

# Package ‘deepredef’

October 23, 2020

**Title** Deep Learning Prediction of Effectors

**Version** 0.1.0

**Description** A tool that contains trained deep learning models for predicting effector proteins. 'deepredef' has been trained to identify effector proteins using a set of known experimentally validated effectors from either bacteria, fungi, or oomycetes. Documentation is available via several vignettes, and the paper by Kristianingsih and MacLean (2020) <doi:10.1101/2020.07.08.193250>.

**License** MIT + file LICENSE

**URL** <https://github.com/ruthkr/deepredef/>

**BugReports** <https://github.com/ruthkr/deepredef/issues/>

**Depends** R (>= 2.10)

**Imports** Biostrings, dplyr, ggplot2, ggthemes, keras, magrittr, purrr, reticulate, rlang, seqinr, tensorflow

**Suggests** covr, kableExtra, knitr, rmarkdown, stringr, testthat

**VignetteBuilder** knitr

**biocViews**

**Config/reticulate** list( packages = list( list(package = ``tensorflow", version = ``2.0.0", pip = FALSE) ) )

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**NeedsCompilation** no

**Author** Ruth Kristianingsih [aut, cre, cph]  
(<<https://orcid.org/0000-0003-1873-6203>>)

**Maintainer** Ruth Kristianingsih <[ruth.kristianingsih30@gmail.com](mailto:ruth.kristianingsih30@gmail.com)>

**Repository** CRAN

**Date/Publication** 2020-10-23 10:30:06 UTC

## R topics documented:

aasset_to_df . . . . .	2
aas_to_df . . . . .	3
deepredef . . . . .	3
encode_integer . . . . .	4
encode_one_hot . . . . .	4
ensemble_weighted . . . . .	5
fasta_to_df . . . . .	5
get_ensemble_method . . . . .	6
install_tensorflow . . . . .	6
load_model . . . . .	7
package_glob . . . . .	8
plot.tbl_deepredef . . . . .	8
prediction_mapper . . . . .	9
predict_effector . . . . .	9
summary.tbl_deepredef . . . . .	11
<b>Index</b>	<b>12</b>

---

aasset_to_df	<i>Convert AAStringset class to dataframe</i>
--------------	---

---

### Description

Convert AAStringset class to dataframe

### Usage

```
aasset_to_df(aas)
```

### Arguments

aas                   AAStringset class object.

### Value

Returns data in data frame.

### Examples

```
input <- "MSHMTFNTWKAGLWRLAAAVALSLLPVVARAAVPGITGPTFDLTAQPGRANQPDGASVYSWGYGCNPRTPVPGFLPSVNPLAGQ"
input_aas <- Biostrings::AAString(input)
input_df <- aasset_to_df(input_aas)
```

---

aas_to_df	<i>Convert AAString class to dataframe</i>
-----------	--

---

**Description**

Convert AAString class to dataframe

**Usage**

```
aas_to_df(aas)
```

**Arguments**

aas                   AAString class object.

**Value**

Returns data in data frame.

**Examples**

```
input_fasta <- system.file("extdata/example/fungi_sample.fasta", package = "deepredef")
input_aas <- Biostrings::readAAStringSet(input_fasta)
input_df <- aasset_to_df(input_aas)
```

---

deepredef	<i>deepredef package</i>
-----------	--------------------------

---

**Description**

Effector protein predictor using Deep Learning models.

**Details**

See the README on [GitHub](#)

---

encode_integer	<i>Integer encoder</i>
----------------	------------------------

---

**Description**

encode\_integer integer-encodes sequence in a string format.

**Usage**

```
encode_integer(sequence, max_length = 4034)
```

**Arguments**

sequence	Sequence in a string format.
max_length	Maximum length of sequence to encode.

**Value**

Integer encoded sequence.

