GADDNET: A Platform for Connecting Researchers via the Genes and Diseases They Studied and Will Study

Emma Yang

Abstract

Authors, Genes, Diseases, and Drugs in a Network (GADDNET) is a web-based platform that connects researchers and institutions via the genes and diseases they study. GADDNET can help researchers to explore understudied genes and enhance opportunities for collaborations. The platform integrates multiple datasets and presents these datasets to the user as a network centered on a researcher, a gene, a drug, or a disease. The nodes in this network are connected based on published genes and proteins, diseases, and drugs, as well as predicted relevant genes. GADDNET has the potential to dramatically accelerate the progress of drug and target discovery.

The goal of the project is to create a web-based and mobile app that dynamically creates networks linking data centered around a specific researcher, a gene, a drug, or a disease. For example, the app uses the researcher's name to look up the genes, the drugs, and the diseases they have published. These genes, drugs and diseases are found in PubMed abstracts. The genes, drugs, and diseases found are used to identify other researchers who have published research about the same genes, drugs, and diseases. The marquee feature of the platform is that the app also provides genes and drugs predicted to be similar in function to the genes and drugs the researcher has published. These predicted genes and drugs are found based on gene-gene and drug-drug similarity matrices constructed from several different resources. These connections

may help researchers identify understudied genes that could become key drug targets and drivers of disease mechanisms, as well as identify opportunities for drug repurposing.

Introduction

The GADDNET platform aims to bridge the gap between genes that could be important to advancing drug discovery research and the amount of research attention those genes are receiving. Much of the human genome is not being actively studied, while well-researched genes continue to be funded and to be the focus of most investigators.

However, an investigation by Stoeger et al. [ref] on the reasons why potentially important genes are ignored found that the discrepancy is not related to the function of these genes. In fact, they found that biomedical research, especially genetics and genomics research, is guided largely by previous experimentation and the characteristics of genes rather than the physiological significance of individual genes or their connection to diseases. Researchers keep going back to relatively well-understood genes and deepening research in those areas, rather than reaching into new genes and developing an initial understanding of those genes.

This is mainly because access to information about connections between diseases and under-studied genes is not widely available. However, swaths of data are available to make computational predictions about such associations. Providing access to such predictions is one of the goals GADDNET aims to achieve.

GADDNET aggregates connections between genes and investigators that already study the diseases these genes, but also to gene-disease associations that are computationally predicted.

The platform gives authors an intuitive way to access information about their own and others research focus, to allow researchers to broaden their attention to consider additional genes and drugs that they may wish to investigate. By putting this data in the hands of investigators, the platform has the potential to accelerate drug discovery research and to enable the identification of understudied drug targets. It will also help to broaden the distribution of genes that are being actively researched and expanding the pool of well-understood genes and their functions in relation to other genes, drugs, and disease mechanisms. With access to this network visualization tool, researchers have the ability to discover lesser-known genes that could be potentially become important to advancing their research and broadening the distribution of research that define gene function.

Development Goals

GADDNET aims to combine data from PubMed, Geneshot and ARCHS4 (from the Ma'ayan Lab), and DISEASES (from the Jensen Lab) to create a web-based search engine that visualizes this data as a network. The expected outcome is that investigators will use the platform to discover other lesser-known genes that they may decide to study. In addition, patients and clinicians will be able to find information about a disease, who is studying it, and which genes are associated with it.

The user will be able to generate a network based on five types of queries: investigator, gene, drug, institution, and disease. An investigator query would generate a network centered around a person and will show the genes, diseases and drugs that they published based on abstracts on PubMed. In addition, predicted genes, and other authors relevant to those genes, drugs, and

diseases will be added to the network. A gene query would create a network centered around a specific gene, showing the top authors who have published the most papers about those genes, the connected diseases, drugs, and predicted genes. The disease query would create a network surrounding a disease, showing genes related to the disease and the other diseases, authors, drugs, and predicted genes related to those genes.

