

# Package ‘enhancerHomologSearch’

June 30, 2022

**Type** Package

**Title** Identification of putative mammalian orthologs to given enhancer

**Version** 1.2.0

**Description** Get ENCODE data of enhancer region via H3K4me1 peaks and search homolog regions for given sequences. The candidates of enhancer homolog regions can be filtered by distance to target TSS. The top candidates from human and mouse will be aligned to each other and then exported as multiple alignments with given enhancer.

**BugReports** <https://github.com/jianhong/enhancerHomologSearch/issues>

**URL** <https://jianhong.github.io/enhancerHomologSearch>

**Depends** R (>= 4.1.0), methods

**Imports** BiocGenerics, Biostrings, BSgenome, BiocParallel, BiocFileCache, GenomeInfoDb, GenomicRanges, httr, IRanges, jsonlite, motifmatchr, Matrix, rtracklayer, Rcpp, S4Vectors, stats, utils

**Suggests** knitr, rmarkdown, BSgenome.Drerio.UCSC.danRer10, BSgenome.Hsapiens.UCSC.hg38, BSgenome.Mmusculus.UCSC.mm10, TxDb.Hsapiens.UCSC.hg38.knownGene, org.Hs.eg.db, TxDb.Mmusculus.UCSC.mm10.knownGene, org.Mm.eg.db, MotifDb, testthat, TFBSTools

**biocViews** Sequencing, GeneRegulation, Alignment

**License** GPL (>= 2)

**Encoding** UTF-8

**VignetteBuilder** knitr

**RoxygenNote** 7.1.2

**LinkingTo** Rcpp

**git\_url** <https://git.bioconductor.org/packages/enhancerHomologSearch>

**git\_branch** RELEASE\_3\_15

**git\_last\_commit** 45c23eb

**git\_last\_commit\_date** 2022-04-26

**Date/Publication** 2022-06-30

**Author** Jianhong Ou [aut, cre] (<<https://orcid.org/0000-0002-8652-2488>>),  
Valentina Cigliola [dct],  
Kenneth Poss [fnd]

**Maintainer** Jianhong Ou <jianhong.ou@duke.edu>

## R topics documented:

alignment . . . . .	2
alignmentOne . . . . .	4
conservedMotifs . . . . .	4
Enhancers-class . . . . .	5
getENCODEdata . . . . .	7
motifs . . . . .	8
queryEncode . . . . .	9
saveAlignments . . . . .	10
searchTFBPS . . . . .	10
<b>Index</b>	<b>12</b>

---

alignment	<i>Output</i>
-----------	---------------

---

## Description

Do pairwise alignment for query enhancer to target genome

## Usage

```
alignment(
  query,
  subject,
  method = c("ClustalW", "Muscle"),
  cluster = c("nj", "upgma", "upgmamax", "upgmamin", "upgmb"),
  substitutionMatrix = c("iub", "clustalw"),
  gapOpening = ifelse(method[1] == "ClustalW", 15, 400),
  gapExtension = ifelse(method[1] == "ClustalW", 6.66, 0),
  maxiters = ifelse(method[1] == "ClustalW", 3, 16),
  order = c("aligned", "input"),
  ...
)
```

**Arguments**

query	An object of DNASTringSet to represent enhancer
subject	An list of objects of <a href="#">Enhancers</a> .
method	specifies the multiple sequence alignment to be used; currently, "ClustalW", and "Muscle" are supported. Default is "Muscle"
cluster	The clustering method which should be used. Possible values are "nj" (default) and "upgma". In the original ClustalW implementation, this parameter is called clustering.
substitutionMatrix	substitution matrix for scoring matches and mismatches; The valid choices for this parameter are "iub" and "clustalw". In the original ClustalW implementation, this parameter is called matrix.
gapOpening	gap opening penalty; the default is 400 for DNA sequences and 420 for RNA sequences. The default for amino acid sequences depends on the profile score settings: for the setting le=TRUE, the default is 2.9, for sp=TRUE, the default is 1,439, and for sv=TRUE, the default is 300. Note that these defaults may not be suitable if custom substitution matrices are being used. In such a case, a sensible choice of gap penalties that fits well to the substitution matrix must be made.
gapExtension	gap extension penalty; the default is 0.
maxiters	maximum number of iterations; the default is 16.
order	how the sequences should be ordered in the output object; if "aligned" is chosen, the sequences are ordered in the way the multiple sequence alignment algorithm orders them. If "input" is chosen, the sequences in the output object are ordered in the same way as the input sequences.
...	Parameters can be used by Muscle, or ClustalW.

**Value**

An object of [Enhancers](#).

