Approach to the project

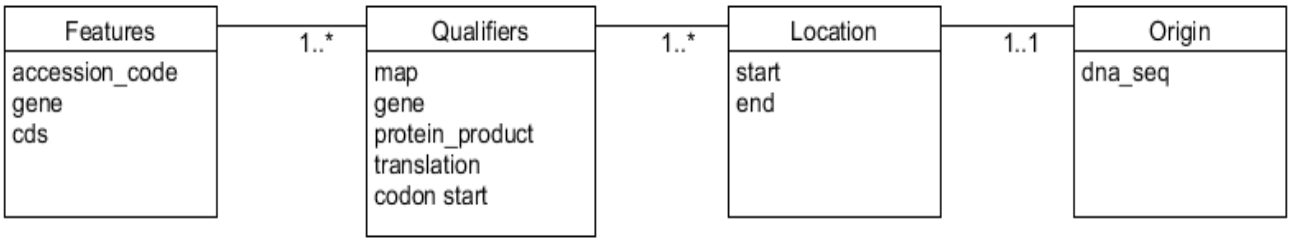
As a group we decided towards a simplistic approach for the project given each of our experiences coding ranged from beginner to intermediate. Based on this we simply suggested which layer we would want to be responsible and decided on this from the onset with no conflicts. As a group we communicated largely though messages and meeting, occasionally in person to discuss the progress of the project. Personally I can say there were no issues with this as we were able to communicate any concerns or development ideas well enough, however more could have been done to integrate the layers of this project also ensuring the APIs were defined properly.

In terms of the group's approach to meeting the project requirements, progress on each layer was discussed occasionally to determine what factors needed to be addressed for the final implementation. Discussing definitions for the APIs within the first weeks took longer than anticipated but once this was established following the presentation sessions, we mostly worked independently. In regards to my own approach for meeting the project requirements, my main focus throughout its duration was on the file parser

Development cycle

From the onset one of the first things that was decided was the information that was to be displayed on the webpage. On the basis of this the design would be simplistic in that the software would retrieve data stored in the database directly with little formatting required. Once the first steps to define the APIs were carried out the development process would run over the remaining weeks until just before submission as there was no final deadline we set for ourselves for everything to be complete. We did however a few deadlines set by each member for any changes that needed to be committed to the code following problems raised through discussion.

I started the process of developing the database layer with constructing the functions which would return information from the database. Given the tables weren't defined at the start these were simply functions which executed SQL code for the retrieval of any data within the database. Inspecting the GenBank file showed a total of 147 entries based on unique accession codes.



The consideration I made for the design of the database relationships weren't comprehensive as they probably should have been. Based on the nature of the project and what data had to be retrieved, I separated the information to be parsed into two entities:

1) Sequence – DNA sequence obtained from the origin
2) Attributes – attributes of that sequence being the features of the coding sequence as well as the accession code

```
MySQL  localhost:3306 ssl  test  SQL > describe seq;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| dna_seq        | longtext    | YES  |     | NULL    |       |
| accession_code | varchar(30) | NO   | PRI | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
2 rows in set (0.0013 sec)

MySQL  localhost:3306 ssl  test  SQL > describe attribute;
+-----------------+-------------+------+-----+---------+-------+
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| accession_code  | varchar(30) | NO   | PRI | NULL    |       |
| gene_id         | varchar(10) | YES  |     | NULL    |       |
| protein_product | varchar(20) | YES  |     | NULL    |       |
| chromosomal_loc | varchar(10) | YES  |     | NULL    |       |
| cds             | varchar(30) | NO   |     | NULL    |       |
| protein_seq     | mediumtext  | NO   |     | NULL    |       |
+-----------------+-------------+------+-----+---------+-------+
6 rows in set (0.0041 sec)
```

The relationships are based on the context of the GenBank file rather that their biological context. This way it would be simpler to write scripts that could select rows matching a given query as all the search terms could be found within one table.

Performance of the development cycle

Much of the structure was decided within two or three weeks of receiving the project as well as the general design of the webpage. The process felt somewhat disjointed and could have included better communication to make development easier. I feel as a result we weren't able to accomplish everything that was set out from the onset. In hindsight code testing on my part should have been implemented better as the methods used were suboptimal in that I only used manual testing for the module and this made constructing the parser very time consuming. The subroutines were called in separate scripts to test if they returned known data values in the correct format.

What worked and what didn't – problems and solutions

Problems which showed up in developing the GenBank parser were mainly due to insufficient testing of the functions before assembly of the script. As a result some of the functions felt overly complicated at the onset as I was trying to compensate for errors. Following examples from GitHub I initially decided to parse the file by placing an iterator of the file in each function and resetting the file pointer. This was essentially reading the file line by line repetitively. As this proved to be highly inefficient and would also not work given the way the class instance was initialised. I instead wrote the function to receive the first entry only based on where the file pointer is set. Parsing was therefore dependent on calling the functions in a set order and a separate script was written to iterate over the file. I also wrote another function which would identify when the file end was reached. For the purpose of the project I inserted a line at the end of the file to be parse to make this easier, rather than leaving it as white space. Two of the considerations I made were to firstly, remove entries that were missing any of the required features and secondly, remove entries with had coding sequences that reference other entries. Considering this removed a lot of the data, this would not be ideal in a normal case scenario.

In terms of what went well, what I can say is that the structure of the GenBank file is consistent in parts that matter making the task not exactly easier but manageable. Documentation for the two type of DB APIs I used, MySQL connector and pymysql cursors, were easily transferable with minimal changes to be made in syntax. The config file also made transitioning between two databases easier.

Overall the project has introduced many new concepts relating to program development that I may not have appreciated before. One of these is how several layers of code can be integrated into one software package. Additionally, how useful GitHub as well as experience with command line can be software development.