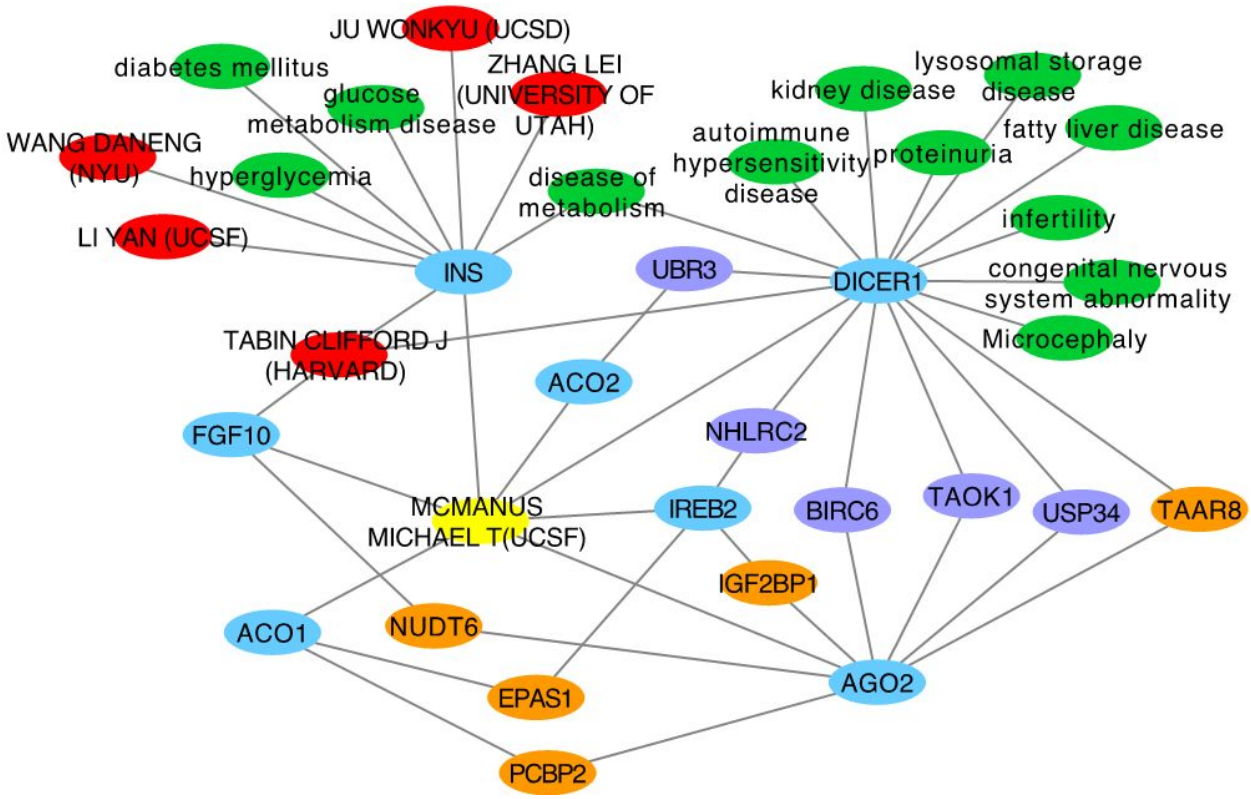

Figure 1. End goal for network of genes, investigators, and diseases (credit: Ma'ayan Lab)

The platform will also leverage natural language processing (NLP) to allow an investigator or a patient to make a query using a question, rather than a structured query. Integrating NLP into the platform will allow complex questions that query information from multiple sources and multiple types of queries to be combined into a single question. The question format will make it easier

and more intuitive to navigate the GADDNET interface. For example, the user could ask, “What investigators are researching the BRCA1 gene?” or “What genes are predicted to be associated with Alzheimer’s disease?” The platform would then generate the network from the relevant query that answers the user’s question.

Materials and Methods

GADDNET was developed around the design of a search engine that would allow the user to search the data available through the platform from multiple different perspectives: searching for investigators, genes, drugs, and diseases. The network design had to be interactive so that the user could explore the network to find information pertinent to their interests. Because of the magnitude of the dataset and the potential scale of the results of many queries, the web app had to be scalable and optimized to prevent slowdown when the user interacted with it.

