

Package ‘infercnv’

August 3, 2025

Type Package

Title Infer Copy Number Variation from Single-Cell RNA-Seq Data

Version 1.25.0

Date 2023-12-01

BugReports <https://github.com/broadinstitute/inferCNV/issues>

Description Using single-cell RNA-Seq expression to visualize CNV in cells.

biocViews Software, CopyNumberVariation, VariantDetection, StructuralVariation, GenomicVariation, Genetics, Transcriptomics, StatisticalMethod, Bayesian, HiddenMarkovModel, SingleCell

Depends R(>= 4.0)

License BSD_3_clause + file LICENSE

LazyData TRUE

VignetteBuilder knitr

Suggests BiocStyle, knitr, rmarkdown, testthat

RoxygenNote 7.2.3

NeedsCompilation no

SystemRequirements JAGS 4.x.y

Imports graphics, grDevices, RColorBrewer, gplots, futile.logger, stats, utils, methods, ape, phyclus, Matrix, fastcluster, parallelDist, dplyr, HiddenMarkov, ggplot2, edgeR, coin, caTools, digest, RANN, igraph, reshape2, rjags, fitdistrplus, future, foreach, doParallel, Seurat, BiocGenerics, SummarizedExperiment, SingleCellExperiment, tidyr, parallel, coda, gridExtra, argparse

URL <https://github.com/broadinstitute/inferCNV/wiki>

Collate 'SplatterScrape.R' 'data.R' 'inferCNV.R' 'inferCNV_BayesNet.R' 'inferCNV_HMM.R' 'inferCNV_constants.R' 'inferCNV_heatmap.R' 'inferCNV_hidden_spike.R' 'inferCNV_i3HMM.R' 'inferCNV_mask_non_DE.R' 'inferCNV_meanVarSim.R' 'inferCNV_ops.R' 'inferCNV_simple_sim.R' 'inferCNV_tumor_subclusters.R' 'inferCNV_tumor_subclusters.random_smoothed_trees.R' 'infercnv_sampling.R' 'noise_reduction.R' 'seurat_interaction.R'

git_url <https://git.bioconductor.org/packages/infercnv>

git_branch devel

git_last_commit e916d44

git_last_commit_date 2025-04-15

Repository Bioconductor 3.22

Date/Publication 2025-08-03

Author Timothy Tickle [aut],
Itay Tirosh [aut],
Christophe Georgescu [aut, cre],
Maxwell Brown [aut],
Brian Haas [aut]

Maintainer Christophe Georgescu <cgeorges@broadinstitute.org>

Contents

infercnv-package	2
add_to_seurat	3
apply_median_filtering	4
color.palette	5
CreateInfercnvObject	5
filterHighPNormals	7
HMM_states	8
infercnv-class	8
inferCNVBayesNet	9
infercnv_annots_example	11
infercnv_data_example	11
infercnv_genes_example	11
infercnv_object_example	12
MCMC_inferCNV-class	12
mcnc_obj	13
plot_cnv	13
plot_per_group	15
plot_subclusters	17
run	18
sample_object	25
validate_infercnv_obj	26
Index	27

infercnv-package	<i>infercnv: Infer Copy Number Variation from Single-Cell RNA-Seq Data</i>
------------------	--

Description

Using single-cell RNA-Seq expression to visualize CNV in cells.

Details

The main functions you will need to use are `CreateInfercnvObject()` and `run(infercnv_object)`. For additional details on running the analysis step by step, please refer to the example vignette.

Author(s)

Maintainer: Christophe Georgescu <cgeorges@broadinstitute.org>

Authors:

- Timothy Tickle <ttickle@broadinstitute.org>
- Itay Tirosh <tirosh@broadinstitute.org>
- Maxwell Brown <mbrown@broadinstitute.org>
- Brian Haas <bhaas@broadinstitute.org>

See Also

Useful links:

- <https://github.com/broadinstitute/inferCNV/wiki>
- Report bugs at <https://github.com/broadinstitute/inferCNV/issues>

add_to_seurat

add_to_seurat()

Description

Add meta.data about CNAs to a Seurat object from an `infercnv_obj`

Usage

```
add_to_seurat(
  seurat_obj = NULL,
  assay_name = "RNA",
  infercnv_output_path,
  top_n = 10,
  bp_tolerance = 2e+06,
  column_prefix = NULL
)
```

Arguments

<code>seurat_obj</code>	Seurat object to add meta.data to (default: NULL)
<code>assay_name</code>	Name of the assay in the Seurat object if provided. (default: "RNA")
<code>infercnv_output_path</code>	Path to the output folder of the infercnv run to use
<code>top_n</code>	How many of the largest CNA (in number of genes) to get.
<code>bp_tolerance</code>	How many bp of tolerance to have around feature start/end positions for top_n largest CNVs.
<code>column_prefix</code>	String to add as a prefix to the Seurat metadata columns. Only applied to the <code>seurat_obj</code> , if supplied. Default is NULL


```
data(infercnv_object_example)

infercnv_object_example <- infercnv::apply_median_filtering(infercnv_object_example)
# plot result object
```

color.palette	<i>Helper function allowing greater control over the steps in a color palette.</i>
---------------	--

