2/15/2017

Software Design Documentation

Team U.I. Fit – Version 2.0

Team Members:
Charles Chatwin
Matthew Burns
Tanner Brelje
Joshua Gutman

Sponsor: Dr. Abolfazl Razi
Mentor: Dr. Abolfazl Razi

The purpose of this document is to provide a timeline and implementation plan for the BioNetFit software solution.

TABLE OF CONTENTS

INTRODUCTION

The field of molecular biology is an ever advancing science that requires tedious experimentation and research. In order to generate concrete research results, many molecular biologists must place precious time and resources into large-scale experiments. These experiments are used to show the process of biochemical molecular interaction, or in layman's terms, the result of the reaction between two or more molecules. In order to aid this meticulous process, Northern Arizona University graduate Brandon Thomas, in tandem with an experienced team of graduate students and molecular biologists, created the program BioNetFit. BioNetFit is a command line tool that was created to provide a fast and easy method to simulate complex molecular bonds. These simulations allow researchers to later run the tests in a real environment in a way that gives them a degree of confidence in the expected interaction.

Built on top of BioNetGen and NFSim, BioNetFit allows researchers to simulate a single experiment many times with a range of parameters. The results of each of these simulations is compared to an experimental results file that the researchers upload. Because the combinations of different parameters scale non-linearly, various methods are used to select the best combinations of parameters, including genetic algorithms, simulated annealing, and a few others. After the simulation has been run many times (usually thousands), BioNetFit creates a final output file that shows information about the molecules involved in the reaction.

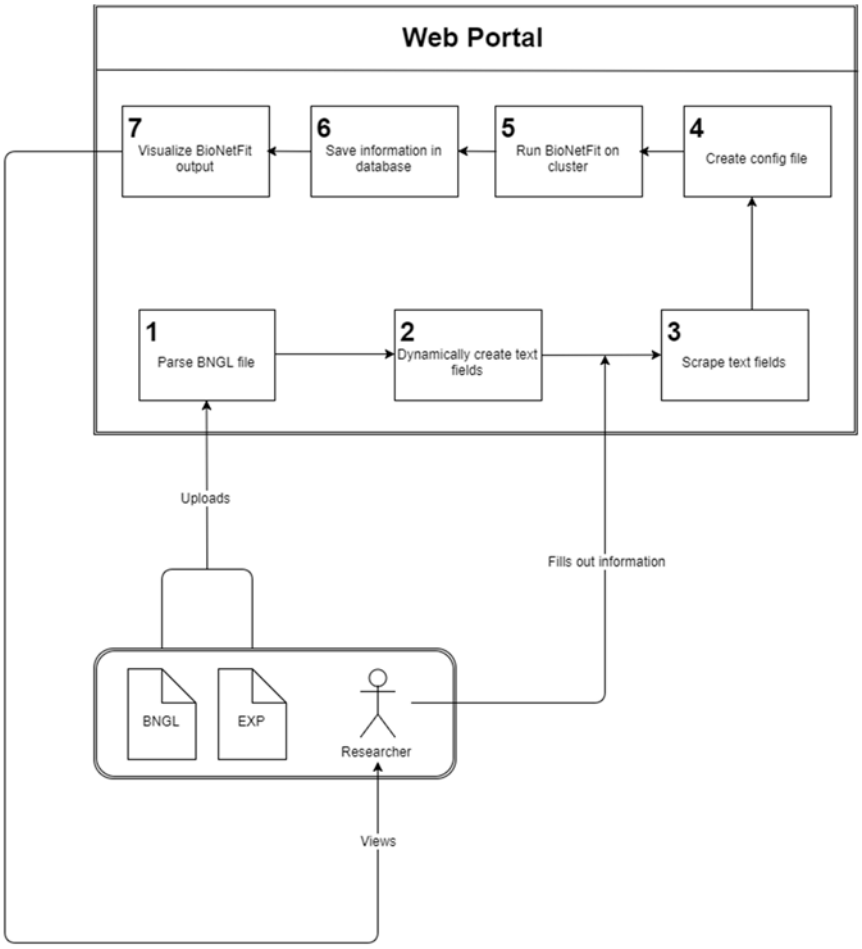
While the program is incredibly useful, its implementation is not user friendly. It exists as a unix-only, command line tool. Researchers who are not technically proficient will struggle at getting the program to run and will attempt to find other avenues when faced with having to spend time learning a new system to run their tests. Additionally, the results currently output by the program are difficult to digest, and do not lend themselves easily to analysis for data collection by researchers. Without much labeling or clear direction in the results, researchers will have to extract results from a command line output, which as previously discussed, is problematic for many scientists. Lastly, the program cannot run large scale experiments in a short amount of time. This lessens the practicality of the program as the experiment becomes more and more grand in scale. In order to make BioNetFit the stellar application it can be, these issues need to be addressed. If done correctly, the program could see widespread use in molecular biology labs all over the world.

