Package ‘infercnv’

May 29, 2024

Type Package

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

Version 1.20.0

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, phyclust, 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 RELEASE_3_19

git_last_commit 441a7d9

git_last_commit_date 2024-04-30

Repository Bioconductor 3.19

Date/Publication 2024-05-29

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 ............................................................ 3
add_to_seurat ............................................................... 3
apply_median_filtering .................................................... 4
color.palette ............................................................... 5
CreateInfercnvObject ..................................................... 6
filterHighPNormals .......................................................... 8
HMM_states ................................................................. 8
infercnv-class .............................................................. 9
inferCNVBayesNet .......................................................... 10
infercnv_annots_example .................................................. 12
infercnv_data_example .................................................... 12
infercnv_genes_example ................................................... 13
infercnv_object_example .................................................. 13
MCMC_inferCNV-class ....................................................... 14
mcmc_obj ................................................................. 14
plot_cnv ................................................................. 14
plot_per_group ............................................................ 17
plot_subclusters .......................................................... 19
run ......................................................................... 21
sample_object ............................................................ 27
validate_infercnv_obj ..................................................... 29

Index 30
infercnv-package

infercnv-package

infercnv: Infer Copy Number Variation from Single-Cell RNA-Seq Data

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](https://github.com/broadinstitute/inferCNV/wiki)
- Report bugs at [https://github.com/broadinstitute/inferCNV/issues](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
)
```
apply_median_filtering

**Arguments**

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

**Value**

seurat_obj

---

**Description**

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

**Usage**

```r
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

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

---

**color.palette**

*Helper function allowing greater control over the steps in a color palette.*

**Description**

Helper function allowing greater control over the steps in a color palette. Source: [http://menugget.blogspot.com/2011/11/define-color-steps-for-colorramp.html#more](http://menugget.blogspot.com/2011/11/define-color-steps-for-colorramp.html#more)

**Usage**

```r
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

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:

<table>
<thead>
<tr>
<th>Gene</th>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
<th>Value 4</th>
<th>Value 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDX11L1</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>ASH7P</td>
<td>2.231939</td>
<td>7.186235</td>
<td>5.284944</td>
<td>0.965009</td>
<td>7.186235</td>
</tr>
<tr>
<td>FAM138A</td>
<td>0.170999</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>OR4F5</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>OR4F29</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Chr</th>
<th>Start</th>
<th>Stop</th>
<th>Gene</th>
</tr>
</thead>
<tbody>
<tr>
<td>chr1</td>
<td>11869</td>
<td>14412</td>
<td>DDX11L1</td>
</tr>
<tr>
<td>chr1</td>
<td>14363</td>
<td>29806</td>
<td>WASH7P</td>
</tr>
<tr>
<td>chr1</td>
<td>34554</td>
<td>36081</td>
<td>OR4F5</td>
</tr>
<tr>
<td>chr1</td>
<td>69091</td>
<td>70008</td>
<td>OR4F29</td>
</tr>
<tr>
<td>chr1</td>
<td>367640</td>
<td>368634</td>
<td>OR4F16</td>
</tr>
<tr>
<td>chr1</td>
<td>621059</td>
<td>622053</td>
<td>...</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>V1</th>
<th>V2</th>
<th>Gene</th>
<th>Cell_Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>V2</td>
<td>MGH54_P2_C12</td>
<td>Microglia/Macrophage</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

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

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"))
**filterHighPNormals**  
*filterHighPNormals: 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 inferCNV 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**

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

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

**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

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 orderreference_grouped_cell_indices <list> mapping [‘group_name’] to c(cell column indices) for reference (normal) cellsobservation_grouped_cell_indices <list> mapping [‘group_name’] to c(cell column indices) for observation (tumor) cells
tumor_subclusters <list> stores subclustering of tumors if requestedoptions <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: Run Bayesian Network Mixture Model To Obtain Posterior Probabilities For HMM Predicted States

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,
  plotingProbs = 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.
plotingProbs Option for adding plots of Cell and CNV probabilities. (Default: TRUE)
inferCNVBayesNet

quietly: Option to print descriptions along each step. (Default: TRUE)
diagnostics: Option to plot Diagnostic plots and tables. (Default: FALSE)
HMM_type: The type of HMM that was ran, either ‘i3’ or ‘i6’. Determines how many state were predicted by the HMM.
k_obs_groups: Number of groups in which to break the observations. (default: 1)
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)
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)
no_plot: (boolean) Option set by infercnv::run() for producing visualizations.
useRaster: Option to use rasterization when plotting

Value

Returns a MCMC_inferCNV_obj and 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.

Examples

data(infercnv_data_example)
data(infercnv_annots_example)
data(infercnv_genes_example)
data(HMM_states)
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"))

out_dir = tempfile()
infercnv_object_example <- infercnv::run(infercnv_object_example,
cutoff=1,
out_dir=out_dir,
cluster_by_groups=TRUE,
analysis_mode="samples",
denoise=TRUE,
HMM=TRUE,
um_threads=2,
no_plot=TRUE)
mcmc_obj <- infercnv::inferCNVBayesNet(infercnv_obj = infercnv_object_example,
HMM_states = HMM_states,
file_dir = out_dir,
postMcmcMethod = "removeCNV",
out_dir = out_dir,
resume_file_token = "HMMi6.hmm_mode-samples",
quietly = TRUE,
infercnv_data_example

