**User guide**

1. **Requirements and installation**

Model construction was performed using python (3.8.3, Anaconda distribution version 5.3.1, <https://www.anaconda.com/>) with the following packages: FastNLP (0.6.0), pytorch (1.8.1), transformers (4.12.5), bidict (0.22.0) and pyteomics (4.5.5).

Data analysis for FLR estimation was performed using python (3.8.3) with the following packages: pandas (1.0.5) and numpy (1.18.5).

Install these packages by using “pip install” command:

pip install fastNLP

pip install torch

pip install transformers

pip install bidict

pip install pyteomics

pip install pandas

pip install numpy

These commands will install the latest stable release of the libraries. If you want to install a specific version of the library, you can specify the version number in the command. Take numpy as an example:

pip install numpy==1.18.5

Once the library is installed, you can verify the installation by importing the library in Python and checking the version number:

import numpy as np

print(np.\_\_version\_\_)

The packages can be installed within a few seconds using a computer with the recommended specs (16 GB RAM, 8 cores@1.8 GHz) and internet speed of 100 Mbps.

1. **Example data**

All the example data can be found in the "demo\_data" folder.

Sequence searching results:

msms\_sample.txt / msms\_sample\_multi.txt

*The msms\_sample.txt file contains the MaxQuant results for singly phosphorylated peptides, while the msms\_sample\_multi.txt file contains the MaxQuant results for multiply phosphorylated peptides. In practical application, the msms.txt file of the Maxquant results can be accessed to obtain the desired file. You can also use searching results from other tools by modifying the format into the MaxQuant type.*

Phospho\_STY\_Sites.txt

*Phospho\_STY\_Sites.txt is obtained from MaxQuant searching results. In practical application, the Phospho (STY)Sites.txt file of the Maxquant results can be accessed to obtain the desired file. You can also use the search results containing protein site information from other tools and modify the format.*

Spectra files

mgf\_demo.mgf / mgf\_multi\_demo.mgf

*The mgf\_demo.mgf file contains the spectra for singly phosphorylated peptides, while the mgf\_multi\_demo.mgf file contains the spectra for multiply phosphorylated peptides To obtain an MGF file, you can use a program or tool that is capable of converting your data into the MGF format. Here, we use MsConvert from the ProteoWizard Package (3.0.11579) with peak picking setting to convert experimental spectra (raw files) into Mascot generic format (MGF) format.*

DeepFLR files:

These files serve as important inputs and outputs for each step of the software. They can be obtained by running the corresponding code, and their filenames can be customized by the user.

mono\_target\_decoy\_msms\_sample.csv / multi\_target\_decoy\_msms\_sample.csv

*“mono\_target\_decoy\_msms\_sample.csv” is the output file after python Targetdecoy\_phosphopeptides\_generation\_mono.py, containing target and decoy phosphopeptides sequences for singly phosphorylated peptides. “multi\_target\_decoy\_msms\_sample.csv” is the output file after python Targetdecoy\_phosphopeptides\_generation\_multi.py, containing target and decoy phosphopeptides sequences for multiply phosphorylated peptides.*

mono\_target\_decoy\_msms\_samplemodelmonomz.csv

*This file contains the predicted spectra for input target or decoy sequences.*

mono\_target\_decoy\_msms\_samplemodelmonomz\_modelresult.csv

*This file contains the cosine similarity between predicted spectra and experimental spectra.*

FLRPSM\_demo.csv

*This file contains the delta score, the corresponding estimated FLR and the number of PSMs obtained under the cutoff.*

DeepFLR\_phosphosites\_obtained\_sample.csv

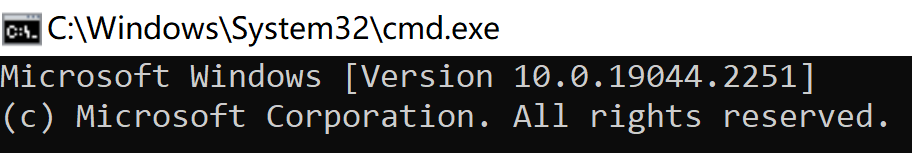
*This file contains the phosphosite localization information for desired estimated FLR.*

Fine-tuning data

finetune\_demo.csv

*This file can be used to fine-tune DeepFLR.*

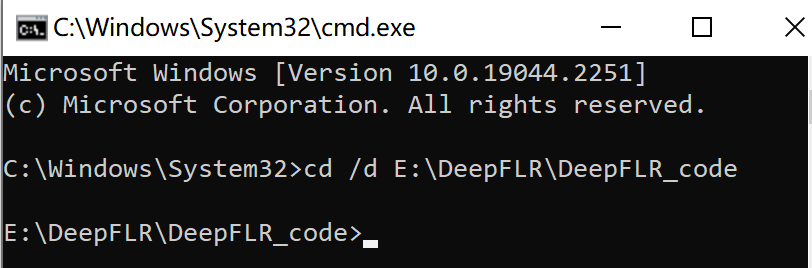
1. **Run the release version of DeepFLR using command line**
2. Open the “cmd.exe”.



(2) Entry to the folder including DeepFLR code files.

