Install Quanformer

1. Create conda environment (Preinstall conda/miniconda)

```
conda create -n quanformer python=3.8
```

2. Activate environment

```
conda activate quanformer
```

3. Clone quanformer

Note: Make sure *checkpoint0029.pth* in /resources/ is normal (>300MB)。

```
git clone https://github.com/LinShuhaiLAB/QuanFormer.git
```

4. Install pytorch

Windows/Linux with NVIDIA GPU.

```
pip install torch==1.13.1+cu117 torchvision==0.14.1+cu117 torchaudio==0.13.1 --extra-index-url https://download.pytorch.org/whl/cu117
```

Linux without NVIDIA GPU.

```
pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cpu```
```

Windows/mac OS without NVIDIA GPU.

```
pip install torch torchvision torchaudio
```

5. Install requirements

```
cd QuanFormer
pip install -r requirements.txt
```

Use Quanformer

1. Parameter Descriptions

General parameters

- `--type`
  - **Default Value**: `'mzML'`
  - **Description**: Type of raw data files, currently only supports the mzML format.
- `--ppm`
  - **Default Value**: `10`
  - **Description**: PPM value for ROI extraction.
- `--source`
  - **Default Value**: `"resources/example"`
  - **Description**: Path to the raw data directory.
- `--feature`
  - **Default Value**: `"resources/test_feature.csv"`
  - **Description**: The path of the feature file. If it is not empty, the targeted mode will be used. If it is empty, it is the untargeted mode, and the parameters required for the untargeted mode need to be set.
- `--images_path`
  - **Default Value**: `"resources/example/output"`
  - **Description**: Path to the output ROI files.
- `--output`
  - **Default Value**: `"resources/example/output/area.csv"`
  - **Description**: Path to the output files.
- `--threshold`

    - **Default Value**: `0.99`
    - **Description**: Keep only predictions with 0.99 confidence.

- `--model`
    - **Default Value**: `"resources/checkpoint0029.pth"`
    - **Description**: Path to the peak detection model.
- `--roi_plot`
    - **Default Value**: `True`
    - **Description**: Whether to plot ROIs. Must be set to `True` on the first use.
- `--plot`
    - **Default Value**: `True`
    - **Description**: Whether to plot predictions.
- `--num_classes`
    - **Default Value**: `1`
    - **Description**: Number of classes.
- `--smooth_sigma`
    - **Default Value**: `0`
    - **Description**: Sigma value for smoothing.
- `--processes_number`
    - **Default Value**: `1`
    - **Description**: Number of processes.

Untargeted mode parameters for centWave algorithm.

- `--polarity`
    - **Default Value**: `'positive'`
    - **Description**: Polarity.
- `--minWidth`
    - **Default Value**: `5`

  - **Description**: Min peak width

- `--maxWidth`

  - **Default Value**: `50`
  - **Description**: Max peak width.

- `--s2n`

  - **Default Value**: `5`
  - **Description**: Signal-to-noise ratio.

- `--noise`

  - **Default Value**: `100`
  - **Description**: Noise level.

- `--mzDiff`

  - **Default Value**: `0.005`
  - **Description**: m/z difference.

- `--prefilter`

  - **Default Value**: `3`
  - **Description**: Pre-filtering parameter.

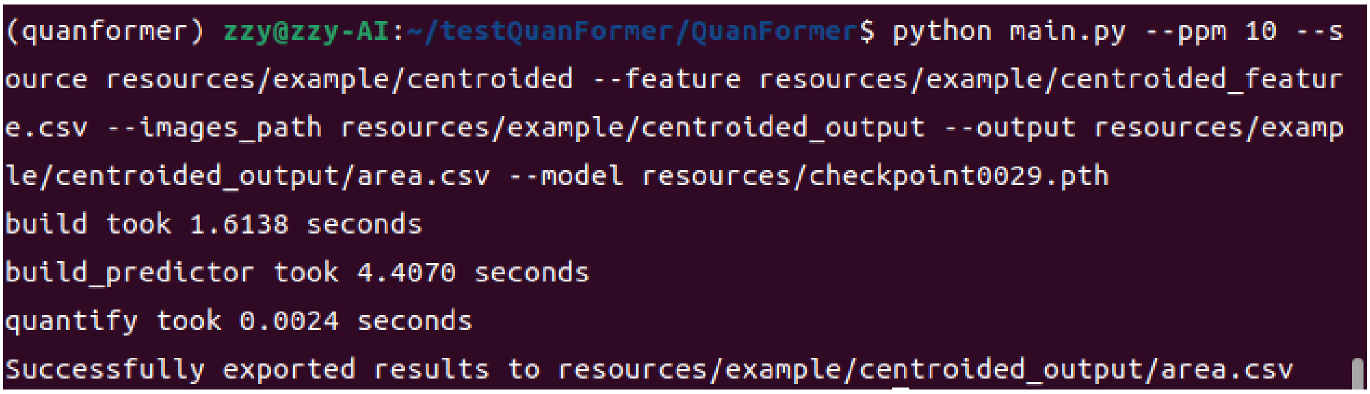
2. Run in command line mode.

