Package 'mSigHdp'

October 30, 2020

Title Mutational signature extraction using hdp (Hierarchical Dirichlet Process)

Version 1.1.2001

Description Adapts HDP to mutational signature extraction. Calls hdpx for mutational signature analysis.mSigHdp
incorporates burnin and Gibbs sampling processes, followed up by extracting signatures from raw clusters
collected by Gibbs sampling, and improved diagnostic plotting with visualization on signature and exposures.

License GPL-3

Encoding UTF-8

LazyData true

Language en-US

BuildManual no

biocViews

Roxygen list(markdown = TRUE)

Depends R (>= 4.0)

RoxygenNote 7.1.1

Remotes github::steverozen/hdpx@*release,
github::steverozen/ICAMSxtra@*release

Imports hdpx (>= 0.3.0),
ICAMS (>= 2.2.4),
reshape2

Suggests ICAMSxtra (>= 0.0.2),
testthat,
utils

R topics documented:

`AnalyzeAndPlotretval`

Evaluate and plot retval from `CombinePosteriorChains` *or* `CombineChainsAndExtractSigs` *This function now works for both NR's pipeline and Mo's pipeline*

Description

Evaluate and plot retval from `CombinePosteriorChains` or `CombineChainsAndExtractSigs` This function now works for both NR's pipeline and Mo's pipeline

Usage

```
AnalyzeAndPlotretval(
  retval,
  input.catalog,
  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` or `CombineChainsAndExtractSigs`
`input.catalog`	input catalog matrix or path to file with input catalog
`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

---

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

Description

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

Usage

```
ChainBurnin(
  hdp.state,
  seedNumber = 1,
  burnin = 5000,
  cpiter = 3,
  burnin.verbosity = 0,
  burnin.multiplier = 2,
  burnin.checkpoint = TRUE
)
```

Arguments

hdp.state
: An hdpState-class object or a list representation of an hdpState-class object.

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

burnin
: Pass to hdp_burnin burnin. The number of burn-in iterations

cpiter
: Pass to hdp_burnin cpiter. The number of iterations of concentration parameter sampling to perform after each iteration.

burnin.verbosity
: Pass to hdp_burnin verbosity.Verbosity of debugging statements.

burnin.multiplier
: A checkpoint setting. burnin.multiplier rounds of burnin iterations will be run. After each round, a burn-in chain will be save for checkpoint. A total number of 10,000 iterations is recommended for most analysis. However, number of iterations can be adjusted based on the size of dataset. According to our experience, 20,000 iterations are needed when analyzing all PCAWG7 genomes (2,780 samples). The burnin can be continued from a checkpoint file with ExtendBurnin.

burnin.checkpoint
: Default is False. If True, a checkpoint for burnin will be created.

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.

`ChainsDiagnosticPlot` *Diagnostic plot for a hdpSampleMulti object*

Description

Diagnostic plot for a hdpSampleMulti object

Usage

```
ChainsDiagnosticPlot(retval, input.catalog, 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.
`input.catalog`	ground truth catalog
`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. Return from MultipleSetupAndPosterior.
verbose	If TRUE then message progress information.

Value

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

CombineChainsAndExtractSigs

Extract components and exposures from multiple posterior sample chains This function returns signatures with high confidence (found in more than 90% #' posterior samples)

Description

Extract components and exposures from multiple posterior sample chains This function returns signatures with high confidence (found in more than 90% #' posterior samples)

Usage

```
CombineChainsAndExtractSigs(
  clean.chlist,
  input.catalog,
  multi.types,
  verbose = TRUE,
  cos.merge = 0.9,
  confident.prop = 0.9,
  noise.prop = 0.1,
  hc.cutoff = 0.1
)
```

Arguments

clean.chlist	A list of hdpSampleChain-class objects. Each element is the result of one posterior sample chain.
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. HDP structure as one parent node for all tumors 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" HDP structure as a grandparent node for whole data and one parent node for each tumor type

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.

