Package ‘EncDNA’

October 12, 2022

Type    Package
Title   Encoding of Nucleotide Sequences into Numeric Feature Vectors
Version 1.0.2
Date    2019-05-28
Author  Prabina Kumar Meher
Maintainer Prabina Kumar Meher <meherprabin@yahoo.com>
Depends R(>= 3.3.0)
biocViews
Imports  Biostrings
LazyData TRUE

Description We describe fifteen different splice site sequence encoding schemes that have been used in earlier studies for mapping of splice site sequences into numeric feature vectors. These encoding schemes will also be helpful for transforming other nucleotide sequences into numeric forms, provided they are of equal length. These encoding schemes will help the computational biologist working in the field of classification (binary or multiclass) or prediction involving nucleic acid sequences of equal length.

License GPL (>= 2)
NeedsCompilation no
Repository CRAN
Date/Publication 2019-05-28 06:00:03 UTC

R topics documented:

- APR.Feature ................................................................. 2
- Bayes.Feature ............................................................ 3
- Density.Feature .......................................................... 4
- droso ................................................................. 5
- Maldoss.Feature ........................................................ 6
- MM1.Feature .............................................................. 7
- MM2.Feature .............................................................. 8
- MN.Fdtf.Feature ......................................................... 10
**APR.Feature**

Adjacent position relationship feature.

**Description**

This feature was proposed by Li *et al.* (2012). In fact this is similar to the PN.FDTF encoding scheme (Huang *et al.*, 2006). In this encoding, correlation between adjacent nucleotides are taken into account. For any nucleotide sequence with \( n \) nucleotides, every two consecutive positions between 1 and \( n \), i.e., \((1, 2), (2, 3), \ldots (n - 1, n)\) constitute an APR feature set. For each pair of positions, frequencies of 16 dinucleotides are first computed for both positive and negative dataset, and then the difference matrix is obtained by subtracting the \( 16 \times (n - 1) \) dinucleotide frequency matrix of positive set from that of negative set. The difference matrix is then be used for encoding of nucleotide sequences. In this encoding procedure each sequence with \( n \) nucleotides can be encoded into a vector of \((n - 1)\) numeric observations.

**Usage**

```r
APR.Feature(positive_class, negative_class, test_seq)
```

**Arguments**

- **positive_class**
  - Nucleotide sequence dataset of positive class, must be an object of class `DNAStringSet`.
- **negative_class**
  - Nucleotide sequence dataset of negative class, must be an object of class `DNAStringSet`.
- **test_seq**
  - Nucleotide sequences to be encoded into numeric feature vectors, must be an object of class `DNAStringSet`.

**Details**

The class `DNAStringSet` can be obtained by using the function `readDNAStringSet` available in `Biostrings` package Bioconductor. Here, the sequences must be supplied in FASTA format. Both positive and negative datasets are required for this encoding scheme.

**Value**

A numeric matrix of order \( m \times (n - 1) \), where \( m \) is the number of sequences in `test_seq` and \( n \) is the length of sequence.
Bayes.Feature

Author(s)
Prabina Kumar Meher, Indian Agricultural Statistics Research Institute, New Delhi-110012, INDIA

References

See Also
PN.Fdtf.Feature, WAM.Feature

Examples
data(droso)
positive <- droso$positive
negative <- droso$negative
test <- droso$test
pos <- positive[1:200]
neg <- negative[1:200]
tst <- test
enc <- APR.Feature(positive_class=pos, negative_class=neg, test_seq=tst)
enc

Bayes.Feature

Projecting nucleotide sequences into numeric feature vectors using Bayes kernel encoding approach.

Description
This sequence encoding technique was introduced by Zhang et al. (2006) for prediction of splice sites. In this encoding technique, positional frequencies of nucleotides are computed for both positive and negative datasets, which are then used for encoding of any nucleotide sequence of same length. Each sequence of length $L$ can be encoded into a numeric feature vector of length $2L$. Both positive and negative classes of sequences are required for sequence encoding.