2.1 Targeted Mode

Here is an example command showing how to use these parameters in **targeted mode**, quanformer can run in both centroided and profile data：

2.1.1 Centroided data

```
python main.py --ppm 10 --source resources/example/centroided --feature
resources/example/centroided_feature.csv --images_path
resources/example/centroided_output --output
resources/example/centroided_output/area.csv --model
resources/checkpoint0029.pth
```

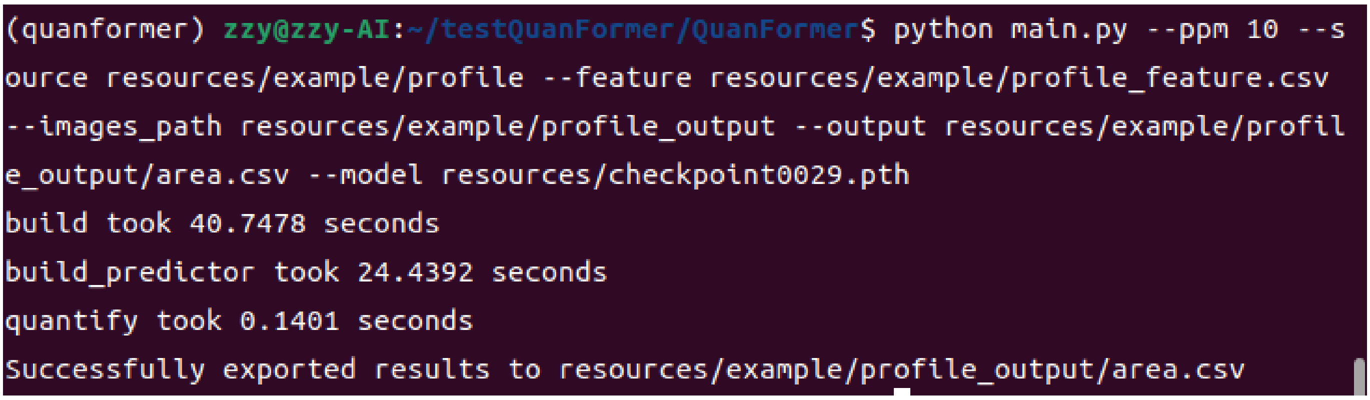


```
(quanformer) zzy@zzy-AI:~/testQuanFormer/QuanFormer$ python main.py --ppm 10 --s
ource resources/example/centroided --feature resources/example/centroided_featur
e.csv --images_path resources/example/centroided_output --output resources/examp
le/centroided_output/area.csv --model resources/checkpoint0029.pth
build took 1.6138 seconds
build_predictor took 4.4070 seconds
quantify took 0.0024 seconds
Successfully exported results to resources/example/centroided_output/area.csv
```

2.1.2 Profile data

example download link(https://drive.google.com/drive/folders/1JopRY0mgMxRGg45iXiBgbT-i7uG3M3tS?usp=drive_link)

```
python main.py --ppm 10 --source resources/example/profile --feature
resources/example/profile_feature.csv --images_path
resources/example/profile_output --output
resources/example/profile_output/area.csv --model resources/checkpoint0029.pth
```



```
(quanformer) zzy@zzy-AI:~/testQuanFormer/QuanFormer$ python main.py --ppm 10 --s
ource resources/example/profile --feature resources/example/profile_feature.csv
--images_path resources/example/profile_output --output resources/example/profil
e_output/area.csv --model resources/checkpoint0029.pth
build took 40.7478 seconds
build_predictor took 24.4392 seconds
quantify took 0.1401 seconds
Successfully exported results to resources/example/profile_output/area.csv
```

2.2 Install R before running in untargeted mode

R version 4.4.2, xcms version 4.4.0 In view of the possible problems in downloading R packages, I have packaged my R dependency packages and put them in the following link：(https://drive.google.com/file/d/1oEIANtyXztyRkKUcWpUh3jznCG2trHwv/view?usp=drive_link)

Before using untargeted mode, you should check whether R is installed, open the terminal and input:

```
R --version
```

If R and Rscript are not installed, they can be installed through the following commands. (https://cran.r-project.org/bin/linux/ubuntu/)

```
sudo apt-get update
 # update indices
sudo apt update -qq
# install two helper packages we need
sudo apt install --no-install-recommends software-properties-common dirmngr
# add the signing key (by Michael Rutter) for these repos
# To verify key, run gpg --show-keys /etc/apt/trusted.gpg.d/cran_ubuntu_key.asc
# Fingerprint: E298A3A825C0D65DFD57CBB651716619E084DAB9
wget -qO- https://cloud.r-project.org/bin/linux/ubuntu/marutter_pubkey.asc | sudo tee -a /etc/apt/trusted.gpg.d/cran_ubuntu_key.asc
# add the R 4.0 repo from CRAN -- adjust 'focal' to 'groovy' or 'bionic' as needed
sudo add-apt-repository "deb https://cloud.r-project.org/bin/linux/ubuntu $(lsb_release -cs)-cran40/"

sudo apt install --no-install-recommends r-base
sudo apt-get install libxml2-dev
```

