Ramapo College

Virus Protein Interactions in Plants Database

Final Report

Ritwik Katiyar

Advanced Bioinformatics

Dr. Stuart

10th May 2019

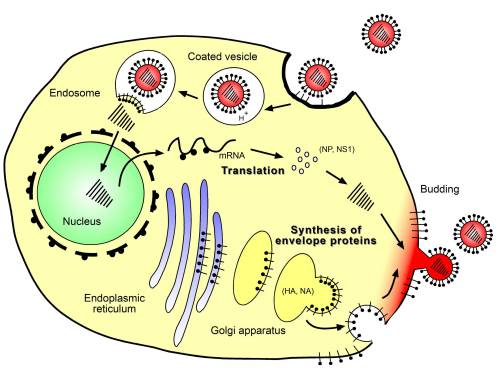
Abstract

Viral proteins and viral Si-RNA’s (Small interfering RNA’s) are important to understand in order to come up with ways to counteract viruses. Hence, a database which could record viral proteins along with the type of host the virus infects could prove beneficial in understanding what the viruses are capable of and thereby giving a step ahead against counteracting viruses and protecting agricultural crops for viral diseases.

Introduction & Background

In 1892 a Russian botanist, Dimitrii Ivanovsky conducts and experiment showing that a pathogen other than bacteria was infecting the tobacco plants and that the organism was able to pass through porcelain filters that would normally retain bacteria. Then in 1898 Marcus Beijerinck made similar observations and concludes that the pathogen must be a distinct organism and officially coins the term “Virus” (Lecoq). Since then, the study of viruses has extended into its own subfield of virology. Many different viruses have been indentified, named and sequenced. Various viral databases have also been created that record the potential viral hosts, record viral sequence, or simply record the existence of a specific virus. Since the discovery of viruses we have been able to create numerous vaccinations and nearly eliminate major viral diseases in humans. We have also created numerous vaccinations for domestic animals as well as for wild animals and even for plants; although, plant vaccinations are simply referred to as pesticides. With the use of pesticides we have insured the safety of our crops. However, there is still a lot that we do not understand about viruses. (Smith)

Viruses are simple parasitic orgasms that require a host in order to survive. Since viruses do not possess a proper cellular structure, they require a host to replicate. Viruses come in various shapes and sizes and comprise mostly of either DNA or RNA. The nucleic acid sequence can also be single or double stranded DNA or RNA. Viruses also possess the capability to encode various types of proteins ranging from as little as 3-4 proteins to 100-200 proteins (Lodish). Viruses depend on a relatively simple mechanism to infect their hosts. First a virus breaks and enters through the membrane (or membranes) of the host cell. Since, viruses are usually extremely small and have a simple structure, a single virus only contains limited amount of RNA or DNA, in other words a very limited number of genes. Viruses use these limited genes to not only code for proteins that can be used to simply allow the virus to replicate within a host cell, but also influence the proteins produced by the host cell (Smith) (Lodish). This allows the virus to hijack the host cell and replicate. Eventually replicating enough to burst out of the cell and infect other neighboring cells.



**Figure 1: Replication cycle of influenza-A virus.** Binding and entry of the virus, fusion with endosomal membrane and release of viral RNA, replication within the nucleus, synthesis of structural and envelope proteins, budding and release of virions capable of infecting neighboring epithelial cells. (Kamps)

Obviously many plants, animals and even some single celled organisms possess a natural anti viral defense systems. However, many viruses also come equipped with a counter host anti viral defense mechanisms.

One of the most important strategies of plants against viral infections is known as si-RNA- mediated (small interfering RNA mediated) gene silencing. Si-RNA-mediated gene silencing allows for defensive signal to spread to other neighboring cells otherwise the mechanism itself is similar to the mi-RNA (micro-RNA) mediated gene silencing. Plant mi-RNAs play two main functions in the antiviral defense; mi-RNAs target the viral RNA/DNA and prevent the virus from reproducing, and mi-RNAs trigger the synthesis/biogenesis of si-RNA (which is responsible for the plant cells primary antiviral response). However, viruses have a counter defense system against the plant cells known as viral suppressors of RNA silencing (VSRs), which interfere with host RNA silencing in multiple ways. (However, VSRs additional functionality includes assistance in viral replication, encapsidation and movement.) VSRs primarily use two methods in dealing with plants defense system; first being the suppression and the assembly of AGOs (Argonautes; protein family that plays vital role in RNA silencing processes) into RISCs (RNA-induced silencing complex; meant to silence the viral DNA/RNA) and the second being interacting with AGOs (Argonautes) to degrade the protein. The figure below shows the two viral counters to the plant anti viral mechanisms in action. (Rui) (Kamthan)

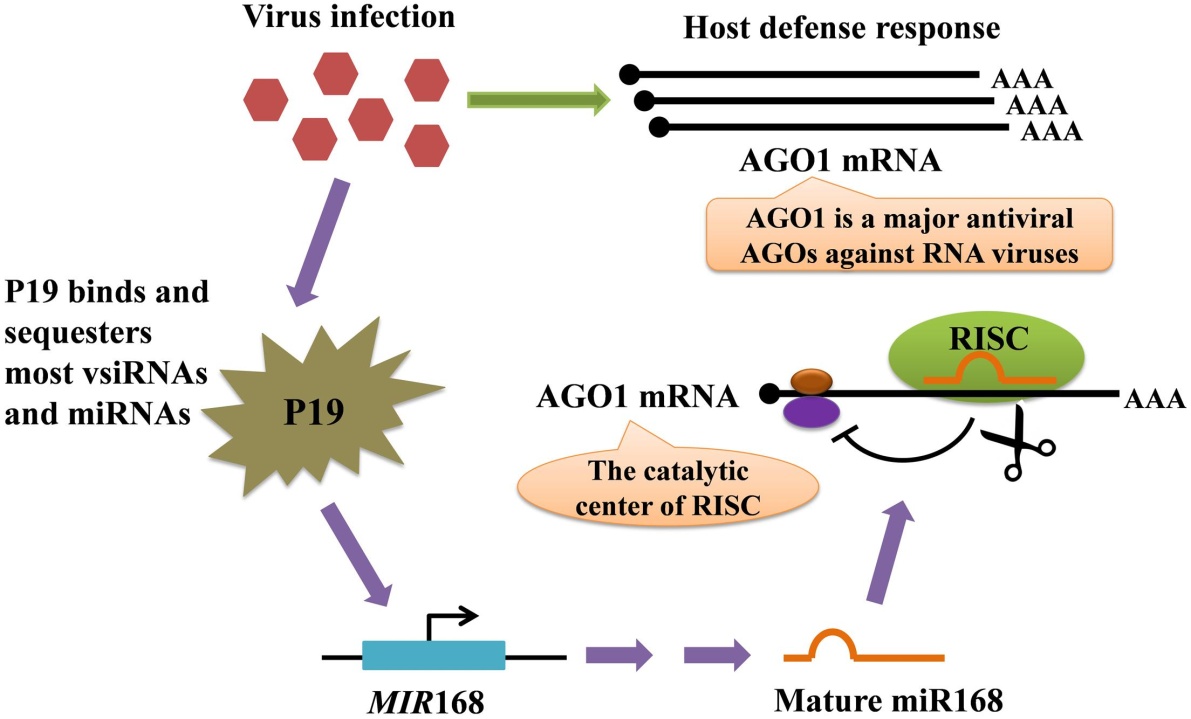


Figure 2: represents a model for the regulation of AGO mRNA level mediated by protein P19-induced miR168. The AGO mRNAs are made when a virus infection is detected within the cell. While the plant cell is creating mRNAs for AGO the virus produces P19 VSR which binds to the virus- encoded siRNAs and with host cell miRNAs which prevents the miRNA loading into AGO. The P19 protein however, does not bind to MIR168 micro RNA (MIR168, represses the AGO1 mRNA); resulting in low production of AGO1 mRNA. (Rui) (Pratt)

Based on the information, it is clear that viral proteins and viral si-RNA’s are important to understand in order to come up with ways to counteract viruses, and there by protect agricultural crops form viral diseases. Hence, a database which could record viral proteins along with the type of host the virus infects could prove beneficial in understanding what the viruses are capable of and thereby giving a step ahead against countering viruses and protecting agricultural crops; and thereby preventing massive crop loss due to viral outbreaks.

