Software Requirements Specifications

for

JAX Front-end Project

Version 1.6 approved

Authored by Sarah Crane, Harshil Patel, Prince Safo, Kate Stenberg

Prepared & Edited by Sarah Crane

CS5500 – Foundations for Software Engineering, Professor Gary Cantrell

Northeastern University, Roux Institute

March 12, 2025

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Sarah Crane	2-11-25	Initial Draft	1.0
Harshil Patel, Prince Safo, Kate Stenberg, Sarah Crane	2-16-25	Preliminary Draft	1.1
Sarah Crane	2-26-25	Revised Outline	1.2
Harshil Patel, Prince Safo, Kate Stenberg, Sarah Crane	3-2-25	Draft 2	1.3
Harshil Patel, Prince Safo, Kate Stenberg, Sarah Crane	3-12-25	Edits	1.4
Harshil Patel, Prince Safo, Kate Stenberg, Sarah Crane	3-13-25	Tech Review Edits	1.5
Harshil Patel, Prince Safo, Kate Stenberg, Sarah Crane, Salim Salim	3-17-25	Final edits	1.6

Team Signatures

Name	**Date**
Prince Safo	03/01/2025
Harshil Patel	03/01/2025
Sarah Crane	03/01/2025
Kate Stenberg	03/12/2025

1. Introduction

1.1 Purpose

This software requirements specification document (SRS hereafter) will lay out the requirements of the Jackson Laboratories Front-End project for Northeastern's Spring 2025 Portland campus CS5500: Software Engineering course. It will detail the goals for the project that the team intends to accomplish, including both functional and non-functional requirements, as determined by the client and the team members.

1.2 Project Scope

This project will be focused on implementing two of the GeneWeaver analysis tools in the web application. In a similar vein to the implementation of the GeneSet graphs, which is intended to show how closely linked genesets are, this project will focus on the front-end UI/UX implementation of two analysis tools that are currently documented but missing from the web application. Those tools are: similar variant set and find variant. Specifically, this project will focus on the user interface for these tools, as the logic and algorithms have already been designed and coded. We will alter the GeneWeaver web app to include these two tools in the Analysis Tools section, mirror the existing UI/UX, and allow users to access them.

1.3 Conventions / Definitions, Acronyms, and Abbreviations

Gene set: a list of genomic features, free text descriptive content, ontology annotations, and gene association scores [1]
Jax: Jackson Laboratories, the client on this project
MoSCoW Method: M - *Must have*, S - *Should have*, C - *Could have*, W - *Won't have*. [2]
SRS: Software Requirements Specification [3]
API: application programming interface
TBD: to be determined

1.4 Client and Stakeholder

The main stakeholders for this project include the client, John Bluis of the Jackson Laboratory, and the 'Front-End' team of graduate students conducting this project, namely Harshil Patel, Prince Safo, Kate Stenberg, and Sarah Crane. Additionally, Professor Gary Cantrell, the Roux Institute's CS5500 course instructor for the Spring of 2025 acts as the final primary stakeholder for this project. The team acknowledges that there may be other users in the future who might benefit from the work of this project (see section 1.5 below).

1.5 User Characteristics & Intended Audience

GeneWeaver is meant for use most particularly by:

- employees of Jackson Laboratories
- students
- researchers
- educators
- any others with guest or user access to the GeneWeaver website
- any others able to download and navigate the open-source code for the project

More generally, users of this program are scientists. These users may have coding knowledge (they may be able to run GeneWeaver locally) or they may not (in which case they can use the web application). They have statistical knowledge that enables them to run the analysis tools. The web app is not for the general public; users must have either approved guest access or verified accounts to use it.

2. Overall Description

2.1 Product Description

GeneWeaver [4] is an application that allows for gene comparison and analysis. It comprises a database of gene sets and a set of analysis tools. GeneWeaver is an open-source project; all the code is available on GitHub for users to run locally. There is also a web application, which is easier and quicker to use. The broad purpose of the software is to allow users to search for gene sets and compare them with other gene sets based on specified criteria. Users are able to search for or add gene sets, add them to their projects, run analysis tools on them, and share them with other users.

This team will enhance the web app by adding interfaces for two tools that are currently only usable locally: Find Variants and Similar Variant Set. On the web app, there is a page specifically for Analysis tools, which has interfaces for the rest of the analysis tools observed in the GitHub repository. However, Find Variants and Similar Variant Set are absent from this interface. Our team will create and add interfaces to this page that will be able to interact with the gene set database according to the logic of these analysis tools as laid out in the documentation.

