Sarah Griffiths Essay

Link to my github page https://github.com/sazzaras/Chromosome_15

1. Approach to the project

    1. Interaction with the team

Our group had a very friendly attitude with each other during the project and we got on well. Deciding who would work on each section was easy as everyone was flexible. Archana was keen to consolidate her database experience after the previous data management module, I really enjoy working with python and Sergio and Fabio were both happy to have a go at the front end.

We communicated throughout the project although more at the beginning and end - via a slack group. Sergio was particularly helpful thanks to his experience.

Where there was initial disagreement on specifications and requirements we discussed and listened to each other to come to solutions and make decisions. For example, initially some in the group were keen to use JSON mark-up language but others in our group wanted to stick with XML that we had learnt in the lectures. In the end we used neither but we had an interesting debate about the advantages and disadvantages of each mark-up language. It was important that everyone had opportunity to speak and I took on board other peoples ideas.

    2. Overall project requirements

We first discussed what was required on the front end – this helped us identify the important information that needed to be extracted from the data file we had.

We looked at the data file itself to see what data it contained and thought about how we could make the most of the data. There was some minor disagreement about if it was necessary to extract citation data, which I argued it wasn't, a good solution was to agree necessities and nice to have features - using MoSCoW.

In our second session we had a go at modelling the data – Archana finalised the data model independently and sent us a copy of it via slack. I feel it would have been good to keep an up to date record of this in git so that I would have been prepared for the changes that were made after.

As time went on it became apparent that we needed to consider adding additional parts such as a table that contains codon frequency from all coding regions and a table containing restriction enzymes so flexibility is important.

    3. Requirements for my contribution

I worked out the specific requirements for my contribution by working slowly through the specification and making notes. I made a list of functions I thought I needed to create and explained these during the meeting to check others agreed with my interpretation.

Early on I had a go at writing little programmes to test my ideas and better understand what I would require from the database so that I could ensure Archana was aware of my needs whilst she designed the database.

I was also able to articulate to the front end what outputs they could expect and what input would be required from the form.

I should have created a formal 'middle layer specification' document asserting what I required and what my code would output at the start of the project instead of relying just on communication because ambiguity meant I couldn't firmly finish my layer until the data access layer was finalised.

As I worked on the code itself I recognised other bits of data will be needed such as entire lists of genes/accessions/ids etc and what sort of queries the logic layer will need to make of the database. I took a note of these and suggested queries that Archana could use in the Pymysql layer. I offered to work on the pymysql myself but I think Archana was keen to practice this.

2. Performance of the development cycle

Our development cycle was a limiting factor in the progress of our project - in the final week of the project there was more communication and sharing of code but with little time to no time for testing - as the output from back end was different than I expected - I was being expected make major adjustments very late on.

Everyone was working to different timelines due to varying commitments but there was little communication or progress updates. It would have been beneficial to agree milestones we would like to have reached by certain times. I do however appreciate and was mindful of the fact that others in the team had other commitments or modules that required attention.

I uploaded some dummy code and scripts to git hub in March to ensure front end had some data to work with but didn’t get any feedback on these until May when there was little time until the project was due and as I didn’t receive any dummy data the testing of my code was fairly limited until the database was finished on the 7th of May but not on a server so I couldn't test it - which left me with less than a day to calculate total codon frequency, return it to the database and use the data to calculate the ratios. I had prepared functions to deal with the data but wasn’t 100% sure they would work with the data returned. I am conscious that the parsing of the data was no easy task and perhaps I and other members of our group should have reached out to help Archana more.

When we gave our initial presentation, we had a general idea about how we wanted our website to work but didn’t have clear API’s. I did suggest that we agree on a list of standard terms we should stick to such as ‘proteinId’ or ‘geneId’ so that each layer knew what to expect and in future I would be more assertive about this.

3. The development process

We met 3 times at the beginning of the project where we discussed our interpretations of the requirements. On our first meeting I felt we could have made more of the time if everyone had thoroughly read through the requirements and documentation to begin with - because we spent this session reading the document.

We spent most of the 3 sessions discussing the requirements and debated what the correct interpretation of them was. Some members of my group weren’t convinced that we needed full lists of genes/proteins/accessions/ncbis’ so I spent a lot of time persuading them to include these but I fully appreciate that there are different ways to interpret the requirements and different ways to achieve outcomes so I was never overly pushy.

In our final meeting I shared some of my code so far with the group and they offered constructive criticism and ideas which I took on board. Fabio also showed us a basic website format that he had created which was very helpful to understand his expectations from middle layer.

It would have been good if everyone in the group had made use of git hub throughout the project rather than in the last couple of days so we could see how each other’s code was developing and offer ideas and support.

4. Code testing

See the test.py file for tests written. One of my tests compared the translated sequence to that of the GenBank file. This helped me realise that the code I had written was not translating my sequence correctly. This was because I am foolish and thought that you had to splice the sequence at stop/start codons and concatenate those strings. I quickly realised the problem was you simply have to translate the codon sequence.

I also incorporated a made-up restriction enzyme in to my code and pasted it in and outside of the coding region to check this function.

When coming back to code I had written with in descript variables I had to use the (print) statements or 'devil debugging' to make sense of what I had done - which helped me understand the value of writing understandable variables in the first place.

