SINEs Clustering

Atara Zohar 318286150
Moria Maman 207595505
Naama Hartuv 315745828
Yael Hava 313417420
Advisor: Dr. Anat Paskin-Cherniavsky

1 Abstract

Until a few years ago it was thought that death caused by aging was a normal death. Unlike death caused by disease, death caused by aging was considered a natural condition and therefore no efforts and actions were taken to prevent it.

As researchers over the years have been able to find treatments for various, unrelated to age, infectious diseases, a person's average life expectancy has increased. However, the researchers noticed that there are diseases that are likely to be caused as a result of the aging process, such as cancer, heart diseases, Alzheimer's and more. Therefore, in recent years, the possibility has come to mind that it may be possible to extend life expectancy. Since then researchers have been trying to find the main cause of aging in order to prevent it and fight it.

In our research we will focus on developing a tool for researchers in the field of SINEs in aging. The SINEs are Different parts of DNA that over time mutate and replicate themselves. The replication of the SINEs can interfere with the proper functioning of DNA, which can accelerate the aging process. Our goal is to be able to cluster the SINEs into similar families and then refine those families, in order to find smaller segments that will represent SINE more reliably.

2 Background

People are making a lot of efforts to maintain a healthier and longer lifestyle. Activities such as fitness and maintaining a diet can help, but they can not prevent the aging process.

Researchers believe that in the future we can live longer lives. They believe that the aging process is caused by an abnormal process in our body, and that process is probably the cause of various diseases in old age, and so aging can be related as a disease.

Various studies have believed that several processes are the main causes of the aging process. Most of these processes involve DNA damage. The goal of the researchers is to understand how those processes are created and how they affect our bodies. It has not yet been understood how the processes are related to each other and how they work in general. All of these processes happen in very specific places in our body and therefore do not affect more complicated systems in our body. Therefore, the researchers hopes that the better we understand how those processes work and affect each other, the simpler the conclusions will be and the way to conclude them will be shorter.

In recent years, the interest in researching the aging process has been rising, and many articles are being written on the subject.

3 About The Research

The research is conducted by Dr. Andrei Gudkov of the Roswell Cancer Research Department. It deals with the aging process that focuses on the process of cumulative damage to DNA. Cumulative damage is an outbreak of early viruses that are manifested as an accumulation of mutations formed in DNA and can cause disruption of normal DNA function. These mutations are called "retrotransposons" which are genetic factors that copy themselves into RNA and then convert back into DNA and are inserted into the genome.

There are different types of retrotransposons and some are called SINEs, these are small sequences that are not encoded and appear in millions of copies within the DNA. The researchers believe that the appearance of retrotransposons may cause the appearance of aging cells that contribute to the development of diseases and the aging process.

4 Project Goal

Our project goal is to help the researchers to verify their hypothesis - that SINEs are part of the factors that helps the aging process. By refining and shortening the representation of a SINE.

The researchers currently search for SINE by comparing it to a single long string (named B1). The problem with this comparison is that there is a large variance, so a lot of SINEs can be missed, and it is possible to mistakenly identify certain sequences as SINEs even when it is not. It happens because the average distance from B1 is quite large.

In our project, instead of one point, we found 3 strings that represented the centers. Now, instead of comparing the sequence only with B1, they will be able to comparing with any of the three centers we found and take the nearest one.

In addition, the refining of the SINE's structure will increase the reliability of the SINEs identification. The average distance size will be reduced and be more accurate, and in addition researchers will be able to identify SINE in a more reliable and accurate way.

5 Research Phases

The process we performed consists of two phases. In the first phase getting rid of data duplications and in the second phase clustering SINEs to families.

5.1 Phase one - Get rid of data duplication

The process of reading DNA is a complicated and problematic process. For example, to map DNA sequence of one mouse, one must look at millions of cells and from them try to map parts of the overall DNA sequence. In fact, the DATA we worked with came from millions of mouse cells, of which only a thousandth manage to survive, which makes it possible to map only part of the DNA sequence, so the resulting data is not so reliable.

For example: (and then the sketch)

There are millions of cells and from each cell only a small part (thousandth) of the DNA copy can be mapped. Each section was able to map in about 30 cells. Therefore, in the final data there are many duplicates that must be got rid of so that we do not count the same areas several times and thus we get more reliable information.

In order to get rid of duplications, we inserted barcodes into a unique data structure. A barcode is a 36-string sequence of letters that appears before the SINE in the DNA segment. To get rid of duplications we use barcodes instead of SINEs because the SINEs enter random places in DNA, so there may be a situation where we see the same SINE in two different places. If the same SINE is seen twice, it is impossible to know if it is from the same place in the DNA, or if it is in two different places in the DNA. In case the SINE is from the same place in the DNA we want to count it as one, and in case it is not from the same place, we want to keep it as two different appearances. Therefore, we will use the barcode so that we can identify whether it is a duplication of the same SINE or two different SINEs.

