

# *RTFBSDB* package

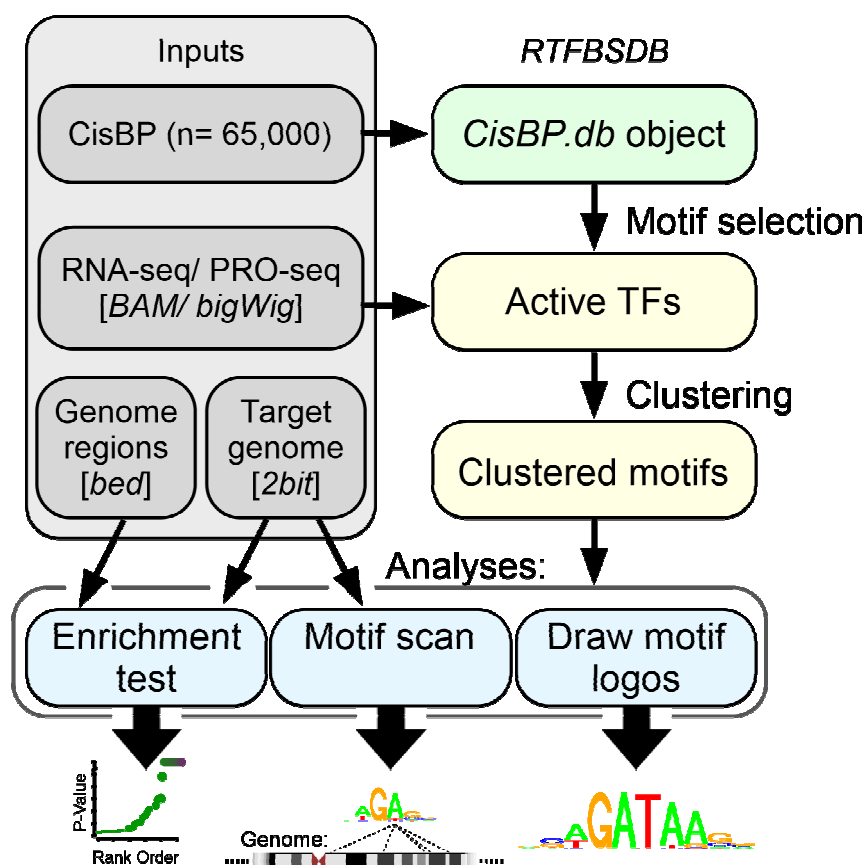
Rtfbsdb version:0.39

Date: 01/26/2017

## 1. Overview

The *rtfbsdb* package provides a convenient platform to find and analyze transcription factor (TF) binding sites in the R environment. Experimentally derived and predicted motifs are imported from the Cis-BP database<sup>[1]</sup>, or other data sources (e.g. Jaspar<sup>[2]</sup>, MotifDb<sup>[3]</sup>, HOCOMOCO<sup>[4]</sup>), which contain thousands of motifs in virtually any species of interest. The following instructions will show you how to use the *rtfbsdb* package to search the genome for motif occurrences, how to identify motifs enriched between sets of DNA sequences, and how to visualize motifs in R.

### 1.1 Flowchart of data analysis



## 1.2 External dependencies

The *rtfbsdb* package depends not only on other R packages, but also on several UNIX shell commands and other bioinformatics tools. Before you go through the following instructions, please check the requisite commands are available in your operating system \$PATH variable and execute normally when run on a command line.

| Command                            | Package                 | Download Link   |
|------------------------------------|-------------------------|---|
| <i>sort-bed, starch, starchcat</i> | bedops <sup>[5]</sup>   | <a href="http://bedops.readthedocs.org/en/latest/index.html">http://bedops.readthedocs.org/en/latest/index.html</a> |
| <i>twoBitInfo, twoBitToFa</i>      | Kent source             | <a href="http://hgdownload.cse.ucsc.edu/admin/exe/">http://hgdownload.cse.ucsc.edu/admin/exe/</a>                   |
| <i>bedtools</i>                    | bedtools <sup>[6]</sup> | <a href="http://bedtools.readthedocs.org/en/latest/">http://bedtools.readthedocs.org/en/latest/</a>                 |
| <i>awk, zcat</i>                   | Linux/Unix command      |   |

## 1.3 How to cite this package

Wang, Z., Martins, A. L., & Danko, C. G. (2016). **RTFBSDB: an integrated framework for transcription factor binding site analysis**. *Bioinformatics*, 32(19), 3024-3026.

