Package 'naddaR'

January 25, 2018

Title Prediction of Protein Conserved Regions Using NADDA Algorithm

Version 0.99.0

Author Armen Abnousi <a.abnousi@gmail.com>

Maintainer Armen Abnousi <a.abnousi@gmail.com>

Description Generates protein sequence profiles using frequency of the k-mers present in the sequences and predicts conserved regions using those profiles. Can operate in parallel or on single processor.

biocViews Clustering,
MultipleSequenceAlignment

Depends R (>= 3.4.1),
pbdMPI,
data.table,
Biostrings,
utils,
stats

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Imports Rdpack

RdMacros Rdpack

R topics documented:

count_kmers *Counting k-mers in the dataset.*

Description

counts the number of times each k-mer appears in the dataset and returns a dataframe indicating these counts. Each k-mer in each sequence is counted at most once, i.e., if there are multiple occureneces of a k-mer in one sequence, only one of them is counted.

Usage

```
count_kmers(obj, klen = 6, parallel = TRUE, nproc = ifelse(parallel,
  comm.size(), 1), distributed = FALSE)
```

Arguments

obj	A filepath to a fasta file containing protein sequences or an AAStringSet object containing the sequences
klen	length of the k-mers to be used
parallel	Indicating whether the operation should be p erformed in parallel
nproc	Currently not supported. Will use all processors available to the job on cluster
distributed	A boolean, indicating whether the data is spread among multiple processors.

Details

If *parallel* is set to **TRUE** and *distributed* is set to **FALSE**, the method distributes the data between different processors and sets *distributed* to **TRUE**. Otherwise, if the *parallel* is set to **FALSE** and *distributed* is set to **TRUE**, the kmer frequencies are computed on each processor separately but then communicated between each other, and therefore at the end all processors have the same set of frequencies for kmers stored, using which they will generate frequency profiles for their chunk of sequences. If you prefer to run the operation in serial, set both *parallel* and *distributed* to **FALSE**.

Value

Returns a dataframe with two columns. Each row includes one k-mer and an integer indicating the number of times that k-mer appears in the input dataset. Each k-mer in each sequence is counted at most once, i.e., if there are multiple occureneces of a k-mer in one sequence, only one of them is counted.

Author(s)

Armen Abnousi

References

Abnousi A, Broschat SL and Kalyanaraman A (2016). "A Fast Alignment-Free Approach for De Novo Detection of Protein Conserved Regions." *PloS one*, **11**(8), pp. e0161338.

See Also

generate_instances for generation of training and test instances for each index in each sequence.
generate_profiles for generation of sequence k-mer frequency profiles for each sequence
comm.size for writing a distributed data object to a single file

Examples

```
library(pbdMPI)
## Generate a set of three example protein sequences
seqs <- AAStringSet(c("seq1"="MLVVD",
                      "seq2"="PVVRA",
                      "seq3"="LVVR"))
## Count the kmers and generate a dataframe of the frequencies
freqs <- count_kmers(seqs, klen = 3, parallel = FALSE)
head(freqs)
##    kmer count
##1:  LVV  2
##2:  MLV  1
##3:  PVV  1
##4:  VRA  1
##5:  VVD  1
##6:  VVR  2
```

---

generate_instances *Generate Instances for Indices in Protein Sequences*

Description

Tconstructs a dataframe where each row corresponds to one index of one protein sequence from the input dataset. It can be used to generate training and test sets to train a NADDA classification model or to predict the conserved indices of input sequences based on a trained model.

Usage

```
generate_instances(obj, labeled = TRUE, parallel = TRUE,
  nproc = ifelse(parallel, comm.size(), 1), groundtruth = NULL,
  truth_filename = NULL, klen = 6, normalize = TRUE, impute = TRUE,
  winlen = 20, imputing_length = winlen%/%2, distributed = FALSE)
```

Arguments

obj	A filepath to a fasta file containing protein sequences or an AAStringSet object containing the sequences
labeled	TRUE if the method is called to construct a training set using a Pfam or InterPro labels or FALSE otherwise.
parallel	Indicating whether the operation should be performed in parallel
nproc	Currently not supported. Will use all processors available to the job on cluster

`groundtruth`	A character string. Can be Pfam or InterPro
`truth_filename`	The filepath to the labels file for generating the training set based on it
`klen`	length of the k-mers to be used
`normalize`	A boolean value, indicating whether the k-mer frequencies should be normalized
`impute`	A boolean value, indicating whether imputed values should be inserted at the beginning and the end of the profiles
`winlen`	An integer, size the window used for generation of each instance
`imputing_length`	An integer, number of frequencies from the beginning and end of a sequence profile that should be used to impute the new values
`distributed`	A boolean, indicating whether the data is spread among multiple processors.

Details

Current version only supports Pfam and InterPro output files for generation of training set. The output from Pfam output file needs to be tabularized (replacing spaces with tabs).

If *parallel* is set to **TRUE** and *distributed* is set to **FALSE**, the method distributes the data between different processors and sets *distributed* to **TRUE**. Otherwise, if the *parallel* is set to **FALSE** and *distributed* is set to **TRUE**, the kmer frequencies are computed on each processor separately but then communicated between each other, and therefore at the end all processors have the same set of frequencies for kmers stored, using which they will generate frequency profiles and instances of their chunk of sequences. If you prefer to run the operation in serial, set both *parallel* and *distributed* to **FALSE**.

