

# Package ‘mSigHdp’

October 26, 2020

**Title** Mutational signature extraction using hdp (Hierarchical Dirichlet Process)

**Version** 1.0.4

**Description**

Adapts HDP to mutational signature extraction. Calls hdp for mutational signature analysis.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Language** en-US

**BuildManual** no

**biocViews**

**Roxygen** list(markdown = TRUE)

**Depends** R (>= 3.5)

**RoxygenNote** 7.1.1

**Remotes** github::steverozen/hdp@\*release,  
github::steverozen/ICAMSxtra@\*release

**Imports** hdp (>= 0.3.0),  
ICAMS (>= 2.2.3),  
reshape2

**Suggests** ICAMSxtra (>= 0.0.2),  
testthat,  
utils

## R topics documented:

AnalyzeAndPlotretval . . . . .	2
ChainBurnin . . . . .	3
ChainsDiagnosticPlot . . . . .	4
CleanChlist . . . . .	4
CombineChainsAndExtractSigs . . . . .	5
CombinePosteriorChains . . . . .	6
ComponentDiagnosticPlotting . . . . .	8
ExtendBurnin . . . . .	9
GenerateAverageCluster . . . . .	9
Generateppindex . . . . .	10
GeneratePriorppindex . . . . .	10

MultipleSetupAndPosterior . . . . .	11
NewRunHdpParallel . . . . .	13
PlotSamplesHighSigExp . . . . .	16
PrepInit . . . . .	17
PriorSetupAndActivate . . . . .	18
RunHdpParallel . . . . .	19
SetupAndActivate . . . . .	22
SetupAndPosterior . . . . .	23

<b>Index</b>	<b>26</b>
--------------	-----------

---

## 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 exists.
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 <code>hdpState-class</code> 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 = 4000,
  cpiter = 3,
  burnin.verbosity = 0,
  burnin.multiplier = 1,
  burnin.checkpoint = FALSE
)
```

## Arguments

<code>hdp.state</code>	An <code>hdpState-class</code> object or a list representation of an <code>hdpState-class</code> object.
<code>seedNumber</code>	An integer that is used to generate separate random seeds for the call to <code>dp_activate</code> , and before the call of <code>hdp_burnin</code> .
<code>burnin</code>	Pass to <code>hdp_burnin</code> burnin.
<code>cpiter</code>	Pass to <code>hdp_burnin</code> cpiter.
<code>burnin.verbosity</code>	Pass to <code>hdp_burnin</code> verbosity.
<code>burnin.multiplier</code>	A checkpoint setting. <code>burnin.multiplier</code> rounds of burnin iterations will be run. After each round, a burn-in chain will be save for checkpoint.
<code>burnin.checkpoint</code>	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	<p>output from CombinePosteriorChains. A list with the following elements:</p> <p><b>signature</b> The extracted signature profiles as a matrix; rows are mutation types, columns are samples (e.g. tumors).</p> <p><b>exposure</b> The inferred exposures as a matrix of mutation counts; rows are signatures, columns are samples (e.g. tumors).</p> <p><b>multi.chains</b> A <code>hdpSampleMulti-class</code> object. This object has the method <code>chains</code> which returns a list of <code>hdpSampleChain-class</code> objects. Each of these sample chains objects has a method <code>final_hdpState</code> (actually the methods seems to be just <code>hdp</code>) that returns the <code>hdpState</code> from which it was generated.</p>
input.catalog	ground truth catalog
out.dir	Directory that will be created for the output; if <code>overwrite</code> is <code>FALSE</code> then abort if <code>out.dir</code> already exists.
verbose	If <code>TRUE</code> 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.

---

`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 = 0.12
)
```

**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.

<code>cos.merge</code>	The cosine similarity threshold for merging raw clusters from the posterior sampling chains into "components" i.e. signatures; passed to <code>extract_components_from_clusters</code>
<code>confident.prop</code>	Passed to <code>interpret_components</code> . clusters with at least <code>confident.prop</code> of posterior samples are high confident signatures
<code>noise.prop</code>	Passed to <code>interpret_components</code> . Clusters with less than <code>noise.prop</code> of posterior samples are noise signatures
<code>hc.cutoff</code>	passed to <code>extract_components_from_clusters</code> . 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 `code-noise.signature`

**extracted.retval** A list object returned from code `interpret_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

<code>retval</code>	output from <code>CombineChainsAndExtractSigs</code>
<code>input.catalog</code>	ground truth catalog
<code>out.dir</code>	Directory that will be created for the output; if <code>overwrite</code> is <code>FALSE</code> then abort if <code>out.dir</code> already exists.
<code>verbose</code>	If <code>TRUE</code> then message progress information.