I tried to include a few tests to check my output for example a test to count how many codons were in my frequency dictionary which should be 64 unless there are unusual codons or outliers and a check to see how many amino acids were expected and how many were in the final translation.

I created tests after I had written my code but I can see there is a benefit in designing tests before writing the code so that you understand what the outcome of your code should be.

We didn't have much time to test our entire project because we didn't attempt to integrate all the layers until the 7th May. A good test of our overall project would be to upload different chromosomes data but we didn't have time to do this.

5. Known issues

Our project didn't work as expect and we ran out of time to get it working. Archana didn't manage to upload her code until 1am on the 7th May and I struggled to get it working on my PC. Sergio had planned to put all the pieces on a server he had access to. Unfortunately, he could not get this to work - so I tried to put it on the hope server. I thought I had managed to get the data access layer to work after a little debugging but when testing I realised it wasn't returning the gene data. Archana's database was being created but for some reason without the gene table. I communicated this via slack and Archana tried to help me getting it working but we ran out of time.

Furthermore, even if I had managed to get the data access layer working my code doesn't make use of all database data. Archana changed a few things in the database layer such as returning CDs join instead of start and end sites. There was a mis communication between Archana and I so when I finally saw the data I thought of how I could parse the CDS start and end myself but ran out of time to implement this - given a little more time I could have worked with the CDS join.

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

I was proud that when Sergio looked at my user guide he said he could understand how my code worked. I had to re-write my user guide about 3 times because I really struggle explaining things. I was glad that I managed to make my code usable with a very minimal number of functions required to be called from the front end.

Something I know I need to work on is over all readability of my code - I wanted to create variables in such a way that comments were not required for understanding how the code works but this can be challenging as you can't repeat variable names and sometimes an explanation of a variable can be 4 to 5 words long. Eg. testForExpectedNumberOfCodons.

One thing I found challenging was trying to find ways to work on my code before having data from the DB to work with. Using dummy code removes a layer of complexity from my code but in some ways, this forced me to write the code in a way that could work with any data access layer and require minimal editing once the data access layer was provided the day before It was due in.

Initially I asked if the data access layer return variables as opposed to a dictionary. I googled it and there was a very simple solution on stack overflow *(locals().update(d])* which made it effortless to convert the dictionary in to variables anyway and from this I learnt that you need to find ways to work with the format of data you are given.

At the start of the project I used variables such as x, y and made use of random words. However, after editing my code multiple times I quickly realised how important it is to write it in a way that is read-able by others and to my future self. I often left days or weeks in between working on my code and making good use of variables really helped me understand what I had done.

At the end of each session of working on my code I tried to make a list of a few ideas of improvement or things to work on next time. This made it easier for me when I came back to work on it and I would continue this practice.

I did notify my group early on that I would be busy over the bank holiday so would have limited time to work on it and encouraged everyone to finish a week before so that we would have time to write our essays etc. but this never happened. I still did my best to facilitate the last-minute putting together of the project and stayed up until midnight working on it - which I imagine is a common state of affairs.

When Archana uploaded her code to slack I did feel a little frustrated that the style of both her user guide and code very closely resembled mine, but I take this as a compliment that my style was clear and it was good to use a consistent style for the entire project. (This may have changed since uploading this essay to git)

7. Alternative strategies

At the beginning I suggested we make some very basic dummy code – a dummy db, dummy pymysql, dummy middle layer and dummy front end just to make sure we understood how the layers communicated and I think this would have been a valuable exercise and given us a basic structure to work from.

I tried to split my code up in to several classes as there was pressure from my group to achieve this but I found this threw a lot of errors. I decided that as there was only a small number of functions - the code was easy enough to use and understand without classes. I know using OOP has benefits for large amounts of code but in this case, It seemed to be a hinderance. It is something I will use in future.

I did have a go at using pyCharm and sublime as suggested by Sergio although I have decided I prefer the more basic layout of IDLE.

I also suggested that in data access layer we could use 'or' statements to the AccessGeneData function so that the database could be searched on any of the terms – Gene/protein name/accession number/ncbi id. And I offered to write this myself but I think Archana didn't want to make any alterations at this late stage – which was fair enough.

8. Personal insights

Whilst waiting for the back end – I decided to create my own dummy data base and dummy pymysql code to help me better understand the communication between layers. This helped me understand that I needed to write python functions that took variables returned from the data access layer as arguments. This also put me in a better position to discuss requirements with back end and gave me an opportunity to practice DB and pymysql.

It’s important to agree API’s with more clarification – we failed to do this and it made it hard for each layer to work independently and made the end of the project stressful - making sure all layers communicated and had the data they required.

One of the most important things I learnt is the value of carefully designing and thinking about your code before diving in and writing it. I really enjoy problem solving so am always keen to jump straight in to problems but taking a step back would save me time and effort in the long run and perhaps allow me to come up with more efficient solutions to problems.

I really enjoy programming and something that draws me to the subject is working constructively with others so this group project has been valuable. It is important to always be open minded to new ideas, new languages and listen to others whilst being prepared to help others. Even though we didn’t finished the project it was still very valuable and I know what I would do differently next time.

I am a passive person and I think assertiveness would help with future team projects but I think confidence will grow with experience. I feel a little bit defeated that we/i didn’t manage to get it to work in time for the hand in but I will keep having a go with it because it’s a good challenge.