Tutorial on the usage of ECL-PF

Provided by Yu's Group
Laboratory for Bioinformatics and Computational Biology
ECE Department, HKUST, Hong Kong
Contact information:
czhouau@connect.ust.hk (ZHOU, Chen)

Introduction

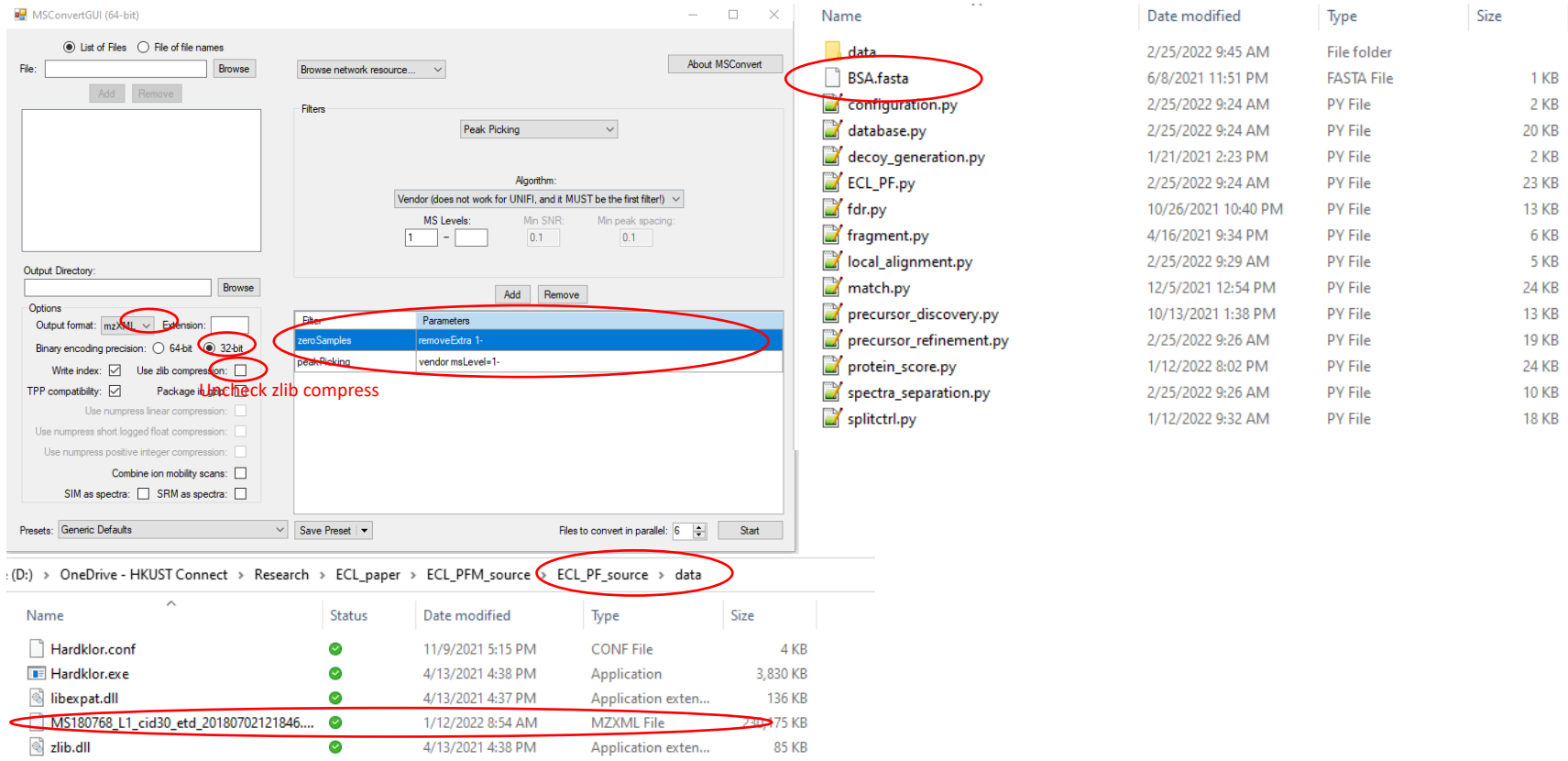
- ECL-PF is a powerful XL-MS tool designed specifically for cleavable data analysis.
- It is written in Python. Basic Python v3.6 or above environment is needed including numpy module, pyteomics module and lxml module. ECL-PF is running in command line for now. GUI version will be updated soon.
- Any question regarding to technical part should email to czhouau@connect.ust.hk (ZHOU, Chen)

