Package 'seqArchRplus'

May 9, 2024

Type Package

Title Downstream analyses of promoter sequence architectures and HTML report generation

Version 1.5.0

Description seqArchRplus facilitates downstream analyses of promoter sequence architectures/clusters identified by seqArchR (or any other tool/method). With additional available information such as the TPM values and interquantile widths (IQWs) of the CAGE tag clusters, seqArchRplus can order the input promoter clusters by their shape (IQWs), and write the cluster information as browser/IGV track files. Provided visualizations are of two kind: per sample/stage and per cluster visualizations. Those of the first kind include: plot panels for each sample showing per cluster shape, TPM and other score distributions, sequence logos, and peak annotations. The second include per cluster chromosome-wise and strand distributions, motif occurrence heatmaps and GO term enrichments. Additionally, seqArchRplus can also generate HTML reports for easy viewing and comparison of promoter architectures between samples/stages.

License GPL-3

URL https://github.com/snikumbh/seqArchRplus

BugReports https://github.com/snikumbh/seqArchRplus/issues

Depends R (>= 4.2), GenomicRanges, IRanges, S4Vectors

Imports BiocParallel, Biostrings, BSgenome, ChIPseeker, cli, clusterProfiler, cowplot, factoextra, GenomeInfoDb, ggplot2, ggimage, graphics, grDevices, gridExtra, heatmaps, magick, methods, RColorBrewer, scales, seqArchR, seqPattern, stats, utils

Suggests BSgenome.Dmelanogaster.UCSC.dm6, BiocStyle, CAGEr (>= 2.0.2), covr, knitr (>= 1.22), org.Dm.eg.db, pdftools, rmarkdown (>= 1.12), slickR, TxDb.Dmelanogaster.UCSC.dm6.ensGene, xfun

VignetteBuilder knitr

biocViews Annotation, Visualization, ReportWriting, GO, MotifAnnotation, Clustering

Encoding UTF-8

LazyData false

RoxygenNote 7.2.3

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

git_branch devel

git_last_commit 67eccab

git_last_commit_date 2024-04-30

Repository Bioconductor 3.20

Date/Publication 2024-05-08

Author Sarvesh Nikumbh [aut, cre, cph]
(<https://orcid.org/0000-0003-3163-4447>)

Maintainer Sarvesh Nikumbh <sarvesh.nikumbh@gmail.com>

Contents

curate_clusters 2
form_combined_panel 6
generate_all_plots 7
generate_html_report 10
handle_tc_from_cage 13
iqw_tpm_plots 15
order_clusters_iqw 17
per_cluster_annotations 18
per_cluster_go_term_enrichments 21
per_cluster_seqlogos 23
per_cluster_strand_dist 25
plot_motif_heatmaps 27
plot_motif_heatmaps2 28
seqArchRplus 30
seqs_acgt_image 31
write_seqArchR_cluster_track_bed 32

Index **36**

`curate_clusters` *Curate clusters from seqArchR result*

Description

seqArchR result stores the clusters obtained at every iteration. It is possible that the previously chosen agglomeration and/or distance method used with hierarchical clustering does not yield reasonable clusters. This function enables minor curation of the clusters obtained from the hierarchical clustering step.

Usage

```
curate_clusters(
  sname,
  use_aggl = "ward.D",
  use_dist = "euclid",
  seqArchR_result,
  iter,
  pos_lab = NULL,
  regularize = TRUE,
  topn = 50,
  use_cutk = 2,
  need_change = NULL,
  change_to = NULL,
  final = FALSE,
  dir_path = NULL
)
```

Arguments

sname	The sample name
use_aggl	The agglomeration method to be used for hierarchical clustering. This is passed on to 'seqArchR::collate_seqArchR_result'. See argument 'aggl_method' in collate_seqArchR_result.
use_dist	The distance method to be used for hierarchical clustering. This is passed on to collate_seqArchR_result. See argument 'dist_method' in collate_seqArchR_result.
seqArchR_result	The seqArchR result object.
iter	Specify which iteration of the seqArchR result should be used for obtaining clusters.
pos_lab	The position labels
regularize	Logical. Specify TRUE if the basis vector comparison is to be regularized. Requires you to set `topn` which is set to 50 as default. See argument 'regularize' in collate_seqArchR_result.
topn	Numeric. The top N features (nucleotide-position pairs) that will be used for distance computation, rest will be ignored. See argument 'topn' in collate_seqArchR_result.
use_cutk	Value of K (number of clusters) for cutting the hierarchical clustering tree.
need_change	A list of elements (cluster IDs in the input clusters) that need re-assignment. Elements
change_to	A list of elements (cluster IDs in the input clusters) to be assigned to those that need re-assignment. In case there is a candidate that needs to be put into a new, independent cluster of itself, use 0 (as numeric). Both `need_change` and `change_to` should be empty lists if no re-assignment is to be performed.
final	Logical, set to TRUE or FALSE
dir_path	The /path/to/the/directory where files will be written. Default is NULL.

Details

This function helps the user work through the curation in at most three steps.

1. This function performs hierarchical clustering on the seqArchR clusters (of the specified iteration) to obtain a clustering result. The resulting clustering is visualized by a dendrogram, color-coded cluster assignments, and corresponding sequence logos. Using this visualization, the user can identify/estimate the (nearly right) number of clusters to cut the dendrogram. The **first call** uses K = 2.

2. Visually examine and count the tentative number of clusters (K) deemed right. Because these architectures are now arranged by the hierarchical clustering order, identifying this tentative value of K is much easier. Call the function this **second** time with the identified value of K. Look at the visualization now generated to determine if it is good enough, i.e., it requires only minor re-assignments.

3. Identify cases of cluster assignments that you wish to re-assign to different clusters. These can be noted as a list and supplied in a subsequent call to the function.

4. In the **final call** to the function, set 'final = TRUE', supply the re- assignments as two lists 'need_change' and 'change_to'.

More on re-assignments using arguments `need_change` and `change_to`: If any element is to be put into a new cluster, use a numeral 0 in change_to. This can be done in either scenario: when any element is re-assigned as a singleton cluster of itself, or as clustered with some other element coming from some other existing cluster. Consider, for instance, among some 33 clusters identified by seqArchR, the following re-assignments are executed.

Original_clustering <- list(c(1), c(2,3), c(4,5), c(6, 7)) need_change <- list(c(5), c(2), c(3, 6)) change_to <- list(1, 0, 0)

In the above, element 5 is re-assigned to the cluster containing element 1. Element 2 is re-assigned to a new, singleton cluster of itself, while elements 3 and 6 (which initially can belong to same/any two clusters) are collated together. Note that it is important to use `c()`.

Also see examples below.

Value

This function returns a list holding (a) 'curation_plot': plot showing the dendrogram + sequence logos, (b) 'clust_assignments': the cluster (re-)assignments performed, and (c) 'clusters_list': the sequence clusters as a list.

to help perform curation and document it.

When 'final = FALSE', the 'curation_plot' shows the dendrogram + sequence logos of clusters (ordered by `hclust` ordering). The 'clusters_list' holds the hclust ordered clusters. If the 'dir_path' is specified, a PDF file showing the same figure is also written at the location using the default filename '<Sample_name>_dend_arch_list_reg_top50_euclid_complete_<K>clusters.pdf'.

