**410.712.82 - Advanced Practical Computer Concepts for Bioinformatics**

**Johns Hopkins University, Krieger School of Arts and Sciences**

**Advanced Academic Programs, Biotechnology Studies**

Name: Brian Poon

UNIX Username: bpoon1

**Final Project – Proposal**

# Background and Purpose

In trying to determine what ideas are suitable to take on as a final project, my primary challenge was determining the boundaries of the project relative to the goal of the project. This was a particular challenge because each project idea was scrutinized according to that boundary line and the allotted timeline. On one hand, the idea may not be as exciting. Perhaps the idea has been done before and provides a lack of challenge. On the other hand, the idea may be absolutely unique and revolutionary. However, the time necessary to achieve a functional project outcome would and the difficulty in overcoming challenges such as resources makes it clear that the thought is too complex to accomplish in the given time frame. During the brainstorming for this project, a thought had come to mind after viewing the variety of bioinformatics tools available online. Many of these tools are specific in nature or stem from large, established host with enormous database resources. However, if one was to use these tools and host-databases, it would require inputting the same sequences or starting information in each of them. Hence, the thought came to mind for this idea: a Swiss army knife for bioinformatics. A tool where it brings together the tools from multiple resources for a high-level overview of a particular DNA, RNA, or protein sequence. These high-level overviews can helpful during initial introduction of a sequence, such as that of a student learning about an assigned sequence, or for someone engaged in research to broadly look for leads on multiple sources. The input to the tool can either an identifier or a sequence. The output display will depend on what the input was. If it was an identifier, then the displayed results will be from databases, such as GenBank or Ensembl. If it was a sequence, then results from other bioinformatics tools would be shown in a way akin to if the sequence was searched individually in those tools such as BLAST. Additionally, if a sequence was the input or an identifier led to a sequence, then a basic analysis summary of the sequence can be displayed to contain information such as G/C content. Lastly, the results will be recorded in the backend for retrieving the search information.

# Technical Details

Starting Page: The starting page will consist of HTML, CSS, and JavaScript. The starting page will come with both a large and small input text field. The large input text field will be for inputting sequences. The small input text field will be for inputting identifier information. The user must select which input type, either sequence or information, their input is. Changing selections will cause the text field not selected to be greyed out. The most useful aspect of any tool is the options that it comes with. An option to hopefully incorporate is the option to upload sequences as FASTA files. Additionally, some options to consider for the results page would be using checkboxes to identify what specific resources wants to be used. Once the submission button is clicked, it will lead to the results page.

Results Page: The results page will be comprised of HTML and CSS. However, it will use Computer Gateway Interface (CGI) written using Python to perform its tasks, then publish the results on the HTML results page. An additional option to incorporate would be the ability to download the result page in a printable format. The results page will consist of the following information:

General Summary: The *General Summary* portion will include basic analysis of the sequence. It can include the *type* of sequence (DNA, RNA, or Protein), the breakdown of residues or nucleotides, and the length of the sequence. There can be special outputs shown based on the type of the sequence. Sequences containing DNA can have G/C content ratios.

API Outputs: These will be outputs based on what kind of information was submitted initially. If a sequence was initially submitted, then the CGI will perform an API call to sequence-related tools, such as BLAST, and display the result. If an identifier information was submitted, then the CGI will perform an API call to databases, such as Ensembl or GenBank, and display the results in a segregated manner as to let the user know the source of the information.

IVG Output: This particular output is more ambitious and will depend on the project timeline. However, it is uniquely different in that it is a genome browser that is embedded into the HTML through JavaScript and has its own section here. The IGV option will begin on the starting page as a selection option for identifier inputs. The results page will only display the genome browser if the identifier leads to a sequence in a database with the genome and position referenced. If that is available, the data will be pulled and used for the IGV output to display the alignment visually.

Database Storage: As mentioned earlier, the searches performed will be retained on a database for retrieval of information. The information will be stored within its own table in a MySQL database. Each record will be stored by a unique identifier (UID) that is based on the date and time, with the time down to the seconds. The challenge in determining how to save information from this project in a MySQL database is the type of information that is saved. Without vetting through the data types returned from all API calls, it is not possible to determine early on if post-call information can be saved without data loss from incompatible data types. However, it is possible to save the pre-call information, which is the API string or variables, used to perform the API call. In this case, it’ll save the step of inputting the initial information on the starting page and making the selection options.