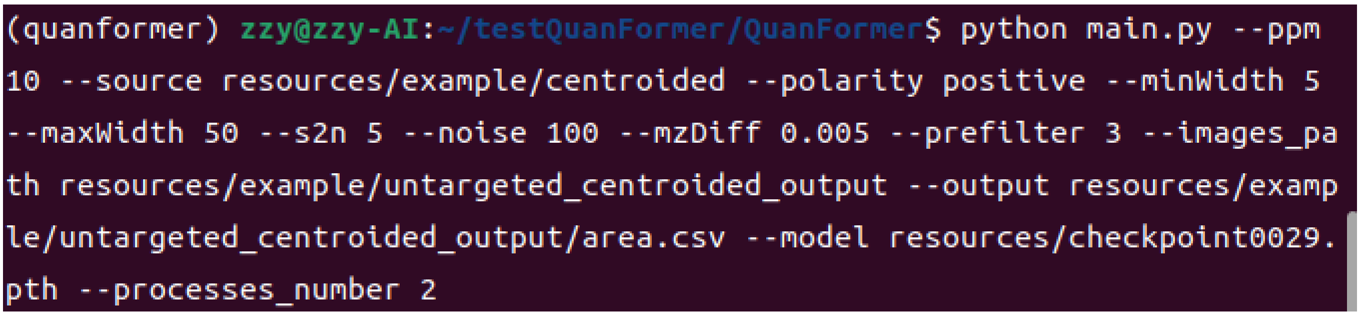
Then, run the following commands to install packages. (https://www.bioconductor.org/packages/release/bioc/html/xcms.html)

```
sudo R

if (!require("BiocManager", quietly = TRUE))
    install.packages("BiocManager")

BiocManager::install("xcms")
BiocManager::install("MSnbase")
install.packages("dplyr")
```