2.2 Product Perspective

Two existing analysis tools, similar variant set and find variant, are currently available within the general GeneWeaver application, but missing from the web application. By adding these two tools to the web application researchers, students, and general users of the GeneWeaver web app will have an increased library of tools with which users can access and analyze gene sets within their projects.

2.3 Constraints and Operating Environment

The GeneWeaver API and web application is based upon an open source, Apache License 2.0, and is available via GitHub repositories. This project will work within the API standards laid forth by GeneWeaver and the Jackson Laboratory to help improve the quality and reproducibility of the scientific environment fostered by the stakeholders.

2.4 User Documentation Requirements

Some user documentation for this project is already available. The Find Variants tool has instructions in both the GitHub documentation section and the general GeneWeaver documentation page, while the Similar Variant Set tool only has documentation on GitHub and lacks instructions in the general GeneWeaver documentation. To improve user support, we will write clear and easy-to-follow instructions for both tools, ensuring they are accessible even without specialized genomics knowledge.

Additionally, we will provide basic test cases to help users understand how the tools function. To enhance the user experience, we will create a Figma prototype to visualize the interface before development and conduct usability testing to gather feedback for improvements and provide a report if we get free access to some of the open-source tools.

Furthermore, we will explore adding visual aids, such as step-by-step screenshots or short videos, to make learning more interactive and engaging. These efforts aim to provide a seamless and user-friendly experience for all users.

3. Specific Requirements

3.1 Functional Requirement: Similar Variant Set

3.1.1 Description and Priority

The Similar Variant Set feature allows users to identify and retrieve genetic variants that are similar to a given variant based on predefined criteria such as sequence similarity, functional impact, or disease association. This functionality enhances the analysis and interpretation of genetic data, aiding researchers in identifying meaningful genetic relationships.

3.2.1 Stimulus and Response

Stimulus: A user selects a genetic variant and requests a list of similar variants.
Response: The system retrieves and displays a ranked list of similar variants based on similarity metrics, including but not limited to sequence similarity, functional annotations, and clinical significance.

3.1.3 Functional Requirements

The system should allow for the following processes (priority indicated in parentheses):

- Users should input a variant ID or select a variant from existing datasets to initiate the search for similar variants (M)
- Users can refine their search by applying various similarity criteria, such as sequence similarity, phenotype relevance, and disease association, ensuring that the retrieved variants align with their specific research needs (M)
- Once a query is submitted, the system should efficiently search the database and rank similar variants based on predefined metrics. To maintain efficiency, the system should retrieve and display results within a reasonable time frame, ideally under five seconds for most queries. (S)
- The application should log all failed login attempts and trigger an alert after consecutive failed attempts. (C)
- The system should consistently provide accurate and reproducible gene set relationships, ensuring correctness in graph generation and analysis. It must handle errors gracefully without data corruption or unexpected failures (M)
- The user interface should be intuitive, allowing researchers to easily navigate, interpret, and manipulate gene set graphs with minimal training. Clear visualization and well-labeled elements should enhance the user experience. (S)

3.2 Functional Requirement: Find Variant

3.2.1 Description and Priority

The Find Variants tool is designed to help a user find gene sets analogous to a target gene set in other species. The tool starts with a given gene set (or multiple gene sets) and returns the corresponding gene set(s) in other species. Currently, the logic is designed to work between humans and mice or between mice and humans. The logic of the tool already exists, but the web app does not have a

current way to interface with the tool. The goal of this team will be to create this interface so that the user can access the analysis tool from the web app.
This is a high-priority task (M on the MoSCoW scale).

3.2.1 Stimulus and Response

Stimulus: The user clicks on Analysis Tools.
Response: A menu of analysis tools appears on the left-hand side, including Find Variants.

Stimulus: The user clicks on Find Variants.
Response: A dropdown window opens. The system will prompt the user to select genesets from a prepared list. The system will prompt the user to choose either “Human to Mouse” or “Mouse to Human”. The system will also prompt the user to select either “eQTL” or “transcript” (or both) as the relationship type.

Stimulus: The user clicks Run
Response: The system will query the geneset database based on the attributes selected and return variants related to the given gene based on the selected relationship. Variants found through the eQTL relationship will also include the tissue name. Variants found through the transcript relationship will also include transcript IDs.

