PRIDE Data Downloader

This documentation provides basic information on the usage and design of the PRIDE Data Downloader and Converter. The PRIDE Data Donwloader is a tool implemented in Python 3.7 that offers access to PRIDE projects and converts the contained PSMs into a portable .csv format.

I. INTRODUCTION

The tool is used as a command line utility with the goal of downloading, extracting, parsing and converting PSMs from PRIDE projects. The user is able to perform feature selection on the PRIDE database. Configuration of the tools parameters is accomplished by utilizing the command line arguments or the accompanied .ini configuration file.

II. CONCEPT

The tool first requests the specified number of projects with the desired settings from the PRIDE archive utilizing the PRIDE RESTful API. The json response is decoded and the enclosed download links for the individual assays are extracted. The files (mzid and mgf tuples) are downloaded and saved with the json description of the project. Afterwards the parsing and processing of the downloaded file tuples starts. The MZID and MGF Parser extract the relevant features which are matched to form a CSV file containing the calculated data (y and b series matches, length calculations, modifications per length ...) for each PSM.

III. IMPLEMENTATION

The tool is implemented in Python 3.7 and utilizes a variety of additional modules to perform the tasks described in the concept. The MZID parser is implemented utilizing the xml.sax module of Python. MGF are parsed by evaluating the content of each line. The tool is able to utilize multiple processes for generating the CSV files. Multiprocessing is implemented with a data parallel approach and the multiprocessing module of Python.

IV. REQUIREMENTS

Besides the offical modules provided by Python the downloader also utilizes external modules available through PyPi.

External Package	Usage	URL
Pyteomics	Provides the masses of molecules and amino acids for the CSV calculations.	`https://pyteomics.readthedocs.io/en/latest/`
numba	Offers JIT-compiled functions in llvm for a significant performance increase.	`https://numba.pydata.org/`

V. CONFIGURATION

The tool can be configured by utilizing the command line parameters or the config.json file. The tool also utilizes a modifications.csv in the top level folder of the downloader. This file contains all modifications that are either valid or invalid for processing with this tool. During the execution of the tool a debug.log is created by the Python logging module.

Configuration by JSON

```
{
"accessions": ["PXD007612"],
"csv": true,
"memory": 0.9,
"number": 20,
"pages": 1,
"single_file": false,
"ini": true,
"instruments": null,
"json": false,
"species": null,
"folder": "data_pride",
"submission": "COMPLETE",
"cores": 4,
"features": ["Charge", "sumI",
    "norm_high_peak_intensity",
    "Num_of_Modifications",
    "Pep_Len", "Num_Pl", "mh(group)", "mh(domain)"
    "uniqueDM", "uniqueDMppm",
    "Sum_match_intensities",
    "Log_sum_match_intensity", "b+_ratio",
    "b++_ratio", "y+_ratio", "y++_ratio",
    "b+_count", "b++_count", "y+_count",
    "y++_count", "b+_long_count",
    "b++_long_count", "y+_long_count",
    "y++_long_count",
    "median_matched_frag_ion_errors",
    "mean_matched_frag_ion_errors",
    "iqr_matched_frag_ion_errors", "Class_Label",
    "ClassLabel_Decision"]
}
```

A. Command line arguments

Execution with command line arguments!

```
python app.py —<Argument> –<Short>
```

For the execution with command line arguments two alternative arguments are provided. A short argument and a long argument. These arguments are interchangable and can be applied in any desirable order.

Argmuent	Short	Description
accessions	A	Specify certain projects by accessions to download.
csv	C	Generates a csv file for each available file tuple!
memory	M	Limits the RAM for the program to the given ratio of the available RAM!
number	N	Maximal number of projects per page with fitting metadata to include.
pages	P	Maximal number of project pages to search.
single_file	O	Only download a single file tuple for each available project!
instruments	I	MS/MS instruments used in projects. String used by PRIDE
json	J	Generates a json file from each available project!
species	S	Species evaluated in projects. NCBI Taxonomy ID
folder	F	Folder containing downloaded data relative to the python script!
submission	SUB	SubmissionType for projects.
ini	INI	Executes the setting in the config.ini and disregards command line arguments!
cores	CO	Maximum number of cores.
features	FE	List of features columns to include in csv.

VI. EXAMPLES

Executing a pride download with the parameters specified in config/config.json .

```
python app.py –INI
```

Executing a pride search (filtering 1 page of 20 entries) and download a single file with csv generation.

```
python app.py –C –N 20 –P 1 –O
python app.py ––csv ––number 20 ––page 1 ––single
```

Executing a single file pride download for a specific accession with csv generation.

```
python app.py –A PXD007612 –C –O
```