Requirements & Materials Used

|  |  |
| --- | --- |
| Requirements & Materials | Summary |
| Visual Studio Code | Used to write all the code |
| Python and Bio-python | Python was used as a base to bio-python and postgresql. Bio-python was used to grab viral sequence information and viral IDs |
| Flask, HTML and CSS | To visualize all the data within the data and create an app to showcase all the data |
| Postgresql (Elephant SQL) | Database where the data was stored |
| R, R-studio and Taxize | Used to grab scientific names and common names of plants |
| Beautiful Soup | Used to grab information from Plant viruses online website |
| NCBI and Conserved Domains Database | NCBI was used from bio-python and CDD was used to search for various motifs and domains within a sequence of translated CDS’s |
| Microsoft Excel | To better organize the raw data gathered from various sources |
| Matplotlib and Pandas | To create the graph on the home page of the website. |

Table 1: Shows a comprehensive list of requirements for the following methods of setting up the database

Method

1. The first step to creating any such database is to develop a schema for the database.

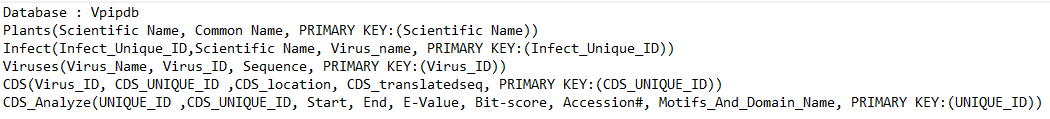


Figure 3: displays the schema of the proposed database.

1. Next to select the plants for the database. Since there are numerous agricultural crops grown around the world and a lack of data linking a specific virus to a particular crop; it is important to focus on the top grown crops around the world. To do that, the data about the amount of agricultural crops grown around the world was taken from Food and agricultural organization (FAO). This data was then sorted and only the top 10 plants were kept.
2. FAO presents the data with the use of common names hence to get the scientific names of the data a simple R script was used that would grab the genus of the plants based on the common name.

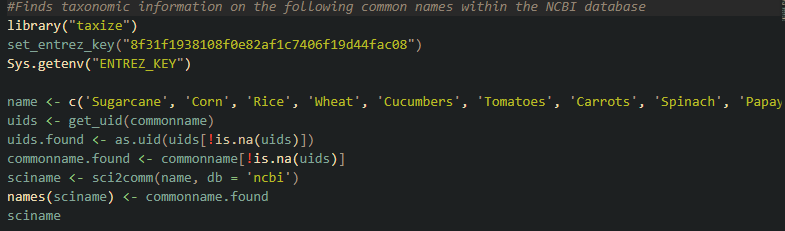


Figure 4: Shows the R-script that finds taxonomic information from common names within NCBI

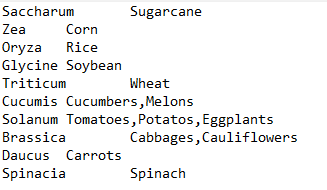


Figure 5: The data was then organized using excel

1. With this information in hand a script was written using beautiful soup to gather information from the Plant and viruses online website. The data that was extracted from Plant and viruses online website was the data on plant susceptibility to a specific virus , all the plant species that the data was available for and all the viruses that were recorded by the website.



Figure 6: The first script used to get the list of hosts used by the website

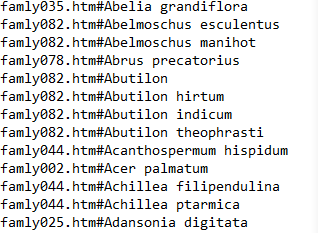


Figure 7: Shows the output from the script

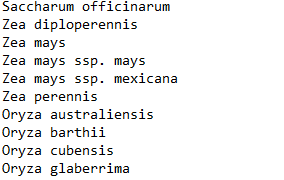


Figure 8: Using Excel the output data was organized

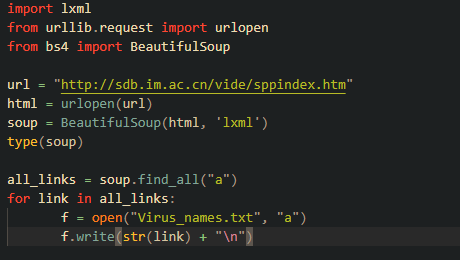


Figure 9: the second script used to extract all viruses that infect the said host from the website

Once the list of all plant and the viruses used by the Plant Viruses Online website was gathered, the list was then organized and arranged using excel to contain appropriate data. The viruses were also then linked with their host plants; figure 10 shows the final version of what the data looked like.

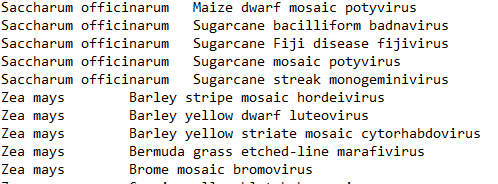


Figure 10: shows how the data looks like once it’s been organized in the tab delimited format. The first column shows the species of plant. The second column shows the virus that infects the plant.

1. With the use of a similar script to that shown in figure 4, the common names were determined for each and every plant.



Figure 11: The data for the plants table mentioned in the schema in figure 3. Scientific Names on the left and the common names on the right

1. Once the viruses were connected to their respective host plants the NCBI viruses ID’s was determined using a script. Then another script was used to grab the sequences for the viruses based on the determined ID.



Figure 12: Shows the script that extracts a single NCBI ID from the search of the virus

.



Figure 13: shows the script used to extract sequence information based on the ID obtained from previous script

1. The resulting data was then arranged using excel



Figure 14: shows how the data looked in excel. The first column shows the name of the virus. The second column shows the ID found. The third column shows the complete sequence of the virus.

1. After the sequences were obtained the next step was to obtain information on potential CDS regions for each virus. Not all viruses contain a CDS region and for viruses without a CDS region no further data was necessary to obtain.

