Angela Sofia Burkhart Colorado

7711 2021, Module 4

Day 3 Written Report

**Motivating Problem:**

When picking genes of interest in relation to a certain phenotype most researchers have used databases and previous literature to identify these genes and study them. Unfortunately, we know that even the most well studied genes are not necessarily the only ones impacting the trait, but it can be difficult to weed out those that may be more relevant than others. Furthermore, not all genes may have the same scale of impact but can remain important to maintenance of a certain phenotypes. In addition, this problem is exacerbated by the fact that in the age of big data one sequencing experiment can lead to hundreds of significantly differentially expressed genes. As a result, being able to illustrate and quantify potentially relevant genes is important for future investigations and hypothesis testing while at the same time eliminating some potentially less impactful genes from the research. This ability would increase efficiency, save time and money, and ideally yield relevant results when examining the genotype causing a certain phenotype.

**Computational Problem:**

To address this problem, we used a networks-based approach wherein subnetworks of genes associated with Fanconi anemia (FA), a disease caused by genomic instability, were created, and optimized (using a genetic algorithm) to be more connected. Using this population of highly connected subnetworks we got two main results. The first was a gene score for each gene associated with FA by evaluating how connected all FA associated genes will be in each optimized subnetwork. The second result that we wanted to determine was whether these optimized subnetworks tended to more connected than subnetworks generated from random noninformative loci. A p-value was then calculated to evaluate whether the difference in connectivity was significant. Ideally a significant p-value would indicate that the optimization had worked. On another note, we also wanted to evaluate the performance of the genetic algorithm therefore, summary statistics were calculated. The combination of these three outputs provides one with networks of FA relevant genes as well as gene scores that indicate which genes are theoretically more relevant to the disease phenotype.

**Specific Approach:**

This script has one main function that calls on several others but is divided into four main parts. The first is the genetic algorithm, and its output. Here 5,000 random FA associated subnetworks with 12 genes each (one for each FA loci) are picked. A selection score is then calculated for each subnetwork by summing the weighted edges in the network. Those networks are then mated where each network is one parent, and the second parent is chosen by selectively picking networks with higher selection scores[5]. These two parent networks then create a child network wherein a gene at each locus in randomly picked from the two options given by the parents. After 5,000 child networks are created if the average change in edge density is over 5% then the children become the parents and the new generation is created from them [5]. Each gene in each generation also has a 5% probability of being mutated. In other words, a new gene from the same loci will be picked 1 in 20 times. However, once the change in edge density is not significant the genetic algorithm stops, and one is left with an optimized set of 5,000 FA associated networks. In addition, wile the genetic algorithm is running, the selection score or edge densities of every subnetwork is calculated for every new generation of subnetworks. This creates a distribution of edge densities of which the mean, standard deviation, and variance is recorded. Furthermore, frequency histograms are plotted to show the change in distribution between each generation (Figures 1A, 1B, and 1C).

The second part of this script is where the noninformative loci are generated and a p-value is calculated to compare the FA associated population of subnetworks to the noninformative populations of subnetworks. This is done by first binning all the genes in the STRING database by the number of genes they can connect to in a network. Tasan et al. used 128 bins therefore that is the default for this script as well. Consequently, 127 bins of equal size were created, but the last bin contained fewer genes as it was the remainder of the total number of genes divided by the number of bins. Once each gene had its own bin, one population of 5,000 subnetworks from noninformative loci were created. The loci were determined by the bins that the genes in the FA associated subnetworks were in. Therefore, if the gene from FA locus 0 in the first FA subnetwork was 5, then the first gene in the first noninformative subnetwork was chosen from bin 5. This process was then repeated 1,000 times leading to 1,000 populations of 5,000 networks [3]. Average selection scores were then calculated for all 1,000 populations as well as the one FA population. In order to calculate the p-value the number of average edge densities from the noninformative populations that were higher than average edge density of the FA population was summed and divided by the number of populations, which in this example 1,000. If the resulting proportion was less than 0.05 then FA edge density would be significantly more connected.

The third portion of this script was where the top 10 subnetworks with the highest selection scores were written into 10 files to be visualized in Cytoscape (Figure 2) [4]. This section would take all the genes in the network find which were connected and record the weight of the connection so that Cytoscape could visualize edge densities.