When 'final = TRUE', 'clusters_list' holds the clusters with re-assignments executed, and 'curation_plot' of dendrogram + sequence logos now has an additional panel showing the sequence logos upon collation of clusters as specified by the re-assignments. Also the cluster IDs in the dendrograms have colors showing the re-assignments, i.e., elements that were re-assigned to different clusters, also have appropriate color changes reflected.

When 'dir_path' is provided, the curation plot is written to disk using the same filename as before except for suffix 'final' attched to it.

Author(s)

Sarvesh Nikumbh

Examples

```
library(Biostrings)
raw_seqs <- Biostrings::readDNAStringSet(
                          filepath = system.file("extdata",
                            "example_promoters45.fa.gz",
                            package = "seqArchRplus",
                            mustWork = TRUE)
                        )

use_clusts <- readRDS(system.file("extdata", "example_clust_info.rds",
         package = "seqArchRplus", mustWork = TRUE))

seqArchR_result <- readRDS(system.file("extdata", "seqArchR_result.rds",
         package = "seqArchRplus", mustWork = TRUE))

## The example seqArchR result generally holds the raw sequences, but in
## this case, the result object is devoid of them in order to reduce size of
## example data
if(!("rawSeqs" %in% names(seqArchR_result)))
    seqArchR_result$rawSeqs <- raw_seqs

use_aggl <- "complete"
use_dist <- "euclid"

## get seqArchR clusters custom curated
seqArchR_clusts <- seqArchRplus::curate_clusters(sname = "sample1",
    use_aggl = use_aggl, use_dist = use_dist,
    seqArchR_result = seqArchR_result, iter = 5,
    pos_lab = NULL, regularize = FALSE, topn = 50,
    use_cutk = 5, final = FALSE, dir_path = tempdir())

## Form the lists need_change and change_to for re-assignments
need_change <- list(c(2))
change_to <- list(c(1))

## This fetches us clusters with custom/curated collation in _arbitrary_
## order. See the next function call to order_clusters_iqw that orders
## these clusters by their median/mean IQW

seqArchR_clusts <- seqArchRplus::curate_clusters(sname = "sample1",
    use_aggl = use_aggl, use_dist = use_dist,
    seqArchR_result = seqArchR_result, iter = 5,
    pos_lab = NULL, regularize = FALSE, topn = 50,
    use_cutk = 5,
    need_change = need_change, change_to = change_to,
    final = TRUE, dir_path = tempdir())
```

`form_combined_panel` *Form a combined panel of three plots*

Description

For a given sample, this function forms a combined panel of three plots namely, the IQW-TPM boxplots, cluster sequence logos, and annotations per cluster. All of these individual plots can be generated using existing seqArchRplus functions

Usage

```
form_combined_panel(iqw_tpm_pl, seqlogos_pl, annot_pl)
```

Arguments

`iqw_tpm_pl`	The IQW-TPM plot generated using `iqw_tpm_plots`
`seqlogos_pl`	The sequence logos oneplot obtained from `per_cluster_seqlogos` by setting the argument `` `one_plot = TRUE` ``
`annot_pl`	The annotations oneplot obtained from `per_cluster_annotations` by setting the argument `` `one_plot = TRUE` ``

Details

This functionality requires cowplot. The combined plot panels are arranged horizontally.

Value

This function returns a `ggplot` plot object.

Author(s)

Sarvesh Nikumbh

`generate_all_plots` *Generate all plots for a given sample*

Description

Generate all plots for a given sample

Usage

```
generate_all_plots(
  sname,
  bed_info_fname,
  custom_colnames = NULL,
  seqArchR_clusts,
  raw_seqs,
  cager_obj = NULL,
  tc_gr = NULL,
  use_q_bound = FALSE,
  use_as_names = NULL,
  order_by_iqw = TRUE,
  use_median_iqw = TRUE,
  iqw = TRUE,
  tpm = TRUE,
  cons = FALSE,
  pos_lab = NULL,
  txdb_obj = NULL,
  orgdb_obj = NULL,
  org_name = NULL,
  qLow = 0.1,
  qUp = 0.9,
  tss_region = c(-500, 100),
  raw_seqs_mh = NULL,
  motifs = c("WW", "SS", "TATAA", "CG"),
  motif_heatmaps_flanks = c(50, 100, 200),
  motif_heatmaps_res = 300,
  motif_heatmaps_dev = "png",
  dir_path,
  txt_size = 25
)
```

Arguments

`sname`	The sample name
`bed_info_fname`	The BED filename with information on tag clusters. See details for expected columns (column names)/information
`custom_colnames`	Specify custom column/header names to be used with the BED file information

`seqArchR_clusts`
: The seqArchR clusters' list

`raw_seqs`
: The sequences corresponding to the cluster elements (also available from the seqArchR result object)

`cager_obj`
: The CAGEr object. This expects that `clusterCTSS` has been run beforehand. Default is NULL

`tc_gr`
: The tag clusters as a `GRanges`. Default is NULL

`use_q_bound`
: Logical. Write the lower and upper quantiles as tag cluster boundaries in BED track files with tag clusters. Default is TRUE

`use_as_names`
: Specify the column name from info_df which you would like to use as names for display with the track. By default, 'use_names' is NULL, and the sequence/tag cluster IDs are used as names

`order_by_iqw`
: Logical. Set TRUE to order clusters by the IQW median or mean. Set argument 'use_median_iqw' to TRUE to use the median, else will use mean if FALSE

`use_median_iqw`
: Logical. Set TRUE if the median IQW for each cluster is used to order the clusters. Otherwise, the mean IQW will be used when set to FALSE

`iqw`, `tpm`, `cons`
: Logical. Specify TRUE when the corresponding plots should be included

`pos_lab`
: The position labels

`txdb_obj`
: The TranscriptsDB object for the organism

`orgdb_obj`
: The OrgDb object for the organism

`org_name`
: The organism name. This is used in the tracknames for tracks writte nas BED files

`qLow`, `qUp`
: Numeric values between 0 and 1. These are required when cager_obj is provided instead of the tag clusters 'tc_gr'

`tss_region`
: For ChIPseeker, "region range of TSS"

`raw_seqs_mh`
: Specify the sequences to be used for motif heatmaps, if they are different than the sequences clustered by seqArchr. Default is NULL, when 'raw_seqs" are used

`motifs`
: Specify a character vector of motif words (using IUPAC notation) to be visualized as a heatmap

`motif_heatmaps_flanks`
: Specify a vector of different flank values to be considered for visualization. Same size flanks are considered upstream as well as downstream, hence one value suffices for eac hvisualization. When a vector 'c(50, 100, 200)' is specified, three motif heatmap files (three separate PNG files) are created, each with one flank size. The motif heatmap file will contain separate heatmaps for each of the specified motifs in the 'motifs' argument

`motif_heatmaps_res`
: The resolution for the motif heatmaps. Default is 300 ppi

`motif_heatmaps_dev`
: The device to be used for plotting. Default is "png". Other options available are "tiff", "pdf", and "svg"

`dir_path`
: The path to the directory where files are saved

`txt_size` The text size to be used for the plots. This is some high value because the plots are written with ‘dpi=300‘ and are often large in size, especially the combined panel plots

Details

The expected columns (and column names) in the BED file are "chr", "start", "end", "width", "strand", "score", "nr_ctss", "dominant_ctss", "domTPM", "q_<qLow>", "q_<qUp>", "IQW", "tpm". Depending on the values for arguments qLow and qUp, the corresponding column names are formed. For example, if ‘qLow‘ and ‘qUp‘ are 0.1 and 0.9, the column names are "q_0.1" and "q_0.9". These columns are mostly present by default in the CAGEr tag clusters.

