

# Package ‘BASiNETEntropy’

January 20, 2025

**Title** Classification of RNA Sequences using Complex Network and Information Theory

**Version** 0.99.6

**Description** It makes the creation of networks from sequences of RNA, with this is done the abstraction of characteristics of these networks with a methodology of maximum entropy for the purpose of making a classification between the classes of the sequences. There are two data present in the 'BASiNET' package, ``mRNA", and ``ncRNA" with 10 sequences. These sequences were taken from the data set used in the article (LI, Aimin; ZHANG, Junying; ZHOU, Zhongyin, 2014) <[doi:10.1186/1471-2105-15-311](https://doi.org/10.1186/1471-2105-15-311)>, these sequences are used to run examples.

**License** GPL-3

**Encoding** UTF-8

**Depends** R (>= 4.1.0)

**Imports** igraph, Biostings, randomForest

**biocViews** Software, BiologicalQuestion, GenePrediction, FunctionalPrediction, Network, Classification

**RoxygenNote** 7.2.0

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Murilo Montanini Breve [aut] (<<https://orcid.org/0000-0002-5781-0480>>),  
Matheus Henrique Pimenta-Zanon [aut]  
(<<https://orcid.org/0000-0003-0698-6360>>),  
Fabricio Martins Lopes [aut, cre]  
(<<https://orcid.org/0000-0002-8786-3313>>)

**Maintainer** Fabricio Martins Lopes <[fabricio@utfpr.edu.br](mailto:fabricio@utfpr.edu.br)>

**Repository** CRAN

**Date/Publication** 2023-08-16 18:24:35 UTC

## Contents

classify	2
createedges	3
creatingDataframe	4
curveofentropy	4
entropy	5
filtering	5
matrixmultiplication	6
maxentropy	6
preprocessing	7
selectingEdges	7
training	8
<b>Index</b>	<b>9</b>

---

classify	<i>Performs the classification methodology using complex network and entropy theories</i>
----------	---

---

### Description

Given three or two distinct data sets, one of mRNA, one of lncRNA and one of sncRNA. The classification of the data is done from the structure of the networks formed by the sequences, that is filtered by an entropy methodology. After this is done, the classification starts.

### Usage

```
classify(
  mRNA,
  lncRNA,
  sncRNA = NULL,
  trainingResult,
  save_dataframe = NULL,
  save_model = NULL,
  predict_with_model = NULL
)
```

### Arguments

mRNA	Directory where the file .FASTA lies with the mRNA sequences
lncRNA	Directory where the file .FASTA lies with the lncRNA sequences
sncRNA	Directory where the file .FASTA lies with the sncRNA sequences (optional)
trainingResult	The result of the training, (three or two matrices)
save_dataframe	save when set, this parameter saves a .csv file with the features in the current directory. No file is created by default.

save\_model        save when set, this parameter saves a .rds file with the model in the current directory. No file is created by default.

predict\_with\_model        predict the input sequences with the previously generated model.

**Value**

Results

**Author(s)**

Murilo Montanini Breve

**Examples**

```
library(BASiNETEntropy)
arqSeqMRNA <- system.file("extdata", "mRNA.fasta", package = "BASiNETEntropy")
arqSeqLNCRNA <- system.file("extdata", "ncRNA.fasta", package = "BASiNETEntropy")
load(system.file("extdata", "trainingResult.RData", package = "BASiNETEntropy"))
r_classify <- classify(mRNA=arqSeqMRNA, lncRNA=arqSeqLNCRNA, trainingResult = trainingResult)
```

---

createedges

*Creates an untargeted graph from a biological sequence*

---

**Description**

A function that from a biological sequence generates a graph not addressed having as words vertices, this being able to have its size parameter set by the 'word' parameter. The connections between words depend of the 'step' parameter that indicates the next connection to be formed

**Usage**

```
createedges(sequence, word = 3, step = 1)
```

**Arguments**

sequence        It is a vector that represents the sequence

word            This integer parameter decides the size of the word that will be formed

step            It is the integer parameter that decides the step that will be taken to make a new connection

**Value**

Returns the array used to creates the edge list

**Author(s)**

Murilo Montanini Breve

creatingDataframe      *Creates a feature matrix using complex network topological measures*

---

**Description**

A function that from the complex network topological measures create the feature matrix.

