

ReFrag

Table Of Contents

What is ReFrag	1
Quick Start Guide	1
Installation	2
Input Files.....	2
Running ReFrag	3
Parameters.....	4
Understanding ReFrag Output	6

What is ReFrag

ReFrag is an implementation of the ReCom concept ([Laguillo-Gómez et al., 2023](#)) designed to be run as a post-processing step after an open MSFragger search. It is compatible with both DDA and DIA data. When using ReFrag, DIA data can be searched in a “pseudo-DDA” workflow, using a curated list of theoretical Δ mass values to correct errors caused by the uncertainty in precursor masses contained within the same fragmentation window.

ReFrag has been developed at the Cardiovascular Proteomics Lab / Proteomics Unit at CNIC (Spanish National Centre for Cardiovascular Research). For support please consult this documentation, submit an [issue on GitHub](#), or contact the developer via e-mail: andrea (dot) laguillo (at) cnic (dot) es.

Quick Start Guide

This section provides a brief overview of the basic requirements to run ReFrag. If this is your first time using ReFrag, or if you’re looking for an in-depth explanation of any step, please refer to the full documentation in the following pages.

1. Download the latest release from the [GitHub page](#).
2. Install Python 3.11 and the packages from `python_requirements.txt`.
3. Prepare your `ReFrag.ini` file. Check tolerance values, column names and fixed modifications.
4. Run ReFrag from the command line:

```
$ python ReFrag.py -i [MSFragger files] -r [MGF or mzML files] -d
[Theoretical  $\Delta$ masses file] -c [Configuration file]
```

5. Find the output files in a newly created `refrag` directory in the same path specified by the `-i` option.

Installation

In order to run ReFrag, Python 3 must be installed in your system. The minimum recommended version is Python 3.11.

The latest release of ReFrag can be downloaded from the CNIC's GitHub page: <https://github.com/CNIC-Proteomics/ReFrag/releases/latest>. The `python_requirements.txt` file lists all the required Python packages, and it can be used to install them directly with `pip`:

```
pip install -r python_requirements.txt
```

Input Files

ReFrag requires several files as input:

1. One or more MSFragger output files. These must be tab-separated text files. Output files from other search engines are not explicitly supported, but they may work if they contain columns with the following information:
 - Scan number (column must be named `scannum`).
 - Precursor charge (column must be named `charge`).
 - Precursor neutral mass (column must be named `precursor_neutral_mass`, ReFrag will add the H^+ mass to this value to calculate MH).
 - Plain peptide sequence (column must be named `peptide`).
 - Δ mass value (column must be named `massdiff`).
2. One MS data file for each MSFragger output file. These must be in MGF or mzML format and must have the same name as the corresponding MSFragger file, excluding the file extension. MSFragger and MS data files will be matched by name. Files that cannot be matched will not be searched.
3. A list of theoretical Δ mass values. A reference list which has been curated from Unimod modifications is provided in `dm_list.txt`. This must be a tab-separated text file with the following columns, in this order (column name does not matter):
 - Full Name: The modification name.
 - `mono_mass`: The monoisotopic mass. This column must be numeric.
 - `site`: List of allowed amino acids for the modification. Values must be single-letter amino-acid codes, 'C-term', 'N-term' or 'Anywhere'. Using 'Anywhere' is recommended. Values must be enclosed in single-quotes and separated by commas. The full list must be enclosed in square brackets. If a site other than 'Anywhere' is specified, this modification will only be rescored when placed at that site.
4. A configuration file. The default configuration is provided in the `ReFrag.ini` file. The contents are described in the Parameters section.

Running ReFrag

ReFrag is designed to be run from the command line. To start it, call python followed by the path to the ReFrag.py file. For example:

```
$ python ReFrag.py
```

There are several command-line options available. Use `-h` or `--help` to display a brief description of each option. A more detailed explanation is provided below:

`-i, --infile: (required)` Specifies the path to the MSFragger results file(s). This can be:

- A single file path.
- A directory path (all `.tsv` files in the directory will be read).
- A path containing a wildcard `'*'` (all `.tsv` files matching the pattern will be read).

