Package 'ELViS'

November 13, 2026

Title An R Package for Estimating Copy Number Levels of Viral Genome Segments Using Base-Resolution Read Depth Profile

Version 1.5.0

Description Base-resolution copy number analysis of viral genome. Utilizes base-resolution read depth data over viral genome to find copy number segments with two-dimensional segmentation approach. Provides publish-ready figures, including histograms of read depths, coverage line plots over viral genome annotated with copy number change events and viral genes, and heatmaps showing multiple types of data with integrative clustering of samples.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

VignetteBuilder knitr

biocViews CopyNumberVariation, Coverage, GenomicVariation, BiomedicalInformatics, Sequencing, Normalization, Visualization, Clustering

LazyData false

BugReports https://github.com/hyochoi/ELViS/issues

URL https://github.com/hyochoi/ELViS

Config/testthat/edition 3

Imports reticulate, BiocGenerics, circlize, ComplexHeatmap, data.table, dplyr, GenomicFeatures, GenomicRanges, ggplot2, glue, graphics, grDevices, igraph, IRanges, magrittr, memoise, methods, parallel, patchwork, scales, segclust2d, stats, stringr, txdbmaker, utils, uuid, zoo

Depends R (>= 4.5.0)

Suggests Rsamtools, BiocManager, knitr, testthat (>= 3.0.0)

git_url https://git.bioconductor.org/packages/ELViS

git_branch devel

git_last_commit a918785

git_last_commit_date 2026-04-28

Repository Bioconductor 3.24

Date/Publication 2026-11-12

Author Hyo Young Choi [aut, cph] (ORCID:
<https://orcid.org/0000-0002-7627-8493>),
Jin-Young Lee [aut, cre, cph] (ORCID:
<https://orcid.org/0000-0002-5366-7488>),
Xiaobei Zhao [ctb] (ORCID: <https://orcid.org/0000-0002-5277-0846>),
Jeremiah R. Holt [ctb] (ORCID: <https://orcid.org/0000-0002-5201-5015>),
Katherine A. Hoadley [aut] (ORCID:
<https://orcid.org/0000-0002-1216-477X>),
D. Neil Hayes [aut, fnd, cph] (ORCID:
<https://orcid.org/0000-0001-6203-7771>)

Maintainer Jin-Young Lee <jlee307@uthsc.edu>

Contents

coord_to_grng 2
coord_to_lst 3
depth_hist 3
ELViS 4
ELViS_toy_run_result 5
filt_samples 5
gene_cn_heatmaps 6
get_depth_matrix 7
get_new_baseline 9
integrative_heatmap 10
mtrx_samtools_reticulate 12
norm_fun 12
plot_pileUp_multisample 13
run_ELViS 14
total_aligned_base__host_and_virus 15
toy_example 16

Index **17**

`coord_to_grng` *Convert coordinate string to grng object*

Description

Convert coordinate string to grng object

Usage

```
coord_to_grng(coord)
```

Arguments

`coord`	string in the form of "chr1:123-456" or "chr1:1,234-5,678,912"

Value

GRanges object corresponding to the input coordinate string

Examples

```
coord_to_grng("chr1:123-456")
coord_to_grng("chr1:1,234-5,678,912")
```

`coord_to_lst` *Convert coordinate string to list of chr,start and end*

Description

Convert coordinate string to list of chr,start and end

Usage

```
coord_to_lst(coord)
```

Arguments

`coord`	string in the form of "chr1:123-456" or "chr1:1,234-5,678,912"

Value

a list of 3 elements. Chromosome name, start position and end position.

Examples

```
coord_to_lst("chr1:123-456")
coord_to_lst("chr1:1,234-5,678,912")
```

`depth_hist` *Sample filtering threshold examination plot.*

Description

Sample filtering threshold examination plot.

Usage

```
depth_hist(mtrx, th = 50, title_txt = NULL, smry_fun = max, ...)
```

Arguments

`mtrx`	Matrix or data.frame. Rows are positions and columns are samples.
`th`	Numeric. Sample filtering threshold
`title_txt`	figure title.
`smry_fun`	function to calculate summary metric to apply sample filter threshold to
`...`	additional argument for `smry_fun` argument.

