Package ‘DeProViR’

October 16, 2024

Type Package

Title A Deep-Learning Framework Based on Pre-trained Sequence Embeddings for Predicting Host-Viral Protein-Protein Interactions

Version 1.0.0

Description Emerging infectious diseases, exemplified by the zoonotic COVID-19 pandemic caused by SARS-CoV-2, are grave global threats. Understanding protein-protein interactions (PPIs) between host and viral proteins is essential for therapeutic targets and insights into pathogen replication and immune evasion. While experimental methods like yeast two-hybrid screening and mass spectrometry provide valuable insights, they are hindered by experimental noise and costs, yielding incomplete interaction maps. Computational models, notably DeProViR, predict PPIs from amino acid sequences, incorporating semantic information with GloVe embeddings. DeProViR employs a Siamese neural network, integrating convolutional and Bi-LSTM networks to enhance accuracy. It overcomes the limitations of feature engineering, offering an efficient means to predict host-virus interactions, which holds promise for antiviral therapies and advancing our understanding of infectious diseases.

License MIT+ file LICENSE

Encoding UTF-8

URL `https://github.com/mrbakhsh/DeProViR`

BugReports `https://github.com/mrbakhsh/DeProViR/issues`

Depends keras

Imports caret, data.table, dplyr, fmsb, ggplot2, grDevices, pROC, PRROC, readr, stats, BiocFileCache, utils

VignetteBuilder knitr

Suggests rmarkdown, tensorflow, BiocStyle, RUnit, knitr, BiocGenerics

biocViews Proteomics, SystemsBiology, NetworkInference, NeuralNetwork, Network

RoxygenNote 7.3.1

git_url https://git.bioconductor.org/packages/DeProViR

git_branch RELEASE_3_19

git_last_commit 07d78d3

git_last_commit_date 2024-04-30

Repository Bioconductor 3.19

Date/Publication 2024-10-16

Author Matineh Rahmatbakhsh [aut, trl, cre]

Maintainer Matineh Rahmatbakhsh <matinerb.94@gmail.com>

Contents

`encodeHostSeq` *Host Protein Sequence Encoding with GloVe Embedding Vectors*

Description

This function first first encodes amino acids as a sequence of unique 20 integers though tokenizer. The padding token was added to the front of shorter sequences to ensure a fixed-length vector of defined size L (i.e., here is 1000). Embedding matrix is then constructed to transform amino acid tokens to pre-training embedding weights, in which rows represent the amino acid tokens created earlier, and columns correspond to 100-dimension weight vectors derived from GloVe word-vector-generation vector map.

Usage

```
encodeHostSeq(trainingSet, embeddings_index)
```

Arguments

`trainingSet`	a data.frame containing training information
`embeddings_index`	embedding outputted from `gloveImport`

Value

A list containing Embedding matrix and tokenization

Examples

```
# Download and load the index
embeddings_index <- gloveImport()
#load training set
dt <- loadTrainingSet()
#encoding
encodeHostSeq <- encodeHostSeq(dt, embeddings_index)
```

encodeViralSeq   *Viral Protein Sequence Encoding with GloVe Embedding Vectors*

Description

This function first first encodes amino acids as a sequence of unique 20 integers though tokenizer. The padding token was added to the front of shorter sequences to ensure a fixed-length vector of defined size L (i.e., here is 1000). Embedding matrix is then constructed to transform amino acid tokens to pre-training embedding weights, in which rows represent the amino acid tokens created earlier, and columns correspond to 100-dimension weight vectors derived from GloVe word-vector-generationvector map.

Usage

```
encodeViralSeq(trainingSet, embeddings_index)
```

Arguments

`trainingSet`	a data.frame containing training information
`embeddings_index`	embedding outputted from `gloveImport`

Value

A list containing Embedding matrix and tokenization

Examples

```
# Download and load the index
embeddings_index <- gloveImport()
#load training set
dt <- loadTrainingSet()
#encoding
encoded_seq <- encodeViralSeq(dt, embeddings_index)
```

gloveImport *Cache and Load Pre-Trained Word Vectors*

Description

This function cache and loads pre-trained GloVe vectors (100d).