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

`confident.prop` Pass to `interpret_components`. clusters with at least `confident.prop` of posterior samples are high confident signatures

`noise.prop` Pass to `interpret_components`. Clusters with less than `noise.prop` of posterior samples are noise signatures

`hc.cutoff` Pass to `extract_components_from_clusters`. The cutoff of height of hierarchical clustering dendrogram

Value

Invisibly, a list with the following elements:

signature The extracted signature profiles as a matrix; rows are mutation types, columns are signatures including high confident signatures -'hdp' signatures and moderate confident signatures - 'potential hdp' signatures.

signature.post.samp.number A dataframe with two columns. The first column corresponds to each signature in `signature` and the second columns contains the number of posterior samples that found the raw clusters contributing to the signature.

signature.cdc A `comp_dp_counts` like dataframe. Each column corresponds to the sum of all `comp_dp_counts` matrices of the raw clusters contributing to each signature in codesignature

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

noise.signature The extracted signature profiles as a matrix; rows are mutation types, columns are signatures with less than `noise.prop` of posterior samples.

noise.post.samp.number A data frame with two columns. The first column corresponds to each signature in `noise.signature` and the second columns contains the number of posterior samples that found the raw clusters contributing to the signature.

noise.cdc A `comp_dp_counts` like data frame. Each column corresponds to the sum of all `comp_dp_counts` matrices of the raw clusters contributing to each signature in codenoise.signature

extracted.retval A list object returned from codeinterpret_components.

`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,
  verbose = TRUE,
  cos.merge = 0.9,
  categ.CI = 0.95,
  exposure.CI = 0.95,
  min.sample = 1,
  diagnostic.folder = NULL
)
```

Arguments

clean.chlist	A list of hdpSampleChain-class objects. Each element is the result of one posterior sample chain.
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").
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 number the range $[0, 1]$. The level of the confidence interval used in step 4 of hdp_merge_and_extract_components. This governs when "averaged raw cluster" get assigned to component 0, i.e. if the the confidence interval overlaps 0. Lower values make it less likely that an averaged raw cluster will be assigned to component 0. The CI in question is for the number of mutations in a given mutation class (e.g. ACA > AAA, internally called a "category"). If, for every mutation class, this CI overlaps 0, then the averaged raw cluster goes to component 0.
exposure.CI	A number in the range $[0, 1]$. The level of the confidence interval used in step 5 of hdp_merge_and_extract_components. The CI in question here for the total number of mutations assigned to an averaged raw cluster.
min.sample	A "component" (i.e. signature) must have at least this many samples; passed to hdp_merge_and_extract_components.
diagnostic.folder	If provided, diagnostic plots for hdp.0 components are provided

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`.

clust_hdp0_ccc4 A matrix containing aggregated spectra of raw clusters moving to hdp.0 after non-zero category selection step in `hdp_merge_and_extract_components`.

clust_hdp0_ccc5 A matrix containing aggregated spectra of raw clusters moving to hdp.0 after non-zero observation selection step in `hdp_merge_and_extract_components`.

`ComponentDiagnosticPlotting` *Diagnostic plot for a hdpSampleMulti object. This function is compatible with the return object from Liu's extract_components_from_clusters*

Description

Diagnostic plot for a hdpSampleMulti object. This function is compatible with the return object from Liu's extract_components_from_clusters

Usage

```
ComponentDiagnosticPlotting(retval, input.catalog, out.dir, verbose)
```

Arguments

`retval`	Return from `CombineChainsAndExtractSigs`
`input.catalog`	Input spectra catalog as a matrix or in `ICAMS` format.
`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.

`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. HDP structure as one parent node for all tumors If TRUE, the HDP analysis will infer tumor types based on the string before "::" in their names. e.g. tumor type for "Ovary-AdenoCA::S.500" would be "Ovary-AdenoCA" HDP structure as a grandparent node for whole data and one parent node for each tumor type 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. If not FALSE, HDP will give a parent node for each tumor type and a grandparent node for the whole dataset. e.g. c("Ovary-AdenoCA","Kidney-RCC").
input.catalog	Input spectra catalog as a matrix or in ICAMS format.

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

Description

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

Usage

```
GeneratePriorppindex(multi.types, input.catalog, nps)
```

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.
nps	Number of prior signatures

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,
  burnin = 5000,
  burnin.multiplier = 2,
  burnin.checkpoint = TRUE,
  post.n = 200,
  post.space = 100,
  post.cpiter = 3,
  post.verbosity = 0,
  CPU.cores = 20,
  num.child.process = 20,
  gamma.alpha = 1,
  gamma.beta = 20,
  gamma0.alpha = gamma.alpha,
  gamma0.beta = gamma.beta,
  checkpoint.chlist = TRUE,
  checkpoint.1.chain = TRUE,
  prior.sigs = NULL,
  prior.pseudoc = NULL
)
```