## 2. TF site binding identification

### 2.1 Loading TF information from the Cis-BP database

If you are planning to analyze the human or mouse genome, you don't need to download the Cis-BP dataset because the package includes pre-installed data for human and mouse. Otherwise, you need to download the dataset manually for your target genome or use the function in *rtfbsdb* to download it. The Cis-BP database provides a very nice web page to download the TF information for any species or TF family at <http://cisbp.cabr.utoronto.ca/bulk.php>.

In this package, three functions, including *CisBP.extdata*, *CisBP.download* and *CisBP.zipload*, aim to build a CisBP data object. *CisBP.extdata* can load the pre-installed dataset for human and mouse, *CisBP.download* can download and use any species data from the Cis-BP web site, *CisBP.zipload* can decompress a zipped dataset from the Cis-BP web site. For example:

```
## Attach the package firstly
library(rtfbsdb);

##Create db from pre-installed dataset
db <- CisBP.extdata("Homo_sapiens");
db1<- CisBP.extdata("Mus_musculus");
db2<- CisBP.extdata("Drosophila_melanogaster");

## Create db from downloaded dataset
db3<- CisBP.download("Mus_musculus");

## Create db from a zipped downloaded dataset
db4<- CisBP.zipload("ZIP_FILE_FROM_CISBP.zip", species="Mus_musculus");
```

A zipped file downloaded from Cis-BP includes a database of TF definitions and matched position-specific weight matrices (PWMs), which represent the DNA binding preferences of each TF. Note that Cis-BP downloads omit binding information from the TRANSFAC database, which requires a paid license. If the licensed motifs are available, the function *tfbs.importMotifs* can be used to merge these motifs into the *tfbs* object after the motif selecting from Cis-BP data file. The Cis-BP zip file includes images representing motif logos which are not used in this package.

## 2.2 Selecting motif data to create *tfbs* object

Once you have created a Cis-BP object you can load PWM information from a subset of motifs using the function *tfbs.loadFromCisBP*. You can select all motifs in the database (default) or you can select a subset of motifs. For example, selecting motifs binding to TFs in the AP-2 family can be accomplished using:

```
## Select all motifs from CisBP dataset
tfs <- tfbs.createFromCisBP(db);

## Query the CisBP dataset and select the motifs for a transcription factor of interest
tfs.ap2<- tfbs.createFromCisBP(db, family_name="AP-2");
```

The *tfbs* object returned by *tfbs.createFromCisBP* includes each selected PWM, the ENSEMBL IDs of genes encoding TF binding and other relevant data. You can get an overview by the command *show*. For human, 1,964 valid motifs are loaded from Cis-BP (12/8/2015) after removing the database entries without a freely available PWM.

```
> show(tfs);

Species: Homo_sapiens
TF number: 1964
Distance Matrix: NULL
Cluster Matrix: NULL
Expression: NULL

Partial list of TFs
  Motif_ID DBID TF_Name Family_Name Motif_Type MSource_Identifier
2 M5917_1.02 ENSG00000008196 TFAP2B AP-2 SELEX Jolma
3 M5918_1.02 ENSG00000008196 TFAP2B AP-2 SELEX Jolma
4 M5919_1.02 ENSG00000008196 TFAP2B AP-2 SELEX Jolma
.....
```

The *show* function displays main attributes briefly. The specific attributes can be accessed by the *@* function as follows:

```
# Show the structure of 'tfbs' object. It may be very long if the object has many PWMs
> str(tfs);
Formal class 'tfbs' [package "rtfbsdb"] with 10 slots
..@ species      : chr "Homo_sapiens"
..@ ntfs         : int 1964
..@ mgisymbols   : chr [1:1964] ...
..@ filename     : chr [1:1964] ...
..@ pwm         : List of 1964
..@ tf_info      : 'data.frame': 1964 obs. of 26 variables:
...$ TF_ID       : Factor w/ 1734 levels, ...
```

```

...$ Family_ID      : Factor w/ 68 levels, ...
...$ ...
...$ TfSource_Day   : int [1:1744]...
..@ distancematrix  : logi[0, 0 ]
..@ cluster         : logi[0, 0 ]
..@ expressionlevel : 'data.frame': 0 obs. of  0 variables

# Access the first 6 rows of TF information
> head( tfs@tf_info );
      TF_ID Family_ID TSource_ID Motif_ID MSource_ID      DBID
1 T004843_1.02 F035_1.02 TS19_1.02 M5917_1.02 MS20_1.02 ENSG00000008196 ...
2 T004843_1.02 F035_1.02 TS19_1.02 M5918_1.02 MS20_1.02 ENSG00000008196 ...
3 T004843_1.02 F035_1.02 TS19_1.02 M5919_1.02 MS20_1.02 ENSG00000008196 ...
4 T004843_1.02 F035_1.02 TS19_1.02 M6144_1.02 MS18_1.02 ENSG00000008196 ...
5 T004844_1.02 F035_1.02 TS19_1.02 M6146_1.02 MS18_1.02 ENSG00000008197 ...
6 T004845_1.02 F035_1.02 TS19_1.02 M2320_1.02 MS19_1.02 ENSG000000087510 ...

# Check first PWM in the TS19_1.02 M2
> tfs@pwm[[1]] ;
      A      C      G      T
1 -1.4906418 -1.42706393 -2.87136511 -0.7378563
2 -1.9987010 -1.59522204 -0.44463743 -3.8839866
3 -5.0210160 -0.09218827 -2.59198982 -5.0210160
4 -8.0752715 -0.01094959 -7.00000000 -4.5489110
5 -5.6667412 -0.43709209 -3.04075595 -1.1945831
...

