Package 'primirTSS'

May 17, 2024

Title Prediction of pri-miRNA Transcription Start Site

Version 1.22.0

Author Pumin Li [aut, cre], Qi Xu [aut], Jie Li [aut], Jin Wang [aut]

Maintainer Pumin Li <ipumin@163.com>

Description A fast, convenient tool to identify the TSSs of miRNAs by integrating the data of H3K4me3 and Pol II as well as combining the conservation level and sequence feature, provided within both command-line and graphical interfaces, which achieves a better performance than the previous non-cell-specific methods on miRNA TSSs.

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

Depends R (>= 3.5.0)

Imports GenomicRanges (>= 1.32.2), S4Vectors (>= 0.18.2), rtracklayer (>= 1.40.3), dplyr (>= 0.7.6), stringr (>= 1.3.1), tidyr (>= 0.8.1), Biostrings (>= 2.48.0), purrr (>= 0.2.5), BSgenome.Hsapiens.UCSC.hg38 (>= 1.4.1), phastCons100way.UCSC.hg38 (>= 3.7.1), GenomicScores (>= 1.4.1), shiny (>= 1.0.5), Gviz (>= 1.24.0), BiocGenerics (>= 0.26.0), IRanges (>= 2.14.10), TFBSTools (>= 1.18.0), JASPAR2018 (>= 1.1.1), tibble (>= 1.4.2), R.utils (>= 2.6.0), stats, utils

Suggests knitr, rmarkdown

VignetteBuilder knitr

biocViews ImmunoOncology, Sequencing, RNASeq, Genetics, Preprocessing, Transcription, GeneRegulation

URL https://github.com/ipumin/primirTSS

BugReports http://github.com/ipumin/primirTSS/issues

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

git_branch RELEASE_3_19

git_last_commit 3d986ad

git_last_commit_date 2024-04-30

Repository Bioconductor 3.19

Date/Publication 2024-05-16

Contents

`find_tss` *Predict TSSs of miRNA*

Description

Search for putative TSSs of miRNA, together with integrating available data such as H3K4me3 data, Pol II data, miRNA expression data, and protein-coding gene data, as well as provide the transcriptional regulation relationship between TF and miRNA.

Usage

```
find_tss(
  bed_merged,
  expressed_mir = "all",
  flanking_num = 1000,
  threshold = 0.7,
  ignore_DHS_check = TRUE,
  DHS,
  allmirdhs_byforce = TRUE,
  expressed_gene = "all",
  allmirgene_byforce = TRUE,
  seek_tf = FALSE,
  tf_n = 1000,
  min.score = 0.8
)
```

Arguments

bed_merged	Peaks from ChIP-seq data to be provided for analysis can be H3K4me3 peaks, Pol II peaks or both. Notice that peaks are supposed to be merged(see also peak_merge) before find_TSS if using only one kind of peak data, while peaks should be firstly merged and then join together(see also peak_join) if both H3K4me3 data and Pol II are input.
expressed_mir	This parameter allows users to specify certain miRNAs, the TSSs of which they want to search for by providing a list of miRNAs(e.g., expressed miRNAs in a certain cell-line). If expressed_mir is not specified, the default value of the parameter is "all" and the function will acquiescently employ all the miRNAs currently listed on "miRbase" database.
flanking_num	A parameter in Eponine model to detect TSSs. It is concluded that a peak signal with flanking regions of C-G enrichment are important to mark TSSs. The default value is 1000.
threshold	The threshold for candidate TSSs scored with Eponine method. The default value is 0.7.
ignore_DHS_check	The process of DHS_check further assists to filter putative TSSs. When there is a DHS peak that locates within 1 kb upstream of a putative TSS, this predicted TSS will be retained for its character is consistent with that of an authentic TSS. Or the TSSs with no DHSs locating within 1 kb upstream of them would be discarded.
DHS	ChIP-seq data of DNase I hypersensitive sites(DHSs).
allmirdhs_byforce	When we use DHS data to check the validity of TSSs, there is a possibility where no DHSs locates within 1 kb upstream of all putative TSSs and all these putative TSSs might be filtered out by our method resulting no outputs. While "allmirdhs_byforce = TRUE", it ensures to output at least 1 most possible TSS even if the nearest DHS signal locates more than 1 kb upstream of this TSS.
expressed_gene	Users can specify genes expressed in certain cell-lines that are analyzed. Or the default value is "all", which means all the expressed genes annotated on Ensemble will be employed.
allmirgene_byforce	While integrating expressed_gene data to improve prediction, there might be a circumstance where all the putative TSS are discarded. To prevent this condition, users are allowed to use "allmirgene_byforce = TRUE" to ensure at least 1 putative TSS for each miRNA will be output.
seek_tf	With the result of predicted TSSs, seek_tf provides users with an option to predict related TFs for miRNA. The data of transcription factors refer to JASPAR2018 database.
tf_n	TFBS locates on the upstream of the TSS of a certain TF, which is considered as the promoter region. tf_n set the length of promoter region for predicting transcription regulation between miRNAs and TFs.
min.score	The threshold for scoring transcription factor binding sites. A single absolute value between 0 and 1.