Run with test data

- We provide a Dimethyl labeled BSA dataset to run a demo. CBDPS-light is used to cross-link the peptides. All the parameter setting is done in the configuration.py file.
- BSA.fasta and CID-MS2-ETD-MS2 data in .mzXML format are provided in test example.

Step 1

- Convert your data into .mzXML format and find a suitable FASTA file. User can choose MSConvert to transfer your file. Below is the concrete setting. After that, put your FASTA file under directory **root/** and your data (multiple files supported) under directory **root/data/**. Note that if you are using the text example provided by us, you don't need to convert the data format here.



Step 2

- Modify configuration.py file and run command **python configuration.py** to generate ECLPF_conf file.

```python
import json

'''this is the input configuration file'''

conf = {'parse_rule': 'trypsin',  # 'arg-c', 'asp-n', 'pepsin ph1.3', 'pepsin ph2.0' etc.
        'fasta_path': "BSA.fasta",  # path to your fasta file
        'activation_type': ['CID', 'ETD'],  # ['HCD', 'ETD'] or ['CID', 'ETD']
        'max_length': 50,  # maximum peptide length
        'min_length': 5,  # minimum peptide length
        'ms1_tol': 5e-6,  # MS1 mass tolerance in ppm
        'ms2_tol': 2e-5,  # MS2 mass tolerance in ppm
        'miss_cleavage': 2,  # allowed missed cleavage in peptides
        'num_max_mod': 3,  # maximun number of modification allowed, exclude fixed modificati
        'link_site': ['K', '['],  # support multiple sites such as ['K', 'R','[']. '[' means
        'fix_mod': {'car': [57.021464, ['C']]},  # fixed modification. 'car' is the modificat

        'var_mod': {'28': [28.0313, ['K', 'Peptide-nterm']],
                    '34': [34.0631, ['K', 'Peptide-nterm']],
                    'oxi': [15.9949, ['M']]},  # variable modification. 'oxi' is the modifica

        'xl_mass': 509.097,  # dsbu 196.0848, dsso 158.0038, dsbso 308.0388, cbdps 509.097
        'm_short': 54.011,  # dsbu 85.0528, dsso 54.0106, dsbso 54.0106, cbdps 54.011
        'm_long': 455.086}  # dsbu 111.032, dsso 85.9826/103.9932, dsbso 236.0177/254.0283, c

with open('ECLPF_conf', 'w') as f:
    json.dump(conf, f, indent=4, separators=(',', ':'))
```

decoy_generation.py	1/21/2021 2:23 PM	PY File	2 KB
ECL_PF.py	2/25/2022 9:24 AM	PY File	23 KB
ECLPF_conf	2/25/2022 9:50 AM	File	1 KB
fdr.py	10/26/2021 10:40 PM	PY File	13 KB
fragment.py	4/16/2021 9:34 PM	PY File	6 KB
local_alignment.py	2/25/2022 9:20 AM	PY File	5 KB