**Usage**

```
creatingDataframe(measures, tamM, tamLNC, tamSNC)
```

**Arguments**

measures	The complex network topological measures
tamM	mRNA sequence size
tamLNC	lncRNA sequence size
tamSNC	snRNA sequence size

**Value**

Returns the feature matrix in scale 0-1

**Author(s)**

Murilo Montanini Breve

---

curveofentropy      *Creates an entropy curve*

---

**Description**

A function that from the entropy measures and threshold creates an entropy curve.

**Usage**

```
curveofentropy(H, threshold)
```

**Arguments**

H	The 'training' return for the entropy measures
threshold	The 'training' return for the threshold

**Value**

Returns a entropy curve

**Author(s)**

Murilo Montanini Breve

---

entropy                      *Calculates the entropy*

---

**Description**

A function that calculates the entropy

**Usage**

entropy(x)

**Arguments**

x                      The probabilities P0 and P1

**Value**

Returns the entropy

**Author(s)**

Murilo Montanini Breve

---

filtering                      *Filters the edges*

---

**Description**

A function that filters the edges after the maximum entropy is obtained

**Usage**

filtering(edgestoselect, edgestofilter)

**Arguments**

edgestoselect    The selected edges  
edgestofilter    The edges used to filter

**Value**

Returns the filtered edges

**Author(s)**

Murilo Montanini Breve

matrixmultiplication *Compares the matrices*

---

**Description**

A function that compares the matrices 'trainingResult' and the adjacency matrix to produce a filtered adjacency matrix.

**Usage**

```
matrixmultiplication(data, histodata)
```

**Arguments**

data	Adjacency matrix
histodata	'trainingResult' data

**Value**

Returns the filtered adjacency matrix

**Author(s)**

Murilo Montanini Breve

---

maxentropy *Calculates the maximum entropy*

---

**Description**

A function that calculates the maximum entropy

**Usage**

```
maxentropy(histogram)
```

**Arguments**

histogram	The histogram (used in 'training' function)
-----------	---

**Value**

Returns the maximum entropy

**Author(s)**

Murilo Montanini Breve

---

preprocessing	<i>Rescales the results between values from 0 to 1</i>
---------------	--

---

**Description**

Given the results the data is rescaled for values between 0 and 1, so that the length of the sequences does not influence the results. The rescaling of the sequences are made separately

**Usage**

```
preprocessing(datah, tamM, tamLNC, tamSNC)
```

**Arguments**

datah	Array with results numerics
tamM	Integer number of mRNA sequences
tamLNC	Integer number of lncRNA sequences
tamSNC	Integer number of sncRNA sequences

**Value**

Returns the array with the rescaled values

**Author(s)**

Murilo Montanini Breve

---

selectingEdges	<i>Selects the edges of the adjacency matrix</i>
----------------	--

---

**Description**

A function that selects the edges of the adjacency matrix

**Usage**

```
selectingEdges(MAX, data)
```

**Arguments**

MAX	The maximum entropy
data	The adjacency matrix

**Value**

Returns the selected edges of the adjacency matrix

**Author(s)**

Murilo Montanini Breve

---

training

*Trains the algorithm to select the edges that maximize the entropy*

---

**Description**

A function that trains the algorithm to select the edges that maximize the entropy

**Usage**

```
training(mRNA, lncRNA, sncRNA = NULL)
```

**Arguments**

mRNA	Directory where the file .FASTA lies with the mRNA sequences
lncRNA	Directory where the file .FASTA lies with the lncRNA sequences
sncRNA	Directory where the file .FASTA lies with the sncRNA sequences (optional)

**Value**

Returns the edge lists and the 'curveofentropy' function inputs

**Author(s)**

Murilo Montanini Breve



# Index

classify, 2  
createedges, 3  
creatingDataframe, 4  
curveofentropy, 4  
  
entropy, 5  
  
filtering, 5  
  
matrixmultiplication, 6  
maxentropy, 6  
  
preprocessing, 7  
  
selectingEdges, 7  
  
training, 8