Description

Helper function allowing greater control over the steps in a color palette. Source: <http://menugget.blogspot.com/2011/11/color-steps-for-colorramppalette.html#more>

Usage

```
color.palette(steps, between = NULL, ...)
```

Arguments

steps	Vector of colors to change use in the palette
between	Steps where gradients change
...	Additional arguments of colorRampPalette

Value

Color palette

Examples

```
color.palette(c("darkblue", "white", "darkred"),
              c(2, 2))
```

CreateInfercnvObject	<i>CreateInfercnvObject</i>
----------------------	-----------------------------

Description

Creation of an infercnv object. This requires the following inputs: A more detailed description of each input is provided below:

The raw_counts_matrix:

```
MGH54_P16_F12 MGH53_P5_C12 MGH54_P12_C10 MGH54_P16_F02 MGH54_P11_C11 ...
DDX11L1 0.000000 0.000000 0.000000 0.000000 0.000000 WASH7P 0.000000 2.231939 7.186235
5.284944 0.9650009 FAM138A 0.1709991 0.000000 0.000000 0.000000 0.000000 OR4F5 0.000000
0.000000 0.000000 0.000000 0.000000 OR4F29 0.000000 0.000000 0.000000 0.000000 0.000000
...
```

The gene_order_file, contains chromosome, start, and stop position for each gene, tab-delimited:

```
chr start stop DDX11L1 chr1 11869 14412 WASH7P chr1 14363 29806 FAM138A chr1 34554
36081 OR4F5 chr1 69091 70008 OR4F29 chr1 367640 368634 OR4F16 chr1 621059 622053 ...
```

The annotations_file, containing the cell name and the cell type classification, tab-delimited.

```
V1 V2 1 MGH54_P2_C12 Microglia/Macrophage 2 MGH36_P6_F03 Microglia/Macrophage 3
MGH53_P4_H08 Microglia/Macrophage 4 MGH53_P2_E09 Microglia/Macrophage 5 MGH36_P5_E12
Oligodendrocytes (non-malignant) 6 MGH54_P2_H07 Oligodendrocytes (non-malignant) ... 179
93_P9_H03 malignant 180 93_P10_D04 malignant 181 93_P8_G09 malignant 182 93_P10_B10
malignant 183 93_P9_C07 malignant 184 93_P8_A12 malignant ...
```

and the ref_group_names vector might look like so: c("Microglia/Macrophage", "Oligodendrocytes (non-malignant)")

Usage

```
CreateInfercnvObject(
  raw_counts_matrix,
  gene_order_file,
  annotations_file,
  ref_group_names,
  delim = "\t",
  max_cells_per_group = NULL,
  min_max_counts_per_cell = c(100, +Inf),
  chr_exclude = c("chrX", "chrY", "chrM")
)
```

Arguments

raw_counts_matrix	the matrix of genes (rows) vs. cells (columns) containing the raw counts If a filename is given, it'll be read via read.table() otherwise, if matrix or Matrix, will use the data directly.
gene_order_file	data file containing the positions of each gene along each chromosome in the genome.
annotations_file	a description of the cells, indicating the cell type classifications
ref_group_names	a vector containing the classifications of the reference (normal) cells to use for inferring cnv
delim	delimiter used in the input files
max_cells_per_group	maximum number of cells to use per group. Default=NULL, using all cells defined in the annotations_file. This option is useful for randomly subsetting the existing data for a quicker preview run, such as using 50 cells per group instead of hundreds.
min_max_counts_per_cell	minimum and maximum counts allowed per cell. Any cells outside this range will be removed from the counts matrix. default=(100, +Inf) and uses all cells. If used, should be set as c(min_counts, max_counts)
chr_exclude	list of chromosomes in the reference genome annotations that should be excluded from analysis. Default = c('chrX', 'chrY', 'chrM')

Value

infercnv

Examples

[illegible]

<code>filterHighPNormals</code>	<i>filterHighPNormals</i> : Filter the HMM identified CNV's by the CNV's posterior probability of belonging to a normal state.
---------------------------------	--

Description

The following function will filter the HMM identified CNV's by the CNV's posterior probability of belonging to a normal state identified by the function `inferCNVBayesNet()`. Will filter CNV's based on a user desired threshold probability. Any CNV with a probability of being normal above the threshold will be removed.

