

# BCRANK

April 20, 2011

---

BCRANKmatch-class    *Class "BCRANKmatch"*

---

## Description

Holds the [bcrank](#) score for one IUPAC consensus sequence. Several objects of this class are collected in a [BCRANKsearch-class](#) object

## Objects from the Class

Objects are not intended to be created directly but as a result from running [bcrank](#).

## Slots

**consensus:** consensus sequence in IUPAC coding

**bcrankScore:** bcrank score for the consensus

**matchVec:** vector with 0's (no match) and 1's (match) of same length as the ranked DNA sequences

## Methods

**consensus** signature(object = "BCRANKmatch"): Returns the consensus sequence.

**bcrankScore** signature(object = "BCRANKmatch"): Returns the bcrank score.

**matchVector** signature(object = "BCRANKmatch"): Returns a vector with 0's (no match) and 1's (match) of same length and order as the ranked DNA sequences.

## Author(s)

Adam Ameer, <adam.ameur@genpat.uu.se>

## See Also

[bcrank](#), [BCRANKsearch-class](#)

---

`BCRANKout`*BCRANK results for USF1 ChIP-chip data*

---

**Description**

Results from running `bcrank` on USF1 whole genome ChIP-chip data for the human liver cell line HepG2.

**Usage**

```
data(BCRANKout)
```

**Source**

Data from whole genome ChIP-chip experiments on human liver cell line HepG2. (Rada-Iglesias, A., et al. 2007)

**References**

Rada-Iglesias, A., et al. (2007) Whole-genome maps of USF1 and USF2 binding and histone H3 acetylation reveal new aspects of promoter structure and candidate genes for common human disorders. *Genome Research*, Accepted

**See Also**

[bcrank](#)

---

`bcrank`*BCRANK: predicting binding site consensus from ranked DNA sequences*

---

**Description**

This function implements an algorithm for detection of short DNA sequences that are overrepresented in some part of the list. Starting from some initial consensus DNA sequence coded in IUPAC symbols, the method uses a heuristic search to improve the consensus until a local optimum is found. Individual predicted binding sites can be reported by the function [matchingSites](#).

**Usage**

```
bcrank(fafile, startguesses=c(), restarts=10, length=10,  
       reorderings=500, silent=FALSE, plot.progress=FALSE,  
       do.search=TRUE, use.P1=FALSE, use.P2=TRUE, strip.desc=TRUE)
```

**Arguments**

<code>fastafile</code>	a ranked fasta file containing DNA sequences.
<code>startguesses</code>	a character vector with consensus sequences in IUPAC coding to be used as starting sequences in the search. If empty, random start guesses will be generated.
<code>restarts</code>	number restarts of the algorithm when using random start guesses.
<code>length</code>	length of random start guess.
<code>reorderings</code>	number of random reorderings of the DNA sequences performed when calculating score.
<code>silent</code>	reports progress status if FALSE.
<code>plot.progress</code>	if TRUE, the progress is displayed in a plot.
<code>do.search</code>	if FALSE, no search is performed. In that case the start guesses are assigned with scores and reported as results.
<code>use.P1</code>	Use penalty for bases other than A,C,G,T.
<code>use.P2</code>	Use penalty for motifs matching repetitive sequences.
<code>strip.desc</code>	An argument used by the function <code>readFASTA()</code> . It specifies whether or not the ">" marking at the beginning of description lines should be removed.

**Value**

The method returns an object of class `BCRANKresult-class`.

**Author(s)**

Adam Aneur, <adam.ameur@genpat.uu.se>

**References**

Aneur, A., Rada-Iglesias, A., Komorowski, J., Wadelius, C. Identification of candidate regulatory SNPs by combination of transcription factor binding site prediction, SNP genotyping and haploChIP. *Nucleic Acids Res*, 2009, 37(12):e85.

**See Also**

`matchingSites`, `BCRANKresult-class`

**Examples**

```
## Load example fasta file
fastaFile <- system.file("Exfiles/USF1_small.fa", package = "BCRANK")
## Run BCRANK
## Not run: BCRANKout <- bcrank(fastaFile, restarts=20)

## Show BCRANK results
toptable(BCRANKout)
## The top scoring result
topMotif <- toptable(BCRANKout,1)
## Plot BCRANK search path
plot(topMotif)
## Position Weight Matrix
pwm(topMotif, normalize=FALSE)
```

BCRANKresult-class *Class "BCRANKresult"*

---

### Description

Holds the results from running `bcrank`. Contains a number of `BCRANKsearch-class` object, one for each restart of the `bcrank` search.

### Slots

`fname`: the name of the fasta file used for running `bcrank`.  
`toplist`: a list of `BCRANKsearch-class` objects, ranked by their scores.  
`funCall`: the function call that was made to `bcrank`.  
`nrSeqs`: number of sequences in the fasta input file.  
`restarts`: number of restarts used in the `bcrank` search.