```

## 2.3 Selecting motifs recognized by expressed TFs

The *tfbs.loadFromCisBP* function will also select motifs that are recognized by TFs expressed in a cell type or biological system of interest. The expression of each TF is measured using functional data profiling gene expression levels. Currently *rtfbsdb* supports either RNA-seq or PRO-seq. This feature requires arguments specifying: (1) 2-bit formatted genome sequence data (e.g., hg19.2bit), (2) Gencode gene coordinates (GTF-formatted, <http://www.gencodegenes.org/releases/19.html>), and (3) Gene expression data in the desired format. Gene expression data collected using RNA-seq requires an indexed BAM file to calculate gene expression. For PRO-seq, bigwig files representing read densities on the plus and minus strand are required. The *tfbs.createFromCisBP* function tests regions defined in the Gencode file that are same as the motif's gene ID for read densities that exceed a background null model, as presented by Core, Waterfall, and Lis<sup>[7]</sup>.

Here we extract the 19<sup>th</sup> chromosome from the RNA-seq and PRO-seq data to demonstrate how to use the function *tfbs.selectExpressedMotifs*. First example is for RNA-seq data.

```

## Specify the BAM file to filter the expressed TFs only (RNA-seq)
file.bam.chr19<- system.file("extdata","human_PI_fastq_gz_sort_chr19.bam", package="rtfbsdb")

## Get the partial hg19 (chromosome 19 only) file from pre-installed data
file.hg19.twobit.chr19<- system.file("extdata","hg19.chr19.2bit",
                                     package="rtfbsdb")

## Get the partial gencode (chromosome 19 only) file from pre-installed data
file.gencode.gtf.chr19<- system.file("extdata","gencode.v19.annotation.chr19.gtf.gz",
                                     package="rtfbsdb")

## Call the function to do filter

```

```
tfs <- tfbs.selectExpressedMotifs(tfs,
  file.hg19.twobit.chr19,
  file.gencode.gtf.chr19,
  file.bam=file.bam.chr19,
  pvalue.threshold=0.001,
  include.DBID.missing=TRUE,
  seq.datatype="RNA-seq");
```

The second example uses PRO-seq experiment data.

```
## Specify the bigwig files to filter the expressed TFs only (PRO-seq)
file.bigwig.minus.chr19<- system.file("extdata","GSM1480327_K562_PROseq_chr19_minus.bw",
  package="rtfbsdb")
file.bigwig.plus.chr19<- system.file("extdata","GSM1480327_K562_PROseq_chr19_plus.bw",
  package="rtfbsdb")

## Get the partial hg19 (chromosome 19 only) file from pre-installed data
file.hg19.twobit.chr19<- system.file("extdata","hg19.chr19.2bit",
  package="rtfbsdb")

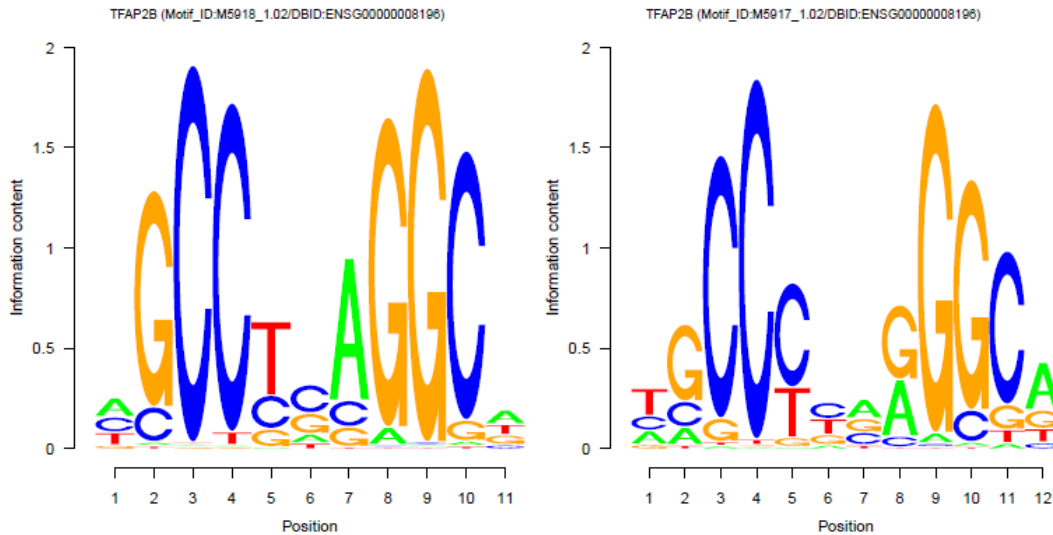
## Get the partial gencode (chromosome 19 only) file from pre-installed data
file.gencode.gtf.chr19<- system.file("extdata","gencode.v19.annotation.chr19.gtf.gz",
  package="rtfbsdb")

## Call the function to do filter
tfs <- tfbs.selectExpressedMotifs(tfs,
  file.hg19.twobit.chr19,
  file.gencode.gtf.chr19,
  file.bigwig.plus.chr19,
  file.bigwig.minus.chr19,
  pvalue.threshold=0.001,
  include.DBID.missing=TRUE,
  seq.datatype="PRO-seq");
```

