Package ‘mSigHdp’

May 29, 2020

Title Mutational signature extraction using hdp (Hierarchical Dirichlet Process)

Version 0.0.0.9009

Description Calls hdp for mutational signature analysis, with performance issues in hdp:::stirling() corrected.

License GPL-3

Encoding UTF-8

LazyData true

Language en-US

biocViews

Imports hdpx,
SynSigGen

Roxygen list(markdown = TRUE)

Depends R (>= 3.5)

RoxygenNote 7.1.0

Remotes github::steverozen/hdpx,
github::steverozen/SynSigGen,
github::WuyangFF95/SynSigEval

Suggests testthat,
ICAMS,
utils,
SynSigEval

R topics documented:

`PlotExposure` *Plot a single exposure plot*

Description

Plot a single exposure plot

Usage

```
PlotExposure(exposures, plot.proportion = FALSE, plot.legend = TRUE, ...)
```

Arguments

`exposures` Exposures as a numerical matrix (or data.frame) with signatures in rows and samples in columns. Rownames are taken as the signature names and column names are taken as the sample IDs. If you want `exp` sorted from largest to smallest use `SortExp`. Do not use column names that start with multiple underscores. The exposures will often be mutation counts, but could also be e.g. mutations per megabase.

`plot.proportion` Plot exposure proportions rather than counts.

`plot.legend` If `TRUE` plot a legend.

`...` Parameters passed to `barplot`.

`PlotExposureByRange` *Plot exposures in multiple plots each with a manageable number of samples.*

Description

Plot exposures in multiple plots each with a manageable number of samples.

Usage

```
PlotExposureByRange(exposures, num.per.line = 30, plot.proportion = FALSE, ...)
```

Arguments

`exposures` Exposures as a numerical matrix (or data.frame) with signatures in rows and samples in columns. Rownames are taken as the signature names and column names are taken as the sample IDs. If you want `exposures` sorted from largest to smallest use `SortExp`. Do not use column names that start with multiple underscores. The exposures will often be mutation counts, but could also be e.g. mutations per megabase.

`num.per.line` Number of samples to show in each plot.

`plot.proportion` Plot exposure proportions rather than counts.

`...` Other arguments passed to `PlotExposure`. If `ylab` is not included, it defaults to a value depending on `plot.proportion`. If `col` is not supplied the function tries to do something reasonable.

`RunAndEvalHdp4` *Run and evaluate hdp*

Description

Run and evaluate hdp

Usage

```
RunAndEvalHdp4(
  input.catalog,
  ground.truth.exp = NULL,
  ground.truth.sig.file = NULL,
  ground.truth.sig.catalog = NULL,
  out.dir,
  CPU.cores = 1,
  seedNumber = 1,
  K.guess,
  multi.types = FALSE,
  remove.noise = FALSE,
  test.only = 0,
  overwrite = FALSE,
  verbose = TRUE,
  num.posterior = 4,
  post.burnin = 4000,
  post.n = 50,
  post.space = 50,
  post.cpiter = 3,
  post.verbosity = 0,
  cos.merge = 0.9,
  min.sample = 1
)
```

Arguments

`input.catalog`
Either a character string, in which case this is the path to a file containing a spectra catalog in `ICAMS` format, or an `ICAMS` catalog.

`ground.truth.exp`
Ground truth exposure matrix or path to file with ground truth exposures. If `NULL` skip checks that need this information.

`ground.truth.sig.file`
Path to file with ground truth signatures.

`ground.truth.sig.catalog`
`ICAMS` catalog with signatures used to construct the ground truth spectra. Specify only one of `ground.truth.sig.file.path` or `ground.truth.sig.catalog`.

out.dir Directory that will be created for the output; if overwrite is FALSE then abort if out.dir already exits.

CPU.cores Number of CPUs to use in running hdp_posterior; this is used to parallelize running the posterior sampling chains, so there is no point in making this larger than num.posterior.

seedNumber An integer that is used to generate separate random seeds for each call to dp_activate, and each call of hdp_posterior; please see the code on how this is done. But repeated calls with same value of seedNumber and other inputs should produce the same results.

K.guess Suggested initial value of the number of signatures, passed to dp_activate as initcc.

multi.types A logical scalar or a character vector. If FALSE, hdp will regard all input spectra as one tumor type.

