# Biocomputing 2 reflective essay

# Approach to the project.

#### Interaction with the team

The group from my perspective was made up of one person who I knew very well from the full-time course and one who I hadn't been introduced to previously. From the first meeting everyone was very keen to stay in close contact with multiple methods of contact being shared. In the first meeting the layers we were keen on were discussed I felt I wanted to do the middle layer as I was interested in the handling and converting of the data. Diego said he wanted to do the middle as well and Georgina was keen not to do the middle and instead either end. It was decided I would have the middle and Diego would take the front as he was interested in furthering his knowledge in html. Whereas I wanted to improve my regex as I had trouble working with it in biocomp1 as well as my general python as it is what I will be working with after this course. So, we both ended up reasonably happy with Georgina taking the base layer. We all met in person after lectures to discuss how to move forward and once after the full-time exams to begin integration. In the meantime, with any questions or discovery's/problems we had easy contact and quick responses from other members of the group. The initial development of the general API was slow however after the lecture where we presented our progress it quickly formed. This was mostly due to more understanding of how to produce this as in the lead up the group was confused how to approach this. At the start of the project we did not use GitHub to its full potential as closer to the deadline especially while integrating mine and Diego's layers with Georgina's draft function's GitHub proved very useful. Small edits to the outputs or changes to fix problems with the handling of data could be done very simply and quickly. This along with the provided file structure streamlined the integration of the layers due to the agreement to not edit the structure too heavily. Towards the deadline most of the integration relied heavily on messages followed by small edits being updated onto GitHub.

### Overall project requirements

From the information provided we first focused on what we thought our individual layers should do. Me and Diego due to being full time students were able to discuss our layers interactions easily. Whereas Georgina due to being part-time was not able to bounce ideas as easily. However, meetings after lectures kept us up to date on what we thought was needed. As a group we discussed and noted down exactly how we intended for our programs to interact and if there were and requests, we all did our best to accommodate. During early stages it was decided that the search term would come from the website as a radio describing the information used to search and a typed query string as well as the display all feature. We also discussed the removal of entry's that contained a join or had data missing from certain fields however at the time of writing this (23:00 07/05) the

database is not available and so my functions do not include a feature to handle these if they appear.

#### Requirements for my contribution

As I had ended up with the middle layer, I first tried to understand how the data would come in from the base layer. As well as this I explained to the front end the rough ideas, I had for how the data would come out of my functions. At this stage the website was still being designed and so it gave me time to finalise this. I first decided how many functions I would require in order to fulfil the development and decided on one function for each main action. This was to ensure I was clear in my goal for each one rather than requiring more information from the front layer in order to inform my functions on what was needed. As well as this as the user given information was sorted early it allowed me to have a sold start on what inputs I would have and how to handle them. At this stage I took time to talk to base layer on how they intended to implement the search wrappers and their inputs. From this I was able to start all functions in the same way packaging the information from the user to be sent to the base layer. As well as this using the dummy data provided in the base layer draft, I was able to unpackage the data received in in a uniform manner.

Soon after this I finalised the outputs of the functions to make it as easy as possible for the front end. As well as this the basic functions that simply recovered relevant information from the base layer were written. The retrieve all using the draft browse from the base layer and the others using the search from the request function.

These were what I decided my layer needed to handle as this ensured all information passed both down through my layer as well as back through being edited or annotated as used. It also prevented both layers from having to interact with each other. Which would have added another layer of complexity to the project as well as being the incorrect way to handle it in my opinion.

Finally, I also wrote a function that highlighted chosen restriction enzymes using regex and symbols to allow for display on the front end. However, during research done by the front end discovered a way to highlight regions using jQuery and Javascript. While my function is working and identifies the binding regions clearly it was decided between us to use the jQuery method as it looks very attractive to the user. As well as this it accepts user inputs to highlight even unsupported restriction enzymes. My function can still be found in blapi.py on GitHub.