Usage
Bayes.Feature(positive_class, negative_class, test_seq)

Arguments

positive_class Nucleotide sequence dataset of positive class, must be an object of class DNAStringSet.
negative_class Nucleotide sequence dataset of negative class, must be an object of class DNAStringSet.
test_seq Nucleotide sequences to be encoded into numeric feature vectors, must be an object of class DNAStringSet.
The class DNAStringSet can be obtained by using the function `readDNAStringSet` available in Biostrings package of Bioconductor. Here, the sequences must be supplied in FASTA format.

A numeric matrix of order $m \times 2n$, where $m$ is the number of sequences in `test_seq` and $n$ is the sequence length.

Prabina Kumar Meher, Indian Agricultural Statistics Research Institute, New Delhi-110012, INDIA


Examples

```r
data(droso)
positive <- droso$positive
negative <- droso$negative
test <- droso$test
pos <- positive[1:200]
neg <- negative[1:200]
tst <- testenc <- Bayes.Feature(positive_class=pos, negative_class=neg, test_seq=tst)enc
```

Each nucleotide sequence is encoded into a numeric vector of same length based on the distribution of nucleotides over the sequence. Here, two classes of dataset are not required for encoding, and each sequence is independently encoded instead. This encoding scheme was introduced by Wei et al. (2013) for prediction of donor and acceptor human splice sites along with the `MM1.Feature`.

Usage

```r
Density.Feature(test_seq)
```
Arguments

test_seq Sequence dataset to be encoded, must be an object of class DNAStringSet.

Details

The class DNAStringSet can be obtained by reading FASTA sequences using the function readDNAS-tringSet available in Biostrings package of Bioconductor.

Value

A numeric matrix of order $m \times n$, where $m$ is the number of sequences in test_seq and $n$ is the length of sequence.

Author(s)

Prabina Kumar Meher, Indian Agricultural Statistics Research Institute, New Delhi-110012, INDIA

References


Examples

data(droso)
test <- droso$test
tst <- test[1:5]
enc <- Density.Feature(test_seq=tst)
enc

Description

An example dataset consisting of true and false donor splice sites of Drosophila melanogaster.

Usage

data("droso")

Format

The dataset belongs to the class DNAStringSet.
Source

This is a sample dataset which was collected from the dataset originally developed by Reese et al. (1997). The dataset can also be accessed at http://www.fruitfly.org/sequence/drosophila-datasets.html.

References


Examples

data(droso)

Maldoss.Feature  Encoding of nucleic acid sequences using di-nucleotide frequency difference between positive and negative class datasets.

Description

In Maldoss (Meher et al., 2016), the authors propose three encoding approaches namely P1, P2 and P3. Out of these three encoding schemes, the accuracies were reported to be higher for P1 as compared to the other two encoding procedures. Here, we describe the sequence encoding based on P1 only. This P1 encoding approach has similarity with that of PN-FDTF encoding (Huang et al., 2006) approach. The difference is only with respect to the logarithmic transformation in case of Maldoss.Feature. In this encoding procedure, both positive and negative class sequences are required for transformation of nucleotide sequences into numeric vectors.

Usage

Maldoss.Feature(positive_class, negative_class, test_seq)

Arguments

positive_class  Sequence dataset of the positive class, must be an object of class DNAStringSet.

negative_class  Sequence dataset of the negative class, must be an object of class DNAStringSet.

test_seq  Sequences to be encoded into numeric vectors, must be an object of class DNAStringSet.

Details

For getting an object of class DNAStringSet, the FASTA sequence dataset must be read in R through the function readDNAStringSet available in Biostrings package of Bioconductor (https://bioconductor.org/packages/release/bioc/html/Biostrings.html).

Value

A numeric matrix of order $m \times (n - 1)$, where $m$ is the number of sequences in test_seq and $n$ is the length of sequence.
Author(s)
Prabina Kumar Meher, Indian Agricultural Statistics Research Institute, New Delhi-110012, INDIA

References

See Also
PN.Fdtf.Feature, MM1.Feature, WAM.Feature

Examples
```r
data(droso)
pos <- droso$positive
.neg <- droso$negative
.test <- droso$test
.pos <- positive[1:200]
.neg <- negative[1:200]
.tst <- .test
.enc <- Maldoss.Feature(positive_class=pos, negative_class=neg, test_seq=tst)
.enc
```

Transforming nucleotide sequences into numeric vectors using first order nucleotide dependency.

