

# Package ‘ASEB’

January 23, 2026

**Title** Predict Acetylated Lysine Sites

**Version** 1.55.0

**Author** Likun Wang <wanglk@hsc.pku.edu.cn> and Tingting Li <ltt@hsc.pku.edu.cn>.

**Description**

ASEB is an R package to predict lysine sites that can be acetylated by a specific KAT-family.

**Maintainer** Likun Wang <wanglk@hsc.pku.edu.cn>

**Depends** R (>= 2.8.0), methods

**Imports** graphics, methods, utils

**License** GPL (>= 3)

**biocViews** Proteomics

**Collate** AllClasses.R AllGenerics.R SequenceInfo.R readSequence.R  
asebSites.R asebProteins.R drawStat.R drawEScurve.R

**LazyLoad** yes

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

**git\_branch** devel

**git\_last\_commit** 193569c

**git\_last\_commit\_date** 2025-10-29

**Repository** Bioconductor 3.23

**Date/Publication** 2026-01-23

## Contents

asebProteins . . . . .	2
asebSites . . . . .	3
drawEScurve . . . . .	5
drawStat . . . . .	6
readSequence . . . . .	7
SequenceInfo . . . . .	8

<b>Index</b>	<b>9</b>
--------------	----------

---

asebProteins

*prediction of all lysine sites on a specific protein that can be acetylated*


---

## Description

This function is used to predict all lysine sites on a specific protein that can be acetylated by a specific KAT-family.

## Usage

```
asebProteins(backgroundSites, predefinedSites, testProteins, outputFile=NULL, permutationTimes=10000)
## S4 method for signature 'character,character,character'
asebProteins(backgroundSites, predefinedSites, testProteins, outputFile=NULL, permutationTimes=10000)
## S4 method for signature 'SequenceInfo,SequenceInfo,SequenceInfo'
asebProteins(backgroundSites, predefinedSites, testProteins, outputFile=NULL, permutationTimes=10000)
```

## Arguments

**backgroundSites** [SequenceInfo](#) object or file name (character(1)) for background peptides set.

**predefinedSites** [SequenceInfo](#) object or file name (character(1)) for KAT special peptides set.

**testProteins** [SequenceInfo](#) object or file name (character(1)) for query Proteins set.

**outputFile** file name for output (character(1)).

**permutationTimes** permutation times (integer(1)), default and recommended: 10000.

## Details

This function is used to predict lysine sites that can be acetylated by a specific KAT-family. The whole process is similar with the GSEA method (permuting gene sets). Please see the references for details.

The first three arguments of method asebProteins can be [SequenceInfo](#) objects or file names. If these arguments are [SequenceInfo](#) objects, this method returns a list to the users besides an output file. Otherwise, this method processes the FASTA format files directly and outputs all results to a file. In this case, this method can process huge number of sites each time without loading any sequences to R workspace.

## Value

The output file contains enrichment scores and P-values for each query site. The asebProteins, SequenceInfo, SequenceInfo also returns a list contains two data.frame objects: results and curveInfo.

**results** contains enrichment scores and P-values for each query site.

**curveInfo** contains information for enrichment score curves.

**Note**

The acetylated lysine sites and their surrounding amino acids (8 on each side) are treated as acetylated peptides.

Example for peptides sequence : "KEHDDIFDKLKEAVKEE".

All input file should follow FASTA format.

**References**

Subramanian, A. et al. (2005) Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc Natl Acad Sci U S A*, **102**, 15545-15550.

Mootha, V.K. et al. (2003) PGC-1alpha-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nat Genet*, **34**, 267-273.

Guttman, M. et al. (2009) Chromatin signature reveals over a thousand highly conserved large non-coding RNAs in mammals. *Nature*, **458**, 223-227.

Li, T.T. et al. Characterization and prediction of lysine (K)-acetyl-transferase (KAT) specific acetylation sites. *Mol Cell Proteomics*, **in press**.

**See Also**

[SequenceInfo](#), [readSequence](#), [asebSites](#), [drawStat](#), [drawEScurve](#).

**Examples**

```
backgroundSites <- readSequence(system.file("extdata", "background_sites.fa", package="ASEB"))
prodefinedSites <- readSequence(system.file("extdata", "predefined_sites.fa", package="ASEB"))
testProteins <- readSequence(system.file("extdata", "proteins_to_test.fa", package="ASEB"))
resultList <- asebProteins(backgroundSites, prodefinedSites, testProteins, permutationTimes=100)
resultList$results[1:2,]
```

---

asebSites

*prediction of lysine sites that can be acetylated*

---

**Description**

This function is used to predict lysine sites that can be acetylated by a specific KAT-family.

