Package 'brainflowprobes'

November 22, 2023

Type Package

Title Plots and annotation for choosing BrainFlow target probe
sequence

Version 1.16.0

Date 2021-08-05

Description Use these functions to characterize genomic regions for
BrainFlow target probe design.

License Artistic-2.0

Encoding UTF-8

LazyData true

Depends R (>= 3.6.0)

Imports Biostrings (>= 2.52.0), BSgenome.Hsapiens.UCSC.hg19 (>=
1.4.0), bumphunter (>= 1.26.0), cowplot (>= 1.0.0), derfinder
(>= 1.18.1), derfinderPlot (>= 1.18.1), GenomicRanges (>=
1.36.0), ggplot2 (>= 3.1.1), RColorBrewer (>= 1.1), utils,
grDevices, GenomicState (>= 0.99.7)

RoxygenNote 7.2.3

Suggests BiocStyle, knitr, RefManageR, rmarkdown, sessioninfo,
testthat (>= 2.1.0), covr

VignetteBuilder knitr

URL https://github.com/LieberInstitute/brainflowprobes

BugReports https://support.bioconductor.org/t/brainflowprobes

biocViews Coverage, Visualization, ExperimentalDesign,
Transcriptomics, FlowCytometry, GeneTarget

Roxygen list(markdown = TRUE)

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

git_branch RELEASE_3_18

git_last_commit d92e0fe

git_last_commit_date 2023-10-24

1

Author Amanda Price [aut, cre] (<<https://orcid.org/0000-0001-7352-3732>>),
Leonardo Collado-Torres [ctb] (<<https://orcid.org/0000-0003-2140-308X>>)

Maintainer Amanda Price <amanda.joy.price@gmail.com>

R topics documented:

---

brainflowprobes-package

brainflowprobes: Plots and annotation for choosing BrainFlow target
probe sequence

Description

Use these functions to characterize genomic regions for BrainFlow target probe design.

Author(s)

Maintainer: Amanda Price <amanda.joy.price@gmail.com> ([ORCID](https://orcid.org))

Other contributors:

- Leonardo Collado-Torres <lcolladotor@gmail.com> ([ORCID](https://orcid.org)) [contributor]

See Also

Useful links:

- <https://github.com/LieberInstitute/brainflowprobes>
- Report bugs at <https://support.bioconductor.org/t/brainflowprobes>

---

brainflowprobes_cov	*Extract coverage data for a set of regions*

Description

This function extracts the data from the BigWig coverage files that is then used by four_panels. This function can take a while to run depending on your internet connection. Furthermore, this function relies on functionality in the rtracklayer package for reading BigWig files which does not work in Windows machines. The data extracted by this function is also used by plot_coverage.

Usage

```
brainflowprobes_cov(REGION, PD = brainflowprobes::pd, VERBOSE = TRUE)
```

Arguments

REGION   Either a single hg19 genomic sequence including the chromosome, start, end, and optionally strand separated by colons (e.g., 'chr20:10199446-10288068:+'), or a string of sequences. Must be character. Chromosome must be proceeded by 'chr'.

PD   A list of data.frames with the sumMapped and files columns. Defaults to the data included in this package.

VERBOSE   A logical value indicating whether to print updates from the process of loading the data from the BigWig files.

Value

A list of region coverage coverage data.frame lists used by four_panels and plot_coverage. That is, a list with one element per dataset in pd (so four: Sep, Deg, Cell, Sort). Each element of the output list is a list with one data.frame per input region. In the case of four_panels_example_cov there was only one input region hence each region coverage data.frame list has one element. A region coverage data.frame has one column per sample and one row per genome base-pair for the given region and dataset.