---

ExtendBurnin	<i>Extend Burn in iteration for a list representation of an <code>hdpState-class</code> object. This list is an output from <code>hdp_burnin</code> or <code>ActivateandBurnin</code>.</i>
--------------	--

---

**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**

<code>hdplist</code>	A list representation of an <code>hdpState-class</code> object
<code>seedNumber</code>	A random seed for setting the environment of <code>hdp_burnin</code> .
<code>burnin</code>	Pass to <code>hdp_posterior</code> burnin.
<code>cpiter</code>	Pass to <code>hdp_posterior</code> cpiter.
<code>verbosity</code>	Pass to <code>hdp_posterior</code> verbosity.

**Value**

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

---

GenerateAverageCluster	<i>Generate average pattern of clusters of each posterior chain from combined list of multiple posterior sample chains</i>
------------------------	--

---

**Description**

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

**Usage**

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

**Arguments**

<code>clean.chlist</code>	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	<i>Generate index for a HDP structure and num.tumor.types for other functions</i>
-----------------	---

---

### 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.

---

GeneratePriorppindex	<i>Generate index for a HDP structure and num.tumor.types for other functions for hdp_prior_init</i>
----------------------	--

---

### 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")`.

<code>input.catalog</code>	Input spectra catalog as a matrix or in <a href="#">ICAMS</a> format.
<code>nps</code>	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,
  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,
  gamma0.alpha = gamma.alpha,
  gamma0.beta = gamma.beta,
  checkpoint.chlist = TRUE,
  checkpoint.l.chain = TRUE,
  prior.sigs = NULL,
  prior.pseudoc = NULL,
  burnin.multiplier = 1,
  burnin.checkpoint = FALSE
)
```

## Arguments

<code>input.catalog</code>	Input spectra catalog as a matrix or in <a href="#">ICAMS</a> format.
<code>seedNumber</code>	A random seeds passed to <a href="#">dp_activate</a> .
<code>K.guess</code>	Suggested initial value of the number of signatures, passed to <a href="#">dp_activate</a> as <code>initcc</code> .
<code>multi.types</code>	A logical scalar or a character vector. If <code>FALSE</code> , The HDP analysis will regard all input spectra as one tumor type.

	<p>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"</p> <p>If <code>multi.types</code> is a character vector, then it should be of the same length as the number of columns in <code>input.catalog</code>, and each value is the name of the tumor type of the corresponding column in <code>input.catalog</code>. e.g. <code>c("SA.Syn.Ovary-AdenoCA", "SA.Syn.Kidney-RCC")</code>.</p>
<code>verbose</code>	If TRUE then message progress information.
<code>post.burnin</code>	Pass to <code>hdp_posterior_sample</code> burnin.
<code>post.n</code>	Pass to <code>hdp_posterior_sample</code> n.
<code>post.space</code>	Pass to <code>hdp_posterior_sample</code> space.
<code>post.cpiter</code>	Pass to <code>hdp_posterior_sample</code> cpiter.
<code>post.verbosity</code>	Pass to <code>hdp_posterior_sample</code> verbosity.
<code>CPU.cores</code>	Number of CPUs to use; there is no point in making this larger than <code>num.child.process</code> .
<code>num.child.process</code>	Number of posterior sampling chains; can set to 1 for testing.
<code>gamma.alpha</code>	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.
<code>gamma.beta</code>	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.
<code>gamma0.alpha</code>	See figure B.1 from Nicola Robert's thesis. The shape parameter ( $\alpha_0$ ) of the gamma distribution priors for the Dirichlet process concentration parameters ( $\gamma_0$ ) for $G_0$ .
<code>gamma0.beta</code>	See figure B.1 from Nicola Robert's thesis. Inverse scale parameter (rate parameter, $\beta_0$ ) of the gamma distribution priors for the Dirichlet process concentration parameters ( $\gamma_0$ ) for $G_0$ .
<code>checkpoint.chlist</code>	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.
<code>checkpoint.1.chain</code>	If TRUE checkpoint the sample chain to current working directory, in a file called <code>sample.chain.seed_number.Rdata</code> .
<code>prior.sigs</code>	A matrix containing prior signatures.
<code>prior.pseudoc</code>	A numeric list. Pseudo 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.
<code>burnin.multiplier</code>	A checkpoint setting. <code>burnin.multiplier</code> rounds of burnin iterations will be run. After each round, a burn-in chain will be save for checkpoint.
<code>burnin.checkpoint</code>	Default is False. If True, a checkpoint for burnin will be created.

**Value**

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

---

NewRunHdpParallel	<i>Extract mutational signatures and optionally compare them to existing signatures and exposures.</i>
-------------------	--

---

**Description**

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

**Usage**

