Assignment 3. Due date (March 13, 11:59pm)

Write a program to identify peptides from their tandem mass spectra.

Command line:

The command line used to call your program will be:

```
run.sh file1.mgf file2.fasta file3.out
```

- file1.mgf contains a list of MS/MS spectra in MGF format
- file2.fasta is the protein database file in FASTA format.
- file3.out is the file where you write the identification results.
- **Note that the file names** used in the command line may be different when we test your program.

The protein sequences in the fasta file may contain characters that are not the single letter code of the 20 common amino acids. You should cut the protein sequence at both sides of those uncommon characters, and then discard these uncommon characters.

Output:

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

```
Id m/z z score peptide protein
```

Then for each spectrum in the mgf file that you have a peptide identification result, print information about the identified peptides in a single line in six fields that are separated by whitespaces.

The fields are:

- id is the index of the spectrum in the mgf file. The first spectrum should hav 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.
- protein is the first 10 characters of the header line (including the greater sign) of the protein that contains the identified peptide. If a peptide appears in multiple proteins, you only need to output any one of the containing proteins.

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).

Test:

The data files we use to test your program will be different to the sample files provided with the assignment. However, the formats will be very similar. So, any reasonable efforts in your program to read the sample files should read the test files correctly.

We will use the target-decoy method to evaluate the performance of your program More specifically, the fasta file we test your program will contain both correct (target) and random (decoy) proteins.

After searching with your program, the FDR of the first n lines of your output is calculated as

$$FDR(n) = \frac{\text{\# of decoy matches in first } n \text{ lines}}{\text{\# of target matches in first } n \text{ lines}}$$

Then we will find the largest n such that $FDR(n) \leq 1\%$. 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 seconds (no more than one minute) 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.

More specifications:

Overall procedure:

1. For each spectrum, you calculate the mass of the peptide by mz × z − 1.0073 × z. Here mz is the mass to charge ratio of the precursor and z is the charge state of the precursor. The –1.0073 ×z in the formula is to subtract the mass of the extra protons due to the charge.
2. Each protein can be digested with trypsin rule: after R or K, and not before P. Although most of the peptides are fully tryptic – meaning that their cutting respects the trypsin rule, there are some peptides that may not respect the trypsin rule. You can decide on your own whether you want to include them. Including those peptides will increase running time, but does not automatically increase your FDR performance due to possibly increased false positives if your scoring function is not good enough.
3. For each peptide, the peptide mass is calculated as the total residue mass +18.0105. The +18.0105 is because of the extra water on the peptide.

4. If a peptide mass matches a spectrum's peptide mass within the error tolerance, then evaluate the peptide-spectrum match with your scoring function.
5. After all proteins are evaluated, output the peptide with the highest matching score to the spectrum.

Note that the description of the procedure is for reference only. You may need to adjust it in order to implement it in an efficient program.

Mass Error tolerance

1. The mass error tolerance of the provided sample mgf file will be similar to the ones used in the testing. Thus, error tolerance values optimized using the sample files should also work for the testing files.
2. If you are unsure or do not want to optimize the error tolerance, 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.

Amino acid residue mass:

1. The amino acid residue's mass can be found at the following webpage. Use the monoisotopic mass in that table. http://education.expasy.org/student_projects/isotopident/htdocs/aa-list.html
2. A special case is the residue Cystein (Cys, or C). It is purposefully modified during the sample preparation before mass spec. The mass of the (modified) residue should be 160.03065 instead of the one given in the above table.
3. The following tool can be used as a reference to check if your mass is calculated correctly: https://www.rapidnovor.com/mass-calculator/

Sample data file supplied:

1. ups.fasta: A fasta file containing a list of proteins used to produce the mass spec data.
2. a3-test1.mgf.zip: A zipped mgf file containing a list of MS/MS spectra.
3. a3-test1.peaksdb.csv: An incomplete list of identified peptides from a3-test.mgf. This may help you debug your program.
4. a3-test2.mgf.zip: A zipped larger test file.

These files are for your development purpose only. The actual marking will use a different mgf file and a different fasta file. In particular, do not try to utilize the protein information in your program to assist the scoring.

Third party tools:

- During your development, feel free to search with an existing tool and compare your result with theirs. But your program should be your own.
- Your program should not depend on any other bioinformatics library.