Created in order to tackle these challenges, Team U.I. Fit, composed of Charles Chatwin, Matthew Burns, Tanner Brelje and Josh Gutman, will develop software solutions in order to turn BioNetFit into the powerful program it can be. Led by client and mentor Dr. Abolfazl Razi, our team will aim to:

- Create an attractive and simple web portal for BioNetFit.
- Visualize the results of BioNetFit into easily digestible data for researchers.
- Implement parallelization in order to run large experiments on a computer cluster.

Firstly, our team will develop a simple and effective Web 2.0 Graphical User Interface that will house the BioNetFit software. This web page will be easily accessible and easily usable by any researcher who wishes to run tests using the BioNetFit software. In the web portal, the user can either create their own BNGL file from

scratch, or upload one that was either previously downloaded, or was saved remotely on the website itself. From here, the user can run the file either locally on their own system or, in the case of a large file, run the file on the computing cluster Monsoon, available on the Northern Arizona University campus. From there, the program will output a configuration file that will be visualized to the user using graphs and charts. Finally, the user can either save and visualize the outputs on the website, or download them to their own local machine. The BNGL files can then be tweaked if wanted, and the experiment can be run again.



In this document, our team will outline the overall design and architecture of our BioNetFit software solution, as well as the specifics of each of the individual modules that will factor into the overall structure of the project. This will include, but is not limited to, full analysis and coverage of the Web 2.0 BioNetWeb GUI, visualization production via Python graphical interfaces and modules, database design and integration, and utilization of NAU Monsoon computing cluster. For each topic, we will delve deeply into the projected implementation of each technical aspect, as well as explain how each solution will meld with each other and form a cohesive unit. Additionally, we will discuss the timeline and plan for implementation within the coming months, as well as any plans and fail safes in the event that issues arise during the implementation. The resulting document will fully outline the project in detail, and will serve as a set of guidelines during the final months of the BioNetFit project.

IMPLEMENTATION OVERVIEW

In order to adequately meet the needs of the client, our proposed software solution will increase both the ease of access as well as the time it takes in order to process a molecular combination. To address these ambitions, we will implement a Web 2.0 GUI to host the BioNetFit program, making it widely accessible to scientists and scholars online. The GUI will also provide visualization to the results of the BioNetFit process, presenting graphs, charts, and easy to digest information to the user. After a full realization of the software, the program will be implemented onto the NAU Monsoon cluster, in order to increase the processing speed of the application. Lastly, information will be needed to be stored for individual users, in order to keep track of who is running what, and knowing where to send information back when results are acquired.

In order to make our vision come to life, the majority of the program will be implemented in the Python programming language. Python is an ever advancing language that is set to become a main force in today's world. By choosing to use Python, we can assure that our implementation will not only be innovative for the present, but adaptable for the future. Additionally, using Python allows us to tap in to a wide variety of 3rd party expansions and packages that work in tandem with the program. Some of these key features that will see use in our implementation are as follows:

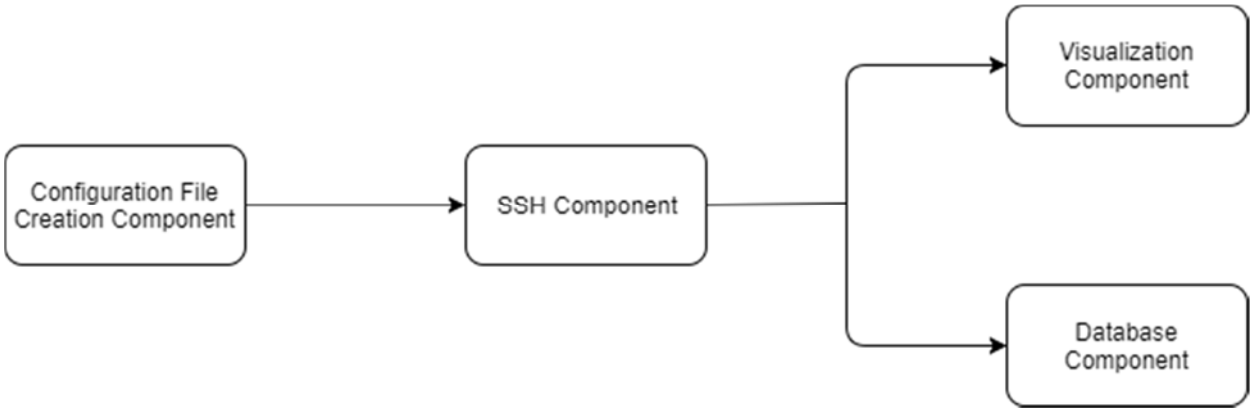
- Django Web Development
- MatPlotLib
- MongoDB w/ Pymongo
- Etc.

Our program will make use of the Django web programming, to bring our vision to life. This will also make use of programming and accessing the Cluster to create an environment that would be accessible to run the BioNetFit from people on the internet, and then allow it to send feedback from the results to whom accessed the BioNetFit program for the particular run of it.

Python and Matplotlib, a subprogram of Python, will be used to control the environment and run the commands for the Web2.0. Matplotlib will be used extensively to generate visualizations to improve the interpretation of the results from BioNetFit as well. Python will be used as the backbones of the Web2.0 Portal in order to keep everything together

MongoDB will be used as a means of holding information on the web for users. This information includes everything essential to separate user information and program runs of BioNetFit for each individual user.

ARCHITECTURAL OVERVIEW



At a high-level, the architecture consists of four basic components:

1. Configuration File Creation
2. SSH
3. Visualization
4. Database

The Configuration File Creation component allows the user to upload a BNGL file to the web portal. This file will be parsed and a custom configuration file template will be generated to make it easy for the user to create a configuration file.

The SSH component allows the user to do several things. The first is to run BioNetFit on NAU's high performance computing cluster *Monsoon*. After the user has uploaded a BNGL file, an EXP file, and has created a configuration file, they can click a button that will send all of those files to Monsoon. It will then start a SLURM job that will run BioNetFit on the Monsoon using the files the user uploaded/created as input. The SSH component is also responsible for retrieving the output files from BioNetFit off of Monsoon and back on to the web portal. This enables the web portal to create visualizations of the output as well as save the output in a database for future review.