## 2.4 Visualizing motif logos

Motif logos can be drawn at any point after loading. To draw a single specific motif, use the function *tfbs.drawLogo* to draw a motif logo. The seqLogo<sup>[8]</sup> package is employed to draw motif logos.

```
## Draw motif logos with only one diagram per page
tfbs.drawLogo(tfs, file.pdf="logos.pdf", motif_id=c("M5917_1.02", "M5918_1.02") )
```

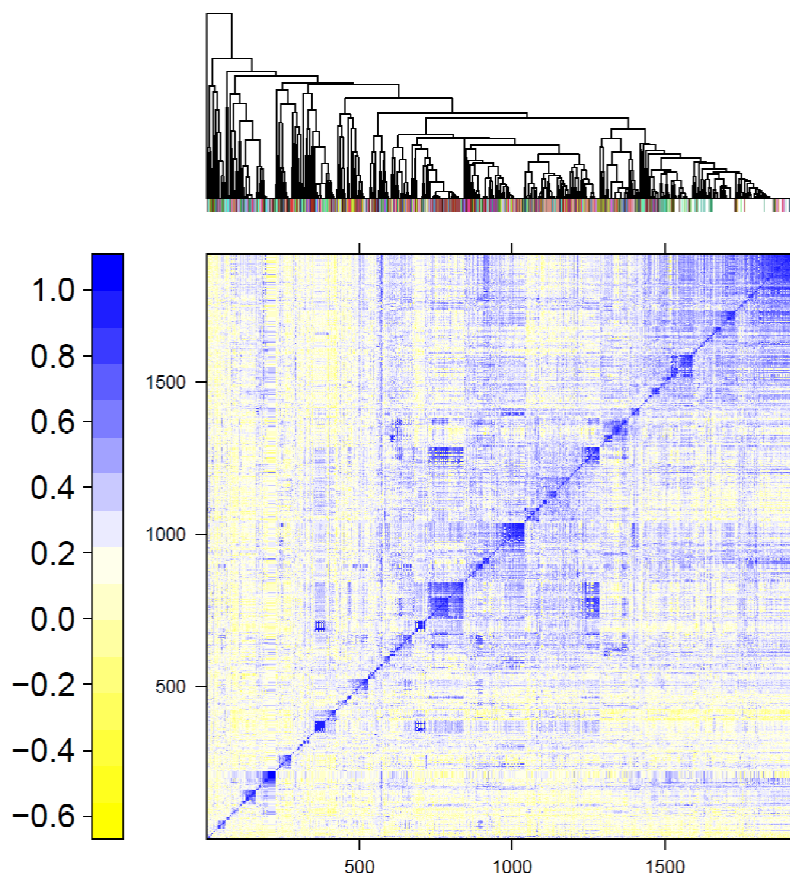


## 2.5 Clustering (optional)

Many of the motifs indexed in Cis-DB share similar underlying DNA sequence preferences. To increase the power of certain tests, motifs with similar DNA sequence binding preferences can optionally be grouped using hierarchical clustering. The clustering function (*tfbs.clusterMotifs*) can be set to use either of two algorithms: hierarchical agglomerative clustering (*agnes* in the R cluster package) and Affinity Propagation Clustering (*apcluster*<sup>[9]</sup>). In order to clustering the motifs, the similarity matrix need to be obtained from comparing each combination of motifs, results in a distance matrix with Pearson's R values. This computational procedure can take a long time to execute if the number of motifs is large, so it is not performed in the constructor of the *tfbs* object. To speed it up, it can be run on multiple cores by setting the *ncores* parameter to a value larger than 1. Optionally, *tfbs.clusterMotifs* will output a heat-map representing the similarity between groups of motifs if an output filename is specified in the *pdf.heatmap* parameter.