**Usage**

```
asebSites(backgroundSites, prodefinedSites, testSites, outputFile=NULL, permutationTimes=10000)
## S4 method for signature 'character,character,character'
asebSites(backgroundSites, prodefinedSites, testSites, outputFile=NULL, permutationTimes=10000)
## S4 method for signature 'SequenceInfo,SequenceInfo,SequenceInfo'
asebSites(backgroundSites, prodefinedSites, testSites, outputFile=NULL, permutationTimes=10000)
```

## Arguments

backgroundSites	<a href="#">SequenceInfo</a> object or file name (character(1)) for background peptides set.
prodefinedSites	<a href="#">SequenceInfo</a> object or file name (character(1)) for KAT special peptides set.
testSites	<a href="#">SequenceInfo</a> object or file name (character(1)) for query peptides set.
outputFile	file name for output (character(1)).
permutationTimes	permutation times (integer(1)), default and recommended: 10000.

## Details

This function is used to predict lysine sites that can be acetylated by a specific KAT-family. It will give an enrichment score and a P-value for each query lysine site. The whole process is similar with the GSEA method (permuting gene sets). Please see the references for details.

The first three arguments of method asebSites can be [SequenceInfo](#) objects or file names. If these arguments are [SequenceInfo](#) objects, this method returns a list to the users besides an output file. Otherwise, this method processes the FASTA format files directly and outputs all results to a file. In this case, this method can process huge number of sites each time without loading any sequences to R workspace.

## Value

The output file contains enrichment scores and P-values for each query site. The asebSites, SequenceInfo, SequenceInfo, also returns a list contains two data.frame objects: results and curveInfo.

results	contains enrichment scores and P-values for each query site.
curveInfo	contains information for enrichment score curves.

## Note

The acetylated lysine sites and their surrounding amino acids (8 on each side) are treated as acetylated peptides.

Example for peptides sequence : "KEHDDIFDKLKEAVKEE".

All input file should follow FASTA format.

## References

- Subramanian, A. et al. (2005) Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc Natl Acad Sci U S A*, **102**, 15545-15550.
- Mootha, V.K. et al. (2003) PGC-1alpha-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nat Genet*, **34**, 267-273.
- Guttman, M. et al. (2009) Chromatin signature reveals over a thousand highly conserved large non-coding RNAs in mammals. *Nature*, **458**, 223-227.
- Li, T.T. et al. Characterization and prediction of lysine (K)-acetyl-transferase (KAT) specific acetylation sites. *Mol Cell Proteomics*, **in press**.

**See Also**

[SequenceInfo](#), [readSequence](#), [asebProteins](#), [drawStat](#), [drawEScurve](#).

**Examples**

```
backgroundSites <- readSequence(system.file("extdata", "background_sites.fa", package="ASEB"))
prodefinedSites <- readSequence(system.file("extdata", "predefined_sites.fa", package="ASEB"))
testSites <- readSequence(system.file("extdata", "sites_to_test.fa", package="ASEB"))
resultList <- asebSites(backgroundSites, prodefinedSites, testSites, permutationTimes=100)
resultList$results[1:2,]
```

---

drawEScurve

*draw Enriment Score curves for specific sites*


---

**Description**

This function is used to draw Enriment Score curves for specific sites.

**Usage**

```
drawEScurve(curveInfoDataFrame, sites=NULL, max_p_value=0.1, min_es=0.2, outputDir=NULL, figKind=c("pdf", "jpeg"))
## S4 method for signature 'data.frame'
drawEScurve(curveInfoDataFrame, sites=NULL, max_p_value=0.1, min_es=0.2, outputDir=NULL, figKind=c("pdf", "jpeg"))
```

**Arguments**

curveInfoDataFrame	
sites	data.frame object: contains curve information for sites, see example.
max_p_value	character vector: only draw curves for sites with ids appear in this vector. default: draw curves for all the sites appear in curveInfoDataFrame.
min_es	numeric(1), only draw curves for sites with p-value less than this value.
outputDir	numeric(1), only draw curves for sites with Enriment Score more than this value.
figKind	character(1), output directory name for all the figures.
	character(1), fig format: "pdf" or "jpeg".

**Details**

This function is used to draw Enrichment Score curves for specific sites. These curves show running-sum process for calculating enrichment score. The data.frame object contains curve information is given by [asebSites](#) or [asebProteins](#).

**See Also**

[SequenceInfo](#), [readSequence](#), [asebSites](#), [asebProteins](#), [drawStat](#).