Usage

```
filterHighPNormals(MCMC_inferCNV_obj, HMM_states, BayesMaxPNormal, useRaster)
```

Arguments

MCMC_inferCNV_obj	MCMC infernCNV object.
HMM_states	InferCNV object with HMM states in expression data.
BayesMaxPNormal	Option to filter CNV or cell lines by some probability threshold.
useRaster	Option to use rasterization when plotting

Value

Returns a list of (MCMC_inferCNV_obj, HMM_states) With removed CNV's.

Examples

[illegible]

HMM_states	<i>infercnv object result of the processing of run() in the HMM example, to be used for other examples.</i>
------------	---

Description

infercnv object result of the processing of run() in the HMM example, to be used for other examples.

Usage

HMM_states

Format

An infercnv object containing HMM predictions

infercnv-class	<i>The infercnv Class</i>
----------------	---------------------------

Description

An infercnv object encapsulates the expression data and gene chromosome ordering information that is leveraged by infercnv for data exploration. The infercnv object is passed among the infercnv data processing and plotting routines.

Details

Slots in the infercnv object include:

Slots

expr.data <matrix> the count or expression data matrix, manipulated throughout infercnv ops

count.data <matrix> retains the original count data, but shrinks along with expr.data when genes are removed.

gene_order <data.frame> chromosomal gene order

reference_grouped_cell_indices <list> mapping [['group_name']] to c(cell column indices) for reference (normal) cells

observation_grouped_cell_indices <list> mapping [['group_name']] to c(cell column indices) for observation (tumor) cells

tumor_subclusters <list> stores subclustering of tumors if requested

options <list> stores the options relevant to the analysis in itself (in contrast with options relevant to plotting or paths)

.hspike a hidden infercnv object populated with simulated spiked-in data

inferCNVBayesNet	<i>inferCNVBayesNet: Run Bayesian Network Mixture Model To Obtain Posterior Probabilities For HMM Predicted States</i>
------------------	--

Description

Uses Markov Chain Monte Carlo (MCMC) and Gibbs sampling to estimate the posterior probability of being in one of six Copy Number Variation states (states: 0, 0.5, 1, 1.5, 2, 3) for CNV's identified by inferCNV's HMM. Posterior probabilities are found for the entire CNV cluster and each individual cell line in the CNV.

Usage

```
inferCNVBayesNet(
  file_dir,
  infercnv_obj,
  HMM_states,
  out_dir,
  resume_file_token,
  model_file = NULL,
  CORES = 1,
  postMcmcMethod = NULL,
  plottingProbs = TRUE,
  quietly = TRUE,
  diagnostics = FALSE,
  HMM_type = HMM_type,
  k_obs_groups = k_obs_groups,
  cluster_by_groups = cluster_by_groups,
  reassignCNVs = TRUE,
  no_plot = no_plot,
  useRaster
)
```

Arguments

file_dir	Location of the directory of the inferCNV outputs.
infercnv_obj	InferCNV object.
HMM_states	InferCNV object with HMM states in expression data.
out_dir	(string) Path to where the output file should be saved to.
resume_file_token	(string) String token that contains some info on settings used to name files.
model_file	Path to the BUGS Model file.
CORES	Option to run parallel by specifying the number of cores to be used. (Default: 1)
postMcmcMethod	What actions to take after finishing the MCMC.
plottingProbs	Option for adding plots of Cell and CNV probabilities. (Default: TRUE)
quietly	Option to print descriptions along each step. (Default: TRUE)
diagnostics	Option to plot Diagnostic plots and tables. (Default: FALSE)


```
no_plot = TRUE)
```

```
infercnv_annots_example
```

Generated classification for 10 normal cells and 10 tumor cells.

Description

Generated classification for 10 normal cells and 10 tumor cells.

Usage

```
infercnv_annots_example
```

Format

A data frame with 20 rows (cells) and 1 columns (classification)

```
infercnv_data_example
```

Generated SmartSeq2 expression data with 10 normal cells and 10 tumor cells. This is only to demonstrate how to use methods, not actual data to be used in an analysis.

Description

Generated SmartSeq2 expression data with 10 normal cells and 10 tumor cells. This is only to demonstrate how to use methods, not actual data to be used in an analysis.

Usage

```
infercnv_data_example
```

Format

A data frame with 8252 rows (genes) and 20 columns (cells)

```
infercnv_genes_example
```

Downsampled gene coordinates file from GrCh37

Description

Downsampled gene coordinates file from GrCh37

Usage

```
infercnv_genes_example
```

Format

A data frame with 10338 rows (genes) and 3 columns (chr, start, end)

`infercnv_object_example`

infercnv object result of the processing of `run()` in the example, to be used for other examples.

Description

infercnv object result of the processing of `run()` in the example, to be used for other examples.

Usage

`infercnv_object_example`

Format

An infercnv object

MCMC_inferCNV-class *MCMC_inferCNV class*

Description

Uses Markov Chain Monte Carlo (MCMC) and Gibbs sampling to estimate the posterior probability of being in one of six Copy Number Variation states (states: 0, 0.5, 1, 1.5, 2, 3) for CNV's identified by inferCNV's HMM. Posterior probabilities are found for the entire CNV cluster and each individual cell line in the CNV.