### Methods

`fname` signature (`object = "BCRANKmatch"`): Returns the fasta file name.  
`toplist` signature (`object = "BCRANKmatch"`, `i=NULL`): If `i` is `NULL`, returns a data frame containing consensus and score for the results for each restart of the `bcrank` search. Otherwise, the `i`'th `BCRANKsearch-class` object in the `toplist` is returned.

### Author(s)

Adam Ameer, <adam.ameur@genpat.uu.se>

### See Also

`bcrank`, `BCRANKsearch-class`,

---

BCRANKsearch-class *Class "BCRANKsearch"*

---

### Description

Holds the whole search path from a single `bcrank` run. Each individual search step is stored in a `BCRANKmatch-class` object. Several objects of this class are collected in a `BCRANKresult-class` object

### Objects from the Class

Objects are not intended to be created directly but as a result from running `bcrank`.

**Slots**

**searchPath**: a collection of `BCRANKmatch-class` objects, containing all bcrank search steps from a start guess to a locally optimal solution.

**final**: a `BCRANKmatch-class` object for the highest scoring consensus sequence (locally optimal solution) in this bcrank run.

**finalPWM**: position weight matrix for the highest scoring consensus sequence.

**finalNrMatch**: number of occurrences of the final consensus sequence in the fasta input file.

**nrIterations**: number of iterations required to move from the start guess to the final solution in this bcrank run.

**Methods**

**searchPath** signature(`object = "BCRANKsearch", i=NULL`): If `i` is `NULL`, returns a data frame containing consensus and score for the whole search path. Otherwise, the `i`'th `BCRANKmatch-class` object in the search path is returned.

**pwm** signature(`object = "BCRANKsearch", normalize=TRUE`): Returns the position weight matrix (pwm) for the highest scoring consensus in this bcrank run. Matrix positions are between 0 and 1 when `normalize` is `TRUE`. When `FALSE`, the number of matching sequences is reported.

**plot** signature(`x = "BCRANKsearch", y = "missing"`): A plot method for the search-Path.

**Author(s)**

Adam Ameer, <adam.ameur@genpat.uu.se>

**See Also**

[bcrank](#), [BCRANKmatch-class](#), [BCRANKresult-class](#)

---

matchingSites

*Report IUPAC consensus occurrences in a fasta file*

---

**Description**

This function reports all occurrences of a consensus sequence in a fasta file. It can be used to extract transcription factor binding sites predicted by BCRANK or other motif search methods.

**Usage**

```
matchingSites(fafile, motifSequence, revComp=TRUE, strip.desc=TRUE)
```

**Arguments**

`fafile` a ranked fasta file containing DNA sequences.

`motifSequence`

a character vector in IUPAC coding representing a DNA sequence.

`revComp`

set to `TRUE` if the reverse complement also be matched.

`strip.desc`

An argument used by the function `readFASTA()`. It specifies whether or not the ">" marking at the beginning of description lines should be removed.

**Value**

Returns a data frame with positions, strand and DNA sequence for the matching sites.

**Author(s)**

Adam Ameer, <adam.ameur@genpat.uu.se>

**References**

Ameer, A., Rada-Iglesias, A., Komorowski, J., Wadelius, C. Identification of candidate regulatory SNPs by combination of transcription factor binding site prediction, SNP genotyping and haploChIP. *Nucleic Acids Res*, 2009, 37(12):e85.

**See Also**

[bcrank](#)

# Index

## \*Topic classes

BCRANKmatch-class, 1  
BCRANKresult-class, 4  
BCRANKsearch-class, 4

## \*Topic datasets

BCRANKout, 2

## \*Topic methods

bcrank, 2  
matchingSites, 5

bcrank, 1, 2, 2, 4-6

BCRANKmatch (*BCRANKmatch-class*), 1

BCRANKmatch-class, 4, 5

BCRANKmatch-class, 1

BCRANKout, 2

BCRANKresult

(*BCRANKresult-class*), 4

BCRANKresult-class, 3-5

BCRANKresult-class, 4

bcrankScore (*BCRANKmatch-class*), 1

BCRANKsearch

(*BCRANKsearch-class*), 4

BCRANKsearch-class, 1, 4

BCRANKsearch-class, 4

consensus (*BCRANKmatch-class*), 1

fname (*BCRANKresult-class*), 4

matchingSites, 2, 3, 5

matchVector (*BCRANKmatch-class*), 1

plot, BCRANKsearch, missing-method

(*BCRANKsearch-class*), 4

pwm (*BCRANKsearch-class*), 4

readFASTA(), 3, 5

searchPath (*BCRANKsearch-class*), 4

toptable (*BCRANKresult-class*), 4