`-r, --rawfile: (required)` Specifies the path to the MS data file(s). This can be:

- A single file path.
- A directory path (all `.MGF` or `.mzML` files in the directory whose name matches one of the MSFragger files will be read).

`-d, --dmfile: (required)` Specifies the path to the list of theoretical Δ mass values.

`-c, --config: (optional)` Specifies the path to the configuration file. If not provided, the `ReFrag.ini` file located in the ReFrag folder will be read.

`-w, --n_workers: (optional)` Specifies the number of CPU cores to be used. If not provided, all available cores will be used. To avoid resource contention when running other tasks alongside ReFrag, it is recommended to set a CPU core limit.

`-s, --scanrange: (optional)` Specifies the scan range to search. A range is defined by a starting scan number and an ending scan number, separated by a comma. For example: `-s 1000,2000`.

`-o, --outdir: (optional)` Specifies the path to the output directory. If not provided, output files will be stored in a newly created `refrag` directory in the same path specified by the `--infile` option.

`-a, --dia: (optional)` Specifies a comma-separated list of numbers used to look for MS data files with a `_chN` suffix, where N is each number. The rest of the file name must still match the MSFragger file names. Matching `*_chN` files will be read together. For example: `-s 2, 3, 4`.

`-v: (optional)` Increase output verbosity for debugging purposes.

`-h, --help: (optional)` Shows the help message and exits.

Parameters

The following parameters are available in the configuration file:

Parameter Name	Default Value	Description
Search		
batch_size	1000	Size (number of PSMs) of each task that will be submitted to a CPU core.
f_tol	20	Fragment mass tolerance, in parts-per-million.
dm_tol	3	Tolerance for matching of theoretical and experimental Δ masses, in Dalton. This is an absolute value. For DDA data, a value of at least 3 is recommended to cover the incorporation of up to two ^{13}C . For DIA data, this value should cover the width of the fragmentation window.
score_mode	0	The method for hyperscore calculation. 0 = "MOD-hyperscore". Equivalent to MSFragger hyperscore. 1 = "HYB-hyperscore". Attempts to match all non-modified fragment ions regardless of the position of the modification. Useful for identification of labile modifications.
full_y	1	How to use the y series for fragment matching. 0 = Include the full y series up to y^n . 1 = Exclude the y^1 ion. Equivalent to MSFragger.
preference	0	If both Δ mass candidates have the same score, prefer: 0 = Experimental Δ mass 1 = Theoretical Δ mass
Spectrum Processing		
top_n	150	Maximum number of peaks (sorted by intensity) to keep from each spectrum.
min_ratio	0.01	Remove peaks less intense than this multiple of the base peak intensity.
bin_top_n	0	Spectrum binning method to be used. 0 = Do not bin spectra. 1 = Bin spectra according to the average aminoacid mass (110 Dalton) and keep the <code>top_n</code> peaks in each bin.
min_fragment_mz	0	Remove peaks with m/z lower than or equal to this value.
max_fragment_mz	0	Remove peaks with m/z greater than or equal to this value. A value of 0 ignores this parameter.
deisotope	0	Remove non-monoisotopic peaks up to 3 ^{13}C , with a tolerance of 0.005 Th. This is an experimental parameter. 0 = Do not perform deisotoping. 1 = Perform deisotoping.
FDR		
prot_column	protein	Name of the column containing protein names.
decoy_prefix	DECOY	The prefix that marks decoy protein IDs.
filter_target	0	Remove Decoys from output (0 = No, 1 = Yes).
filter_fdr	0	Remove PSMs above this FDR threshold. A value of 0 ignores this parameter.

The amino acid masses are also specified in the configuration file. Custom amino acids can be added to this list. The identifier must be a unique single-letter code. Selenocysteine (U) and pyrrolysine (O) are included by default, as well as a placeholder (Z) for either glutamine (Q) or glutamic acid (E).