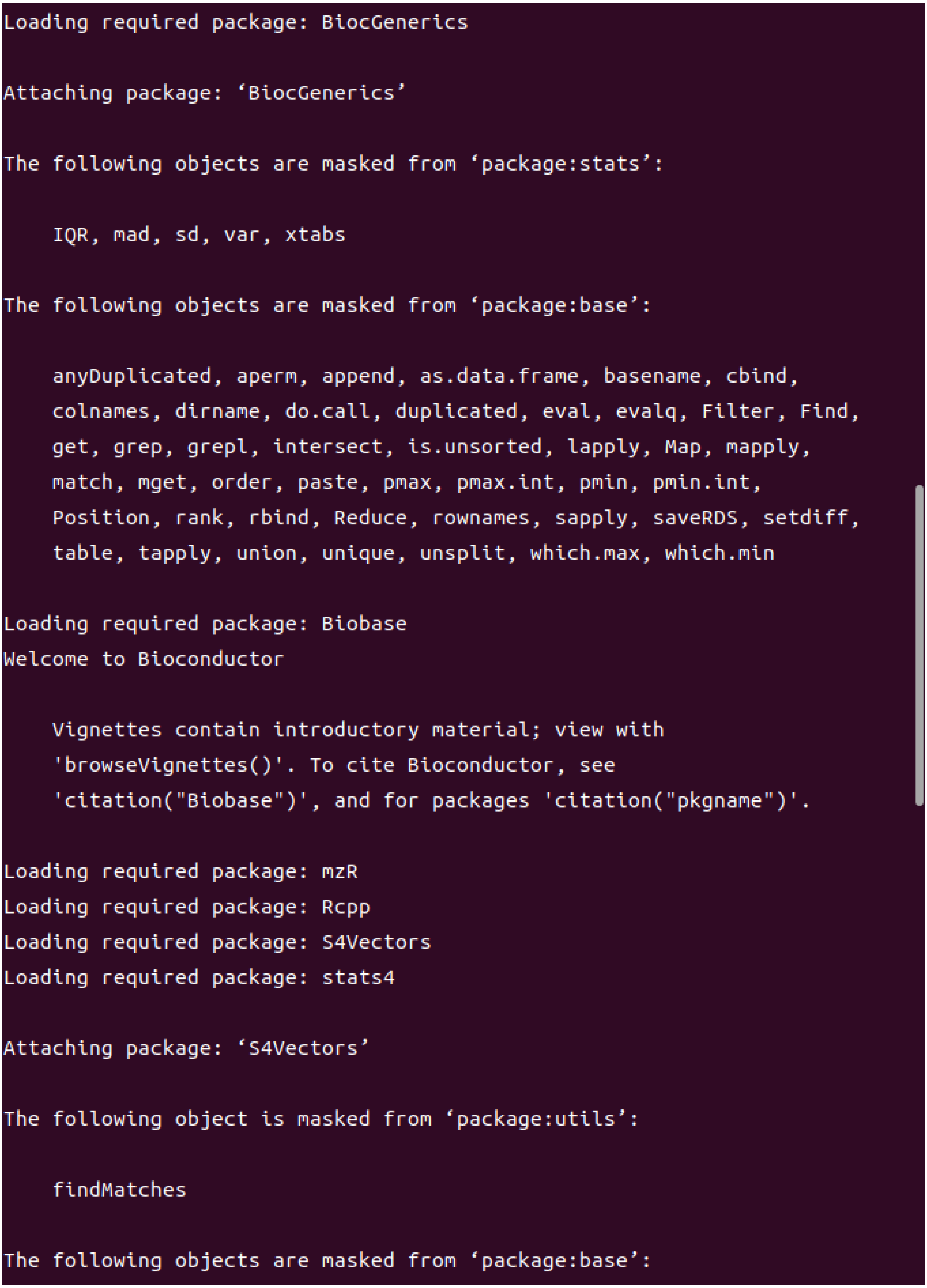
2.3 Untargeted Mode

Finally, to run the untargeted mode, the feature parameter needs to be set to empty or not set the feature parameter (default is empty), and at the same time, additional parameters required by the centWave algorithm such as polarity, peakWidth, s2n, noise, mzDiff, and prefilter need to be set. A complete command for running the untargeted mode is as follows:

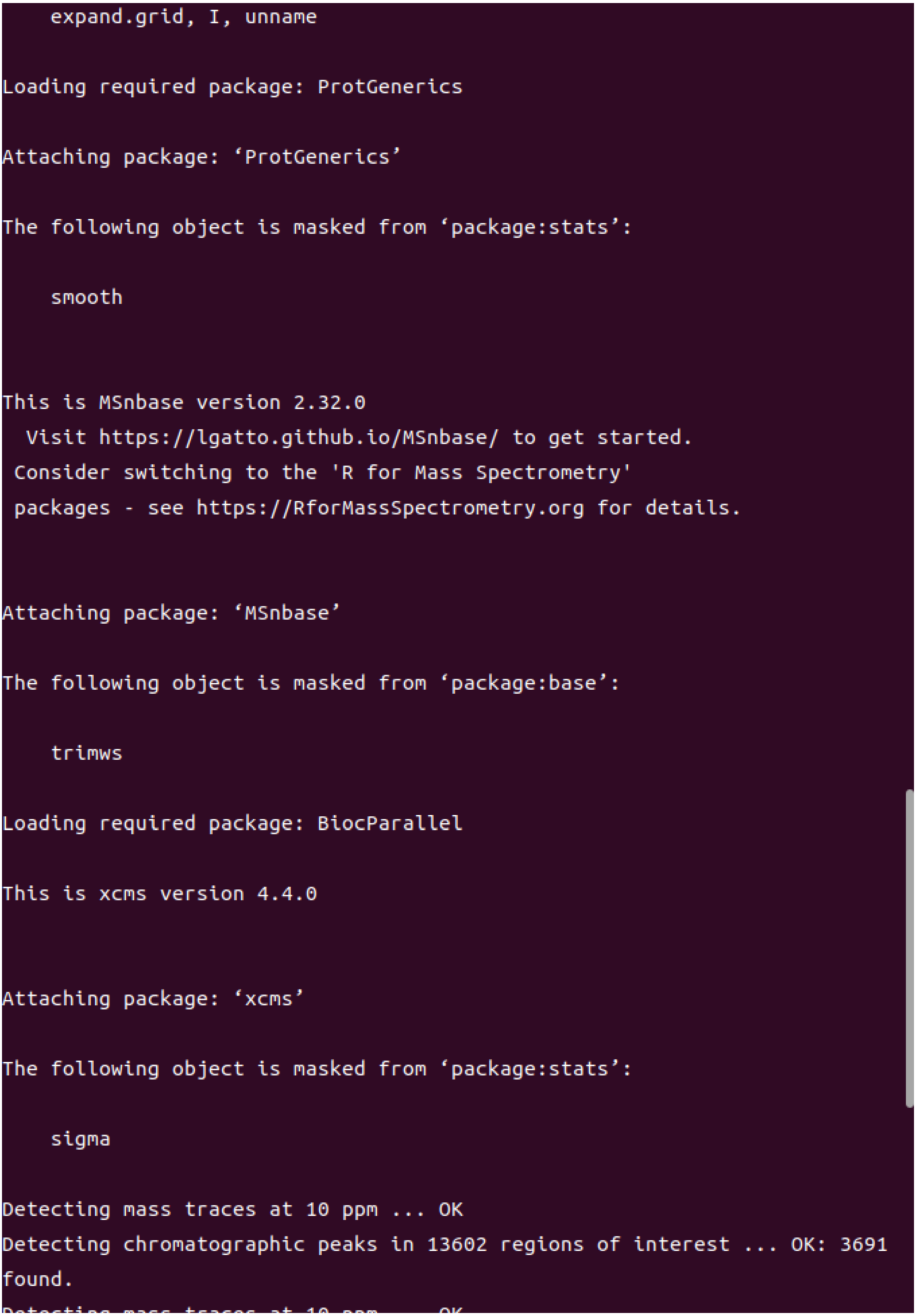
```
python main.py --ppm 10 --source resources/example/centroided --polarity
positive --minWidth 5 --maxWidth 50 --s2n 5 --noise 100 --mzDiff 0.005 --
prefilter 3 --images_path resources/example/untargeted_centroided_output --
output resources/example/untargeted_centroided_output/area.csv --model
resources/checkpoint0029.pth --processes_number 2
```