3.2.3 Functional Requirements

The system should allow for the following process.

- When a user selects “Mouse to Human”: the system should search the database for the given geneset under Mouse geneset and display the matching result from Human geneset (M)
- When a user selects “Human to Mouse”, the system should search the database for the given geneset under Human genesets and display matching results from Mouse genesets. (M)
- When a user selects an eQTL relationship, the system should query the database for gene sets with similar eQTL attributes and display the results.
- Transcript Relationship Search (M)
- When a user selects a transcript relationship, the system should query the database for gene sets with similar transcript attributes. (M)
- The application should log all failed login attempts and trigger an alert after consecutive failed attempts. (C)
- The system should consistently provide accurate and reproducible gene set relationships, ensuring correctness in graph generation and analysis. It must handle errors gracefully without data corruption or unexpected failures (M)
- The user interface should be intuitive, allowing researchers to easily navigate, interpret, and manipulate gene set graphs with minimal training. Clear visualization and well-labeled elements should enhance the user experience. (S)

4: NonFunctional Requirements

4.1 Performance

- The system should respond to user actions within 5 seconds under normal load conditions.
- The application should support at least 100 concurrent users without significant performance degradation.
- The system should support an increasing number of gene sets and complex graph structures without significant degradation in performance. It should be able to accommodate large-scale datasets used in genomics research.

4.2 Availability

- The system should have 99.9% uptime, ensuring high availability.
- In the event of a system crash, recovery should occur within 5 minutes and should be able to auto-restart in case of unexpected failures.

4.3 Security

- User authentication should be protected using OAuth 2.0 and all sensitive data should be encrypted using AES-256 before storage.
- The tool should protect gene set data from unauthorized access and ensure data integrity. If applicable, authentication mechanisms will be in place for restricted functionalities.

4.4 Portability and Compatibility

- The system should support the latest versions of Chrome, Firefox, Edge, and Safari.
- The application should be OS-independent and run on Windows, macOS and Linux.

References

[1] “Genes and genesets,” Genes and GeneSets - GeneWeaver, https://thejacksonlaboratory.github.io/geneweaver-docs/concepts/genes-and-genesets/#genomic-features-genes (accessed Feb. 14, 2025).

[2] “MoSCow method.” Wikipedia, https://en.wikipedia.org/wiki/MoSCoW_method (accessed Feb. 16, 2025).

[3] "IEEE Recommended Practice for Software Requirements Specifications," in *IEEE Std 830-1998* , vol., no., pp.1-40, 20 Oct. 1998, https://ieeexplore.ieee.org/document/720574 (accessed Feb. 16, 2025).

[4] Erich J. Baker, Jeremy J. Jay, Jason A. Bubier, Michael A. Langston, and Elissa J. Chesler. Geneweaver: a web-based system for integrative functional genomics. Nucl. Acids Res. (2012) 40(D1): D1067-D1076.

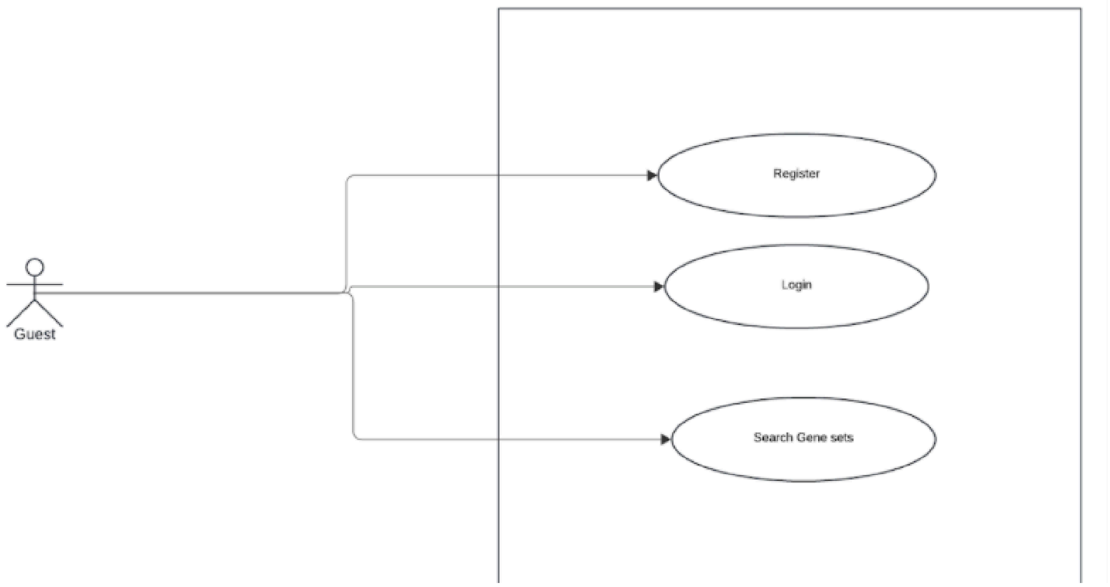
[5] “GeneSet Graph Tool - Figure 2” *GeneWeaver*, The Jackson Laboratory. https://github.com/TheJacksonLaboratory/geneweaver-docs/blob/main/docs/analysis-tools/geneset-graph.md (accessed Mar. 1, 2025).

