Package 'fastCNV'

October 28, 2025

Title fastCNV (Modified)

Version 1.1.1

Description Computes CNV quickly on Seurat objects.

License GPL-3

URL https://must-bioinfo.github.io/fastCNV

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports biomaRt,
circlize,
crayon,
ComplexHeatmap,
dplyr,
FNN,
ggplot2,
ggtree,
grDevices,
grid,
paletteer,
purrr,
scales,
Seurat,
SeuratObject,
stats,
stringr,
ape,
phangorn,
utils,
proxy,
graphics

LazyData true

Depends R (>= 3.5.0)

Suggests fastCNVdata, knitr, rmarkdown, patchwork

VignetteBuilder knitr

NeedsCompilation no

Maintainer Gadea Cabrejas <gadea.cabrejas-saiz@u-paris.fr>

Contents

`annotateCNVTree` *Annotate a Phylogenetic Tree with CNV Events*

Description

This function annotates a phylogenetic tree with copy number variation (CNV) events. It identifies significant CNV events in the provided matrix, links them to clones and ancestral nodes, and updates the tree with this information.

Usage

```
annotateCNVTree(tree, cnv_mat, cnv_thresh = 0.15)
```

Arguments

`tree`	A phylogenetic tree (of class `phylo`) that will be annotated.
`cnv_mat`	A matrix of copy number variation (CNV) values, with samples as rows and regions as columns.
`cnv_thresh`	A numeric threshold to filter significant CNV events. Default is 0.15.

Value

A data frame with the tree data, including annotations for CNV events.

Examples

```
cnv_matrix <- structure(c(0.2, 0.4, 0, 0, 0.1, 0, 0.1, 0.2, 0.2), dim = c(
  3L,
  3L
), dimnames = list(c("Clone 1", "Clone 2", "Clone 3"), c(
  "Region 1",
  "Region 2", "Region 3"
)))
tree <- buildCNVTree(cnv_matrix)
tree_data <- annotateCNVTree(tree, cnv_matrix)
```

---

annotations8umTo16um *Project 8μm Spatial Annotation onto 16μm Spots*

Description

This function projects annotations from a high-resolution (8μm) spatial assay onto a lower-resolution (16μm) spatial assay by finding the nearest 8μm spot to each 16μm spot based on spatial coordinates.

Usage

```
annotations8umTo16um(HDobj, referenceVar)
```

Arguments

HDobj	A Seurat object containing both 8μm and 16μm spatial assays (named Spatial.008um and Spatial.016um).
referenceVar	A character string specifying the name of the metadata column in the 8μm assay to project (e.g., a clustering or annotation label).

Details

The function uses FNN::get.knnx() to find the nearest 8μm spot for each 16μm spot based on tissue coordinates. It assigns the annotation from the closest 8μm spot to each 16μm spot. The new annotation column is added to the metadata of HDobj.

Value

A modified Seurat object with a new metadata column named projected_<referenceVar> containing the projected annotation on 16μm spots.

`buildCNVTree` *Construct a Phylogenetic Tree from a Copy Number Variation (CNV) Matrix*

Description

This function constructs a phylogenetic tree based on a given copy number variation (CNV) matrix. It adds a baseline "Normal" profile only to root the tree, which is not shown in final output. First, it computes pairwise distances between profiles using Euclidean distance, and then applies a specified tree-building function (e.g., Neighbor-Joining) to construct the tree.

Usage

```
buildCNVTree(cnv_matrix, tree_function = nj, dist_method = "euclidean")
```

Arguments

`cnv_matrix`	A matrix representing copy number variation, where rows correspond to samples and columns correspond to genomic regions. Each value represents the CNV at a given region in a sample.
`tree_function`	A function to construct the phylogenetic tree from a distance matrix. The default is `nj` (Neighbor-Joining). Other functions (e.g., `upgma`, `wpgma`) can also be used.
`dist_method`	The distance method to be used.

Value

A rooted phylogenetic tree (of class `phylo`)

Examples

```
# Example usage with Neighbor-Joining (default)
```

`CNVAnalysis` *CNVAnalysis Runs Copy Number Variation (CNV) analysis on a Seurat object or a list of Seurat objects.*

Description

This function performs CNV analysis by calculating genomic scores, applying optional denoising, and optionally scaling the results based on a reference population. It processes single-cell or spatial transcriptomics data, generating an additional assay with genomic scores and adding a new metadata column for CNV fractions.