```
NewRunHdpParallel(
  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,
  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 = 1,
  gamma0.alpha = gamma.alpha,
  gamma0.beta = gamma.beta,
  checkpoint.chlist = TRUE,
  checkpoint.l.chain = TRUE,
  prior.sigs = NULL,
  prior.pseudoc = NULL,
  burnin.multiplier = 1,
  burnin.checkpoint = FALSE
)
```

**Arguments**

`input.catalog`

Input spectra catalog as a matrix or in [ICAMS](#) format.

<code>seedNumber</code>	A random seeds passed to <code>dp_activate</code> .
<code>K.guess</code>	Suggested initial value of the number of signatures, passed to <code>dp_activate</code> as <code>initcc</code> .
<code>multi.types</code>	<p>A logical scalar or a character vector. If <code>FALSE</code>, The HDP analysis will regard all input spectra as one tumor type.</p> <p>If <code>TRUE</code>, 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"</p> <p>If <code>multi.types</code> is a character vector, then it should be of the same length as the number of columns in <code>input.catalog</code>, and each value is the name of the tumor type of the corresponding column in <code>input.catalog</code>. e.g. <code>c("SA.Syn.Ovary-AdenoCA", "SA.Syn.Kidney-RCC")</code>.</p>
<code>verbose</code>	If <code>TRUE</code> then message progress information.
<code>post.burnin</code>	Pass to <code>hdp_posterior_sample</code> burnin.
<code>post.n</code>	Pass to <code>hdp_posterior_sample</code> n.
<code>post.space</code>	Pass to <code>hdp_posterior_sample</code> space.
<code>post.cpiter</code>	Pass to <code>hdp_posterior_sample</code> cpiter.
<code>post.verbosity</code>	Pass to <code>hdp_posterior_sample</code> verbosity.
<code>CPU.cores</code>	Number of CPUs to use; there is no point in making this larger than <code>num.child.process</code> .
<code>num.child.process</code>	Number of posterior sampling chains; can set to 1 for testing.
<code>cos.merge</code>	The cosine similarity threshold for merging raw clusters from the posterior sampling chains into "components" i.e. signatures; passed to <code>extract_components_from_clusters</code> .
<code>confident.prop</code>	Passed to <code>interpret_components</code> . clusters with at least <code>confident.prop</code> of posterior samples are high confident signatures
<code>noise.prop</code>	Passed to <code>interpret_components</code> . Clusters with less than <code>noise.prop</code> of posterior samples are noise signatures
<code>hc.cutoff</code>	passed to <code>extract_components_from_clusters</code> . The cutoff of height of hierarchical clustering dendrogram
<code>ground.truth.sig</code>	Optional. Either a string with the path to file with ground truth signatures or and <code>ICAMS</code> catalog with the ground truth signatures. These are the signatures used to construct the ground truth spectra.
<code>ground.truth.exp</code>	Optional. Ground truth exposure matrix or path to file with ground truth exposures. If <code>NULL</code> skip checks that need this information.
<code>overwrite</code>	If <code>TRUE</code> overwrite <code>out.dir</code> if it exists, otherwise raise an error.
<code>out.dir</code>	Directory that will be created for the output; if <code>overwrite</code> is <code>FALSE</code> then abort if <code>out.dir</code> already exists.
<code>gamma.alpha</code>	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.
<code>gamma.beta</code>	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.

<code>gamma0.alpha</code>	See figure B.1 from Nicola Robert's thesis. The shape parameter ( $\alpha_0$ ) of the gamma distribution priors for the Dirichlet process concentration parameters ( $\gamma_0$ ) for $G_0$ .
<code>gamma0.beta</code>	See figure B.1 from Nicola Robert's thesis. Inverse scale parameter (rate parameter, $\beta_0$ ) of the gamma distribution priors for the Dirichlet process concentration parameters ( $\gamma_0$ ) for $G_0$ .
<code>checkpoint.chlist</code>	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.
<code>checkpoint.l.chain</code>	If TRUE checkpoint the sample chain to current working directory, in a file called <code>sample.chain.seed_number.Rdata</code> .
<code>prior.sigs</code>	A matrix containing prior signatures.
<code>prior.pseudoc</code>	A numeric list. Pseudo 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.
<code>burnin.multiplier</code>	A checkpoint setting. <code>burnin.multiplier</code> rounds of <code>burnin</code> iterations will be run. After each round, a burn-in chain will be save for checkpoint.
<code>burnin.checkpoint</code>	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 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 `signature`