Author(s)

Leonardo Collado-Torres

Examples

```
## This function loads data from BigWig files using the rtracklayer package.
## This functionality is not supported on Windows OS machines!
if (.Platform$OS.type != "windows") {
    ## How long this takes to run will depend on your internet connection.
    example_cov <- brainflowprobes_cov("chr20:10286777-10288069:+",
        PD = lapply(brainflowprobes::pd, head, n = 2)
    )
```

```
    ## Output examination:
    # A list with one element per element in brainflowprobes::pd
    stopifnot(is.list(example_cov))
    stopifnot(identical(
        names(example_cov),
        names(brainflowprobes::pd)
    ))

    # For each dataset, brainflowprobes_cov() returns a list of region
    # coverage data.frames. In this example, there was a single input region.
    stopifnot(all(
        sapply(example_cov, length) ==
            length(
                GenomicRanges::GRanges("chr20:10286777-10288069:+")
            )
    ))

    # Then each data.frame itself has 1 row per genome base-pair in the region
    stopifnot(
        all(
            sapply(example_cov, function(x) {
                nrow(x[[1]])
            }) ==
                GenomicRanges::width(
                    GenomicRanges::GRanges("chr20:10286777-10288069:+")
                )
        )
    )

    # and one column per sample in the dataset unless you subsetted the data
    # like we did earlier when creating "example_cov".
    stopifnot(identical(
        sapply(four_panels_example_cov, function(x) {
            ncol(x[[1]])
        }),
        sapply(pd, nrow)
    ))
}

## This is how the example data included in the package was made:
## Not run:
## This can take about 10 minutes to run!
four_panels_example_cov <- brainflowprobes_cov("chr20:10286777-10288069:+")

## End(Not run)


## If you are interested, you could download all the BigWig files
## in the \code{brainflowprobes::pd} list of data.frames from the
## \code{files} column to your disk. Doing so will greatly increase the
## speed for \code{brainflowprobes_cov} and the functions that depend on
## this data. Then edit \code{brainflowprobes::pd} \code{files} to point to
## your local files.
```

```
## Web location of BigWig files
lapply(brainflowprobes::pd, function(x) head(x$files))
```

---

check_pdf   *Check the PDF file*

Description

This function checks that the PDF file does not exist to avoid overwriting a plot the user has previously made.

Usage

```
check_pdf(PDF = "four_panels.pdf", OUTDIR = tempdir())
```

Arguments

PDF	The name of the PDF file. Defaults to four_panels.pdf.
OUTDIR	The default directory where PDF will be saved to.

Details

This is an utility function used by [four_panels](#) and [plot_coverage](#).

Value

The path to the PDF file if the file doesn't exist. It appends the pdf file extension if it was absent from PDF.

Author(s)

Leonardo Collado-Torres

Examples

```
## Choose a random PDF file
PDF <- paste0("test_", stats::runif(1, max = 1e10))

## Initially this works because the output PDF does not exist.
PDF <- check_pdf(PDF)
## It also adds the PDF extension if the user didn't supply it.
PDF

## Create a dummy PDF file
pdf(file = PDF)
plot(1, 1)
dev.off()
```

```
## Now it doesn't work since the PDF file already exists.
testthat::expect_error(
    check_pdf(basename(PDF)),
    "already exists! Rename or erase it"
)
```

---

four_panels   *Plot expression coverage in four datasets for candidate probe se-*
                                   quences.

Description

four_panels creates four plots for each candidate probe sequence. The first plot (Separation)
shows the adjusted read coverage in cytosolic and nuclear RNA from human postmortem cortex.
The second plot (Degradation) shows coverage in human cortical samples exposed to room tem-
perature for 0-60 minutes. The third plot (Sorted) shows RNA coverage in nuclei that had been
sorted based on reactivity to NeuN-antibody, a neuronal marker. NeuN+ samples are enriched for
neurons, and NeuN- samples are enriched for non-neurons. The fourth plot (Single Cells) shows
the expression coverage in single cells isolated from human temporal lobe.