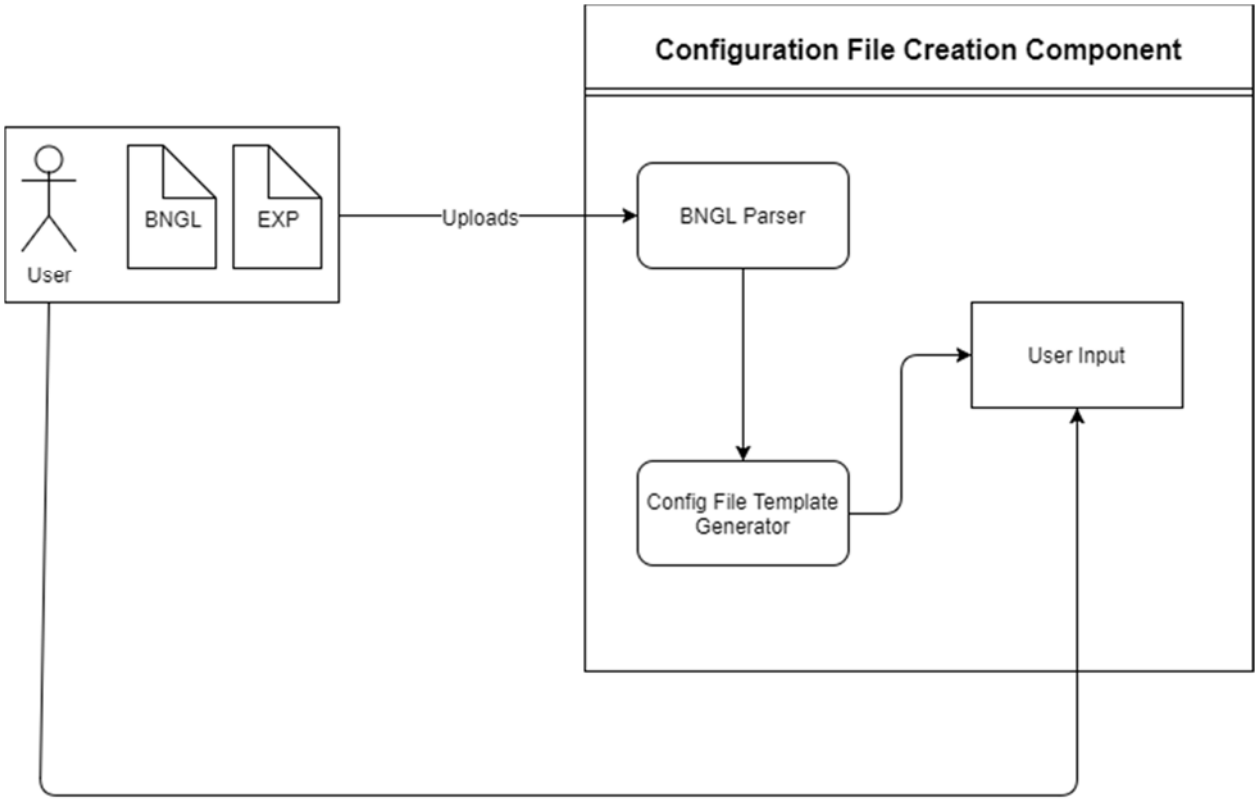
The Visualization component aims to give the user feedback about how well BioNetFit is performing as well as giving the user a quick digest of the information that is present in the results. By implementing these two concepts, the user can both make conclusions about the simulation and improve their use of BioNetFit in the future.

The Database component automatically saves all input and output files from a BioNetFit run and associates them with a user. As long as the user has an account and is logged in, the information will be saved. They can pull up this information (including visualizations) to review the results or make modifications to their input files and rerun BioNetFit.

MODULE AND INTERFACE

CONFIGURATION FILE CREATION

The configuration file creation component allows the user to quickly and easily create a configuration file, which is necessary to run BioNetFit. After creating a configuration file, they may run BioNetFit on NAU's *Monsoon*.



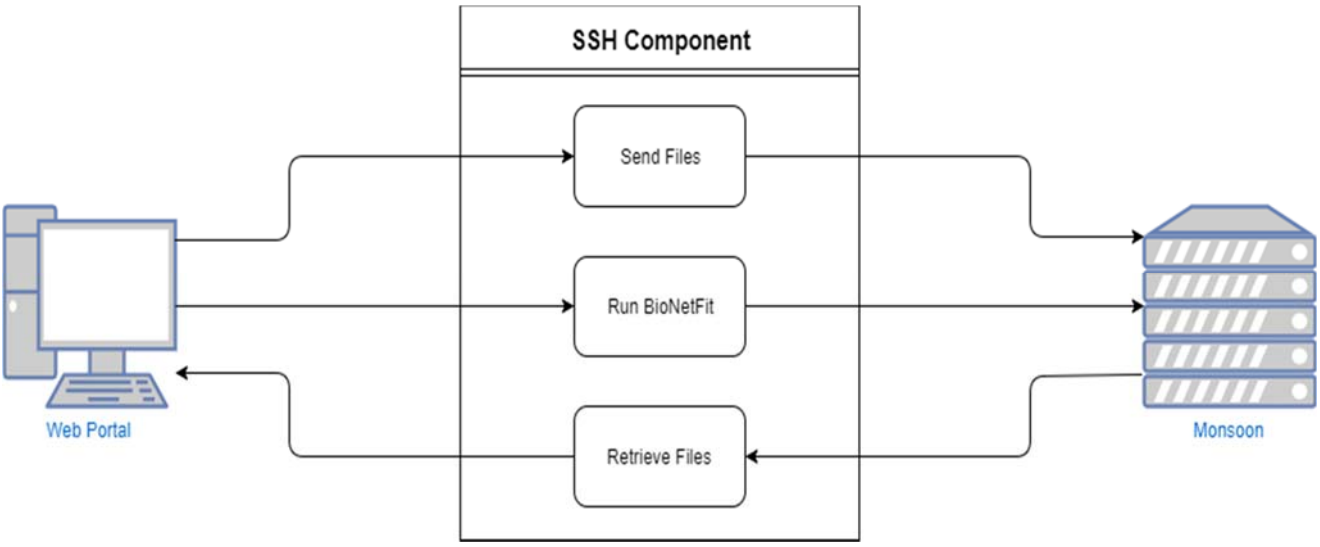
The Configuration File Creation component consists of four steps:

1. User uploads BNGL file (and optionally an EXP file)
2. The BNGL file is parsed looking for free parameters
3. A configuration file template is generated
4. The user fills out the template to create a configuration file

The user can then download the configuration file locally, or run it on NAU's *Monsoon* if certain conditions are met (which is handled by the SSH module). The configuration file template will have all available BioNetFit options as per the official documentation. There will also be other ease-of-use features like descriptions of each option and a live preview of the final configuration file.

SSH

The SSH component is responsible for communicating between the web portal and NAU's high performance computing cluster *Monsoon*.

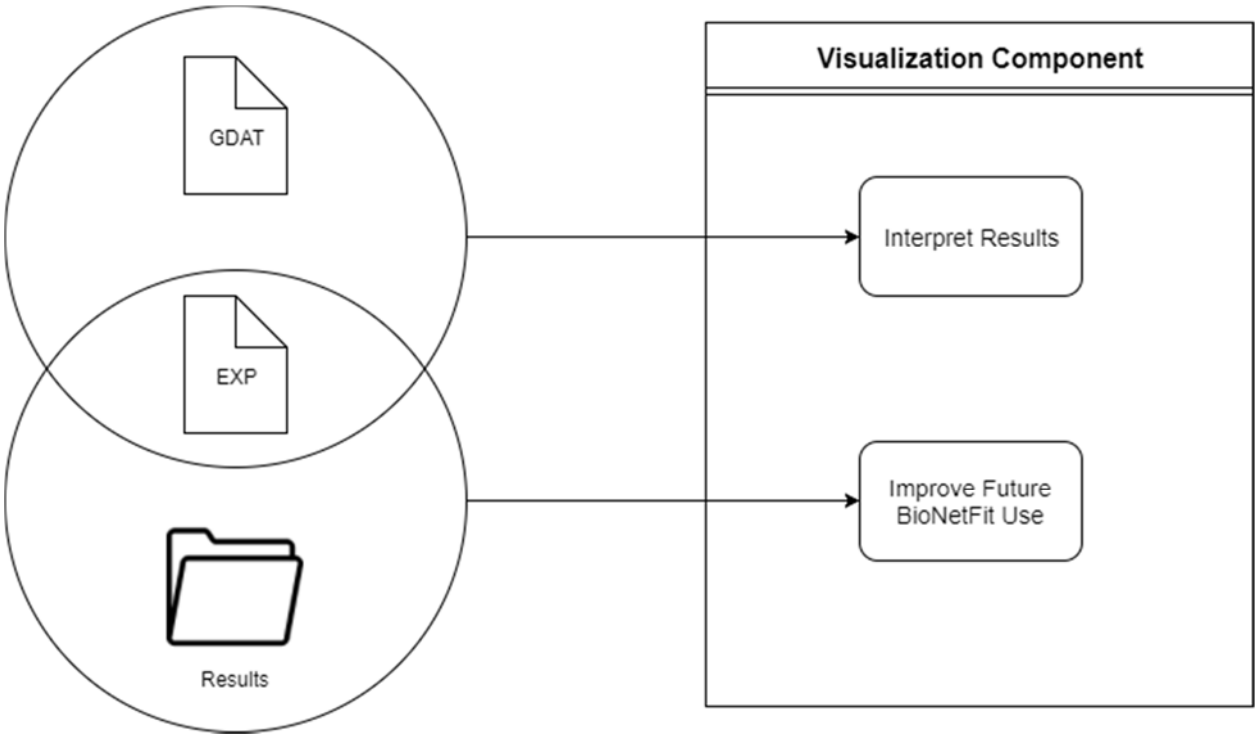


Rather than having sequential steps like the other components, the SSH component acts as an interface to interact with Monsoon. It handles sending files from the web portal to Monsoon, running BioNetFit on Monsoon, and retrieving files from Monsoon to the web portal.

There must be a protocol in place to run BioNetFit on Monsoon that deals with the locations of the input and output files, and how to associate those with users. This protocol must be robust and well thought-out, as changing it would require tediously moving and/or renaming files and directories manually.