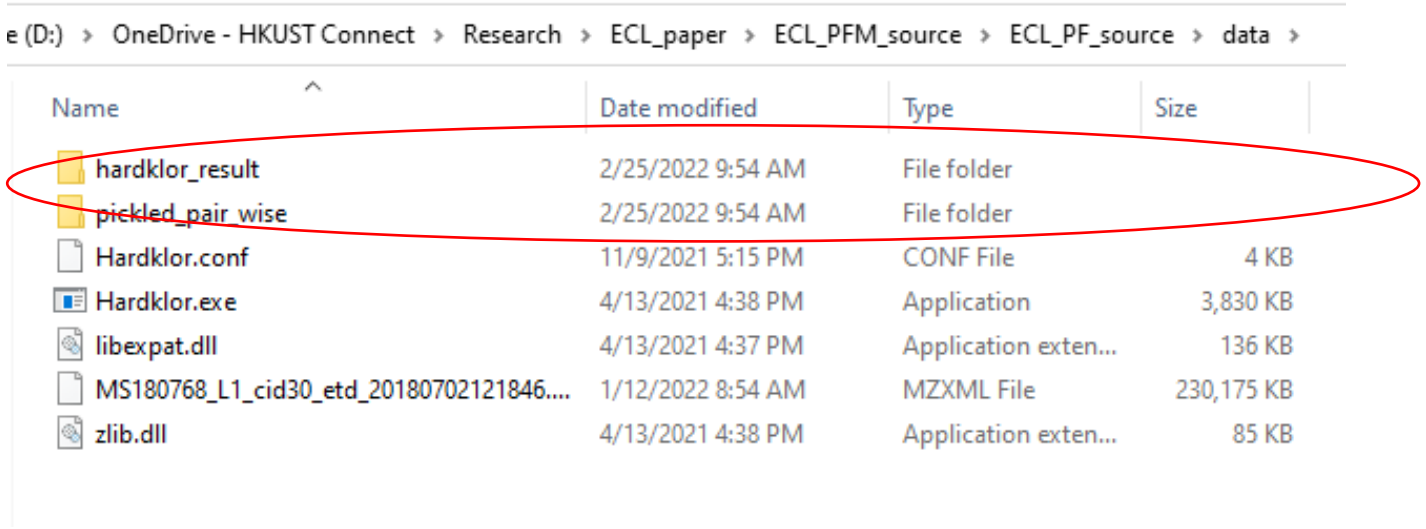
Step 3

- Generate database by running **python database.py**. Then, database directory and protein_name file will generate, and you don't need to run it again next time when you have the same type of data.

Name	Date modified	Type	Size
__pycache__	2/25/2022 9:52 AM	File folder	
database_file	2/25/2022 9:52 AM	File folder	
data	2/25/2022 9:45 AM	File folder	
BSA_decoy.fasta	2/25/2022 9:52 AM	FASTA File	2 KB
database.log	2/25/2022 9:52 AM	Text Document	1 KB
protein_name	2/25/2022 9:52 AM	File	1 KB
BSA.fasta	6/8/2021 11:51 PM	FASTA File	1 KB
configuration.py	2/25/2022 9:24 AM	PY File	2 KB
database.py	2/25/2022 9:24 AM	PY File	20 KB
decoy_generation.py	1/21/2021 2:23 PM	PY File	2 KB
ECL_PF.py	2/25/2022 9:24 AM	PY File	23 KB
ECLPF_conf	2/25/2022 9:50 AM	File	1 KB
fdr.py	10/26/2021 10:40 PM	PY File	13 KB
fragment.py	4/16/2021 9:34 PM	PY File	6 KB
local_alignment.py	2/25/2022 9:29 AM	PY File	5 KB
match.py	12/5/2021 12:54 PM	PY File	24 KB
precursor_discovery.py	10/13/2021 1:38 PM	PY File	13 KB
precursor_refinement.py	2/25/2022 9:26 AM	PY File	19 KB
protein_score.py	1/12/2022 8:02 PM	PY File	24 KB
spectra_separation.py	2/25/2022 9:26 AM	PY File	10 KB
splitctrl.py	1/12/2022 9:32 AM	PY File	18 KB

Step 4

- Run command **python spectra_separation.py** to process and deisotope data. Two more directories (hardklor_result and pickle_pair_wise) will generate under directory **root/data/.**



Step 5

- Run command **python precursor_refinement.py** and **python local_alignment.py**. In this step, you won't observe any change, but the data content has been modified. This step is optional but is strongly recommended.