Usage

```
CNVAnalysis(
  object,
  referenceVar = NULL,
  referenceLabel = NULL,
  pooledReference = TRUE,
  scaleOnReferenceLabel = TRUE,
  assay = NULL,
  thresholdPercentile = 0.01,
  geneMetadata = getGenes(),
  windowSize = 150,
  windowStep = 10,
  saveGenomicWindows = FALSE,
  topNGenes = 7000,
  chrArmsToForce = NULL,
  genesToForce = NULL,
  regionToForce = NULL
)
```

Arguments

`object`	A Seurat object or a list of Seurat objects containing the data for CNV analysis. Each object can be either **single-cell** or **spatial transcriptomics** data.
`referenceVar`	The name of the metadata column in the Seurat object that contains reference annotations.
`referenceLabel`	The label within `referenceVar` that specifies the reference population (can be any type of annotation).
`pooledReference`	Logical. If `TRUE` (default), builds a pooled reference across all samples.
`scaleOnReferenceLabel`	Logical. If `TRUE` (default), scales the results based on the reference population.
`assay`	Name of the assay to run the CNV analysis on. Defaults to the results of `prepareCountsForCNVAnalysis` if available.
`thresholdPercentile`	Numeric. Specifies the quantile range to consider (e.g., `0.01` keeps values between the 1st and 99th percentiles). Higher values filter out more background noise.
`geneMetadata`	A dataframe containing gene metadata, typically from Ensembl.
`windowSize`	Integer. Defines the size of genomic windows for CNV analysis.
`windowStep`	Integer. Specifies the step size between genomic windows.
`saveGenomicWindows`	Logical. If `TRUE`, saves genomic window information in the current directory (default = `FALSE`).
`topNGenes`	Integer. The number of top-expressed genes to retain in the analysis.
`chrArmsToForce`	A chromosome arm (e.g., `"8p"`, `"3q"`) or a list of chromosome arms (e.g., `c("3q", "8p", "17p")`) to force into the analysis. If specified, all genes within the given chromosome arm(s) will be included.
`genesToForce`	A list of genes to force into the analysis (e.g. `c("FOXP3","MUC16","SAMD15")`).
`regionToForce`	Chromosome region to force into the analysis (vector containing chr, start, end).

Value

If given a **single** Seurat object, returns the same object with:

- An **additional assay** containing genomic scores per genomic window.
- A new **CNV fraction column** added to the object's metadata. If given a **list** of Seurat objects, returns the modified list.

`CNVCalling` *CNVCalling Performs Copy Number Variation (CNV) analysis on a Seurat object.*

Description

CNVCalling Performs Copy Number Variation (CNV) analysis on a Seurat object.

Usage

```
CNVCalling(
  seuratObj,
  assay = NULL,
  referenceVar = NULL,
  referenceLabel = NULL,
  scaleOnReferenceLabel = TRUE,
  thresholdPercentile = 0.01,
  geneMetadata = getGenes(),
  windowSize = 150,
  windowStep = 10,
  saveGenomicWindows = FALSE,
  topNGenes = 7000,
  chrArmsToForce = NULL,
  genesToForce = NULL,
  regionToForce = NULL
)
```

Arguments

`seuratObj`	A Seurat object containing the data for CNV analysis. Can be either **single-cell** or **spatial transcriptomics** data.
`assay`	Name of the assay to run the CNV analysis on. Defaults to the results of `prepareCountsForCNVAnalysis` if available.
`referenceVar`	The name of the metadata column in the Seurat object that contains reference annotations.
`referenceLabel`	The label within `referenceVar` that specifies the reference population (can be any type of annotation).
`scaleOnReferenceLabel`	Logical. If `TRUE` (default), scales the results based on the reference population.
`thresholdPercentile`	Numeric. Specifies the quantile range to consider (e.g., `0.01` keeps values between the 1st and 99th percentiles). Higher values filter out more background noise.

geneMetadata	A dataframe containing gene metadata, typically from Ensembl.
windowSize	Integer. Defines the size of genomic windows for CNV analysis.
windowStep	Integer. Specifies the step size between genomic windows.
saveGenomicWindows	Logical. If TRUE, saves genomic window information in the current directory (default = FALSE).
topNGenes	Integer. The number of top-expressed genes to retain in the analysis.
chrArmsToForce	A chromosome arm (e.g., "8p", "3q") or a list of chromosome arms (e.g., c("3q", "8p", "17p")) to force into the analysis. If specified, all genes within the given chromosome arm(s) will be included.
genesToForce	A list of genes to force into the analysis (e.g. c("FOXP3","MUC16","SAMD15")).
regionToForce	Chromosome region to force into the analysis (vector containing chr, start, end).

Value

The same Seurat object provided in seuratObj, with:

- An **additional assay** containing genomic scores per genomic window.
- A new **CNV fraction column** added to the object's metadata.

CNVCallingList *CNVCalling for a List of Seurat Objects Performs Copy Number Variation (CNV) analysis on a list of Seurat objects.*

Description

CNVCalling for a List of Seurat Objects Performs Copy Number Variation (CNV) analysis on a list of Seurat objects.