## Examples

```
backgroundSites <- readSequence(system.file("extdata", "background_sites.fa", package="ASEB"))
prodefinedSites <- readSequence(system.file("extdata", "predefined_sites.fa", package="ASEB"))
testSites <- readSequence(system.file("extdata", "sites_to_test.fa", package="ASEB"))
resultList <- asebSites(backgroundSites, prodefinedSites, testSites, permutationTimes=100)
drawEScurve(resultList$curveInfo, max_p_value=0.1, min_es=0.1, outputDir=tempdir(), figKind="jpeg")
cat("see figures in output dir:", tempdir(),"\n")
```

---

drawStat	<i>draw P-values and enrichment scores for all lysine sites on a specific protein</i>
----------	---

---

## Description

This function is used to show P-values and enrichment scores for all lysine sites on a specific protein.

## Usage

```
drawStat(curveInfoDataFrame, proteinIds=NULL, outputDir=NULL, figKind=c("pdf","jpeg"))
## S4 method for signature 'data.frame'
drawStat(curveInfoDataFrame, proteinIds=NULL, outputDir=NULL, figKind=c("pdf","jpeg"))
```

## Arguments

curveInfoDataFrame	data.frame object: contains curve information for all proteins, see example.
proteinIds	character vector: only draw curves for proteins with ids appear in this vector. default: draw curves for all the proteins appear in curveInfoDataFrame.
outputDir	character(1), output directory name for all the figures.
figKind	character(1), fig format: "pdf" or "jpeg".

## Details

This function is used to draw P-values and enrichment scores for all lysine sites on a specific protein. The X-axis shows positions of all lysine sites on a specific protein, and Y-axis shows the enrichment scores (0~1) and P-values (0~1) for each lysine site. The data.frame object contains curve information is given by [asebProteins](#).

## See Also

[SequenceInfo](#), [readSequence](#), [asebSites](#), [asebProteins](#), [drawEScurve](#).

## Examples

```
backgroundSites <- readSequence(system.file("extdata", "background_sites.fa", package="ASEB"))
prodefinedSites <- readSequence(system.file("extdata", "predefined_sites.fa", package="ASEB"))
testProteins <- readSequence(system.file("extdata", "proteins_to_test.fa", package="ASEB"))
resultList <- asebProteins(backgroundSites, prodefinedSites, testProteins, permutationTimes=100)
#drawEScurve(resultList$curveInfo, max_p_value=0.5, min_es=0, outputDir=tempdir(), figKind="jpeg")
drawStat(resultList$curveInfo, outputDir=tempdir(), figKind="jpeg");
cat("see figures in output dir:", tempdir(),"\n")
```

---

readSequence	<i>read sequences from file</i>
--------------	---------------------------------

---

## Description

This function is used to read sequences from FASTA format file.

## Usage

```
readSequence(file)
```

## Arguments

file	character(1), file name for input. The input file should follow FASTA format.
------	--

## Details

This function return an object of [SequenceInfo](#) that contains sequences and identifiers from FASTA format input file.

## Value

A [SequenceInfo](#) object containing sequences and identifiers from input file.

## See Also

[SequenceInfo](#), [asebSites](#), [asebProteins](#), [drawStat](#), [drawEScurve](#).

## Examples

```
ff <- system.file("extdata", "background_sites.fa", package="ASEB")
readSequence(ff)
```

---

SequenceInfo	<i>"SequenceInfo" class</i>
--------------	-----------------------------

---

### Description

This class is used to store sequences and identifiers for lysine sites or proteins.

### Objects from the Class

Objects from this class are created by constructor SequenceInfo, as outlined below.

### Slots

sequences: "character" containing sequences.

id: "character" containing identifiers.

### Methods

Constructor:

**SequenceInfo** signature(sequences = "character", id = "character"): Create a SequenceInfo object from sequences and their identifiers. The length of id must match that of sequences.

### See Also

[readSequence](#), [asebSites](#), [asebProteins](#), [drawStat](#), [drawEScurve](#).

### Examples

```
showClass("SequenceInfo")
```



# Index

## \* **classes**

SequenceInfo, [8](#)

## \* **methods**

asebProteins, [2](#)

asebSites, [3](#)

drawEScurve, [5](#)

drawStat, [6](#)

readSequence, [7](#)

asebProteins, [2](#), [5–8](#)

asebProteins, character, character, character-method  
(asebProteins), [2](#)

asebProteins, SequenceInfo, SequenceInfo, SequenceInfo-method  
(asebProteins), [2](#)

asebSites, [3](#), [3](#), [5–8](#)

asebSites, character, character, character-method  
(asebSites), [3](#)

asebSites, SequenceInfo, SequenceInfo, SequenceInfo-method  
(asebSites), [3](#)

drawEScurve, [3](#), [5](#), [5](#), [6–8](#)

drawEScurve, data.frame-method  
(drawEScurve), [5](#)

drawStat, [3](#), [5](#), [6](#), [7](#), [8](#)

drawStat, data.frame-method (drawStat), [6](#)

readSequence, [3](#), [5](#), [6](#), [7](#), [8](#)

SequenceInfo, [2–7](#), [8](#)