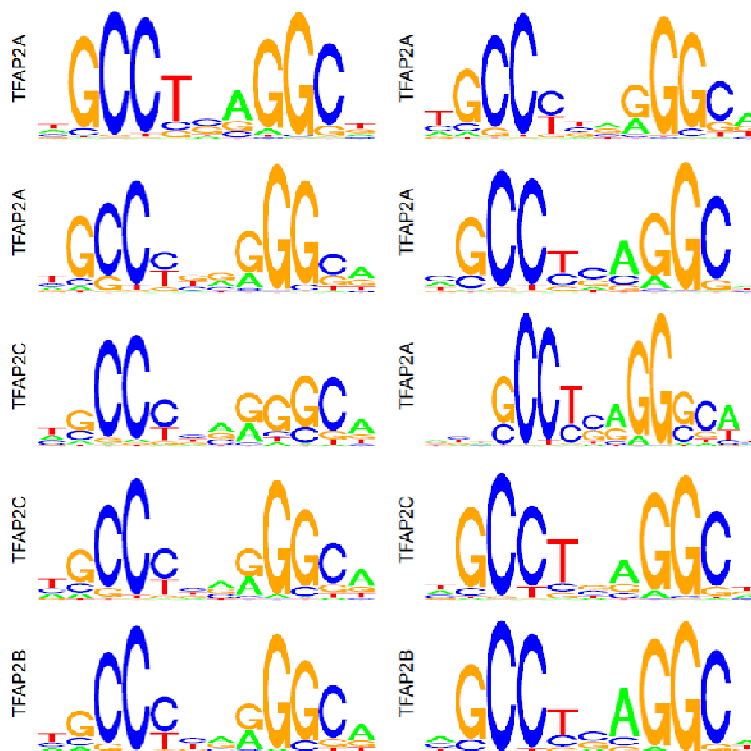
Below is an example of the code to compute the distance matrix, cluster motifs, and generate a figure. The *tfbs.clusterMotifs* function returns a new *tfbs* object with the cluster mapping table.

```
## Generate heatmap and return clustering map
tfs <- tfbs.clusterMotifs(tfs, method="agnes", pdf.heatmap="heatmap.pdf", ncores=25)
```



Besides drawing the heat-map, users can also draw sequence logos for motifs within each cluster using the *tfbs.drawLogosForClusters* function. Each page of the output PDF contains all motif logos that are grouped within a single cluster. This visualization can be useful when checking whether the clustering was conducted using a reasonable number of clusters. The following code demonstrates how to plot group motif logos.

```
## Draw motif logos with one group of TF per page
tfbs.drawLogosForClusters(tfs, "clustering.logos.pdf");
```



## 2.6 Motif selection and plotting

A single motif representing each cluster is used in most downstream analysis. We provide two methods to select which motifs are used to represent each cluster in downstream analyses. The function *tfbs.selectByRandom* randomly selects one motif from each group of clustering. The function *tfbs.selectByGeneExp* selects one motif with the minimum p-value of gene expression from each group of clustering. The index returned from these two functions can be used in the function call of *tfbs.scanTFsite* and *tfbs.enrichmentTest*.

## 2.7 Find TF binding sites across the genome

The first goal of this package is to locate TF binding sites across a genome based on the *rtfbs*<sup>[10]</sup> package. The *tfbs.scanTFsite* function matches user selected PWM(s) across the genome given the DNA sequence formatted as a 2 bit file (e.g., hg19.2bit). If desired, users can restrict the motif search to occur within a set of genomic coordinates specified using a BED-formatted data frame. Below are two examples that show a motif scan across the whole genome and restricted to a specified range.

```
## hg19.2bit is downloaded from http://hgdownload-test.cse.ucsc.edu/goldenPath/hg19/bigZips/
file.twoBit.chr19<- system.file("extdata","hg19.chr19.2bit", package="rtfbsdb")

## Example 1: Scan the whole genome
## Scan 2bit file within whole genome to find motif binding site
r1.scan <- tfbs.scanTFsite( tfs, file.twoBit.chr19, ncores = 3);

## Example 2: Scan a specified range
## Get a data frame from a plain-text bed file for your range of interest
```



```

file.ELF1.chr19<- system.file("extdata","Chipseq-k562-chr19-ELF1.bed", package="rtfbsdb");
bed.chipseq.chr19<- read.table(file.ELF1.chr19);

## build tfbs object by querying motifs in ELF1
tfs.elf1<- tfbs.createFromCisBP(db, tf_name="ELF1");

## Scan 2bit file within all bed regions to find motif binding sites
r2.scan <- tfbs.scanTFsite(tfs.elf1, file.twoBit.chr19, bed.chipseq.chr19, ncores = 3);