Slots

`bugs_model` BUGS model.

`sig` fitted values for cell lines, 1/standard deviation to be used for determining the distribution of each cell line

`mu` Mean values to be used for determining the distribution of each cell line

`group_id` ID's given to the cell clusters.

`cell_gene` List containing the Cells and Genes that make up each CNV.

`cnv_probabilities` Probabilities of each CNV belonging to a particular state from 0 (least likely) to 1 (most likely).

`cell_probabilities` Probabilities of each cell being in a particular state, from 0 (least likely) to 1 (most likely).

`args` Input arguments given by the user

`cnv_regions` ID for each CNV found by the HMM

mcmc_obj	<i>infercnv object result of the processing of inferCNVBayesNet in the example, to be used for other examples.</i>
----------	--

Description

infercnv object result of the processing of inferCNVBayesNet in the example, to be used for other examples.

Usage

```
mcmc_obj
```

Format

An infercnv object containing posterior probability of CNV states

plot_cnv	<i>Plot the matrix as a heatmap, with cells as rows and genes as columns, ordered according to chromosome</i>
----------	---

Description

Formats the data and sends it for plotting.

Usage

```
plot_cnv(
  infercnv_obj,
  out_dir = ".",
  title = "inferCNV",
  obs_title = "Observations (Cells)",
  ref_title = "References (Cells)",
  cluster_by_groups = TRUE,
  cluster_references = TRUE,
  plot_chr_scale = FALSE,
  chr_lengths = NULL,
  k_obs_groups = 1,
  contig_cex = 1,
  x.center = mean(infercnv_obj@expr.data),
  x.range = "auto",
  hclust_method = "ward.D",
  custom_color_pal = NULL,
  color_safe_pal = FALSE,
  output_filename = "infercnv",
  output_format = "png",
  png_res = 300,
  dynamic_resize = 0,
  ref_contig = NULL,
```

```

write_expr_matrix = FALSE,
write_phylo = FALSE,
useRaster = TRUE
)

```

Arguments

<code>infercnv_obj</code>	infercnv object
<code>out_dir</code>	Directory in which to save pdf and other output.
<code>title</code>	Plot title.
<code>obs_title</code>	Title for the observations matrix.
<code>ref_title</code>	Title for the reference matrix.
<code>cluster_by_groups</code>	Whether to cluster observations by their annotations or not. Using this ignores <code>k_obs_groups</code> .
<code>cluster_references</code>	Whether to cluster references within their annotations or not. (dendrogram not displayed)
<code>plot_chr_scale</code>	Whether to scale the chromosome width on the heatmap based on their actual size rather than just the number of expressed genes.
<code>chr_lengths</code>	A named list of chromosomes lengths to use when <code>plot_chr_scale=TRUE</code> , or else chromosome size is assumed to be the last chromosome's stop position + 10k bp
<code>k_obs_groups</code>	Number of groups to break observation into.
<code>contig_cex</code>	Contig text size.
<code>x.center</code>	Value on which to center expression.
<code>x.range</code>	vector containing the extreme values in the heatmap (ie. <code>c(-3,4)</code>)
<code>hclust_method</code>	Clustering method to use for hclust.
<code>custom_color_pal</code>	Specify a custom set of colors for the heatmap. Has to be in the shape <code>color.palette(c("darkblue", "white", "darkred"), c(2, 2))</code>
<code>color_safe_pal</code>	Logical indication of using a color blindness safe palette.
<code>output_filename</code>	Filename to save the figure to.
<code>output_format</code>	format for heatmap image file (default: 'png'), options('png', 'pdf', NA) If set to NA, will print graphics natively
<code>png_res</code>	Resolution for png output.
<code>dynamic_resize</code>	Factor (≥ 0) by which to scale the dynamic resize of the observation heatmap and the overall plot based on how many cells there are. Default is 0, which disables the scaling. Try 1 first if you want to enable.
<code>ref_contig</code>	If given, will focus cluster on only genes in this contig.
<code>write_expr_matrix</code>	Includes writing a matrix file containing the expression data that is plotted in the heatmap.
<code>write_phylo</code>	Write newick strings of the dendrograms displayed on the left side of the heatmap to file.
<code>useRaster</code>	Whether to use rasterization for drawing heatmap. Only disable if it produces an error as it is much faster than not using it.

Value

A list of all relevant settings used for the plotting to be able to reuse them in another plot call while keeping consistent plotting settings, most importantly `x.range`.

