# CS 482/682 Assignment 4

**Due**: Apr 07, 11:59pm

**Task**: Write a program to identify peptides from their tandem mass spectra. We will call your program by the following command line:

python assn4.py --spectra input.mgf --target target.txt –decoy decoy.txt --output output.txt

**Input**:

* input.mgf contains a list of MS/MS spectra in MGF format.
* output.txt is the file where you write the identification results.
* target.txt and decoy.txt should be derived a given **protein** FASTA file, as instructed below.
* file2.fasta is the **protein** database file in FASTA format.
* Note that the file names used in the command line may be different when we test your program.

**Output**:

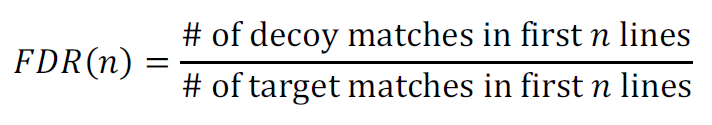
The first line of your output should be a header line as follows:

**Id m/z z score peptide**

* **id** is the index of the spectrum in the MGF file. The first spectrum should have id=0. Then the id number increase by 1 for each additional spectrum.
* **m/z** is the mass-to-charge ratio of the precursor ion of the spectrum, which is read from the mgf file in the line such as “PEPMASS=400.6561”. Despite the tag name “PEPMASS”, the value is actually mass-to-charge ratio.
* **z** is the precursor ion’s charge state, which is read from the data in a line such as “CHARGE=3+”.
* **score** is the score that your program assigns to this peptide spectrum match. A higher score should indicate a more confident match.
* **peptide** is whatever peptide your program identified.

**Evaluation metric**:

We will use the target-decoy method to evaluate the performance of your program. After searching with your program, the FDR of the first 𝑛 lines of your output is calculated as:



Then report the largest 𝑛 such that 𝐹𝐷𝑅(𝑛) ≤ 1%. This value of 𝑛 is the number of identifications you made at 1% FDR; and will be used to grade your program.

We expect your program to finish in reasonable amount of time. For example, using the sample files we provide with the assignment, your program should finish within at most dozens of seconds on a desktop PC. As long as the speed is within this range, your marks will be primarily based on the number of identifications made at 1% FDR.

**Note**:

* Each spectrum should only be matched to the highest-scored peptide. If a spectrum does not find a matching peptide, do not print anything. The spectrum is simply discarded.
* Importantly, you should ensure that the lines are sorted according to the descending order of the scores (highest score in the first line).
* Please be aware that the input FASTA file is a **protein** database, but the spectrum matching is based on the peptides. You don’t need to manually calculate the enzyme-digested peptides from protein on your own. You can simply use the crux toolkit v4.1 (<https://crux.ms/download.html>) and use the crux to generate the target or decoy peptides for you. The command is similar as follows (might slightly differ in different operating systems). Then the target and decoy peptides together with their **mass** (NOT mass-to-charge ratio **m/z**) can be found in “./file2db/tide-index.peptides.target.txt” and “./file2db/tide-index.peptides.decoy.txt”, respectively.

crux-4.1/crux.exe tide-index --peptide-list T --decoy-format peptide-reverse --missed-cleavages 0 --enzyme trypsin --output-dir ./file2db ./file2.fasta ./file2db

* To get the fragmentation for a particular peptide, you can calculate the m/z for b-/y-ions by simply using the pyteomics python package (<https://pyteomics.readthedocs.io/en/latest/examples/example_msms.html>). Or you can also calculate on your own. The amino acid residue’s mass can be found at <http://education.expasy.org/student_projects/isotopident/htdocs/aalist.html> . Use the monoisotopic mass in that table.
* For each spectrum with (**m/z**, **z**), you calculate the **mass** of the peptide by using the equation “**mass** = (**m/z** - 1.0073) \* **z**”. Here -1.0073 in the formula is to subtract the mass of the extra protons due to the charge.
* If a peptide mass matches a spectrum’s peptide mass within the error tolerance, then evaluate the peptide-spectrum match with your scoring function. you can set the precursor mass error tolerance to be **0.1 Da** and the fragment ion mass error tolerance to be **0.5 Da**.

**Submission**:

* The source code.
* A pdf file contains (a) a brief description of your scoring function, (b) the number of identifications at 1% FDR, and (c) the effort to optimize the search (by reducing the unnecessary calculation if two mass are larger than the tolerance, etc.)

**Programming language**: Python

**Sample data file**:

* ups.fasta: A FASTA file containing a list of proteins used to produce the mass spec data.
* test1.mgf: A MGF file containing a list of MS/MS spectra
* test2.mgf: A larger MGF file containing a list of MS/MS spectra

**Hint:**

A white board with writing on it

Description automatically generated