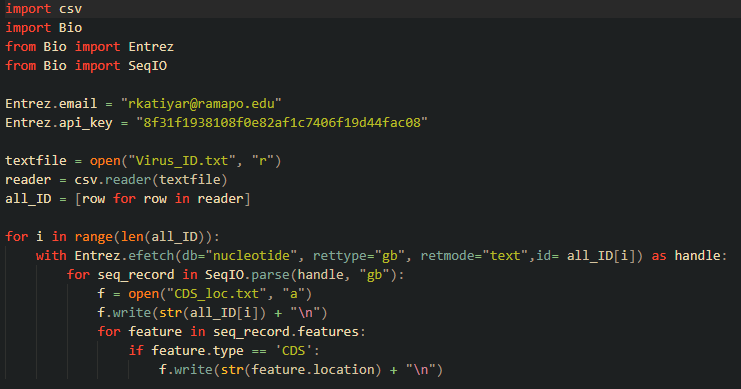


Figure 15: Shows the script for getting the CDS region for each virus

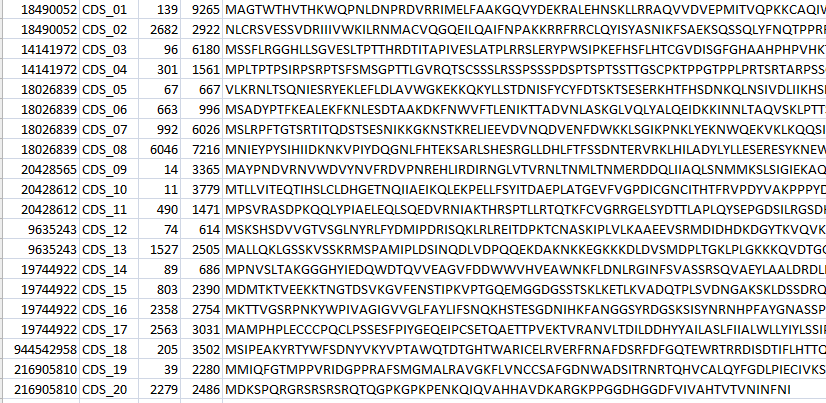


Figure 16: Shows how the data looks in excel. The first column shows the Virus\_ID. The second column shows the Unique\_ID given to every CDS region found. The third column shows the length/ region of CDS within the viral genome. Finally, the last column is the translated protein sequence.

1. Using **NCBI: CDD** data on possible Motifs and Domains within the Translated Protein Sequence was obtained. NCBI:CDD allows for batch sequence searches which looked like:



Figure 17: Shows the CDD batch search.

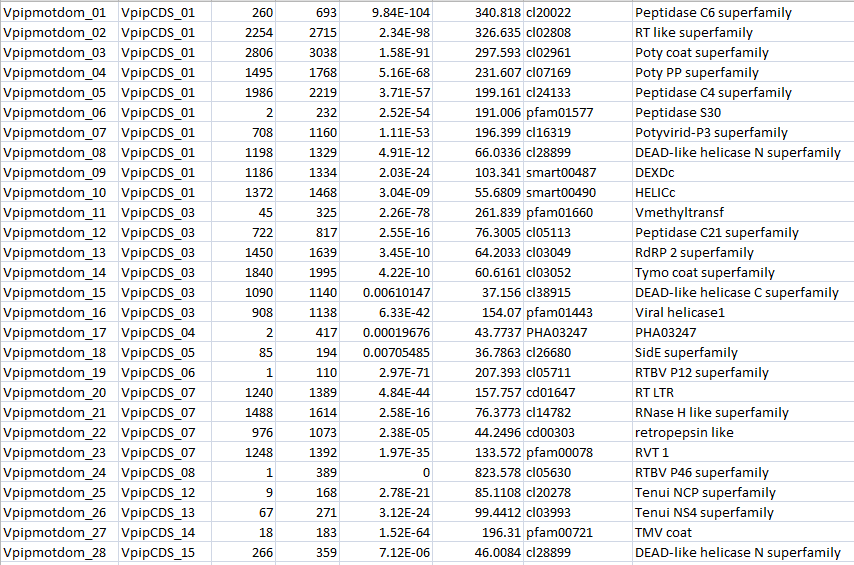
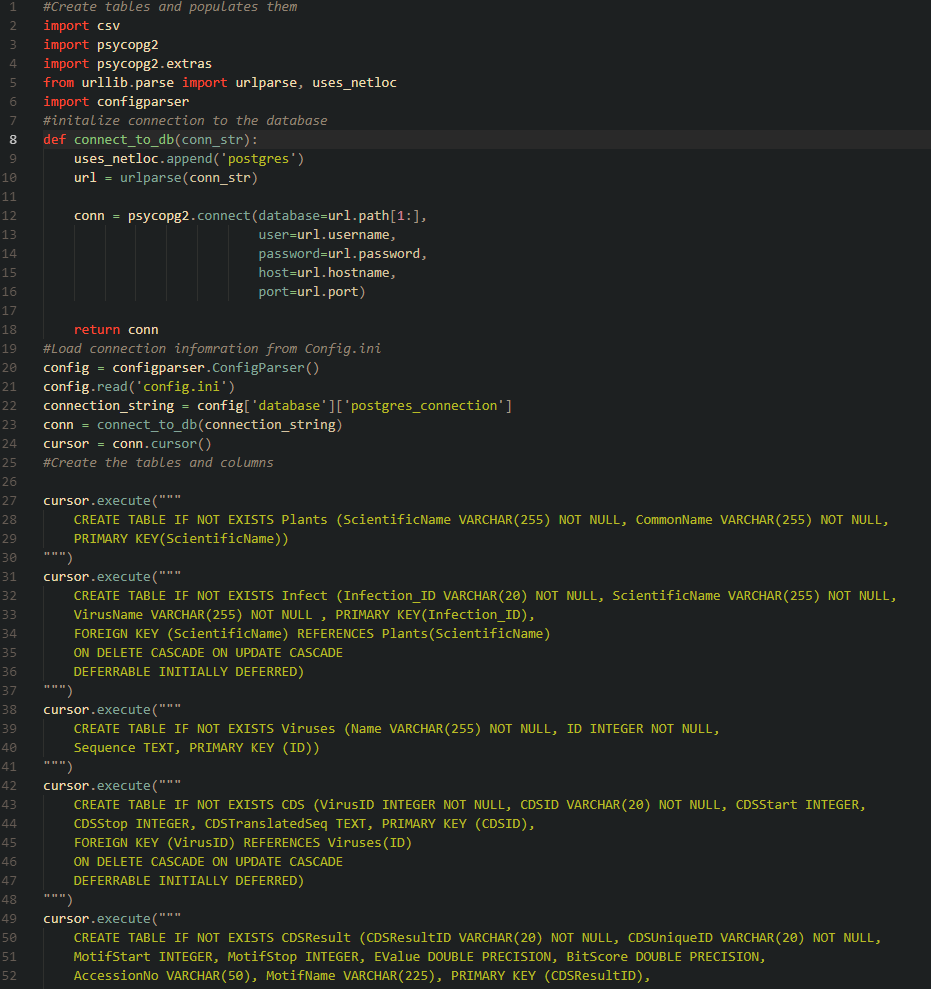


Figure 18: shows the final results that were obtained from the NCBI: CDD database. These results are edited and arranged based on the schema. Look at image 1 for reference.

1. Now that all the data was gathered, the next step was to place the data in the database so that further queries can be written to allow for greater analysis of the data.



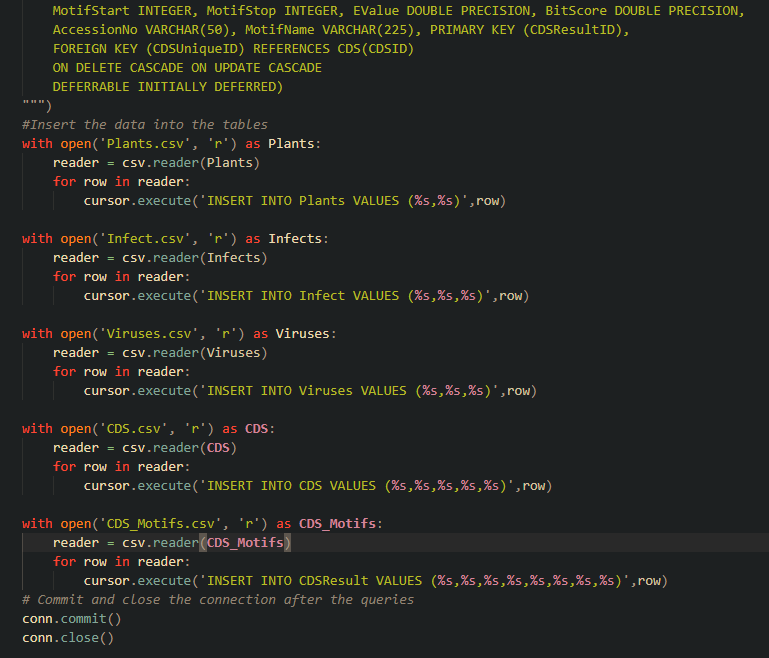


Figure 19 & 20: Shows the scrip used to seed the database.

