4.1 Plan Introduction

This Software Development Plan provides the details of the planned development for the Gene Pages feature for the GRNsight project. The gene pages will add additional functionality to GRNsight allowing for a user to select a protein and have the program automatically query multiple APIs for information about the correct protein from the species.

4.1.1 Project Deliverables

4.1.1.1 Species Menu 3/16

This is a menu that will display the species if the program is able to detect it from the user uploaded workbook. Otherwise the user can interact with the menu and manually select the species that they are working.

4.1.1.2 Tests for Species Identification 3/30

The tests will cover and prove the functionality of all of the pieces of the species menu

4.1.1.3 API Calls 4/13

Taking the species and protein data that is passed to the program Through GRNsight this functionality will query the relevant APIs for the species. Giving each the required data for a response. It will then handle the data and output it for the user.

4.1.1.4 API Tests 4/20

Thorough testing to prove the functionality of the API calls in the gene page.

4.1.1.5 Data Display 4/27

Display the data in a neat and easily readable form for the user. This is not imperative to use, but it is a feature that adds to ease of use.

4.1.1.6 Finish Tests 4/27

Finish tests for data display, and add or update tests as the code evolves.

4.2 Project Resources

GRNsight as a whole is a multi-year, multi-discipline, team project, however the GRNstate section that I am working on is an individual project. GRNsight is hosted through Github and the current live and beta branches can be accessed online. The GRNsight team also holds weekly meetings on Monday where we go over last week's progress, and talk about issues and goals for the current week.

4.2.1 Hardware Resources

A short list with descriptions of what hardware components are needed to work on GRNsight

4.2.1.1 Computer: these will be the minimum requirements for development, most of which will be met by all modern computers.

4.2.1.1.1 2 core CPU running at 1.6 GHz

4.2.1.1.2 2 Gb of ram

4.2.1.2.3 1 Gb of space on a HDD

4.2.2 Software Resources

A short list with descriptions of what software is needed to work on GRNsight

4.2.2.1 IDE

I use VScode, but any text editor will work.

4.2.2.2 NPM

A lot of the app is based around NPM packages. You will need to install before you can run the code.

4.2.2.3 Browser

The browser will allow you to access the localhost where the website is run while developing.

4.2.2.4 Github

Github is where the code is stored and hosted. To access the code you will need an account to gain access as a developer.

4.3 Project Organization

The deliverables each represent a milestone of development. Given the nature of the GRNsight group, and our weekly meetings where we have a discussion on what was accomplished and what is left to be done, it makes it very easy to have a goal of a deliverable, or a piece of the current deliverable ready each Monday. Not only do we need to have a piece of the project, but we are also expected to have written up the accomplishments of last week and the goals of this. This makes the meetings flow more smoothly and lets us focus on issues. Listed are the general deliverables which make up the pieces of the GRNsight project.

4.3.1 Reading species/taxon data and updating GRNstate.

4.3.2 User interface for reading in and updating species.

4.3.3 Querying APIs for different species

4.3.4 Updating the gene page user interface

4.3.5 Testing

4.4 Project Schedule

Weekly meetings on Monday with a goal of a working piece of the final deliverable each meeting. This section will provide more information for the schedule of the Gene Pages for GRNsight.

4.4.1 PERT / GANTT Chart (attached)

4.4.1.1 Pull request and front end branch merge to beta. After working on all the front end components this is an iterative step where merge conflicts are resolved and the client can give feedback on design.

4.4.1.2 Passing data to the gene page. This is the first step to formatting the API calls. The data parsed in the front end needs to be passed to the backend so API calls can be made.

4.4.1.3 API calls. This is the most important part of the project. We take the data formatted by the front end and use it to call online APIs and format and display the data that is returned to the user.

4.4.1.4 Testing. Testing should optimally be completed on all the code that is written, but it is imperative to have the API calls tested.

4.4.1.5 Gene page data display. Format the data returned by the APIs. This step is not critical, but still a goal.

4.4.2 Task / Resource Table

Each development requires only the software and hardware components listed, while the weekly meetings require an area to host.