Value

The first part of the result returns details of predicted TSSs, composed of seven columns: `mir_name`, `chrom`, `stem_loop_p1`, `stem_loop_p2`, `strand` `mir_context`, `tss_type` `gene` and `predicted_tss`:

`mir_name`: Name of miRNA.

`chrom`: Chromosome.

`stem_loop_p1`: The start site of a stem-loop.

`stem_loop_p2`: The end site of a stem-loop.

`strand`: Polynucleotide strands. (`+/-`)

`mir_context`: The relative positon relationship between stem-loop and protein-coding gene. (`intra/inter`)

`tss_type`: Four types of predicted TSSs. See the section below `TSS types` for details. (`host_TSS/intra_TSS/overlap_int`

`gene`: Ensembl gene ID

`predicted_tss`: Predicted transcription start sites(TSSs).

`pri_tss_distance`: The distance between a predicted TSS and the start site of the stem-loop.

TSS types

TSSs are catalogued into 4 types as below.

`host_TSS` The TSSs of miRNA that are close to the TSS of protein-coding gene implying they may share the same TSS, on the condition where `mir_context` is "`intra`". (See above: `Value-mir_context`)

`intra_TSS` The TSSs of miRNA that are NOT close to the TSS of the protein-coding gene, on the condition where `mir_context` is "`intra`".

`overlap_inter_TSS` The TSSs of miRNA are catalogued as "`overlap_inter_TSS`" when the pri-miRNA gene overlaps with Ensembl gene, on the condition where "`mir_context`" is "`inter`".

`inter_inter_TSS` The TSSs of miRNA are catalogued as "`inter_inter_TSS`" when the miRNA gene does NOT overlap with Ensembl gene, on the condition where "`mir_context`" is "`inter`".

(See Xu HUA et al 2016 for more details)

Log

The second part of the result returns logs during the process of prediction: `find_nearest_peak_log` If no peaks locate in the upstream of a stem-loop to help determine putative TSSs of miRNA, we will fail to find the nearest peak and this miRNA will be logged in `find_nearest_peak_log`.

`eponine_score_log` For a certain miRNA, if none of the candidate TSSs scored with `Eponine` `method` meet the threshold we set, we will fail to get an eponine score and this miRNA will be logged in `eponine_score_log`.

`DHS_check_log` For a certain miRNA, if no DHS signals locate within 1 kb upstream of each putative TSSs, these putative TSSs will be filtered out and this miRNA will be logged in `DHS_check_log`.

`gene_filter_log` For a certain miRNA, when integrating expressed_gene data to improve prediction, if no putative TSSs are confirmed after considering the relative position relationship among TSSs, stem-loops and expressed genes, this miRNA will be filtered out and logged in `gene_filter_log`.

Reference

Xu Hua, Luxiao Chen, Jin Wang*, Jie Li* and Edgar Wingender*, Identifying cell-specific microRNA transcriptional start sites. Bioinformatics 2016, 32(16), 2403-10.

