**PROMISE pipeline (Protein modification integrated search engine)**

PROMISE is a distributed pipeline to detect post-translation modification on Mass Spectrometry data. The pipeline is solely for academic research, non-commercial or educational purposes; for other uses, please contact Weizmann institution.

The pipeline is implemented to work on HPC environment with LSF API. The following LSF commands should be supported: bsub, bjobs, bpeek

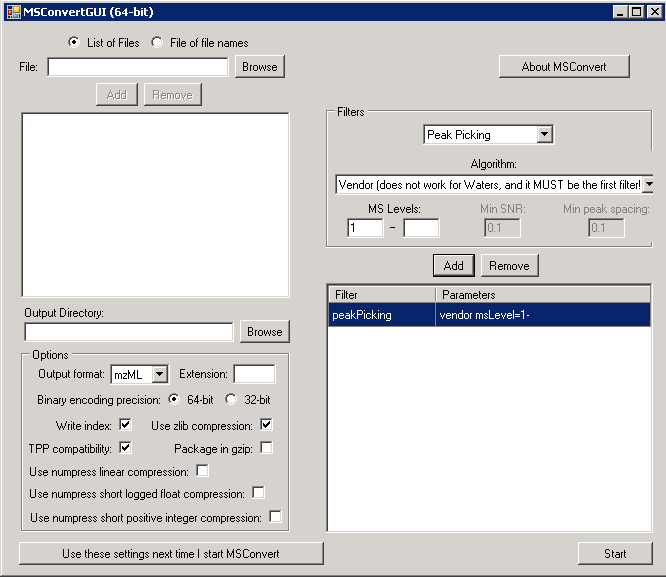
The user should update all the hard-coded links in the code for external software used by the pipeline.

The pipeline was implemented and tested on python 3.6.0

For pipeline description, please refer to the publication: “**Post-translational modifications reshape the antigenic landscape of MHC I-immunopeptidome in tumors”** in nature biotechnology

**Convert raw files to mzML format**

Installation: http://proteowizard.sourceforge.net/download.html



**Run matching phase**

**General pipeline description**

The peptidomics pipeline contains 3 steps:

1. Split MSFragger tasks and run them, dynamically re-split them as needed. The user should supply fragger params

2. Merge success MSFragger outputs

3. Run Philosopher for FDR and quantification

**Commands**

*change directory to your working directory with the mzML files*

**peptidomicsPipeline.py <working directory> <modification file> <fasta>**

*example:*

**python3 peptidomicsPipeline.py . ./modifications.txt ../../database/Human\_SP\_27042020.fasta -params ./fragger.params**

**flags:**

-digest [Tryptic , non-specific] *default is non-specific*

*Will determine which fragger.params to take, and which FDR for protein level.*

-init [integer 1-10] default 1

Determine the initial split, 1 equal regular MSFragger

-params <path to fragger.params file>

*Will override the default params files*

-merge <path to task\_report.csv file>

*Will continue to merge and philosopher tasks according to the success MSFragger tasks listed in the task report file*

-JVM <number of Gbyte>

*MAPP required at list 60, 1D – 20, default is 60*

-queue <queue name>

*Your own HPC queue for the bsub command*

-fdr <float number up to 1>

*Will be used for peptide fdr, default 0.01*

*-redo <path to task report file>*

*If the pipeline crashed, you can rerun it with the redo flag and the latest task report, located in the nobackup temporary folder.*

**Modification file format:**

Syntax:

\* is used to represent any amino acid

[ is a modifier for protein N-terminal

] is a modifier for protein C-terminal

n is a modifier for peptide N-terminal

c is a modifier for peptide C-terminal

Syntax Examples: 15.9949 M (for oxidation on methionine)

79.66331 STY (for phosphorylation)

-17.0265 nQnC (for pyro-Glu or loss of ammonia at peptide N-terminal)

Example:

B:15.99490 M 3,42.01060 [^ 1

P:79.96633 ST 2

Y:79.96633 Y 2

* It has to be at least 2 lines!!
* Identifier – one letter!! (not case sensitive)
* Up to 3 definitions per line
* Recommended – 2 amino acids per line
* The number represents the maximum modification of this type in the same peptide. This limitation reduces the number of theoretical peptides and helps control FDR overfitting issues

**How to check if pipeline run succeeds:**

Check in the log folder for file MSFragger\_tasks\_summary.xlsx

**Run prioritizing phase:**

**parallelModificationAnalysing.py <psm file> <output file> <fasta>**

Example:

python3 /home/labs/yifatlab/assafk/PROMISE/prioritizing/parallelModificationAnalysing.py ./psm.tsv ./modified\_psm.csv /home/labs/yifatlab/assafk/modification/PROMISE\_test/Human\_SP\_27042020-with-contaminate-with-decoy.fasta

Flags:

-queue <queue name>

*Can be “Merbl” or “new-short”, Merbl is our queue and has priority; default is new-short*

-batch <int number>

*Default 500*

*-filter <string>*

*Explain what should not be considered as a modification. For example:*

*-filter "n(42.0106)" : if you want to ignore n-terminus acetylation*

*-filter "n(145.1000),C(57.0215),K(145.1000)" : if you want to ignore Carbamidomethyl, and ITRAQ*

*-filter “C(57.0215)” : should be done in any tryptic analysis*

*-mode [new,append]*

*Append means start where you finished*

*-mem <int>*

*Size of memory assigned to the bsub command in the LSF. Default is 8G*

*-annotation <string>*

*Define the type of modification when searching for alternative modification in step 10. Can receive multiple types separated by comma, default = general. For example:*

*-annotation general: will use only common modifications like acetylation, methylation, phosphorylation, ubiquitination, etc.*

*-annotation general, specific: will also use c terminus alanine tail as a potential modification.*

*The modification type are defined in file X:\assafk\modification\* *modification-masses2.xlsx*

**How to check if pipeline run succeeds:**

All temporary files were deleted

**Troubleshooting**

Check the Analyzed log to find problems