Simulating Germinal Center Dynamics using Genetic Algorithms

[*]University of New Mexico Complex Adaptive Systems. Code available at: https://github.com/AbubakarKasule/Germinal-Center-Genetic-Algorithm

Abubakar Kasule
Computer Science (graduate student)
University of New Mexico
Albuquerque, NM USA
akasule@unm.edu

***Abstract*—In this paper, we model germinal center dynamics using genetic algorithms. Because many aspects of germinal center dynamics are still unknown, we propose a model that takes what is known about germinal centers and speculates on the unknown areas of germinal center dynamics. Our model was effective at accomplishes its task, but we do acknowledge that it is simplified and does not fully replicate even what we do known about germinal centers. However, we do believe that it provides valuable insights into what could be happening in germinal centers.**

***Index Terms*—germinal centers, immunity, epitopes, genetic algorithms**

I. Introduction

The adaptive immune system is the primary mechanism our bodies use to protect themselves from novel pathogens. In the adaptive immune system, structures called germinal centers, which are located in the lymph nodes, help the body protect itself by facilitating the emergence of B-cells with antibodies that can bind to previously unseen pathogens [1] [2]. While germinal center dynamics are not currently well understood, there is a growing body of literature attempting to analyze how they function. The process of creating B-cells with antibodies that bind well to a given antigen is thought to be similar to our understanding of Darwinian evolution. Inside germinal centers, B-cells undergo somatic hyper-mutation and are selectively chosen to proliferate based on the binding affinity between their antibodies and the epitopes of the target antigen [1] [3]. To increase the probability that an antibody with high binding affinity emerges, an enzyme called AID is used to introduce mutations on the portion of an antibody's DNA that encodes the genes responsible for its antibodies.

Because of the proposed evolutionary nature of germinal center dynamics, genetic algorithms are an appropriate tool to computationally simulate germinal center dynamics. Genetic algorithms generally consist of 5 core features: a population of individuals, a function to measure the fitness of an individual in the population, a function to select which individuals in a population should proliferate/reproduce, and a function to determine how new individuals are added to the population via reproduction [4].

In this paper, we model germinal centers using a genetic algorithm to simulate germinal center dynamics. Because there are some gaps in what we known about germinal centers, we made some assumptions on how germinal centers manage to facilitate the evolution of B-cells. Our model successfully managed to produce B-cells with antibody capable of binding with a given antigen with high affinity. Thus, in this paper we propose one possible mechanism for germinal center dynamics.

II. Methods

Genetic Algorithm Structure

Our program is structured using object oriented programming principles. We took this approach as there were several unique elements we wanted to implement independently. We created classes to represent antibodies, epitopes, and germinal centers in our system. Each antibody and epitope has a 24 letter string that represented its DNA sequence as well as an equivalent bit string that stored the same information but in a different format. The DNA string was made 24 base pairs long because epitopes typically contain 6 to 8 amino acids and each amino acid requires 3 base pairs to be represented. The bit string was constructed from the antibody's DNA sequence with each base being represented by a two bit code. For example, adenine (A) was represented with the bit code "00". As a result the bit string representation was twice as long as the DNA representation. Both these representations were necessary in our system since we utilized hamming distance in our fitness function. A bit string was necessary to calculate the hamming distance between the bit string of a candidate antibody and the target epitope. The driver code of our program implicitly represents lymph nodes as it creates multiple instances of the germinal center class, initializes their antibody populations, and begins the evolution process.

Antibody Class

We implemented the antibody class to be able to store the information needed to represent an antibody as well as

the functions necessary for antibody functions within our system. Other than the bit string and DNA sequence stored in each antibody, we also made it so that antibodies stored their generation number, their unique identification number, as well as pointers to their parent antibodies so that we could analyze the evolutionary dynamics in our system.

We also implemented 4 functions relating to antibodies in our system. The first function was a static method used to create new antibodies with randomly generated DNA. This function was used to create naive antibodies for the initial population of antibodies in our system. We also created a static method that created antibodies that were mostly random but contained a certain portion of the target epitope's exact DNA. This function was used to create antibodies that could partially bind to the target epitope for the initial population of antibodies in our system. The third static method we made was responsible for handling antibody reproduction. This function would take to parent antibodies and would randomly select a crossover point. This crossover point would be used to determine what point in the parental DNA would be split and recombined to make the DNA of child antibodies. Each pair of parent antibodies would produce two child antibodies with each child taking the left part of its DNA from one parent and the other half from the other parent. The final function was a mutation function used to model the role of the AID enzyme. Using a set mutation rate, this function was used to change a certain number of base pairs in a given child antibody before it was added to the population of antibodies in the germinal center.