```

The function has two parameters to control how to select binding sites. Specifying a “*threshold*” will return motifs that exceed the log likelihood of the observed N-mer given the PWM minus the log likelihood of the N-mer under a third-order Markov background model. This option controls the specificity of motif discovery very well in most situations. In some cases, the default threshold (6) approximately corresponds to a 10% false discovery rate (FDR), although this varies depending on the DNA sequence composition and motif information content. Using a higher threshold results in a more stringent match to the motif of interest.

Alternatively users can specify a fixed FDR using the *fdr* option. This option simulates a set of sequences under a third-order Markov background model and estimates the motif threshold that satisfies the specified FDR. It takes considerably more time to estimate the FDR. If *fdr* is specified in the parameter of *threshold.type*, the parameter of *threshold* will be used as the *fdr* threshold.

Optionally, multithreaded processing is supported by setting the *ncores* parameter.

The *tfbs.scanTFsite* function returns a list object consisting of four parts:

- 1) *\$result*: the result of the motif scan. The format can be set using the *return.type* parameter. By default (*return.type*="matches") a BED-formatted data frame is returned. The option "*writedb*" writes a starch-compressed (using the *bedops* package) to disk with matches for each motif, and is useful for large searches resulting in millions of motif matches.
- 2) *\$summary*: a summary of TF scan, including the number of binding sites matched for each motif.
- 3) *\$parm*: the values of control parameters, such as *fdr*, *threshold*, *gc.groups*, *background.order*, *background.length*.
- 4) *\$bed*: the bed-formatted loci information with 6 columns.

The parameters and summary information can be printed out using the *show* command as follows:

```

# return.type="matches"
>r1.scan
Return type: matches
FDR threshold: NA
Score threshold: 6
Motifs count: 26
Binding sites: 34518
  TF Name    Motif ID Count
7 M2809_1.01 TFAP2C 2504
4 M5963_1.01 TFAP2B 2283
.....

```

```
# return.type="writedb"
>r1.scan
Return type: writedb
FDR threshold: NA
Score threshold: 6
Binary Bed file: scan.db.db.starch
```

The function *tfbs.reportFinding* can export a simple report to a PDF file which shows the motif logos and the number of binding sites discovered for each motif if the “matches” return type is specified.

## 2.8 Enrichment comparison between two case-control groups

The second goal of *rtfbsdb* is to identify motifs enriched in a user specified set of genomic coordinates compared to a background set. The *tfbs.enrichmentTest* function computes the number of motif occurrences in two sets of genomic coordinates and returns a p-value (Fisher’s exact), and other information. Two groups of genomic coordinates are specified as arguments to *tfbs.enrichmentTest* as the arguments ‘positive’ and ‘negative’. To maximize statistical power, sequence sets are typically no less than a few hundred sequences, and backgrounds can often be much larger (i.e., tens of thousands). For example:

```
## Specify the requisite files from pre-installed data
file.twoBit.chr19<- system.file("extdata","hg19.chr19.2bit", package="rtfbsdb")
file.ELF1.chr19<- system.file("extdata","Chipseq-k562-chr19-ELF1.bed",
  package="rtfbsdb");
file.bg.chr19<- system.file("extdata","k562.chr19.dnase.UW.DUKE.inters.bed",
  package="rtfbsdb");

## Convert the bed file for each condition to a data frame
bed.ELF1.chr19<- read.table(file.ELF1.chr19, header=FALSE);
bed.bg.chr19 <- read.table(file.bg.chr19, header=FALSE);

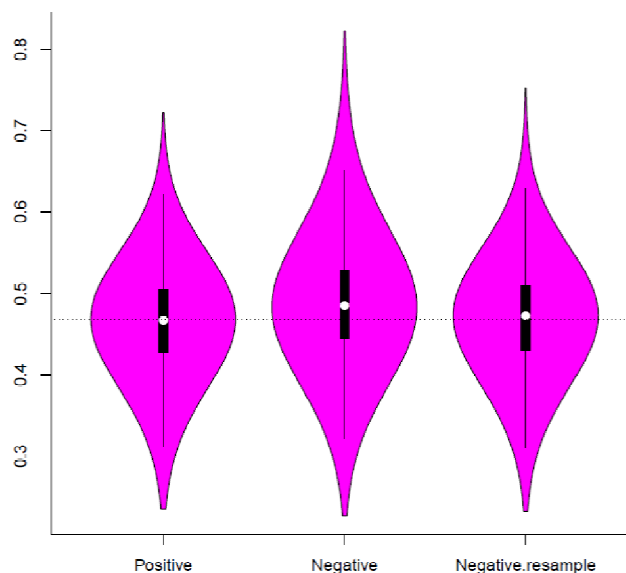
## build tfbs object by querying motifs in ELF1
tfs.elf1<- tfbs.createFromCisBP(db, tf_name="ELF1");

## Compare motifs between each condition
t.comp <- tfbs.enrichmentTest( tfs.elf1,
  file.twoBit.chr19,
  bed.ELF1.chr19,
  bed.bg.chr19,
  gc.correction=TRUE,
  gc.robust.rep=5,
  file.prefix="comp.db",
  ncores = 3);
```