Usage

```
gloveImport(url_path = "https://nlp.stanford.edu/data")
```

Arguments

url_path	URL path to GloVe embedding. Defaults to "https://nlp.stanford.edu/data"

Value

glove embedding

Examples

```
options(timeout=240)
embeddings_index <- gloveImport(url_path = "https://nlp.stanford.edu/data")
```

loadPreTrainedModel *Load Pre-Trained Model Weights*

Description

This function loads the pre-trained model weights constructed previously using modelTraining

Usage

```
loadPreTrainedModel(
  input_dim = 20,
  output_dim = 100,
  filters_layer1CNN = 32,
  kernel_size_layer1CNN = 16,
  filters_layer2CNN = 64,
  kernel_size_layer2CNN = 7,
  pool_size = 30,
  layer_lstm = 64,
  units = 8,
  metrics = "AUC",
  filepath = system.file("extdata", "Pre_trainedModel", package = "DeProViR")
)
```

Arguments

input_dim Integer. Size of the vocabulary, i.e. amino acid tokens. Defults to 20. See keras.

output_dim Integer. Dimension of the dense embedding, i.e., GloVe. Defaults to 100. See keras.

filters_layer1CNN
Integer, the dimensionality of the output space (i.e. the number of output filters in the first convolution). Defaults to 32. See keras

kernel_size_layer1CNN
An integer or tuple/list of 2 integers, specifying the height and width of the convolution window in the first layer. Can be a single integer to specify the same value for all spatial dimensions. Defaults to 16. See keras

filters_layer2CNN
Integer, the dimensionality of the output space (i.e. the number of output filters in the second convolution). Defaults to 64. See keras

kernel_size_layer2CNN
An integer or tuple/list of 2 integers, specifying the height and width of the convolution window in the second layer. Can be a single integer to specify the same value for all spatial dimensions. Defaults to 7. See keras

pool_size Down samples the input representation by taking the maximum value over a spatial window of size pool_size. Defaults to 30.See keras

layer_lstm Number of units in the Bi-LSTM layer. Defaults to 64. See keras

units Number of units in the MLP layer. Defaults to 8. See keras

metrics Vector of metric names to be evaluated by the model during training and testing. Defaults to "AUC". See keras

filepath A character string indicating the path contained pre-trained model weights, i.e., inst/extdata/Pre-trainedModel

Value

Pre-trained model.

Examples

```
Loading_trainedModel <- loadPreTrainedModel()
```

loadTrainingSet *Load Demo Training Set*

Description

This function loads demo training set.

Usage

```
loadTrainingSet(
  training_dir = system.file("extdata", "training_Set", package = "DeProViR")
)
```

Arguments

`training_dir` dir containing a training data.frame .csv Default set to "extdata/training_testSets".

Value

data.frame

Examples

```
dt <- loadTrainingSet()
```

`modelTraining` *Predictive Model Training using k-fold Validation Strategy*

Description

This function first transforms protein sequences to amino acid tokens wherein tokens are indexed by positive integers, then represents each amino acid token by pre-trained co-occurrence embedding vectors learned by GloVe, followed by applying an embedding layer. Then it employs Siamese-like neural network articheture on densly-connected neural net to predict interactions between host and viral proteins.

Usage

```
modelTraining(
  url_path = "https://nlp.stanford.edu/data",
  training_dir = system.file("extdata", "training_Set", package = "DeProViR"),
  input_dim = 20,
  output_dim = 100,
  filters_layer1CNN = 32,
  kernel_size_layer1CNN = 16,
  filters_layer2CNN = 64,
  kernel_size_layer2CNN = 7,
  pool_size = 30,
  layer_lstm = 64,
  units = 8,
  metrics = "AUC",
  cv_fold = 10,
  epochs = 100,
  batch_size = 128,
  plots = TRUE,
  tpath = tempdir(),
  save_model_weights = TRUE,
  filepath = tempdir()
)
```

Arguments

`url_path` URL path to GloVe embedding. Defaults to "https://nlp.stanford.edu/data/glove.6B.zip".

