

# DeepDIA Demo: Training New Models for MS/MS and iRT Prediction

Training new models for MS/MS and iRT prediction using data-dependent acquisition (DDA) data.

## 1. System Requirements

This demo has been tested on a workstation with Intel Xeon E5-2690 v3 CPU, 16 GB RAM, and Microsoft Windows Server 2016 Version 1607 (OS Build 14393.2430) operating system with the following softwares:

- Anaconda 4.2.0 (Python 3.5.2).
- Keras 2.2.4 and TensorFlow 1.11.
- Microsoft R Open 3.5.1.
- RStudio 1.1.447.

A GPU card with Compute Unified Device Architecture (CUDA) is recommended, e.g. NVIDIA GeForce GTX 1050 Ti.

## 2. Demo Data

LC-MS/MS DDA data of HeLa cells on Q Exactive HF are available at ProteomeXchange

(<http://proteomecentral.proteomexchange.org/>) with the data set [PXD005573](#) . (Bruderer, R. et al. Mol. Cell. Proteomics 2017, 16, 2296-2309.)

- C\_D160304\_S251-Hela-2ug-2h\_MSG\_R01\_To.raw
- C\_D160304\_S251-Hela-2ug-2h\_MSG\_R02\_To.raw
- C\_D160304\_S251-Hela-2ug-2h\_MSG\_R03\_To.raw
- C\_D160331\_S209-HPRP-HeLa-05\_MSG\_R01\_To.raw
- C\_D160331\_S209-HPRP-HeLa-10\_MSG\_R01\_To.raw
- C\_D160331\_S209-HPRP-HeLa-15\_MSG\_R01\_To.raw
- C\_D160331\_S209-HPRP-HeLa-20\_MSG\_R01\_To.raw
- C\_D160331\_S209-HPRP-HeLa-25\_MSG\_R01\_To.raw
- C\_D160331\_S209-HPRP-HeLa-50\_MSG\_R01\_To.raw
- C\_D160331\_S209-HPRP-HeLa-FT\_MSG\_R01\_To.raw
- C\_D160401\_S209-HPRP-HeLa-05\_MSG\_R01\_To.raw
- C\_D160401\_S209-HPRP-HeLa-10\_MSG\_R01\_To.raw

- C\_D160401\_S209-HPRP-HeLa-15\_MSG\_R01\_To.raw
- C\_D160401\_S209-HPRP-HeLa-20\_MSG\_R01\_To.raw
- C\_D160401\_S209-HPRP-HeLa-25\_MSG\_R01\_To.raw
- C\_D160401\_S209-HPRP-HeLa-50\_MSG\_R01\_To.raw
- C\_D160401\_S209-HPRP-HeLa-FT\_MSG\_R01\_To.raw

SwissProt *Homo sapiens* database (FASTA) can be downloaded from UniProt (<https://www.uniprot.org/>). The FASTA file (2018-04 version, 20,301 entries)

has been deposited to ProteomeXchange via the iProX partner repository with the data set identifier [PXD014108/IPX0001628000](#) .

- swissprot\_human\_201804\_validated.fasta

The saved project and exported results from SpectroMine are also available at ProteomeXchange/iProX with identifier [PXD014108/IPX0001628000](#) .

- HeLa\_DDA.psar.zip
- HeLa\_DDA.csv.zip

### 3. Prepare Training Data

Training data can be converted from SpectroMine fragment reports (CSV). As an alternative, MaxQuant results ( [msms.txt](#) ) are also supported.

In this demo, SpectroMine reports are used, which should be exported with the schema provided in the [misc/SpectroMine\\_Report\\_Schema](#) folder.

- [FragmentReport.rs](#)

The file name of fragment report should end with [.FragmentReport.csv](#) , e.g. [HeLa.FragmentReport.csv](#) .

SpectroMine Manual is available at <https://biognosys.com/shop/spectromine>.

### 4. Train a MS/MS Model

Prepare an ions file for MS/MS prediction. An ions file can be converted from a SpectroMine fragment report.

Start RStudio, ensure packages [readr](#) and [rjson](#) have been installed.

```
install.packages("readr")
install.packages("rjson")
```

Set the working directory to the fragment report.

```
setwd("{PATH_TO_DATA}")
```

Open `deepms2/R/extract_ions_from_Spectronaut_report.R` and run the script by clicking **Source**.

```
source("{PATH_TO_CODE}/deepms2/R/extract_ions_from_Spectronaut_report.R")
```

We get two output ions files.