[6] “Find Variants Tool - Figure 1” *GeneWeaver*, The Jackson Laboratory. https://github.com/TheJacksonLaboratory/geneweaver-docs/blob/main/docs/analysis-tools/find-variants.md (accessed Mar. 1, 2025).

Appendices

Appendix A: Use Cases and Requirements Gathering Work

A.1 User story 1



As a guest user, we want to browse and explore publicly available gene sets and documentation so that we can understand the platform's capabilities before deciding to register.

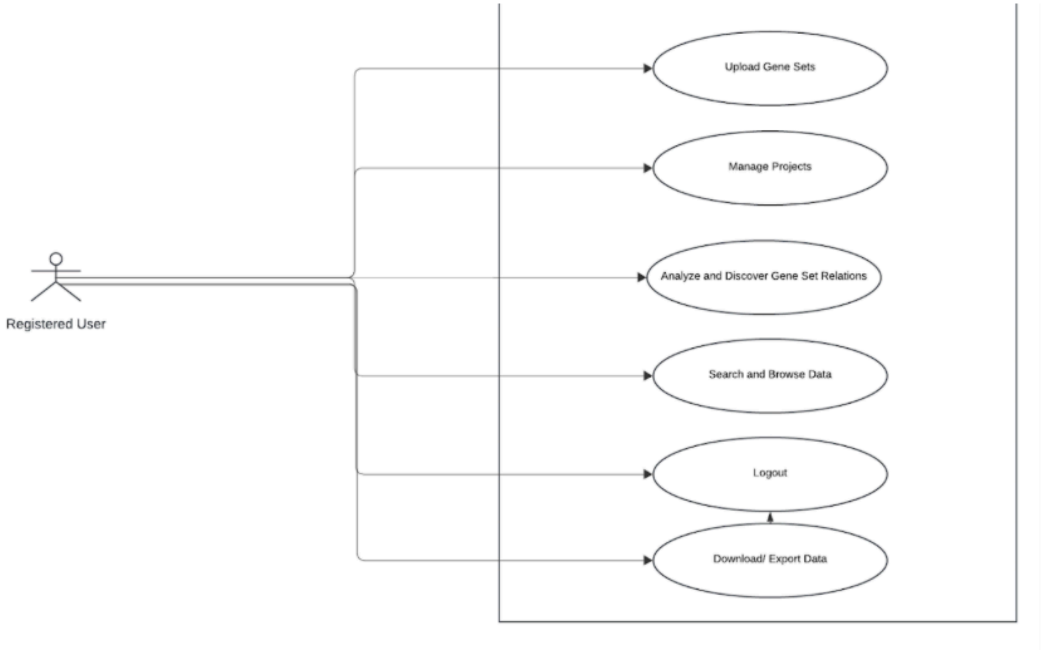
Feature 1.1 - Public Browsing of Gene Sets

This feature enables **Guest** users to discover and explore the publicly available gene sets in GeneWeaver. Its primary goal is to provide a clear entry point for new or casual visitors who wish to learn about the system's offerings without creating an account.

Priority: Must Have (M)

The ability for guests to browse publicly available gene sets is fundamental to encouraging new users to explore the platform and potentially register for full functionality.