Usage

```
CNVCallingList(
  seuratList,
  assay = NULL,
  referenceVar = NULL,
  referenceLabel = NULL,
  scaleOnReferenceLabel = TRUE,
  thresholdPercentile = 0.01,
  geneMetadata = getGenes(),
  windowSize = 150,
  windowStep = 10,
  saveGenomicWindows = FALSE,
  topNGenes = 7000,
  chrArmsToForce = NULL,
  genesToForce = NULL,
  regionToForce = NULL
)
```

Arguments

`seuratList`	A list of Seurat objects containing the data for CNV analysis. Each object can be either **single-cell** or **spatial transcriptomics** data.
`assay`	Name of the assay to run the CNV analysis on. Defaults to the results of `prepareCountsForCNVAnalysis` if available.
`referenceVar`	The name of the metadata column in the Seurat object that contains reference annotations.
`referenceLabel`	The label within `referenceVar` that specifies the reference population (can be any type of annotation).
`scaleOnReferenceLabel`	Logical. If `TRUE` (default), scales the results based on the reference population.
`thresholdPercentile`	Numeric. Specifies the quantile range to consider (e.g., `0.01` keeps values between the 1st and 99th percentiles). Higher values filter out more background noise.
`geneMetadata`	A dataframe containing gene metadata, typically from Ensembl.
`windowSize`	Integer. Defines the size of genomic windows for CNV analysis.
`windowStep`	Integer. Specifies the step size between genomic windows.
`saveGenomicWindows`	Logical. If `TRUE`, saves genomic window information in the current directory (default = `FALSE`).
`topNGenes`	Integer. The number of top-expressed genes to retain in the analysis.
`chrArmsToForce`	A chromosome arm (e.g., `"8p"`, `"3q"`) or a list of chromosome arms (e.g., `c("3q", "8p", "17p")`) to force into the analysis. If specified, all genes within the given chromosome arm(s) will be included.
`genesToForce`	A list of genes to force into the analysis (e.g. `c("FOXP3","MUC16","SAMD15")`).
`regionToForce`	Chromosome region to force into the analysis (vector containing chr, start, end).

Value

A list of Seurat objects, where each:

- Contains an **additional assay** with genomic scores per genomic window.
- Has a new **CNV fraction column** added to its metadata.

`CNVClassification` *CNV Classification Classifies the CNV results into loss, gain, or no alteration for each observation and chromosome arm.*

Description

CNV Classification Classifies the CNV results into loss, gain, or no alteration for each observation and chromosome arm.

Usage

```
CNVClassification(seuratObj, peaks = c(-0.1, 0, 0.1))
```

Arguments

seuratObj	A Seurat object containing the results of the CNV analysis (e.g., from fastCNV).
peaks	A numeric vector containing the thresholds for classifying CNVs. The default is c(-0.1, 0, 0.1), which defines: • Loss: CNV scores below -0.1 • No alteration: CNV scores between -0.1 and 0.1 • Gain: CNV scores above 0.1

Value

The same Seurat object with an additional classification for each observation and chromosome arm in the metadata. The classification can be one of "loss", "gain", or "no_alteration".

CNVCluster *Perform CNV Clustering with Seurat Object*

Description

The CNVcluster function performs hierarchical clustering on a genomic score matrix extracted from a Seurat object. It provides options for plotting a dendrogram, an elbow plot for optimal cluster determination, and cluster visualization on the dendrogram. The resulting cluster assignments are stored in the Seurat object.

Usage

```
CNVCluster(
  seuratObj,
  referenceVar = NULL,
  tumorLabel = NULL,
  k = NULL,
  h = NULL,
  plotDendrogram = F,
  plotClustersOnDendrogram = F,
  plotElbowPlot = F
)
```

Arguments

seuratObj	A Seurat object containing a "genomicScores" assay with a matrix of genomic scores for clustering.
referenceVar	The name of the metadata column in the Seurat object containing reference annotations.
tumorLabel	The label within referenceVar that specifies the tumor/malignant population (can be any type of annotation).
k	Optional. The number of clusters to cut the dendrogram into. If NULL, the optimal number of clusters is determined automatically using the elbow method.
h	Optional. The height at which to cut the dendrogram for clustering. If both k and h are provided, k takes precedence.

`plotDendrogram` Logical. If `TRUE`, plots the dendrogram. Defaults to `FALSE`.

`plotClustersOnDendrogram`
Logical. If `TRUE`, highlights the clusters on the dendrogram. Defaults to `FALSE`.

`plotElbowPlot` Logical. If `TRUE`, plots the elbow plot used for determining the optimal number of clusters. Defaults to `FALSE`.

Details

The function computes a Manhattan distance matrix and performs hierarchical clustering using the Ward.D2 method. If `k` is not provided, the elbow method is applied to determine the optimal number of clusters based on the within-cluster sum of squares (WSS).

The clusters are assigned to the Seurat object under the metadata column `cnv_clusters`.