**Examples**

```
library(BSgenome.Hsapiens.UCSC.hg38)
library(BSgenome.Mmusculus.UCSC.mm10)
library(BSgenome.Drerio.UCSC.danRer10)
LEN <- GRanges("chr4", IRanges(19050041, 19051709))
seqEN <- getSeq(BSgenome.Drerio.UCSC.danRer10, LEN)
aln_hs <- readRDS(system.file("extdata", "aln_hs.rds",
                             package="enhancerHomologSearch"))
genome(aln_hs) <- Hsapiens
aln_mm <- readRDS(system.file("extdata", "aln_mm.rds",
                             package="enhancerHomologSearch"))
genome(aln_mm) <- Mmusculus
al <- alignment(seqEN, list(human=aln_hs, mouse=aln_mm),
               method="ClustalW", order="input")
```

---

alignmentOne                    *Get alignment scores*

---

### Description

Do pairwise alignment for query enhancer to target genome

### Usage

```
alignmentOne(query, subject, block = 1000, bpparam = bpparam(), ...)
```

### Arguments

query	An object of DNASTringSet to represent enhancer
subject	Output of getENCODEdata. An object of <a href="#">Enhancers</a>
block	The size of sequences to do alignment. Increase the size will increase the memory cost. Default 1000.
bpparam	BiocParallel parameters.
...	not used.

### Value

An object of [Enhancers](#).

### Examples

```
library(BiocParallel)
bpparam <- MulticoreParam(workers = 1, tasks=200, progressbar=TRUE)
library(BSgenome.Hsapiens.UCSC.hg38)
peaks <- GRanges("chr1", IRanges(seq(5000, 50000, by=1000), width=1000))
peaks$id <- paste(seq_along(peaks), 1, sep="_")
subj <- Enhancers(genome=Hsapiens, peaks=peaks)
q <- getSeq(Hsapiens, GRanges("chr1", IRanges(90000, width=1000)))
ao <- alignmentOne(q, subj, bpparam=bpparam)
```

---

conservedMotifs                    *check the conserved motifs in the orthologs*

---

### Description

Print the conserved motifs in the alignments

### Usage

```
conservedMotifs(aln, aln_hs, aln_mm, PWMs, queryGenome, background = "genome")
```

**Arguments**

<code>aln</code>	alignment of multiple DNAs. Output of <a href="#">alignment</a> function.
<code>aln_hs, aln_mm</code>	output of <code>linksearchTFBPS</code> for human and mouse.
<code>PWMs</code>	The Position Weight Matrix list represented as a numeric matrix. Object of <a href="#">PWMMatrixList</a> or <a href="#">PFMatrixList</a> .
<code>queryGenome</code>	An object of <a href="#">BSgenome</a> for query enhancer.
<code>background</code>	background nucleotide frequencies. Default is "genome". Refer <a href="#">matchMotifs</a> for details.

**Value**

A list of XStringviews.

**Examples**

```
library(BSgenome.Hsapiens.UCSC.hg38)
library(BSgenome.Mmusculus.UCSC.mm10)
library(BSgenome.Drerio.UCSC.danRer10)
LEN <- GRanges("chr4", IRanges(19050041, 19051709))
seqEN <- getSeq(BSgenome.Drerio.UCSC.danRer10, LEN)
aln_hs <- readRDS(system.file("extdata", "aln_hs.rds",
                             package="enhancerHomologSearch"))
genome(aln_hs) <- Hsapiens
aln_mm <- readRDS(system.file("extdata", "aln_mm.rds",
                             package="enhancerHomologSearch"))
genome(aln_mm) <- Mmusculus
al <- alignment(seqEN, list(human=aln_hs, mouse=aln_mm),
               method="ClustalW", order="input")
data(motifs)
conservedMotifs(al[[1]], aln_hs, aln_mm, motifs[["dist60"]], Drerio)
```

---

Enhancers-class	<i>Class "Enhancers"</i>
-----------------	--------------------------

---

**Description**

An object of class "Enhancers" represents the output of function `getENCODEdata`, which includes the sequences of enhancers and their genomic coordinates.