GADDNET is developed with React.js, with a server written in Node.js. The server is used to access a MySQL database containing the different datasets such as the DISEASES dataset from the Jensen Lab, PubMed data from the PubMed API, and a list of investigators who have published papers about genes, their Open Researcher and Contributor ID (ORCiD) IDs using the ORCiD API. React.js supports the front-end interface of the platform. Data on genes predicted to be relevant by co-expression was sourced from the Geneshot API. The network was created using the d3.js library.

Data Collection and Aggregation

Investigators Included in GADDNET

All of the data sourcing was rooted in a list of investigators who had conducted and published genetics and drug discovery research on PubMed. The list was in a .csv file.

Collecting Data from ORCiD

ORCiD provides a unique identification code for scientific and academic authors and contributors. This allows researchers to obtain a persistent identifier that can be used to claim publications. The ORCiD ID allows the platform to keep track of authors and investigators across articles on PubMed.

The ORCiD numbers for all the investigators in the investigator list were retrieved by using the ORCiD API in a Python script. The ORCiD numbers and the corresponding names of the investigators were stored in a .csv file. The data was then imported into a MySQL database.

Collecting Data from PubMed

The names of the investigators were also used to search PubMed for publications that each investigator has published. The PubMed API was accessed with a Python script that searched the PubMed database for the unique IDs of articles that each author published.

Mapping Articles to the Genes They Discuss

The PubMed articles found from the PubMed API for each author was mapped to the gene that they primarily discussed using the GeneRIF dataset from the National Center for Biotechnology Information (NCBI). Most authors have researched and published work on a gene multiple times,

so the publication count for each gene was also included in the resulting dataset. This dataset was also imported into a table in the MySQL database.

Retrieving Disease Data

Disease-gene associations were sourced from the DISEASES dataset, which was created and is maintained by the Jensen Lab. The disease related to each gene in the dataset was retrieved from this dataset and stored in a MySQL data table.

Developing the GADDNET Server

Since the web app is still in a development build, the MySQL database is running on a local server. A web server written with Node.js queries the SQL database in sequence.

Each of the three types of queries supported by GADDNET (investigator, gene, and disease) has a different sequence of SQL queries. In order to separate each flow of queries, the server establishes three different API endpoints, each of which takes the search term as a parameter. When the GADDNET website requests data from the server, it passes the search term input by the user in the front-end in the URL that reaches the server API.

The sequence of SQL queries depended on what kinds of direct and indirect connections could be made from the data. For example, the first direct connection that could be made for authors and diseases was the genes that the queried person worked on or the searched disease is linked to. On the other hand, there are many more direct connections that can be made for a gene query: the diseases it is linked to, the top investigators working on the gene, and the predicted genes it is co-expressed with from Geneshot. These kinds of connections were taken into account when designing the flow of SQL queries and API requests the Node.js server made.

Since each SQL query and API request takes time to be dispatched and received, the server leverages the asynchronous capabilities of Node.js. Instead of simply returning values of the function that sends out queries, the server uses callbacks to handle API and SQL response data and adds it to the dictionary of data returned to the front-end by the server when all the data has been collected.

Retrieving Data for Investigator Queries (Figure 2)

An investigator query begins with a SQL query which retrieves the genes an investigator has worked on and the publication counts for each of those genes. This list of directly-connected genes is used to drive the rest of the server queries.

The server then builds two SQL queries based on the list of gene symbols retrieved from the initial query. One of the queries searches for other investigators in the database who are working on the same genes, excluding the original investigator the user searched for. The other query retrieves the diseases linked to each gene.

Lastly, the co-expressed genes for each of the genes is retrieved with the Geneshot API. The API request is sent with the request-promise library. In Node.js, a Promise ensures that the server waits for the API response data to be sent back by the API before moving on in the program.