The supplied clusters are ordered by their mean/median interquantile widths before proceeding to generate the visualizations.

Value

A list holding generated plots; some which are directly written to disk are not included in this list.

The included plots are:

- Boxplots of IQW (and TPM and conservation score when available) distributions for each cluster (as a single combined plot) - Annotation percentages per cluster as stacked barplots (as a single combined plot) - Annotation percentages per cluster as stacked barplots as a list - Sequence logos of all cluster architectures (as a single combined plot) - Sequence logos of all cluster architectures as a list - Strand-separated sequence logos of all cluster architectures as a list - Per cluster distribution of tag clusters on chromosomes and strands

In addition, the following plots are written to disk: - Visualization of all clustered sequences as a matrix - Visualization of motif occurrences (for specified motifs) in all clustered sequences

In addition, the individual clusters from seqArchR are written to disk as BED track files that can be viewed in the genome browser/IGV.

Author(s)

Sarvesh Nikumbh

Examples

```
library(GenomicRanges)
library(TxDb.Dmelanogaster.UCSC.dm6.ensGene)
library(ChIPseeker)
library(Biostrings)

bed_fname <- system.file("extdata", "example_info_df.bed.gz",
         package = "seqArchRplus", mustWork = TRUE)

## info_df <- read.delim(file = bed_fname,
##         sep = "\t", header = TRUE)


tc_gr <- readRDS(system.file("extdata", "example_tc_gr.rds",
         package = "seqArchRplus", mustWork = TRUE))
```

```
use_clusts <- readRDS(system.file("extdata", "example_clust_info.rds",
         package = "seqArchRplus", mustWork = TRUE))

raw_seqs <- Biostrings::readDNAStringSet(
                          filepath = system.file("extdata",
                            "example_promoters45.fa.gz",
                            package = "seqArchRplus",
                            mustWork = TRUE)
                        )

raw_seqs_mh <- Biostrings::readDNAStringSet(
                          filepath = system.file("extdata",
                            "example_promoters200.fa.gz",
                            package = "seqArchRplus",
                            mustWork = TRUE)
                        )


all_plots <- generate_all_plots(sname = "sample1",
                    bed_info_fname = bed_fname,
                    seqArchR_clusts = use_clusts,
                    raw_seqs = raw_seqs,
                    tc_gr = tc_gr,
                    use_q_bound = FALSE,
                    order_by_iqw = FALSE,
                    use_median_iqw = TRUE,
                    iqw = TRUE, tpm = TRUE, cons = FALSE,
                    pos_lab = -45:45,
                    txdb_obj = TxDb.Dmelanogaster.UCSC.dm6.ensGene,
                    org_name = "Dmelanogaster22",
                    qLow = 0.1, qUp = 0.9,
                    tss_region = c(-500, 100),
                    raw_seqs_mh = raw_seqs_mh,
                    motifs = c("WW", "SS", "TATAA", "CG"),
                    motif_heatmaps_flanks = c(50, 100, 200),
                    motif_heatmaps_res = 150,
                    motif_heatmaps_dev = "png",
                    dir_path = tempdir(),
                    txt_size = 25)
```

`generate_html_report` *Generate HTML report with scrollable combined panel plots*

Description

This function generates an HTML report with large scrollable combined panels for multiple samples that eases comparison of changes between samples

Usage

```
generate_html_report(
  snames,
  file_type = "PDF",
  img_ht = "1200px",
  img_wd = "1600px",
  page_wd = "1800px",
  render_silently = TRUE,
  dir_path
)
```

Arguments

snames	Sample names to be included
file_type	"PDF" (default) or "SVG" (to be supported in the future). The type of files to look for in the sample-specific results folder
img_ht, img_wd	The height and width (in pixels) of the images in the output HTML report. Default values are '1200px' (height) and '1600px' (width)
page_wd	The width of the body in the HTML. Default is '1800px'.
render_silently	Logical. TRUE or FALSE
dir_path	Specify the '/path/to/directory' where sample-specific results folders are located. This is a required argument and cannot be NULL. A directory named 'combined_results' is created at the given location, and the HTML report is written into it

Details

This functionality requires suggested libraries `slickR` and `pdftools` installed. The function assumes requires that the candidate figure files have combined_panel Note that the combined plot panels are arranged horizontally and therefore are best viewed in wide desktop monitors.

Value

Nothing. Report is written to disk at the provided `dir_path` using the filename 'Combined_panels_report_samples_<s

Author(s)

Sarvesh Nikumbh

Examples

```
## Need these packages to run these examples
if(require("slickR", "pdftools")){

## Make IQW-TPM plots

bed_fname <- system.file("extdata", "example_info_df.bed.gz",
```

```
        package = "seqArchRplus", mustWork = TRUE)

info_df <- read.delim(file = bed_fname,
        sep = "\t", header = TRUE,
        col.names = c("chr", "start", "end", "width",
                "dominant_ctss", "domTPM",
                "strand", "score", "nr_ctss",
                "q_0.1", "q_0.9", "IQW", "tpm"))

use_clusts <- readRDS(system.file("extdata", "example_clust_info.rds",
        package = "seqArchRplus", mustWork = TRUE))

use_dir <- tempdir()

iqw_tpm_pl <- iqw_tpm_plots(sname = "sample1",
                            dir_path = use_dir,
                            info_df = info_df,
                            iqw = TRUE,
                            tpm = TRUE,
                            cons = FALSE,
                            clusts = use_clusts,
                            txt_size = 14)

## Make sequence logos
library(Biostrings)
raw_seqs <- Biostrings::readDNAStringSet(
                           filepath = system.file("extdata",
                             "example_promoters45.fa.gz",
                             package = "seqArchRplus",
                             mustWork = TRUE)
                         )


seqlogo_oneplot_pl <- per_cluster_seqlogos(sname = "sample1",
                                    seqs = raw_seqs,
                                    clusts = use_clusts,
                                    pos_lab = -45:45,
                                    bits_yax = "auto",
                                    strand_sep = FALSE,
                                    one_plot = TRUE,
                                    dir_path = use_dir,
                                    txt_size = 14)

## Need the TxDb object to run these examples

annotations_pl <- NULL
if(require("TxDb.Dmelanogaster.UCSC.dm6.ensGene")){

    annotations_pl <- per_cluster_annotations(sname = "sample1",
                          clusts = NULL,
                          tc_gr = bed_fname,
                          txdb_obj = TxDb.Dmelanogaster.UCSC.dm6.ensGene,
                          one_plot = FALSE,
```

```
                          dir_path = use_dir,
                          tss_region = c(-500,100))
}

## Combine them together
if(!is.null(annotations_pl)){
    panel_pl <- form_combined_panel(iqw_tpm_pl = iqw_tpm_pl,
                    seqlogos_pl = seqlogos_oneplot_pl,
                    annot_pl = annotations_oneplot_pl)
 }else{
    panel_pl <- form_combined_panel(iqw_tpm_pl = iqw_tpm_pl,
                    seqlogos_pl = seqlogos_oneplot_pl)

 }

 cowplot::save_plot(filename = file.path(use_dir,
                                  paste0("sample1_combined_panel.pdf")),
                   plot = panel_pl)

## Call function to generate HTML report
generate_html_report(snames = c("sample1", "sample1"),
                dir_path = use_dir)
}
```

---

handle_tc_from_cage *Handle writing of tag clusters (TCs) from CAGE experiment to disk as BED files.*

Description

Handle writing of tag clusters obtained from a CAGE experiment to disk as BED files. It can also return corresponding promoter sequences as a DNAStringSet object or write them to disk as FASTA files.

