**Simulating Chromatin with Variable Linker Length Design Document**

**Overview:** This project is meant to augment the existing chromatin simulator code from the Spakowitz lab. The goal of the code is to understand 3D structure of chromatin with modifications of a variety of physical parameters including linker length (the distance between histones), linker angle (the angle between different histone connections), and epigenetic marks. For this project I focused on incorporating the ability to add linker length to the Polymer object used to represent chromatin.

**Background:** Understanding the behavior of chromatin from a physics-based perspective will help us better understand and predict the effects of different modifications to the chromatin configuration or environment.

**Goals:** Add the ability to variable linker length to the existing Chromo codebase. Do this by modifying the process by which the Polymer object is assembled to allow for the condition of variable linker length. Change the linker length variable from a float with a single value for each histone-histone connection to a np.ndarray of linker lengths in which each individual linker length can be set to a different value. Additionally, create a test that allows the user to confirm that the image of the chromatin object they have generated resembles the control to verify that the code is functioning correctly.

**Non-goals:** For the scope of this project, I decided not to focus on running the simulation itself with the Polymer object, since this would involve modifying several additional physical parameters and a greater restructuring of the code.

**Future Goals:** Next, I plan to add the ability to run the simulation itself with variable linker lengths, and explore which other physical parameters must be modified. I also plan to make the angles between the interconnected histones modifiable to mirror the biological customizability of linker angle.

**Detailed Design:** The overarching design of the simulator has been previously documented through sphinx files. They can be accessed by going to the /chromo/docs/build/html folder and opening the relevant html files in a web browser. This project included the addition of a run\_simulation.py file that runs the program and calls the test\_polymer\_build.py function which compares the Polymer object generated to a reference control. The user can then visually inspect any potential differences between the Polymer object generated and the control and respond “yes” or “no” in response to the prompt about the degree of similarity.

**Python environment:** There is a *make\_all.sh* file that will conveniently install all the required packages and dependencies needed for the program to function.