Birkbeck University of London
Bioinformatics and Systems Biology
Biocomputing II

Group Project
Reflective Essay

Team: ChromAwesome_22
Kostas Pantelidis

1. Approach to the project

1. Interaction with the team

The interaction with the other members of the group started really well. We had frequent meetings for the first 3 weeks, face to face or group calls. However, because of the current covid-19 situation we could not communicate well after the first 3 weeks. Specifically, after 3 weeks only 3 out of 4 of us could communicate and moreover, there was a big period of 20 days that not even the members left could interact with each other. However, the last 10 days before the deadline we managed to be back on board.

2. Overall project requirements

In our first meetings we set the website requirements and the rest of the team announced me after our group discussion and meetings what I should parse from the GenBank file. I must say that that was clear early so I could focus on my code. Then, after I agreed with the business layer what the api functions should be (week 4), everyone focused on their own code.

3. Requirements for my contribution

It was clear that I had to focus on giving well working api functions to the business layer so that they can check their own work. I created a dummy data file that the api function could use to take data and give the results back to the BL. I also requested specific data from the other layers and in a specific type, for example the return of the function should be a dictionary so that the BL could use easier.

4. Performance of the development cycle

Because of current covid-19 situation, the communication with the other members was not always easy. There were some big periods that we couldn’t communicate and couldn’t test each other code and work. However, at the end we managed to find a way. Unfortunately, we did not have the chance to work the Github idea and git command as we should supposed to do.

5. The development process

Most of the work is Python, using also pymysql and sql queries to create a database.
I created three separate files/scripts.
Two related with my database, one database API script for business layer to use.
Since we set the api function from the start, I decided to focus on parsing the GenBank file first which I knew it would be the difficult part. I had in my mind that I want to create a list of the entries that I can then use it to populate my table. When I already had the first data ready from the genbank file, I created a dummy table in the database, just with 3-4 columns to check if I can populate the database and if the api is functional. Since the were no problems I then continued to parsing my file. The parsing had many stages and many changes as every time I could find something simpler and more functional. I decided to create only one table from start so I didn't have to change much in my database, just to add columns every time I was extracting new data form the GenBank file.

6. Code Testing

As in charge of the database layer, my code wasn't expecting any other code from other layer. We set early the API that the business layer wanted, so I only had to test my code. I created a dummy data file (it's under the name *dummy_data.csv* in dummy folder in GitHub) after we set the API and we had in our mind what the return of the genbank file would be. So, when the business layer had to check the api function they could use this dummy data.
I also created a dummy genbank file with 4-5 entries/genes instead of working on the big original file and tried to include and possible issue I could find in the original genbank file (dummy_GenBank in GitHub), for example missing entries, more than one entries etc.

7. Known issues

As far as I was informed and confirmed by the business layer, the api single function (`dict_entries(a)`) that I was asked from my code to give back to the BL, works fine.
Also the parsing and the table creation and population work also fine and checked in other chromosomes as well.
Unfortunately, I did not have the time to exclude entries that join DNA sequence from another entry as requested. However, the code does take the first splice variant in case of mutations, variations etc as again requested.

8. What worked and what didn't - problems and solutions

As mentioned before, unfortunately we did not work the git commands as we were supposed to. My personal opinion is that lack of knowledge of git from everyone on the group as well as lack of communication, made as focus on our code which we were sending to each other by email. We made some progress in the last few days as far as git and GitHub but that was, I think, too late.
A work that I am proud of is the parsing from the GenBank file using regular expressions.

9. Alternative Strategies

For the API functions:
My first version of the api functions was to return one function for every piece of requested data. Same as far as the parameter with the current version but it would return a single string (or a dictionary) for this specific entry. So, in total 10 functions.
After meetings with the business layer it was decided to give back the current version.

For the database format and table creation:
Another approach I tried was instead of dropping the table every time, in case this already exists, a printed message would show up to the user asking if he wants to drop the table or continue with the existing one and just populate the database. The user could answer with a *YES* or *NO*.
However, although the code worked well, it would not be easy and quick for the user when they try to run the **RUN_database.sh** file to run and populate the database automatically from terminal.

10. Personal insights

The main skill I improved with this project is my coding skills in Python. Specifically the use of regular expressions. I also gained many knowledge on pymysql connections as well as sql scripts. I also understood the importance of git although we did not have the chance to use it a lot. I personally used it and tried to learn more. I also understood the point of API's between layers. Unfortunately, since everyone has their own part, I did not have the chance to deal with HTML code. Overall, it was a good project and although the difficulties, I strongly believe that in a different situation, without corona times, we could have done a lot more.