For information on tag clusters obtained by clustering of CAGE-derived TSS, we refer the reader to the CAGEr vignette, Section 3.7 https://www.bioconductor.org/packages/release/bioc/vignettes/CAGEr/inst/doc/CAGEexp.html#ctss-clustering

Usage

```
handle_tc_from_cage(
  sname,
  tc_gr,
  cager_obj,
  qLow = 0.1,
  qUp = 0.9,
  fl_size_up = 500,
  fl_size_down = 500,
  bsgenome,
```

```
  dir_path = NULL,
  fname_prefix = NULL,
  fname_suffix = NULL,
  write_to_disk = TRUE,
  ret_seqs = TRUE
)
```

Arguments

sname	The sample name
tc_gr	Tag clusters as GRanges. If 'cager_obj' is not provided (is NULL), this argument is required. Also note that the columns should match the columns (with mcols) as returned by tagClusters. If 'tc_gr' is provided, 'cager_obj' is ignored
cager_obj	A CAGEexp object obtained from the CAGEr package, if and when CAGEr was used to process the raw CAGE data
qLow, qUp	The interquantile boundaries to be considered for obtaining tag clusters from the CAGEexp object. See tagClusters
fl_size_up, fl_size_down	Numeric. The size of the flanks in the upstream and downstream directions.
bsgenome	The BSgenome file that will be used to obtain sequences of the organism from.
dir_path	The path to the directory where files will be written. By default, all BED files are written within a subdirectory named "BED", and all FASTA files are written within a subdirectory named "FASTA", both created at the 'dir_path' location.
fname_prefix, fname_suffix	Specify any prefix or suffix string to be used in the filename. This can be the organism name etc. Specify without any separator. By default, an underscore is used as a separator in the filename.
write_to_disk	Logical. Specify TRUE to write files to disk. More specifically, BED files are written to disk only when this is set to TRUE. For promoter sequences, FASTA files are written to disk if this arg is set to TRUE, otherwise not. and a DNAStringSet object is returned if 'ret_seqs' is set to TRUE.
ret_seqs	Logical. Specify TRUE if promoter sequences are to be returned as a DNAStringSet object.

Details

You can use the fname_prefix and fname_suffix arguments to specify strings to be used as prefix and suffix for the files. For example, the organism name can be used as a prefix for the filename. Similarly, for suffix.

Value

If 'ret_seqs' is TRUE, a DNAStringSet object is returned. Depending on 'write_to_disk', files are written to disk at the specified location.

If 'ret_seq = TRUE', the promoter sequences are returned as a DNAStringSet object.

Author(s)

Sarvesh Nikumbh

Examples

```
if(require("BSgenome.Dmelanogaster.UCSC.dm6")){

  tc_gr <- readRDS(system.file("extdata", "example_tc_gr.rds",
         package = "seqArchRplus", mustWork = TRUE))

  seqs <- seqArchRplus::handle_tc_from_cage(sname = "sample1",
                                tc_gr = tc_gr,
                                fl_size_up = 500,
                                fl_size_down = 500,
                                dir_path = NULL,
                                fname_prefix = "pre",
                                fname_suffix = "suff",
                                write_to_disk = FALSE,
                                bsgenome = BSgenome.Dmelanogaster.UCSC.dm6,
                                ret_seqs = TRUE)

}
```

---

iqw_tpm_plots *IQW, TPM plots*

Description

IQW, TPM plots

Usage

```
iqw_tpm_plots(
  sname,
  info_df,
  clusts,
  iqw = TRUE,
  tpm = TRUE,
  cons = TRUE,
  order_by_median = TRUE,
  dir_path = NULL,
  txt_size = 12,
  use_suffix = NULL,
  use_prefix = "C",
  wTitle = TRUE
)
```

Arguments

sname	Sample name
info_df	A DataFrame object holding information on the tag clusters. Expected columns (names) are 'chr', 'start', 'end', 'IQW', 'domTPM', and 'strand'.
clusts	List of sequence ids in each cluster.
iqw	Logical. Specify TRUE if boxplots of interquantile widths (IQW) of the tag clusters corresponding to the promoters in each cluster are to be plotted.
tpm	Logical. Specify TRUE if boxplots of TPM values for all clusters are to be plotted.
cons	Logical. Specify TRUE if boxplots of conservation scores (PhastCons scores) for all clusters. If this is TRUE, an additional column named 'cons' is expected in the 'info_df'.
order_by_median	Logical. Whether to order to clusters by their median (when TRUE) or mean (when FALSE) interquantile widths.
dir_path	The /path/to/the/directory where files will be written. Default is NULL.
txt_size	Specify text size to be used in the plots.
use_suffix, use_prefix	Character. Specify any suffix and/or prefix you wish to add to the filename.
wTitle	Logical. If TRUE, the returned plot will contain a default title, which is the same as the filename. See details.

Details

The plots are written to a file named "Sample_<sample_name>_IQW_TPM_Cons_plot.pdf" if all of 'iqw', 'tpm', and 'cons' are set to TRUE. This is also set as the plot title if 'wTitle' is set to TRUE.

All plots are arranged by the IQWs (smallest on top, largest at the bottom), even if 'iqw' is set to FALSE.

Value

The plot(s) as a ggplot2 object. The order of the plots is IQW, followed by TPM values (when specified), followed by conservation scores (when specified).

Author(s)

Sarvesh Nikumbh

Examples

```
bed_fname <- system.file("extdata", "example_info_df.bed.gz",
         package = "seqArchRplus", mustWork = TRUE)

info_df <- read.delim(file = bed_fname,
         sep = "\t", header = TRUE,
```

```
          col.names = c("chr", "start", "end", "width",
                  "dominant_ctss", "domTPM",
                  "strand", "score", "nr_ctss",
                  "q_0.1", "q_0.9", "IQW", "tpm"))

use_clusts <- readRDS(system.file("extdata", "example_clust_info.rds",
          package = "seqArchRplus", mustWork = TRUE))


iqw_tpm_pl <- iqw_tpm_plots(sname = "sample1",
                            dir_path = tempdir(),
                            info_df = info_df,
                            iqw = TRUE,
                            tpm = TRUE,
                            cons = FALSE,
                            clusts = use_clusts,
                            txt_size = 14)
```

`order_clusters_iqw` *Order clusters by median or mean interquantile widths*

Description

Order clusters by median or mean interquantile widths

Usage

```
order_clusters_iqw(sname, clusts, info_df, order_by_median = TRUE)
```

Arguments

`sname`	The sample name
`clusts`	List of sequence ids in each cluster.
`info_df`	The data.frame with all tag clusters information. The following columns are expected in the data.frame:"chr", "start", "end", "width", "strand", "score", "nr_ctss", "dominant_ctss", "domTPM", "IQW", "tpm" and two additional columns based on qLow and qUp used.
`order_by_median`	Logical. Whether to order to clusters by their median (when TRUE) or mean (when FALSE) interquantile widths.