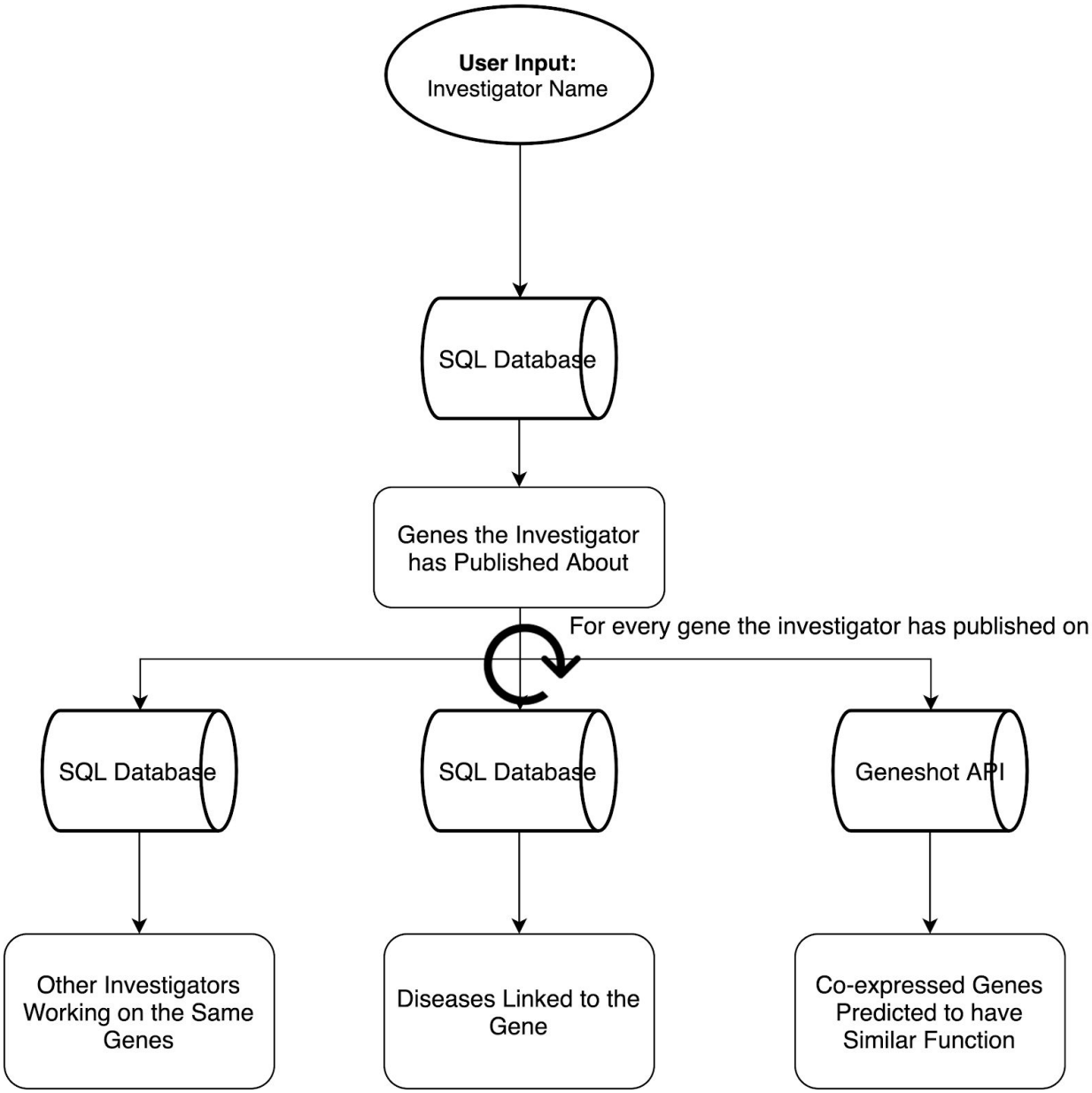

Figure 2. Server Query Flow for Investigator Query

Retrieving Data for Gene Queries (Figure 3)

The gene query only has one level of child nodes, because all of the data collected for GADDNET centers around genes. Therefore, no "in-between" step needs to be made in order to make connections between genes, diseases, and authors.

Two SQL queries are made by the server. One finds all authors who have published papers about the searched gene. The other finds all diseases linked to the gene.

An API call to Geneshot is made to find co-expressed genes. The 3 co-expressed genes with the highest similarity scores are saved. Again, this API call is made with a request encapsulated in a promise.