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 clusters. Usually, the number of clusters is two times of the number of extracted 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. HDP structure as one parent node for all tumors If TRUE, the HDP analysis will infer tumor types based on the string before "::" in their names. e.g. tumor type for "Ovary-AdenoCA::S.500" would be "Ovary-AdenoCA" HDP structure as a grandparent node for whole data and one parent node for each tumor type 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. If not FALSE, HDP will give a parent node for each tumor type and a grandparent node for the whole dataset. e.g. c("Ovary-AdenoCA","Kidney-RCC").
verbose	If TRUE then message progress information.
burnin	Pass to hdp_burnin burnin. The number of burn-in iterations
burnin.multiplier	A checkpoint setting. burnin.multiplier rounds of burnin iterations will be run. After each round, a burn-in chain will be save for checkpoint. A total number of 10,000 iterations is recommended for most analysis. However, number of iterations can be adjusted based on the size of dataset. According to our experience, 20,000 iterations are needed when analyzing all PCAWG7 genomes (2,780 samples). The burnin can be continued from a checkpoint file with ExtendBurnin.
burnin.checkpoint	Default is False. If True, a checkpoint for burnin will be created.
post.n	Pass to hdp_posterior_sample n.The number of posterior samples to collect.
post.space	Pass to hdp_posterior_sample space. The number of iterations between collected samples.
post.cpiter	Pass to hdp_posterior_sample and hdp_burnin cpiter.The number of iterations of concentration parameter sampling to perform after each iteration
post.verbosity	Pass to hdp_posterior_sample verbosity. Verbosity of debugging statements. No need to change unless for development purpose
CPU.cores	Number of CPUs to use; this should be no more than num.child.process.
num.child.process	Number of posterior sampling chains; can set to 1 for testing. We recommend 20 for real data analysis
gamma.alpha	Shape parameter of the gamma distribution prior for the Dirichlet process concentration parameters; in this function the gamma distributions for all Dirichlet processes, except possibly the top level process, are the same.

`gamma.beta` Inverse scale parameter (rate parameter) of the gamma distribution prior for the Dirichlet process concentration parameters; in this function the gamma distributions for all Dirichlet processes, except possibly the top level process, are the same.

We recommend gamma.alpha = 1 and gamma.beta = 20 for single-base-substitution signatures extraction; gamma.alpha = 1 and gamma.beta = 50 for doublet-base-substitution/INDEL signature extraction

`gamma0.alpha` See figure B.1 from Nicola Robert's thesis. The shape parameter (α_0) of the gamma distribution priors for the Dirichlet process concentration parameters (γ_0) for G_0.

`gamma0.beta` See figure B.1 from Nicola Robert's thesis. Inverse scale parameter (rate parameter, β_0) of the gamma distribution priors for the Dirichlet process concentration parameters (γ_0) for G_0.

`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.

`checkpoint.1.chain`
If `TRUE` checkpoint the sample chain to current working directory, in a file called sample.chain.*seed_number*.Rdata.

`prior.sigs` A matrix containing prior signatures.

`prior.pseudoc`
A numeric list. Pesudo counts of each prior signature. Recommended is 1000. In practice, it may be advisable to put lower weights on prior signatures that you do not expect to be present in your dataset, or even exclude some priors entirely.

Value

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

`OldRunHdpParallel` *Deprecated, extract mutational signatures and optionally compare them to existing signatures and exposures.*

Description

Deprecated, This functions uses the original method of combining raw clusters into "components". Use `RunHdpxParallel` instead.

