# XXXXXX
```

`simple.40000.new.sig0` *Temporary test data*

Description

Temporary test data

Usage

```
simple.40000.new.sig0
```

Format

Complicated

`simple.40000.NLOPT_GN_DIRECT_L` *Another temp test output*

Description

Another temp test output

Usage

```
simple.40000.NLOPT_GN_DIRECT_L
```

Format

An object of class `list` of length 12.

`simple.40000.remainder` *Temporary test data*

Description

Temporary test data

Usage

```
simple.40000.remainder
```

Format

Complicated

sp.sigs *SigProfiler 96 SBS* counts.signatures *under genome abundance.*

Description

SigProfiler 96 SBS counts.signatures under genome abundance.

Usage

```
sp.sigs
```

Format

An [ICAMS] counts.spectrum catalog; see Alexandrov et al. https://www.biorxiv.org/content/10.1101/322859v2.

sp.sigs.exome *SigProfiler 96 SBS* counts.signatures *under exome abundance.*

Description

SigProfiler 96 SBS counts.signatures under exome abundance.

Usage

```
sp.sigs.exome
```

Format

An [ICAMS] counts.spectrum catalog; see Alexandrov et al. https://www.biorxiv.org/content/10.1101/322859v2.

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 = 0.5,
  eval_f, m.opts, mc.cores = NULL)
```

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.

`eval_f` The objective function for `nloptr`.

`m.opts` See`DefaultManyOpts`.

`mc.cores` The number of cores to use; if `NULL` use `ncol{spectra}`, except on Windows, where `mc.cores` is always 1.

Value

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

Index