If TRUE, hdp will infer tumor types based on the string before "::" in their names. e.g. tumor type for "SA.Syn.Ovary-AdenoCA::S.500" would be "SA.Syn.Ovary-AdenoCA"

If multi.types is a character vector, then it should be of the same length as the number of columns in input.catalog, and each value is the name of the tumor type of the corresponding column in input.catalog, e.g. c("SA.Syn.Ovary-AdenoC

remove.noise Deprecated; ignored

test.only If > 0, only analyze the first test.only columns in input.catalog.

overwrite If TRUE overwrite out.dir if it exists, otherwise raise an error.

verbose If TRUE then message progress information.

num.posterior

Number of posterior sampling chains; can set to 1 for testing.

post.burnin Pass to hdp_posterior burnin.

post.n Pass to hdp_posterior n.

post.space Pass to hdp_posterior space.

post.cpiter Pass to hdp_posterior cpiter.

post.verbosity

Pass to hdp_posterior verbosity.

cos.merge The cosine similarity threshold for merging raw clusters from the posterior sampling chains into "components" i.e. signatures; passed to hdp_extract_components.

min.sample A "component" (i.e. signature) must have at least this many samples; passed to hdp_extract_components.

Runhdp4 *Run hdp extraction and attribution on a spectra catalog file using hdpx*

Description

Run hdp extraction and attribution on a spectra catalog file using hdpx

Usage

```
Runhdp4(
  input.catalog,
  out.dir,
  CPU.cores = 1,
  seedNumber = 1,
  K.guess,
  multi.types = FALSE,
  remove.noise = FALSE,
  test.only = 0,
  overwrite = FALSE,
  verbose = TRUE,
  num.posterior = 4,
  post.burnin = 4000,
  post.n = 50,
  post.space = 50,
  post.cpiter = 3,
  post.verbosity = 0,
  cos.merge = 0.9,
  min.sample = 1,
  checkpoint.aft.post = NULL,
  plot.extracted.sig = FALSE
)
```

Arguments

`input.catalog`
Either a character string, in which case this is the path to a file containing a spectra catalog in `ICAMS` format, or an `ICAMS` catalog.

`out.dir` Directory that will be created for the output; if `overwrite` is `FALSE` then abort if `out.dir` already exits.

`CPU.cores` Number of CPUs to use in running `hdp_posterior`; this is used to parallelize running the posterior sampling chains, so there is no point in making this larger than `num.posterior`.

`seedNumber` An integer that is used to generate separate random seeds for each call to `dp_activate`, and each call of `hdp_posterior`; please see the code on how this is done. But repeated calls with same value of `seedNumber` and other inputs should produce the same results.

`K.guess` Suggested initial value of the number of signatures, passed to `dp_activate` as `initcc`.

`multi.types` A logical scalar or a character vector. If `FALSE`, hdp will regard all input spectra as one tumor type.

If `TRUE`, hdp will infer tumor types based on the string before "::" in their names. e.g. tumor type for "SA.Syn.Ovary-AdenoCA::S.500" would be "SA.Syn.Ovary-AdenoCA"

If `multi.types` is a character vector, then it should be of the same length as the number of columns in `input.catalog`, and each value is the name of the tumor type of the corresponding column in `input.catalog`, e.g. `c("SA.Syn.Ovary-AdenoC`

`remove.noise` Deprecated; ignored

`test.only` If > 0, only analyze the first `test.only` columns in `input.catalog`.

`overwrite` If `TRUE` overwrite `out.dir` if it exists, otherwise raise an error.

`verbose` If `TRUE` then `message` progress information.

`num.posterior` Number of posterior sampling chains; can set to 1 for testing.

`post.burnin` Pass to `hdp_posterior` `burnin`.

`post.n` Pass to `hdp_posterior` `n`.

`post.space` Pass to `hdp_posterior` `space`.

`post.cpiter` Pass to `hdp_posterior` `cpiter`.

`post.verbosity` Pass to `hdp_posterior` `verbosity`.

`cos.merge` The cosine similarity threshold for merging raw clusters from the posterior sampling chains into "components" i.e. signatures; passed to `hdp_extract_components`.