```r
CORES = 2,
plotingProbs = FALSE,
diagnostics = FALSE,
HMM_type = 'i6',
k_obs_groups = 1,
cluster_by_groups = FALSE,
reassignCNVs = 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  

**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**  

**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_cv**

Plot the matrix as a heatmap, with cells as rows and genes as columns, ordered according to chromosome.

### Description

Formats the data and sends it for plotting.

### Usage

```r
plot_cv(
    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

- `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.
plot_cnv

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)

plot_chr_scale
Whether to scale the chromosome width on the heatmap based on their actual size rather than just the number of expressed genes.

chr_lengths
A named list of chromosomes lengths to use when plot_chr_scale=TRUE, or else chromosome size is assumed to be the last chromosome’s stop position + 10k bp

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.

custom_color_pal
Specify a custom set of colors for the heatmap. Has to be in the shape color.palette(c("darkblue", "white", "darkred"), c(2, 2))

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.

write_phylo
Write newick strings of the dendrograms displayed on the left side of the heatmap to file.

useRaster
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 relevent settings used for the plotting to be able to reuse them in another plot call while keeping consistant plotting settings, most importantly x.range.
Examples

```r
# 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)
```

---

**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**

```r
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

infercnv_obj  infercnv_object
on_references  boolean (default=TRUE), plot references (normal cells).
on_observations boolean (default=TRUE), plot observations data (tumor cells).
sample  Whether unique groups of cells should be sampled from or not. (see other parameters for how sampling is done) (Default: FALSE)
n_cells  Number of cells that should be sampled per group if sampling is enabled (default = 1000).
every_n  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)
above_m  Sample only groups that have at least above_m cells if sampling is enabled. (default: 1000) Does not require every_n to be set.
k_obs_groups  Number of groups to break each group in with cutree (in the color bars on the left side of the plot only). (Default: 1)
base_filename  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”) output_format  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")
write_expr_matrix  Includes writing a matrix file containing the expression data that is plotted in the heatmap. (default: FALSE)
save_objects  Whether to save the infercnv objects generated for each group as RDS. (default: FALSE)
**plot_subclusters**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>png_res</td>
<td>Resolution for png output. (Default: 300)</td>
</tr>
<tr>
<td>dynamic_resize</td>
<td>Factor (&gt;= 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)</td>
</tr>
<tr>
<td>useRaster</td>
<td>Whether to use rasterization for drawing heatmap. Only disable if it produces an error as it is much faster than not using it.</td>
</tr>
<tr>
<td>out_dir</td>
<td>Directory in which to save plots and other outputs.</td>
</tr>
</tbody>
</table>