Value

ggplot2 object

Examples

```
data(mtrx_samtools_reticulate)
th <- 50
depth_hist(mtrx_samtools_reticulate,th=th,smry_fun=max)
depth_hist(mtrx_samtools_reticulate,th=th,smry_fun = quantile,prob=0.95)
depth_hist(mtrx_samtools_reticulate,th=th,smry_fun = mean)
```

ELViS *ELViS : An R Package for Estimating Copy Number Levels of Viral Genome Segments Using Base-Resolution Read Depth Profile.*

Description

Base-resolution copy number analysis of viral genome. Utilizes base-resolution read depth data over viral genome to find copy number segments with two-dimensional segmentation approach. Provides publish-ready figures, including histograms of read depths, coverage line plots over viral genome annotated with copy number change events and viral genes, and heatmaps showing multiple types of data with integrative clustering of samples.

Functions

- get_depth_matrix : Generate a read depth matrix of positions x samples from input BAM files list.
- run_ELViS : Run ELViS using input raw depth matrix.
- raw depth matrix.

Author(s)

Maintainer: Jin-Young Lee <jlee307@uthsc.edu> (ORCID) [copyright holder]

Authors:

- Hyo Young Choi <hyoyoung.choi@uthsc.edu> (ORCID) [copyright holder]
- Katherine A. Hoadley <hoadley@med.unc.edu> (ORCID)
- D. Neil Hayes <Neil.Hayes@uthsc.edu> (ORCID) [funder, copyright holder]

Other contributors:

- Xiaobei Zhao <xiaobei_zhao@uthsc.edu> (ORCID) [contributor]
- Jeremiah R. Holt <jholt41@uthsc.edu> (ORCID) [contributor]

See Also

Useful links:

- https://github.com/hyochoi/ELViS
- Report bugs at https://github.com/hyochoi/ELViS/issues

ELViS_toy_run_result *ELViS Toy Example - Run Result*

Description

List containing ELViS run result

Usage

```
data(ELViS_toy_run_result)
```

Format

ELViS_toy_run_result:
A list of

is_reduced_output Indicates whether this results is a reduced form

final_output Final base-resolution segmentation output

final_call Indices of samples in which copy number variants were detected

new_Y_p2 Normalized read depth

filt_samples *Filtering samples based on summary statistic*

Description

Filtering samples based on summary statistic

Usage

```
filt_samples(mtrx, th = 50, smry_fun = max)
```

Arguments

mtrx	matrix or data.frame. Rows are positions and columns are samples.
th	Sample filtering threshold (Default : 50)
smry_fun	function to generate summary value of samples, which is used for filtering. (Default : max)

Value

matrix or data.frame. data matrix with low depth samples filtered out.

Examples

```
data(mtrx_samtools_reticulate)
th<-50
filtered_mtrx <- filt_samples(mtrx_samtools_reticulate,th=th,smry_fun=max)
```

gene_cn_heatmaps *Gene Copy Number Heatmap*

Description

Gene Copy Number Heatmap

Usage

```
gene_cn_heatmaps(
  X_raw,
  result,
  gff3_fn,
  gene_ref,
  baseline = 1,
  exclude_genes,
  col_cn = colorRamp2(c(0.5, 1, 1.5), c(muted("blue"), "white", muted("red"))),
  heatmap_height = unit(1.5, "in")
)
```

Arguments

X_raw	Raw depth matrix
result	Run result
gff3_fn	gene annotation file name
gene_ref	The name of the gene to set as reference for relative gene dosage heatmap
baseline	Vector of state numbers to use as baseline for each sample. If it is single integer, then the given state number is used for all samples. (Default : 1)
exclude_genes	name of genes to exclude from the annotation track (Default : NULL)
col_cn	relative gene dosage color palette. (Default : colorRamp2(c(0.5,1,1.5),c(muted("blue"),"whit
heatmap_height	heatmap height specified using unit function. (Default : unit(1.5,"in"))

Value

a ComplexHeatmap Heatmap List object

Examples

```
# gff3 gene model file
package_name <- "ELViS"
gff3_fn <- system.file("extdata","HPV16REF_PaVE.gff",package = package_name)

# loading precalculated depth matrix
data(mtrx_samtools_reticulate)

# threshold
th <- 50
```

```
# filtered matrix
base_resol_depth <- filt_samples(mtrx_samtools_reticulate,th=th,smry_fun=max)

# viral load data
data(total_aligned_base__host_and_virus)
viral_load <- (10^6)*(apply(base_resol_depth,2,\(x) sum(x)) )/total_aligned_base__host_and_virus

# load ELViS run result
data(ELViS_toy_run_result)
result <- ELViS_toy_run_result

# genes to exclude from plotting
exclude_genes <- c("E6*","E1^E4","E8^E2")

# heatmap of gene dosage
gene_ref<-"E7"

gene_cn <-
  gene_cn_heatmaps(
    X_raw = base_resol_depth,
    result = result,
    gff3_fn = gff3_fn,
    baseline = 1,
    gene_ref = gene_ref,
    exclude_genes = exclude_genes
    )

gene_cn
```