Examples

```
# data(infercnv_data_example)
# data(infercnv_annots_example)
# data(infercnv_genes_example)

# infercnv_object_example <- infercnv::CreateInfercnvObject(raw_counts_matrix=infercnv_data_example,
#                                                         gene_order_file=infercnv_genes_example,
#                                                         annotations_file=infercnv_annots_example,
#                                                         ref_group_names=c("normal"))

# infercnv_object_example <- infercnv::run(infercnv_object_example,
#                                          cutoff=1,
#                                          out_dir=tempfile(),
#                                          cluster_by_groups=TRUE,
#                                          denoise=TRUE,
#                                          HMM=FALSE,
#                                          num_threads=2,
#                                          no_plot=TRUE)

data(infercnv_object_example)

plot_cnv(infercnv_object_example,
         out_dir=tempfile(),
         obs_title="Observations (Cells)",
         ref_title="References (Cells)",
         cluster_by_groups=TRUE,
         x.center=1,
         x.range="auto",
         hclust_method='ward.D',
         color_safe_pal=FALSE,
         output_filename="infercnv",
         output_format="png",
         png_res=300,
         dynamic_resize=0
        )
```

plot_per_group

plot_per_group

Description

Takes an `infercnv` object and subdivides it into one object per group of cells to allow plotting of each group on a separate plot. If references are selected, they will appear on the observation heatmap area as it is larger.

Usage

```
plot_per_group(
  infercnv_obj,
  on_references = TRUE,
  on_observations = TRUE,
  sample = FALSE,
  n_cells = 1000,
  every_n = NULL,
  above_m = 1000,
  k_obs_groups = 1,
  base_filename = "infercnv_per_group",
  output_format = "png",
  write_expr_matrix = TRUE,
  save_objects = FALSE,
  png_res = 300,
  dynamic_resize = 0,
  useRaster = TRUE,
  out_dir
)
```

Arguments

<code>infercnv_obj</code>	<code>infercnv_object</code>
<code>on_references</code>	boolean (default=TRUE), plot references (normal cells).
<code>on_observations</code>	boolean (default=TRUE), plot observations data (tumor cells).
<code>sample</code>	Whether unique groups of cells should be sampled from or not. (see other parameters for how sampling is done) (Default: FALSE)
<code>n_cells</code>	Number of cells that should be sampled per group if sampling is enabled (default = 1000) .
<code>every_n</code>	Sample 1 cell every_n cells for each group that has above_m cells, if sampling is enabled. If subclusters are defined, this will make sure that at least one cell per subcluster is sampled. Requires above_m to be set to work, overriding n_cells parameter. (Default: NULL)
<code>above_m</code>	Sample only groups that have at least above_m cells if sampling is enabled. (default: 1000) Does not require every_n to be set.
<code>k_obs_groups</code>	Number of groups to break each group in with cutree (in the color bars on the left side of the plot only). (Default: 1)
<code>base_filename</code>	Base prefix for the output files names. Will be followed by OBS/REF to indicate the type of the group, and the group name. (Default: "infercnv_per_group")
<code>output_format</code>	Output format for the figure. Choose between "png", "pdf" and NA. NA means to only write the text outputs without generating the figure itself. (default: "png")
<code>write_expr_matrix</code>	Includes writing a matrix file containing the expression data that is plotted in the heatmap. (default: FALSE)
<code>save_objects</code>	Whether to save the infercnv objects generated for each group as RDS. (default: FALSE)
<code>png_res</code>	Resolution for png output. (Default: 300)

dynamic_resize Factor (≥ 0) by which to scale the dynamic resize of the observation heatmap and the overall plot based on how many cells there are. Default is 0, which disables the scaling. Try 1 first if you want to enable. (Default: 0)

useRaster Whether to use rasterization for drawing heatmap. Only disable if it produces an error as it is much faster than not using it.

out_dir Directory in which to save plots and other outputs.

Value

void

Examples

```
# data(infercnv_data_example)
# data(infercnv_annots_example)
# data(infercnv_genes_example)

# infercnv_object_example <- infercnv::CreateInfercnvObject(raw_counts_matrix=infercnv_data_example,
#                                                         gene_order_file=infercnv_genes_example,
#                                                         annotations_file=infercnv_annots_example,
#                                                         ref_group_names=c("normal"))

# infercnv_object_example <- infercnv::run(infercnv_object_example,
#                                         cutoff=1,
#                                         out_dir=tempfile(),
#                                         cluster_by_groups=TRUE,
#                                         denoise=TRUE,
#                                         HMM=FALSE,
#                                         num_threads=2,
#                                         no_plot=TRUE)

data(infercnv_object_example)

infercnv::plot_per_group(infercnv_object_example, out_dir=tempfile())
```

plot_subclusters	<i>Plot a heatmap of the data in the infercnv object with the subclusters being displayed as annotations.</i>
------------------	---

Description

Formats the data and sends it for plotting.

Usage

```
plot_subclusters(
  infercnv_obj,
  out_dir,
  output_filename = "subcluster_as_annotations"
)
```

Arguments

infercnv_obj infercnv object
 out_dir Directory in which to output.
 output_filename Filename to save the figure to.