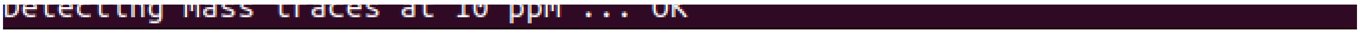
```
(quanformer) zzy@zzy-AI:~/testQuanFormer/QuanFormer$ python main.py --ppm
10 --source resources/example/centroided --polarity positive --minWidth 5
--maxWidth 50 --s2n 5 --noise 100 --mzDiff 0.005 --prefilter 3 --images_pa
th resources/example/untargeted_centroided_output --output resources/examp
le/untargeted_centroided_output/area.csv --model resources/checkpoint0029.
pth --processes_number 2
```



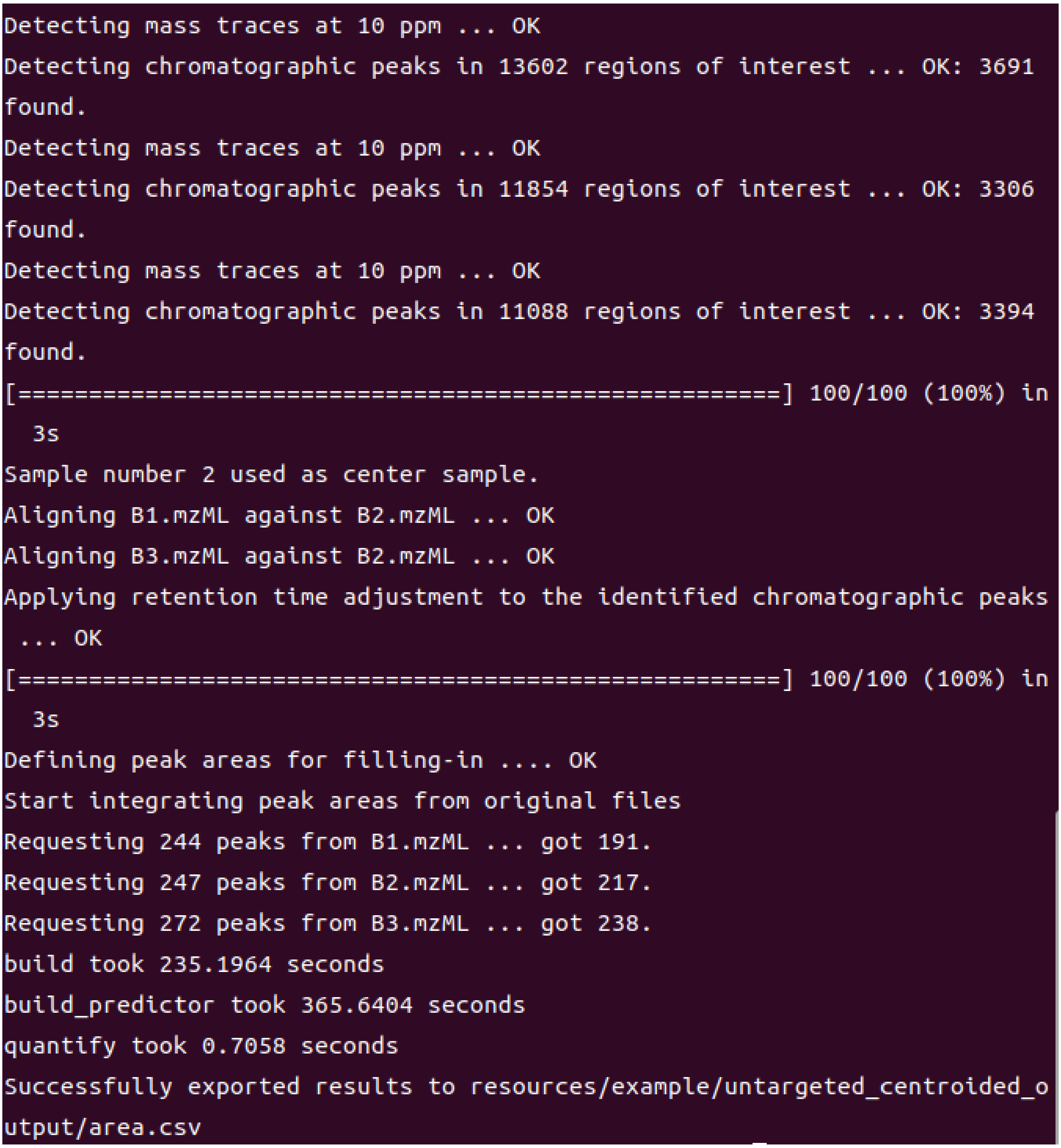
```
Loading required package: BiocGenerics

Attaching package: ‘BiocGenerics’

The following objects are masked from ‘package:stats’:

    IQR, mad, sd, var, xtabs

The following objects are masked from ‘package:base’:

    anyDuplicated, aperm, append, as.data.frame, basename, cbind,
    colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
    get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
    match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
    Position, rank, rbind, Reduce, rownames, sapply, saveRDS, setdiff,
    table, tapply, union, unique, unsplit, which.max, which.min

Loading required package: Biobase
Welcome to Bioconductor

    Vignettes contain introductory material; view with
    'browseVignettes()'. To cite Bioconductor, see
    'citation("Biobase")', and for packages 'citation("pkgname")'.

Loading required package: mzR
Loading required package: Rcpp
Loading required package: S4Vectors
Loading required package: stats4

Attaching package: ‘S4Vectors’

The following object is masked from ‘package:utils’:

    findMatches

The following objects are masked from ‘package:base’:
```



```
    expand.grid, I, unname

Loading required package: ProtGenerics

Attaching package: ‘ProtGenerics’

The following object is masked from ‘package:stats’:

    smooth


This is MSnbase version 2.32.0
  Visit https://lgatto.github.io/MSnbase/ to get started.
 Consider switching to the 'R for Mass Spectrometry'
 packages - see https://RforMassSpectrometry.org for details.


Attaching package: ‘MSnbase’

The following object is masked from ‘package:base’:

    trimws

Loading required package: BiocParallel

This is xcms version 4.4.0


Attaching package: ‘xcms’

The following object is masked from ‘package:stats’:

    sigma

Detecting mass traces at 10 ppm ... OK
Detecting chromatographic peaks in 13602 regions of interest ... OK: 3691
found.
Detecting mass traces at 10 ppm ... OK
```



```
Detecting mass traces at 10 ppm ... OK
```