Value

A Seurat object with an additional metadata column, `cnv_clusters`, containing the cluster assignments.

`CNVPerChromosomeArm` *CNV Per Chromosome Arm Computes the CNV fraction of each spot/cell per chromosome arm, then stores the results into the metadata.*

Description

CNV Per Chromosome Arm Computes the CNV fraction of each spot/cell per chromosome arm, then stores the results into the metadata.

Usage

```
CNVPerChromosomeArm(seuratObj)
```

Arguments

`seuratObj` A Seurat object, typically the output from the `fastCNV()` function, containing genomic scores for CNV analysis.

Value

The function returns the same Seurat object with the CNV fraction for each chromosome arm added to the metadata.

CNVTree — *Build, annotate and plot a Phylogenetic Tree from a seurat Object containing the CNV results from* `fastCNV()`

Description

Build, annotate and plot a Phylogenetic Tree from a seurat Object containing the CNV results from `fastCNV()`

Usage

```
CNVTree(
  seuratObj,
  healthyClusters = NULL,
  values = "scores",
  cnv_thresh = 0.15,
  tree_function = nj,
  dist_method = "euclidean",
  clone_cols = TRUE
)
```

Arguments

`seuratObj`	A Seurat object containing CNV data and metadata.
`healthyClusters`	A numeric vector or `NULL`. If provided, clusters specified in this vector will be labeled as "Benign" instead of "Clone". Default is `NULL`.
`values`	one of 'scores' or 'calls'. 'scores' returns the mean CNV score per cluster, while 'calls' uses `cnv_thresh` to establish a cut-off for gains and losses, returning a matrix of CNV calls (0=none, 1=gain, -1=loss).
`cnv_thresh`	A numeric threshold to filter significant CNV events. Default is 0.15.
`tree_function`	A function to construct the phylogenetic tree from a distance matrix. The default is `nj` (Neighbor-Joining). Other functions (e.g., `upgma`, `wpgma`) can also be used.
`dist_method`	The distance method to be used.
`clone_cols`	a color palette to color the clones. If NULL, points are not colored. If TRUE, clones are colored using default color palette. If a palette is given, clones are colored following the palette, with values passed to `scale_color_manual`.

computeAverageExpression — *Compute average expression for patients*

Description

This function calculates the average gene expression for each patient across different cell types. It first retrieves patient data from LN, then extracts the corresponding count data from LrawcountsByPatient, and calculates the mean expression.

Usage

```
computeAverageExpression(LN, LrawcountsByPatient)
```

Arguments

LN	A list where each element represents a cell type with sublists containing patient data.
LrawcountsByPatient	A named list where each element contains count data for a specific patient.

Value

A named vector containing the average expression for each patient.

---

fastCNV *fastCNV calls all of the internal functions needed to compute the putative CNV on a Seurat object or a list of Seurat objects*

Description

This function orchestrates the CNV analysis on a Seurat object (or multiple objects). It calls internal functions such as `prepareCountsForCNVAnalysis`, `CNVAnalysis`, `CNVPerChromosomeArm`, `CNVcluster`, and `PlotCNVResults` to compute the CNVs, perform clustering, and generate heatmaps. The results are saved in the metadata of the Seurat object(s), with options for generating and saving plots.

Usage

```
fastCNV(
  seuratObj,
  sampleName,
  referenceVar = NULL,
  referenceLabel = NULL,
  assay = NULL,
  prepareCounts = TRUE,
  aggregFactor = 15000,
  seuratClusterResolution = 0.8,
  aggregateByVar = TRUE,
  reClusterSeurat = FALSE,
  pooledReference = TRUE,
  scaleOnReferenceLabel = TRUE,
  thresholdPercentile = 0.01,
  geneMetadata = getGenes(),
  windowSize = 150,
  windowStep = 10,
  saveGenomicWindows = FALSE,
  topNGenes = 7000,
  chrArmsToForce = NULL,
  genesToForce = NULL,
  regionToForce = NULL,
  getCNVPerChromosomeArm = TRUE,
```

```
  getCNVClusters = TRUE,
  tumorLabel = NULL,
  k_clusters = NULL,
  h_clusters = NULL,
  plotDendrogram = FALSE,
  plotClustersOnDendrogram = FALSE,
  plotElbowPlot = FALSE,
  mergeCNV = TRUE,
  mergeThreshold = 0.98,
  doPlot = TRUE,
  denoise = TRUE,
  printPlot = FALSE,
  savePath = ".",
  outputType = "png",
  clustersVar = "cnv_clusters",
  splitPlotOnVar = clustersVar,
  referencePalette = "default",
  clusters_palette = "default"
)
```