Epitope Class

The epitope class stored much of the same information stored in the antibody class. One major difference between the two is that epitope only had one parent as opposed to two. This is also represented in the reproduction function for the epitopes as it only creates a single mutated clone from a parent epitope. Other than a mutation function function that functions much like the antibody mutation function, the epitope class only contains one static method used to generate an epitope with a random DNA sequence.

Germinal Center Class

The germinal center class contains an array that stores the antibodies that it currently contains as well as an array that stores the epitopes currently contained within the antibody. One addition array, used to store the highest fitness value in each generation of antibodies, is used to store the progress made by the germinal center as the evolution process proceeds.

The germinal center class contains 3 other methods that are essential to modelling the behaviour of germinal centers. The first method is responsible for creating the initial population of antibodies within the germinal center. This method will initialize a population of antibodies where half the created antibodies will be completely naive with the other half being partially binding antibodies. The second method is responsible for calculating the fitness of a given antibody in the germinal center. In our model, fitness is derived by calculating the average hamming distance between the selected antibody and every epitope in the germinal center and subtracting that value from the theoretical maximum fitness of our model (set to 48 as that is the length of the bit strings in our model as such a fitness of 48 indicates an antibody that binds perfectly to every epitope available).

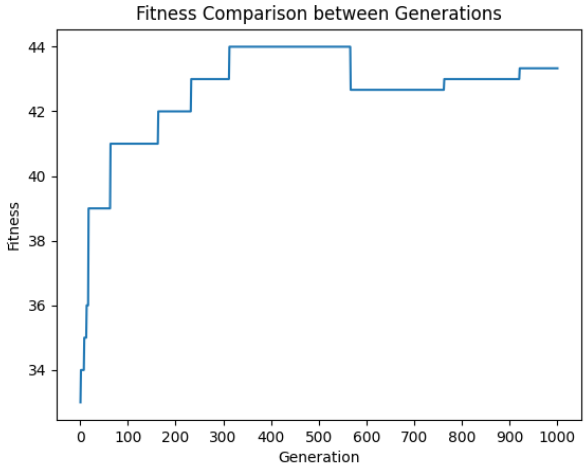

Fig. 1. Line graph showing the fitness of the best performing antibody in each generation.

The most important method in the germinal center class is responsible for selecting the antibodies whose genes should proliferate through the population and reproducing them with each other. This method sorts antibodies in the germinal center's antibody population by fitness and then removes the half of the population that is performing worse than the other half. The remaining half of the population is then made to reproduce in pairs in order to replenish the antibody population size with new candidate antibodies. This method will also occasionally add a new mutated clone of the initial epitope to the germinal center's epitope population. This is done to bias the genetic algorithm towards antibodies that bind well to the initial epitope as well as any possible mutated versions of the initial epitope, thus preventing over fitting to the initial epitope.

III. Results

Fitness

Fig. 1 shows the fitness value of the best performing antibody in each generation in one germinal center over 1000 generations. There are two notable features of Fig. 1 that tell us a lot about the evolutionary dynamics of our germinal centers. The first major feature is the early jump in fitness that occurs from generation 0 to generation 100. We hypothesize that this sharp jump in fitness comes from the algorithm

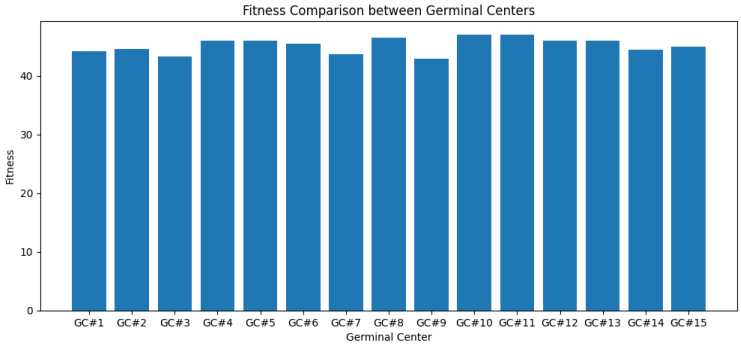

Fig. 2. Bar graph showing the fitness of the best performing antibody in each germinal center at the end of 1000 generations.