1. Next to display the data in a table properly from the database a Flask application was set up and quires were written to allow user to further use the gathered data and to create a web interface.



Figure 21: Shows the main flask app code. For full code simply open the “Main.py” file in the application folder.



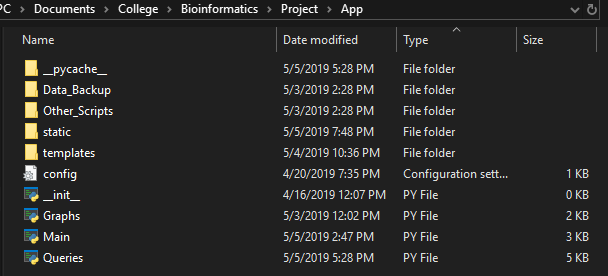
Figure 22: The figure shows the queries that were then called to the database and displayed on the flask application using HTML (to display on browser) and CSS (to style the display).

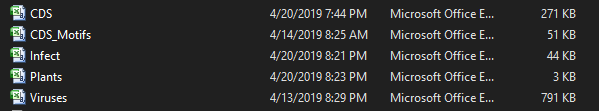
Results

Over All Statistics for the application

|  |
| --- |
| 10 Plant genuses and 73 Plant species were recorded |
| 193 Viruses were recorded that infect at-least one plant species |
| Over all 764 Recorded infections |
| 486 Coding regions identified for various viruses |
| 717 Various motifs identified from the coding regions |

Folder structure of the application





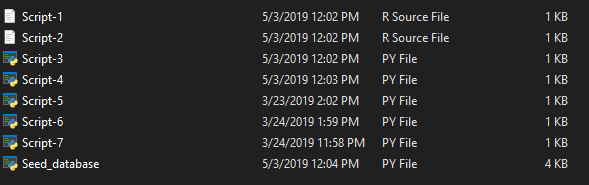


Figure 23 – a: Shows the folder structure of the application and how the scripts were arranged. The \_pycache\_ folder is created at use of the application and holds information on the usage of the app. The Data\_Backup folder (figure 23 - b) contains all the data that was gathered in csv format and tab-delimited format. The Other\_Scripts folder (figure 23- c) contains all the scripts that were used for gathering the data including script used to seed the database. Static and Templates contain CSS and HTML files for the application. \_init\_.py file simply initializes the application and should be empty. The Graphs.py script creates the graph based on the data in the database. Main.py is the main application and is called to run the application. Queries.py connects the database and the application together and calls queries to the database.

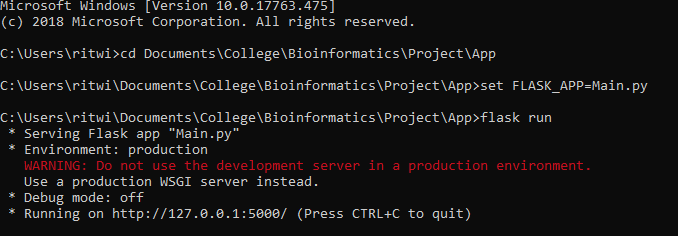


Figure 24: Shows how to run the application once everything was set up.

Application

Home and About pages

The application contains a home page, about page, a browse section (allows the user to browse the all the data that was gathered for the project) and a search bar (which can be set according to what the user wants to search for.

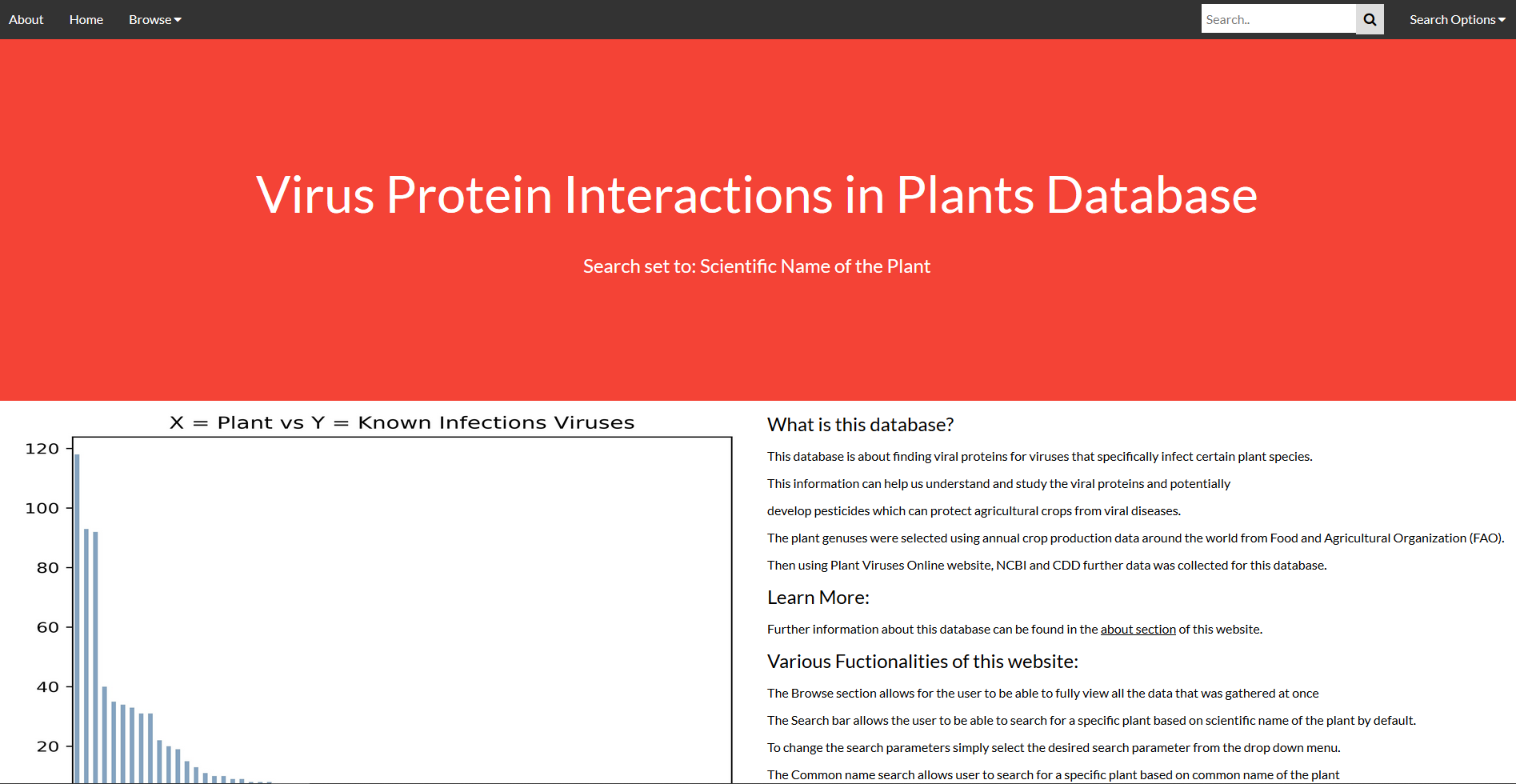


Figure 25: Shows the home page of the application. The graph showcases all the recorded infections for a particular plant species. The home page of the application goes in more detail about the database and about the graph itself.