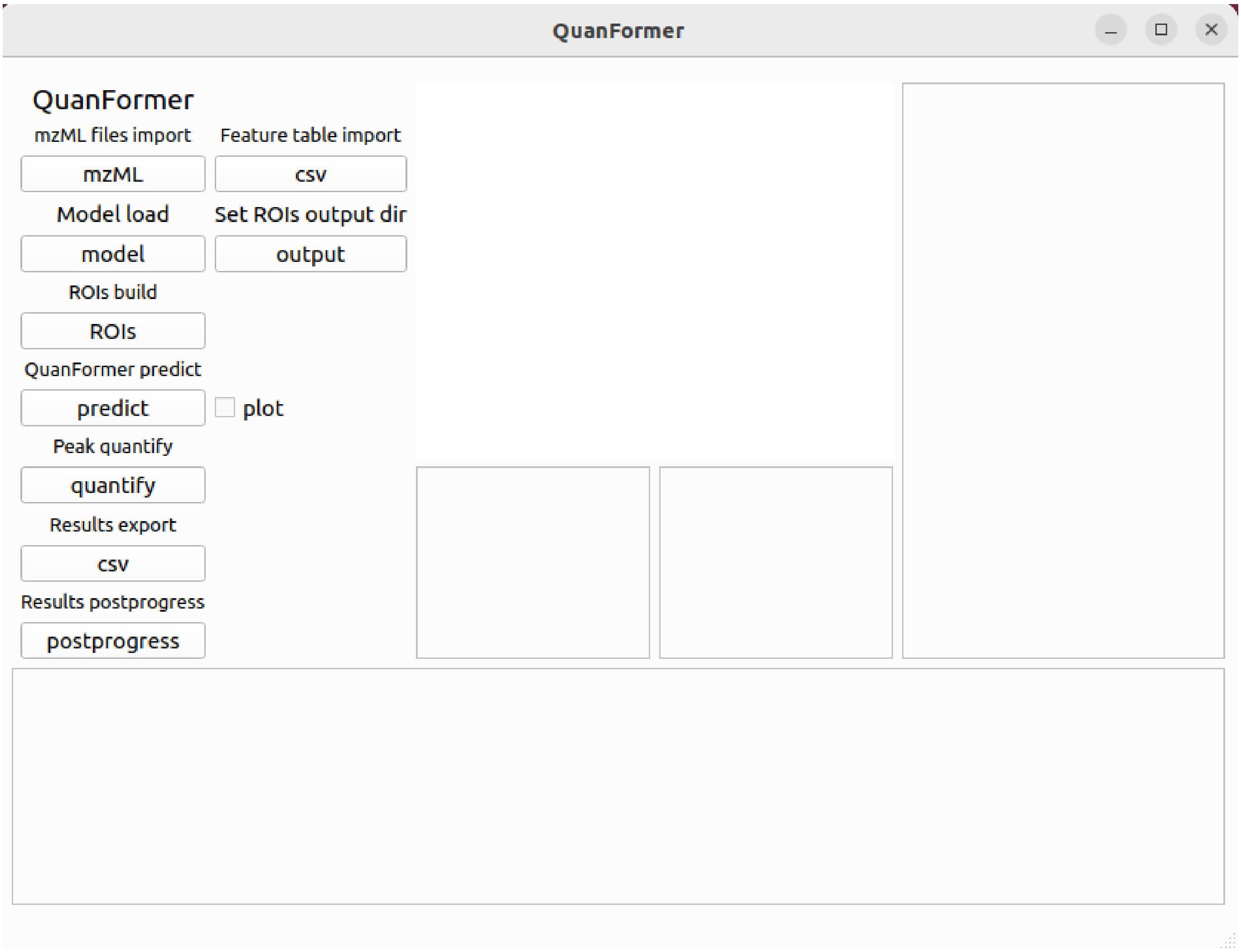
```
Detecting mass traces at 10 ppm ... OK
Detecting chromatographic peaks in 13602 regions of interest ... OK: 3691
found.
Detecting mass traces at 10 ppm ... OK
Detecting chromatographic peaks in 11854 regions of interest ... OK: 3306
found.
Detecting mass traces at 10 ppm ... OK
Detecting chromatographic peaks in 11088 regions of interest ... OK: 3394
found.
[======================================================] 100/100 (100%) in
  3s
Sample number 2 used as center sample.
Aligning B1.mzML against B2.mzML ... OK
Aligning B3.mzML against B2.mzML ... OK
Applying retention time adjustment to the identified chromatographic peaks
 ... OK
[======================================================] 100/100 (100%) in
  3s
Defining peak areas for filling-in .... OK
Start integrating peak areas from original files
Requesting 244 peaks from B1.mzML ... got 191.
Requesting 247 peaks from B2.mzML ... got 217.
Requesting 272 peaks from B3.mzML ... got 238.
build took 235.1964 seconds
build_predictor took 365.6404 seconds
quantify took 0.7058 seconds
Successfully exported results to resources/example/untargeted_centroided_o
utput/area.csv
```

