Motivation

Visual networks have been applied to understand the relationships between proteins and genes [1]. These relationship paths allow us to understand how genes or any biomolecular processes change when comparing wild types to disease states [1]. In this study, we are attempting to visualize all the gene interactions that occur within Fanconi Anemia (FA). FA is a rare genetic disease that results in improper response to DNA damage [2]. Here we are provided with an input file (GMT format) that contains all the loci associated with FA. In this study, a program named *simple\_path* was developed that takes in these input files and generates .SIF files. These files can later be uploaded to a network visualization software known as *Cytoscape* [3], where you can visualize how the genes interact in a single locus and all loci.

Computational problem

With the provided STRING.txt data set, where each row contains information of the gene interactions and its weights, will generate a .SIF file that will be used for uploading on cytoscape. The provided input file will be used to extract all interaction pairs that exist within each locus and will be searched within the STRING.txt dataset. Possible limitations is redundancy, where the STRING.txt contains reversed interaction pairs that have the same score. This can affect the network that is plotted on Cytoscape.

Specific Approach:

With the given input file, the gene interactions within each loci will be extracted and queried into the STRING.txt dataset. The obtained matched pairs will be written into a .SIF file and then loaded into Cytoscape to visualize genetic interactions.

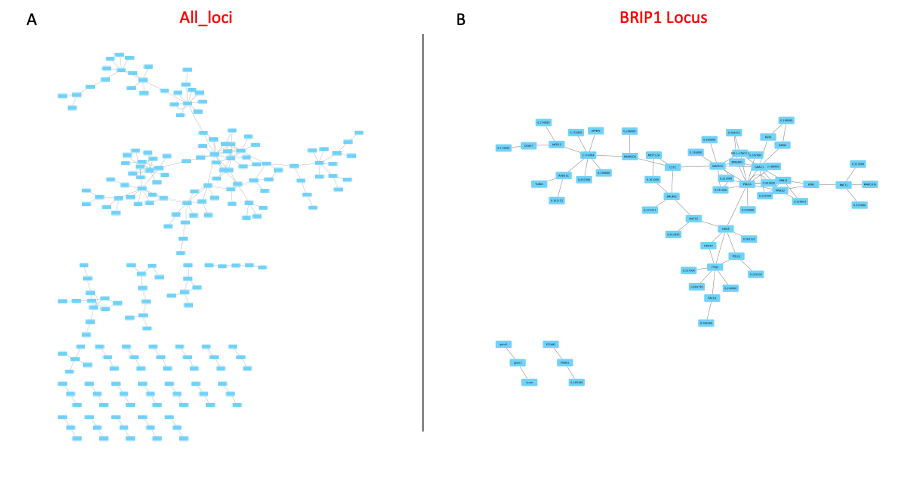
Specific Implementation:

The input file will be parsed and converted into a python dictionary (similar to as Hashmap), where each key will be the name of the locus and the values will be an array of genes associated. Similar to parsing the input file, the STRING.txt will be converted into a python class object, named StringDB, that utilizes python’s dictionary internally when using its class specific methods. The StringDB has a method called get\_pair\_score() where it takes in the locus name, array of associated genes and interaction types. In the method, the array gets passed through one of python standard packages called itertools using the permutations function, where it creates all possible combinations within the array. These pairs are passed as queries to the StringDB object and returns the score. The method returns a python dictionary that contains the locus\_name and the two genes with its interaction score. The returned dictionary is passed through a function save\_as\_sif()where it outputs a .SIF file per locus and a .SIF where all loci results are merged.

This implementation has its advantages. The use of python’s dictionary has a complexity of O(1) when submitting a query. However, python dictionaries consume more memory than other data types like lists or tuples. In addition, another limitation is the use of the permutations as it returns a generator type. Generators are iterators that are not loaded into memory until called. Once the generator is used, the data contained within no longer exists in memory. One major drawback of using generators is speed. It is much slower when compared to lists, tuples, or dictionaries when doing an iterative process. Some alternative approaches would be to change the data types into pandas DataFrames, as it has the capability to vectorize processes and replaces all iterative processes done with the generator [4].

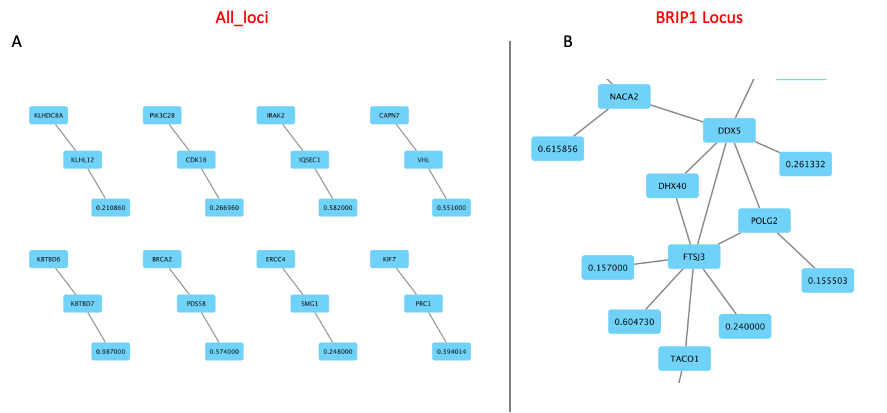
Results:

Running the command python simple\_path.py -i Input.gmt.txt -o FA\_net\_out -t pp, -db ./Data/STRING.txt will output a folder named with the provided output name along with a unique ID (Example: Fa\_net\_out-092321-152431), where the unique id consists of date and time format (MMDDYY-HHMMSS). This directory will have two groups of .SIF files. The one labeled with the all\_loci indicates that the .SIF file contains all loci interactions into one file and the rest of the files are interactions in a single locus. The generated .SIF files were uploaded to a program called cytoscape (fig 1) to visualize the interactions. However,



***Figure 1:*** *Visualization networks of both All Loci and BRIP1 locus*

upon closer inspection, there are some inconsistencies when plotting the .SIF files onto cytoscape (fig 2). Some of the data were not plotted into a net like format therefore -



***Figure 2:*** *Showing inconsistency plots from both the all\_loci and BRIP1 plots.*

causing the interactions between these genes to be meaningless (fig 2A). In addition some of the nodes are weight values and not gene names (fig 2B).

Conclusion

The developed .SIF from simple\_path script did not provide the most accurate network interaction The results show that there are some inconsistent seperations of genes and some genes were linking to interaction values. This issue can be narrowed down to either the format of the .SIF file created by simple\_path or how cytoscape internally processes weights. These issues can be easily tested by just removing the weight values to check if indeed the values are what's causing these inconsistent sperations. In addition, future performance implementation will be added by incorporating the Pandas DataFrame implementations to allow vectorized processes.

Sources:

1. Merico, Daniele, David Gfeller, and Gary D. Bader. "How to Visually Interpret Biological Data Using Networks." *Nature Biotechnology* 27.10 (2009): 921-24. Print.
2. Nalepa, Grzegorz, and D. Wade Clapp. "Fanconi Anaemia and Cancer: An Intricate Relationship." *Nature Reviews Cancer* 18.3 (2018): 168-85. Print.
3. Shannon, P. "Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks." *Genome Research* 13.11 (2003): 2498-504. Print.
4. "Pandas." *Pandas*. Web. 24 Sept. 2021.