Usage

```
four_panels(
  REGION,
  PDF = "four_panels.pdf",
  OUTDIR = tempdir(),
  JUNCTIONS = FALSE,
  COVERAGE = NULL,
  CODING_ONLY = FALSE,
  VERBOSE = TRUE
)
```

Arguments

REGION   Either a single hg19 genomic sequence including the chromosome, start, end,
                    and optionally strand separated by colons (e.g., 'chr20:10199446-10288068:+'),
                    or a string of sequences. Must be character. Chromosome must be proceeded by
                    'chr'.

PDF   The name of the PDF file. Defaults to four_panels.pdf.

OUTDIR   The default directory where PDF will be saved to.

JUNCTIONS   A logical value indicating if the candidate probe sequence spans splice junctions
                    (Default=FALSE).

COVERAGE   The output of brainflowprobes_cov for the input REGION. Defaults to NULL but
                    it can be pre-computed and saved separately since this is the step that takes the
                    longest to run. Also, this is the only step that depends on rtracklayer's function-
                    ality for reading BigWig files which does not run on Windows OS. So it could
                    be run on a non-Windows machine, saved, and then shared with Windows users.

CODING_ONLY   A logical vector of length 1 specifying whether to subset the Annotated Genes to only the coding genes. That is, whether to subset the genes by whether they have a non-NA CSS value. The Annotated Genes are downloaded with GenomicState::GenomicStateHub().

VERBOSE   A logical value indicating whether to print updates from the process of loading the data from the BigWig files.

Value

four_panels() first annotates the input candidate probe sequence(s) in REGION using bumphunter::matchGenes(), and then cuts the expression coverage for each sequence from each sample in four different datasets (see the BrainFlow publication for references) using derfinder::getRegionCoverage(). The coverage is normalized to the total mapped reads per sample and kilobase width of each probe region before log2 transformation. The four plots are labeled by the dataset and the plots are topped by the sequence coordinates, sequence width, and the name of the nearest gene.

A good candidate probe sequence will have several characteristics. In the Separation data, the sequence should be relatively highly expressed in nuclear RNA, at least in your age of interest. The sequence should also show stable expression over the 60 minutes of room temperature exposure in the Degradation data. The sequence should also be expressed in the appropriate NeuN fraction (depending on cell type specificity) in the Sorted dataset, and also be expressed in the right cell type in the Single Cell dataset.

four_panels() saves the results as four_panels.pdf in a temporary directory unless otherwise specified with OUTDIR.
if(JUNCTIONS), this means that the candidate probe sequence spans splice junctions. In this case, the character vector of regions should represent the coordinates of each exon spanned in the sequence. if(JUNCTIONS), four_panels() will sum the coverage of each exon and plot that value for each dataset instead of creating an independent set of plots for each exon. This is a way to avoid deflating coverage by including lowly-expressed intron coverage in the plots.

Author(s)

Amanda J Price

Examples

```
## Here we use the pre-saved example coverage data such that this example
## will run fast!
four_panels("chr20:10286777-10288069:+",
    COVERAGE = four_panels_example_cov
)
## Not run:
## Without using COVERAGE, this function reads BigWig files from the web
## using rtracklayer and this functionality is not supported on Windows
## machines.
if (.Platform$OS.type != "windows") {
    ## This example takes 10 minutes to run!
    four_panels("chr20:10286777-10288069:+")
}
```

```
## These examples will take several minutes to run depending on your
## internet connection
four_panels(c(
    "chr20:10286777-10288069:+",
    "chr18:74690788-74692427:-",
    "chr19:49932861-49933829:-"
))

PENK_exons <- c(
    "chr8:57353587-57354496:-",
    "chr8:57358375-57358515:-",
    "chr8:57358985-57359040:-",
    "chr8:57359128-57359292:-"
)

## General syntax
four_panels(PENK_exons,
    JUNCTIONS = TRUE,
    PDF = "PDF_file.pdf", OUTDIR = "/path/to/directory/"
)

four_panels("chr20:10286777-10288069:+",
    PDF = "PDF_file.pdf", OUTDIR = "/path/to/directory/"
)


## Explore the effect of changing CODING_ONLY
## Check how gene name changes in the title of the plot
## (everything else stays the same)
cov <- brainflowprobes_cov("chr10:135379301-135379311:+")
four_panels("chr10:135379301-135379311:+", COVERAGE = cov)
four_panels("chr10:135379301-135379311:+",
    COVERAGE = cov,
    PDF = "coding_only_four_panels", CODING_ONLY = TRUE
)

## End(Not run)
```