Lastly, the final output of this script was a file containing the name of the FA associated locus, its description and list of genes at each locus with their respective gene score. The gene score was calculated by taking each FA associated subnetwork and calculating the sum of the weighted edges of every gene at that locus in each subnetwork. For example, every gene from FA locus 0 was put into the first subnetwork and the sum was computed. Then the same was done for the genes in FA locus 1 and so and so forth. This was then done for the full population of FA subnetworks [5].

**Specific Implementation:**

There were several portions of this script that may have led to biased results. First was the fact that since the noninformative populations were not each run through the genetic algorithm the p-value calculated was 0 as no randomly chosen network had the level of connectivity that FA ones did. The reason for this decision was twofold. The first was runtime, overall, the program takes around 5 minutes to fully execute at the moment. However, that would be significantly increased if the genetic algorithm was applied to all 1,000 populations of 5,000 noninformative subnetworks. The second reason was that in using noninformative genes located in the same bin as the original FA associated gene, this bias might be mitigated. However, unfortunately the code used to bin the genes did not take into account the weight of the edges and instead only counted each edge as 1 (since that is what I calculated in Module 2). Consequently, these noninformative populations may not reflect neither an optimized subnetworks nor similarly connected genes. Nevertheless, it was necessary to get 1,000 populations of subnetworks in order to create a distribution of edge density averages (that was calculated using weighted edges), therefore the code does reflect this part of the assignment. One final thing to consider as well is that the last bin can have considerable smaller number of genes since it is the remainder of the total number of genes divided by the specified number of bins. In this case, most bins contain around 132 genes but the last contains 33.

In addition, there is nothing in the genetic algorithm that prevents a parent network from mating with itself. This is unfortunately not representative of what typically happens in sexual reproduction therefore it does not model reality, and it could have been corrected with the use of an IF statement. Furthermore, in the Tasan et al. paper the selection scores were calculated by taking unweighted edge density cubing it and then diving that number by the sum of all the selection scores. In essence, each selection score eventually became a proportion. In my implementation I chose instead to take the weighted edge density of a subnetwork, round to two decimal places and then multiply by 100 to get whole numbers. In doing so, while the magnitude of the scores was not increased the addition of the weighted edge density and the multiplication kept the variability of the scores. I also needed whole numbers to create what I called a priority list. This list had each selection score’s index value (from 0 to 5,000) and the number of times the index number appeared in the priority list was the selection score value. This list is what was used to preferentially pick parent networks with higher connectivity as their selection scores would be higher and their index values would therefore have a higher chance of being picked from the list.

When looking at the visualized histograms (Figures 1A, 1B, and 1C) for each generation it is important to see that there is a significant change over each generation where the mean, the standard deviation and the variance increases. After running it several times, these results remain consistent and in fact the program averages 15-20 generations before it is fully optimized, which proves that it is working as the one in the Tasan et al. paper. In addition, the visualization of the network (Figure 2) also shows that the mutation portion of the genetic algorithm is working because there is one node (ALG1) that remains unconnected. This is representative of the biology involved in inheritance of genes where most come unchanged from the parents, but some can be mutated potentially creating a new phenotype which, in turn, is how evolution occurs.

Another thing of note however, is that the weight of interactions between the genes tended to not be very different, for example, many edges only had a 0.001 difference in weight therefore when visualizing in Cytoscape, the edge densities do not show a lot of variation. This could have potentially been corrected by cubing the edge densities changing the order of magnitude and emphasizing the differences in weight.

Figure 1: Example histograms of the selection scores for generations 0, 8 and 16 in one run of the genetic algorithm

Chart, histogram

Description automatically generatedChart, histogram

Description automatically generatedChart, histogram

Description automatically generated

C

B

A

Figure 2: Highest scoring network after genetic algorithm visualized in Cytoscape

Diagram

Description automatically generated

Citations:

1. Discussed possible pseudocode and implementation of problem with Katherina Cortes.
2. Dr. Mazen CPBS7711 lecture. Analysis and Computational design.
3. Kechris, Katerina (2021). *Re-sampling and Simulation Based Methods*[PDF slides].
4. Talon Arbuckle visualized text files in Cytoscape.
5. Taşan, Murat, et al. “Selecting Causal Genes from Genome-Wide Association Studies via Functionally Coherent Subnetworks.” *Nature News*, Nature Publishing Group, 22 Dec. 2014, [www.nature.com/articles/nmeth.3215](http://www.nature.com/articles/nmeth.3215).