`training_dir` dir containing viral-host training set. See `loadTrainingSet`

`input_dim` Integer. Size of the vocabulary, i.e. amino acid tokens. Defults to 20. See `keras`.

`output_dim` Integer. Dimension of the dense embedding, i.e., GloVe. Defaults to 100. See `keras`.

`filters_layer1CNN` Integer, the dimensionality of the output space (i.e. the number of output filters in the first convolution). Defaults to 32. See `keras`

`kernel_size_layer1CNN` An integer or tuple/list of 2 integers, specifying the height and width of the convolution window in the first layer. Can be a single integer to specify the same value for all spatial dimensions.Defaults to 16. See `keras`

`filters_layer2CNN` Integer, the dimensionality of the output space (i.e. the number of output filters in the second convolution). Defaults to 64. See `keras`

`kernel_size_layer2CNN` An integer or tuple/list of 2 integers, specifying the height and width of the convolution window in the second layer. Can be a single integer to specify the same value for all spatial dimensions. Defaults to 7. See `keras`

`pool_size` Down samples the input representation by taking the maximum value over a spatial window of size pool_size. Defaults to 30.See `keras`

`layer_lstm` Number of units in the Bi-LSTM layer. Defaults to 64. See `keras`

`units` Number of units in the MLP layer. Defaults to 8. See `keras`

`metrics` Vector of metric names to be evaluated by the model during training and testing. Defaults to "AUC". See `keras`

`cv_fold` Number of partitions for cross-validation. Defaults to 10.

`epochs` Number of epochs to train the model. Defaults to 100. See `keras`

`batch_size` Number of samples per gradient update.Defults to 128. See `keras`

`plots` PDF file containing perfromance measures. Defaults to TRUE. See `performancePlots`

`tpath` A character string indicating the path to the project directory. If the directory is missing, PDF file containing perfromance measures will be stored in the Temp directory. See `performancePlots`

`save_model_weights` If TRUE, save the trained weights. Defaults to TRUE.

`filepath` A character string indicating the path to save the model weights. Default to tempdir().

Value

Trained model and perfromance measures.

performancePlots *Model Performance Evalution*

Description

This function plots model performance

Usage

```
performancePlots(pred_label, y_label, tpath = tempdir())
```

Arguments

pred_label	predicted labels
y_label	Ground truth labels
tpath	A character string indicating the path to the project directory. If the directory is missing, PDF file will be stored in the Temp directory.

Value

Pdf file containing perfromanc plots

Examples

```
pred_label <- seq(0,1, length.out = 100)
truth_label <- rep(c(0,1), each = 50)
perf <- performancePlots(pred_label,
                         truth_label,
                         tpath = tempdir())
```

predInteractions *Predict Unknown Interactions*

Description

This function initially constructs an embedding matrix from the viral or host protein sequences and then predicts scores for unknown interactions. Interactions with scores greater than 0.5 are more likely to indicate interaction.

Usage

```
predInteractions(
  url_path = "https://nlp.stanford.edu/data",
  Testingset,
  trainedModel
)
```

Arguments

`url_path` URL path to GloVe embedding. Defaults to "https://nlp.stanford.edu/data/glove.6B.zip".

`Testingset` A data.frame containing unknown interactions. For demo, we can use the file in extdata/test_Set.

`trainedModel` Pre-trained model stored in extdata/Pre_trainedModel or the training model "$merge_model" achieved by `modelTraining`.

Value

Probability scores for unknown interactions

Examples

```
trainedModel <- loadPreTrainedModel()
# load test set (i.e., unknown interactions)
testing_set <- data.table::fread(system.file("extdata", "test_Set",
                               "test_set_unknownInteraction.csv",
                                package = "DeProViR"))
# now predict interactions
options(timeout=240)
predInteractions <- predInteractions(url_path = "https://nlp.stanford.edu/data",
                                     testing_set,
                                     trainedModel)
```

Index