Description
The concept of sequence encoding using Markov model (1st order) was introduced by Ho and Rajapakse (2005) for prediction of splice sites. However, this encoding scheme has been comprehensively used by Baten et al. (2006) for prediction of splice sites. In this encoding procedure, first order dependencies between nucleotides in nucleotide sequence are accounted. Only the positive class dataset is used for estimation of dependencies in terms of probabilities, which are then used for encoding.

Usage
```r
MM1.Feature(positive_class, test_seq)
```

Arguments
- `positive_class`: Sequence dataset of the positive class, must be an object of class `DNAStringSet`
- `test_seq`: Sequences to be encoded into numeric vectors, must be an object of class `DNAStringSet`
Details

The FASTA sequences should be read into R using the function `readDNAStringSet` available in the `Biostrings` package. This encoding is similar to PN.FDTF feature, as far as the dependency among nucleotides in a sequence is concerned. The only difference is the use of positive class only in stead of both positive and negative classes in PN.FDTF. This encoding approach has similarity with WAM features (Meher et al. 2016) in which the dinucleotide dependencies are considered.

Value

A numeric matrix of order $m \times (n - 1)$, where $m$ is the number of sequences in `test_seq` and $n$ is the length of sequence.

Author(s)

Prabina Kumar Meher, Indian Agricultural Statistics Research Institute, New Delhi-110012, INDIA

References


See Also

PN.Fdtf.Feature, WAM.Feature

Examples

```r
data(droso)
positive <- droso$positive
test <- droso$test
pos <- positive[1:200]
tst <- test
enc <- MM1.Feature(positive_class=pos, test_seq=tst)
enc
```

---

**MM2.Feature**

*Mapping nucleotide sequences onto numeric feature vectors based on second order nucleotide dependencies.*
**Description**
This encoding procedure is similar to the MM1 encoding. The only difference is consideration of second order dependencies unlike first order in MM2. Feature. This technique was first conceptualized by Rajapakse and Ho (2005), and adopted by Maji and Garg (2014). The number of parameters to be estimated in MM2 is 64, which is higher than that of MM1 i.e., 16. Further, only the positive class dataset is used for encoding of sequences.

**Usage**

```r
MM2.Feature(positive_class, test_seq)
```

**Arguments**

- `positive_class`  Sequence dataset of the positive class, must be an object of class `DNAStringSet`.
- `test_seq`  Sequences to be encoded into numeric vectors, must be an object of class `DNAStringSet`.

**Details**
For getting an object of class `DNAStringSet`, the FASTA sequences should be read using the function `readDNAStringSet` available in the `Biostrings` package.

**Value**
A numeric matrix of order \(m \times (n - 2)\), where \(m\) is the number of sequences in `test_seq` and \(n\) is the length of sequence.

**Author(s)**
Prabina Kumar Meher, Indian Agricultural Statistics Research Institute, New Delhi-110012, INDIA

**References**


**See Also**

- `MM1.Feature`, `WAM.Feature`

**Examples**

```r
data(droso)
positive <- droso$positive
test <- droso$test
pos <- positive[1:200]
tst <- test
enc <- MM2.Feature(positive_class=pos, test_seq=tst)
enc
```

MN.Fdtf.Feature

Sequence encoding with nucleotide frequency difference between two classes of sequence datasets.

Description

In this encoding procedure, at first, frequency of each nucleotide at each position is computed for both positive and negative classes datasets. Then, the frequency matrix of the positive set is substracted from that of negative set. The sequences are then encoded into numeric vectors after passing them through this difference matrix. So, both positive and negative datasets are necessary for encoding of sequences. This concept was introduced by Huang et al. (2006), and was also used by Pashaei et al. (2016) to generate features for prediction of splice sites along with other features. This has similarity with Bayes kernel encoding (Zhang et al., 2006), where both frequency matrices are used for encoding instead of the difference matrix.

Usage

MN.Fdtf.Feature(positive_class, negative_class, test_seq)

Arguments

positive_class  Sequence dataset of the positive class, must be an object of class DNAStringSet.
negative_class  Sequence dataset of the negative class, must be an object of class DNAStringSet.
test_seq       Sequences to be encoded into numeric vectors, must be an object of class DNAStringSet.