`get_depth_matrix` *Generate a read depth matrix of positions x samples from input BAM files list*

Description

Generate a read depth matrix of positions x samples from input BAM files list

Usage

```
get_depth_matrix(
  bam_files,
  mode = "samtools_reticulate",
  coord_or_target_virus_name,
  is_virus = TRUE,
  N_cores = detectCores(),
  min_mapq = 30,
  min_base_quality = 0,
  max_depth = 1e+05,
  modules = NULL,
  envs = NULL,
```

```
  tmpdir = tempdir(),
  samtools = NULL,
  condaenv = "env_samtools",
  condaenv_samtools_version = "auto"
)
```

Arguments

bam_files	Vector containing bam file names in character
mode	Mode of read depth calculation. Either of c("samtools_reticulate","samtools_custom","Rsamt are acceptable. If run on Windows OS, it will coerced to "Rsamtools" (Default : "samtools_reticulate")
coord_or_target_virus_name	The name of the target virus. This should be equal to the name of the sequence in the FASTA file reads are aligned to.
is_virus	logical indicating if the coord_or_target_virus_name is for viral genome(TRUE) or non-viral genome(FALSE) (default : TRUE)
N_cores	Number of cores to use for parallel processing (Default : min(10,available cores))
min_mapq	Minimum MAPQ. (Default : 30)
min_base_quality	Minimum basecall quality score (Default : 0)
max_depth	(Rsamtools) Maximum read depth. (Default : 1e5)
modules	(samtools) Environment modulefile name. (Default : NULL)
envs	(samtools) Environmental variables for samtools. (Default : NULL)
tmpdir	(samtools) Temporary file directory (Default : tempdir())
samtools	(samtools) Absolute path to samtools executable (Default : NULL)
condaenv	(samtools_reticulate) Name of the conda environment in which samtools are installed. If no environment with this name is available, one will be created. (Default : "env_samtools")
condaenv_samtools_version	(samtools_reticulate) The version of samtools to install in the conda environment using basilisk. If set to "auto", the latest available version of samtools will be installed. (Default : "auto")

Value

a matrix of positions x samples containing base-resolution raw read depth

Examples

```
package_name <- "ELViS"

# The name of the target virus
# in the reference sequence FASTA file used for alignment.
# Can be check by samtools view -H input.bam
target_virus_name <- "gi|333031|lcl|HPV16REF.1|"

# get bam file pathes
ext_path <- system.file("extdata",package = package_name)
bam_files <- list.files(ext_path,full.names = TRUE,pattern = "bam$")
```

```
# number of threads to use
N_cores <- 1L

# get read depth matrix
tmpdir <- tempdir()

mtrx_samtools_reticulate <-
 get_depth_matrix(
  bam_files = bam_files,coord_or_target_virus_name = target_virus_name,is_virus = TRUE
 ,mode = "samtools_reticulate"
 ,N_cores = N_cores
 ,min_mapq = 30
 ,tmpdir=tempdir()
 ,condaenv = "env_samtools"
 )
```

`get_new_baseline` *Get new baselines according to criteria user designates*

Description

Get new baselines according to criteria user designates

Usage

```
get_new_baseline(result, mode = "longest")
```

Arguments

`result`	Run result
mode	Indicate how new baseline should be set ("longest","shortest")

Value

a integer vector indicating new baseline index for each sample

Examples

```
# its usage example is given in vignette in detail

data(ELViS_toy_run_result)
result <- ELViS_toy_run_result

get_new_baseline(result,mode="longest")
```

`integrative_heatmap` *Plot heatmaps based on simple integrative clustering of multiple matrices*

Description

Plot heatmaps based on simple integrative clustering of multiple matrices

Usage

```
integrative_heatmap(
  X_raw,
  result,
  gff3_fn,
  exclude_genes,
  col_pal_gene = col_yarrr_info2,
  col_cn = colorRamp2(c(0.5, 1, 1.5), c(muted("blue"), "white", muted("red"))),
  col_y = colorRamp2(c(0.5, 1, 2), c(muted("blue"), "white", muted("red"))),
  col_z = colorRamp2(c(-4, 0, 4), c(muted("blue"), "white", muted("red"))),
  col_x_scaled = "auto",
  col_vl = "auto",
  baseline = 1,
  matrices_to_plot = "all",
  matrices_integ_cluster = "all",
  total_aligned_base__host_and_virus = NULL,
  return_data_matrices = FALSE
)
```