Value

Returns a dataframe with one row for each instance. Each row contains *winlen* k-mer frequencies around an index of a protein. The index number is stored in *position* column. Name of the sequence is stored in *name* column. If a training set is constructed, one column indicating whether it is a conserved index or not and a second column indicating the number of proteins in the dataset that have a similar conserved region are added to the returned dataframe.

Author(s)

Armen Abnousi

References

Abnousi A, Broschat SL and Kalyanaraman A (2016). "A Fast Alignment-Free Approach for De Novo Detection of Protein Conserved Regions." *PloS one*, **11**(8), pp. e0161338.

See Also

`generate_profiles` for generation of frequency profiles for each sequence
`comm.size` for writing a distributed data object to a single file

Examples

```
## Generate a set of three example protein sequences
seqs <- AAStringSet(c("seq1"="MLVVD",
                      "seq2"="PVVRA",
                      "seq3"="LVVR"))
## Count the kmers and generate a dataframe of the frequencies
ins <- generate_instances(seqs, labeled = FALSE, parallel = FALSE, klen = 3, impute = TRUE, winlen = 5, normalize
head(ins)
##    name      position        freqn2  freqn1  freqindex       freqp1  freqp2
##    seq1      1               1.5     1.5     1.0             2.0     1.0
##    seq1      2               1.5     1.0     2.0             1.0     1.5
##    seq1      3               1.0     2.0     1.0             1.5     1.5
##    seq1      4               2.0     1.0     1.5             1.5     1.5
##    seq1      5               1.0     1.5     1.5             1.5     1.5
##    seq2      1               1.5     1.5     1.0             2.0     1.0
```

---

generate_profiles *Generate Protein k-mer frequency profiles*

Description

constructs a dataframe where each row corresponds to one index of one protein sequence from the input dataset. It can be used to generate training and test sets to train a NADDA classification model or to predict the conserved indices of input sequences based on a trained model.

Usage

```
generate_profiles(obj, klen = 6, parallel = TRUE, nproc = ifelse(parallel,
  comm.size(), 1), normalize = TRUE, impute = TRUE, winlen = 20,
  imputing_length = winlen%/%2, distributed = FALSE)
```

Arguments

obj	A filepath to a fasta file containing protein sequences or an AAStringSet object containing the sequences
klen	length of the k-mers to be used
parallel	Indicating whether the operation should be performed in parallel
nproc	Currently not supported. Will use all processors available to the job on cluster
normalize	A boolean value, indicating whether the k-mer frequencies should be normalized
impute	A boolean value, indicating whether imputed values should be inserted at the beginning and the end of the profiles
winlen	An integer, size the window used for generation of each instance
imputing_length	An integer, number of frequencies from the beginning and end of a sequence profile that should be used to impute the new values
distributed	A boolean, indicating whether the data is spread among multiple processors.

Details

If *parallel* is set to **TRUE** and *distributed* is set to **FALSE**, the method distributes the data between different processors and sets *distributed* to **TRUE**. Otherwise, if the *parallel* is set to **FALSE** and *distributed* is set to **TRUE**, the kmer frequencies are computed on each processor separately but then communicated between each other, and therefore at the end all processors have the same set of frequencies for kmers stored, using which they will generate frequency profiles for their chunk of sequences. If you prefer to run the operation in serial, set both *parallel* and *distributed* to **FALSE**.

Value

Returns a list with one vector for each protein sequence in the dataset. A vector for sequence *s* contains |s| - klen + 1 indices if *impute* is set to **FALSE** (where |s| is the length of the sequence). Otherwise it will include one index for each position in the sequence but also *winlen %\% 2* indices at the beginning and end of each sequence.

Author(s)

Armen Abnousi

References

Abnousi A, Broschat SL and Kalyanaraman A (2016). "A Fast Alignment-Free Approach for De Novo Detection of Protein Conserved Regions." *PloS one*, **11**(8), pp. e0161338.

See Also

`generate_instances` for generation of training and test instances for each index in each sequence
`comm.size` for writing a distributed data object to a single file

Examples

```
## Generate a set of three example protein sequences
seqs <- AAStringSet(c("seq1"="MLVVD",
                      "seq2"="PVVRA",
                      "seq3"="LVVR"))
## Count the kmers and generate a dataframe of the frequencies
profs <- generate_profiles(seqs, klen = 3, parallel = FALSE, winlen = 5, normalize = FALSE)
head(profs)
profs
##[[1]]
##[[1]]$freqs
##[1] 1.5 1.5 1.0 2.0 1.0 1.5 1.5 1.5 1.5
##[[1]]$seq
##[1] "seq1"
##
##[[2]]
##[[2]]$freqs
##[1] 1.5 1.5 1.0 2.0 1.0 1.5 1.5 1.5 1.5
##[[2]]$seq
##[[1]] "seq2"
##
```

```
##[[3]]
##[[3]]$freqs
##[1] 2 2 2 2 2 2 2 2
##[[3]]$seq
##[1] "seq3"
```

Index