---

four_panels_example_cov

Example base-pair region coverage data

Description

A list of base-pair region coverage data.frame lists used for exemplifying the package functionality. This is the data extracted for the example region chr20:10286777-10288069:+ by looping through the phenotype tables stored in pd and using the getRegionCoverage function. This can be reproduced using the brainflowprobes_cov function in this package.

Format

A list with the base-pair coverage output from brainflowprobes_cov with the example region used throughout the package documentation ('chr20:10286777-10288069:+').

Sep base-pair coverage region data.frame list for pd$Sep,

Deg base-pair coverage region data.frame list for pd$Deg,

Cell base-pair coverage region data.frame list for pd$Cell,

Sort base-pair coverage region data.frame list for pd$Sort.

See Also

four_panels plot_coverage

Examples

```
if (FALSE) {
    ## Takes about 10 minutes to run!
    four_panels_example_cov <- brainflowprobes_cov(
        "chr20:10286777-10288069:+"
    )
}
```

---

get_nearest_annotation

Compute the nearest annotation to the annoated genes in brainflowprobes

---

Description

For a given set of genomic regions, this function computes the nearest annotation information using the Annotated Genes required by this package. The Annotated Genes are actually provided by GenomicState::GenomicStateHub().

Usage

```
get_nearest_annotation(gr, CODING_ONLY = FALSE)
```

Arguments

gr   A GenomicRanges::GRanges() object.

CODING_ONLY   A logical vector of length 1 specifying whether to subset the Annotated Genes to only the coding genes. That is, whether to subset the genes by whether they have a non-NA CSS value. The Annotated Genes are downloaded with GenomicState::GenomicStateHub().

Details

This is an utility function used by region_info, four_panels and plot_coverage.

Value

The `bumphunter::matchGenes()` output for the annotation information using the Annotated Genes for Gencode version 31 on hg19 coordinates (subset to only the coding elements if `CODING_ONLY` was set to `TRUE`).

Author(s)

Leonardo Collado-Torres

Examples

```
gr <- GenomicRanges::GRanges("chr10:135379301-135379311:+")

get_nearest_annotation(gr)
get_nearest_annotation(gr, CODING_ONLY = TRUE)
```

---

get_region_cov   *Check or compute the region coverage from the datasets in brainflow-*
                              probes

Description

This utility function checks the user-provided region data.frame coverage list (`COVERAGE`) or computes a new one using brainflowprobes_cov. This is used by four_panels and plot_coverage.

Usage

```
get_region_cov(
  REGION,
  COVERAGE = NULL,
  VERBOSE = TRUE,
  PD = brainflowprobes::pd
)
```

Arguments

REGION   Either a single hg19 genomic sequence including the chromosome, start, end, and optionally strand separated by colons (e.g., 'chr20:10199446-10288068:+'), or a string of sequences. Must be character. Chromosome must be proceeded by 'chr'.

COVERAGE   The output of brainflowprobes_cov for the input `REGION`. Defaults to `NULL` but it can be pre-computed and saved separately since this is the step that takes the longest to run. Also, this is the only step that depends on rtracklayer's functionality for reading BigWig files which does not run on Windows OS. So it could be run on a non-Windows machine, saved, and then shared with Windows users.

VERBOSE   A logical value indicating whether to print updates from the process of loading the data from the BigWig files.

