Package ‘mSigHdp’

January 11, 2022

Title Mutational signature extraction using hdp (Hierarchical Dirichlet Process)

Version 1.2.1

Description Mutational signature discovery using hierarchichal Dirichlet
process mixture modeling. mSigHdp stands for 'mutational
signature (discovery using) hierarchical dirichlet processes.
This packages uses https://github.com/steverozen/hdpx for the
hierarchical Dirichlet process implementation.

License GPL-3

Encoding UTF-8

Language en-US

BuildManual no

biocViews

Roxygen list(markdown = TRUE)

Depends R (>= 4.0)

RoxygenNote 7.1.2

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

Imports hdpx (>= 0.3.8),
ICAMS (>= 2.2.4),
reshape2,
data.table

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,
  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.

`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. Therefore we set the default of `burnin` to 1000 and `burnin.multiplier` to 10. However, number of iterations can be adjusted based on the size of dataset. The dataset with more mutations require longer burn-ins. According to our experience, 50,000 iterations are needed when analyzing all PCAWG7 genomes (2,780 samples). The burnin can be continued from a checkpoint file with ExtendBurnin.

`burnin.checkpoint` If `TRUE`, a checkpoint for burn-in 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.

`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.

`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,
  verbose = TRUE,
  high.confidence.prop = 0.9,
  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.
`verbose`	If `TRUE` then `message` progress information.
`high.confidence.prop`	Pass to `interpret_components`. raw clusters (mutation cluster) found in >= `high.confidence.prop` proportion of posterior samples are signatures with high confidence.
`hc.cutoff`	Pass to `extract_components_from_clusters`. The cutoff of height of hierarchical clustering dendrogram, used in combining raw clusters (mutation clusters) into agreggated clusters.

Value

Invisibly, a list with the following elements:

signature The extracted signature profiles as a matrix; rows are mutation types, columns are signatures with high confidence.

signature.post.samp.number A data frame 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 numeric data frame. Each column corresponds to the sum of all mutations contributing to each signature in `signature`

exposureProbs The inferred exposures as a matrix of mutation probabilities; rows are signatures, columns are samples (e.g. tumors). This is similar to `signature.cdc` but every column was normalized to sum of 1

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

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

low.confidence.cdc A numeric data frame. Each column corresponds to the sum of all mutations contributing to each signature in `low.confidence.signature`

extracted.retval A list object returned from codeextract_components_from_clusters.

`ComponentDiagnosticPlotting`

Generate multiple plots for for a hdpSampleMulti object.

Description

Generate multiple plots for for a hdpSampleMulti object.

Usage

```
ComponentDiagnosticPlotting(
  retval,
  input.catalog,
  out.dir,
  verbose,
  IS.ICAMS = T
)
```

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.

`IS.ICAMS` If TRUE, then plot diagnostics.hdp.signature.exposure.each.sample.pdf.

Details

Generates the plots diagnostics.hdp.signature.exposure.each.sample.pdf, diagnostics.component.distribution.in.posterior. diagnostics.likelihood.pdf, diagnostics.numcluster.pdf, diagnostics.signatures.pdf

---

`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, and the HDP structure will have one parent node for all tumors.

If `TRUE`, Sample IDs in `input.catalog` must have the form *sample_type*::*sample_id*.

If a character vector, then its length must be `ncol(input.catalog)`, and each value is the sample type of the corresponding column in `input.catalog`, e.g. `c(rep("Type-A",23),rep("Type-B",10))` for 23 Type-A samples and 10 Type-B samples.

If not `FALSE`, HDP will have one parent node for each sample type and one grandparent node.

`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,
  posterior.checkpoint = FALSE
)
```

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, and the HDP structure will have one parent node for all tumors.

  If `TRUE`, Sample IDs in `input.catalog` must have the form *sample_type*::*sample_id*.

  If a character vector, then its length must be `ncol(input.catalog)`, and each value is the sample type of the corresponding column in `input.catalog`, e.g. `c(rep("Type-A",23),rep("Type-B",10))` for 23 Type-A samples and 10 Type-B samples.

  If not `FALSE`, HDP will have one parent node for each sample type and one grandparent node.

`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. Therefore we set the default of `burnin` to 1000 and `burnin.multiplier` to 10. However, number of iterations can be adjusted based on the size of dataset. The dataset with more mutations require longer burn-ins. According to our experience, 50,000 iterations are needed when analyzing all PCAWG7 genomes (2,780 samples). The burnin can be continued from a checkpoint file with `ExtendBurnin`.

`burnin.checkpoint`
: If `TRUE`, a checkpoint for burn-in 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.

`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.

`posterior.checkpoint`
If `TRUE` checkpoint the posterior sampling after every 10 posterior samples collected

Value

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

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 = F,
  ylab_numdata = "Number of data items",
  ylab_exp = "Component exposure",
  leg.title = "Component",
  cex.names = 0.6,
  cex.axis = 0.7,
  mar = c(4, 4, 2, 0.5),
  oma = c(1.5, 1.5, 1, 1)
)
```

Arguments

`retval` An object return from `extract_ccc_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, and the HDP structure will have one parent node for all tumors.