A.2 User Story 2



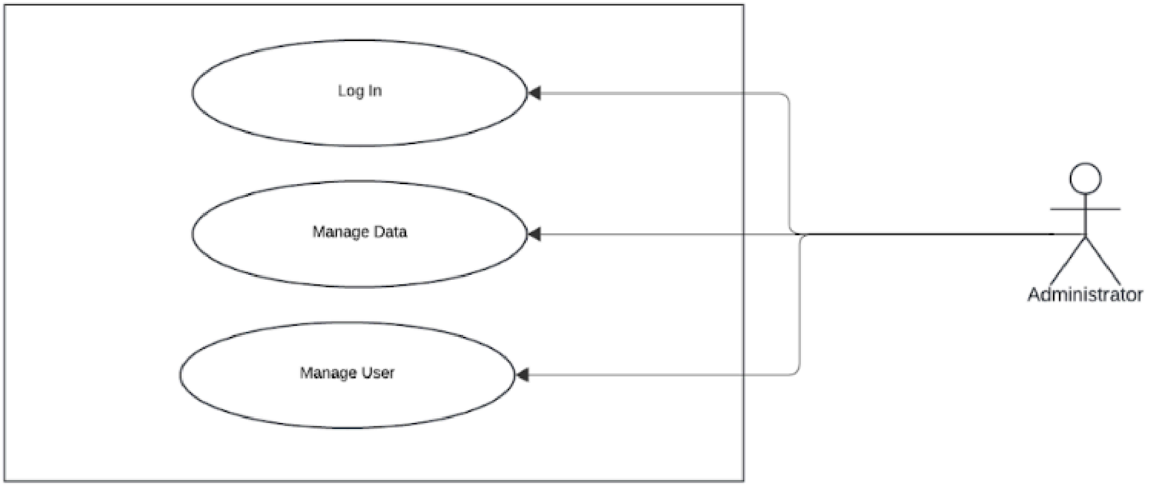
Feature 1.2 - Registered User

As a **registered user**, we want to upload and organize gene sets, so we can easily manage and analyze my research data.

Priority: Must Have (M)

The ability for guests to browse publicly available gene sets is fundamental to encouraging new users to explore the platform and potentially register for full functionality.

A.3 User Story 3



Feature 1.3 - Administrator

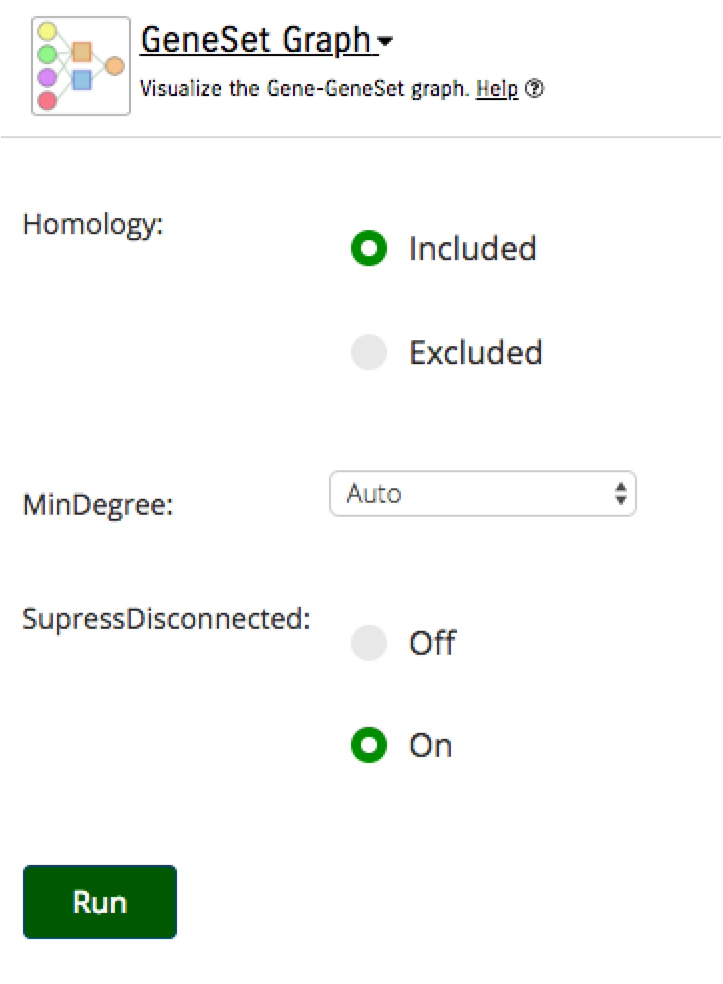
As an **administrator**, we want to manage user accounts and data submissions, so that the system remains secure and reliable for all users. The feature would be an administrative control page.