**Usage**

```
Enhancers(genome, peaks, TFBP, TFBP0)

## S4 method for signature 'Enhancers'
x$name

## S4 replacement method for signature 'Enhancers'
```

```
x$name <- value

## S4 method for signature 'Enhancers,ANY'
distance(x)

## S4 replacement method for signature 'Enhancers'
distance(x) <- value

## S4 method for signature 'Enhancers'
tfbp(x)

## S4 method for signature 'Enhancers'
query_tfbp(x)

## S4 method for signature 'Enhancers'
getSeq(x, ...)

## S4 method for signature 'Enhancers,ANY'
subsetByOverlaps(
  x,
  ranges,
  maxgap = -1L,
  minoverlap = 0L,
  type = c("any", "start", "end", "within", "equal"),
  invert = FALSE,
  ...
)

## S4 method for signature 'Enhancers'
subset(x, ...)

## S4 method for signature 'Enhancers'
seqinfo(x)

## S4 method for signature 'Enhancers'
genome(x)

## S4 replacement method for signature 'Enhancers'
genome(x) <- value

## S4 method for signature 'Enhancers'
peaks(x)

## S4 replacement method for signature 'Enhancers'
peaks(x) <- value

## S4 method for signature 'Enhancers'
show(object)
```

**Arguments**

genome	An object of <a href="#">BSgenome</a> .
peaks	An object of <a href="#">GRanges</a> .
TFBP	An object of <a href="#">IgcMatrix</a> .
TFBP0	An vector of logical. "Enhancers"
x	An object of "Enhancers"
name	Slot name.
value	The values.
...	parameters can be passed to upstream functions.
ranges, maxgap, minoverlap, type, invert	parameters used by <a href="#">subsetByOverlaps</a>
object	An object of "Enhancers"

**Value**

An object of Enhancers.

**Slots**

genome	An object of <a href="#">BSgenome</a> .
peaks	An object of <a href="#">GRanges</a> .
TFBP	An object of <a href="#">IgcMatrix</a> .
TFBP0	An vector of logical.

**Examples**

```
Enhancers()
```

---

```
getENCODEdata
```

*Download enhancer sequences from ENCODE*

---

**Description**

Query enhancer peak and extract sequences from ENCODE

**Usage**

```
getENCODEdata(
  genome,
  markers = "H3K4me1",
  window_size = 1000L,
  step = 50L,
  ...
)
```

**Arguments**

genome	An object of <a href="#">BSgenome</a> .
markers	Enhancer markers. Default 'H3K4me1'. For active enhancer, it can be set as c('H3K4me1', 'H3K27ac')
window_size, step	The size of windows and steps to split the peaks into small pieces. These parameter is used because the width of histone marker peaks are different sizes. Break the peaks into small pieces can increase the matching score and align the matching for different peaks into same size. The window_size is also be used for overlapping detection of multiple histone markers.
...	Parameters can be passed to <a href="#">queryEncode</a>

**Value**

An object of [Enhancers](#) with genome, and peaks. The peaks is an object of GRanges. The genome is an object of BSgenome.

**Examples**

```
library(BSgenome.Hsapiens.UCSC.hg38)
hs <- getENCODEdata(genome=Hsapiens,
                    partialMatch=c(biosample_summary="spinal cord"))
```

---

motifs

---

*Pre-clustered motifs from human and mouse*


---

**Description**

The data were extracted from MotifDb package (v 1.34.0) and grouped by motifStack package (v 1.37.2). The data were packaged as PFMatrixList object by TFBSTools (v 1.30.0)

**Usage**

```
data(motifs)
```

**Format**

a list of PFMatrixList. The names of the list is the group distance.

**Source**

MotifDb package. Source code for the data generation is in extdata folder

**Examples**

```
data(motifs)
names(motifs)
motifs[[1]]
```



---

queryEncode	<i>query data from ENCODE by predefined criteria</i>
-------------	--

---

## Description

Search ENCODE data by querying the ENCODE Portal using its REST API.

## Usage

```
queryEncode(
  exactMatch,
  partialMatch = character(0),
  API_url = "https://www.encodeproject.org/search/",
  ...
)
```

## Arguments

exactMatch	character. Exact-match keywords refer to search results that perfectly match all the keywords in the search query, exactly as entered. It is a named character vector. The names will be the keys and characters will be the values for search.
partialMatch	character. Partial-match refer to search results that contain the search query. It is a named character vector. The names will be the keys and characters will be the values for search. The value starting from '!' indicates logical negation(NOT). The value starting from '>', '>=', '<', '==', '<=' indicates string comparison.
API_url	character. The ENCODE REST API url.
...	Not used.

## Value

A list of search results

## Examples

```
res <- queryEncode(
  exactMatch=c(
    target.label="H3K4me1",
    replicates.library.biosample.donor.organism.scientific_name="Homo sapiens",
    assembly="GRCh38",
    assay_term_name="ChIP-seq"),
  partialMatch=c(biosample_summary="heart"))
```

saveAlignments            *output alignments*

---

### Description

Save enhancer homologs to file in phylip format.

### Usage

```
saveAlignments(al, output_folder = tempdir(), motifConsensus = NULL)
```

### Arguments

al                        output of [alignment](#).  
output\_folder    output folder.  
motifConsensus    Transcription factor binding consensus.