`min.sample` A "component" (i.e. signature) must have at least this many samples; passed to `hdp_extract_components`.

`checkpoint.aft.post` If non-`NULL`, a file path to checkpoint the list of values returned from the calls to `hdp_posterior` as a .Rdata file.

`plot.extracted.sig` If `TRUE` then plot the extracted signatures.

Details

Creates several files in `out.dir`. These are: call.and.session.info.txt, hdp.diagnostics.pdf, Runhdp4.retval.Rdata, extracted.signatures.csv, extracted.signature.pdf (optional), inferred.exposures.csv.

Value

The same list as returned by `RunhdpInternal4`.

`RunhdpInternal4` *Run hdp extraction and attribution on a spectra catalog file*

Description

Run hdp extraction and attribution on a spectra catalog file

Usage

```
RunhdpInternal4(
  input.catalog,
  CPU.cores = 1,
  seedNumber = 1,
  K.guess,
  multi.types = FALSE,
  verbose = TRUE,
  num.posterior = 4,
  post.burnin = 4000,
  post.n = 50,
  post.space = 50,
  post.cpiter = 3,
  post.verbosity = 0,
  cos.merge = 0.9,
  min.sample = 1,
  checkpoint.aft.post = NULL
)
```

Arguments

`input.catalog`
Input spectra catalog as a matrix or in `ICAMS` format.

`CPU.cores` Number of CPUs to use in running `hdp_posterior`; this is used to parallelize running the posterior sampling chains, so there is no point in making this larger than `num.posterior`.

`seedNumber` An integer that is used to generate separate random seeds for each call to `dp_activate`, and each call of `hdp_posterior`; please see the code on how this is done. But repeated calls with same value of `seedNumber` and other inputs should produce the same results.

`K.guess` Suggested initial value of the number of signatures, passed to `dp_activate` as `initcc`.

`multi.types` A logical scalar or a character vector. If `FALSE`, hdp will regard all input spectra as one tumor type.

If `TRUE`, hdp will infer tumor types based on the string before "::" in their names. e.g. tumor type for "SA.Syn.Ovary-AdenoCA::S.500" would be "SA.Syn.Ovary-AdenoCA"

If `multi.types` is a character vector, then it should be of the same length as the number of columns in `input.catalog`, and each value is the name of the tumor type of the corresponding column in `input.catalog`, e.g. `c("SA.Syn.Ovary-AdenoC`

`verbose` If `TRUE` then `message` progress information.

`num.posterior`
Number of posterior sampling chains; can set to 1 for testing.

`post.burnin` Pass to `hdp_posterior` `burnin`.

`post.n` Pass to `hdp_posterior` `n`.

`post.space` Pass to `hdp_posterior` `space`.

`post.cpiter` Pass to `hdp_posterior` `cpiter`.

`post.verbosity`
Pass to `hdp_posterior` `verbosity`.

`cos.merge` The cosine similarity threshold for merging raw clusters from the posterior sampling chains into "components" i.e. signatures; passed to `hdp_extract_components`.

`min.sample` A "component" (i.e. signature) must have at least this many samples; passed to `hdp_extract_components`.

`checkpoint.aft.post`
If non-`NULL`, a file path to checkpoint the list of values returned from the calls to `hdp_posterior` as a .Rdata file.

Value

A list with the following elements:

signature The extracted signature profiles as a matrix; rows are mutation types, columns are samples (e.g. tumors).

exposure The inferred exposures as a matrix of mutation counts; rows are signatures, columns are samples (e.g. tumors).

exposure.p `exposure` converted to proportions.

multi.chains A `hdpSampleMulti-class` object. This object has the method `chains` which returns a list of `hdpSampleChain-class` objects. Each of these sample chains objects has a method `final_hdpState` (actually the methods seems to be just `hdp`) that returns the `hdpState` from which it was generated.

SortExp — *Sort columns of an exposure matrix from largest to smaller (or vice versa).*

Description

Sort columns of an exposure matrix from largest to smaller (or vice versa).

Usage

```
SortExp(exposures, decreasing = TRUE)
```

Arguments

`exposures`	The exposures to sort; columns are samples.
`decreasing`	If `TRUE` sort from largest to smallest.

Index