Details

For getting an object of class DNAStringSet, the sequence dataset must be read in FASTA format through the function readDNAStringSet available in the Biostrings package of Bioconductor (https://bioconductor.org/packages/release/bioc/html/Biostrings.html).

Value

A numeric matrix of order $m \times n$, where $m$ is the number of sequences in test_seq and $n$ is the sequence length.

Note

This feature does not take into consideration the dependencies among nucleotides in the sequence.

Author(s)

Prabina Kumar Meher, Indian Agricultural Statistics Research Institute, New Delhi-110012, INDIA
References


See Also

`WMM.Feature, Bayes.Feature, PN.Fdtf.Feature`

Examples

```r
data(droso)
positive <- droso$positive
negative <- droso$negative
test <- droso$test
pos <- positive[1:200]
neg <- negative[1:200]
tst <- test
c <- MN.Fdtf.Feature(positive_class=pos, negative_class=neg, test_seq=tst)
c
```

---

**PN.Fdtf.Feature**

Conversion of nucleotide sequences into numeric feature vectors based on the difference of dinucleotide frequency.

**Description**

Dinucleotide frequency matrix is first computed for both positive and negative classes. Then, frequency matrix of the positive class is subtracted from that of negative class. The sequences are then passed through this difference matrix to encode them into numeric feature vectors. Similar to the `MN.Fdtf.feature`, both positive and negative classes are necessary for encoding of nucleotide sequences. This was also conceptualized by Huang *et al.* (2006). This has also been used by Pashaei *et al.* (2016) as one of the features for prediction of splice sites along with the other features.

**Usage**

```r
PN.Fdtf.Feature(positive_class, negative_class, test_seq)
```

**Arguments**

- `positive_class` Sequence dataset of the positive class, must be an object of class `DNAStringSet`.
- `negative_class` Sequence dataset of the negative class, must be an object of class `DNAStringSet`.
- `test_seq` Sequences to be encoded into numeric vectors, must be an object of class `DNAStringSet`. 
Details

For getting an object of class DNAStringSet, the sequence dataset must be read in FASTA format through the function `readDNAStringSet` available in `Biostings` package of Bioconductor (https://bioconductor.org/packages/release/bioc/html/Biostings.html).

Value

A numeric matrix of order $m \times (n - 1)$, where $m$ is the number of sequences in `test_seq` and $n$ is the sequence length.

Note

Both positive and negative classes datasets are essential for the encoding. This feature has similarity with that of `MM1.Feature` and `WAM.Feature` with respect to the first order dependency. Unlike `MN.Fdtf.Feature`, this feature takes into account the first order dependencies of nucleotides in the sequence.

Author(s)

Prabina Kumar Meher, Indian Agricultural Statistics Research Institute, New Delhi-110012, INDIA

References


See Also

`MN.Fdtf.Feature, WAM.Feature, MM1.Feature`

Examples

data(droso)
positive <- droso$positive
negative <- droso$negative
test <- droso$test
pos <- positive[1:200]
neg <- negative[1:200]
tst <- test
enc <- PN.Fdtf.Feature(positive_class=pos, negative_class=neg, test_seq=tst)
enc
Description

This encoding scheme was devised by Li et al. (2012). Frequencies of 4 nucleotides are first computed at each position for both positive and negative datasets, resulting in two $4 \times L$ probability tables for the two classes for sequence length $L$. A $4 \times L$ statistical difference table is obtained by elementwise substraction of the two probability distribution tables, which is then used for encoding of sequences. Further, as per sparse encoding, the nucleotides A, T, G and C can be encoded as $(1,0,0,0)$, $(0,1,0,0)$, $(0,0,1,0)$ and $(0,0,0,1)$ respectively. The value 1 of sparse encoding is then replaced with the difference values obtained from the difference table for encoding nucleotide at each postion. Thus, it can be said that POS feature encoding is a blending of MN-FDTF (Huang et al., 2006) and Sparse encoding (Meher et al., 2016) technique.

