

Package ‘infercnv’

October 16, 2019

Type Package

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

Version 1.0.4

Date 2019-09-16

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(>= 3.6)

License BSD_3_clause + file LICENSE

LazyData TRUE

VignetteBuilder knitr

Suggests BiocStyle, knitr, rmarkdown, testthat

RoxygenNote 6.1.1

NeedsCompilation no

SystemRequirements JAGS 4.x.y

Imports graphics, grDevices, RColorBrewer, gplots, futile.logger, stats, utils, methods, ape, Matrix, fastcluster, dplyr, HiddenMarkov, ggplot2, edgeR, coin, caTools, reshape, rjags, fitdistrplus, future, foreach, doParallel, 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' 'noise_reduction.R'

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

git_branch RELEASE_3_9

git_last_commit 5aaa67f

git_last_commit_date 2019-09-16

Date/Publication 2019-10-15

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

Maintainer Christophe Georgescu <cgeorges@broadinstitute.org>

R topics documented:

infercnv-package	2
annots	3
apply_median_filtering	3
CreateInfercnvObject	4
data	6
filterHighPNormals	6
genes	7
HMM_states	7
infercnv-class	8
inferCNVBayesNet	8
infercnv_obj	10
MCMC_inferCNV-class	10
mcmc_obj	11
plot_cnv	11
run	13
validate_infercnv_obj	17
Index	18

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>

annots

Generated classification for 10 normal cells and 10 tumor cells.

Description

Generated classification for 10 normal cells and 10 tumor cells.

Usage

annots

Format

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

apply_median_filtering

apply_median_filtering

Description

Apply a median filtering to the expression matrix within each tumor bounds

Usage

```
apply_median_filtering(infercnv_obj, window_size = 7,
  on_observations = TRUE, on_references = TRUE)
```

Arguments

infercnv_obj infercnv_object
 window_size Size of the window side centered on the data point to filter (default = 7).
 on_observations
 boolean (default=TRUE), run on observations data (tumor cells).
 on_references boolean (default=TRUE), run on references (normal cells).

Value

infercnv_obj with median filtering applied to observations

Examples

```
# data(data)
# data(annots)
# data(genes)

# infercnv_obj <- infercnv::CreateInfercnvObject(raw_counts_matrix=data,
#                                               gene_order_file=genes,
#                                               annotations_file=annots,
#                                               ref_group_names=c("normal"))

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

data(infercnv_object)

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

CreateInfercnvObject *CreateInfercnvObject*

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 = NULL, 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=NULL 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

```

data(data)
data(annots)
data(genes)

infercnv_obj <- infercnv::CreateInfercnvObject(raw_counts_matrix=data,
                                              gene_order_file=genes,
                                              annotations_file=annots,
                                              ref_group_names=c("normal"))

```

data	<i>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.</i>
------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

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

```
data
```

Format

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

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

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)
```

Arguments

MCMC_inferCNV_obj	MCMC inferCNV object.
HMM_states	InferCNV object with HMM states in expression data.
BayesMaxPNormal	Option to filter CNV or cell lines by some probability threshold.

Value

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

Examples

```
data(mcmc_obj)

mcmc_obj_hmm_states_list <- infercnv::filterHighPNormals( MCMC_inferCNV_obj = mcmc_obj,
                                                         HMM_states       = HMM_states,
                                                         BayesMaxPNormal  = 0.5)
```

genes	<i>Downsampled gene coordinates file from GrCh37</i>
-------	------------------------------------------------------

Description

Downsampled gene coordinates file from GrCh37

Usage

```
genes
```

Format

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

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
 .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,
  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)
```



```

CORES           = 2,
plottingProbs   = FALSE,
diagnostics     = FALSE,
HMM_type        = 'i6',
k_obs_groups    = 1,
cluster_by_groups = FALSE)

```

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

Description

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

Usage

```
infercnv_obj
```

Format

An infercnv object

MCMC_inferCNV-class	<i>MCMC_inferCNV class</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.

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

States States that are identified and (depending on posterior MCMC input methods) modified.

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,
  k_obs_groups = 3, contig_cex = 1,
  x.center = mean(infercnv_obj@expr.data), x.range = "auto",
  hclust_method = "ward.D", color_safe_pal = FALSE,
  output_filename = "infercnv", output_format = "png", png_res = 300,
  dynamic_resize = 0, ref_contig = NULL, write_expr_matrix = FALSE)
```

Arguments

infercnv_obj	infercnv object
out_dir	Directory in which to save pdf and other output.
title	Plot title.
obs_title	Title for the observations matrix.
ref_title	Title for the reference matrix.
cluster_by_groups	Whether to cluster observations by their annotations or not. Using this ignores k_obs_groups.
cluster_references	Whether to cluster references within their annotations or not. (dendrogram not displayed)

k_obs_groups	Number of groups to break observation into.
contig_cex	Contig text size.
x.center	Value on which to center expression.
x.range	vector containing the extreme values in the heatmap (ie. c(-3,4))
hclust_method	Clustering method to use for hclust.
color_safe_pal	Logical indication of using a color blindness safe palette.
output_filename	Filename to save the figure to.
output_format	format for heatmap image file (default: 'png'), options('png', 'pdf', NA) If set to NA, will print graphics natively
png_res	Resolution for png output.
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.
ref_contig	If given, will focus cluster on only genes in this contig.
write_expr_matrix	Includes writing a matrix file containing the expression data that is plotted in the heatmap.

