User Guide of DeepDetect

Version 1.0.0

Last revised February 26, 2023

Jinghan Yang

Contents:

1 Introduction....2

2 Installation....2

2.1 Environment Requirements....2

2.2 Downloading DeepDetect....2

3 Running DeepDetect....3

3.1 Parameters....3

3.2 Running the command....4

4 Output file formats....4

1 Introduction

DeepDetect is a command line tool written in Python. It is based on a bidirectional long short-term memory (BiLSTM) network for the prediction of peptide detectability enhanced by peptide digestibility. This tool supports eight commonly used proteases, covering Trypsin, ArgC, Chymotrypsin, GluC, LysC, AspN, LysN, and LysargiNase. DeepDetect is freely available at http://fugroup.amss.ac.cn/software/DeepDetect/DeepDetect.html.

2 Installation

2.1 Environment Requirements

DeepDetect is a program based on Keras using TensorFlow backend. An installed Python environment (v3.6) is preferred with the following packages: os, sys, re, getopt, NumPy (≥1.14.5), TensorFlow (v1.10.0), and Keras (v2.2.4). For convenience, we strongly recommend the user to install Anaconda 3 (64-bit) in the local computer, and all the packages can be installed through pip.ss

2.2 Downloading DeepDetect

DeepDetect can be freely available from http://fugroup.amss.ac.cn/software/DeepDetect/DeepDetect.html. The user can download the released version of DeepDetect executable program "DeepDetect.zip" (Figure 1.a), including a help document "User Guide of DeepDetect.pdf". Besides, the source code "SourceCode.zip" (Figure 1.b) also can be accessed for the convenience of the community, containing three folders – "bilstm_training" for BiLSTM training, "deepdetect_pred" for deepdetect prediction in the Python environment, and "diann_library_reduction" for spectral library reduction with DeepDetect. The "Datasets.zip" involves the nineteen data sets and fasta files used in the paper, as well as a sample fasta file and its predicted results by DeepDetect (Figure 1.c).