PD A list of data.frames with the `summMapped` and `files` columns. Defaults to the data included in this package.

Value

If `COVERAGE` is provided and all checks pass, then this function returns `COVERAGE`. Otherwise, it computes a new region coverage data.frame list using brainflowprobes_cov.

Author(s)

Leonardo Collado-Torres

Examples

```
## If all checks pass, then it returns the COVERAGE
stopifnot(identical(
    get_region_cov(COVERAGE = four_panels_example_cov),
    four_panels_example_cov
))
```

---

pd *A list of phenotype data.frames*

Description

A list of four phenotype data frames used throughput the package.

Format

A list of four data.frames:

Sep phenotype information for samples from https://www.biorxiv.org/content/10.1101/567966v1; a data.frame with 23 rows and 15 columns. Column descriptions: SampleID is the sample name, Zone is the RNA fraction of the sample ("Nucleus" or "Cytosol"), Age, Sex, and Race list these demographic characteristics, Fetal categorizes each sample age as "Fetal" or "Adult", Library categorizes the RNAseq library preparation method as polyA selection ("polyA") or rRNA depletion ("RiboZero"), RIN_fraction is the RNA Integrity Number for each sample, summMapped is the number of mapped reads, Shortlabels, Label, col, and LabelFrac are columns of information for plotting, BigWig is the name of the BigWig file for each sample, and files lists the URL for the BigWig online.

Deg phenotype information for samples from https://www.pnas.org/content/114/27/7130; a data.frame with 40 rows and 16 columns. Column descriptions: DegradationTime is the number of minutes the brain tissue for each sample was left on the benchtop at room temperature before RNA was extracted, AgeDeath is the age of the donor at time of death, Dx lists whether the donor was diagnosed as schizophrenic or was a neurotypical control, Sex and Race list these demographic characteristics for each sample, pH lists the pH of the brain at collection, PMI is the postmortem interval between death and brain harvesting, BrNum is the donor number, RIN is the RNA Integrity Number for the RNA sample, LibraryProtocol

categorizes the RNAseq library preparation method as polyA selection ("polyA") or rRNA depletion ("Ribo"), SampleID is the ID for the sample, totalAssignedGene is the proportion of reads mapping to a gene body, sumMapped is the number of mapped reads, BigWig is the name of the BigWig file, SampleID_library is the Sample ID and library column values together, and files lists the URL for the BigWig online.

Cell phenotype information for samples from https://www.pnas.org/content/112/23/7285; a data.frame with 466 rows and 11 columns. Column descriptions: geo_accession is the accession number for each sample in the Gene Expression Omnibus, Age is the numeric age of each sample, AgeGroup categorizes each sample age as "prenatal" or "postnatal", sub_tissue lists whether the sample was from cortex or hippocampus, Cluster_color is the color for each sample for plotting, Cell_type is the cell identity assigned to each sample, RunName is the run name assigned to each sample in the Sequence Read Archive (SRA), SubjectID is the subject label, sumMapped is the number of mapped reads, BigWig is the name of the BigWig file, and files lists the URL for the BigWig online.

Sort phenotype information for samples from https://www.biorxiv.org/content/10.1101/428391v2; a data.frame with 12 rows and 12 columns. Column descriptions: Description categorizes the RNAseq library preparation method as polyA selection ("PolyA") or rRNA depletion ("Ribo"), SubjectID is the subject number, CellType lists whether the sample was labeled by NeuN antibody (neuronal, "NeuN_Plus") or not (non-neuronal, "NeuN_Minus"), SampleID combines the sample number, cell type and library of each sample in one column, BrNum is the donor ID, RIN is the RNA Integrity Number for each RNA sample, Age is the numeric age at death, sumMapped is the number of mapped reads, Label and col provide information for plotting, BigWig is the name of each BigWig file, and files lists the URL for the BigWig online.

