Programming Project 02 - Author Map

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**	**5**
Points	3	1	1	2	3

Author Map - Introduction

Collaboration is an instrumental component of the scientific community; progress and discovery would have been severely hindered if research groups refused to work together. As a result, some groups tend to work with one another quite often and thus one will see the same authors listed on multiple publications despite them belonging to different institutes or even areas of expertise. These types of relationships are of interest to other scientists as it allows them to quickly identify work related to what they are currently investigating.

Aims

The primary goal of this project is to create a pipeline that generates a "co-author" network which details how strongly connected the queried author is to his/her co-authors.

1. **Extract all publications for a given author** using PubMed or PMC and compile a list of co-authors, i.e. other authors listed on their publications
2. **Count cooccurrences** of pairs of authors to quantify the number of publications that any pair of author both appear on
3. **Generate a network** depicting how strongly related every pair of authors are
4. **Create a frontend** which allows one to generate a co-author network for a given author name

Tasks

Task 1 - *Finding the Shared Work for One Author* (3 pts)

- Using tools available from NCBI, write code that downloads a list of publications (and the authors) for a given author's name
- Use this information to compile a list of co-authors i.e. a list of names that were listed as a co-author on at least one publication with the original queried individual

Task 2 - *Finding the Shared Work for All-Pairs of Authors* (1 pt)

- At this point, you only have information on the publications shared between the original queried author in Task 1 and the others. Write new methods to download publications for *every* pair of authors collected in Task 1
  - *Example* - Using the code developed in Task 1, I search for information on author **A**. From this, we compile a list of related authors to **A** and end up with a list such as: [**A**, **B**, **C**] where **A**, **B**, and **C** are all different names. Now, we have information on publications between **A** and **B**, as well as **A** and **C**. In this task you need to find the shared publication information for **B** and **C**, and any other combination of names in the list extracted from Task 1.

Task 3 - *Count Em Up* (1 pt)

- Based on the information collected and parsed in Task 2, count the number of publications shared between every combination of authors

Task 4 - *Build a Network* (2 pts)

- Build an undirected graph in which nodes represent authors (i.e. names) and edges represent at least 1 shared publication between the pair of nodes
- Annotate the edges with the number of shared publications. The line thickness should correspond to the number of shared publications i.e. the more publications shared between the pair of authors, the thicker the edge
- Add additional functionality so that one can visualize the network
- Allow one to be able to export the network in several formats including png, jpg, svg, and pdf

Task 5 - *GUI* (3 pts)

- Construct a web interface that allows one to search for a given name and build a co-author network depicting shared publications. Your interface should include the following features:
    - A text box + submit button for inputting the author's name
    - A table of all found related authors
    - The ability to checkmark two author names and see the number of publications shared between the two
    - An area depicting the resulting co-author network
    - The ability for one to choose which file format to download the network image