Package ‘mSigHdp’

July 10, 2020

Title Mutational signature extraction using hdp (Hierarchical Dirichlet Process)

Version 0.0.0.9026

Description Calls hdpx for mutational signature analysis.

License GPL-3

Encoding UTF-8

LazyData true

Language en-US

biocViews

Imports hdpx (>= 0.1.5.0014),
ICAMS (>= 2.1.2.9010),
SynSigGen

Roxygen list(markdown = TRUE)

Depends R (>= 3.5)

RoxygenNote 7.1.1

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

Suggests testthat,
utils,
SynSigEval

R topics documented:

ActivateAndBurnin *Prepare an hdpState-class object and run the Gibbs sampling burnin.*

Description

Prepare an `hdpState-class` object and run the Gibbs sampling burnin.

Usage

```
ActivateAndBurnin(
  input.catalog,
  seedNumber = 1,
  K.guess,
  multi.types = FALSE,
  verbose = TRUE,
  burnin = 4000,
  cpiter = 3,
  burnin.verbosity = 0,
  gamma.alpha = 1,
  gamma.beta = 1
)
```

Arguments

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

`seedNumber` An integer that is used to generate separate random seeds for the call to `dp_activate`, and before the call of `hdp_burnin`.

`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`, The HDP analysis will regard all input spectra as one tumor type.

If `TRUE`, the HDP analysis 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-AdenoCA","SA.Syn.Kidney-RCC")`.

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

`burnin` Pass to `hdp_burnin` `burnin`.

`cpiter` Pass to `hdp_burnin` `cpiter`.

`burnin.verbosity`
: Pass to `hdp_burnin` `verbosity`.

`gamma.alpha` Shape parameter of gamma distribution from which the Dirichlet process concentration parameters are drawn; in this function the gamma distributions for all Dirichlet processes are the same.

`gamma.beta` Inverse scale parameter (rate parameter) of gamma distribution from which the Dirichlet process concentration parameters are drawn; in this function the gamma distributions for all Dirichlet processes are the same.

Value

A list with 2 elements:

`hdplist` A list representation of an `hdpState-class` object.

likelihood A numeric vector with the likelihood at each iteration.

`AnalyzeAndPlotretval` *Evaluate and plot retval from CombinePosteriorChains*

Description

Evaluate and plot retval from CombinePosteriorChains

Usage

```
AnalyzeAndPlotretval(
  retval,
  out.dir = NULL,
  ground.truth.sig = NULL,
  ground.truth.exp = NULL,
  verbose = TRUE,
  overwrite = TRUE,
  diagnostic.plot = TRUE
)
```

Arguments

`retval` the output from function CombinePosteriorChains

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

`ground.truth.sig`
: Optional. Either a string with the path to file with ground truth signatures or and `ICAMS` catalog with the ground truth signatures. These are the signatures used to construct the ground truth spectra.

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

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

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

`diagnostic.plot`
: If `TRUE` plot diagnostic plot. This is optional because there are cases having error

`ChainsDiagnosticPlot` *Diagnostic plot for a hdpSampleMulti object*

Description

Diagnostic plot for a hdpSampleMulti object

Usage

```
ChainsDiagnosticPlot(retval, out.dir, verbose)
```

Arguments

`retval` output from CombinePosteriorChains.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).

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.

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

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

`CleanChlist` *If the job of Gibbs sampling from `MultipleSetupAndPosterior` has an error caught by R, the corresponding element of chlist has class try-error. If the job is stopped with, e.g. a segfault, the `chlist` element is NULL.*

Description

If the job of Gibbs sampling from `MultipleSetupAndPosterior` has an error caught by R, the corresponding element of chlist has class try-error. If the job is stopped with, e.g. a segfault, the `chlist` element is NULL.

Usage

```
CleanChlist(chlist, verbose = FALSE)
```

Arguments

`chlist` A list of `hdpSampleChain-class` objects.