Usage

```r
POS.Feature(positive_class, negative_class, test_seq)
```

Arguments

- **positive_class**: Sequence dataset of the positive class, must be an object of class `DNAStringSet`.
- **negative_class**: Sequence dataset of the negative class, must be an object of class `DNAStringSet`.
- **test_seq**: Sequences to be encoded into numeric vectors, must be an object of class `DNAStringSet`.

Details

The `DNAStringSet` object can be obtained by reading the sequences in FASTA format using the function `readDNAStringSet` available in the `Biostrings` package of Bioconductor.

Value

A numeric matrix of order $m \times 4n$, where $m$ is the number of sequences in `test_seq` and $n$ is the length of sequence.

Note

In this encoding procedure, dependencies of nucleotides are not taken into consideration. Both positive and negative datasets are required for encoding of nucleotide sequences. Each sequence of length $L$ can be transformed into a numeric vector of length $4 \times L$ with this encoding technique.

Author(s)

Prabina Kumar Meher, Indian Agricultural Statistics Research Institute, New Delhi-110012, INDIA
References


See Also

*MN.Fdtf.Feature*, *Bayes.Feature*, *WMM.Feature*

Examples

```r
data(droso)
positive <- droso$positive
negative <- droso$negative
test <- droso$test
pos <- positive[1:200]
eg <- negative[1:200]
tst <- test
enc <- POS.Feature(positive_class=pos, negative_class=neg, test_seq=tst)
enc
```

**Predoss.Feature**

*Encoding nucleotide sequences using all possible di-nucleotide dependencies.*

**Description**

In this encoding, not only the adjacent dependencies are considered, but also the association that exists among non-adjacent nucleotides. In MM1, PN.FDTF features, only the dependencies between adjacent nucleotides are taken into account. Though all possible pair-wise dependencies are first introduced by Meher et al. (2014) for predicting splice sites through probabilistic approach, the same authors further used this association to encode the splice site dataset for prediction using machine learning classifiers (Meher et al., 2016).

**Usage**

```r
Predoss.Feature(positive_class, negative_class, test_seq)
```

**Arguments**

- `positive_class` Sequence dataset of the positive class, must be an object of class DNAStringSet.
- `negative_class` Sequence dataset of the negative class, must be an object of class DNAStringSet.
- `test_seq` Sequences to be encoded into numeric vectors, must be of an object of class DNAStringSet.
Details

This encoding approach will be helpful for transformation of nucleotide sequences into numeric feature vectors, which can subsequently be used as input in several supervised learning models for classification.

Value

A numeric matrix of order $m \times n^2$, where $m$ is the number of sequences in \texttt{test_seq} and $n$ is the length of sequence.

Note

Dimension of the feature space will increase geometrically with increase in the length of the sequence.

Author(s)

Prabina Kumar Meher, Indian Agricultural Statistics Research Institute, New Delhi-110012, INDIA

References


Examples

data(droso)
positive <- droso$positive
negative <- droso$negative
test <- droso$test
pos <- positive[1:200]
neg <- negative[1:200]
tst <- test
enc <- Predoss.Feature(positive_class=pos, negative_class=neg, test_seq=tst)
enc

\begin{table}[h]
\centering
\begin{tabular}{ll}
\hline
\textbf{SAE.Feature} & \textit{Encoding of nucleotide sequences based on sum of absolute error (SAE) of each sequence.} \\
\hline
\end{tabular}
\end{table}
Description

The sum of absolute error (SAE) concept was introduced by Meher et al. (2014) for prediction of donor splice sites, and was subsequently used by the same authors (Meher et al., 2016) for encoding of splice site motif for prediction using supervised learning model. In this encoding technique also all possible pair-wise nucleotide dependencies are considered.

Usage

`SAE.Feature(positive_class, negative_class, test_seq)`

Arguments

- `positive_class`  Sequence dataset of the positive class, must be an object of class `DNAStringSet`.
- `negative_class`  Sequence dataset of the negative class, must be an object of class `DNAStringSet`.
- `test_seq`        Sequences to be encoded into numeric vectors, must be an object of class `DNAStringSet`.

Details

In this encoding approach a vector of two observations will be obtained for each sequence. This two values correspond to the values obtained, when only positive class and both positive & negative datasets are used for encoding. This encoding scheme is invariant to the length of the sequence. Thus, both positive and negative classes datasets are required for encoding of sequence.