Examples

```
bed_merged <- data.frame(
                chrom = c("chr1", "chr1", "chr1", "chr1", "chr2"),
                start = c(9910686, 9942202, 9996940, 10032962, 9830615),
                end = c(9911113, 9944469, 9998065, 10035458, 9917994),
                stringsAsFactors = FALSE)
bed_merged <- as(bed_merged, "GRanges")

## Not run:
ownmiRNA <- find_tss(bed_merged, expressed_mir = "hsa-mir-5697",
                     ignore_DHS_check = TRUE,
                     expressed_gene = "all",
                     allmirgene_byforce = TRUE)

## End(Not run)
```

---

peak_join *Integrate H3K4me3 data and Pol II data.*

Description

Integrate peaks from H3K4me3 and Pol II data. To conduct the overlappd ranges for the further analysis by imposing H3K4me3 peaks on Pol II peaks, if both of these two different kinds of ChIP-seq data are available.

Usage

```
peak_join(peak1, peak2)
```

Arguments

peak1	H3K4me3 peaks. Merged peak data as GRange object by fucntion peak_merge
peak2	Pol II peaks. Merged peak data as GRange object by fucntion peak_merge

Value

A GRanges object. The joined peaks for the following analysis to search for TSSs.

Detail

Peak1 and peak2 are signals seperately from the ChIP-seq data of H3K4me3 and Pol II data that to be integrated. The data is GRange object containing three columns Chrom, Ranges, Strand. And the order of these two kinds of data when input as peak1 and peak2 can be swapped.

Examples

```
peak_df1 <- data.frame(chrom = c("chr1", "chr1", "chr1", "chr2"),
                       start = c(100, 460, 600, 70),
                       end = c(200, 500, 630, 100),
                       stringsAsFactors = FALSE)
peak1 <-  as(peak_df1, "GRanges")

peak_df2 <- data.frame(chrom = c("chr1", "chr1", "chr1", "chr2"),
                       start = c(160, 470, 640, 71),
                       end = c(210, 480, 700, 90),
                       stringsAsFactors = FALSE)
peak2 <-  as(peak_df2, "GRanges")

peak_join(peak1, peak2)
```

`peak_merge` *Merge adjacent peaks within H3K4me3 or Pol II data.*

Description

Merge the adjacent segments provided as `GRange` object from original data. This function will merge adjacent peaks the distance between which is less than n base pairs apart and then return the merged segments.

Usage

```
peak_merge(peak, n = 250)
```

Arguments

`peak`	A `GRange` object. The peaks to be merged from one certain ChIP-seq data, such as H3K4me3 data or Pol II data.
`n`	A number. `n` stipulates the distance(bp, base pair) between two seperate peaks within which they should be merged.

Value

A GRanges object. The merged peaks for the following analysis to search for TSSs.

Examples

```
peak_df <- data.frame(chrom = c("chr1", "chr2", "chr1"),
                      chromStart = c(450, 460, 680),
                      chromEnd = c(470, 480, 710),
                      stringsAsFactors = FALSE)
peak <-  as(peak_df, "GRanges")

peak_merge(peak, n =250)
```

plot_primiRNA *Plot the result of prediction for miRNA*

Description

For each miRNA, plot the position of TSS, pri-miRNA, related Ensemble gene, eponine socre and conservation score according to the result of prediction using primirTSS.

Usage

```
plot_primiRNA(
  expressed_mir,
  bed_merged,
  flanking_num = 1000,
  threshold = 0.7,
  ignore_DHS_check = TRUE,
  DHS,
  allmirdhs_byforce = TRUE,
  expressed_gene = "all",
  allmirgene_byforce = TRUE
)
```

Arguments

expressed_mir	This parameter allows users to specify certain miRNAs, the TSSs of which they want to search for by providing a list of miRNAs(e.g. expressed miRNAs in a certain cell-line). If expressed_mir is not specified, the default value of the parameter is "all" and the function will acquiescently employ all the miRNAs currently listed on "miRbase" database.
bed_merged	Peaks from ChIP-seq data to be provided for analysis can be H3K4me3 peaks, Pol II peaks or both. Notice that peaks are supposed to be merged(see also peak_merge) before find_TSS if using only one kind of peak data, while peaks shoud be firstly merged and then join together(see also peak_join) if both H3K4me3 data and Pol II are input.
flanking_num	A parameter in Eponine model to detect TSSs. It is concluded that a peak signal with flanking regions of C-G enrichment are important to mark TSSs. The default value is 1000.
threshold	Threshold for candidate TSSs scored with Eponine method. The default value is 0.7.
ignore_DHS_check	The process of DHS_check further assist to filter putative TSSs. When there are a DHS peak that locates within 1 kb upstream of a putative TSS, this predicted TSS will be retain for it character is consistent with that of an authentic TSS. Or the TSSs with no DHSs locating within 1 kb upstream of them would be discard.
DHS	ChIP-seq data of DNase I hypersensitive sites(DHSs).