Arguments

`seuratObj`	Seurat object or list of Seurat objects to perform the CNV analysis on.
`sampleName`	Name of the sample or a list of names corresponding to the samples in the `seuratObj`.
`referenceVar`	The variable name of the annotations in the Seurat metadata to be used as reference.
`referenceLabel`	The label given to the observations you want as reference (can be any type of annotation).
`assay`	Name of the assay to run the CNV on. Takes the results of `prepareCountsForCNVAnalysis` by default if available.
`prepareCounts`	If `FALSE`, will not run the `prepareCountsForCNVAnalysis` function (default = `TRUE`).
`aggregFactor`	The number of counts per spot desired (default = 15 000). If less than 1,000, will not run the `prepareCountsForCNVAnalysis` function.
`seuratClusterResolution`	The resolution wanted for the Seurat clusters (default = 0.8).
`aggregateByVar`	If `referenceVar` is given, determines whether to use it to pool the observations (default = `TRUE`).
`reClusterSeurat`	Whether to re-cluster if the Seurat object given already has a `seurat_clusters` slot in its metadata (default = `FALSE`).
`pooledReference`	Default is `TRUE`. Will build a pooled reference across all samples if `TRUE`.
`scaleOnReferenceLabel`	If `TRUE`, scales the results depending on the normal observations (default = `TRUE`).
`thresholdPercentile`	Which quantiles to take (default 0.01). For example, `0.01` will take quantiles between 0.01-0.99. Background noise appears with higher numbers.

`geneMetadata` List of genes and their metadata (default uses genes from Ensembl version 113).

`windowSize` Size of the genomic windows for CNV analysis (default = 150).

`windowStep` Step between the genomic windows (default = 10).

`saveGenomicWindows`
If `TRUE`, saves the information of the genomic windows in the current directory (default = `FALSE`).

`topNGenes` Number of top expressed genes to keep (default = 7000).

`chrArmsToForce` A chromosome arm (e.g., `"8p"`, `"3q"`) or a list of chromosome arms (e.g., `c("3q", "8p", "17p")`) to force into the analysis.

`genesToForce` A list of genes to force into the analysis (e.g. `c("FOXP3","MUC16","SAMD15")`).

`regionToForce` Chromosome region to force into the analysis (vector containing chr, start, end).

`getCNVPerChromosomeArm`
If `TRUE`, will save the CNV per chromosome arm into the metadata.

`getCNVClusters` If `TRUE`, will perform clustering on the CNV scores and save them in the metadata of the Seurat object as `cnv_clusters`.

`tumorLabel` The label within `referenceVar` that specifies the tumor/malignant population (can be any type of annotation).

`k_clusters` Optional. Number of clusters to cut the dendrogram into. If `NULL`, the optimal number of clusters is determined automatically using the elbow method.

`h_clusters` Optional. The height at which to cut the dendrogram for clustering. If both `k` and `h` are provided, `k` takes precedence.

`plotDendrogram` Logical. Whether to plot the dendrogram (default = `FALSE`).

`plotClustersOnDendrogram`
Logical. Whether to highlight clusters on the dendrogram (default = `FALSE`).

`plotElbowPlot` Logical. Whether to plot the elbow plot used for determining the optimal number of clusters (default = `FALSE`).

`mergeCNV` Logical. Whether to merge the highly correlated CNV clusters.

`mergeThreshold` A numeric value between 0 and 1. Clusters with correlation greater than this threshold will be merged. Default is 0.98.

`doPlot` If `TRUE`, will build a heatmap for each of the samples (default = `TRUE`).

`denoise` If `TRUE`, the denoised data will be used in the heatmap (default = `TRUE`).

`printPlot` If `TRUE`, the heatmap will be printed in the console (default = `FALSE`, the plot will only be saved in a PDF).

`savePath` Path to save the heatmap plot. If `NULL`, the plot won't be saved (default = `.`).

`outputType` Specifies the file format for saving the plot, either `"png"` or `"pdf"` (default = `"png"`).

`clustersVar` The variable name of the clusters in the Seurat metadata (default = `"cnv_clusters"`).

`splitPlotOnVar` The name of the metadata column to split the observations during the `plotCNVResults` step, if different from `referenceVar`.

`referencePalette`
The color palette that should be used for `referenceVar` (default = `"default"`).

`clusters_palette`
The color palette that should be used for `clustersVar` (default = `"default"`).

Value

A list of Seurat objects after all the analysis is complete. Heatmaps of the CNVs for every object in `seuratObj` are generated and saved in the specified path (default = current working directory).

`fastCNV_10XHD` *fastCNV_10XHD calls all of the internal functions needed to compute the putative CNV on a Seurat Visium HD object or a list of Seurat Visium HD objects*

Description

This function orchestrates the CNV analysis on a Seurat Visium HD object (or multiple objects). It calls internal functions such as `CNVAnalysis` and `PlotCNVResults` to compute the CNVs and generate heatmaps. The results are saved in the metadata of the Seurat object(s), with options for generating and saving plots.