The users can use the command such as cd /d E:\DeepFLR\DeepFLR\_code to entry the folder. The path “E:\DeepFLR\DeepFLR\_code” indicates the folder containing

the python files for DeepFLR.



1. Generate target and decoy phosphopeptides sequences

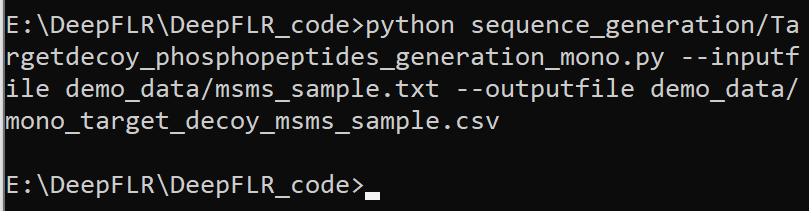
python sequence\_generation/Targetdecoy\_phosphopeptides\_generation\_mono.py --inputfile demo\_data/msms\_sample.txt --outputfile demo\_data/mono\_target\_decoy\_msms\_sample.csv

This command can be used to generate target and decoy phosphopeptide lists of singly phosphorylated peptides.

The description of the parameters of the command line:

“--inputfile”: the filename for the input file (sequence searching software results).

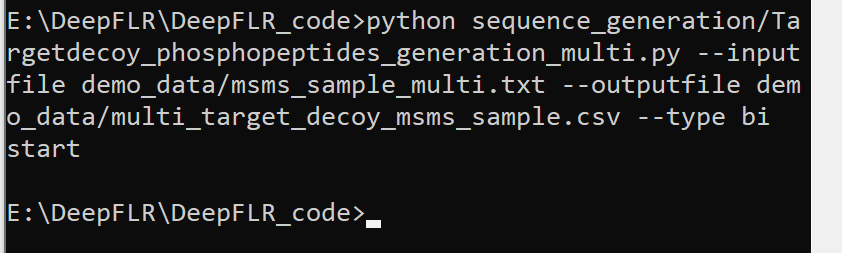
“--outputfile”: the desired output file name (target and decoy sequences).



The output file contains the target and decoy phosphopeptides sequences for singly phosphorylated peptides.

For multiply phosphorylated peptides, to generate target and decoy phosphopeptides, you can just change python Targetdecoy\_phosphopeptides\_generation\_mono.py to python Targetdecoy\_phosphopeptides\_generation\_multi.py. Besides, we provide an alternative optional parameter “type” (bi or multi) to decide whether only bi-phosphorylated peptides are processed or all the multiply phosphorylated peptides are processed. The default type is “bi”. For multiply phosphorylated peptides, the target and decoy phosphopeptide sequences are generated by:

python sequence\_generation/Targetdecoy\_phosphopeptides\_generation\_multi.py --inputfile demo\_data/msms\_sample\_multi.txt --outputfile demo\_data/multi\_target\_decoy\_msms\_sample.csv --type bi

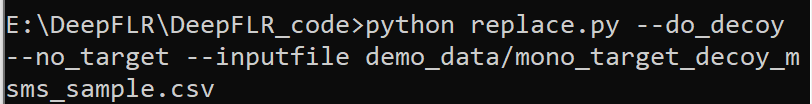


(4) Use DeepFLR to predict spectra for the generated target and decoy phosphopeptides.

python replace.py --do\_decoy --no\_target --inputfile demo\_data/mono\_target\_decoy\_msms\_sample.csv

This command is used to predict spectra for the phosphopeptide sequences.

The output file name is inputfilename+modelmonomz.csv by default in the same folder as the input file.



The output file contains predicted spectra.

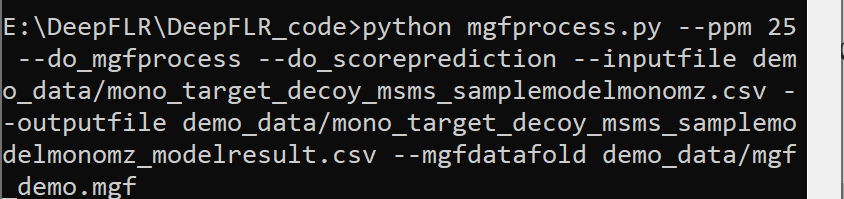
(5) Obtain the cosine similarity between the predicted spectra and the experimental spectra

python mgfprocess.py --ppm 25 --do\_mgfprocess --do\_scoreprediction --inputfile demo\_data/mono\_target\_decoy\_msms\_samplemodelmonomz.csv --outputfile demo\_data/mono\_target\_decoy\_msms\_samplemodelmonomz\_modelresult.csv --mgfdatafold demo\_data

The description of the parameters of the command line:

“--mgfdatafold”: the path containing the mgf files.

“--ppm”: the fragment tolerance in ppm (optional, default: 25).



The "score" column in the file represents the cosine similarity between predicted spectra and experimental spectra.