Value

A numeric matrix of order \( m \times 2 \), where \( m \) is the number of sequences in `test_seq`.

Author(s)

Prabina Kumar Meher, Indian Agricultural Statistics Research Institute, New Delhi-110012, INDIA

References


See Also

`MM1.Feature`, `WAM.Feature`
Examples

```r
data(droso)
positive <- droso$positive
negative <- droso$negative
test <- droso$test
pos <- positive[1:200]
neg <- negative[1:200]
tst <- test
enc <- SAE.Feature(positive_class=pos, negative_class=neg, test_seq=tst)
enc
```

Sparse.Feature

Nucleotide sequence encoding with 0 and 1.

Description

In this encoding approach A, T, G and C are encoded as (1,1,1), (1,0,0), (0,1,0) and (0,0,1). This was introduced by Golam Bari et al. (2014). Besides, each nucleotide can also be encoded with four bits i.e., A as (1,0,0,0), T as (0,1,0,0), G as (0,0,1,0) and C as (0,0,0,1) as followed in Meher et al. (2016).

Usage

`Sparse.Feature(test_seq)`

Arguments

- **test_seq**: Sequence dataset to be encoded into numeric vector containing 0 and 1, must be an object of class `DNAStringSet`.

Details

Each sequence is encoded independently, without the need of positive and negative classes datasets.

Value

A vector of length $4 \times n$ for sequence of $n$ nucleotides long in `test_seq`.

Note

For larger sequence length, high dimensional feature vector will be generated.

Author(s)

Prabina Kumar Meher, Indian Agricultural Statistics Research Institute, New Delhi-110012, INDIA
References


Examples

```r
data(droso)
test <- droso$test
tst <- test
enc <- Sparse.Feature(test_seq=tst)
enc
```

**Trint.Dist.Feature**

Tri-nucleotide distribution-based encoding of nucleotide sequences.

**Description**

This encoding scheme was first time adopted by Wei *et al.* (2013) for prediction of splice sites along with MM1 features. In this encoding technique, distribution of trinucleotides are taken into consideration independently for the exon and intron regions of splice site motifs.

**Usage**

`Trint.Dist.Feature(test_seq)`

**Arguments**

- `test_seq` 
  Sequence dataset to be transformed into numeric feature vectors. There should be at least two sequences, must be an object of class `DNASTringSet`.

**Details**

This encoding scheme is independent of positive and negative datasets. In other words, each sequence can be encoded independently. Further, nucleotide sequence of any length will be transformed into a numeric vector of 64 observations corresponding to 64 combinations of trinucleotides.

**Value**

A numeric matrix of order \(m \times 64\), where \(m\) is the number of sequences in `test_seq`.

**Author(s)**

Prabina Kumar Meher, Indian Agricultural Statistics Research Institute, New Delhi-110012, INDIA
References


Examples

```r
data(droso)
test <- droso$test
tst <- test
enc <- Trint.Dist.Feature(test_seq=tst)
enc
```

Description

Unlike weighted matrix method (WMM), first order nucleotide dependencies are accounted in weighted array model (WAM). The WAM was introduced by Zhang and Marr (1993) for locating splicing signal on nucleotide sequences. The WAM was employed by Meher et al. (2016) for encoding of splice site motifs.

Usage

```r
WAM.Feature(positive_class, negative_class, test_seq)
```

Arguments

- `positive_class`  Sequence dataset of the positive class, must be an object of class `DNAStringSet`.
- `negative_class`  Sequence dataset of the negative class, must be an object of class `DNAStringSet`.
- `test_seq`        Sequences to be encoded into numeric vectors, must be an object of class `DNAStringSet`.

Details

In this encoding approach, a vector of two observations will be obtained for each sequence, corresponds to the situation when only positive class and both positive & negative datasets are used for encoding. This encoding scheme is also invariant to the length of the sequence.

Value

A numeric matrix of order $m \times 2$, where $m$ is the number of sequences in `test_seq`.