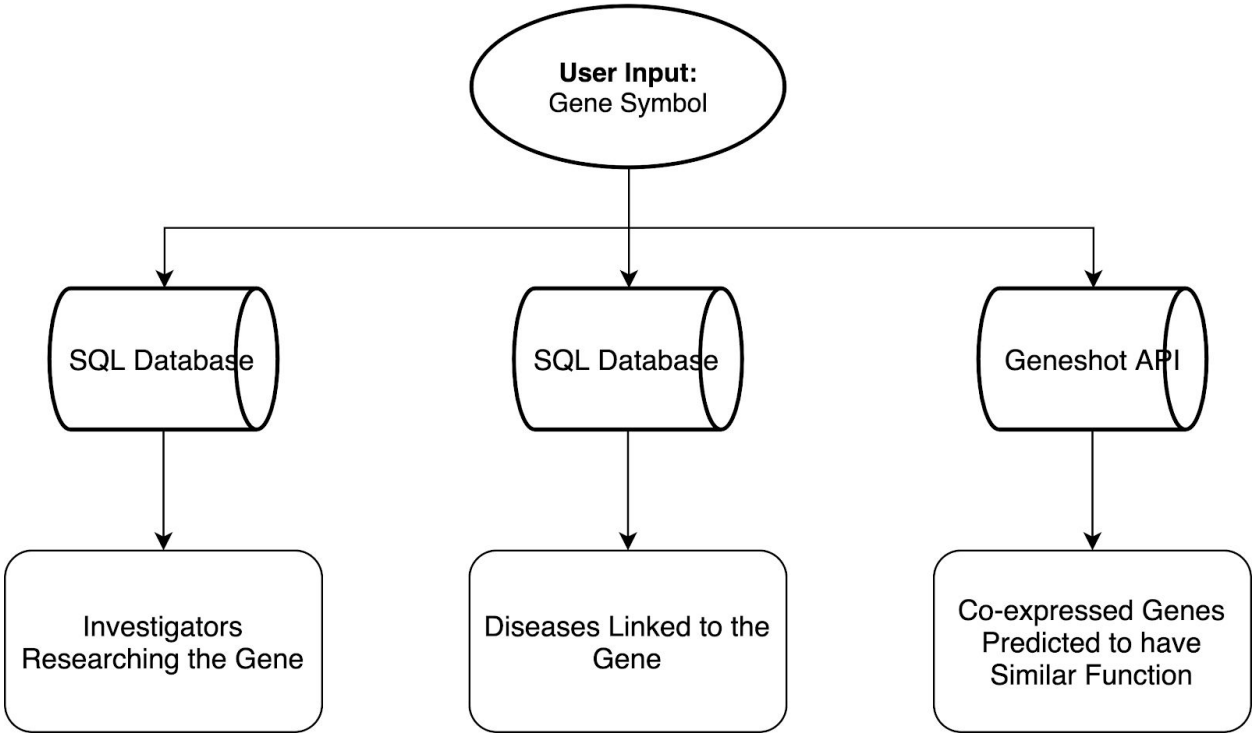

Figure 3. Server Query Flow for Gene Query

Retrieving Data for Disease Queries (Figure 4)

The disease query begins with a SQL query that retrieves all genes linked to the given disease.

From there, the server goes through a process similar to the gene query. First, investigators who have published about the genes found and author diseases linked to the genes are found. Then, the genes are sent to the Geneshot API where co-expressed genes are found.