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

Value

Invisibly, the clean, non-error `chlist` This is a list of `hdpSampleChain-class` objects.

`CombinePosteriorChains` *Extract components and exposures from multiple posterior sample chains*

Description

Extract components and exposures from multiple posterior sample chains

Usage

```
CombinePosteriorChains(
  clean.chlist,
  input.catalog,
  multi.types,
  cluster.method = "kmedians",
  verbose = TRUE,
  cos.merge = 0.9,
  categ.CI = 0.95,
  exposure.CI = 0.95,
  min.sample = 1
)
```

Arguments

`clean.chlist` It collects the output of multiple independent hdp_posterior calls.

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

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

If `TRUE`, the HDP analysis 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-AdenoCA","SA.Syn.Kidney-RCC")`.

`cluster.method` A `kccaFamily` object input for `flexclust` package

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

`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`.

`categ.CI` A numeric object between 0 and 1. The level of confidence interval used in step 4 of hdp_merge_and_extract_components

`exposure.CI` A numeric object between 0 and 1. The level of confidence interval used in step 5 of hdp_merge_and_extract_components

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

Value

Invisibly, 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).

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.

sum_raw_clusters_after_cos_merge A matrix containing aggregated spectra of raw clusters after cosine similarity merge step in `hdp_merge_and_extract_components`

sum_raw_clusters_after_nonzero_categ A matrix containing aggregated spectra of raw clusters after non-zero category selecting step in `hdp_merge_and_extract_components`

`ExtendBurnin` *Extend Burn in iteration for a list representation of an `hdpState-class` object. This list is an output from `hdp_burnin` or `ActivateandBurnin`.*

Description

Extend Burn in iteration for a list representation of an `hdpState-class` object. This list is an output from `hdp_burnin` or `ActivateandBurnin`.

Usage

```
ExtendBurnin(hdplist, seedNumber = 1, burnin = 4000, cpiter = 3, verbosity = 0)
```

Arguments

`hdplist` A list representation of an `hdpState-class` object

`seedNumber` A random seed for setting the environment of `hdp_burnin`.

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

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

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

Value

A list with hdp object after burn-in iteration and likelihood of iteration

GenerateAverageCluster *Generate average pattern of clusters of each posterior chain from combined list of multiple posterior sample chains*

Description

Generate average pattern of clusters of each posterior chain from combined list of multiple posterior sample chains

Usage

```
GenerateAverageCluster(clean.chlist)
```

Arguments

`clean.chlist` A list of multiple (or one) posterior sample chains.

Value

A list of matrices containing the average pattern of clusters within each posterior chain and a list of matrices containing the sum of each cluster in each posterior chain

Generateppindex *Generate index for a HDP structure and num.tumor.types for other functions*

Description

Generate index for a HDP structure and num.tumor.types for other functions

Usage

```
Generateppindex(multi.types, input.catalog)
```

Arguments

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

If `TRUE`, the HDP analysis 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-AdenoCA","SA.Syn.Kidney-RCC")`.

`input.catalog`

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

MultipleSetupAndPosterior

Activate hierarchical Dirichlet processes and run posterior sampling in parallel.

Description

Activate hierarchical Dirichlet processes and run posterior sampling in parallel.

Usage

```
MultipleSetupAndPosterior(
  input.catalog,
  seedNumber = 1,
  K.guess,
  multi.types = FALSE,
  verbose = TRUE,
  post.burnin = 4000,
  post.n = 50,
  post.space = 50,
  post.cpiter = 3,
  post.verbosity = 0,
  CPU.cores = 1,
  num.child.process = 4,
  gamma.alpha = 1,
  gamma.beta = 1,
  checkpoint.chlist = TRUE
)
```

Arguments

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

`seedNumber`
: A random seeds passed to `dp_activate`.

`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`, The HDP analysis will regard all input spectra as one tumor type.