Note: if "FileNotFoundError: [Errno 2] No such file or directory: 'sources/example/xcms_peak_list.csv'" appears in the terminal window, it usually means that the R environment or the dependent packages are not installed correctly. **Please return Step 2.2**.

3. Run in GUI mode

3.1 Targeted Mode

```
python GUI/ms-main.py
```



1. Click mzML button to choose the directory of mzML data, like* /resources/example/centroided*.

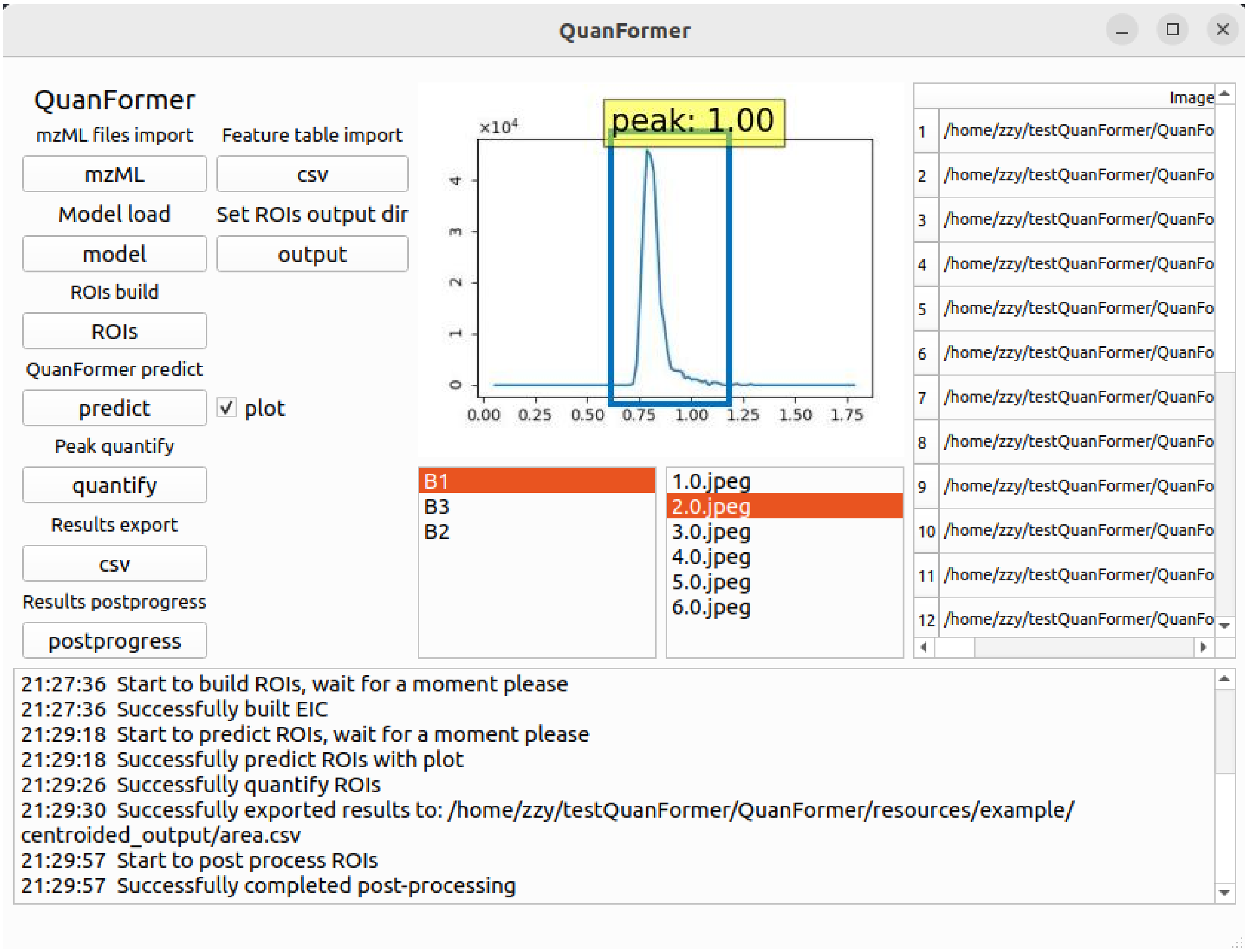
2. Click csv button to choose the feature table, like */resources/example/centroided_feature.csv.*