Value

infercnv_obj the modified infercnv object that was plotted where subclusters are assigned as annotation groups

Examples

```
# data(infercnv_data_example)
# data(infercnv_annots_example)
# data(infercnv_genes_example)

# infercnv_object_example <- infercnv::CreateInfercnvObject(raw_counts_matrix=infercnv_data_example,
#                                                             gene_order_file=infercnv_genes_example,
#                                                             annotations_file=infercnv_annots_example,
#                                                             ref_group_names=c("normal"))

# infercnv_object_example <- infercnv::run(infercnv_object_example,
#                                           cutoff=1,
#                                           out_dir=tempfile(),
#                                           cluster_by_groups=TRUE,
#                                           denoise=TRUE,
#                                           HMM=FALSE,
#                                           num_threads=2,
#                                           no_plot=TRUE)

data(infercnv_object_example)

plot_subclusters(infercnv_object_example,
                  out_dir=tempfile(),
                  output_filename="subclusters_as_annots"
                  )
```

run	<i>run() : Invokes a routine inferCNV analysis to Infer CNV changes given a matrix of RNASeq counts.</i>
-----	--

Description

Function doing the actual analysis before calling the plotting functions.

Usage

```
run(
  infercnv_obj,
  cutoff = 1,
```

```

min_cells_per_gene = 3,
out_dir = NULL,
window_length = 101,
smooth_method = c("pyramidal", "runmeans", "coordinates"),
num_ref_groups = NULL,
ref_subtract_use_mean_bounds = TRUE,
cluster_by_groups = TRUE,
cluster_references = TRUE,
k_obs_groups = 1,
hclust_method = "ward.D2",
max_centered_threshold = 3,
scale_data = FALSE,
HMM = FALSE,
HMM_transition_prob = 1e-06,
HMM_report_by = c("subcluster", "consensus", "cell"),
HMM_type = c("i6", "i3"),
HMM_i3_pval = 0.05,
HMM_i3_use_KS = FALSE,
BayesMaxPNormal = 0.5,
sim_method = "meanvar",
sim_foreground = FALSE,
reassignCNVs = TRUE,
analysis_mode = c("subclusters", "samples", "cells"),
tumor_subcluster_partition_method = c("leiden", "random_trees", "qnorm", "pheight",
  "qgamma", "shc"),
tumor_subcluster_pval = 0.1,
k_nn = 20,
leiden_method = c("PCA", "simple"),
leiden_function = c("CPM", "modularity"),
leiden_resolution = "auto",
leiden_method_per_chr = c("simple", "PCA"),
leiden_function_per_chr = c("modularity", "CPM"),
leiden_resolution_per_chr = 1,
per_chr_hmm_subclusters = FALSE,
per_chr_hmm_subclusters_references = FALSE,
z_score_filter = 0.8,
denoise = FALSE,
noise_filter = NA,
sd_amplifier = 1.5,
noise_logistic = FALSE,
outlier_method_bound = "average_bound",
outlier_lower_bound = NA,
outlier_upper_bound = NA,
final_scale_limits = NULL,
final_center_val = NULL,
debug = FALSE,
num_threads = 4,
plot_steps = FALSE,
inspect_subclusters = TRUE,
resume_mode = TRUE,
png_res = 300,
plot_probabilities = TRUE,

```

```

save_rds = TRUE,
save_final_rds = TRUE,
diagnostics = FALSE,
remove_genes_at_chr_ends = FALSE,
prune_outliers = FALSE,
mask_nonDE_genes = FALSE,
mask_nonDE_pval = 0.05,
test.use = "wilcoxon",
require_DE_all_normals = "any",
hspike_aggregate_normals = FALSE,
no_plot = FALSE,
no_prelim_plot = FALSE,
write_expr_matrix = FALSE,
write_phylo = FALSE,
output_format = "png",
plot_chr_scale = FALSE,
chr_lengths = NULL,
useRaster = TRUE,
up_to_step = 100
)

```

Arguments

<code>infercnv_obj</code>	An infercnv object populated with raw count data
<code>cutoff</code>	Cut-off for the min average read counts per gene among reference cells. (default: 1)
<code>min_cells_per_gene</code>	minimum number of reference cells requiring expression measurements to include the corresponding gene. default: 3
<code>out_dir</code>	path to directory to deposit outputs (default: NULL, required to provide non NULL) ## Smoothing params
<code>window_length</code>	Length of the window for the moving average (smoothing). Should be an odd integer. (default: 101)#'
<code>smooth_method</code>	Method to use for smoothing: c(runmeans,pyramidal,coordinates) default: pyramidal #####
<code>num_ref_groups</code>	The number of reference groups or a list of indices for each group of reference indices in relation to reference_obs. (default: NULL)
<code>ref_subtract_use_mean_bounds</code>	Determine means separately for each ref group, then remove intensities within bounds of means (default: TRUE) Otherwise, uses mean of the means across groups. #####
<code>cluster_by_groups</code>	If observations are defined according to groups (ie. patients), each group of cells will be clustered separately. (default=FALSE, instead will use k_obs_groups setting)
<code>cluster_references</code>	Whether to cluster references within their annotations or not. (dendrogram not displayed) (default: TRUE)