`allmirdhs_byforce`
: When we use DHS data to check the validity of TSSs, there is possibility where no DHSs locates within 1 kb upstream of all putative TSSs and all these putative TSSs might be filtered out by our method resulting no outputs. While "`allmirdhs_byforce = TRUE`", it ensures to output at least 1 most possible TSS even if the nearest DHS signal locates more than 1 kb upsteam of this TSS.

`expressed_gene`
: Users can speicify genes expressed in certain cell-lines that is analyzed. Or the default value is "`all`", which means all the expressed genes annotated on Ensemble will be employed.

`allmirgene_byforce`
: While integrating expressed_gene data to improve prediction, there might be a circumstance where all the putative TSS are discarded. To prevent this condition, users are allowed to use "`allmirgene_byforce = TRUE`" to ensure at least 1 putative TSS for each miRNA will be output.

Details

NOTICE that this function is used for visualizing the predicted result of ONLY ONE specific miRNA every single time.

Value

There will be six tracks plotted as return:

`Chrom`: Position of miRNA on the chromosome.

`hg38`: Reference genome coordinate in hg38.

`pri-miRNA`: Position of pri-miRNA.

`Ensemble genes`: Position of related protein-coding gene.

`eponine score`: Score of best putative TSS conducted by eponine method.

`conservation score`: Conservation score should be integrated with eponine score to find out putative TSSs.

Examples

```
expressed_mir <- "hsa-mir-5697"
bed_merged <- data.frame(
  chrom = c("chr1", "chr1", "chr1", "chr1", "chr2"),
  start = c(9180799, 9201483, 9234339, 9942202, 9830615),
  end = c(9183889, 9202580, 9235853, 9944469, 9917994),
  stringsAsFactors = FALSE
)
bed_merged <- as(bed_merged, "GRanges")
## Not run:
plot_primiRNA(expressed_mir, bed_merged)

## End(Not run)
```

primirTSS *primirTSS: Search for putative TSSs of miRNA*

Description

A fast, convenient tool to identify the TSSs of miRNAs by integrating the data of H3K4me3 and Pol II as well as combining the conservation level and sequence feature, provided within both command-line and graphical interfaces, which achieves a better performance than the previous non-cell-specific methods on miRNA TSSs.

Detail

See `find_tss` for deailted instruction to predict TSSs of miRNA; See `run_primirTSSapp` to predict TSSs of miRNA using graphical web interface.

Author(s)

Maintainer: Pumin Li <ipumin@163.com>

Other contributors: Qi Xu <xuqi@vip.qq.com>

See Also

Useful links: Xu HUA et al

run_primirTSSapp *Predict TSSs of miRNA using a graphical web interface.*

Description

A graphical web interface is provided for users to achieve the functions of `find_tss` and `plot_primiRNA` to intuitively and conveniently predict putative TSSs of miRNA.

Usage

```
run_primirTSSapp()
```

Details

Users can refer documents of the two functions mentioned ABOVE for details.

Value

A graphical interface.

Examples

```
## Not run:
run_primirTSSapp()

## End(Not run)
```

---

trans_cor *transform one hg coordinates to another*

Description

Convert coordinates between different genomes when necessary.

Usage

```
trans_cor(peak, hg_from, hg_to)
```

Arguments

peak	A GRange object. The genome, the coordinates of which need to be coverted.
hg_from	The genome are coverting from. This parameter can be "hg18", "hg19" or "hg38", etc.
hg_to	Which type the genome is converting to. This parameter can be "hg18", "hg19" or "hg38", etc. NOTICE hg_from and hg_to should be different from each other.

Value

A GRanges object.

Examples

```
peak_df <- data.frame(chrom = c("chr7", "chr7", "chr7"),
                      chromStart = c(128043908, 128045075, 128046242),
                      chromEnd = c(128045074, 128046241, 128047408),
                      stringsAsFactors = FALSE)
peak <-  as(peak_df, "GRanges")

trans_cor(peak, "hg19", "hg38")
```

Index