- HeLa\_charge2.ions.RData
- HeLa\_charge3.ions.RData

The screenshot shows the RStudio interface with the following components:

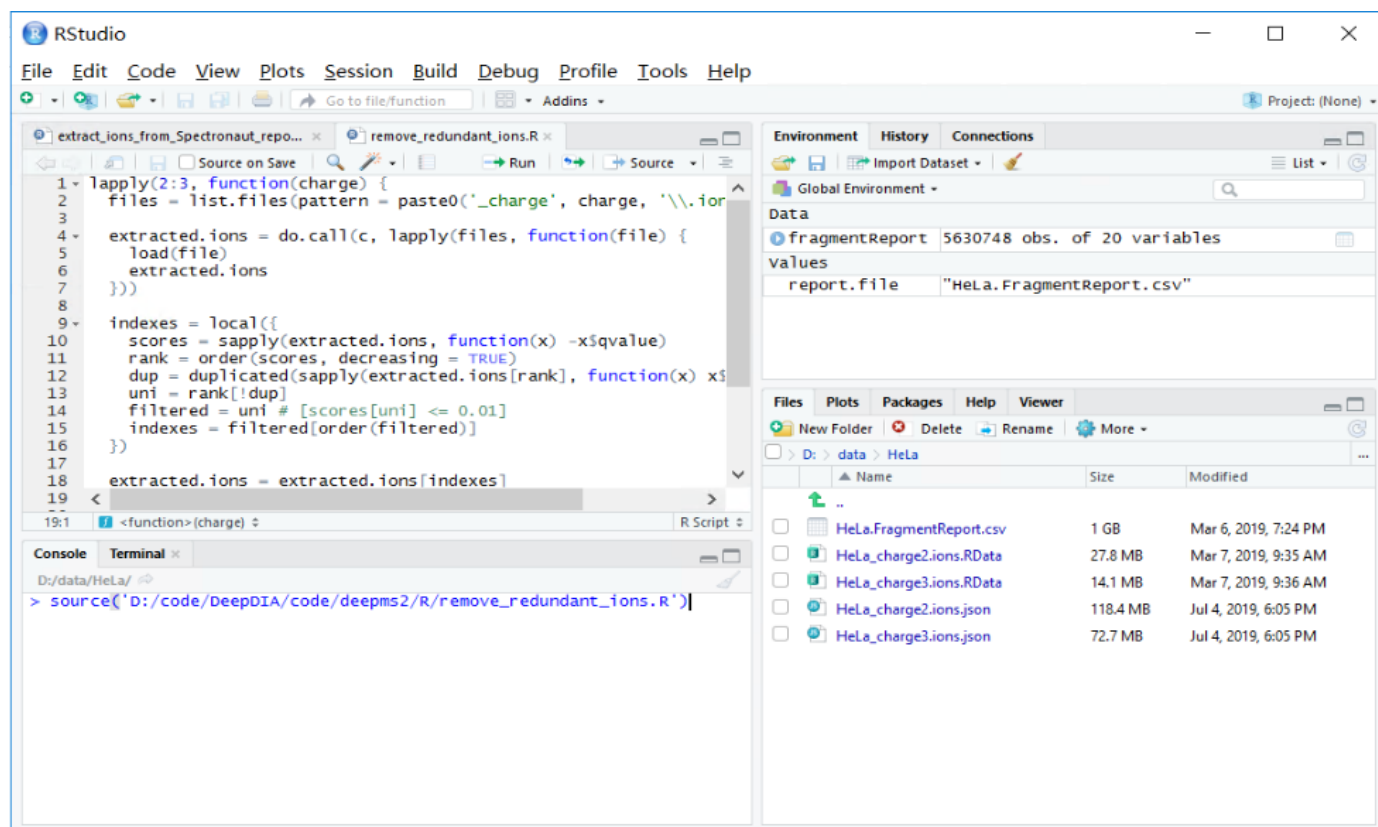
- Script Editor:** Displays the R script `extract_ions_from_Spectronaut_report.R`. The script uses `readr` to read a CSV file, filters for unique peptides based on PSM, MS2ScanNumber, PEP.StrippedSequence, PEP.Charge, PG.ProteinAccessions, and R.FileName, and then saves the results as `HeLa_charge2.ions.RData` and `HeLa_charge3.ions.RData`.
- Environment:** Shows the `fragmentReport` object with 5630748 observations and 20 variables. The `report.file` variable is set to `"HeLa.FragmentReport.csv"`.
- Files:** A file explorer showing the directory `D:/data/HeLa/` containing three files:

Name	Size	Modified
HeLa.FragmentReport.csv	1 GB	Mar 6, 2019, 7:24 PM
HeLa_charge2.ions.RData	27.8 MB	Mar 7, 2019, 9:35 AM
HeLa_charge3.ions.RData	14.1 MB	Mar 7, 2019, 9:36 AM
- Console:** Shows the execution of `setwd("D:/data/HeLa/")` and `source("D:/code/DeepDIA/code/deepms2/R/extract_ions_from_Spectronaut_report.R")`. It also displays the output of `col_types` for the `fragmentReport` object.