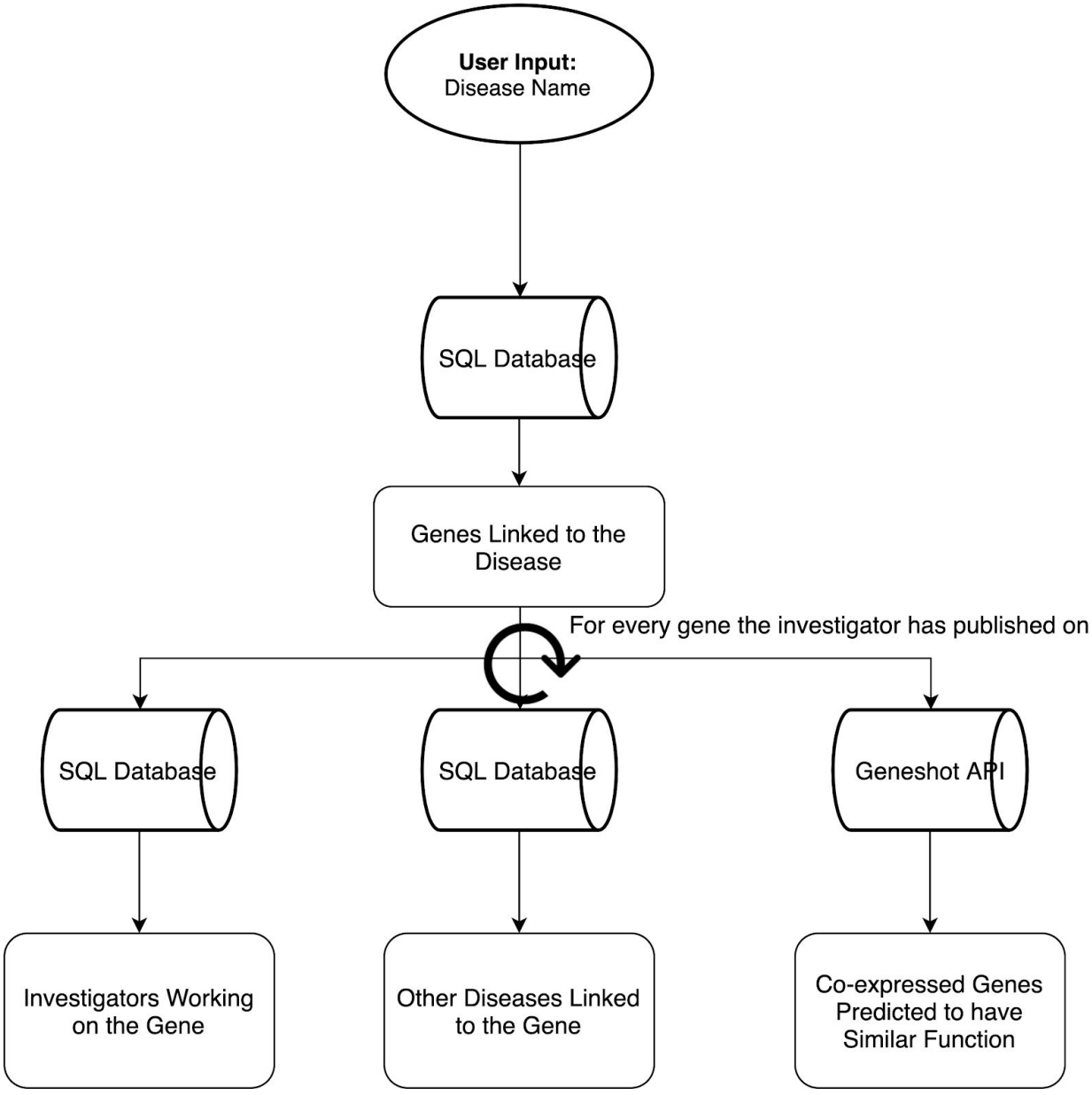

Figure 4. Server Query Flow for Disease Query

Server Response Data Structure

All of the data is sent to the API endpoint that the front-end platform requested in the JSON format, allowing the data to be easily parsed through. All of the data is returned in four columns: source, target, weight, and data type. The source and the target represents the two nodes, or datapoints, that the connection found links. The data type indicates how the given node is connected to the original query, such as gene or disease. The weight has different meanings depending on the data type. For genes an investigator has worked on, the weight indicates the number of works the author has published about a gene. For diseases a gene is connected to, the weight indicates the z-score of the link, showing how significant the connection between the gene and the disease is. For predicted genes, the weight indicates the similarity score, which is derived from the gene-gene similarity matrix behind Geneshot.

Visualizing GADDNET Data

Constructing the Network

The data is visualized so that it can be displayed by the d3 network. After a user submits a search term and the relevant SQL and API queries are made through the server, the resulting collection of data is parsed based on where it came from: disease dataset, predicted genes, or genes the author worked on. The color of the node representing each row of data depends on what type of data it is. Gene data, disease data, and author data all have different colors. The data is parsed so that it is organized as a collection of nodes and links, and is displayed in the user interface.