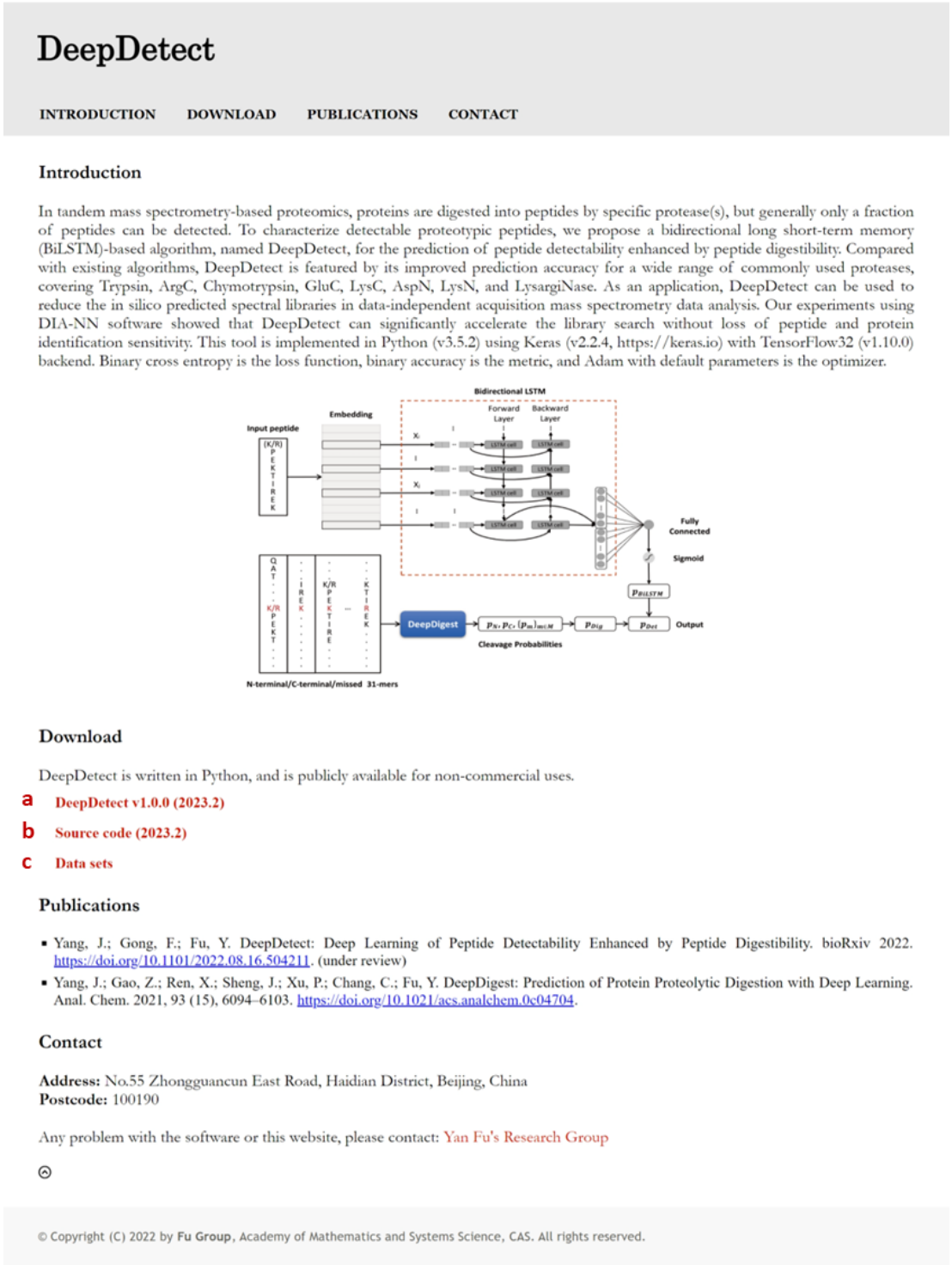


Figure 1. The screenshot of the DeepDetect webpage.

3 Running DeepDetect

3.1 Parameters

Users can set the custom parameters which are annotated in Table 1.

Table 1. Annotations of the parameters.

Parameter name	Meaning
input	The path of protein sequence file in FASTA format (.fasta)
output	The path of output file in TXT format (.txt)
regular	The regular expression used to extract the protein id (default: ">(.*?)\s")
protease	The digestion protease (default: Trypsin)
missed_cleavages	The maximum number of missed cleavages allowed in each theoretical peptide fragment (default: 2)
min_len	The minimum length of the theoretical peptide fragments (default: 7)
max_len	The maximum length of the theoretical peptide fragments (default: 47)

3.2 Running the command

First, unpack the "exe.zip" file in the tool folder. Then, open the command interpreter "cmd.exe" and run DeepDetect in the path of the exe tool by the following format:

>DeepDetect.exe --input=*the path of protein sequence file* --output=*the path of output file* --regular=">(.*?)\s" --protease=Trypsin --missed_cleavages=2 --min_len =7 --max_len=47

An example is as follows (Figure 2):

```
E:\DeepDetect\tool\exe>DeepDetect.exe --input=.\SampleData\Sample_Data.fasta --output=.\SampleData\Sample_Data.txt --regular=">(.*?)\s" --protease=Trypsin --missed_cleavages=2 --min_len=7 --max_len=47
```

Figure 2. Illustration of running DeepDetect.

4 Output file formats

Once the calculation is done, DeepDetect generates a .txt file in the output directory. The detailed description of each column in this file is shown in Table 2.

Table 2. Descriptions of headers in the result file.

Name	Description
Protein_id	The identity of the protein from which the peptide is digested
Peptide_sequence	The sequence of the theoretical digested peptide
Peptide detectability	The predicted peptide detectability of the peptide