Usage

```
OldRunHdpParallel(
  input.catalog,
  seedNumber = 1,
  K.guess,
  multi.types = FALSE,
  verbose = TRUE,
  post.burnin = 5000,
```

```
  post.n = 200,
  post.space = 100,
  post.cpiter = 3,
  post.verbosity = 0,
  CPU.cores = 20,
  num.child.process = 20,
  cos.merge = 0.9,
  min.sample = 1,
  categ.CI = 0.95,
  exposure.CI = 0.95,
  ground.truth.sig = NULL,
  ground.truth.exp = NULL,
  overwrite = TRUE,
  out.dir = NULL,
  gamma.alpha = 1,
  gamma.beta = 20,
  gamma0.alpha = gamma.alpha,
  gamma0.beta = gamma.beta,
  checkpoint.chlist = TRUE,
  checkpoint.1.chain = TRUE,
  prior.sigs = NULL,
  prior.pseudoc = NULL,
  burnin.multiplier = 2,
  burnin.checkpoint = 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 clusters. Usually, the number of clusters is two times of the number of extracted 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. HDP structure as one parent node for all tumors If `TRUE`, the HDP analysis will infer tumor types based on the string before "::" in their names. e.g. tumor type for "Ovary-AdenoCA::S.500" would be "Ovary-AdenoCA" HDP structure as a grandparent node for whole data and one parent node for each tumor type 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`. If not `FALSE`, HDP will give a parent node for each tumor type and a grandparent node for the whole dataset. e.g. `c("Ovary-AdenoCA","Kidney-RCC")`.
`verbose`	If `TRUE` then `message` progress information.
`post.n`	Pass to `hdp_posterior_sample` `n`.The number of posterior samples to collect.

post.space Pass to hdp_posterior_sample space. The number of iterations between collected samples.

post.cpiter Pass to hdp_posterior_sample and hdp_burnin cpiter.The number of iterations of concentration parameter sampling to perform after each iteration

post.verbosity
Pass to hdp_posterior_sample verbosity. Verbosity of debugging statements. No need to change unless for development purpose

CPU.cores Number of CPUs to use; this should be no more than num.child.process.

num.child.process
Number of posterior sampling chains; can set to 1 for testing. We recommend 20 for real data analysis

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_merge_and_extract_components.

categ.CI A number the range $[0, 1]$. The level of the confidence interval used in step 4 of hdp_merge_and_extract_components. This governs when "averaged raw cluster" get assigned to component 0, i.e. if the the confidence interval overlaps 0. Lower values make it less likely that an averaged raw cluster will be assigned to component 0. The CI in question is for the number of mutations in a given mutation class (e.g. ACA > AAA, internally called a "category"). If, for every mutation class, this CI overlaps 0, then the averaged raw cluster goes to component 0.

exposure.CI A number in the range $[0, 1]$. The level of the confidence interval used in step 5 of hdp_merge_and_extract_components. The CI in question here for the total number of mutations assigned to an averaged raw cluster.

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 the gamma distribution prior for the Dirichlet process concentration parameters; in this function the gamma distributions for all Dirichlet processes, except possibly the top level process, are the same.

gamma.beta Inverse scale parameter (rate parameter) of the gamma distribution prior for the Dirichlet process concentration parameters; in this function the gamma distributions for all Dirichlet processes, except possibly the top level process, are the same.

We recommend gamma.alpha = 1 and gamma.beta = 20 for single-base-substitution signatures extraction; gamma.alpha = 1 and gamma.beta = 50 for doublet-base-substitution/INDEL signature extraction

gamma0.alpha See figure B.1 from Nicola Robert's thesis. The shape parameter (α_0) of the gamma distribution priors for the Dirichlet process concentration parameters (γ_0) for G_0.

`gamma0.beta` See figure B.1 from Nicola Robert's thesis. Inverse scale parameter (rate parameter, β_0) of the gamma distribution priors for the Dirichlet process concentration parameters (γ_0) for G_0.

`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.

`checkpoint.1.chain`
If `TRUE` checkpoint the sample chain to current working directory, in a file called sample.chain.*seed_number*.Rdata.

`prior.sigs` A matrix containing prior signatures.

`prior.pseudoc`
A numeric list. Pesudo counts of each prior signature. Recommended is 1000. In practice, it may be advisable to put lower weights on prior signatures that you do not expect to be present in your dataset, or even exclude some priors entirely.

`burnin.multiplier`
A checkpoint setting. `burnin.multiplier` rounds of `burnin` iterations will be run. After each round, a burn-in chain will be save for checkpoint. A total number of 10,000 iterations is recommended for most analysis. However, number of iterations can be adjusted based on the size of dataset. According to our experience, 20,000 iterations are needed when analyzing all PCAWG7 genomes (2,780 samples). The burnin can be continued from a checkpoint file with `ExtendBurnin`.

`burnin.checkpoint`
Default is False. If True, a checkpoint for burnin will be created.

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`.

clust_hdp0_ccc4 A matrix containing aggregated spectra of raw clusters moving to hdp.0 after non-zero category selection step in `hdp_merge_and_extract_components`.

clust_hdp0_ccc5 A matrix containing aggregated spectra of raw clusters moving to hdp.0 after non-zero observation selection step in `hdp_merge_and_extract_components`.

PlotSamplesHighSigExp

Plot hdp signature exposure in each sample. This function returns the plot of top 5 samples with the highest exposure to a signature. Each spectrum's title is in the form of: SampleName(Proportion of Signature Assginment) This function is here because it is specific for signature extraction application.

Description

Plot hdp signature exposure in each sample. This function returns the plot of top 5 samples with the highest exposure to a signature. Each spectrum's title is in the form of: SampleName(Proportion of Signature Assginment) This function is here because it is specific for signature extraction application.

Usage