# Design and Development

The timeline of the project is a critical factor that determines the amount of tools and databases that can be added into this project. The project goal is to create a single nexus for other bioinformatics tools, such as a *Travelocity* for bioinformatics searches if one will find a real-world analogy online. In that sense, the primary process flow is non-specific and generalized in order to integrate tools/databases into the project. The challenge here would be to fine-tune the incorporations in such a way that the data is seamlessly transferred between the source and the project for the user. A visualization of the process in a high-level overview can be seen in the flowchart on *Figure 1* below. The figure also identifies the primary method for carrying out the portions of the process. The portions for HTML on the starting page will incorporate JavaScript (JS) primarily for client-side validations and controls. The JS used on the result page will mostly be for displaying of information. Each source will have its own sectional area and visibility of relevant sections will be controlled using JS. The option for downloading, if given the time and resources to do so, will most likely use JS to flatten the HTML into a PDF format for download.

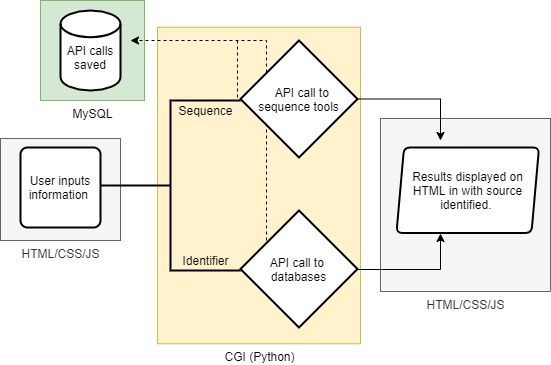


Figure . Overall process flow

It was mentioned earlier about certain options or integrations that can be added given the timeline of the project. At the minimum, the project will incorporate one database source and one sequence tool source. This is to demonstrate proof of concept prior to scaling up with additional tools or external databases. The database currently chosen is the *GenBank* database, which has API available for programmatic searches. The sequence tool currently chosen is *BLAST* because it is a predominant tool in bioinformatics and has an API available. Additional databases and tools currently thought of can be seen in *Figure 2*, along with the sequence-related general summary results. Bolded names are methods that are going to be incorporated at a minimum in the project. Ideally, an identifier search would incorporate the sequence from the identifier found in the database to be used, but the incorporation of that option may exceed the timeline allotted. Future thoughts is to provide a direct linkage for identifier input to pull sequence information after selection by the user for use in the process for sequence inputs.

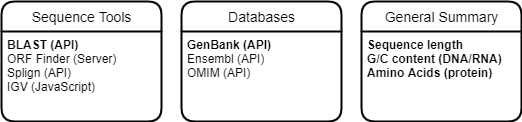


Figure . List of tools, databases, and summary information

Now that we have an idea of what the project is to incorporate and some direction as to how that would work, the next step is to visualize what the user would expect to see. The general layout for the starting page and result pages are shown in *Figure 3-5*. Currently, the result pages have two options because of the two types of input information that can be provided from the user. In *Figure 4* and *Figure 5*, the dotted lines indicate additional sections added, with visibility controlled by JavaScript, for additional tools selected for in the starting page. A possible future workflow is to use the top portion of the results page for identifier inputs to contain an option for selecting the referenced sequence from the database as the sequence to use for input into the sequence results page.

Retrieving historical searches

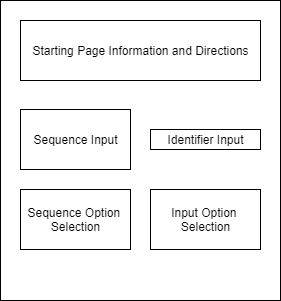


Figure a. Starting Page Layout

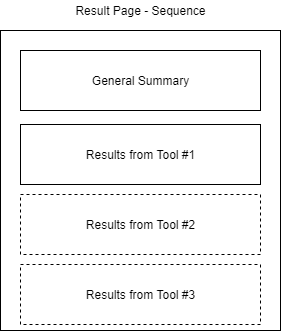


Figure . Result page for sequence inputs

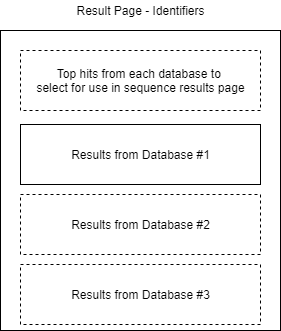


Figure . Result page for identifier inputs

# Reference Links

1. https://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE\_TYPE=BlastDocs&DOC\_TYPE=DeveloperInfo
2. https://rest.ensembl.org/
3. https://www.ncbi.nlm.nih.gov/home/develop/api/
4. https://github.com/igvteam/igv.js#igvjs
5. https://www.ncbi.nlm.nih.gov/sutils/splign/splign.cgi
6. https://www.ncbi.nlm.nih.gov/orffinder/
7. https://omim.org/api