# Performance of the development cycle.

The overall pace of development was fast out of the gate and lots happening in the final weeks however it slowed in between these periods. However, this was in part due to the looming exams during the first week of May and the preparations needed for them. The personal meetings slowed during this revision period and for the week of the exams very little was said, and personal progress was instead focused on between revision sessions. Between me and front end the development was simple as we saw each other very often

and so updated each other very frequently on issues and approaches. This was also in part due to the heavy interactions between our two layers. On the other side lots was said during the development of the draft function for searching and once this was developed the focus turned onto the database design and the parsing of data. As well as this there was an ongoing discussion on what data should be made available with each additional variable in the database api. A middle ground was reached by including options supplying both individual data as well as grouped data. This allowed for less calling between the layers compared to individual function calls for each type of data. E.g. DNA code and splice information as one nested list compared to multiple calls of the function. After the exam period the final versions of the front and middle layers were worked on and interact properly utilising the draft data and functions provided by dbapi.py. This ensured the three layers worked properly with each other according to api's. This iteration of the webpage did not however search instead the base layer returned dummy data based on the radio. At 23:00 07/05 the full database is not available and so the front and business layers still run on dummy data retrieved from the base layer through the middle layer.

## The development process.

For my functions I spent about the same amount of time designing them as writing them. I begun with the protein converting function. This was chosen due to an idea on how to approach it based on previous personal coding however it made following functions much simpler as elements could be reutilised for other functions. The first part developed was the handling of the data. I used my own dummy data and produced a fake search function to be used until the base layers was produced. I first finished the for I in range of a given string and [i:i+3] aspects of the function first. This allowed me to easily move through a given string and extract the codons to be converted. This functionality proved very useful across most of my functions produced. After this I moved onto the coding sequence function which utilised this feature in a different way instead of moving through codons it moved through regexed digits in pairs. These pairs outline splice sites on the DNA and so by using these the coding sequence is removed. This functionality was again used in both the coding regions and any coding re functions. When given access to the draft search function of the base layer I received the data in the same structure as my dummy data a nested list. I however had a different arrangement of data to the draft and so had to change my selections. As well as this I hadn't considered the alternative splicing and so had to include a feature to remove DNA to align the start codon and in scenarios re attach it. A final problem that I found through testing was that the protein sequence was incorrect. It was as if there was a frame shift occurring. I realised that the splice was occurring 1 to far forward due to python's splice feature treating the first digit as 0. I fixed this by lowering the number by 1 every time a splice occurred using data from the gene bank file.

# Code testing.

After the discussion lecture with the professionals I started using PyCharm to error test. This highlighted regions that had issues as well as allowing me to improve these areas easily. At the start of the process I was using print to show me where I thought errors were occurring but after the error testing lecture I moved away from this practice. However, at that point

lots of the code I had issues with initially had been solved. One issue that I encountered later was that the protein sequence was incorrect when I was using the gene bank entries. to test my functions. I realised why it was occurring however when I was focusing elsewhere and so was able to rectify the situation. However, I would not have noticed without using varying dummy data as I initially thought it was error caused by my poor initial dummy data.

#### Known issues.

The main known issue is that the database is not functional at the current time (23:00 07/05). Due to this the website only considers the dummy data stored in the dbapi.py file. Before the deadline in an hour we will attempt to integrate the base layer if it becomes available but if there are any issues, we may not have time.

## What worked and what didn't – problems and solutions.

The first two layers work well on the dummy data however there isn't the full picture as there is no way to access the database at this time. A solution would be to have used another groups database as a last second replacement for the wrong chromosome however we were not able to reach other groups to borrow their chromosomes datasets.

## Alternative strategies.

One alternative would be to use my function for the main method to identify restriction enzymes. However, we believed that the JavaScript was a more integrative and better-looking approach, so it was never utilised.

# Personal insights.

I think as a group we should have had more in person meetings to drive forward the work like weekly lab meetings. This would have forced us to have had something to show each time and would have made us progress more evenly throughout. I learned a lot in how to integrate my functions into both a database and to be utilised by another user which I haven't been exposed to previously. I also learned a lot about working with others and how useful GitHub is to pass information between group members easily.