```
PlotSamplesHighSigExp(
  retval,
  hdpsample,
  input.catalog,
  col_comp = NULL,
  incl_numdata_plot = TRUE,
  ylab_numdata = "Number of data items",
  ylab_exp = "Component exposure",
  leg.title = "Component",
  cex.names = 0.6,
  cex.axis = 0.7,
  mar = c(1, 4, 2, 0.5),
  oma = c(1.5, 1.5, 1, 1)
)
```

Arguments

retval	An object return from extract_ccc_cdc_from_hdp
hdpsample	A hdpSampleChain-class or hdpSampleMulti-class object including output from extract_components_from_clusters
input.catalog	Input spectra catalog as a matrix or in ICAMS format.
col_comp	Colours of each component, from 0 to the max number. If NULL, default colors will be used
incl_numdata_plot	Logical - should an upper barplot indicating the number of data items per DP be included? (Default TRUE)
ylab_numdata	Vertical axis label for numdata plot
ylab_exp	Vertical exis label for exposure plot
leg.title	Legend title
cex.names	Expansion factor for bar labels (dpnames) in exposure plot
cex.axis	Expansion factor for vertical-axis annotation

mar	See ?par
oma	See ?par

`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 = TRUE,
  K.guess,
  gamma.alpha = 1,
  gamma.beta = 1,
  gamma0.alpha = gamma.alpha,
  gamma0.beta = gamma.beta
)
```

Arguments

`multi.types`	A logical scalar or a character vector. If `FALSE`, The HDP analysis will regard all input spectra as one tumor type. HDP structure as one parent node for all tumors If `TRUE`, the HDP analysis will infer tumor types based on the string before "::" in their names. e.g. tumor type for "Ovary-AdenoCA::S.500" would be "Ovary-AdenoCA" HDP structure as a grandparent node for whole data and one parent node for each tumor type 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`. If not `FALSE`, HDP will give a parent node for each tumor type and a grandparent node for the whole dataset. e.g. `c("Ovary-AdenoCA","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 clusters. Usually, the number of clusters is two times of the number of extracted signatures. Passed to `dp_activate` as `initcc`.
`gamma.alpha`	Shape parameter of the gamma distribution prior for the Dirichlet process concentration parameters; in this function the gamma distributions for all Dirichlet processes, except possibly the top level process, are the same.

`gamma.beta` Inverse scale parameter (rate parameter) of the gamma distribution prior for the Dirichlet process concentration parameters; in this function the gamma distributions for all Dirichlet processes, except possibly the top level process, are the same.

We recommend gamma.alpha = 1 and gamma.beta = 20 for single-base-substitution signatures extraction; gamma.alpha = 1 and gamma.beta = 50 for doublet-base-substitution/INDEL signature extraction

`gamma0.alpha` See figure B.1 from Nicola Robert's thesis. The shape parameter (α_0) of the gamma distribution priors for the Dirichlet process concentration parameters (γ_0) for G_0.

`gamma0.beta` See figure B.1 from Nicola Robert's thesis. Inverse scale parameter (rate parameter, β_0) of the gamma distribution priors for the Dirichlet process concentration parameters (γ_0) for G_0.

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

Description

Generate an HDP Gibbs sampling chain from a spectra catalog.

Usage

```
PriorSetupAndActivate(
  prior.sigs,
  prior.pseudoc,
  gamma.alpha = 1,
  gamma.beta = 1,
  K.guess,
  gamma0.alpha = gamma.alpha,
  gamma0.beta = gamma.beta,
  multi.types = F,
  input.catalog,
  verbose = TRUE,
  seedNumber = 1
)
```

Arguments

`prior.sigs` A matrix containing prior signatures.