The data structure into which we inserted the barcodes is made up of a dictionary whose keys are all the 9 permutations of the letters A, T, C, G and N. A, T, C and G are the letters representing the DNA and the letter N represents all cases where the DNA reading has not been clearly identified. The value of each key is a dictionary whose key is a barcode that contains the same permutation. As mentioned earlier, a barcode is a sequence of letters - the same letters that make up DNA. As we mentioned earlier, a barcode is a sequence of letters- the same letters that DNA is composed from. The value of each key is the ID of that barcode, so each full barcode has only one ID that represents it. The unique data structure helped us retrieve the barcodes in a convenient and fast way.

The code we received had a software engineering problem that we had to solve. In order to make the process parallel, there were processes that ran in parallel on the data but posed a problem for us in running the code, so we had to remove them.

5.1.1 Graph Production

- We went through the barcode file, which still has duplicates. We added a barcode to be a vertex in the graph only if its ID matches the one in the dictionary (in order not to insert duplicates).

- We have created for each barcode a tuple named match that keeps all the barcodes that are different from it at an Edit Distance of 3 and below. Edit Distance counts the minimum number of operations required to transform one string into the other. the operations are removal, insertion, or substitution of a character in the string.

- We went through the match list of a barcode and created edges between this barcode and the other barcodes in its list, provided that identical barcodes would not be connected.

We got a graph that connects similar barcodes in Edit Distance up to 3, where we can find connected components. Each component will describe a set of barcodes that are similar among themselves. If the binding component is a clique, it means that the whole group is similar to each other and we can take only one barcode sample that will represent the group.

The image (which should be added) shows the distribution of the cliques' sizes, each one represents a set of similar barcodes that can be removed and thus avoid duplications. Out of XXX in total we reached YYY .....

5.1.2 Graph Analysis

Now, we have analyzed the findings obtained from the graph:

- We found the number of connected components in the graph and saw how many of them are cliques and how many are not.
- We checked the size of the biggest connected component (whether it's a clique or not).
- We built a histogram of the component sizes and from it we concluded the distribution of the sizes of the connected components.

Building the graph from the large file of barcodes took plenty of time, around 4 days. The barcodes file contains about 3,000,000 lines, with each line containing a barcode. So, to get findings faster, we ran the algorithm on smaller parts of the barcode file - first 10,000 lines, then 50,000 lines and finally 200,000 lines. We built a graph for the small parts and saw that the results obtained are close to the expected results. Therefore, in order to get faster results in building the graph from all the data, we used the multiple processors in the machine.

The machine we ran on contains 8 processors, so we decided to divide the barcode file into 7 parts, so that each part of the barcode file will run on one processor and in total uses 7 of the processors. We were able to reduce the running time to about half a day.

5.2 Phase two - Clustering the SINEs to families

Now, after getting rid of the duplicates we wanted to cluster the SINEs into groups. Each group will have a center, i.e. a central SINE, which around it, the other SINEs, who are relatively closest to it, are gathered.

5.2.1 Finding a suitable algorithm

At first, we wanted to use the k-means algorithm. This algorithm uses a function that calculates distance using Euclidean Distance[1]. Euclidean Distance is used to calculate distance between numeric values. Our data contains strings and therefore we could not use this algorithm.

After further inquiries, we realized that the k-medoids algorithm could be used. Unlike the k-means algorithm, with the k-medoids algorithm it is possible to use a function that calculates distance between strings. We used an implementation of the k-medoids algorithm[2], which uses the Jaro - Winkler Distance[3] function. Jaro - Winkler Distance is a method of measuring distance between strings.

5.2.2 Using k-medoids algorithm

The k-medoids algorithm is a clustering algorithm. it divides the SINEs into k groups, and each group has a central point (SINE). The other points are clustered around the central point, and selected according to the distance in the similarity between them and the central point. k points will choose to be central as the sum of the distances between them and the other points is lower.

The disadvantage of the k-medoids algorithm is that its runtime is $O(n^2)$. Therefore, in order for us to get results in a reasonable time, we sampled 1,000 random samples from the SINEs file that we ran on the algorithm to get an approximation of the result.

6 Future work

- We divided the graph into connected components and examined all the connected components that are a clique. Now, we would like to go over the connected components that are not a clique and check how many edges

[1] https://en.wikipedia.org/wiki/Euclidean_space
[2] https://github.com/oscarbyrne/string-clustering
[3] https://en.wikipedia.org/wiki/Jaro-Winkler_distance

are missing for the component to form a clique. If the number of missing edges is low, we will consider it a clique. This way we will reach a higher level of accuracy in reducing duplications.

- At this point we run the algorithm k-medoids on DNA from mouse liver cells. We would like to run it not only on liver cells but also on lung cells and see the results.
- We ran the algorithm on only some of the data to get results faster. We would like to run the algorithm on all data to draw broader conclusions for the study.
- To try to filter out noises of DNA segments that are not close enough, in a particular Edit Distance, to any of the points.
- Find a way to represent the SINE strings using numeric vectors, so that the k-means algorithm can be used, thus reducing the runtime to O(n).