- 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 `code-noise.signature`
- extracted.retval** A list object returned from `codeinterpret_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 Assgiment) 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 Assgiment) 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 <a href="#">extract_ccc_cdc_from_hdp</a>
hdpsample	A <a href="#">hdpSampleChain-class</a> or <a href="#">hdpSampleMulti-class</a> object including output from <a href="#">extract_components_from_clusters</a>
input.catalog	input catalog for samples
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	<i>Initialize hdp object Allocate process index for hdp initialization. Prepare for <a href="#">hdp_init</a></i>
----------	--

---

## 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

<code>multi.types</code>	<p>A logical scalar or a character vector. If FALSE, The HDP analysis will regard all input spectra as one tumor type.</p> <p>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"</p> <p>If <code>multi.types</code> is a character vector, then it should be of the same length as the number of columns in <code>input.catalog</code>, and each value is the name of the tumor type of the corresponding column in <code>input.catalog</code>. e.g. <code>c("SA.Syn.Ovary-AdenoCA", "SA.Syn.Kidney-RCC")</code>.</p>
<code>input.catalog</code>	Input spectra catalog as a matrix or in <a href="#">ICAMS</a> format.
<code>verbose</code>	If TRUE then message progress information.
<code>K.guess</code>	Suggested initial value of the number of signatures, passed to <a href="#">dp_activate</a> as <code>initcc</code> .
<code>gamma.alpha</code>	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.
<code>gamma.beta</code>	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.

<code>gamma0.alpha</code>	See figure B.1 from Nicola Robert's thesis. The shape parameter ( $\alpha_0$ ) of the gamma distribution priors for the Dirichlet process concentration parameters ( $\gamma_0$ ) for $G_0$ .
<code>gamma0.beta</code>	See figure B.1 from Nicola Robert's thesis. Inverse scale parameter (rate parameter, $\beta_0$ ) of the gamma distribution priors for the Dirichlet process concentration parameters ( $\gamma_0$ ) 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

<code>prior.sigs</code>	A matrix containing prior signatures.
<code>prior.pseudoc</code>	A numeric list. Pseudo 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.
<code>gamma.alpha</code>	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.
<code>gamma.beta</code>	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.
<code>K.guess</code>	Suggested initial value of the number of signatures, passed to <code>dp_activate</code> as <code>initcc</code> .
<code>gamma0.alpha</code>	See figure B.1 from Nicola Robert's thesis. The shape parameter ( $\alpha_0$ ) of the gamma distribution priors for the Dirichlet process concentration parameters ( $\gamma_0$ ) for $G_0$ .

<code>gamma0.beta</code>	See figure B.1 from Nicola Robert's thesis. Inverse scale parameter (rate parameter, $\beta_0$ ) of the gamma distribution priors for the Dirichlet process concentration parameters ( $\gamma_0$ ) for $G_0$ .
<code>multi.types</code>	A logical scalar or a character vector. If <code>FALSE</code> , The HDP analysis will regard all input spectra as one tumor type. If <code>TRUE</code> , 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 <code>multi.types</code> is a character vector, then it should be of the same length as the number of columns in <code>input.catalog</code> , and each value is the name of the tumor type of the corresponding column in <code>input.catalog</code> . e.g. <code>c("SA.Syn.Ovary-AdenoCA", "SA.Syn.Kidney-RCC")</code> .
<code>input.catalog</code>	Input spectra catalog as a matrix or in <a href="#">ICAMS</a> format.
<code>verbose</code>	If <code>TRUE</code> then message progress information.
<code>seedNumber</code>	A random seeds passed to <a href="#">dp_activate</a> .

**Value**

Invisibly, an [hdpState-class](#) object as returned from [dp\\_activate](#).

---

<code>RunHdpParallel</code>	<i>Extract mutational signatures and optionally compare them to existing signatures and exposures.</i>
-----------------------------	--

---

**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,
  ground.truth.sig = NULL,
  ground.truth.exp = NULL,
```

```

    overwrite = TRUE,
    out.dir = NULL,
    gamma.alpha = 1,
    gamma.beta = 1,
    gamma0.alpha = gamma.alpha,
    gamma0.beta = gamma.beta,
    checkpoint.chlist = TRUE,
    checkpoint.l.chain = TRUE,
    prior.sigs = NULL,
    prior.pseudoc = NULL,
    burnin.multiplier = 1,
    burnin.checkpoint = FALSE
)