Usage

```
fastCNV_10XHD(
  seuratObjHD,
  sampleName,
  referenceVar = NULL,
  referenceLabel = NULL,
  assay = "Spatial.016um",
  pooledReference = TRUE,
  scaleOnReferenceLabel = TRUE,
  thresholdPercentile = 0.01,
  geneMetadata = getGenes(),
  windowSize = 150,
  windowStep = 10,
  saveGenomicWindows = FALSE,
  topNGenes = 7000,
  chrArmsToForce = NULL,
  genesToForce = NULL,
  regionToForce = NULL,
  getCNVPerChromosomeArm = TRUE,
  getCNVClusters = FALSE,
  tumorLabel = NULL,
  k_clusters = NULL,
  h_clusters = NULL,
  plotDendrogram = FALSE,
  plotClustersOnDendrogram = FALSE,
  plotElbowPlot = FALSE,
  mergeCNV = TRUE,
  mergeThreshold = 0.98,
  doPlot = TRUE,
  denoise = TRUE,
  printPlot = FALSE,
  savePath = ".",
  outputType = "png",
  clustersVar = "cnv_clusters",
  clusters_palette = "default",
  splitPlotOnVar = clustersVar,
  referencePalette = "default"
)
```

Arguments

seuratObjHD	Seurat object or list of Seurat objects to perform the CNV analysis on.
sampleName	Name of the sample or a list of names corresponding to the samples in the seuratObj.
referenceVar	The variable name of the annotations in the Seurat metadata to be used as reference.
referenceLabel	The label given to the observations you want as reference (can be any type of annotation).
assay	Name of the assay to run the CNV on. Takes the results of prepareCountsForCNVAnalysis by default if available.
pooledReference	Default is TRUE. Will build a pooled reference across all samples if TRUE.
scaleOnReferenceLabel	If TRUE, scales the results depending on the normal observations (default = TRUE).
thresholdPercentile	Which quantiles to take (default 0.01). For example, 0.01 will take quantiles between 0.01-0.99. Background noise appears with higher numbers.
geneMetadata	List of genes and their metadata (default uses genes from Ensembl version 113).
windowSize	Size of the genomic windows for CNV analysis (default = 150).
windowStep	Step between the genomic windows (default = 10).
saveGenomicWindows	If TRUE, saves the information of the genomic windows in the current directory (default = FALSE).
topNGenes	Number of top expressed genes to keep (default = 7000).
chrArmsToForce	A chromosome arm (e.g., "8p", "3q") or a list of chromosome arms (e.g., c("3q", "8p", "17p")) to force into the analysis.
genesToForce	A list of genes to force into the analysis (e.g. c("FOXP3","MUC16","SAMD15")).
regionToForce	Chromosome region to force into the analysis (vector containing chr, start, end).
getCNVPerChromosomeArm	If TRUE, will save the CNV per chromosome arm into the metadata.
getCNVClusters	If TRUE, will perform clustering on the CNV scores and save them in the metadata of the Seurat object as cnv_clusters.
tumorLabel	The label within referenceVar that specifies the tumor/malignant population (can be any type of annotation).
k_clusters	Optional. Number of clusters to cut the dendrogram into. If NULL, the optimal number of clusters is determined automatically using the elbow method.
h_clusters	Optional. The height at which to cut the dendrogram for clustering. If both k and h are provided, k takes precedence.
plotDendrogram	Logical. Whether to plot the dendrogram (default = FALSE).
plotClustersOnDendrogram	Logical. Whether to highlight clusters on the dendrogram (default = FALSE).
plotElbowPlot	Logical. Whether to plot the elbow plot used for determining the optimal number of clusters (default = FALSE).
mergeCNV	Logical. Whether to merge the highly correlated CNV clusters.

`mergeThreshold` A numeric value between 0 and 1. Clusters with correlation greater than this threshold will be merged. Default is 0.98.

`doPlot` If `TRUE`, will build a heatmap for each of the samples (default = `TRUE`).

`denoise` If `TRUE`, the denoised data will be used in the heatmap (default = `TRUE`).

`printPlot` If `TRUE`, the heatmap will be printed in the console (default = `FALSE`, the plot will only be saved in a PDF).

`savePath` Path to save the heatmap plot. If `NULL`, the plot won't be saved (default = `.`).

`outputType` Specifies the file format for saving the plot, either `"png"` or `"pdf"` (default = `"png"`).

`clustersVar` The name of the metadata column containing cluster information (default = `"cnv_clusters"`).

`clusters_palette` A color palette for `clustersVar`. You can provide a custom palette as a vector of color codes (e.g., `c("#F8766D", "#A3A500", "#00BF7D")`).

`splitPlotOnVar` The name of the metadata column to split the observations during the `plotCNVResults` step, if different from `referenceVar`.

