

**Francine Bianca Oca**  
**CHEM 274B Final Project - Individual Reflection**

**PART 3.1**

My specific role of the project initially was Setup, however after completing the setup portion, changes needed to be made which Kassady fixed, and since she made those changes, I worked on the Compute part.

**PART 3.2**

I contributed a good amount of work to this project (while being in another state as well), by initially creating the repository as well as the initial source/test/output code using our specific allele model just to see if our example was possible to translate into a grid. Although it was possible using our model, we were running into issues with making the library more general purpose, which I helped in by doing the compute services. The whole group did a fair share in coordinating meetings as well as sending text updates to each other about changes in the repository/files.

**PART 3.3**

Most of the challenges that we faced were mostly in regards to the general purpose library setup. We all had a good idea about the specific rules for our allele model and the way we wanted it set up/outputted. However, we struggled to make it based on a general library, of which we all worked together to solve those issues by zoom calls. Another big challenge for us was the time difference Kassady and I had with Korede. So, a lot of coordination and communication in terms of repo updates was critical to make sure we were on the same page despite the time difference.

**PART 3.4**

For the algorithmic and performance analysis of our library, it's important to note that we used a two-dimensional vector of integers to represent our two-dimensional grid. To store the cell states, the states are assigned to integers to allow for simpler simulation transitions. In terms of our boundary conditions, the boundaries are created in a way where a toroidal structure is formed, to allow for a continuous space, which can be computationally expensive, but necessary to demonstrate specific models. In some models, the toroidal structure is beneficial in avoiding issues relating to edge cells. Because we are working with two-dimensional grids, the functions that allows users to set specific rules (onedim\_rule1(), onedim\_rule2(), onedim\_rule3(), twodim\_rule1(), twodim\_rule2(), twodim\_rule3()) iterate over the rows and columns. The CA library allows for code reusability due to the setter and getter functions, which allows for universal usage for any model that the user would like to simulate. The time and space complexity for updating the grid is  $O(N*M)$ , since updating the grid requires the iteration over each row/column, and calculating the neighborhood for that grid. Space complexity might require a bit more overhead due to the temporary storing of temporary vectors, which hold cell states, that are used in some functions. To further optimize the program, parallelization could've been used to make some calculations independent from each other (if possible).

### **PART 3.5**

For problem 2, our model worked as we intended it to work. The change in cell states throughout each generation followed the genetic rules that we put into place depending on starting recessive frequency. Now of course genetic inheritance is a lot more complicated than what is demonstrated by our model, however, it is generally a good basis to search for certain patterns in genetics as a starting point. To improve its accuracy, an awesome (but complicated) idea is to incorporate genetic drift and genetic mutations. For us to do that accurately would require more complicated logic, and a larger grid to accurately represent those complications in genetics. Again, in the simplicity of our model, it is still an effective way to showcase common patterns in genetics.

### **PART 3.6**

I think we were mainly focused on our specific allele model more than the general purpose library. So something I would've done differently is solidify our general purpose library to make the process of this project smoother. Another thing I would've done differently is be more knowledgeable about GitHub, since we were all having many issues with pushing/pulling. If we had the extra time, I would've also implemented a way to parallelize some of the operations in our code, for a better efficient model simulation. Other than that, the group coordination and communication was great and we were all aware of any changes that were made and tasks that we each had to contribute to finish this project.