```

### Arguments

<code>input.catalog</code>	Input spectra catalog as a matrix or in <a href="#">ICAMS</a> format.
<code>seedNumber</code>	A random seeds passed to <a href="#">dp_activate</a> .
<code>K.guess</code>	Suggested initial value of the number of signatures, passed to <a href="#">dp_activate</a> as <code>initcc</code> .
<code>multi.types</code>	<p>A logical scalar or a character vector. If FALSE, The HDP analysis will regard all input spectra as one tumor type.</p> <p>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"</p> <p>If <code>multi.types</code> is a character vector, then it should be of the same length as the number of columns in <code>input.catalog</code>, and each value is the name of the tumor type of the corresponding column in <code>input.catalog</code>. e.g. <code>c("SA.Syn.Ovary-AdenoCA", "SA.Syn.Kidney-RCC")</code>.</p>
<code>verbose</code>	If TRUE then message progress information.
<code>post.burnin</code>	Pass to <a href="#">hdp_posterior_sample</a> burnin.
<code>post.n</code>	Pass to <a href="#">hdp_posterior_sample</a> n.
<code>post.space</code>	Pass to <a href="#">hdp_posterior_sample</a> space.
<code>post.cpiter</code>	Pass to <a href="#">hdp_posterior_sample</a> cpiter.
<code>post.verbosity</code>	Pass to <a href="#">hdp_posterior_sample</a> verbosity.
<code>CPU.cores</code>	Number of CPUs to use; there is no point in making this larger than <code>num.child.process</code> .
<code>num.child.process</code>	Number of posterior sampling chains; can set to 1 for testing.
<code>cos.merge</code>	The cosine similarity threshold for merging raw clusters from the posterior sampling chains into "components" i.e. signatures; passed to <a href="#">hdp_extract_components</a> .
<code>min.sample</code>	A "component" (i.e. signature) must have at least this many samples; passed to <a href="#">hdp_merge_and_extract_components</a> .
<code>categ.CI</code>	A number the range [0, 1]. The level of the confidence interval used in step 4 of <a href="#">hdp_merge_and_extract_components</a> . 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.