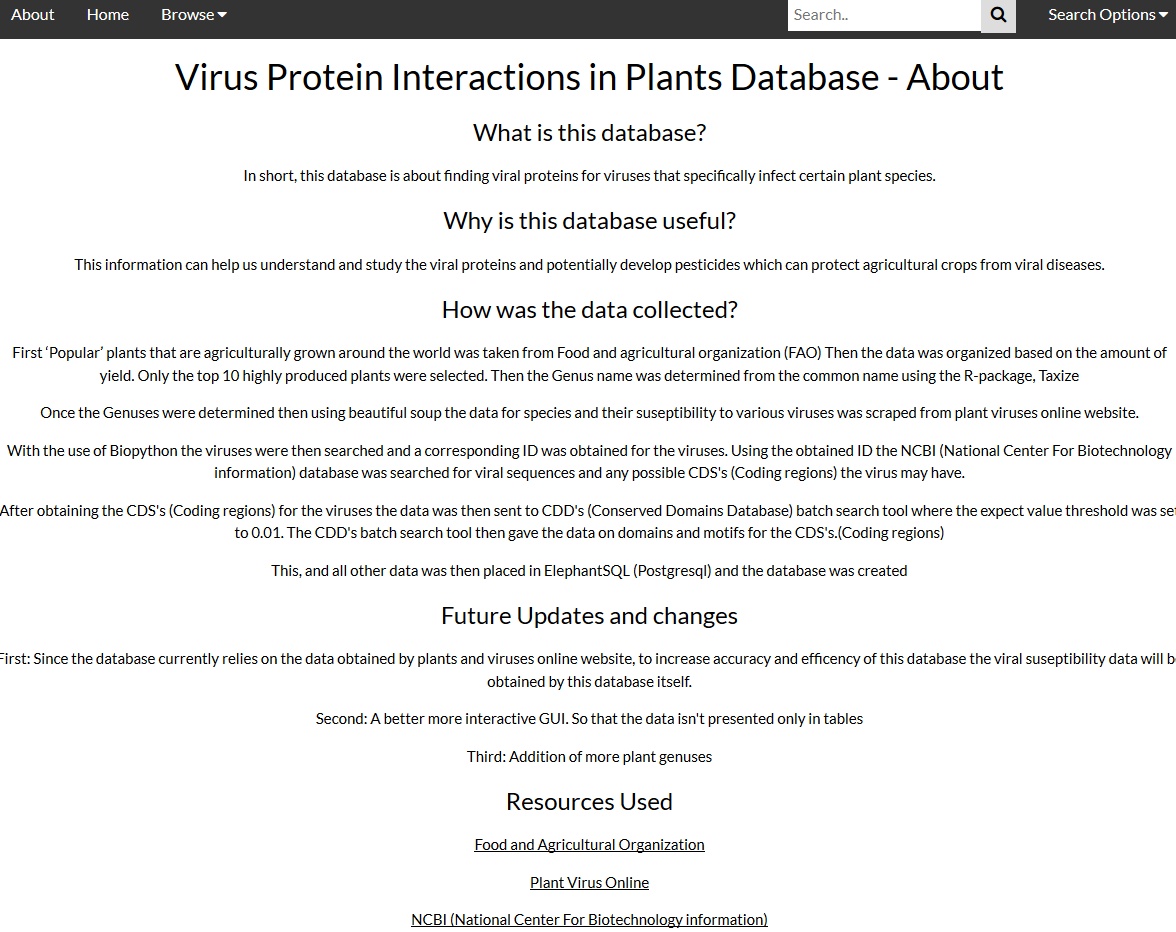


Figure 26: Shows the about page of the application. The about page summarizes how the data was obtained and what recourses were used for the creation of the application.

The Browse section

Contains 5 different tables that showcase all the data from the database.

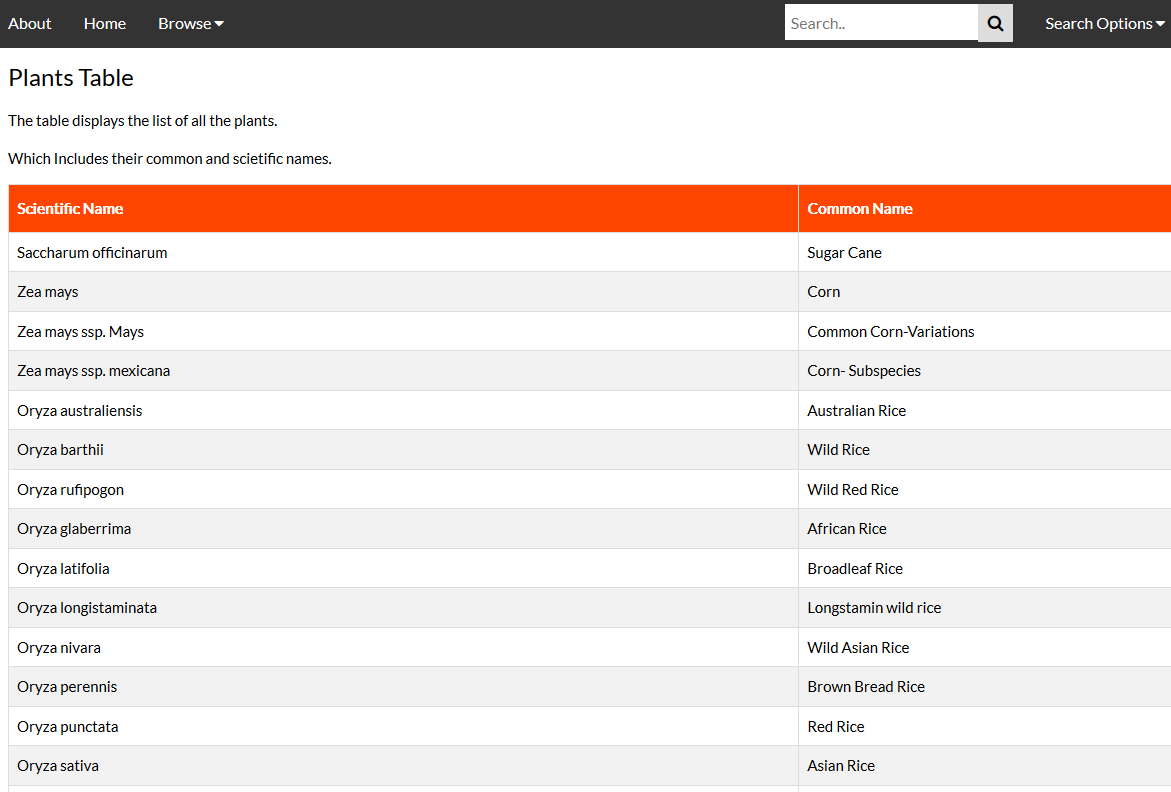


Figure 27: Shows the first table within the browse section.