If `TRUE`, Sample IDs in `input.catalog` must have the form *sample_type*::*sample_id*.

If a character vector, then its length must be `ncol(input.catalog)`, and each value is the sample type of the corresponding column in `input.catalog`, e.g. `c(rep("Type-A",23),rep("Type-B",10))` for 23 Type-A samples and 10 Type-B samples.

If not `FALSE`, HDP will have one parent node for each sample type and one grandparent node.

`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 generate diagnostic plots to help understand the results: e.g. the stability each extracted signature and the tumors that drive the extraction of each signature.*

Description

Extract mutational signatures and optionally generate diagnostic plots to help understand the results: e.g. the stability each extracted signature and the tumors that drive the extraction of each signature.

Usage

```
RunHdpxParallel(
  input.catalog,
  seedNumber = 123,
  K.guess,
  multi.types = TRUE,
  verbose = TRUE,
  burnin = 1000,
  burnin.multiplier = 10,
  burnin.checkpoint = FALSE,
  post.n = 200,
  post.space = 100,
  post.cpiter = 3,
  post.verbosity = 0,
  CPU.cores = 20,
  num.child.process = 20,
  high.confidence.prop = 0.9,
  hc.cutoff = 0.1,
  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,
  posterior.checkpoint = FALSE
)
```

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, and the HDP structure will have one parent node for all tumors.

  If `TRUE`, Sample IDs in `input.catalog` must have the form *sample_type*::*sample_id*.

  If a character vector, then its length must be `ncol(input.catalog)`, and each value is the sample type of the corresponding column in `input.catalog`, e.g. `c(rep("Type-A",23),rep("Type-B",10))` for 23 Type-A samples and 10 Type-B samples.

  If not `FALSE`, HDP will have one parent node for each sample type and one grandparent node.

`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. Therefore we set the default of `burnin` to 1000 and `burnin.multiplier` to 10. However, number of iterations can be adjusted based on the size of dataset. The dataset with more mutations require longer burn-ins. According to our experience, 50,000 iterations are needed when analyzing all PCAWG7 genomes (2,780 samples). The burnin can be continued from a checkpoint file with `ExtendBurnin`.

`burnin.checkpoint`
: If `TRUE`, a checkpoint for burn-in 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

`high.confidence.prop`
Pass to `interpret_components`. raw clusters (mutation cluster) found in >= `high.confidence.prop` proportion of posterior samples are signatures with high confidence.

`hc.cutoff` Pass to `extract_components_from_clusters`. The cutoff of height of hierarchical clustering dendrogram, used in combining raw clusters (mutation clusters) into agreggated clusters.

`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.

`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.

`posterior.checkpoint`
If `TRUE` checkpoint the posterior sampling after every 10 posterior samples collected

Value

Invisibly, a list with the following elements:

signature The extracted signature profiles as a matrix; rows are mutation types, columns are signatures with high confidence.

signature.post.samp.number A data frame 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 numeric data frame. Each column corresponds to the sum of all mutations contributing to each signature in `signature`

exposureProbs The inferred exposures as a matrix of mutation probabilities; rows are signatures, columns are samples (e.g. tumors). This is similar to `signature.cdc` but every column was normalized to sum of 1

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

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

low.confidence.cdc A numeric data frame. Each column corresponds to the sum of all mutations contributing to each signature in `low.confidence.signature`

extracted.retval A list object returned from codeextract_components_from_clusters.

`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, and the HDP structure will have one parent node for all tumors.

  If `TRUE`, Sample IDs in `input.catalog` must have the form *sample_type*::*sample_id*.

  If a character vector, then its length must be `ncol(input.catalog)`, and each value is the sample type of the corresponding column in `input.catalog`, e.g. `c(rep("Type-A",23),rep("Type-B",10))` for 23 Type-A samples and 10 Type-B samples.

  If not `FALSE`, HDP will have one parent node for each sample type and one grandparent node.

`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,
  post.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,
  posterior.checkpoint = F
)
```

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, and the HDP structure will have one parent node for all tumors.

  If `TRUE`, Sample IDs in `input.catalog` must have the form *sample_type*::*sample_id*.

  If a character vector, then its length must be `ncol(input.catalog)`, and each value is the sample type of the corresponding column in `input.catalog`, e.g. `c(rep("Type-A",23),rep("Type-B",10))` for 23 Type-A samples and 10 Type-B samples.

  If not `FALSE`, HDP will have one parent node for each sample type and one grandparent node.

`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.

`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

`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. Therefore we set the default of `burnin` to 1000 and `burnin.multiplier` to 10. However, number of iterations can be adjusted based on the size of dataset. The dataset with more mutations require longer burn-ins. According to our experience, 50,000 iterations are needed when analyzing all PCAWG7 genomes (2,780 samples). The burnin can be continued from a checkpoint file with `ExtendBurnin`.

`burnin.checkpoint`
: If `TRUE`, a checkpoint for burn-in 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.

`posterior.checkpoint`
: If `TRUE` checkpoint the posterior sampling after every 10 posterior samples collected

Value

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

Index