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(data)
# data(annots)
# data(genes)

# infercnv_obj <- infercnv::CreateInfercnvObject(raw_counts_matrix=data,
#                                               gene_order_file=genes,
#                                               annotations_file=annots,
#                                               ref_group_names=c("normal"))

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

data(infercnv_object)

plot_cnv(infercnv_obj,
         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
)

```

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

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"),
    num_ref_groups = NULL, ref_subtract_use_mean_bounds = TRUE,
    cluster_by_groups = FALSE, 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 = TRUE, BayesMaxPNormal = 0.5,
    sim_method = "meanvar", sim_foreground = FALSE,
    analysis_mode = c("samples", "subclusters", "cells"),
    tumor_subcluster_partition_method = c("random_trees", "qnorm",
    "pheight", "qgamma", "shc"), tumor_subcluster_pval = 0.1,
    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, resume_mode = TRUE,
    png_res = 300, plot_probabilities = 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)

```

Arguments

infercnv_obj	An infercnv object populated with raw count data
cutoff	Cut-off for the min average read counts per gene among reference cells. (default: 1)

```

min_cells_per_gene      minimum number of reference cells requiring expression measurements to include the corresponding gene. default: 3
out_dir                 path to directory to deposit outputs (default: NULL, required to provide non NULL)
                        ## Smoothing params
window_length          Length of the window for the moving average (smoothing). Should be an odd integer. (default: 101)#'
smooth_method          Method to use for smoothing: c(runmeans,pyramidal) default: pyramidal
                        #####
num_ref_groups          The number of reference groups or a list of indices for each group of reference indices in relation to reference_obs. (default: NULL)
ref_subtract_use_mean_bounds
                        Determine means separately for each ref group, then remove intensities within bounds of means (default: TRUE) Otherwise, uses mean of the means across groups.
                        #####
cluster_by_groups      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)
cluster_references     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. (de-
                  fault=0.5, note zero turns it off)
                  ##### ## Tumor subclustering
sim_method       method for calibrating CNV levels in the i6 HMM (default: 'meanvar')
sim_foreground   don't use... for debugging, developer option.
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('random_trees', 'qnorm') ran-
                  dom_trees: (default) slow but best. Uses permutation statistics w/ tree con-
                  struction. qnorm: defines tree height based on the quantile defined by the tu-
                  mor_subcluster_pval
tumor_subcluster_pval
                  max p-value for defining a significant tumor subcluster (default: 0.1)
                  ##### ## 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.
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)
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.
no_plot     don't make any of the images. Instead, generate all non-image outputs as part
              of the run. (default: FALSE)
no_prelim_plot
              don't make the preliminary infercnv image (default: FALSE)

```

Value

infercnv_obj containing filtered and transformed data

Examples

```

data(data)
data(annots)
data(genes)

infercnv_obj <- infercnv::CreateInfercnvObject(raw_counts_matrix=data,
                                              gene_order_file=genes,
                                              annotations_file=annots,
                                              ref_group_names=c("normal"))

infercnv_obj <- infercnv::run(infercnv_obj,
                             cutoff=1,
                             out_dir=tempfile(),
                             cluster_by_groups=TRUE,
                             denoise=TRUE,

```



```
HMM=FALSE,  
num_threads=2,  
no_plot=TRUE)
```

```
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

*Topic **classes**

MCMC_inferCNV-class, 10

*Topic **datasets**

annots, 3

data, 6

genes, 7

HMM_states, 7

infercnv_obj, 10

mcmc_obj, 11

_PACKAGE (infercnv-package), 2

annots, 3

apply_median_filtering, 3

CreateInfercnvObject, 4

data, 6

filterHighPNormals, 6

genes, 7

HMM_states, 7

infercnv (infercnv-class), 8

infercnv-class, 8

infercnv-package, 2

infercnv_obj, 10

inferCNVBayesNet, 8

MCMC_inferCNV (MCMC_inferCNV-class), 10

MCMC_inferCNV-class, 10

mcmc_obj, 11

plot_cnv, 11

run, 13

validate_infercnv_obj, 17