<code>exposure.CI</code>	A number in the range $[0, 1]$ . The level of the confidence interval used in step 5 of <code>hdp_merge_and_extract_components</code> . The CI in question here for the total number of mutations assigned to an averaged raw cluster.
<code>ground.truth.sig</code>	Optional. Either a string with the path to file with ground truth signatures or and <a href="#">ICAMS</a> catalog with the ground truth signatures. These are the signatures used to construct the ground truth spectra.
<code>ground.truth.exp</code>	Optional. Ground truth exposure matrix or path to file with ground truth exposures. If <code>NULL</code> skip checks that need this information.
<code>overwrite</code>	If <code>TRUE</code> overwrite <code>out.dir</code> if it exists, otherwise raise an error.
<code>out.dir</code>	Directory that will be created for the output; if <code>overwrite</code> is <code>FALSE</code> then abort if <code>out.dir</code> already exists.
<code>gamma.alpha</code>	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.
<code>gamma.beta</code>	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.
<code>gamma0.alpha</code>	See figure B.1 from Nicola Robert's thesis. The shape parameter ( $\alpha_0$ ) of the gamma distribution priors for the Dirichlet process concentration parameters ( $\gamma_0$ ) for $G_0$ .
<code>gamma0.beta</code>	See figure B.1 from Nicola Robert's thesis. Inverse scale parameter (rate parameter, $\beta_0$ ) of the gamma distribution priors for the Dirichlet process concentration parameters ( $\gamma_0$ ) for $G_0$ .
<code>checkpoint.chlist</code>	If <code>TRUE</code> , checkpoint the (unclean) <code>chlist</code> to "initial.chlist.Rdata" in the current working directory, and checkpoint the clean <code>chlist</code> to "clean.chlist.Rdata" in the current working directory.
<code>checkpoint.l.chain</code>	If <code>TRUE</code> checkpoint the sample chain to current working directory, in a file called <code>sample.chain.seed_number.Rdata</code> .
<code>prior.sigs</code>	A matrix containing prior signatures.
<code>prior.pseudoc</code>	A numeric list. Pseudo 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.
<code>burnin.multiplier</code>	A checkpoint setting. <code>burnin.multiplier</code> rounds of <code>burnin</code> iterations will be run. After each round, a burn-in chain will be save for checkpoint.
<code>burnin.checkpoint</code>	Default is <code>False</code> . If <code>True</code> , a checkpoint for <code>burnin</code> 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`.

---

SetupAndActivate	<i>Generate an HDP Gibbs sampling chain from a spectra catalog.</i>
------------------	---

---

**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**

<code>input.catalog</code>	Input spectra catalog as a matrix or in <a href="#">ICAMS</a> format.
<code>seedNumber</code>	A random seeds passed to <a href="#">dp_activate</a> .
<code>K.guess</code>	Suggested initial value of the number of signatures, passed to <a href="#">dp_activate</a> as <code>initcc</code> .

<code>multi.types</code>	<p>A logical scalar or a character vector. If <code>FALSE</code>, The HDP analysis will regard all input spectra as one tumor type.</p> <p>If <code>TRUE</code>, 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"</p> <p>If <code>multi.types</code> is a character vector, then it should be of the same length as the number of columns in <code>input.catalog</code>, and each value is the name of the tumor type of the corresponding column in <code>input.catalog</code>. e.g. <code>c("SA.Syn.Ovary-AdenoCA", "SA.Syn.Kidney-RCC")</code>.</p>
<code>verbose</code>	If <code>TRUE</code> then message progress information.
<code>gamma.alpha</code>	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.
<code>gamma.beta</code>	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.
<code>gamma0.alpha</code>	See figure B.1 from Nicola Robert's thesis. The shape parameter ( $\alpha_0$ ) of the gamma distribution priors for the Dirichlet process concentration parameters ( $\gamma_0$ ) for $G_0$ .
<code>gamma0.beta</code>	See figure B.1 from Nicola Robert's thesis. Inverse scale parameter (rate parameter, $\beta_0$ ) of the gamma distribution priors for the Dirichlet process concentration parameters ( $\gamma_0$ ) 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,
  post.burnin = 4000,
  post.n = 50,
  post.space = 50,
  post.cpiter = 3,
  post.verbosity = 0,
  gamma.alpha = 1,
```