**Value**

void

**Examples**

```r
# 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**

*Plot a heatmap of the data in the infercnv object with the subclusters being displayed as annotations.*

**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_annotations"
)
**run**

**run()**: 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**

```r
run(
  infercnv_obj,
  cutoff = 1,
  min_cells_per_gene = 3,
  out_dir = NULL,
  window_length = 101,
  smooth_method = c("pyramidinal", "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,
)
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
## 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,pyramidinal,coordinates) default: pyramidinal

###

**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. (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 partitionned 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 (default: 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". (default: "modularity")
leiden_resolution_per_chr
resolution parameter for the Leiden algorithm for the per chromosome subclustering (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 subclusters 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 threshold to filter genes used for subclustering. Applied based on reference genes to automatically ignore genes with high expression variability 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_bounda 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.

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 tumor/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 comparisons to normal cells options: "any", "most", "all" (default: "any")

hspike_aggregate_normals: instead of trying to model the different normal groupings individually, just merge them in the hspike.

write_expr_matrix: Whether to write text files with the content of matrices when generating plots (default: FALSE)

write_phylo: Whether to write newick strings of the dendrograms displayed on the left side of the heatmap to file (default: FALSE)

output_format: 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")

plot_chr_scale: Whether to scale the chromosome width on the heatmap based on their actual size rather than just the number of expressed genes.
sample_object 27

chr_lengths  A named list of chromosomes lengths to use when plot_chr_scale=TRUE, or else chromosome size is assumed to be the last chromosome’s stop position + 10k bp

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

up_to_step  run() only up to this exact step number (default: 100 » 23 steps currently in the process)

Value

infercnv_obj containing filtered and transformed data

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,
    analysis_mode="samples",
    no_plot=TRUE)

Description

Apply sampling on an infercnv object to reduce the number of cells in it and allow faster plotting or have all groups take up the same height on the heatmap

Usage

sample_object(
    infercnv_obj,
    n_cells = 100,
    every_n = NULL,
above_m = NULL,
on_references = TRUE,
on_observations = TRUE
)

Arguments

infercnv_obj    infercnv_object
n_cells    Number of cells that should be sampled per group (default = 100).
every_n    Sample 1 cell every_n cells for each group. 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.
above_m    Sample groups that have at least above_m cells. Requires every_n to be set to work, overriding n_cells parameter
on_references    boolean (default=TRUE), sample references (normal cells).
on_observations    boolean (default=TRUE), sample observations data (tumor cells).

Value

sampled infercnv_obj

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_object_example <- infercnv::sample_object(infercnv_object_example, n_cells=5)
# plot result object
validate_infercnv_obj

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, 14

* datasets
  HMM_states, 8
  infercnv_annots_example, 12
  infercnv_data_example, 12
  infercnv_genes_example, 13
  infercnv_object_example, 13
  mcmc_obj, 14
  _PACKAGE (infercnv-package), 3
  add_to_seurat, 3
  apply_median_filtering, 4
  color.palette, 5
  CreateInfercnvObject, 6
  filterHighPNormals, 8
  HMM_states, 8
  infercnv (infercnv-class), 9
  infercnv-class, 9
  infercnv-package, 3
  infercnv_annots_example, 12
  infercnv_data_example, 12
  infercnv_genes_example, 13
  infercnv_object_example, 13
  inferCNVBayesNet, 10
  MCMC_inferCNV (MCMC_inferCNV-class), 14
  MCMC_inferCNV-class, 14
  mcmc_obj, 14
  plot_cnv, 15
  plot_per_group, 17
  plot_subclusters, 19
  run, 21
  sample_object, 27
  validate_infercnv_obj, 29