This command automatically produced JSON files from MGF files. If there are already JSON files for the MGF files or you don't need mgfprocess, you can speed up the process by omitting the “mgfprocess” step using the command:

python mgfprocess.py --ppm 25--do\_scoreprediction --inputfile demo\_data/mono\_target\_decoy\_msms\_samplemodelmonomz.csv --outputfile demo\_data/mono\_target\_decoy\_msms\_samplemodelmonomz\_modelresult.csv --mgfdatafold demo\_data

(6) Obtain the correlation between the delta score cutoff and the estimated FLR

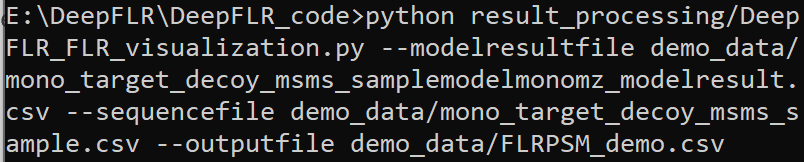
python result\_processing/DeepFLR\_FLR\_visualization.py --modelresultfile demo\_data/mono\_target\_decoy\_msms\_samplemodelmonomz\_modelresult.csv --sequencefile demo\_data/mono\_target\_decoy\_msms\_sample.csv --outputfile demo\_data/FLRPSM\_demo.csv

This command can be used to get DeepFLR FLR estimation.

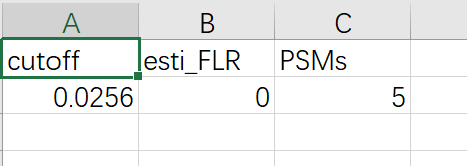
The description of the parameters of the command line:

“--modelresultfile”: the filename of the output file from mgfprocess.py.

“--sequencefile”: the filename of output file from Targetdecoy\_phosphopeptides\_generation\_{mono,multi}.py.



The output file contains the delta score, the corresponding estimated FLR and the number of PSMs obtained under the cutoff.



Here, “cutoff” means the delta score cutoff, “esti\_FLR” means the estimates FLR, “PSMs” means the number of PSMs.

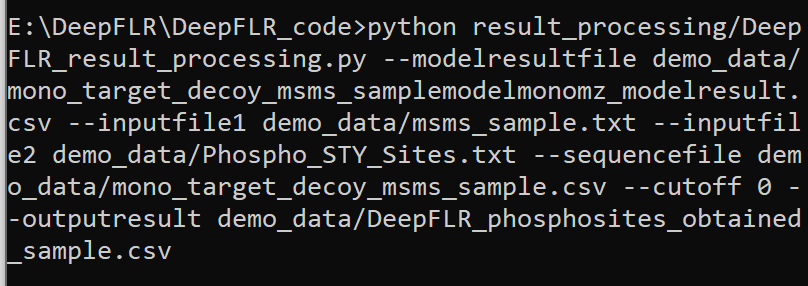
(7) Obtain the phosphosite localization information for desired estimated FLR

python result\_processing/DeepFLR\_result\_processing.py --modelresultfile demo\_data/mono\_target\_decoy\_msms\_samplemodelmonomz\_modelresult.csv --inputfile1 demo\_data/msms\_sample.txt --inputfile2 demo\_data/Phospho\_STY\_Sites.txt --sequencefile demo\_data/mono\_target\_decoy\_msms\_sample.csv --cutoff 0 --outputresult demo\_data/DeepFLR\_phosphosites\_obtained\_sample.csv

This command can be used to obtain the identification results from DeepFLR at a given estimated FLR.

The description of the parameters of the command line:

“--cutoff”: the delta score cutoff obtained from FLR determination/DeepFLR FLR visualization.py for target estimated FLR. You can visualize the correlation between the delta score and target estimated FLR from the step (6).



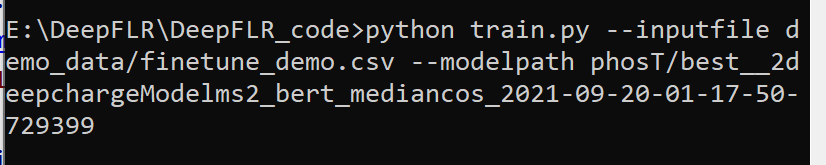
The " model\_proteinsite" column in the output file represents the phosphosites localized by DeepFLR.

1. **Fine-tune or re-train DeepFLR**

python train.py --inputfile demo\_data/finetune\_demo.csv --modelpath phosT/best\_\_2deepchargeModelms2\_bert\_mediancos\_2021-09-20-01-17-50-729399

This command is used to fine-tune DeepFLR.

This command will automatically turn the input file into the right format JSON for fine-tuning and the JSON file is in the same folder as the input file. Here, the input file (training or fine-tuning data) is in the SpectroMine type. You can also use searching results from other tools by modifying the format into SpectroMine type. “modelpath” refers to the path to the DeepFLR model. This model is provided in the folder “phosT”.



python train.py --inputfile demo\_data/finetune\_demo.csv --type train

This command can be used to re-train DeepFLR.

**All the demo, output files and code are provided. A sentence with gray background indicates that it is a sentence of code. If you have any further questions, please don’t hesitate to ask us via:** [**liang\_qiao@fudan.edu.cn**](mailto:liang_qiao@fudan.edu.cn)**.**