VISUALIZATION

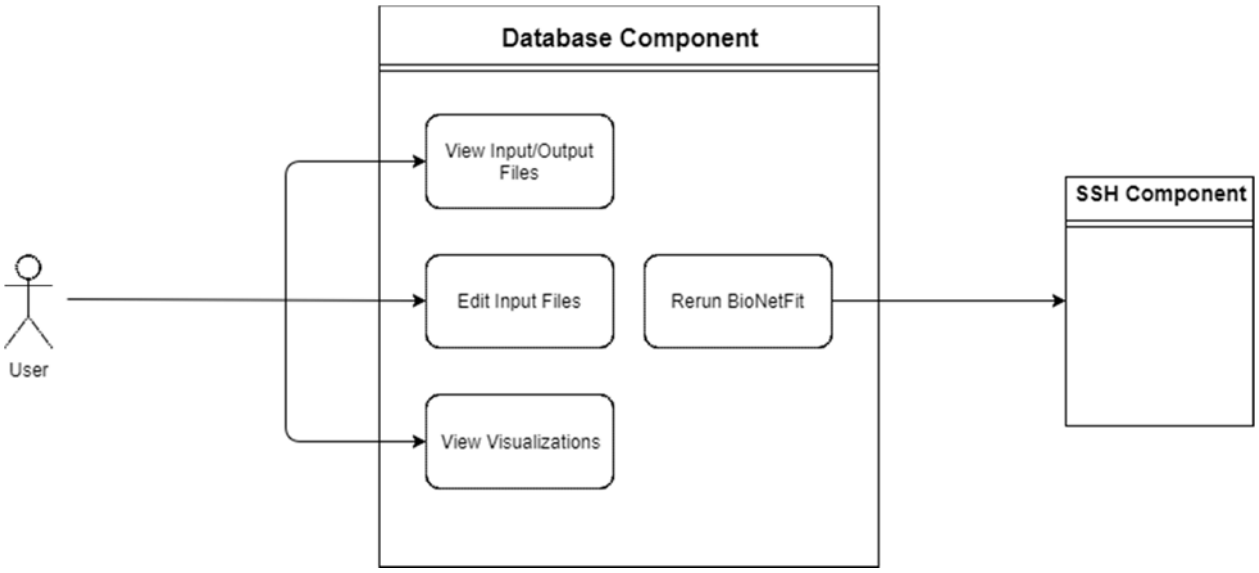
The Visualization component helps the user to interpret the results of their BioNetFit runs and improve their use of BioNetFit in the future.



In general, visualizations that help interpret results will use individual GDAT files (which show the values of observables over time in a generation) and EXP files. Visualizations that help the user improve their usage of BioNetFit will use the entirety of the results folder, looking at GDAT files from every simulation in every generation. Some visualizations may be useful for both interpreting the results and improving BioNetFit usage.

DATABASE

The Database component is responsible for saving the files associated with a BioNetFit run for a specific user.



The Database component will allow the user to view input files (BNGL, EXP, configuration file), output files (best-fit GDAT files and generational GDAT files), and visualizations. It will also allow the user to quickly edit any of the input files and rerun BioNetFit on Monsoon using these new files.