Arguments

`X_raw`	Raw depth matrix
`result`	Run result
`gff3_fn`	gene annotation file name
`exclude_genes`	name of genes to exclude from the annotation track (Default : NULL)
`col_pal_gene`	color palette for gene colors
`col_cn`	Color scheme for copy number heatmap (Default :`colorRamp2(c(0.5,1,1.5),c(muted("blue"),"`
`col_y`	Color scheme for normalized read depth(Y) heatmap (Default : `colorRamp2(c(0.5,1,2),c(muted(`
`col_z`	Color scheme for Z-score heatmap (Default : `colorRamp2(c(-4,0,4),c(muted("blue"),"white",`
`col_x_scaled`	Color scheme for scaled raw depth(X) heatmap (Default : `"auto"`)
`col_vl`	Color scheme for positional viral load heatmap (Default : `"auto"`)
`baseline`	Vector of state numbers to use as baseline for each sample. If it is single integer, then the given state number is used for all samples. (Default : 1)
`matrices_to_plot`	Names and orders of the matrices to show as heatmap. Any permutation of `c("CN","Y","Z","X_Scaled","Viral_Load")` of any length is allowed. The vertical orders of stacked heatmaps follows the order of this vector. If set to `"all"`, `c("CN","Y","Z","X_Scaled","Viral_Load")` is used. (Default : `"all"`)

`matrices_integ_cluster`
: Names of the matrices to be used for integrative clustering for column orders. Any combination of `c("CN","Y","Z","X_Scaled","Viral_Load")` of length > 1 is allowed. If the length is less then 2, then it is ignored and the first matrix specified in `matrices_to_plot` argument is used for column ordering. The vertical orders of stacked heatmaps follows the order of this vector. If set to `"all"`, `c("CN","Y","Z","X_Scaled","Viral_Load")` is used. (Default : `"all"`)

`total_aligned_base__host_and_virus`
: Total aligned bases for each sample(i.e. from picard,gatk,qualimap). Used to calculate positional load of viral DNA. Makes sense if regions in host genome are also included in the target panel. Ignored if set to NULL. (Default : NULL)

`return_data_matrices`
: boolean whether to return the data matrices used. (Default : `FALSE`)

Value

A ComplexHeatmap Heatmap List object vertically stacked

Examples

```
# gff3 gene model file
package_name <- "ELViS"
gff3_fn <- system.file("extdata","HPV16REF_PaVE.gff",package = package_name)

# loading precalculated depth matrix
data(mtrx_samtools_reticulate)

# threshold
th <- 50

# filtered matrix
base_resol_depth <- filt_samples(mtrx_samtools_reticulate,th=th,smry_fun=max)

# viral load data
data(total_aligned_base__host_and_virus)
viral_load <- (10^6)*(apply(base_resol_depth,2,\(x) sum(x)) )/total_aligned_base__host_and_virus

# load ELViS run result
data(ELViS_toy_run_result)
result <- ELViS_toy_run_result

# genes to exclude from plotting
exclude_genes <- c("E6*","E1^E4","E8^E2")

# heatmap based on integrative clustering
integ_ht_result <- integrative_heatmap(
  X_raw = base_resol_depth,
  result = result,
  gff3_fn = gff3_fn,
  exclude_genes = exclude_genes,
  baseline=1,
  total_aligned_base__host_and_virus = total_aligned_base__host_and_virus
  )

integ_ht_result
```

mtrx_samtools_reticulate *ELViS Toy Example - Base-Resolution Raw Read Depth*

Description

Base-resolution raw read depth profile over viral genome

Usage

```
data(mtrx_samtools_reticulate)
```

Format

`mtrx_samtools_reticulate`:
A matrix with 7906 rows and 120 columns

norm_fun *Normalization - scaling by certain quantile*

Description

Normalization - scaling by certain quantile

Usage

```
norm_fun(x, probs = 0.75)
```

Arguments

x	numeric vector to normalize.
probs	a single numeric value of probabilities in [0,1] used for normalization.(Default = 0.75)

Value

numeric vector of normalized values

Examples

```
norm_fun(seq_len(5))
# [1] 0.25 0.50 0.75 1.00 1.25
```

`plot_pileUp_multisample` *Get a list of pile-up plots over multiple samples*

Description

Get a list of pile-up plots over multiple samples

Usage

```
plot_pileUp_multisample(
  result,
  X_raw,
  target_indices = NULL,
  plot_target = "x",
  gff3_fn,
  baseline = 1,
  annot_margin = 0.01,
  arrow_spacing = 0.05,
  geme_name_space = 0.5,
  col_pal = col_yarrr_info2,
  col_cn_baseline = "#708C98",
  col_pal_cn = col_yarrr_info2[-5],
  exclude_genes = NULL,
  annot_plot_ratio = 0.3
)
```