Value

The list of clusters ordered by their mean/median interquantile widths (shortest first).

Author(s)

Sarvesh Nikumbh

Examples

```
bed_fname <- system.file("extdata", "example_info_df.bed.gz",
         package = "seqArchRplus", mustWork = TRUE)

info_df <- read.delim(file = bed_fname,
         sep = "\t", header = TRUE,
         col.names = c("chr", "start", "end", "width",
                 "dominant_ctss", "domTPM",
                 "strand", "score", "nr_ctss",
                 "q_0.1", "q_0.9", "IQW", "tpm"))

use_clusts <- readRDS(system.file("extdata", "example_clust_info.rds",
         package = "seqArchRplus", mustWork = TRUE))


ordered_clusts <- seqArchRplus::order_clusters_iqw(
                                  sname = "sample1",
                                  clusts = use_clusts,
                                  info_df = info_df,
                                  order_by_median = TRUE)
```

`per_cluster_annotations`

per_cluster_annotations

Description

This function helps annotate the genomic regions specified in 'tc_gr' with features, namely, promoter-TSS (transcription start site), exons, 5'UTR, 3'UTR, introns and (distal) intergenic regions. This requires that the annotations are available as a TxDb object. The selected genomic regions can be specified as a single GenomicRanges object. These regions can be specified directly as a BED file (when available) or select specific regions from a larger set of regions based on some clustering.

When working with CAGE data, if the CAGEr package was used and the corresponding CAGEexp object is available, it can also be used – see 'cager_obj' argument.

Usage

```
per_cluster_annotations(
  sname = NULL,
  clusts = NULL,
  tc_gr = NULL,
  cager_obj = NULL,
  qLow = 0.1,
```

```
  qUp = 0.9,
  txdb_obj = NULL,
  tss_region = NULL,
  orgdb_obj = NULL,
  one_plot = TRUE,
  dir_path = NULL,
  txt_size = 12,
  use_suffix = NULL,
  use_prefix = "C",
  n_cores = 1
)
```

Arguments

sname	Sample name. Default is NULL. This is a required argument if the CAGEexp object is provided. See 'cager_obj' argument
clusts	List of sequence IDs in each cluster. This can be NULL only when a BED file is passed to the argument 'tc_gr'
tc_gr	Tag clusters as GRanges or a BED file (specify filename with path). If 'cager_obj' is not provided (i.e., it is NULL), this argument is required. It will be ignored only if 'cager_obj' is provided. Default is NULL
cager_obj	A CAGEexp object obtained from the CAGEr package, if and when CAGEr was used to process the raw CAGE data
qLow, qUp	The interquantile boundaries to be considered for obtaining tag clusters from the CAGEexp object. See tagClusters
txdb_obj	A TxDb object storing transcript metadata
tss_region	For ChIPseeker
orgdb_obj	Organism-level annotation package
one_plot	Logical. Default is TRUE. If set to FALSE the barplots of annotations per cluster are returned as a list, else all are condensed into single plot
dir_path	Specify the /path/to/directory to store results
txt_size	Adjust text size for the plots
use_suffix, use_prefix	Character. Specify any suffix and/or prefix you wish to add to the cluster labels
n_cores	Numeric. If you wish to parallelize annotation of peaks, specify the number of cores. Default is 1 (serial)

Details

When annotations for only selected clusters are required, alter the 'clusts' argument to specify only those selected clusters. Because the 'clusts' list holds the IDs of sequences belonging to each cluster, the corresponding records are selected from the 'tc_gr' GRanges object. This approach requires that sequence IDs in 'clusts' are directly associated with the ranges in 'tc_gr'. Also, see examples.

Value

When 'one_plot = TRUE', a single plot where annotation barplots for each cluster are put together (ordered as per the clusters in 'clusts'). Otherwise, a list of annotation barplots is returned (again ordered by the clusters in 'clusts').

Author(s)

Sarvesh Nikumbh

Examples

```
## Need the TxDb object to run these examples
if(require("TxDb.Dmelanogaster.UCSC.dm6.ensGene")){

library(GenomicRanges)
library(TxDb.Dmelanogaster.UCSC.dm6.ensGene)
library(ChIPseeker)

bed_fname <- system.file("extdata", "example_info_df.bed.gz",
         package = "seqArchRplus", mustWork = TRUE)

info_df <- read.delim(file = bed_fname,
         sep = "\t", header = TRUE)

tc_gr_from_df <- GenomicRanges::makeGRangesFromDataFrame(info_df,
                                                  keep.extra.columns = TRUE)

tc_gr <- readRDS(system.file("extdata", "example_tc_gr.rds",
         package = "seqArchRplus", mustWork = TRUE))

use_clusts <- readRDS(system.file("extdata", "example_clust_info.rds",
         package = "seqArchRplus", mustWork = TRUE))

tdir <- tempdir()

# Get annotations for all clusters in use_clusts
annotations_pl <- per_cluster_annotations(sname = "sample1",
                         clusts = use_clusts,
                         tc_gr = tc_gr_from_df,
                         txdb_obj = TxDb.Dmelanogaster.UCSC.dm6.ensGene,
                         one_plot = FALSE,
                         dir_path = tdir,
                         tss_region = c(-500,100))

# Get annotations for selected clusters in use_clusts
# -- First two clusters
selected_clusts <- lapply(seq(2), function(x) use_clusts[[x]])
# OR
# -- Mixed set of clusters, say 1 and 3 out of total 3
selected_clusts <- lapply(c(1,3), function(x) use_clusts[[x]])
#
annotations_pl <- per_cluster_annotations(sname = "sample1",
```

```
                          clusts = selected_clusts,
                          tc_gr = tc_gr,
                          txdb_obj = TxDb.Dmelanogaster.UCSC.dm6.ensGene,
                          one_plot = FALSE,
                          dir_path = tdir,
                          tss_region = c(-500,100))

# Alternatively, you can also directly specify a BED file to the `tc_gr`
# argument. This is useful when one may not have access to the CAGEexp
# object, but only clusters' information is available in a BED file.
#

annotations_pl <- per_cluster_annotations(sname = "sample1",
                          clusts = NULL,
                          tc_gr = bed_fname,
                          txdb_obj = TxDb.Dmelanogaster.UCSC.dm6.ensGene,
                          one_plot = FALSE,
                          dir_path = tdir,
                          tss_region = c(-500,100))
}
```

`per_cluster_go_term_enrichments` *Perform per cluster GO term enrichment analysis*

Description

This function helps identify GO terms enriched per cluster. This requires that the annotations are available as a TxDb object. The selected genomic regions can be specified as a single GenomicRanges object. These regions can be specified directly as a BED file (when available) or select specific regions from a larger set of regions based on some clustering.