  If `TRUE`, the HDP analysis 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-AdenoCA","SA.Syn.Kidney-RCC")`.

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

`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`.

`CPU.cores` Number of CPUs to use; there is no point in making this larger than `num.child.process`.

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

`gamma.alpha` Shape parameter of gamma distribution from which the Dirichlet process concentration parameters are drawn; in this function the gamma distributions for all Dirichlet processes are the same.

`gamma.beta` Inverse scale parameter (rate parameter) of gamma distribution from which the Dirichlet process concentration parameters are drawn; in this function the gamma distributions for all Dirichlet processes are the same.

`checkpoint.chlist`
If `TRUE`, checkpoint the (unclean) chlist to "initial.chlist.Rdata" in the current working directory. and checkpoint the clean chlist to "clean.chlist.Rdata" in the current working directory.

Value

Invisibly, the clean `chlist` (output of `CleanChlist`). This is a list of `hdpSampleChain-class` objects.

`ParallelPosteriorafterBurnin`

Generate an HDP Gibbs sampling chain from a spectra catalog.

Description

Generate an HDP Gibbs sampling chain from a spectra catalog.

Usage

```
ParallelPosteriorafterBurnin(
  retval,
  seedNumber = 1,
  verbose = TRUE,
  post.burnin = 4000,
  post.n = 50,
  post.space = 50,
  post.cpiter = 3,
  post.verbosity = 0,
  num.child.process = 2,
  CPU.cores = 2
)
```

Arguments

`retval` A list object containing hdplist after burn-in iteration and likelihood from `BurninIteration`.

`seedNumber` Pass to `hdp_posterior`

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

`post.burnin` Pass to `hdp_posterior` `burnin`. This can be set to a small number

`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`.

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

`CPU.cores` Number of CPUs to use; there is no point in making this larger than `num.child.process`.

Value

Invisibly, an `hdpSampleChain-class` object as returned from `hdp_posterior`.

PrepInit *Initialize hdp object Allocate process index for hdp initialization. Prepare for* `hdp_init`

Description

Initialize hdp object Allocate process index for hdp initialization. Prepare for `hdp_init`

Usage

```
PrepInit(
  multi.types,
  input.catalog,
  verbose,
  K.guess,
  gamma.alpha = 1,
  gamma.beta = 1
)
```

Arguments

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

If `TRUE`, the HDP analysis 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-AdenoCA","SA.Syn.Kidney-RCC")`.

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

verbose
If TRUE then message progress information.

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

gamma.alpha
Shape parameter of gamma distribution from which the Dirichlet process concentration parameters are drawn; in this function the gamma distributions for all Dirichlet processes are the same.

gamma.beta
Inverse scale parameter (rate parameter) of gamma distribution from which the Dirichlet process concentration parameters are drawn; in this function the gamma distributions for all Dirichlet processes are the same.

RunHdpParallel *Extract mutational signatures and optionally compare them to existing signatures and exposures.*

Description

Extract mutational signatures and optionally compare them to existing signatures and exposures.

Usage

```
RunHdpParallel(
  input.catalog,
  seedNumber = 1,
  K.guess,
  multi.types = FALSE,
  verbose = TRUE,
  post.burnin = 4000,
  post.n = 50,
  post.space = 50,
  post.cpiter = 3,
  post.verbosity = 0,
  CPU.cores = 1,
  num.child.process = 4,
  cos.merge = 0.9,
  min.sample = 1,
  categ.CI = 0.95,
  exposure.CI = 0.95,
  cluster.method = "kmedians",
  ground.truth.sig = NULL,
  ground.truth.exp = NULL,
  overwrite = TRUE,
  out.dir = NULL,
  gamma.alpha = 1,
  gamma.beta = 1,
  checkpoint.chlist = TRUE
)
```

Arguments

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

`seedNumber` A random seeds passed to `dp_activate`.