`prior.pseudoc`
A numeric list. Pesudo counts of each prior signature. Recommended is 1000. In practice, it may be advisable to put lower weights on prior signatures that you do not expect to be present in your dataset, or even exclude some priors entirely.

`gamma.alpha` Shape parameter of the gamma distribution prior for the Dirichlet process concentration parameters; in this function the gamma distributions for all Dirichlet processes, except possibly the top level process, are the same.

gamma.beta	Inverse scale parameter (rate parameter) of the gamma distribution prior for the Dirichlet process concentration parameters; in this function the gamma distributions for all Dirichlet processes, except possibly the top level process, are the same.
K.guess	Suggested initial value of the number of signatures, passed to dp_activate as initcc.
gamma0.alpha	See figure B.1 from Nicola Robert's thesis. The shape parameter (α_0) of the gamma distribution priors for the Dirichlet process concentration parameters (γ_0) for G_0.
gamma0.beta	See figure B.1 from Nicola Robert's thesis. Inverse scale parameter (rate parameter, β_0) of the gamma distribution priors for the Dirichlet process concentration parameters (γ_0) for G_0.
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.
seedNumber	A random seeds passed to dp_activate.

Value

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

RunHdpxParallel *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

```
RunHdpxParallel(
  input.catalog,
  seedNumber = 123,
  K.guess,
  multi.types = FALSE,
  verbose = TRUE,
  burnin = 5000,
  burnin.multiplier = 2,
  burnin.checkpoint = FALSE,
```

```
  post.n = 200,
  post.space = 100,
  post.cpiter = 3,
  post.verbosity = 0,
  CPU.cores = 20,
  num.child.process = 20,
  cos.merge = 0.9,
  confident.prop = 0.9,
  noise.prop = 0.5,
  hc.cutoff = 0.1,
  ground.truth.sig = NULL,
  ground.truth.exp = NULL,
  overwrite = TRUE,
  out.dir = NULL,
  gamma.alpha = 1,
  gamma.beta = 20,
  gamma0.alpha = gamma.alpha,
  gamma0.beta = gamma.beta,
  checkpoint.chlist = TRUE,
  checkpoint.1.chain = TRUE,
  prior.sigs = NULL,
  prior.pseudoc = NULL
)
```

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 clusters. Usually, the number of clusters is two times of the number of extracted 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. HDP structure as one parent node for all tumors If `TRUE`, the HDP analysis will infer tumor types based on the string before "::" in their names. e.g. tumor type for "Ovary-AdenoCA::S.500" would be "Ovary-AdenoCA" HDP structure as a grandparent node for whole data and one parent node for each tumor type 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`. If not `FALSE`, HDP will give a parent node for each tumor type and a grandparent node for the whole dataset. e.g. `c("Ovary-AdenoCA","Kidney-RCC")`.
`verbose`	If `TRUE` then `message` progress information.
`burnin`	Pass to `hdp_burnin` `burnin`. The number of burn-in iterations
`burnin.multiplier`	A checkpoint setting. `burnin.multiplier` rounds of `burnin` iterations will be run. After each round, a burn-in chain will be save for checkpoint. A total number of 10,000 iterations is recommended for most analysis. However,

number of iterations can be adjusted based on the size of dataset. According to our experience, 20,000 iterations are needed when analyzing all PCAWG7 genomes (2,780 samples). The burnin can be continued from a checkpoint file with ExtendBurnin.

burnin.checkpoint
: Default is False. If True, a checkpoint for burnin will be created.

post.n
: Pass to hdp_posterior_sample n.The number of posterior samples to collect.

post.space
: Pass to hdp_posterior_sample space. The number of iterations between collected samples.

post.cpiter
: Pass to hdp_posterior_sample and hdp_burnin cpiter.The number of iterations of concentration parameter sampling to perform after each iteration

post.verbosity
: Pass to hdp_posterior_sample verbosity. Verbosity of debugging statements. No need to change unless for development purpose

CPU.cores
: Number of CPUs to use; this should be no more than num.child.process.

num.child.process
: Number of posterior sampling chains; can set to 1 for testing. We recommend 20 for real data analysis

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

confident.prop
: Pass to interpret_components. clusters with at least confident.prop of posterior samples are high confident signatures

noise.prop
: Pass to interpret_components. Clusters with less than noise.prop of posterior samples are noise signatures

hc.cutoff
: Pass to extract_components_from_clusters. The cutoff of height of hierarchical clustering dendrogram

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 the gamma distribution prior for the Dirichlet process concentration parameters; in this function the gamma distributions for all Dirichlet processes, except possibly the top level process, are the same.

gamma.beta
: Inverse scale parameter (rate parameter) of the gamma distribution prior for the Dirichlet process concentration parameters; in this function the gamma distributions for all Dirichlet processes, except possibly the top level process, are the same.