Usage

```
per_cluster_go_term_enrichments(
  sname = NULL,
  clusts = NULL,
  tc_gr = NULL,
  cager_obj = NULL,
  qLow = 0.1,
  qUp = 0.9,
  txdb_obj = NULL,
  tss_region = NULL,
  orgdb_obj = NULL,
  use_keytype = "ENTREZID",
  one_file = TRUE,
  bar_or_dot = "dot",
  dir_path = NULL,
```

```
  txt_size = 12,
  n_cores = 1
)
```

Arguments

sname	Sample name. Default is NULL. This is a required argument if the CAGEexp object is provided. See ‘cager_obj‘ argument
clusts	List of sequence IDs in each cluster. This can be NULL only when a BED file is passed to the argument ‘tc_gr‘
tc_gr	Tag clusters as GRanges or a BED file (specify filename with path). If ‘cager_obj‘ is not provided (i.e., it is NULL), this argument is required. It will be ignored only if ‘cager_obj‘ is provided. Default is NULL
cager_obj	A CAGEexp object obtained from the CAGEr package, if and when CAGEr was used to process the raw CAGE data
qLow, qUp	The interquantile boundaries to be considered for obtaining tag clusters from the CAGEexp object. See tagClusters
txdb_obj	A TxDb object storing transcript metadata
tss_region	For ChIPseeker
orgdb_obj	Organism-level annotation package
use_keytype	Either of "ENTREZID" or "ENSEMBL". Required for use with enrichGO
one_file	Logical. Default is TRUE. If set to FALSE the plots of GO terms enriched per cluster are returned as a list, else all are written to a single file as separate pages
bar_or_dot	Specify "dot" for dotplot (default), or "bar" for barplot
dir_path	Specify the /path/to/directory to store results
txt_size	Adjust text size for the plots
n_cores	For future use

Details

Both ‘txdb_obj‘ and ‘orgdb_obj‘ are required.

Per cluster, the enriched GO terms are visualized as a dot plot which shows the enriched terms on the vertical axis and the ratio of genes that are enriched for the given GO term vs. the total genes in the cluster.

Value

The list of dot plots showing GO term enrichments per cluster. This is a list of ggplot2 plots.

When ‘one_file‘ is set to TRUE (default), in addition to returning the list of dot plots, these plots are also written to disk as a PDF, with one plot per page.

Author(s)

Sarvesh Nikumbh

Examples

```
library(GenomicRanges)
library(TxDb.Dmelanogaster.UCSC.dm6.ensGene)
library(ChIPseeker) ## important to load this package
library(org.Dm.eg.db)

bed_fname <- system.file("extdata", "example_info_df.bed.gz",
         package = "seqArchRplus", mustWork = TRUE)

info_df <- read.delim(file = bed_fname, sep = "\t", header = TRUE)

tc_gr_from_df <- GenomicRanges::makeGRangesFromDataFrame(info_df,
                                                  keep.extra.columns = TRUE)

tc_gr <- readRDS(system.file("extdata", "example_tc_gr.rds",
         package = "seqArchRplus", mustWork = TRUE))

use_clusts <- readRDS(system.file("extdata", "example_clust_info.rds",
         package = "seqArchRplus", mustWork = TRUE))

tdir <- tempdir()

# Get GO term enrichments for all clusters in use_clusts
go_pl <- per_cluster_go_term_enrichments(sname = "sample1",
                         clusts = use_clusts[1:2],
                         tc_gr = tc_gr_from_df,
                         txdb_obj = TxDb.Dmelanogaster.UCSC.dm6.ensGene,
                         dir_path = tdir,
                         one_file = FALSE,
                         tss_region = c(-500,100),
                         orgdb_obj = "org.Dm.eg.db")
```

`per_cluster_seqlogos` *Plot per cluster sequence logos*

Description

Plot per cluster sequence logos

Usage

```
per_cluster_seqlogos(
  sname,
  seqs = NULL,
  clusts,
  pos_lab = NULL,
```

```
  bits_yax = "max",
  strand_sep = FALSE,
  one_plot = TRUE,
  info_df = NULL,
  txt_size = 12,
  save_png = FALSE,
  dir_path = NULL
)
```

Arguments

sname	The sample name
seqs	The raw sequences as a DNAStringSet. These are also available as part of the seqArchR result object.
clusts	List of sequence ids in each cluster.
pos_lab	The position labels
bits_yax	The yaxis limits. Possible values are "full", and "auto". See argument in plot_ggseqlogo_of_seqs for more details
strand_sep	Logical. Whether sequences are to be separated by strand.
one_plot	Logical. Whether all sequence logos should be combined into one grid (with ncol = 1)?
info_df	The information data.frame
txt_size	The font size for text
save_png	Logical. Set TRUE if you would like to save the architectures sequence logos as PNG files
dir_path	The /path/to/the/directory where plot will be saved. Default is NULL

Details

Plots the sequence logos of all clusters

Value

If ‘one_plot‘ is TRUE, one plot as a grid of all sequence logos is returned.

If ‘one_plot‘ is FALSE, a set of ggplot2-based sequence logos for all clusters is saved to disk with the default filename ‘Architectures_0-max.pdf‘ in the location provided by ‘dir_path‘. This is a multi-page PDF document with the sequence logo for each cluster on a separate page. Also, the list of plots is returned.

With ‘strand_sep = TRUE‘, the ‘one_plot‘ option is not available. In other words, only a list of plots is returned. When ‘dir_path‘ is not NULL, the default filename used is ‘Architectures_0-max_strand_separated.pdf‘.

Author(s)

Sarvesh Nikumbh

Examples

```
library(Biostrings)
raw_seqs <- Biostrings::readDNAStringSet(
                          filepath = system.file("extdata",
                            "example_promoters45.fa.gz",
                            package = "seqArchRplus",
                            mustWork = TRUE)
                        )

use_clusts <- readRDS(system.file("extdata", "example_clust_info.rds",
         package = "seqArchRplus", mustWork = TRUE))

seqlogo_pl <- per_cluster_seqlogos(sname = "sample1",
                                   seqs = raw_seqs,
                                   clusts = use_clusts,
                                   pos_lab = -45:45,
                                   bits_yax = "auto",
                                   strand_sep = FALSE,
                                   one_plot = TRUE,
                                   dir_path = tempdir(),
                                   txt_size = 14)
```

---

per_cluster_strand_dist

Visualize as barplots how promoters in each cluster are distributed on different chromosomes and strands

---

Description

Visualize as barplots how promoters in each cluster are distributed on different chromosomes and strands

Usage

```
per_cluster_strand_dist(
  sname,
  clusts,
  info_df,
  dir_path = NULL,
  colrs = c("#FB8072", "#80B1D3"),
  txt_size = 14,
  fwidth = 20,
  fheight = 3
)
```

Arguments

sname	The sample name
clusts	List of sequence ids in each cluster.
info_df	The information data.frame
dir_path	Specify the path to the directory on disk where plots will be saved
colrs	Specify colors used for two strands. By default, the fourth and fifth color from the palette 'Set3' are used
txt_size	Size of the text in the plots (includes plot title, y-axis title, axis texts, legend title and text)
fwidth	Width of the individual plots in file
fheight	Height of the plots in file

Value

A list of plots showing the per cluster division of promoters on chromosomes and strands. These plots are also written to disk in file named "Per_cluster_strand_distributions.pdf"

Author(s)