`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`, The HDP analysis will regard all input spectra as one tumor type.

If `TRUE`, the HDP analysis 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-AdenoCA","SA.Syn.Kidney-RCC")`.

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

`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`.

`CPU.cores` Number of CPUs to use; there is no point in making this larger than `num.child.process`.

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

`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`.

`categ.CI` A numeric object between 0 and 1. The level of confidence interval used in step 4 of hdp_merge_and_extract_components

`exposure.CI` A numeric object between 0 and 1. The level of confidence interval used in step 5 of hdp_merge_and_extract_components

`cluster.method`
A `kccaFamily` object input for `flexclust` package

`ground.truth.sig`
Optional. Either a string with the path to file with ground truth signatures or and `ICAMS` catalog with the ground truth signatures. These are the signatures used to construct the ground truth spectra.

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

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

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

`gamma.alpha` Shape parameter of gamma distribution from which the Dirichlet process concentration parameters are drawn; in this function the gamma distributions for all Dirichlet processes are the same.

`gamma.beta` Inverse scale parameter (rate parameter) of gamma distribution from which the Dirichlet process concentration parameters are drawn; in this function the gamma distributions for all Dirichlet processes are the same.

`checkpoint.chlist`

If `TRUE`, checkpoint the (unclean) chlist to "initial.chlist.Rdata" in the current working directory. and checkpoint the clean chlist to "clean.chlist.Rdata" in the current working directory.

Value

Invisibly, 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).

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.

`SetupAndActivate` *Generate an HDP Gibbs sampling chain from a spectra catalog.*

Description

Generate an HDP Gibbs sampling chain from a spectra catalog.

Usage

```
SetupAndActivate(
  input.catalog,
  seedNumber = 1,
  K.guess,
  multi.types = FALSE,
  verbose = TRUE,
  gamma.alpha = 1,
  gamma.beta = 1
)
```

Arguments

`input.catalog`

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

`seedNumber` A random seeds passed to `dp_activate`.

`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, The HDP analysis will regard all input spectra as one tumor type. If TRUE, the HDP analysis 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-AdenoCA","SA.Syn.Kidney-RCC").
verbose	If TRUE then message progress information.
gamma.alpha	Shape parameter of gamma distribution from which the Dirichlet process concentration parameters are drawn; in this function the gamma distributions for all Dirichlet processes are the same.
gamma.beta	Inverse scale parameter (rate parameter) of gamma distribution from which the Dirichlet process concentration parameters are drawn; in this function the gamma distributions for all Dirichlet processes are the same.

Value

Invisibly, an hdpState-class object as returned from dp_activate.

SetupAndPosterior *Generate an HDP Gibbs sampling chain from a spectra catalog.*

Description

Generate an HDP Gibbs sampling chain from a spectra catalog.

Usage

```
SetupAndPosterior(
  input.catalog,
  seedNumber = 1,
  K.guess,
  multi.types = FALSE,
  verbose = TRUE,
  post.burnin = 4000,
  post.n = 50,
  post.space = 50,
  post.cpiter = 3,
  post.verbosity = 0,
  gamma.alpha = 1,
  gamma.beta = 1
)
```

Arguments

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

`seedNumber`
: A random seeds passed to `dp_activate`.

`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`, The HDP analysis will regard all input spectra as one tumor type.

  If `TRUE`, the HDP analysis 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-AdenoCA","SA.Syn.Kidney-RCC")`.

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

`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`.

`gamma.alpha`
: Shape parameter of gamma distribution from which the Dirichlet process concentration parameters are drawn; in this function the gamma distributions for all Dirichlet processes are the same.

`gamma.beta`
: Inverse scale parameter (rate parameter) of gamma distribution from which the Dirichlet process concentration parameters are drawn; in this function the gamma distributions for all Dirichlet processes are the same.

Value

Invisibly, an `hdpSampleChain-class` object as returned from `hdp_posterior`.

Index