3. Click model button to load the model weight, like */resources/checkpoint0029.pth*, make sure the weight is normal (>300MB), or download the weights separately from the Github repository can solve the above problem. Just go to https://github.com/LinShuhaiLAB/QuanFormer/blob/main/resources/checkpoint0029.pth and click Download raw file button.

4. Click output button and choose an empty output folder.

5. Click ROIs button and wait for a moment.

6. Check the plot option(or not, reduce time), after clicking the "predict" button, wait for the log column to output "Successfully predict ROIs with plot". Click a different choice in the list column to display the results.


QuanFormer
QuanFormer
mzML files import
Feature table import
mzML
csv
Model load
Set ROIs output dir
model
output
ROIs build
ROIs
QuanFormer predict
predict
☑ plot
Peak quantify
quantify
Results export
csv
Results postprogress
postprogress
peak: 1.00
×10⁴
0
1
2
3
4
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
B1
B3
B2
1.0.jpeg
2.0.jpeg
3.0.jpeg
4.0.jpeg
5.0.jpeg
6.0.jpeg
Image
1 /home/zzy/testQuanFormer/QuanFo
2 /home/zzy/testQuanFormer/QuanFo
3 /home/zzy/testQuanFormer/QuanFo
4 /home/zzy/testQuanFormer/QuanFo
5 /home/zzy/testQuanFormer/QuanFo
6 /home/zzy/testQuanFormer/QuanFo
7 /home/zzy/testQuanFormer/QuanFo
8 /home/zzy/testQuanFormer/QuanFo
9 /home/zzy/testQuanFormer/QuanFo
10 /home/zzy/testQuanFormer/QuanFo
11 /home/zzy/testQuanFormer/QuanFo
12 /home/zzy/testQuanFormer/QuanFo

```
21:27:36  Start to build ROIs, wait for a moment please
21:27:36  Successfully built EIC
21:29:18  Start to predict ROIs, wait for a moment please
21:29:18  Successfully predict ROIs with plot
21:29:26  Successfully quantify ROIs
21:29:30  Successfully exported results to: /home/zzy/testQuanFormer/QuanFormer/resources/example/
centroided_output/area.csv
21:29:57  Start to post process ROIs
21:29:57  Successfully completed post-processing
```

7. Click quantify button.

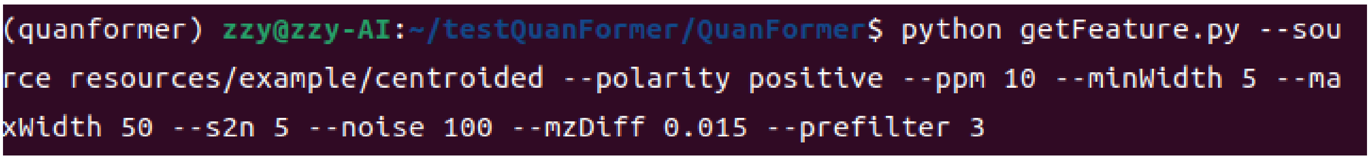
8. Click the csv button to create a new empty csv format file.

9. Click the postprogress button to progress the csv.

3.2 Untargeted Mode

In view of the fact that the centWave module requires additional environmental configuration and is relatively time-consuming during operation, and after searching for ROIs, what is finally obtained is a feature table that can be read by the feature button in the GUI mode. Therefore, we suggest to first run the ROIs search module based on the centWave algorithm in the command-line mode:

```
python getFeature.py --source resources/example/centroided --polarity positive
--ppm 10 --minWidth 5 --maxWidth 50 --s2n 5 --noise 100 --mzDiff 0.015 --
prefilter 3
```



```
(quanformer) zzy@zzy-AI:~/testQuanFormer/QuanFormer$ python getFeature.py --sou
rce resources/example/centroided --polarity positive --ppm 10 --minWidth 5 --ma
xWidth 50 --s2n 5 --noise 100 --mzDiff 0.015 --prefilter 3
```

After the operation is completed, a feature table in.csv format is generated (*./resources/peak_list.csv*). Then *python GUI/ms-main.py* and import it. Other operations are the same as targeted analysis (Step 3.1).