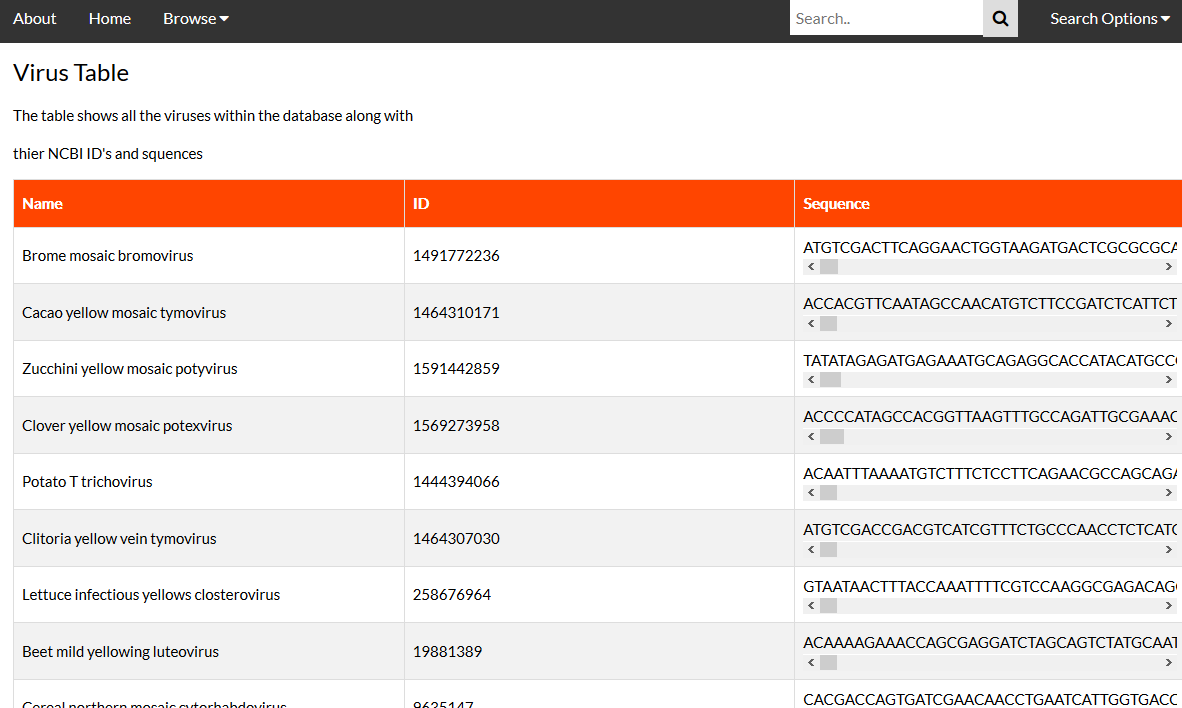


Figure 28: Shows the second table within the browse section. The scroll under the sequences allows the user to view the whole sequence without the need to scroll horizontally on the page itself.

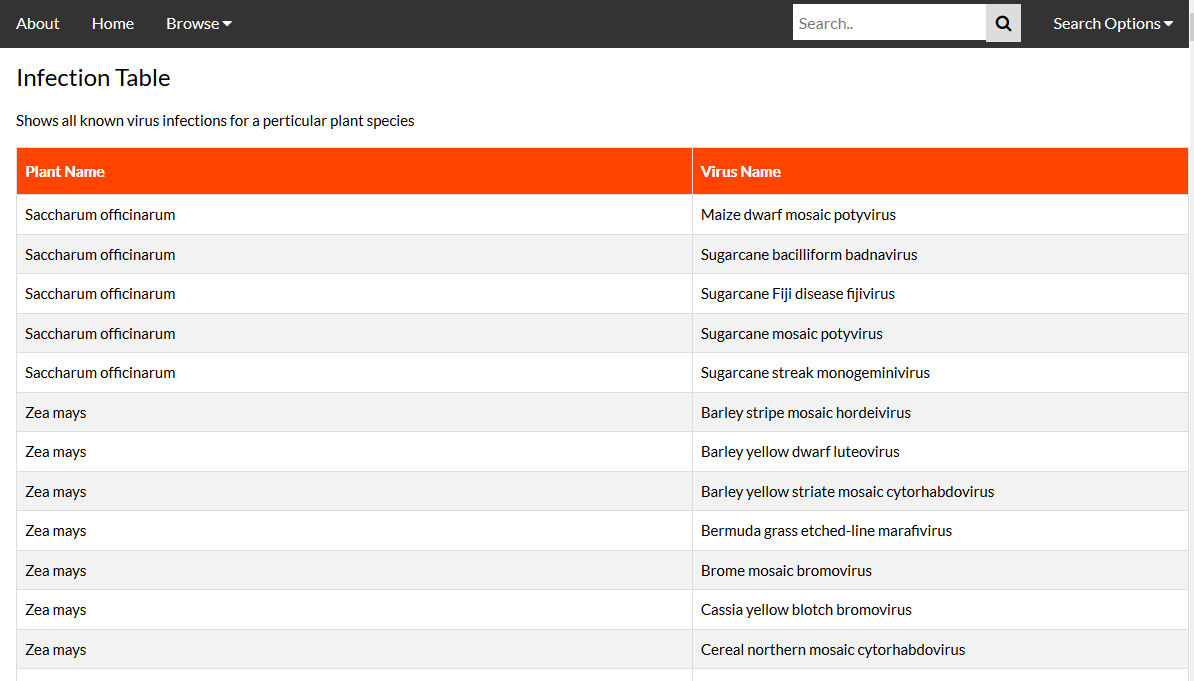


Figure 29: Shows the third table within the browse section.

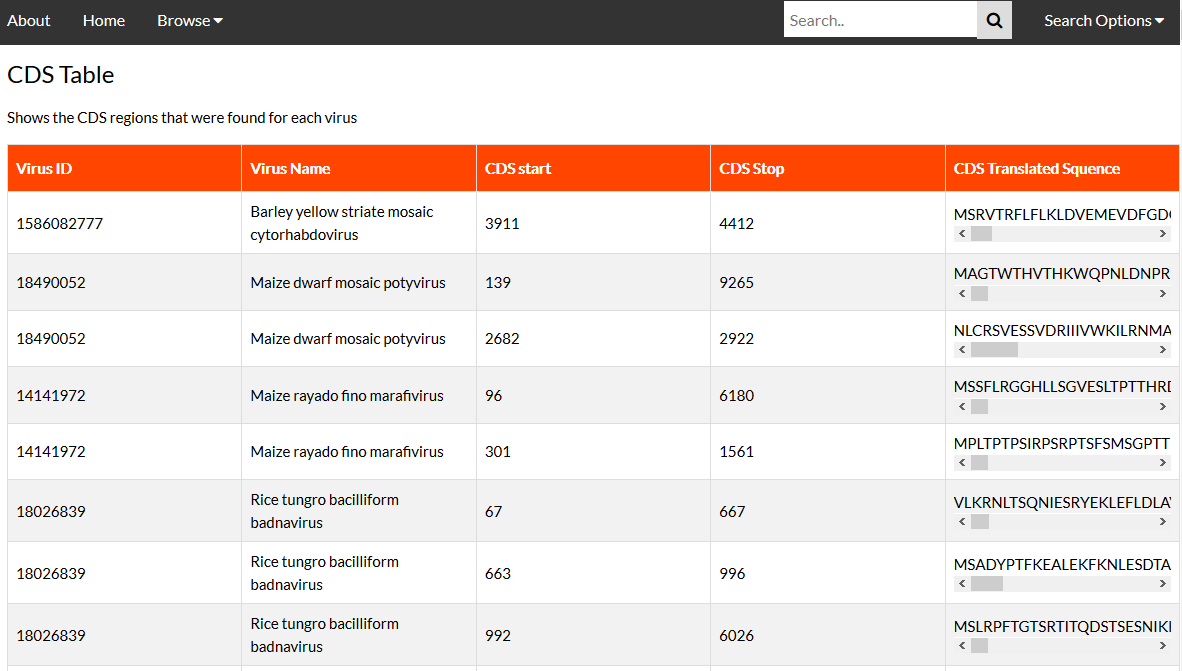


Figure 30: Shows the forth table within the browse section. The scroll under the sequences allows the user to view the whole sequence without the need to scroll horizontally on the page itself.