See Also

four_panels brainflowprobes_cov https://github.com/LieberInstitute/brainflowprobes/blob/devel/data-raw/create_sysdata.R

Examples

```
##  pd <- list(Sep = pdSep, Deg = pdDeg, Cell = pdCell, Sort = pdSort)
```

---

plot_coverage	*Plot coverage in nuclear and cytoplasmic RNA for candidate probe sequences.*

Description

`plot_coverage` plots the coverage of nuclear and cytoplasmic RNA from adult and prenatal human prefrontal cortex across a candidate or series of candidate probe sequences for BrainFlow. If the sequence spans splice junctions, the plot will include the introns. A good candidate sequence will be highly and evenly expressed in nuclear RNA.

Usage

```
plot_coverage(
  REGION,
  PDF = "regionCoverage_fractionedData.pdf",
  OUTDIR = tempdir(),
  COVERAGE = NULL,
  CODING_ONLY = FALSE,
  VERBOSE = TRUE
)
```

Arguments

REGION   Either a single hg19 genomic sequence including the chromosome, start, end, and optionally strand separated by colons (e.g., 'chr20:10199446-10288068:+'), or a string of sequences. Must be character. Chromosome must be proceeded by 'chr'.

PDF   The name of the PDF file. Defaults to `regionCoverage_fractionedData.pdf`.

OUTDIR   The default directory where `PDF` will be saved to.

COVERAGE   The output of [brainflowprobes_cov](#) for the input `REGION`. Defaults to `NULL` but it can be pre-computed and saved separately since this is the step that takes the longest to run. Also, this is the only step that depends on rtracklayer's functionality for reading BigWig files which does not run on Windows OS. So it could be run on a non-Windows machine, saved, and then shared with Windows users.

CODING_ONLY   A logical vector of length 1 specifying whether to subset the Annotated Genes to only the coding genes. That is, whether to subset the genes by whether they have a non-NA CSS value. The Annotated Genes are downloaded with `GenomicState::GenomicStateHub()`.

VERBOSE   A logical value indicating whether to print updates from the process of loading the data from the BigWig files.

Value

`plot_coverage` plots all input sequences using [`derfinderPlot::plotRegionCoverage()`](#). It returns a plot for each input candidate sequence listed in REGION. Each plot includes coverage of the sequence(s) in nuclear (N) and cytoplasmic (C) RNA isolated from adult (A) and fetal (F) prefrontal cortex, sequenced using two different library preparation methods. PolyA+ libraries (P) were generated using selection for polyadenylated transcripts, and RiboZero (R) libraries were generated using a ribosomal RNA depletion protocol.

Each plot also shows the overlapping genes beneath the coverage, and the genomic location. The title lists the nearest gene, the position of the sequence relative to the gene's canonical transcriptional start site (TSS), and further annotation information as described in the 'region' column from [`bumphunter::matchGenes()`](#).

`plot_coverage` saves the results as regionCoverage_fractionedData.pdf in a temporary directory unless otherwise specified with OUTDIR.

Author(s)

Amanda J Price

Examples

```
## Here we use the pre-saved example coverage data such that this example
## will run fast!
plot_coverage("chr20:10286777-10288069:+",
    COVERAGE = four_panels_example_cov
)

## Without using COVERAGE, this function reads BigWig files from the web
## using rtracklayer and this functionality is not supported on Windows
## machines.
if (.Platform$OS.type != "windows") {
    plot_coverage("chr20:10286777-10288069:+",
        PDF = "regionCoverage_fractionedData_fromScratch.pdf"
    )
}
## Not run:

## These examples will take a few minutes to run!
plot_coverage(c(
    "chr20:10286777-10288069:+",
    "chr18:74690788-74692427:-",
    "chr19:49932861-49933829:-"
))

candidates <- c(
    "chr20:10286777-10288069:+",
    "chr18:74690788-74692427:-",
    "chr19:49932861-49933829:-"
)

## General syntax:
plot_coverage(candidates,
    PDF = "PDF_file.pdf",
    OUTDIR = "/path/to/directory/"
)

plot_coverage("chr20:10286777-10288069:+",
    PDF = "PDF_file.pdf", OUTDIR = "/path/to/directory/"
)

## Explore the effect of changing CODING_ONLY
## Check how gene name and distance to TSS changes in the title of the plot
## (everything else stays the same)
cov <- brainflowprobes_cov("chr10:135379301-135379311:+")
plot_coverage("chr10:135379301-135379311:+", COVERAGE = cov)
plot_coverage("chr10:135379301-135379311:+",
    COVERAGE = cov,
    PDF = "coding_only_plot_coverage", CODING_ONLY = TRUE
```

```
)

## End(Not run)
```