Step 6

- Run command **python ECL_PF** to match the peptides.

Name	Date modified	Type	Size
__pycache__	2/25/2022 9:56 AM	File folder	
data	2/25/2022 9:54 AM	File folder	
database_file	2/25/2022 9:52 AM	File folder	
BSA.fasta	6/8/2021 11:51 PM	FASTA File	1 KB
BSA_decoy.fasta	2/25/2022 9:52 AM	FASTA File	2 KB
configuration.py	2/25/2022 9:24 AM	PY File	2 KB
database.log	2/25/2022 9:57 AM	Text Document	1 KB
database.py	2/25/2022 9:24 AM	PY File	20 KB
decoy_generation.py	1/21/2021 2:23 PM	PY File	2 KB
ECL_PF.py	2/25/2022 9:24 AM	PY File	23 KB
ECLPF_conf	2/25/2022 9:50 AM	File	1 KB
fdr.py	10/26/2021 10:40 PM	PY File	13 KB
fragment.py	4/16/2021 9:34 PM	PY File	6 KB
local_alignment.py	2/25/2022 9:29 AM	PY File	5 KB
match.py	12/5/2021 12:54 PM	PY File	24 KB
MS180768_L1_cid30_etd_20180702121846.csv	2/25/2022 9:57 AM	Microsoft Excel C...	48 KB
precursor_discovery.py	10/13/2021 1:38 PM	PY File	13 KB
precursor_refinement.py	2/25/2022 9:26 AM	PY File	19 KB
protein_name	2/25/2022 9:52 AM	File	1 KB
protein_score.py	1/12/2022 8:02 PM	PY File	24 KB
spectra_separation.py	2/25/2022 9:26 AM	PY File	10 KB
splitctrl.py	1/12/2022 9:32 AM	PY File	18 KB

Step 7

- Run command **python splitctrl.py result_file.csv fdr** to filter the result, e.g., **python splitctrl.py MS180768_L1_cid30_etd_20180702121846.csv 0.01**. The FDR cut-off is set in the last step so that you can change it to observe different FDR cut-off result.

Name	Date modified	Type	Size
__pycache__	2/25/2022 9:56 AM	File folder	
data	2/25/2022 9:54 AM	File folder	
database_file	2/25/2022 9:52 AM	File folder	
BSA.fasta	6/8/2021 11:51 PM	FASTA File	1 KB
BSA_decoy.fasta	2/25/2022 9:52 AM	FASTA File	2 KB
configuration.py	2/25/2022 9:24 AM	PY File	2 KB
database.log	2/25/2022 9:57 AM	Text Document	1 KB
database.py	2/25/2022 9:24 AM	PY File	20 KB
decoy_generation.py	1/21/2021 2:23 PM	PY File	2 KB
ECL_PF.py	2/25/2022 9:24 AM	PY File	23 KB
ECLPF_conf	2/25/2022 9:50 AM	File	1 KB
fdr.py	10/26/2021 10:40 PM	PY File	13 KB
fragment.py	4/16/2021 9:34 PM	PY File	6 KB
local_alignment.py	2/25/2022 9:29 AM	PY File	5 KB
match.py	12/5/2021 12:54 PM	PY File	24 KB
MS180768_L1_cid30_etd_20180702121846.csv	2/25/2022 9:57 AM	Microsoft Excel C...	48 KB
MS180768_L1_cid30_etd_20180702121846_final.csv	2/25/2022 9:58 AM	Microsoft Excel C...	18 KB
precursor_discovery.py	10/13/2021 1:38 PM	PY File	13 KB
precursor_refinement.py	2/25/2022 9:26 AM	PY File	19 KB
protein_name	2/25/2022 9:52 AM	File	1 KB
protein_score.py	1/12/2022 8:02 PM	PY File	24 KB
spectra_separation.py	2/25/2022 9:26 AM	PY File	10 KB
splitctrl.py	1/12/2022 9:32 AM	PY File	18 KB

Thanks for using ECL-PF!