quickly selecting antibodies with partially binding regions and reproducing them with each other. This shows that the crossover of ”good” genes is a highly effective evolutionary dynamic. Relying on random mutation alone, which probably is the case for generations 100 to 500, takes a significantly longer time to achieve a similar boost in fitness. The other major feature is the significant drop in fitness seen around generation 600. This massive drop in fitness is probably the result of the addition of a new epitope that the dominant antibody was unable to effectively bind to due to it over fitting itself to the epitopes that were present in the germinal center prior to the addition of this new mutated epitope. Fig. 2 shows how, after 1000 generations, all 15 germinal centers managed to achieve similar fitness levels.

Antibody Lineage

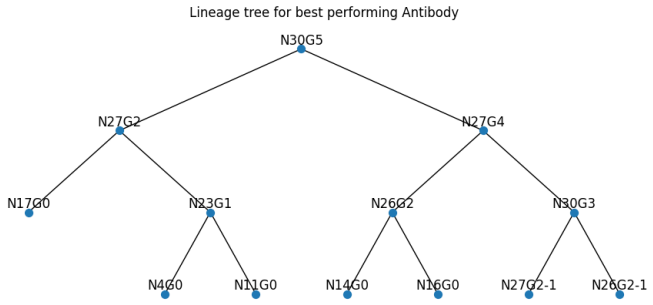

Fig. 3. Genealogy tree showing the lineage of the best performing antibody after 5 generations. Note: nodes are labeled with the following convention - N30G5 refers to node number 30 that was created in generation number 5 and N27G2-1 refers to the second instance of node number 27 that was created in generation number 2.

Fig. 3 took the best performing antibody in a germinal center after 5 generations and shows us the lineage of this node. This lineage tree’s most interesting feature is that the Antibody N27G2 appears in multiple points in this tree. N27G2’s prevalence in this tree indicates that it had genes that were very good at binding to the epitopes present in the germinal center.

IV. Discussion & Conclusion

While we believe that our model was a decent representation of germinal center dynamics, there are many known limitations to our model. The first major limitation is how we calculate fitness in our model. While using hamming distance is convenient, it does not actually give us much information on the actually binding affinity between Antibodies and epitopes. Even while using hamming distance as a toy model, there are issues with representing base pairs as bit strings. Since hamming distance measures the number of bit flips needed to make two bit strings match, the pair of bit strings ”00” and ”11” and the pair of bit strings ”01” and ”10” have a hamming distance of 2. On the other hand, the pair of bit strings ”00” and ”01” and the pair of bit strings ”11” and ”10” have a hamming distance of 1. Since these are the bit strings that we are using to represent our base pairs, we are implying that certain base pairs are closer to each other than to other base pairs. This is not the case in actually binding mechanics in the real world.

If we were to try to re-implement our model to more accurately represent germinal center dynamics, we would have to develop a much more robust fitness function that actually takes into account the binding affinity between an antibody and an epitope given their respective DNA sequences.

V. Reference & Contribution

Abubakar Kasule: Wrote the code for the genetic algorithm and authored this report.

References

[1] Microbe.tv, “Immune 52: B cell boot camp with Gabriel Victora,” Immune — A podcast on the host defense systems that protect against disease, 31-Jan-2022. [Online]. Available: https://www.microbe.tv/immune/immune-52/. [Accessed: 20-Mar-2022].

[2] Laidlaw, B.J., Ellebedy, A.H. The germinal centre B cell response to SARS-CoV-2. Nat Rev Immunol 22, 7–18 (2022). https://doi.org/10.1038/s41577-021-00657-1

[3] Pae J, Jacobsen JT, Victora GD. Imaging the different timescales of germinal center selection. Immunol Rev. 2022 Mar;306(1):234-243. doi: 10.1111/imr.13039. Epub 2021 Nov 25. PMID: 34825386.

[4] P. Kiely, “Introduction to genetic algorithms,” FloydHub Blog, 25-May-2020. [Online]. Available: https://blog.floydhub.com/introduction-to-genetic-algorithms/. [Accessed: 20-Mar-2022].