---

region_info   *Print relevant info about candidate probe sequence.*

Description

region_info returns annotation of a single potential probe sequence or list of sequences and, if
specified, prints the resuts in a .csv file.

Usage

```
region_info(
  REGION,
  CSV = TRUE,
  SEQ = TRUE,
  OUTDIR = tempdir(),
  CODING_ONLY = FALSE
)
```

Arguments

REGION   Either a single hg19 genomic sequence including the chromosome, start, end,
                and optionally strand separated by colons (e.g., 'chr20:10199446-10288068:+'),
                or a string of sequences to be annotated. Must be character. Chromosome must
                be proceeded by 'chr'.

CSV   A logical(1) value indicating if the results should be exported in a .csv file.

SEQ   A 'logical(1)" value indicating if the base sequence should be returned.

OUTDIR   If a .csv file is to be exported, this parameter indicates the path where the file
                should be saved. By default the file will be saved in a temporary directory.

CODING_ONLY   A logical vector of length 1 specifying whether to subset the Annotated Genes
                to only the coding genes. That is, whether to subset the genes by whether
                they have a non-NA CSS value. The Annotated Genes are downloaded with
                GenomicState::GenomicStateHub().

Value

This function annotates all input sequences using bumphunter::matchGenes(). It returns a data
frame where each row is a genomic sequence specified in REGION. The columns c('seqnames',
'start', 'end', 'width', 'strand') list the chromosome, range, sequence length, and strand of the
REGION. The columns c('name', 'annotation', 'description', 'region', 'distance', 'subregion', 'in-
sideDistance', 'exonnumber', 'nexons', 'UTR', 'geneL', 'codingL', 'Geneid', 'subjectHits') are
described in bumphunter::matchGenes() documentation.

If SEQ=TRUE, a column 'Sequence' will be included. This is recommended for sending the probe sequence to be synthesized.

If CSV=TRUE, a .csv file called region_info.csv will be saved to a temporary directory unless otherwise specified in OUTDIR.

Author(s)

Amanda J Price

Examples

```
x <- region_info("chr20:10286777-10288069:+", CSV = FALSE)
head(x)

## You can easily transform this data.frame to a GRanges object
GenomicRanges::GRanges(x)

y <- region_info(
    c(
        "chr20:10286777-10288069:+",
        "chr18:74690788-74692427:-",
        "chr19:49932861-49933829:-"
    ),
    CSV = FALSE, SEQ = FALSE
)
head(y)

candidates <- c(
    "chr20:10286777-10288069:+",
    "chr18:74690788-74692427:-",
    "chr19:49932861-49933829:-"
)
region_info(candidates, CSV = FALSE)

## Explore the effect of changing CODING_ONLY
## Check how the "distance", "name", "Geneid" among other values change
region_info("chr10:135379301-135379311:+", CSV = FALSE)
region_info("chr10:135379301-135379311:+", CSV = FALSE, CODING_ONLY = TRUE)
## Not run:
region_info(candidates, OUTDIR = "/path/to/directory/")

region_info("chr20:10286777-10288069:+", OUTDIR = "/path/to/directory")

## End(Not run)
```

Index