Sarvesh Nikumbh

Examples

```
library(RColorBrewer)
bed_fname <- system.file("extdata", "example_info_df.bed.gz",
         package = "seqArchRplus", mustWork = TRUE)

info_df <- read.delim(file = bed_fname,
         sep = "\t", header = TRUE,
         col.names = c("chr", "start", "end", "width",
                 "strand", "score", "nr_ctss",
                 "dominant_ctss", "domTPM",
                 "q_0.1", "q_0.9", "IQW", "tpm"))

use_clusts <- readRDS(system.file("extdata", "example_clust_info.rds",
         package = "seqArchRplus", mustWork = TRUE))

pair_colrs <- RColorBrewer::brewer.pal(n = 5, name = "Set3")[4:5]
per_cl_strand_pl <- per_cluster_strand_dist(sname = "sample1",
                                            clusts = use_clusts,
                                            info_df = info_df,
                                            dir_path = tempdir(),
                                            colrs = pair_colrs
                                            )
```

`plot_motif_heatmaps` *Plot heatmaps of motifs occurring in seqArchR clusters*

Description

Plot heatmaps of motifs occurring in seqArchR clusters

Usage

```
plot_motif_heatmaps(
  sname,
  seqs,
  flanks = c(50),
  clusts,
  use_colors = NULL,
  motifs,
  dir_path,
  fheight = 500,
  fwidth = 500,
  funits = "px",
  n_cores = 1,
  res = 150,
  dev = "png"
)
```

Arguments

`sname`	The sample name
`seqs`	The sequences as a DNAStringSet object
`flanks`	Flank size. The same flank is used upstream and downstream. A vector of values is also accepted when more than one flanks should be visualized.
`clusts`	List of sequence Ids in each cluster.
`use_colors`	Specify colors to use
`motifs`	A vector of motifs to be visualized in the sequence. This can be any words formed by the IUPAC code. For example, TATAA, CG, WW, SS etc.
`dir_path`	The path to the directory
`fheight, fwidth, funits`	Height and width of the image file, and the units in which they are specified. Units are inches for PDF and SVG, and pixels for PNG and TIFF devices
`n_cores`	Numeric. If you wish to parallelize, specify the number of cores. Default is 1 (serial)
`res`	Numeric. The resolution in ppi to be used for producing PNG or TIFF files. Default is 150
`dev`	Specify the device type. One of 'png', 'tiff', 'pdf', or 'svg'. Default is 'png'

Value

Nothing. Images are written to disk using the provided filenames.

Author(s)

Sarvesh Nikumbh

Examples

```
library(Biostrings)
raw_seqs <- Biostrings::readDNAStringSet(
                          filepath = system.file("extdata",
                            "example_promoters200.fa.gz",
                            package = "seqArchRplus",
                            mustWork = TRUE)
                        )

use_clusts <- readRDS(system.file("extdata", "example_clust_info.rds",
         package = "seqArchRplus", mustWork = TRUE))

plot_motif_heatmaps(sname = "sample1", seqs = raw_seqs,
                    flanks = c(10, 20, 50),
                    clusts = use_clusts,
                    motifs = c("WW", "SS", "TATAA", "CG", "Y"),
                    dir_path = tempdir(),
                    fheight = 800, fwidth = 2400)
```

---

plot_motif_heatmaps2 *Plot heatmaps of motifs occurring in seqArchR clusters*

Description

This function uses the seqPattern package. It is recommended to use this function rather than 'plot_motif_heatmaps'

Usage

```
plot_motif_heatmaps2(
  sname,
  seqs,
  flanks = c(50),
  clusts,
  use_colors = NULL,
  motifs,
  dir_path,
  fheight = 1.5 * fwidth,
  fwidth = 2000,
```

```
  hm_scale_factor = 0.75,
  n_cores = 1,
  type = c("png", "jpg"),
  ...
)
```

Arguments

sname	The sample name
seqs	The sequences as a DNAStringSet object
flanks	Flank size. The same flank is used upstream and downstream. A vector of values is also accepted when more than oone flanks should be visualized
clusts	List of sequence Ids in each cluster
use_colors	Specify colors to use
motifs	A vector of motifs to be visualized in the sequence. This can be any words formed by the IUPAC code. For example, TATAA, CG, WW, SS etc.
dir_path	The path to the directory
fheight, fwidth	Height and width of the individual heatmap plots in pixels
hm_scale_factor	Factor to scale the color scale bar w.r.t. the heatmaps. Values in (0,1]. Useful when specifying more than two motifs at once. Note that combining/specifying more than 3 or 4 motifs in one call to the function may result in a sub-optimally combined plot
n_cores	Numeric. If you wish to parallelize, specify the number of cores. Default is 1 (serial)
type	Specify either of "png" or "jpg" to obtain PNG or JPEG files as output
...	Additional arguments passed to plotPatternDensityMap

Value

Nothing. Images are written to disk using the provided filenames. In addition, two legends are printed to separate files: the color legend and the clustering legend, which can be then combined with the heatmaps. The heatmaps themselves have the cluster numbers marked on the vertical axis.

Author(s)

Sarvesh Nikumbh

Examples

```
library(Biostrings)
raw_seqs <- Biostrings::readDNAStringSet(
                          filepath = system.file("extdata",
                            "example_promoters200.fa.gz",
                            package = "seqArchRplus",
                            mustWork = TRUE)
                        )
```

```
use_clusts <- readRDS(system.file("extdata", "example_clust_info.rds",
         package = "seqArchRplus", mustWork = TRUE))

plot_motif_heatmaps2(sname = "sample1", seqs = raw_seqs,
                     flanks = c(50),
                     clusts = use_clusts,
                     motifs = c("WW", "SS"),
                     dir_path = tempdir())
```

`seqArchRplus` *seqArchRplus: A package for downstream analyses of promoter sequence architectures/clusters identified by seqArchR*

Description

This package facilitates various downstream analyses and visualizations for clusters/architectures identified in promoter sequences. If a seqArchR result object is available, and in addition, the CAGEr object or information on the tag clusters, this package can be used for the following:

Details

- Order the sequence architectures by the interquantile widths (IQWs) of the tag clusters they originate from. See 'CAGEr' vignette for more information. `order_clusters_iqw`

- Visualize distributions of IQW, TPM and conservation scores per cluster `iqw_tpm_plots`

- Visualize the percentage annotations of promoter sequence per cluster `per_cluster_annotations`

- Visualize heatmaps of occurrences of motifs (as words) in sequences `plot_motif_heatmaps` and `plot_motif_heatmaps2`

- Visualize the above plots as (publication ready) combined panels viewable for different samples as HTML reports

- Following per cluster visualizations:

- sequence logos of architectures (including strand-separated ones) `per_cluster_seqlogos`

- distributions of promoters on different chromosomes/strands `per_cluster_strand_dist`

- GO term enrichments `per_cluster_go_term_enrichments`

- Produce BED track files of seqArchR clusters for visulization in a genome browser or IGV `write_seqArchR_cluster_track_bed`

- Curate seqArchR clusters `curate_clusters`

- (future) Generate HTML reports that help you navigate this wealth of information with ease, and enable insights and hypotheses generation

Functions for data preparation and manipulation

- `prepare_data_from_FASTA`
- `get_one_hot_encoded_seqs`

Functions for visualizations

- plot_arch_for_clusters
- plot_ggseqlogo_of_seqs
- viz_bas_vec
- viz_seqs_acgt_mat
- viz_pwm

Author(s)

Sarvesh Nikumbh

---

seqs_acgt_image *Visualize all sequences as an image*

Description

Visualize all sequences as an image

Usage

```
seqs_acgt_image(
  sname,
  seqs,
  seqs_ord,
  pos_lab,
  xt_freq = 5,
  yt_freq = 500,
  f_height = 1200,
  f_width = 600,
  dir_path = NULL
)
```

Arguments

sname	The sample name
seqs	The sequences
seqs_ord	The order of sequences
pos_lab	The position labels
xt_freq	The frequency of xticks
yt_freq	The frequency of yticks
f_height, f_width	The height and width for the PNG image.
dir_path	Specify the /path/to/directory to store results. Default is NULL.