The *rtfbsdb* package will find motifs that are enriched in a reference set (positive) relative to background (i.e. negative). If the background set can’t be specified, the function will stochastically extract the genomic loci adjacent to the positive set, for example:

```
## Compare motifs without background set
t.comp <- tfbs.enrichmentTest( tfs.elf1,
  file.twoBit.chr19,
  bed.ELF1.chr19,
  gc.correction=TRUE,
  gc.correction.pdf="elf1.gc.correction.pdf",
  gc.robust.rep=5,
  file.prefix="comp.db",
  ncores = 3);
```

Notably, this type of analysis is often confounded by systematic differences in the GC content between groups. To address this limitation, *tfbs.enrichmentTest* will check whether the mean of the GC content differs significantly between the two groups and shows a p-value (Wilcox test) and a Violin plot. If the p-value is significant, and the parameter of *gc.correction* is set to TRUE (default), the *tfbs.enrichmentTest* function will resample the background (i.e. negative) sequences stochastically to reduce the difference in GC content. The following is a Violin plot to demonstrate the effects of negative correction.



The output of *tfbs.enrichmentTest* includes a data frame with 8 columns, including motif ID, TF name, occurrence in positive group, expectation in negative group, enrichment ratio, p-value of fisher test, and correction value by multiple comparison methods (e.g., Bonferroni or FDR). The *show* command prints significant motifs and other meta data in the result object, as shown below.

```
> show(t.comp)
Negative correction: TRUE
p-value correction: BH
Significant p-value: 0.05
Threshold type : fdr
```

```

TF binding threshold: 0.05
TF binding background.order: 3
TF binding background.length: 1e+05
Total Motif: 336

```

Significant Motifs(or top 20):

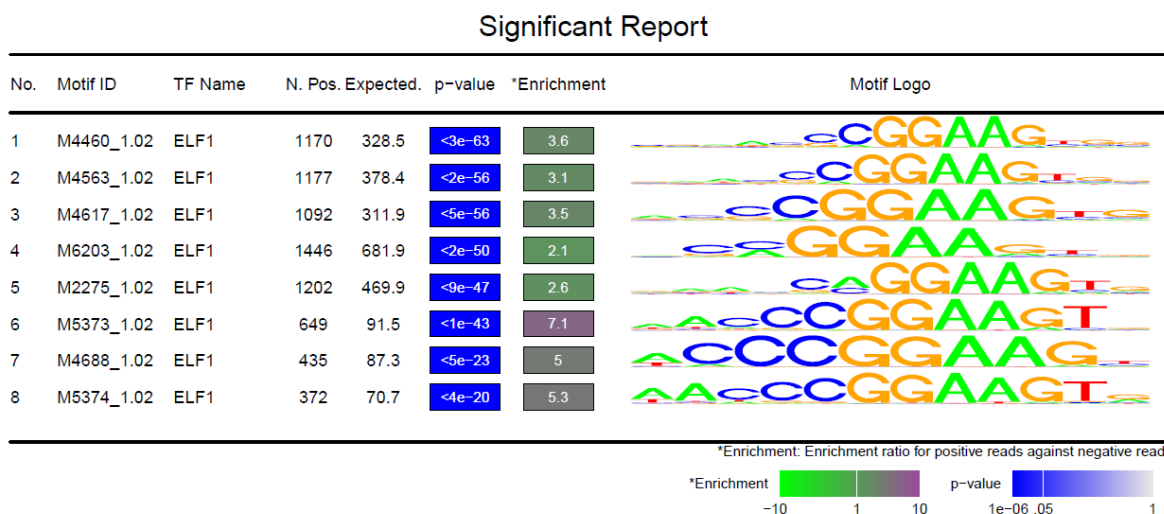
|       | motif.id   | tf.name | Npos | Nneg | pv.adj       | fold.enrichment |
|-------|------------|---------|------|------|--------------|-----------------|
| 165   | M3432_1.01 | MEIS1   | 64   | 740  | 1.230174e-05 | 2.194749        |
| 28    | M5440_1.01 | MEIS3   | 32   | 261  | 1.310005e-05 | 3.111330        |
| ..... |            |         |      |      |              |                 |