Arguments

`result`	analysis result
`X_raw`	input raw depth matrix
`target_indices`	sample indices to plot
`plot_target`	target data type to plot (Default : `"x"`)
`gff3_fn`	gene annotation file name
`baseline`	the state index to set as baseline (Default : `1`)
`annot_margin`	minimum of margin between gene annotations allowed. As a fraction of plotting area. (Default : `0.01`)
`arrow_spacing`	gene annotation arrow spacing. As a fraction of plotting area. (Default : `0.05`)
`geme_name_space`	the height of white space reserved for gene names in the annotation. (Default : `0.5`)
`col_pal`	gene color palette
`col_cn_baseline`	color for baseline (Default : `"#708C98"`)
`col_pal_cn`	color palette for non-baseline copy number states
`exclude_genes`	name of genes to exclude from the annotation track (Default : NULL)
`annot_plot_ratio`	ratio of the annotation plot under the pileup plot

Value

a list of pile-up ggplot object

Examples

```
# gff3 gene model file
package_name <- "ELViS"
gff3_fn <- system.file("extdata","HPV16REF_PaVE.gff",package = package_name)

# loading precalculated depth matrix
data(mtrx_samtools_reticulate)

# threshold
th <- 50

# filtered matrix
base_resol_depth <- filt_samples(mtrx_samtools_reticulate,th=th,smry_fun=max)

data(ELViS_toy_run_result)
result <- ELViS_toy_run_result

# get line plots for shape-change samples
gg_lst_x <-
plot_pileUp_multisample(
  result = result,
  X_raw = base_resol_depth,
  plot_target = "x",
  gff3 = gff3_fn,
  baseline=1,
  exclude_genes = c("E6*","E1^E4","E8^E2"),
  target_indices = result$final_call$cnv_samples[seq_len(3)]
  )

gg_lst_x[[1]]
```

---

run_ELViS *Run ELViS using input raw depth matrix*

Description

Run ELViS using input raw depth matrix

Usage

```
run_ELViS(
  X,
  N_cores = min(10L, detectCores()),
  reduced_output = TRUE,
  verbose = FALSE,
  save_intermediate_data = FALSE,
```

```
  save_dir = "save_dir",
  overwrite = FALSE
)
```

Arguments

X	Raw depth matrix of position x samples
N_cores	The number of cores to use (Default : min(10L,detectCores()))
reduced_output	logical indicating whether to return only reduced output
verbose	logical whether to print out information for debugging
save_intermediate_data	logical indicating whether to save intermediate data as rds file. (default : FALSE)
save_dir	Name of the directory to save intermediate files in. (default : "save_dir")
overwrite	logical indicating whether to overwrite intermediate files. (default : FALSE)

Value

list containing ELViS run results

Examples

```
data(mtrx_samtools_reticulate)
th<-50
filtered_mtrx <- filt_samples(mtrx_samtools_reticulate,th=th,smry_fun=max)

result <- run_ELViS(filtered_mtrx[,seq_len(5)],N_cores=1L)
```

---

total_aligned_base__host_and_virus

ELViS Toy Example - Total Aligned Base

Description

Total aligned base both to host and viral genome.

Usage

```
data(total_aligned_base__host_and_virus)
```

Format

total_aligned_base__host_and_virus:
A vector of length 120

`toy_example` *ELViS Toy Example - Metadata*

Description

Metadata of samples in the toy examples

Usage

```
data(toy_example)
```

Format

`toy_example`**:**
A data frame with 120 rows and 6 columns:

VarType Variant type. Set to control if there is no variant
Copies_Altered The number of copies duplicated or deleted
Event_Size Variant size
Mean_Depth Mean read depth

Index

∗ **datasets**
ELViS_toy_run_result, 5
mtrx_samtools_reticulate, 12
total_aligned_base__host_and_virus, 15
toy_example, 16

coord_to_grng, 2
coord_to_lst, 3

depth_hist, 3

ELViS, 4
ELViS-package *(*ELViS*)*, 4
ELViS_toy_run_result, 5

filt_samples, 5

gene_cn_heatmaps, 6
get_depth_matrix, 7
get_new_baseline, 9

integrative_heatmap, 10

mtrx_samtools_reticulate, 12

norm_fun, 12

plot_pileUp_multisample, 13

run_ELViS, 14

total_aligned_base__host_and_virus, 15
toy_example, 16