`referencePalette` A color palette for `referenceVar`. You can provide a custom palette as a vector of color codes (e.g., `c("#FF0000", "#00FF00")`).

Value

A Seurat object or a list of Seurat objects after all the analysis is complete. Heatmaps of the CNVs for every object in `seuratObj` are generated and saved in the specified path (default = current working directory).

`geneMetadata` *Genes Data from Ensembl Version 113*

Description

Data downloaded from the Ensembl website (version 113), containing detailed gene information for approximately 76,000 genes. The dataset includes Ensembl gene IDs, HUGO nomenclature (HGNC symbol), Entrez gene IDs, chromosome locations, gene biotype, and gene length for each gene.

Usage

```
data(geneMetadata)
```

Format

An object of class `list`, containing gene information as described above.

Source

Ensembl Genome Browser, Version 113: https://www.ensembl.org/index.html

Examples

```
data(geneMetadata)
hgnc <- geneMetadata$hgnc_symbol
entrez <- geneMetadata$entrezgene_id
```

`generateCNVClonesMatrix` *Generate CNV Matrix for CNV Clusters by Chromosome Arm*

Description

This function generates a matrix of metacells where each metacell corresponds to a CNV cluster. The CNV matrix is calculated by chromosome arm. If specified, certain clusters will be labeled as "Benign" rather than "Clone".

Usage

```
generateCNVClonesMatrix(
  seuratObj,
  healthyClusters = NULL,
  values = "scores",
  cnv_thresh = 0.15
)
```

Arguments

`seuratObj`	A Seurat object containing CNV data and metadata.
`healthyClusters`	A numeric vector or `NULL`. If provided, clusters specified in this vector will be labeled as "Benign" instead of "Clone". Default is `NULL`.
`values`	one of 'scores' or 'calls'. 'scores' returns the mean CNV score per cluster, while 'calls' uses `cnv_thresh` to establish a cut-off for gains and losses, returning a matrix of CNV calls (0=none, 1=gain, -1=loss).
`cnv_thresh`	A numeric threshold to filter significant CNV events. Default is 0.15.

Value

A matrix of CNVs with row names corresponding to the clone or benign labels and columns representing the chromosome arms, with values corresponding to CNV scores or CNV calls.

getGenes *Download Gene Information from Ensembl*

Description

This function retrieves gene information from the Ensembl database using the specified filters. It can either fetch the latest data or use cached data if available.

Usage

```
getGenes(filters = NULL, cache = TRUE)
```

Arguments

filters	A character vector of filters to be applied in the query. These filters determine which genes and their associated information are returned from the Ensembl database.
cache	Logical. If TRUE, the function will use cached data if available. If FALSE, it will download the latest version of the gene data from Ensembl.

Value

A list containing gene information retrieved from Ensembl, with each element representing data for a specific gene (e.g., gene IDs, descriptions, associated attributes).

mergeCNVClusters *Merge CNV Clusters in a Seurat Object*

Description

This function merges CNV clusters in a Seurat object based on the correlation of their average CNV profiles across chromosome arms. Clusters with correlation greater than a user-specified threshold are merged into a single cluster.

Usage

```
mergeCNVClusters(seuratObj, mergeThreshold = 0.98)
```

Arguments

seuratObj	A Seurat object containing fastCNV's results by chromosome arm, and CNV clustering.
mergeThreshold	A numeric value between 0 and 1. Clusters with correlation greater than this threshold will be merged. Default is 0.98.

Value

A Seurat Object with updated CNV clusters, where highly correlated clusters have been merged.

`plotCNVResults` *Plot CNV Results into a Heatmap Builds a heatmap to visualize the CNV results based on genomic scores.*

Description

Plot CNV Results into a Heatmap Builds a heatmap to visualize the CNV results based on genomic scores.

Usage

```
plotCNVResults(
  seuratObj,
  referenceVar = NULL,
  clustersVar = "cnv_clusters",
  splitPlotOnVar = clustersVar,
  denoise = TRUE,
  savePath = ".",
  printPlot = FALSE,
  referencePalette = "default",
  clusters_palette = "default",
  outputType = "png"
)
```

Arguments

`seuratObj`	A Seurat object containing the genomic scores computed previously.
`referenceVar`	The name of the metadata column in the Seurat object containing reference annotations.
`clustersVar`	The name of the metadata column containing cluster information (default = `"cnv_clusters"`).
`splitPlotOnVar`	The name of the metadata column used to split the heatmap rows (e.g., cell type or cluster) (default = `clustersVar`).
`denoise`	If `TRUE`, the denoised data will be used in the heatmap (default = `TRUE`).
`savePath`	The path where the heatmap will be saved. If `NULL`, the plot will not be saved (default = `"."`).
`printPlot`	Logical. If `TRUE`, prints the heatmap to the console.
`referencePalette`	A color palette for `referenceVar`. You can provide a custom palette as a vector of color codes (e.g., `c("#FF0000", "#00FF00")`).
`clusters_palette`	A color palette for `clustersVar`. You can provide a custom palette as a vector of color codes (e.g., `c("#F8766D", "#A3A500", "#00BF7D")`).
`outputType`	Character. Specifies the file format for saving the plot, either `"png"` or `"pdf"`.