### Value

The I/O status.

### Examples

```
al <- readRDS(system.file("extdata", "al.rds",  
                          package="enhancerHomologSearch"))  
tmpfolder <- tempdir()  
library(MotifDb)  
motifs <- query(MotifDb, "JASPAR_CORE")  
consensus <- sapply(motifs, consensusString)  
consensus <- DNAStrngSet(gsub("\\?", "N", consensus))  
saveAlignments(al, output_folder=tmpfolder, motifConsensus=consensus)
```

---

searchTFBPS            *Transcription Factor Binding Pattern Similarity (TFBPS) search*

---

### Description

Search the TFBPs for query in subject.

### Usage

```
searchTFBPS(query, subject, PWMs, queryGenome, background = "genome", ...)
```

**Arguments**

query	An object of DNASTringSet to represent enhancer
subject	Output of getENCODEdata. An object of <a href="#">Enhancers</a>
PWMs	The Position Weight Matrix list represented as a numeric matrix. Object of <a href="#">PWMMatrixList</a> or <a href="#">PFMatrixList</a> .
queryGenome	An object of <a href="#">BSgenome</a> for query data.
background	background nucleotide frequencies. Default is "genome". Refer <a href="#">matchMotifs</a> for details.
...	Parameters will be passed to <a href="#">matchMotifs</a> except 'out' and 'genome'.

**Value**

An object of [Enhancers](#).

**Examples**

```
library(BSgenome.Hsapiens.UCSC.hg38)
peaks <- GRanges("chr1", IRanges(seq(5000, 50000, by=1000), width=1000))
peaks$id <- paste(seq_along(peaks), 1, sep="_")
subj <- Enhancers(genome=Hsapiens, peaks=peaks)
q <- getSeq(Hsapiens, GRanges("chr1", IRanges(90000, width=1000)))
data(motifs)
ao <- searchTFBPS(q, subj, motifs[["dist60"]], queryGenome=Hsapiens)
```

# Index

- \* **datasets**
  - motifs, 8
- \$, Enhancers-method (Enhancers-class), 5
- \$<- , Enhancers-method (Enhancers-class), 5
- alignment, 2, 5, 10
- alignmentOne, 4
- BSgenome, 5, 7, 8, 11
- coerce (Enhancers-class), 5
- coerce, Enhancers, GRanges-method (Enhancers-class), 5
- conservedMotifs, 4
- distance (Enhancers-class), 5
- distance, Enhancers, ANY-method (Enhancers-class), 5
- distance, Enhancers-method (Enhancers-class), 5
- distance<- (Enhancers-class), 5
- distance<- , Enhancers, ANY-method (Enhancers-class), 5
- distance<- , Enhancers-method (Enhancers-class), 5
- Enhancers, 3, 4, 8, 11
- Enhancers (Enhancers-class), 5
- Enhancers-class, 5
- genome (Enhancers-class), 5
- genome, Enhancers-method (Enhancers-class), 5
- genome<- (Enhancers-class), 5
- genome<- , Enhancers, BSgenome-method (Enhancers-class), 5
- genome<- , Enhancers-method (Enhancers-class), 5
- getENCODEdata, 5, 7
- getSeq, Enhancers-method (Enhancers-class), 5
- GRanges, 7
- lgCMatrix, 7
- matchMotifs, 5, 11
- motifs, 8
- peaks (Enhancers-class), 5
- peaks, Enhancers-method (Enhancers-class), 5
- peaks<- (Enhancers-class), 5
- peaks<- , Enhancers, GRanges-method (Enhancers-class), 5
- peaks<- , Enhancers-method (Enhancers-class), 5
- PFMatrixList, 5, 11
- PWMMatrixList, 5, 11
- query\_tfbp (Enhancers-class), 5
- query\_tfbp, Enhancers, ANY-method (Enhancers-class), 5
- query\_tfbp, Enhancers-method (Enhancers-class), 5
- queryEncode, 8, 9
- saveAlignments, 10
- searchTFBPS, 10
- seqinfo, Enhancers-method (Enhancers-class), 5
- show, Enhancers-method (Enhancers-class), 5
- subset, Enhancers-method (Enhancers-class), 5
- subsetByOverlaps, 7
- subsetByOverlaps, Enhancers, ANY-method (Enhancers-class), 5
- subsetByOverlaps, Enhancers-method (Enhancers-class), 5

tfbp (Enhancers-class), 5  
tfbp, Enhancers, ANY-method  
    (Enhancers-class), 5  
tfbp, Enhancers-method  
    (Enhancers-class), 5