k_obs_groups Number of groups in which to break the observations. (default: 1)

hclust_method Method used for hierarchical clustering of cells. Valid choices are: "ward.D", "ward.D2", "single", "complete", "average", "mcquitty", "median", "centroid". default("ward.D2")

max_centered_threshold
 The maximum value a value can have after centering. Also sets a lower bound of -1 * this value. (default: 3), can set to a numeric value or "auto" to bound by the mean bounds across cells. Set to NA to turn off.

scale_data perform Z-scaling of logtransformed data (default: FALSE). This may be turned on if you have very different kinds of data for your normal and tumor samples. For example, you need to use GTEx representative normal expression profiles rather than being able to leverage normal single cell data that goes with your experiment.
 #####
 ## Downstream Analyses (HMM or non-DE-masking) based on tumor subclusters

HMM when set to True, runs HMM to predict CNV level (default: FALSE)

HMM_transition_prob
 transition probability in HMM (default: 1e-6)

HMM_report_by cell, consensus, subcluster (default: subcluster) Note, reporting is performed entirely separately from the HMM prediction. So, you can predict on subclusters, but get per-cell level reporting (more voluminous output).

HMM_type HMM model type. Options: (i6 or i3): i6: infercnv 6-state model (0, 0.5, 1, 1.5, 2, >2) where state emissions are calibrated based on simulated CNV levels. i3: infercnv 3-state model (del, neutral, amp) configured based on normal cells and HMM_i3_pval

HMM_i3_pval p-value for HMM i3 state overlap (default: 0.05)

HMM_i3_use_KS boolean: use the KS test statistic to estimate mean of amp/del distributions (ala HoneyBadger). (default=TRUE)
 ## Filtering low-conf HMM preds via BayesNet P(Normal)

BayesMaxPNormal
 maximum P(Normal) allowed for a CNV prediction according to BayesNet. (default=0.5, note zero turns it off)

sim_method method for calibrating CNV levels in the i6 HMM (default: 'meanvar')

sim_foreground don't use... for debugging, developer option.

reassignCNVs (boolean) Given the CNV associated probability of belonging to each possible state, reassign the state assignments made by the HMM to the state that has the highest probability. (default: TRUE)
 ##### ## Tumor subclustering

analysis_mode options(samples|subclusters|cells), Grouping level for image filtering or HMM predictions. default: samples (fastest, but subclusters is ideal)

tumor_subcluster_partition_method
 method for defining tumor subclusters. Options('leiden', 'random_trees', 'qnorm')
 leiden: Runs a nearest neighbor search, where communities are then partitioned with the Leiden algorithm. random_trees: Slow, uses permutation statistics w/ tree construction. qnorm: defines tree height based on the quantile defined by the tumor_subcluster_pval

```

tumor_subcluster_pval
    max p-value for defining a significant tumor subcluster (default: 0.1)
k_nn
    number k of nearest neighbors to search for when using the Leiden partition
    method for subclustering (default: 20)
leiden_method
    Method used to generate the graph on which the Leiden algorithm is applied,
    one of "PCA" or "simple". (default: "PCA")
leiden_function
    Whether to use the Constant Potts Model (CPM) or modularity in igraph. Must
    be either "CPM" or "modularity". (default: "CPM")
leiden_resolution
    resolution parameter for the Leiden algorithm using the CPM quality score (de-
    fault: auto)
leiden_method_per_chr
    Method used to generate the graph on which the Leiden algorithm is applied
    for the per chromosome subclustering, one of "PCA" or "simple". (default:
    "simple")
leiden_function_per_chr
    Whether to use the Constant Potts Model (CPM) or modularity in igraph for the
    per chromosome subclustering. Must be either "CPM" or "modularity". (de-
    fault: "modularity")
leiden_resolution_per_chr
    resolution parameter for the Leiden algorithm for the per chromosome subclus-
    tering (default: 1)
per_chr_hmm_subclusters
    Run subclustering per chromosome over all cells combined to run the HMM
    on those subclusters instead. Only applicable when using Leiden subclustering.
    This should provide enough definition in the predictions while avoiding sub-
    clusters that are too small thus providing less evidence to work with. (default:
    FALSE)
per_chr_hmm_subclusters_references
    Whether the per chromosome subclustering should also be done on references,
    which should not have as much variation as observations. (default = FALSE)
z_score_filter
    Z-score used as a treshold to filter genes used for subclustering. Applied based
    on reference genes to automatically ignore genes with high expression variabil-
    ity such as MHC genes. (default: 0.8)
##### ## de-noising parameters #####
denoise
    If True, turns on denoising according to options below
noise_filter
    Values +/- from the reference cell mean will be set to zero (whitening effect)
    default(NA, instead will use sd_amplifier below.
sd_amplifier
    Noise is defined as mean(reference_cells) +/- sdev(reference_cells) * sd_amplifier
    default: 1.5
noise_logistic
    use the noise_filter or sd_amplifier based threshold (whichever is invoked) as the
    midpoint in a logistic model for downscaling values close to the mean. (default:
    FALSE)
##### ## Outlier pruning
outlier_method_bound
    Method to use for bounding outlier values. (default: "average_bound") Will
    preferentially use outlier_lower_bound and outlier_upper_bound if set.