Example of the data stored: (Note – Files will be attached based on this user information.)

	F_name	L_name	pass	email
User 1	Joe	Brown	Catlover1	Sajv@man.com
User 2	Reggie	Can	Mmelll	werst@cox.net
User 3	Sandy	Hotts	Poridge1001	sand@hotmail.co

IN CASE OF FAILURE:

In order to prevent a crash and failure of the database system from destroying the saved information, the database will be backed up weekly to either a file system resembling that of an encrypted excel spreadsheet, or to a local implementation of the database on one of the admin's home computers.

PROJECT MANAGEMENT

In order to adequately manage the project and ensure that the trajectory of the project, the team will be hosting weekly meetings with Dr. Razi. In these meetings, we will be demonstrating key aspects of the program as well as providing weekly documentation that updates our mentor about the current stasis of the project.

Currently, demonstrations done within the meetings have included:

- Database creation and management
- Basic website implementation
- SSH examples and connectivity
- BioNetFit bug fixes

In the future, other demonstrations will include:

- Databasing linked with website
- Full website implementation
- Multiple types of visualizations
- SSH Security demonstrations

The weekly documentation provided to Dr. Razi will include the following:

- Attendance of team members at weekly meetings
- Updates to the progress of the implementation
- Items that were accomplished within the week
- Items to be accomplished within the next week.
- Future items on the team's radar.

With these items, the team expects to keep themselves on track given what should be a difficult school semester. These items also keep the mentor, Dr. Razi, informed and in the production cycle, ensuring that developments within the program are to his liking.

IMPLEMENTATION PLAN

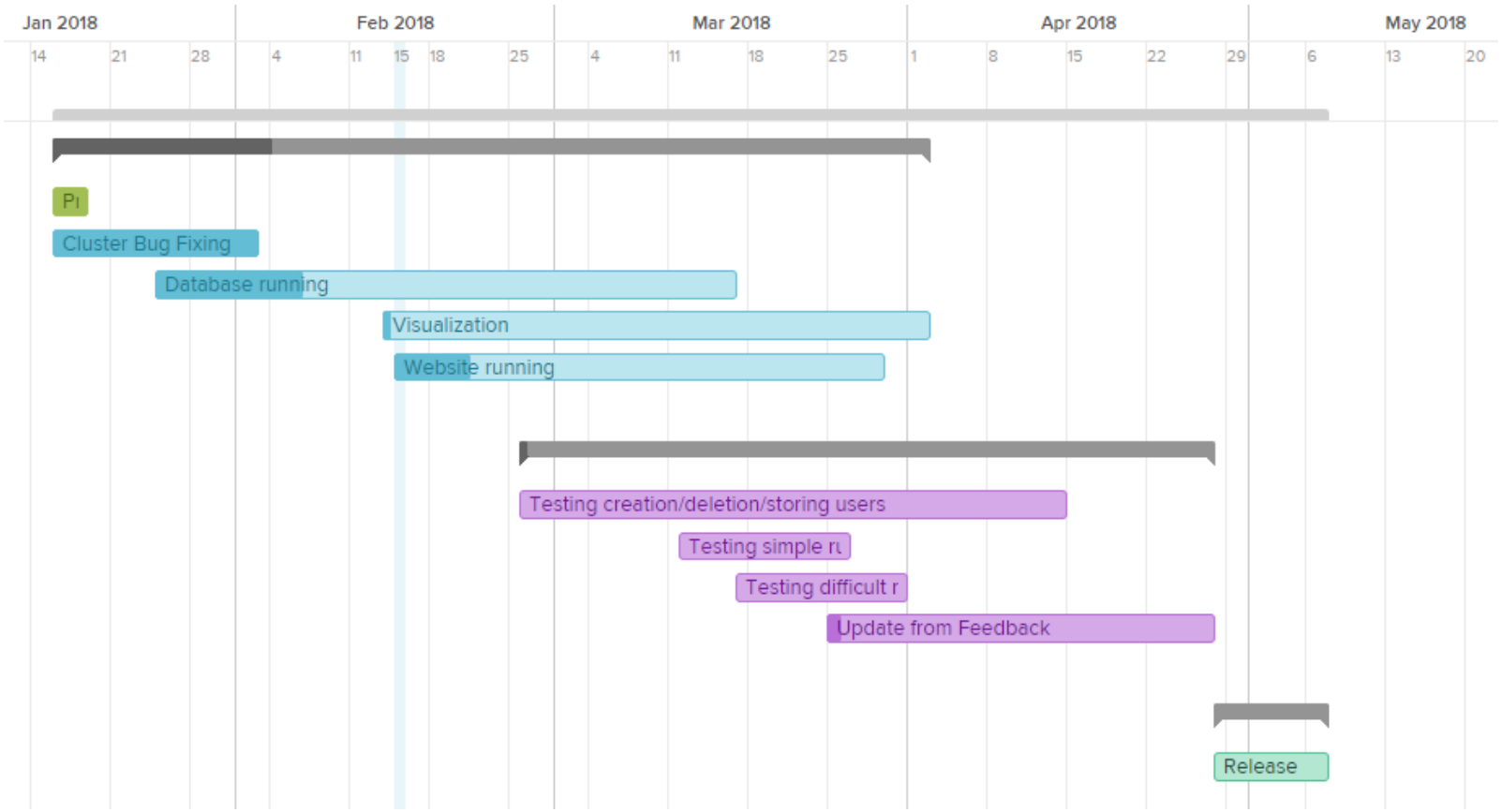
In this section, we will delve into the timeline of events that will contribute to the full construction of the BioNetFit software solution. In order to increase the digestibility of our implementation plan, we will divide the process in to multiple different phases. The phases of implementation are as follows (note: some phases may have reached completion before document publication; phases subject to change):