  We recommend gamma.alpha = 1 and gamma.beta = 20 for single-base-substitution signatures extraction; gamma.alpha = 1 and gamma.beta = 50 for doublet-base-substitution/INDEL signature extraction

`gamma0.alpha` See figure B.1 from Nicola Robert's thesis. The shape parameter (α_0) of the gamma distribution priors for the Dirichlet process concentration parameters (γ_0) for G_0.

`gamma0.beta` See figure B.1 from Nicola Robert's thesis. Inverse scale parameter (rate parameter, β_0) of the gamma distribution priors for the Dirichlet process concentration parameters (γ_0) for G_0.

`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.

`checkpoint.1.chain`
If `TRUE` checkpoint the sample chain to current working directory, in a file called sample.chain.*seed_number*.Rdata.

`prior.sigs` A matrix containing prior signatures.

`prior.pseudoc`
A numeric list. Pesudo counts of each prior signature. Recommended is 1000. In practice, it may be advisable to put lower weights on prior signatures that you do not expect to be present in your dataset, or even exclude some priors entirely.

Value

Invisibly, a list with the following elements:

signature The extracted signature profiles as a matrix; rows are mutation types, columns are signatures including high confident signatures -'hdp' signatures and moderate confident signatures - 'potential hdp' signatures.

signature.post.samp.number A dataframe with two columns. The first column corresponds to each signature in `signature` and the second columns contains the number of posterior samples that found the raw clusters contributing to the signature.

signature.cdc A `comp_dp_counts` like dataframe. Each column corresponds to the sum of all `comp_dp_counts` matrices of the raw clusters contributing to each signature in codesignature

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

noise.signature The extracted signature profiles as a matrix; rows are mutation types, columns are signatures with less than `noise.prop` of posterior samples

noise.post.samp.number A data frame with two columns. The first column corresponds to each signature in `noise.signature` and the second column contains the number of posterior samples that found the raw clusters contributing to the signature.

noise.cdc A `comp_dp_counts` like data frame. Each column corresponds to the sum of all `comp_dp_counts` matrices of the raw clusters contributing to each signature in codenoise.signature

extracted.retval A list object returned from codeinterpret_components.

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,
  gamma0.alpha = gamma.alpha,
  gamma0.beta = gamma.beta
)
```

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 clusters. Usually, the number of clusters is two times of the number of extracted 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. HDP structure as one parent node for all tumors If TRUE, the HDP analysis will infer tumor types based on the string before "::" in their names. e.g. tumor type for "Ovary-AdenoCA::S.500" would be "Ovary-AdenoCA" HDP structure as a grandparent node for whole data and one parent node for each tumor type 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. If not FALSE, HDP will give a parent node for each tumor type and a grandparent node for the whole dataset. e.g. c("Ovary-AdenoCA","Kidney-RCC").
verbose	If TRUE then message progress information.
gamma.alpha	Shape parameter of the gamma distribution prior for the Dirichlet process concentration parameters; in this function the gamma distributions for all Dirichlet processes, except possibly the top level process, are the same.

`gamma.beta` Inverse scale parameter (rate parameter) of the gamma distribution prior for the Dirichlet process concentration parameters; in this function the gamma distributions for all Dirichlet processes, except possibly the top level process, are the same.

We recommend gamma.alpha = 1 and gamma.beta = 20 for single-base-substitution signatures extraction; gamma.alpha = 1 and gamma.beta = 50 for doublet-base-substitution/INDEL signature extraction

`gamma0.alpha` See figure B.1 from Nicola Robert's thesis. The shape parameter (α_0) of the gamma distribution priors for the Dirichlet process concentration parameters (γ_0) for G_0.

`gamma0.beta` See figure B.1 from Nicola Robert's thesis. Inverse scale parameter (rate parameter, β_0) of the gamma distribution priors for the Dirichlet process concentration parameters (γ_0) for G_0.

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,
  burnin = 5000,
  post.n = 50,
  post.space = 50,
  cpiter = 3,
  post.verbosity = 0,
  gamma.alpha = 1,
  gamma.beta = 20,
  gamma0.alpha = gamma.alpha,
  gamma0.beta = gamma.beta,
  checkpoint.1.chain = TRUE,
  burnin.multiplier = 2,
  burnin.checkpoint = TRUE,
  prior.sigs = NULL,
  prior.pseudoc = NULL
)
```