```

gamma.beta = 1,
gamma0.alpha = gamma.alpha,
gamma0.beta = gamma.beta,
checkpoint.1.chain = TRUE,
burnin.multiplier = 1,
burnin.checkpoint = FALSE,
prior.sigs = NULL,
prior.pseudoc = NULL
)

```

### Arguments

<code>input.catalog</code>	Input spectra catalog as a matrix or in <a href="#">ICAMS</a> format.
<code>seedNumber</code>	A random seeds passed to <a href="#">dp_activate</a> .
<code>K.guess</code>	Suggested initial value of the number of signatures, passed to <a href="#">dp_activate</a> as <code>initcc</code> .
<code>multi.types</code>	<p>A logical scalar or a character vector. If <code>FALSE</code>, The HDP analysis will regard all input spectra as one tumor type.</p> <p>If <code>TRUE</code>, 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"</p> <p>If <code>multi.types</code> is a character vector, then it should be of the same length as the number of columns in <code>input.catalog</code>, and each value is the name of the tumor type of the corresponding column in <code>input.catalog</code>.</p> <p>e.g. <code>c("SA.Syn.Ovary-AdenoCA", "SA.Syn.Kidney-RCC")</code>.</p>
<code>verbose</code>	If <code>TRUE</code> then message progress information.
<code>post.burnin</code>	Pass to <a href="#">hdp_posterior_sample</a> <code>burnin</code> .
<code>post.n</code>	Pass to <a href="#">hdp_posterior_sample</a> <code>n</code> .
<code>post.space</code>	Pass to <a href="#">hdp_posterior_sample</a> <code>space</code> .
<code>post.cpiter</code>	Pass to <a href="#">hdp_posterior_sample</a> <code>cpiter</code> .
<code>post.verbosity</code>	Pass to <a href="#">hdp_posterior_sample</a> <code>verbosity</code> .
<code>gamma.alpha</code>	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.
<code>gamma.beta</code>	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.
<code>gamma0.alpha</code>	See figure B.1 from Nicola Robert's thesis. The shape parameter ( $\alpha_0$ ) of the gamma distribution priors for the Dirichlet process concentration parameters ( $\gamma_0$ ) for $G_0$ .
<code>gamma0.beta</code>	See figure B.1 from Nicola Robert's thesis. Inverse scale parameter (rate parameter, $\beta_0$ ) of the gamma distribution priors for the Dirichlet process concentration parameters ( $\gamma_0$ ) for $G_0$ .
<code>checkpoint.1.chain</code>	If <code>TRUE</code> checkpoint the sample chain to current working directory, in a file called <code>sample.chain.seed_number.Rdata</code> .



<code>burnin.multiplier</code>	A checkpoint setting. <code>burnin.multiplier</code> rounds of burnin iterations will be run. After each round, a burn-in chain will be save for checkpoint.
<code>burnin.checkpoint</code>	Default is False. If True, a checkpoint for burnin will be created.
<code>prior.sigs</code>	A matrix containing prior signatures.
<code>prior.pseudoc</code>	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, an `hdpSampleChain-class` object as returned from `hdp_posterior`.

# Index

AnalyzeAndPlotretval, 2

ChainBurnin, 3

chains, 4, 8, 22

ChainsDiagnosticPlot, 4

CleanChlist, 4

CombineChainsAndExtractSigs, 5, 8

CombinePosteriorChains, 6

comp\_dp\_counts, 6, 15

ComponentDiagnosticPlotting, 8

dp\_activate, 3, 11, 14, 17–20, 22–24

ExtendBurnin, 9

extract\_ccc\_cdc\_from\_hdp, 16

extract\_components\_from\_clusters,  
6, 14, 16

final\_hdpState, 4, 8, 22

GenerateAverageCluster, 9

Generateppindex, 10

GeneratePriorppindex, 10

hdp\_burnin, 3, 9

hdp\_extract\_components, 7, 20

hdp\_init, 17

hdp\_merge\_and\_extract\_components,  
7, 8, 20, 22

hdp\_posterior, 9, 25

hdp\_posterior\_sample, 12, 14, 20, 24

hdpState-class, 3, 9

ICAMS, 2, 5, 7, 10, 11, 13, 14, 17, 19–22, 24

interpret\_components, 6, 14, 15

MultipleSetupAndPosterior, 11

NewRunHdpParallel, 13

PlotSamplesHighSigExp, 16

PrepInit, 17

PriorSetupAndActivate, 18

RunHdpParallel, 19

SetupAndActivate, 22

SetupAndPosterior, 23