**Examples**

```
sample_seq <- "MSHMTFNTWKAGLWRLAAAAVLSLLPVVARAAVPGITGPTFDLTAQPGRANQPDGASVYSWGYGCNPRTPVPGFLPSVNPLAGQ"  
encoded_seq <- encode_integer(sample_seq)
```

---

encode_one_hot	<i>One-hot encoder</i>
----------------	------------------------

---

**Description**

encode\_one\_hot one-hot-encodes sequence in a string format.

**Usage**

```
encode_one_hot(sequence, max_length = 4034)
```

**Arguments**

sequence	Sequence in a string format.
max_length	Maximum length of sequence to encode.

**Value**

One-hot encoded sequence.

**Examples**

```
sample_seq <- "MSHMTFNTWKAGLWRLAAAAVL SLLPVVARAAVPGITGPTFDLTAQPGRANQPDGASVYSWGYGCNPRTVPGFLPSVNPLAGQ"
encoded_seq <- encode_one_hot(sample_seq)
```

---

ensemble_weighted	<i>Weighted ensemble</i>
-------------------	--------------------------

---

**Description**

Weighted ensemble

**Usage**

```
ensemble_weighted(pred_list, weights)
```

**Arguments**

pred_list	List of deep learning models.
weights	Accuracy values from evaluation on the validation dataset.

**Value**

Returns the prediction results from weighted ensemble.

---

fasta_to_df	<i>Convert fasta format to dataframe</i>
-------------	--

---

**Description**

Convert fasta format to dataframe

**Usage**

```
fasta_to_df(fasta_path)
```

**Arguments**

fasta_path	Path of FASTA file.
------------	---------------------

**Value**

Returns data in data frame.

**Examples**

```
input <- system.file("extdata/example/fungi_sample.fasta", package = "depredeff")
input_df <- fasta_to_df(input)
```

---

get\_ensemble\_method     *Get ensemble methods*

---

### Description

This function is used to get the ensemble methods used for each taxon group. If weights are needed for a particular ensemble, then the weights will automatically follow.

### Usage

```
get_ensemble_method(taxon)
```

### Arguments

taxon                    taxon group

### Value

Returns ensemble method and weights.

---

install\_tensorflow     *Install the TensorFlow backend*

---

### Description

TensorFlow will be installed into an "r-tensorflow" virtual or conda environment. Note that "virtualenv" is not available on Windows (as this isn't supported by TensorFlow).

### Usage

```
install_tensorflow(
  method = c("conda", "virtualenv"),
  conda = "auto",
  version = "2.0.0",
  extra_packages = NULL,
  ...
)
```

### Arguments

method                    Installation method ("conda" or "virtualenv").

conda                      Path to conda executable (or "auto" to find conda using the PATH and other conventional install locations).

version                    TensorFlow version to install ( by default, "2.0.0").

extra\_packages            Additional PyPI packages to install along with TensorFlow.

...                        Other arguments passed to [reticulate::py\\_install\(\)](#).

## Custom Installation

Custom installations of TensorFlow are not supported yet by deepredef.

## Additional Packages

If you wish to add additional PyPI packages to your TensorFlow environment you can either specify the packages in the `extra_packages` argument of `install_tensorflow()`, or alternatively install them into an existing environment using the `reticulate::py_install()` function.

Notice that this may have undesired side-effects on Windows installations.

## Examples

```
## Not run:  
# Default installation  
library(deepredef)  
install_tensorflow()  
  
## End(Not run)
```

---

load\_model

*Load model*

---

## Description

`load_model()` loads model saved in hdf5 format

## Usage

```
load_model(taxon = c("bacteria", "fungi", "oomycete"))
```

## Arguments

taxon            Name of taxon.

## Value

Returns all of the hyperparameters and parameters of particular model from specific taxon chosen.