Run `deepms2/R/remove_redundant_ions.R` to get a unique MS/MS spectrum for each peptide.

We get two output ions files.

- HeLa\_charge2.ions.json
- HeLa\_charge3.ions.json



Move them into separate folders `charge2` and `charge3`

```
mkdir charge2
mv HeLa_charge2.ions.json charge2
mkdir charge3
mv HeLa_charge3.ions.json charge3
```

Run `deepms2/py/train.py` in the `charge2` directory.


```
cd charge2
python {PATH_TO_CODE}/deepms2/py/train.py
```

Expected run time depends on the number of peptide spectra and the performance of the computer. In this demo, this command may take several hours to a day.

In the `models` folder, we find the trained model (with checkpoints during training) for charge 2+ peptides.

(D:) > data > HeLa > charge2

名称	修改日期	类型	大小
models	2019/7/4 18:11	文件夹	
HeLa_charge2.ions.json	2019/7/4 18:05	JSON 文件	121,290 KB
training.log	2019/7/4 18:11	文本文档	0 KB

```
管理员: Windows PowerShell
FS.D:\data\HeLa\charge2> python D:\code\DeepDIA\code\deepms2\py\train.py
Using TensorFlow backend.
Train on 38321 samples, validate on 18876 samples
Epoch 1/100
2019-07-04 18:11:35.202298: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2
2019-07-04 18:11:36.182681: E tensorflow/core/grappler/optimizers/dependency_optimizer.cc:666] Iteration = 0, topological sort failed with message: The graph couldn't be sorted in topological order.
2019-07-04 18:11:36.209512: E tensorflow/core/grappler/optimizers/dependency_optimizer.cc:666] Iteration = 1, topological sort failed with message: The graph couldn't be sorted in topological order.
2019-07-04 18:11:36.460778: E tensorflow/core/grappler/optimizers/dependency_optimizer.cc:666] Iteration = 0, topological sort failed with message: The graph couldn't be sorted in topological order.
2019-07-04 18:11:36.485950: E tensorflow/core/grappler/optimizers/dependency_optimizer.cc:666] Iteration = 1, topological sort failed with message: The graph couldn't be sorted in topological order.
3296/38321 [=>.....] - ETA: 2:23 - loss: 0.0028 - cosine_similarity: -0.6618_
```

Train the model for charge 3+ following the same steps.

## 5. Train an iRT Model

Prepare an iRT file for iRT prediction.

An iRT file can be converted from a SpectroMine fragment report.

Start RStudio and set the working directory to the fragment report.

```
setwd("{PATH_TO_DATA}")
```

Open `deeprrt/R/extract_irt_from_Spectronaut_report.R` and run the script by clicking **Source**.

```
source("{PATH_TO_CODE}/deeprrt/R/extract_irt_from_Spectronaut_report.R")
```

We get the output iRT file.

- HeLa\_charge2.irt.csv

The screenshot shows the RStudio environment with the following components:

- Source Editor:** Contains an R script named `extract_irt_from_Spectronaut_report.R`. The script uses `readr` to read a CSV file, processes the data to remove duplicates, and creates a data frame with columns for sequence, iRT, and empirical iRT.
- Environment:** Shows two data objects: `irt` (69576 obs. of 3 variables) and `psmReport` (5630748 obs. of 20 variables). A value for `report.file` is shown as "HeLa.FragmentReport.csv".
- Files Panel:** Displays the file explorer for the `D:/data/HeLa` directory, showing `HeLa.FragmentReport.csv` (1 GB) and `HeLa.irt.csv` (2.9 MB).
- Console:** Shows the execution of `setwd` and `source` commands, followed by a message indicating the column specification for the CSV file.

Run `deeprt/py/train.py`.

```
python {PATH_TO_CODE}/deeprt/py/train.py
```

Expected run time depends on the number of peptide spectra and the performance of the computer. In this demo, this command may take several hours to a day.

The screenshot displays the file system and the execution of the training script:

- File Explorer:** Shows the `D:\data\HeLa` directory containing `models` (folder), `HeLa.FragmentReport.csv` (1,062,516 bytes), `HeLa.irt.csv` (2,972 KB), and `training.log` (0 KB).
- PowerShell Terminal:** Shows the command `python D:\code\DeepDIA\code\deeprt\py\train.py` being executed. The output indicates the use of TensorFlow, training on 46615 samples, and provides a progress bar for the first epoch.

In the `models` folder, we find the trained model (with checkpoints during training).