- Phase 1: General bug fixing of the BioNetFit software.
    - At the time of writing this, the team is currently in the process of retrieving and fixing a working local version of the BioNetFit software. This local software, once tested and proved to be working, will serve as a base for local implementations of the BioNetFit software solution. Additionally, the working version will be sent to the cluster to serve as a similar base for implementation.
- Phase 2: Individual Testing
    - During this phase, each member of the group will be working individually, in order to fully grasp the technology assigned to them within the Work Distribution Memo. This will include the production of demonstration applications and video tutorials on the specific subject.
- Phase 3: Local Implementation
    - At this time, the group will commence assembling the puzzle that is the BioNetFit software solution. Each member of the group will bring their specified expertise together in order to generate a local prototype of the software solution. This prototype will attempt to be a full and complete realization of the software that will be easily transferred to the cluster.
- Phase 4: Cluster Implementation
    - Here, the group will implement their original local design onto the computing cluster. This process should not require much time in the realm of programming, but may require time when considering the addition of SSH and cluster computing.
- Phase 5: Bugfixing
    - Once the program is installed to the computing cluster, the program will undergo rigorous testing in order to ensure that the process runs fluidly and smoothly every time.

- Phase 6: Official Release
    - At this point, the program as described by both the project description and feasibility report should be fully implemented. It is here that the group will begin their final reports on the project, as well as prepare for any presentations that will be done.
- Phase 7: Additional Contributions
    - If there is additional time at the conclusion of the project, minor additions/academic assistance for Dr. Razi may be done at this time. This is an optional process.

GANTT CHART

The following Gantt chart displays the likely plan for the production of the actual software, including release versions and specific implementations.



CONCLUSION

In conclusion, we have discussed our plan for the implementation of the BioNetFit software solution, including the specifics of software design, and the likely timeline for the completion of the project as a whole. We have firstly reintroduced the project as a whole, this time with a concrete idea of the goals and ambitions that the software will aim to achieve. We then have discussed the likely plans for implementation, covering what we imagine the final product of the software solution to both achieve and visualize. We have also covered the main four facets of the program as a whole, these being configuration file generation, SSH, visualization and databasing. Lastly, we have laid out the ground work for what we believe to be a solid timeline and plan for creating the software. This timeline is made out of seven phases, each of which will mark a significant addition to the project upon its completion. Overall, we hope that this document has made clear what our implementation plan is for the software, as well as serve as a guiding point for our future endeavors on the project.