Programming Project 04 - Compute the Number of Peptides of a Given Total Mass

Group Projects

These projects serve as a way for you to learn how to build a programmatic tool in a group setting. Often in this field, large packages and software are built in a collaborative effort, and knowing how to effectively communicate ideas, tasks, and workflow is an essential skill. Here, you will work as part of a group and demonstrate your ability to compose a cohesive (and working) tool comprised of components created by others *and* yourself.

Deadline

February 08, 2021 - 12:00/noon (UTC+1:00)

Submission Guidelines

1. Clone your repository to your local workstation/computer if you have not done so
2. Structure and work on your submission.
3. Commit your work to the git repository.
4. Create a git tag.
   - Tag name must be equivalent to ”GroupProject”.
   - Tag can be made via the command line or using the GitLab GUI
5. Be sure to include a PDF of your presentation in your repository once it is finished

Package Requirements

- All code comprising the backend portion (i.e. the code responsible for downloading, parsing, and formatting the information) must be compiled as an installable Python package
- The package must contain the following:
  - A working CLI (see CLI Requirements below)
  - A clear and descriptive `README.md` that details what the package is, what it does, how it can be used, and examples of how to use the CLI
  - The necessary dependencies so that the package works immediately upon installation in a new virtual environment
  - Working unit tests that test at least 70% of the code in the package

CLI Requirements

- Within the python package described in the package requirements section, there must also be a working command line interface (CLI)
- CLI methods must contain proper documentation that is accessible via the command line
- CLI method documentation should contain:
  - Explanations of the arguments and options involved with the method
  - Brief description of what the method is used for

Use of External Libraries

In general, one can make use of an external library or package that can aid in accomplishing a small subtask, such as a combinatorial problem, interface with an API, etc., but you cannot use a library or package capable of solving **all** of your tasks. You are of course allowed to use modified code from your previous individual assignments (including that of PLAB1) where applicable. If you do choose to use an external resource to perform part of one of your tasks, it must be properly explained in the presentation. If you have any questions or concerns about whether a particular resource is allowed, please feel free to ask via email or issue.

General Remarks

- The tasks are purposely written in such as manner as to require you, as a group, to figure out what tools are needed, what information needs to be gathered, and what resources should be used
- All code-based work is to be done in GitLab
- Use GitLab Issues to track and assign individual tasks and required work
- The software package (backend code) and web application (frontend) can be stored in separate folders in the root directory of your repository as shown here (you can rename these folders as you please):

```
├── frontend
└── project_package
```

Grading (10 pts):

Task	1	2	3	4
Points	3	2	2	3

Compute the Number of Peptides of a Given Total Mass - Introduction

Mass spectrometry (MS) is a widely used scientific technique for determining molecular weights and charges of molecules in a sample, or to a purified sample. It is used extensively in the biochemical field for calculating the molecular weight of isolated proteins as well as identifying possible modifications made to the protein itself (post-translational modifications). MS works by generating a spectrum (known as a mass spectrum) which plots the mass-to-charge ratio (*m/z*) of the ions in the sample. By using the values derived from such a spectrum, one can determine with a high degree of accuracy what the compound (or peptide) is.

Aims

1. **Parse raw MS files** for their *m/z* values
2. **Predict which peptides** could result from the derived *m/z* values
3. **Determine the proteins** that the predicted peptides could be derived from
4. **Create a frontend** which allows one to upload a raw MS output file and obtain a list possible proteins

Tasks

Task 1 - *Parse the Data* (3 pts)

- Given a mzML or mzXML file of a peptide mass spectrum, parse it for its relevant values
  - e.g. intensity, *m/z*, etc

Task 2 - *From Data to Peptides* (2 pts)

- Based on the values extracted in Task 1, compile a list of the most likely peptides that the spectrum may represent

Task 3 - *Protein Prediction* (2 pts)

- Assemble a list of possible proteins that the amino acid sequences (peptides) generated in Task 2 may represent
  - This may require using tools available from UniProt, EBI, NCBI, BLAST, or elsewhere

Task 4 - *GUI* (3 pts)

- Construct a web interface that allows one to upload a mzML or mzXML file and get a list of possible proteins it may be derived from. Your interface should include the following features:
  - An upload button that allows one to upload a mzML or mzXML file
  - A table of relevant values derived from the uploaded file
  - A collapsable table of the possible peptides that the spectrum values may represent (and their amino acid sequences)
  - A collapsable table of the possible proteins that each peptide could be derived from. In the case of multiple proteins, show the most likely based on the similarity search values

Hint

- http://www.peptideatlas.org/ is a good source for peptide MS files