Priority: Must Have (M)

The system shall allow administrators to review, approve, or disable user accounts and provide tools to monitor and remove problematic datasets.

Appendix B: Analysis Models

B.1 GeneSet Graph Tool



GeneSet Graph ▾

Visualize the Gene-GeneSet graph. Help ⓘ

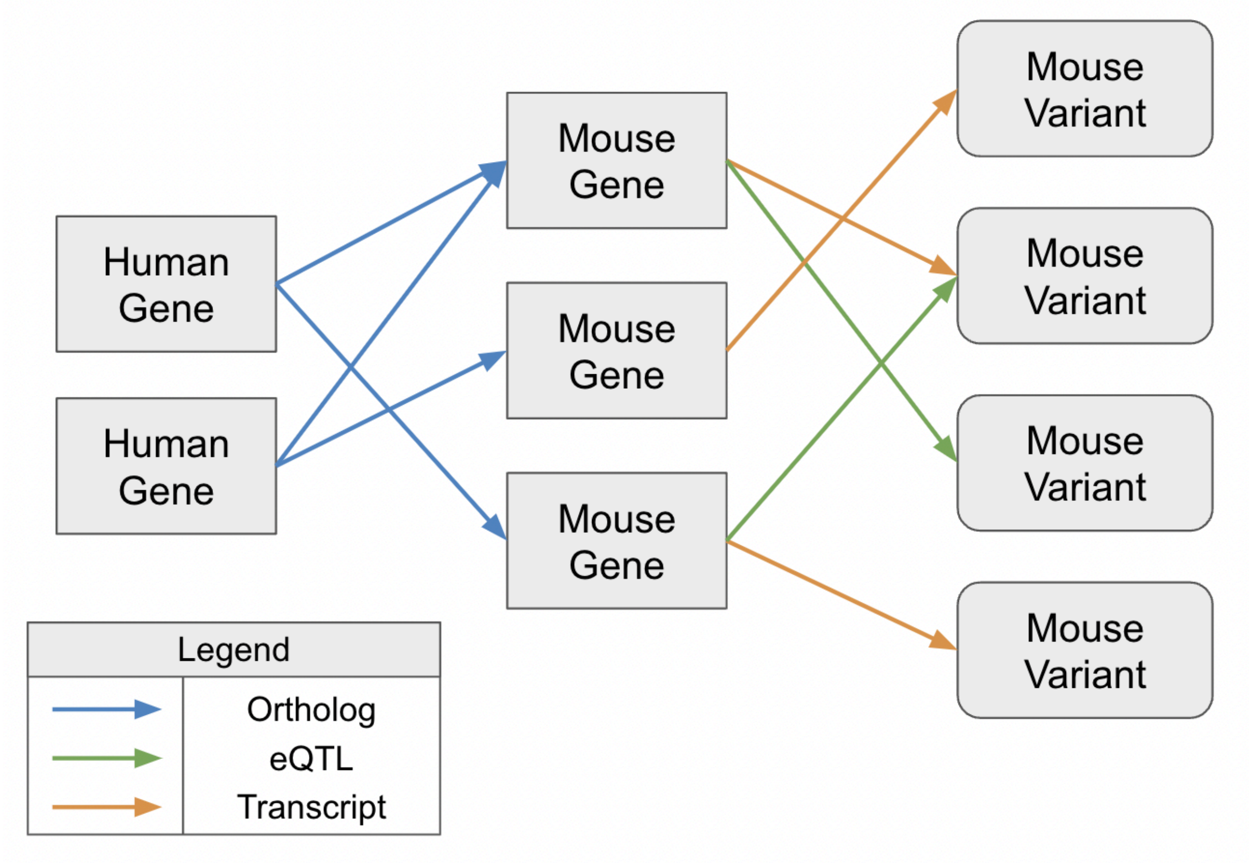
Homology: ● Included ○ Excluded

MinDegree: Auto

SupressDisconnected: ○ Off ● On

“*Figure 2*: GeneSet Graph Selection Icon.” [5]

B.2 Find Variants Tool



“*Figure 1*: Cut of the graph database to map the relationships from a set of human genes to mouse variants.” [6]

Appendix C: Issues List

TBD