Amino Acids					
A	71.037114	H	137.058912	T	101.047679
R	156.101111	I	113.084064	U	150.953630
N	114.042927	L	113.084064	W	186.079313
D	115.026943	K	128.094963	Y	163.063329
C	103.009185	M	131.040485	V	99.068414
E	129.042593	F	147.068414	O	132.089878
Q	128.058578	P	97.052764	Z	129.042594
G	57.021464	S	87.032028		

Fixed modifications in any amino acid or the C-terminal and N-terminal positions must also be specified in the configuration file. Any custom amino acid included in the previous list must also be included in this one, even if it does not have a fixed modification. By default, carbamidomethylation of cysteine (C) is included.

Fixed Modifications					
A	0	H	0	T	0
R	0	I	0	U	0
N	0	L	0	W	0
D	0	K	0	Y	0
C	57.021464	M	0	V	0
E	0	F	0	O	0
Q	0	P	0	Z	0
G	0	S	0		
N-term	0	C-term	0		

Finally, there are parameters regarding other masses, logging, and debugging. For general ReFrag usage, these parameters do not need to be modified.

Masses					
m_proton	1.007276	m_hydrogen	1.007825	m_oxygen	15.994915

Parameter Name	Default Value	Description
Logging		
create_log	1	Create log file (0 = No, 1 = Yes).
create_ini	1	Create a copy of the configuration file in the input directory (0 = No, 1 = Yes). Specifying custom parameters in the command line will always create a copy of this file.
Debug		
debug_scores	0	Report full score profiles for both MOD and HYB hyperscores (0 = No, 1 = Yes).

Understanding ReFrag Output

ReFrag produces two output files for each MSFragger file that was provided as input: one PSM table and one summary table. The PSM table contains all the columns that were present in the MSFragger file, and some additional ones, which are described here:

Column Name	Description
Information about the peptide with its experimental Δmass	
REFRAG_MH	MH of the peptide, calculated by adding the H^+ mass to the value in the <code>precursor_neutral_mass</code> column.
REFRAG_exp_MZ	m/z of the peptide, calculated from the MH value.
REFRAG_exp_DM	Experimental Δ mass determined by MSFragger.
REFRAG_exp_ions_matched	Number of matched ions for this Δ mass (recalculated by ReFrag).
REFRAG_exp_hyperscore	Hyperscore for this Δ mass (recalculated by ReFrag).
Information about the non-modified (NM) peptide	
REFRAG_nm_ions_matched	Number of matched non-modified ions.
REFRAG_nm_hyperscore	Hyperscore for the non-modified peptide.
Information about the best-scoring Δmass candidate (whether experimental, NM or theoretical)	
REFRAG_hyperscore	Hyperscore for this Δ mass in the best-scoring position.
REFRAG_score_range	List of hyperscores across all possible positions.
REFRAG_site_range	Peptide sequence with the best-scoring site(s) in lowercase.
REFRAG_DM	Best-scoring Δ mass.
REFRAG_site	Best-scoring position, formatted as the one-letter amino acid code followed by the numeric position within the peptide (1-based).
REFRAG_sequence	The peptide sequence including the mass of any fixed modifications between brackets, after the modified amino acid.
REFRAG_ions_matched	Number of matched ions for this Δ mass.
REFRAG_sum_intensity	Total intensity of the matched ions.
REFRAG_name	Name of the best-scoring Δ mass per the user-provided list.
REFRAG_sp_score	Spectral match score, similar in concept to the SEQUEST Sp score.
REFRAG_Label	Target and Decoy labels.
REFRAG_FDR	Global FDR value.

The summary file contains metadata: date and time the search was finished, MSFragger file name, MS data file name, theoretical Δ mass list file name, search time, total number of PSMs, number and percentage of “refragged” PSMs (those for which a theoretical Δ mass was found which scored better than the experimental Δ mass), number and percentage of “refragged” target PSMs.

It also contains a table with all the modifications that have been identified, and the corresponding number of PSMs.