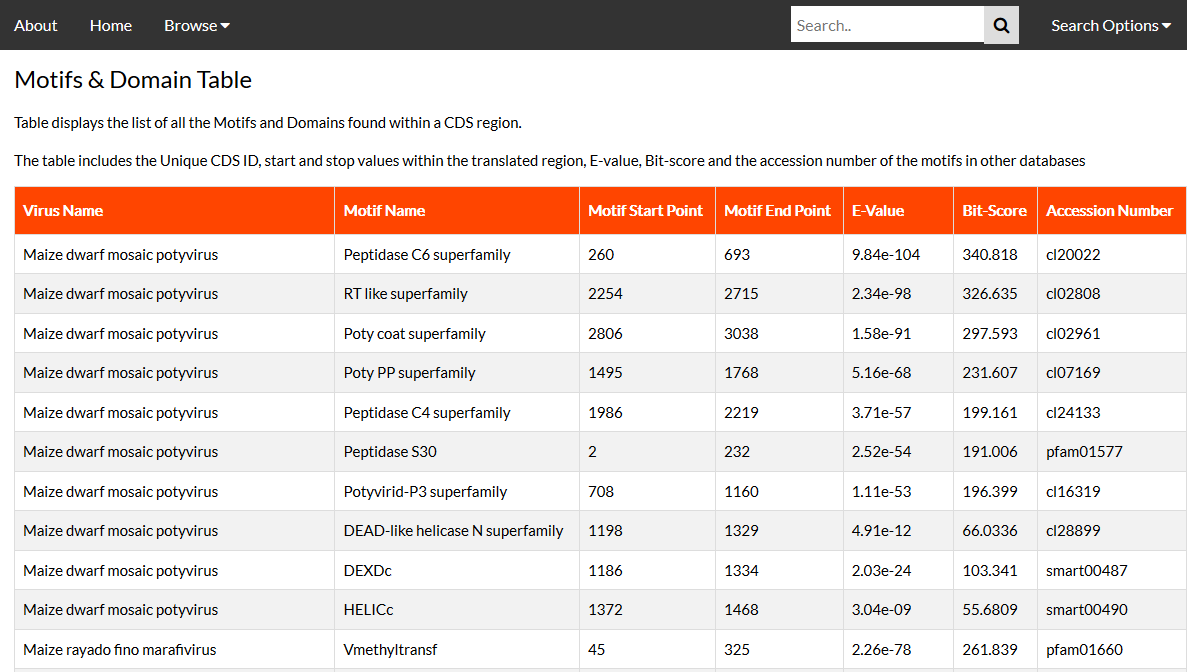


Figure 31: Shows the fifth table within the browse section. The accession numbers for the motifs can be used in the CDD website to search for more information on the motifs that were found.

Search Section

The search section allows the user to search by the scientific name by default. However, it also allows the user to search by common name, virus and motif as well. The search option for scientific and common name of the plant is for the user to see the search results for a specified plant rather than needing to browse through the browse section of the application. The Virus search allows the user to search for a specific virus and view the virus sequence, cds and motifs found for the cds that a virus might posses. The motif search results allows user to search for various viruses that may possess a specific motif; this search also provides the accession number for the motifs.

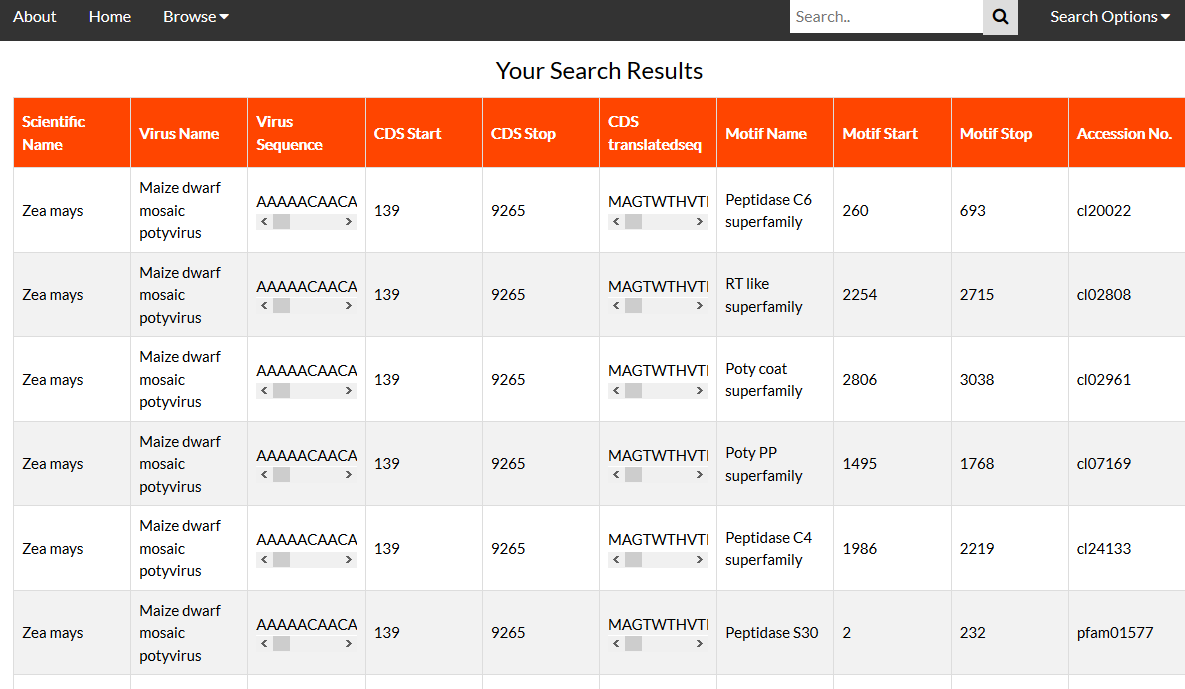


Figure 32: Shows an example of the default search results where “Zea Mays” plant was searched.