---

package_glob	<i>Wildcard Expansion on File Paths</i>
--------------	---

---

**Description**

Wildcard Expansion on File Paths

**Usage**

```
package_glob(..., pattern)
```

**Arguments**

...	Path
pattern	Pattern

**Value**

Glob

---

plot.tbl_deepredef	<i>Plot the results of prediction</i>
--------------------	---------------------------------------

---

**Description**

Plot the results of prediction

**Usage**

```
## S3 method for class 'tbl_deepredef'  
plot(x, ...)
```

**Arguments**

x	tbl_deepredef object
...	additional arguments ignored.

**Value**

class distribution plot



### Examples

```
# FASTA input
input_fasta <- system.file("extdata/example/fungi_sample.fasta", package = "deepredef")

pred_result <- deepredef::predict_effector(
  input = input_fasta,
  taxon = "fungi"
)

plot(pred_result)
```

---

*prediction\_mapper*      *Prediction mapper helper*

---

### Description

Prediction mapper helper

### Usage

```
prediction_mapper(sequence_list, model_list)
```

### Arguments

`sequence_list`    List of sequences input.  
`model_list`        List of models.

### Value

Returns list of prediction result of each sequence.

---

*predict\_effector*      *Predict effector*

---

### Description

`predict_effector` is used to predict effector protein given amino acid protein sequences.

**Usage**

```

predict_effector(input, taxon)

## S3 method for class 'character'
predict_effector(input, taxon)

## S3 method for class 'data.frame'
predict_effector(input, taxon)

## S3 method for class 'AAStringSet'
predict_effector(input, taxon)

## S3 method for class 'AAString'
predict_effector(input, taxon)

## Default S3 method:
predict_effector(input, taxon)

```

**Arguments**

<code>input</code>	Input data that contains amino acid sequence(s). It can be in fasta format, strings, AAString, AAStringset, and dataframe.
<code>taxon</code>	Taxon group of input data. Available taxons are bacteria, fungi, and oomycete.

**Value**

`predict_effector` returns an object of class "tbl\_deepredef" or for multiple responses of class c("tbl\_deepredef", "data.frame").

An object of class "tbl\_deepredef" is a data frame containing at least the following components:

<code>sequence</code>	the sequence(s) from the input data.
<code>s_score</code>	score obtained from sigmoid function showing how likely the sequences to be an effector.
<code>prediction</code>	class prediction for each sequence, obtained from <code>s_score</code> . If the value of <code>s_score</code> $\geq 0.5$ , it will be classified as an effector. Otherwise, it will be classified as a non-effector.

**Examples**

```

# FASTA input
input_fasta <- system.file("extdata/example/fungi_sample.fasta", package = "deepredef")

pred_result <- deepredef::predict_effector(
  input = input_fasta,
  taxon = "fungi"
)

```

---

```
summary.tbl_deepredef
```

*Get the summary of the prediction results*

---

### **Description**

Get the summary of the prediction results

### **Usage**

```
## S3 method for class 'tbl_deepredef'  
summary(object, ...)
```

### **Arguments**

object	Results of prediction from <code>deepredef::predict_effector()</code> .
...	Additional arguments ignored.

### **Examples**

```
# FASTA input  
input_fasta <- system.file("extdata/example/fungi_sample.fasta", package = "deepredef")  
  
pred_result <- deepredef::predict_effector(  
  input = input_fasta,  
  taxon = "fungi"  
)  
  
summary(pred_result)
```

# Index

aas\_to\_df, 3  
aasset\_to\_df, 2  
  
deepredef, 3  
  
encode\_integer, 4  
encode\_one\_hot, 4  
ensemble\_weighted, 5  
  
fasta\_to\_df, 5  
  
get\_ensemble\_method, 6  
  
install\_tensorflow, 6  
  
load\_model, 7  
  
package\_glob, 8  
plot.tbl\_deepredef, 8  
predict\_effector, 9  
prediction\_mapper, 9  
  
reticulate::py\_install(), 6, 7  
  
summary.tbl\_deepredef, 11