Value

Nothing. PNG images are written to disk at the provided 'dir_path' using the filename '<sample_name>_ClusteringImage.png'.

Author(s)

Sarvesh Nikumbh

Examples

```
library(Biostrings)
raw_seqs <- Biostrings::readDNAStringSet(
                          filepath = system.file("extdata",
                            "example_promoters45.fa.gz",
                            package = "seqArchRplus",
                            mustWork = TRUE)
                        )

use_clusts <- readRDS(system.file("extdata", "example_clust_info.rds",
         package = "seqArchRplus", mustWork = TRUE))

seqs_acgt_image(sname = "sample1",
                seqs = raw_seqs,
                seqs_ord = unlist(use_clusts),
                pos_lab = -45:45,
                dir_path = tempdir())
```

---

write_seqArchR_cluster_track_bed

Write seqArchR cluster information in BED files viewable as browser tracks

---

Description

Writes the seqArchR clusters as BED tracks for viewing in IGV or any genome browser

Usage

```
write_seqArchR_cluster_track_bed(
  sname,
  clusts = NULL,
  info_df,
  use_q_bound = TRUE,
  use_as_names = NULL,
  one_zip_all = FALSE,
  org_name = NULL,
  dir_path,
```

```
  include_in_report = FALSE,
  strand_sep = FALSE
)
```

Arguments

sname	Sample name
clusts	List of sequence ids in each cluster
info_df	The data.frame holding information to be written to the BED file. Expected columns are "chr", "start", "end", "names", "strand", "domTPM", and "dominant_ctss". Out of these, only "dominant_ctss" column is optional; when present this is visualised as thickStart and thickEnd. The "domTPM" column is treated as scores for visualizing by default. One can also visualize PhastCons score
use_q_bound	Logical. Write the lower and upper quantiles as tag cluster boundaries. Default is TRUE
use_as_names	Specify the column name from info_df which you would like to use as names for display with the track. This is NULL by default, and the sequence/tag cluster IDs are used as names.
one_zip_all	Logical. Specify TRUE when the facility to download BED files for all clusters with one click is desired, otherwise FALSE. This is relevant only when include_in_report is TRUE. Default is FALSE
org_name	Organism name
dir_path	The /path/to/the/directory where a special folder named 'Cluster_BED_tracks' (by default) is created and all BED files are written inside it. This is a required argument, and cannot be NULL
include_in_report	Logical. Specify TRUE when this function is invoked to write BED files to disk *and* provide downloadable links in the HTML report. The corresponding chunk in the Rmarkdown report should set the parameter 'result='asis''. By setting this to FALSE, BED files are written to disk, but no downloadable links are provided. Note: This should be TRUE, when 'one_zip_all' argument is set to TRUE (see below). Requires the package 'xfun'
strand_sep	Logical. Specify TRUE if records for each strand are to be written in separate BED files

Details

Note on links in HTML: For providing downloadable links in the HTML report, the complete BED files are encoded into base64 strings and embedded with the HTML report itself. This considerably increases the size of the HTML file, and can slow down loading of the HTML file in your browser.

Note on BED files: The output BED files have selected columns provided in the 'info_df'. These are "chr", "start", "end", "name", "score" (see more info below), "strand", "dominant_ctss". By default, the sequence/tag cluster IDs are used as names. If `use_as_names` is specified, information from that column in the 'info_df' is used as "name".

If conservation score (e.g., PhastCons) is available, it is used as the score, otherwise the TPM value of the dominant CTSS (domTPM) is used. The final two columns (when dominantCTSS

column is present), are the 'thickStart' and 'thickEnd' values corresponding to the BED format. The 'thickEnd' column is the dominant_ctss position.

Importantly, the lower and upper quantile boundaries are used as the start and end coordinates of the cluster when 'use_q_bound' is set to TRUE (the default).

Value

When `include_in_report = FALSE`, the cluster information is written to disk as BED track files that can be viewed in the genome browser or IGV. Otherwise, a str object holding HTML text is returned that can be included in the report as downloadable links for each cluster BED file (use `cat` function). When `one_zip_all = TRUE`, a link to download all files zipped into one is also provided to enable convenience.

Author(s)

Sarvesh Nikumbh

Examples

```
bed_fname <- system.file("extdata", "example_info_df.bed.gz",
         package = "seqArchRplus", mustWork = TRUE)

info_df <- read.delim(file = bed_fname,
         sep = "\t", header = TRUE)

use_clusts <- readRDS(system.file("extdata", "example_clust_info.rds",
         package = "seqArchRplus", mustWork = TRUE))


# Write seqArchR clusters of promoters/CAGE tag clusters as BED track files
# All possible variations are enlisted here

# Using quantiles information as the tag cluster boundaries
# via arg `use_q_bound = TRUE` and using custom names for each.
# Create a new/custom column, and specify the new column name for argument
# `use_as_names`. Notice that any custom names can be obtained by this
# approach.

info_df$use_names <- paste(rownames(info_df), info_df$domTPM, sep = "_")

write_seqArchR_cluster_track_bed(sname = "sample1",
                                 clusts = use_clusts,
                                 info_df = info_df,
                                 use_q_bound = FALSE,
                                 use_as_names = "use_names",
                                 dir_path = tempdir()
                                 )

# Generating textual output that can be included in HTML reports.
# This requires package xfun.

cat_str <- write_seqArchR_cluster_track_bed(sname = "sample1",
```

```
clusts = use_clusts,
info_df = info_df,
use_q_bound = FALSE,
use_as_names = "use_names",
dir_path = tempdir(),
include_in_report = TRUE,
one_zip_all = TRUE
)
```

Index

clusterCTSS, *8*
collate_seqArchR_result, *3*
curate_clusters, 2, *30*

DNAStringSet, *14*, *24*

enrichGO, *22*

form_combined_panel, 6

generate_all_plots, 7
generate_html_report, 10
get_one_hot_encoded_seqs, *30*
GRanges, *8*, *14*, *19*, *22*

handle_tc_from_cage, 13

iqw_tpm_plots, *6*, 15, *30*

order_clusters_iqw, 17, *30*

per_cluster_annotations, *6*, 18, *30*
per_cluster_go_term_enrichments, 21, *30*
per_cluster_seqlogos, *6*, 23, *30*
per_cluster_strand_dist, 25, *30*
plot_arch_for_clusters, *31*
plot_ggseqlogo_of_seqs, *24*, *31*
plot_motif_heatmaps, 27, *30*
plot_motif_heatmaps2, 28, *30*
plotPatternDensityMap, *29*
prepare_data_from_FASTA, *30*

seqArchRplus, 30
seqs_acgt_image, 31

tagClusters, *14*, *19*, *22*

viz_bas_vec, *31*
viz_pwm, *31*
viz_seqs_acgt_mat, *31*

write_seqArchR_cluster_track_bed, *30*,
        32