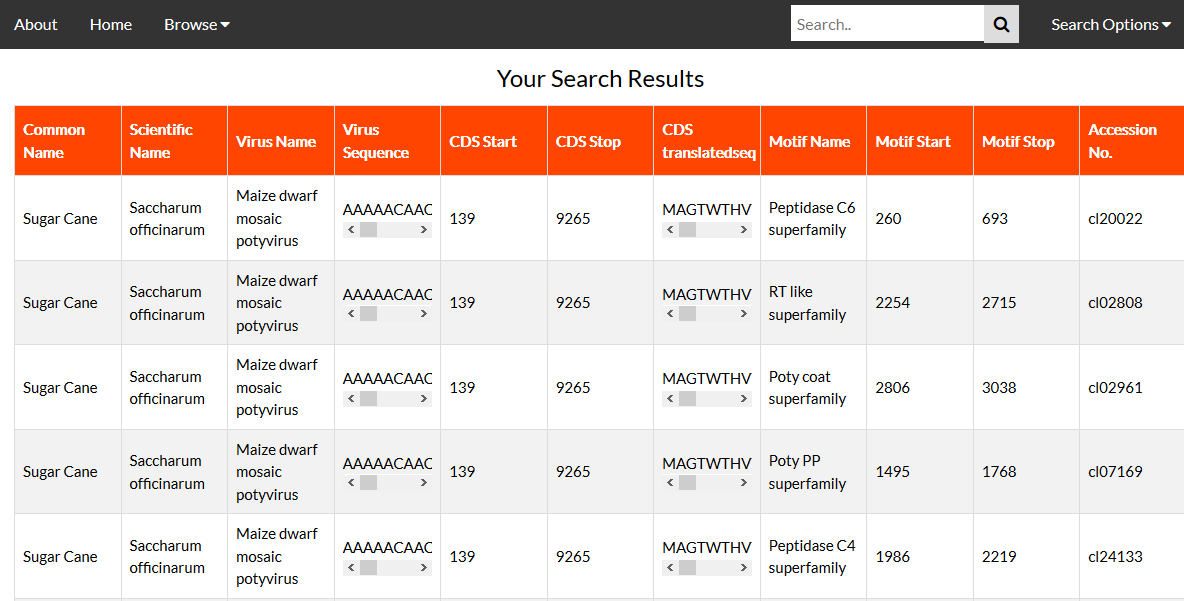


Figure 33: Shows an example where “Sugar Cane” was searched by setting the search settings to “common name” by using the dropdown menu next to the search bar.

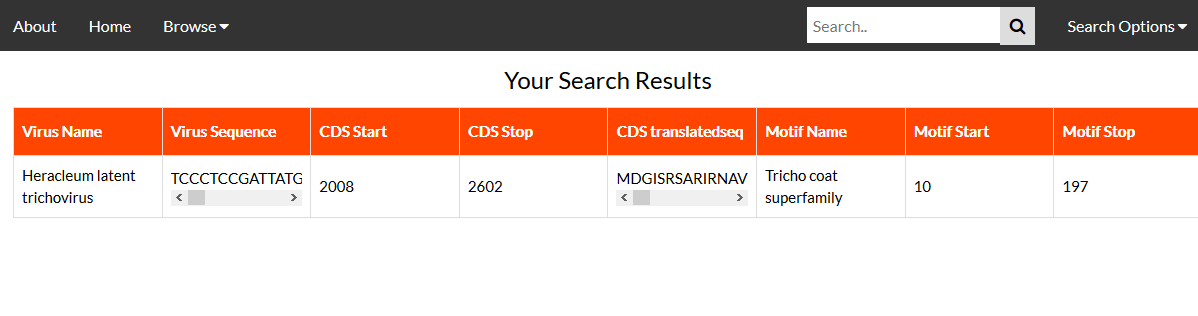


Figure 34: Shows an example where “Heracleum latent trichovirus” was searched by setting the search settings to “viruses” by using the dropdown menu next to the search bar.

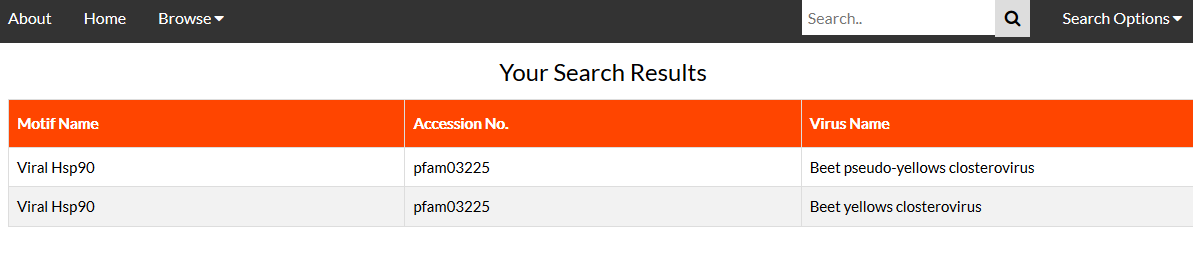


Figure 35: Shows an example where “Viral Hsp90” Motif was searched by setting the search settings to “Motif” by using the dropdown menu next to the search bar.

Discussion & Conclusion

Overall the application was connected and the data was retrieved from the database properly. The overall purpose of the application was fulfilled; however there is a lot of room for improvements. Future updates for the application would involve creating a proper GUI for viewing the sequences and results. The results would not contain redundancy. The reason for redundancy in the current version of the application is simply due to the join statements on the various tables, which are displayed without any edits made to the joined tables. The search bar needs to made more user friendly. Currently to change the search parameter one must use the drop down menu. The drop down menu simply directs the user to a version of the home page where the search bar itself has been changed based on what the user wishes to search for. Another update that needs to be made for the search bar is to make the searches not case sensitive. Currently the searches are very case sensitive and require the user to enter the exact search term. For example, in Figure 33: if the user had searched for “sugar cane” instead of “Sugar Cane” no results would be shown to the user. Additionally, more plant genuses need to be added to the database. The reason why only 10 genuses were used in the current version of the database is because Elephant SQL has a limit of 128 MB of free usage. Once the data exceeds 128 MB the user is required to purchase more space for the database. Finally the application currently requires data from Plant and viruses online website; the website itself isn’t update all too much. Hence, to remove reliance on Plant and viruses online website, an algorithm that generates viral infection data based on plant species is required, within the application. Creating this algorithm would remove reliance on Plant and viruses online website. Despite these future improvements the database is able to record viral proteins along with the type of host the virus usually infects. This could prove beneficial in understanding what the viruses are capable of and thereby giving us a step ahead against countering viruses and protecting agricultural crops.

References

1. (Lecoq) Lecoq, H. “Discovery of the First Virus, the Tobacco Mosaic Virus: 1892 or 1898?” *Comptes Rendus De L'Academie Des Sciences. Serie III, Sciences De La Vie*, U.S. National Library of Medicine, Oct. 2001, www.ncbi.nlm.nih.gov/pubmed/11570281.
2. (Smith) Smith ,H. “Mechanisms of Viral Pathogenicity” *Department of microbiology, University of Birmingham, Birmingham, England*, Vol.36,NO.3 Bacteriological Reviews. American Society of Microbiology, Sept. 1972, https://mmbr.asm.org/content/mmbr/36/3/291.full.pdf
3. (Kamps) Kamps, Bernd Sebastian, et al. *Influenza Book | Pathogenesis and Immunology*, www.influenzareport.com/ir/pathogen.htm.
4. (Lodish) Lodish H, Berk A, Zipursky SL, et al. Molecular Cell Biology. 4th edition. New York: W. H. Freeman; 2000. Section 6.3, Viruses: Structure, Function, and Uses. Available from:
5. (Rui) Rui, Liu Sheng, et al. “MicroRNA-Mediated Gene Silencing in Plant Defense and Viral Counter-Defense.” *Frontiers in Microbiology* , vol. 08, no. 1664-302x, 2017, p. 1801., doi:10.3389/fmicb.2017.01801.
6. (Pratt) Pratt, Ashley J and Ian J MacRae. “The RNA-induced silencing complex: a versatile gene-silencing machine” *Journal of biological chemistry* vol. 284,27 (2009): 17897-901.
7. (Kamthan) Kamthan, Ayushi et al. “Small RNAs in plants: recent development and application for crop improvement” *Frontiers in plant science* vol. 6 208. 2 Apr. 2015, doi:10.3389/fpls.2015.00208