Author(s)

Prabina Kumar Meher, Indian Agricultural Statistics Research Institute, New Delhi-110012, INDIA
References


See Also

`MM1.Feature`, `PN.Fdtf.Feature`

Examples

```r
data(droso)
pd <- droso$positive
nd <- droso$negative
tst <- droso$tst
pos <- pd[1:200]
lat <- nd[1:200]
tst <- tst
enc <- WMM.Feature(positive_class=pos, negative_class=lat, test_seq=tst)
enc
```

---

**WMM.Feature**

Weighted matrix model based mapping of nucleotide sequences into vectors of numeric observations.

**Description**

The weighted matrix model (WMM) was developed by Staden (1984) for prediction of splice sites. In this technique, position weight matrix are computed from the aligned positive dataset which is then used for computing the likelihood of any nucleotide sequence being a positive or negative class. The position weight matrix was later on used by Meher et al. (2016) for encoding of splice site motifs for prediction using supervised learning models.

**Usage**

`WMM.Feature(positive_class, negative_class, test_seq)`

**Arguments**

- `positive_class`  Sequence dataset of the positive class, must be an object of class `DNAStringSet`.
- `negative_class`  Sequence dataset of the negative class, must be an object of class `DNAStringSet`.
- `test_seq`        Sequences to be encoded into numeric vector, must be an object of class `DNAStringSet`.
WMM.Feature

Details

In this encoding approach, a vector of two observations will be obtained for each sequence, corresponds to the situation when only positive class and both positive and negative datasets are used for encoding. This encoding scheme is also invariant to the length of the sequence.

Value

A numeric matrix of order \( m \times 2 \), where \( m \) is the number of sequences in \( \text{test}_\text{seq} \).

Author(s)

Prabina Kumar Meher, Indian Agricultural Statistics Research Institute, New Delhi-110012, INDIA

References


See Also

Bayes.Feature, MN.Fdtf.Feature

Examples

data(droso)
positive <- droso$positive
negative <- droso$negative
test <- droso$test
pos <- positive[1:200]
neg <- negative[1:200]
tst <- test
enc <- WMM.Feature(positive_class=pos, negative_class=neg, test_seq=tst)
enc
Index

* All possible pair-wise dependency
  SAE.Feature, 15
* Binary classification
  Predoss.Feature, 14
* Binary encoding
  Sparse.Feature, 17
* Dinucleotide dependency
  WAM.Feature, 19
* Dinucleotide frequency
  APR.Feature, 2
  Maldoss.Feature, 6
  PN.Fdtf.Feature, 11
* First order Markov model
  MM1.Feature, 7
* Multiclass classification
  Predoss.Feature, 14
* Nucleotide dependencies
  PN.Fdtf.Feature, 11
  Predoss.Feature, 14
* Nucleotide dependency
  APR.Feature, 2
  MM1.Feature, 7
  MM2.Feature, 8
* Nucleotide frequency
  MN.Fdtf.Feature, 10
  POS.Feature, 13
* Position weight matrix
  Bayes.Feature, 3
  MN.Fdtf.Feature, 10
  POS.Feature, 13
  WMM.Feature, 20
* Positional independence
  Bayes.Feature, 3
  WMM.Feature, 20
* Second order Markov model
  MM2.Feature, 8
* Sequence encoding
  Density.Feature, 4
  Maldoss.Feature, 6
* Splice sites motifs
  SAE.Feature, 15
* Splice sites
  APR.Feature, 2
  Density.Feature, 4
  PN.Fdtf.Feature, 11
* Tri-nucleotide frequency
  Trint.Dist.Feature, 18
* True and False splice sites
  Maldoss.Feature, 6
* datasets
  droso, 5
  APR.Feature, 2
  Bayes.Feature, 3, 11, 14, 21
  Density.Feature, 4
  DNAStringSet, 2, 3, 5–7, 9–11, 13, 14, 16–20
  droso, 5
  Maldoss.Feature, 6
  MM1.Feature, 7
  MM2.Feature, 8
  MN.Fdtf.Feature, 4, 10, 12, 14, 21
  PN.Fdtf.Feature, 3, 7, 8, 11, 11, 20
  POS.Feature, 4, 13
  Predoss.Feature, 14
  SAE.Feature, 15
  Sparse.Feature, 17
  Trint.Dist.Feature, 18
  WAM.Feature, 3, 7–9, 12, 16, 19
  WMM.Feature, 11, 14, 20