Biocomputing 2 : Reflective Essay
Msc Bioinformatic with Systems Biology

1. Approach to the project

1.1 Interaction with the team

From the day this project was given, the group was in regular contact. On the first day of the project, the different project parts were allocated to each member; this was mostly done according the level of coding skills we all had. I was given the front-end layer, which I was happy to take. As a group, we nominated Florence as the group leader, which she was happy about. She took on the role of arranging meetings at university as well as via email, phone etc.

We were able to have few face to face meetings at university, where we discussed the API's and what was needed in order for us to make this genome browser, what were the important parts in the GenBank record. We also discussed briefly how the website should be laid out; what the summary table should look like and what should be hyperlink that links to the detail page. Due to the lockdown, our group interactions continued by email, group chats and video calls; this went all smoothly.

1.2 Overall project requirements

The project requirement was to create a genome browser to look at genes in Chromosome 6. GenBank data needed to be loaded into a database to provide a web front end to allows users to view and query the data via the middle layer; where they can search by GenBank Accession, Gene Identifier, Protein Product and Chromosomal Location.

The genome browser was composed of three layers:

- Database layer
- Business layer
- Front end

<u>Database layer:</u>

where the Genbank data was parsed and read and any data any relevant was extracted and converted to SQL that could be loaded into the database. The database layer also provided 'data access tier' - Python wrappers to the SQL that returned the data required by the middle layer.

<u>Business layer:</u>

Where requests were taken from the interface and data was extracted from the data layer by calling the , perform any needed processing of the data, including any calculations that need to be done.

<u>Front end layer:</u>

Where web pages were created that allowed user to query the database - via the middle layer and Python/CGI scripts created that accessed the business layer of code when forms were submitted and generate new pages.

1.1 Requirements for my contribution

My requirement for this project was to produce sets of web pages, perform HTML, CSS formatting and generate tables, by taking data from the middle layer and formatting it for presentation, create python/CGI scripts that allowed users to query the database - via the middle layer; searching specified through forms. Searches carried out by: GenBank Accession, Gene Identifier, Protein Product or Chromosomal Location.

2. Performance of the development cycle

From the first few group meetings we had; we were able to discuss about what we thought was important in the GenBank data in order for us to meet the overall requirement of this project. We also discussed the API's and the browser layout.

After these discussions, the database layer person and business layer person started working on their API's. In the meantime, I started working on a draft HTML code and the CGI script once there was something for me to work on from. The business layer person later provided dummy data that I could use while the database layer person was working on their API; this was tested with my CGI script. We set a deadline for when all the layers could provide their draft code so we could test it. Once all the codes were working smoothly, we continued on polishing our code, while also testing every time a major update was made.

3. The development processes

First of all, we all created a GitHub accounts where we all uploaded our files. I started this process off by making a draft HTML code, making sure I was able to display everything properly on the web browser. Once the BLAPI was at a stage that I could start working on my CGI scripts, I started off by using the skeletons provided on GitHub.

For the listall.py CGI script, I started working on the table layout, so I could display GenBank Accession, Gene Identifier, Protein Product and Chromosomal Location. I then worked on making the GenBank Accession the hyperlink that linked it to its detail page which was the search.py CGI script. I worked on making the search HTML page connect with the listall.py page when something was searched, so multiple entries or single entries were returned.

For search.py CGI script I started working on the layout, so I could display GenBank Accession, Gene Identifier, Protein Product and Chromosomal Location, Amino acid Sequence, DNA sequence, dropdown for restriction enzyme, Codon usage table on the detail page.

Once most, if not all, of the code was working without any major issues, I moved on to fixing my HTML code and adding CSS.

4. Code testing

When the database layer person was still working on their code, the business layer person provided dummy data. I was able to test my CGI code against the BLAPI and the dummy data. Once the database layer person had a working version of DBAPI, I was then able to move on to that from the dummy data. Therefore, I used BLAPI and DBAPI to test my CGI scripts, lisall.py and search.py.

For HTML and CSS testing, I used http://validator.w3.org/#validate_by_upload and http://jigsaw.w3.org/css-validator/ respectively, to validate my HTML code and CSS. Any errors that occurred were resolved.

5. Known issues

There were some issues that occurred, one was the codon frequency table and the searching restriction enzyme by sequence. The codon frequency were two long separate tables. Although I managed to merge the codon usage frequency tables, I was unable to change the layout, so it was just not one long table. I was unable to produce a search bar for the restriction enzyme sequence to be searched on as the BLAPI code needed changing and there was not enough time for this.

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

The developmental process of this genome browser was started early, that in itself was a solution to any problems that occurred. Any minor bugs or issues that occurred in the API's were fixed in time.

One issue that occurred for me, was the merging of the codon usage table. Originally, I had two separate tables, one for codon usage frequencies in the chosen gene and one for the codon usage frequencies in chromosome 6. I wanted them to in one table like the example given by Andrew. On Florence's advice I tried to produce 3 tables with each table having 5-6 rows. However, this was proving to be difficult, tables were formatted differently on different web browsers . To overcome this issue, I used data frames to merge the two frequencies into one table. Although this was not the desired outcome, it was a better outcome then the original one.

7. Alternative strategies

In terms of HTML coding, I could've used libraries jQuery and bootstrap to make the browser more presentable. For example, instead of having a long table in the gene summary page, I could have used table pagination so there could be shorter table with multiple pages. Additionally, I could've created a collapsible section for DNA sequence as some sequences were very large and took a lot of space on the page.

8. Personal insights

Overall, I am quite pleased with the way this project turned out and I think this project was a success. The team members all worked really hard and we all worked together very well. We all gave our all and the efforts paid off.

Although this project seemed like a challenge in the beginning, as time went on it seemed to get easier. Florence did a great job of getting the group together and aiding anyone who was need of it. Oliver did a great job of starting with his database early, giving him time to correct any problems that occurred. This in turn helped me, so everything was not left at the last minute to complete.

Having completed a draft working version of the browser weeks before the deadline was an important part in our project process. This gave the whole team time to work on anything that didn't seem right (any bugs, errors) or improve any code. Furthermore, having a draft version completed meant it gave Nini the time to test the codes.

This project has given me an insight to how web developers work and has made me more confident in coding then I was before. Although I had an inkling of how challenging it is developing websites, I did not expect this much hard work to go into it. This project has given me skills that I can use in a real-life situation if I was to have a career in web developing or something of the sort. Additionally, I have become fond of GitHub. Having never used this before this project, I was a bit confused how GitHub was going help and work. However, I have seen how it aided us, as a team, and individually; being able to easily take code that has been updated, checking what was added or deleted in the file, getting access to old files etc.

Lastly, this project was enjoyable but challenging and it was great working with my team members.