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 clusters. Usually, the number of clusters is two times of the number of extracted 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. HDP structure as one parent node for all tumors

  If `TRUE`, the HDP analysis will infer tumor types based on the string before "::" in their names. e.g. tumor type for "Ovary-AdenoCA::S.500" would be "Ovary-AdenoCA" HDP structure as a grandparent node for whole data and one parent node for each tumor type

  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`.

  If not `FALSE`, HDP will give a parent node for each tumor type and a grandparent node for the whole dataset.

  e.g. `c("Ovary-AdenoCA","Kidney-RCC")`.

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

`burnin`
: Pass to `hdp_burnin` `burnin`. The number of burn-in iterations

`post.n`
: Pass to `hdp_posterior_sample` `n`.The number of posterior samples to collect.

`post.space`
: Pass to `hdp_posterior_sample` `space`. The number of iterations between collected samples.

`cpiter`
: Pass to `hdp_burnin` `cpiter`. The number of iterations of concentration parameter sampling to perform after each iteration.

`post.verbosity`
: Pass to `hdp_posterior_sample` `verbosity`. Verbosity of debugging statements. No need to change unless for development purpose

`gamma.alpha`
: Shape parameter of the gamma distribution prior for the Dirichlet process concentration parameters; in this function the gamma distributions for all Dirichlet processes, except possibly the top level process, are the same.

`gamma.beta`
: Inverse scale parameter (rate parameter) of the gamma distribution prior for the Dirichlet process concentration parameters; in this function the gamma distributions for all Dirichlet processes, except possibly the top level process, are the same.

  We recommend gamma.alpha = 1 and gamma.beta = 20 for single-base-substitution signatures extraction; gamma.alpha = 1 and gamma.beta = 50 for doublet-base-substitution/INDEL signature extraction

`gamma0.alpha`
: See figure B.1 from Nicola Robert's thesis. The shape parameter (α_0) of the gamma distribution priors for the Dirichlet process concentration parameters (γ_0) for G_0.

`gamma0.beta`
: See figure B.1 from Nicola Robert's thesis. Inverse scale parameter (rate parameter, β_0) of the gamma distribution priors for the Dirichlet process concentration parameters (γ_0) for G_0.

`checkpoint.1.chain`
: If `TRUE` checkpoint the sample chain to current working directory, in a file called sample.chain.*seed_number*.Rdata.

`burnin.multiplier`
: A checkpoint setting. `burnin.multiplier` rounds of `burnin` iterations will be run. After each round, a burn-in chain will be save for checkpoint. A total number of 10,000 iterations is recommended for most analysis. However, number of iterations can be adjusted based on the size of dataset. According to our experience, 20,000 iterations are needed when analyzing all PCAWG7 genomes (2,780 samples). The burnin can be continued from a checkpoint file with `ExtendBurnin`.

`burnin.checkpoint`
: Default is False. If True, a checkpoint for burnin will be created.

`prior.sigs`
: A matrix containing prior signatures.

`prior.pseudoc`
: A numeric list. Pesudo counts of each prior signature. Recommended is 1000. In practice, it may be advisable to put lower weights on prior signatures that you do not expect to be present in your dataset, or even exclude some priors entirely.

`post.cpiter`
: Pass to `hdp_posterior_sample` and `hdp_burnin` `cpiter`.The number of iterations of concentration parameter sampling to perform after each iteration

Value

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

`test.ground.truth.sig` *test.ground.truth.sig*

Description

Ground truth signature for testing.

Ground truth exposure for testing.

Usage

```
test.ground.truth.sig

test.ground.truth.exposure
```

Format

An `ICAMS` catalog (each column is a COSMICv3 signature, each row is a mutation type, e.g. ACT > AGT).

An exposure matrix (each row is a COSMICv3 signature used to generate `test.spectra` , each column is a sample).

test.spectra *test.spectra*

Description

Synthetic SBS96 spectra for testing.

Usage

```
test.spectra
```

Format

An ICAMS catalog (each column is a sample, each row is a mutation type, e.g. ACT > AGT).

Index