Value

This function generates a heatmap and saves it as a `.pdf` or `.png` file in the specified path (default = working directory).

plotCNVResultsHD *Plot Visium HD CNV Results into a Heatmap Builds a heatmap to visualize the Visium HD CNV results based on genomic scores.*

Description

Plot Visium HD CNV Results into a Heatmap Builds a heatmap to visualize the Visium HD CNV results based on genomic scores.

Usage

```
plotCNVResultsHD(
  seuratObjHD,
  referenceVar = NULL,
  clustersVar = "cnv_clusters",
  splitPlotOnVar = clustersVar,
  denoise = TRUE,
  savePath = ".",
  printPlot = FALSE,
  referencePalette = "default",
  clusters_palette = "default",
  outputType = "png"
)
```

Arguments

seuratObjHD	A Seurat object containing the genomic scores computed previously.
referenceVar	The name of the metadata column in the Seurat object containing reference annotations.
clustersVar	The name of the metadata column containing cluster information (default = "cnv_clusters").
splitPlotOnVar	The name of the metadata column used to split the heatmap rows (e.g., cell type or cluster) (default = clustersVar).
denoise	If TRUE, the denoised data will be used in the heatmap (default = TRUE).
savePath	The path where the heatmap will be saved. If NULL, the plot will not be saved (default = ".").
printPlot	Logical. If TRUE, prints the heatmap to the console.
referencePalette	A color palette for referenceVar. You can provide a custom palette as a vector of color codes (e.g., c("#FF0000", "#00FF00")).
clusters_palette	A color palette for clustersVar. You can provide a custom palette as a vector of color codes (e.g., c("#F8766D", "#A3A500", "#00BF7D")).
outputType	Character. Specifies the file format for saving the plot, either "png" or "pdf".

Value

This function generates a heatmap and saves it as a .pdf or .png file in the specified path (default = working directory).

plotCNVTree *Plot an Annotated Phylogenetic Tree with CNV Events*

Description

This function generates a plot of an annotated phylogenetic tree using ggtree. It displays tip labels, tip points, and labels for CNV events associated with each node.

Usage

```
plotCNVTree(tree_data, clone_cols = NULL)
```

Arguments

tree_data	A data frame containing tree structure and annotations, typically produced by annotateCNVtree.
clone_cols	a color palette to color the clones. If NULL, points are not colored. If TRUE, clones are colored using default color palette. If a palette is given, clones are colored following the palette, with values passed to scale_color_manual.

Value

A ggplot object representing the annotated phylogenetic tree.

Examples

```
cnv_matrix <- structure(c(0.2, 0.4, 0, 0, 0.1, 0, 0.1, 0.2, 0.2), dim = c(
  3L,
  3L
), dimnames = list(c("Clone 1", "Clone 2", "Clone 3"), c(
  "Region 1",
  "Region 2", "Region 3"
)))
tree <- buildCNVTree(cnv_matrix)
tree_data <- annotateCNVTree(tree, cnv_matrix)
plotCNVTree(tree_data)
```

prepareCountsForCNVAnalysis

Aggregate Observations by Cell Type for CNV Analysis Aggregates observations with the same cell types to increase counts per observation, improving Copy Number Variation (CNV) computation.

Description

Aggregate Observations by Cell Type for CNV Analysis Aggregates observations with the same cell types to increase counts per observation, improving Copy Number Variation (CNV) computation.

Usage

```
prepareCountsForCNVAnalysis(
  seuratObj,
  sampleName = NULL,
  referenceVar = NULL,
  aggregateByVar = T,
  aggregFactor = 15000,
  seuratClusterResolution = 0.8,
  reClusterSeurat = F
)
```

Arguments

`seuratObj`	A Seurat object containing the data.
`sampleName`	A character string specifying the sample name.
`referenceVar`	The name of the metadata column in the Seurat object that contains reference annotations.
`aggregateByVar`	Logical. If `TRUE` (default), aggregates observations based on `referenceVar` annotations.
`aggregFactor`	Integer. The target number of counts per observation (default = `15 000`).
`seuratClusterResolution`	Numeric. The resolution used for Seurat clustering (default = `0.8`).
`reClusterSeurat`	Logical. If `TRUE`, re-runs clustering on the Seurat object.

Value

A Seurat object with:

- A new assay called **"AggregatedCounts"** containing the modified count matrix.
- **Seurat clusters** stored in the metadata.

Index