This object can be used to write a PDF report for the comparison. The `tfbs.reportEnrichment` function draws a motif list with visual p-value bar, enrichment ratio bar, and motif logos. The following command demonstrates how to print out the significant motifs for which adjusted p-values are less than 0.01. The screen shot of PDF report follows this example.

```

## Output the report for all significant motifs
tfbs.reportEnrichment(tfs, t.comp, file.pdf="test-tfcomp.pdf", report.title="Significant Report",
  sig.only=T, pv.threshold=0.01, pv.adj="fdr");

```



## 2.9 SessionInfo()

Session information of the R console used to write this vignette is shown below. It demonstrates the R packages necessary for successful *rtfbsdb* installation.

```

> sessionInfo()
R version 3.1.0 (2014-04-10)
Platform: x86_64-unknown-linux-gnu (64-bit)

locale:
 [1] LC_CTYPE=en_US.UTF-8  LC_NUMERIC=C

```

```
[3] LC_TIME=en_US.UTF-8    LC_COLLATE=en_US.UTF-8
[5] LC_MONETARY=en_US.UTF-8 LC_MESSAGES=en_US.UTF-8
[7] LC_PAPER=en_US.UTF-8    LC_NAME=C
[9] LC_ADDRESS=C            LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
```

attached base packages:

```
[1] stats    graphics grDevices utils    datasets methods base
```

other attached packages:

```
[1] rtfbsdb_0.1.8 vioplot_0.2 sm_2.2-5.4
```

loaded via a namespace (and not attached):

```
[1] apcluster_1.4.1    bigWig_0.2-9      cluster_2.0.3
[4] grid_3.1.0        lattice_0.20-33   latticeExtra_0.6-26
[7] Matrix_1.2-1      parallel_3.1.0    RColorBrewer_1.1-2
[10] Rcpp_0.12.0       rphast_1.6        rtfbs_0.3.4
```

### 3 Links

- (1) Cis-BP database: <http://cisbp.ccbbr.utoronto.ca/bulk.php>
- (2) Twobit files: <http://hgdownload-test.cse.ucsc.edu/goldenPath/>
- (3) Gencode files: <http://www.gencodegenes.org/>

### 4 References

- [1] Weirauch, M. T., Yang, A., Albu, M., Cote, A. G., Montenegro-Montero, A., Drewe, P., ... & Zheng, H. (2014). Determination and inference of eukaryotic transcription factor sequence specificity. *Cell*, 158(6), 1431-1443.
- [2] Mathelier, A., Zhao, X., Zhang, A. W., Parcy, F., Worsley-Hunt, R., Arenillas, D. J., ... & Lim, J. (2013). JASPAR 2014: an extensively expanded and updated open-access database of transcription factor binding profiles. *Nucleic acids research*, gkt997.
- [3] Shannon P. (2015). MotifDb: An Annotated Collection of Protein-DNA Binding Sequence Motifs. *R package version 1.12.1*.
- [4] Kulakovskiy, I. V., Medvedeva, Y. A., Schaefer, U., Kasianov, A. S., Vorontsov, I. E., Bajic, V. B., & Makeev, V. J. (2013). HOCOMOCO: a comprehensive collection of human transcription factor binding sites models. *Nucleic acids research*, 41(D1), D195-D202.
- [5] Neph, S., Kuehn, M. S., Reynolds, A. P., Haugen, E., Thurman, R. E., Johnson, A. K., ... & Sandstrom, R. (2012). BEDOPS: high-performance genomic feature operations. *Bioinformatics*, 28(14), 1919-1920.
- [6] Quinlan, A. R., & Hall, I. M. (2010). BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, 26(6), 841-842.
- [7] Core, L. J., Waterfall, J. J., & Lis, J. T. (2008). Nascent RNA sequencing reveals widespread pausing and divergent initiation at human promoters. *Science*, 322(5909), 1845-1848.
- [8] Bembom, O. (2014). seqLogo: Sequence logos for DNA sequence alignments. *R package version, 1(0)*.

- [9] Bodenhofer, U., Kothmeier, A., & Hochreiter, S. (2011). APCluster: an R package for affinity propagation clustering. *Bioinformatics*, 27(17), 2463-2464.
- [10] rtfbs: Transcription Factor Binding Site Identification Tool. *R package* version 0.3.4.