```

```

outlier_lower_bound
    Outliers below this lower bound will be set to this value.
outlier_upper_bound
    Outliers above this upper bound will be set to this value.
##### ## Misc options
final_scale_limits
    The scale limits for the final heatmap output by the run() method. Default "auto".
    Alt, c(low,high)
final_center_val
    Center value for final heatmap output by the run() method.
debug
    If true, output debug level logging.
num_threads
    (int) number of threads for parallel steps (default: 4)
plot_steps
    If true, saves infercnv objects and plots data at the intermediate steps.
inspect_subclusters
    If true, plot subclusters as annotations after the subclustering step to easily see
    if the subclustering options are good. (default = TRUE)
resume_mode
    leverage pre-computed and stored infercnv objects where possible. (default=TRUE)
png_res
    Resolution for png output.
plot_probabilities
    option to plot posterior probabilities (default: TRUE)
save_rds
    Whether to save the current step object results as an .rds file (default: TRUE)
save_final_rds
    Whether to save the final object results as an .rds file (default: TRUE)
diagnostics
    option to create diagnostic plots after running the Bayesian model (default:
    FALSE)
##### ## Experimental options
remove_genes_at_chr_ends
    experimental option: If true, removes the window_length/2 genes at both ends
    of the chromosome.
prune_outliers
    Define outliers loosely as those that exceed the mean boundaries among all cells.
    These are set to the bounds.
    ## experimental opts involving DE analysis
mask_nonDE_genes
    If true, sets genes not significantly differentially expressed between tumor/normal
    to the mean value for the complete data set (default: 0.05)
mask_nonDE_pval
    p-value threshold for defining statistically significant DE genes between tu-
    mor/normal
test.use
    statistical test to use. (default: "wilcoxon") alternatives include 'perm' or 't.'
require_DE_all_normals
    If mask_nonDE_genes is set, those genes will be masked only if they are are
    found as DE according to test.use and mask_nonDE_pval in each of the com-
    parisons to normal cells options: "any", "most", "all" (default: "any")
    other experimental opts
hspike_aggregate_normals
    instead of trying to model the different normal groupings individually, just merge
    them in the hspike.

```



```
#                               denoise=TRUE,  
#                               HMM=FALSE,  
#                               num_threads=2,  
#                               no_plot=TRUE)  
  
data(infercnv_object_example)  
  
infercnv_object_example <- infercnv::sample_object(infercnv_object_example, n_cells=5)  
# plot result object
```

```
validate_infercnv_obj  validate_infercnv_obj()
```

Description

validate an infercnv_obj ensures that order of genes in the @gene_order slot match up perfectly with the gene rows in the @expr.data matrix. Otherwise, throws an error and stops execution.

Usage

```
validate_infercnv_obj(infercnv_obj)
```

Arguments

```
infercnv_obj    infercnv_object
```

Value

none

Index

* classes

MCMC_inferCNV-class, [12](#)

* datasets

HMM_states, [8](#)

infercnv_annots_example, [11](#)

infercnv_data_example, [11](#)

infercnv_genes_example, [11](#)

infercnv_object_example, [12](#)

mcmc_obj, [13](#)

_PACKAGE (infercnv-package), [2](#)

add_to_seurat, [3](#)

apply_median_filtering, [4](#)

color.palette, [5](#)

CreateInfercnvObject, [5](#)

filterHighPNormals, [7](#)

HMM_states, [8](#)

infercnv (infercnv-class), [8](#)

infercnv-class, [8](#)

infercnv-package, [2](#)

infercnv_annots_example, [11](#)

infercnv_data_example, [11](#)

infercnv_genes_example, [11](#)

infercnv_object_example, [12](#)

inferCNVBayesNet, [9](#)

MCMC_inferCNV (MCMC_inferCNV-class), [12](#)

MCMC_inferCNV-class, [12](#)

mcmc_obj, [13](#)

plot_cnv, [13](#)

plot_per_group, [15](#)

plot_subclusters, [17](#)

run, [18](#)

sample_object, [25](#)

validate_infercnv_obj, [26](#)