Natural Language Processing

The natural language processing in GADDNET is powered by the Google Natural Language API (NLP API). When the user submits a query, GADDNET sends the question to the NLP API. The

API parses out which words are pertinent to the query that will produce the network that answers the user's question. For example, if the user asked, "Who is working on BRCA1?," the NLP API would return the word "BRCA1" as it was the subject of the question.

GADDNET then searches through each database to find out whether the word the NLP picked out was a person, gene, or disease. Based on what the subject of the question was, the search engine then uses the server to make the corresponding query and to display the network that answers the user's question.

Results

The GADDNET platform generates networks of genes, diseases, and investigators based on a user's search. The interface is divided into four parts, where each part allows the user to make a different type of query. Each part of the platform has its own tab in the user interface, allowing the user to select what kind of query they would like to make.

1. Natural Language Input (Figure 5 & 6)

The first tab, labeled "Ask Me Anything," allows the user to generate a network based on a natural language question. When the user inputs the question, GADDNET uses the results of the natural language analysis to direct the main word indicating the search term to the correct query. For example, if the user asks, "Who is working on BRCA1?," GADDNET will show the network for the BRCA1 gene, including the nodes with the investigators who work on the gene.

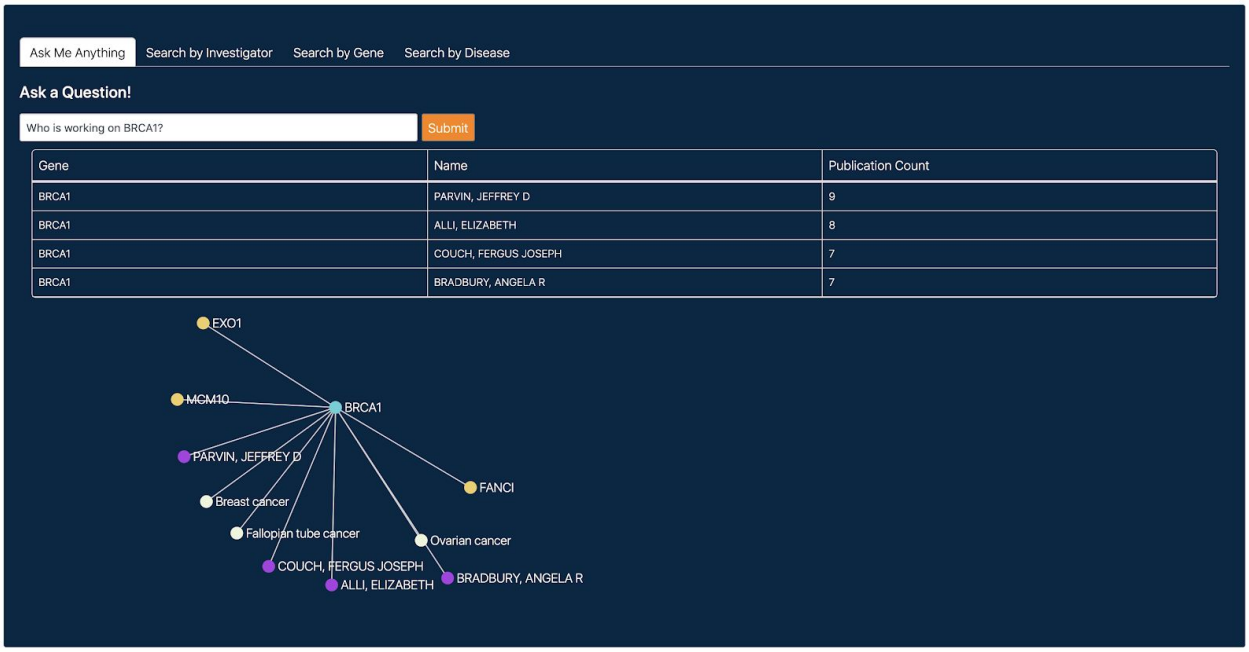

Figure 5. Natural language input example for BRCA1 gene query

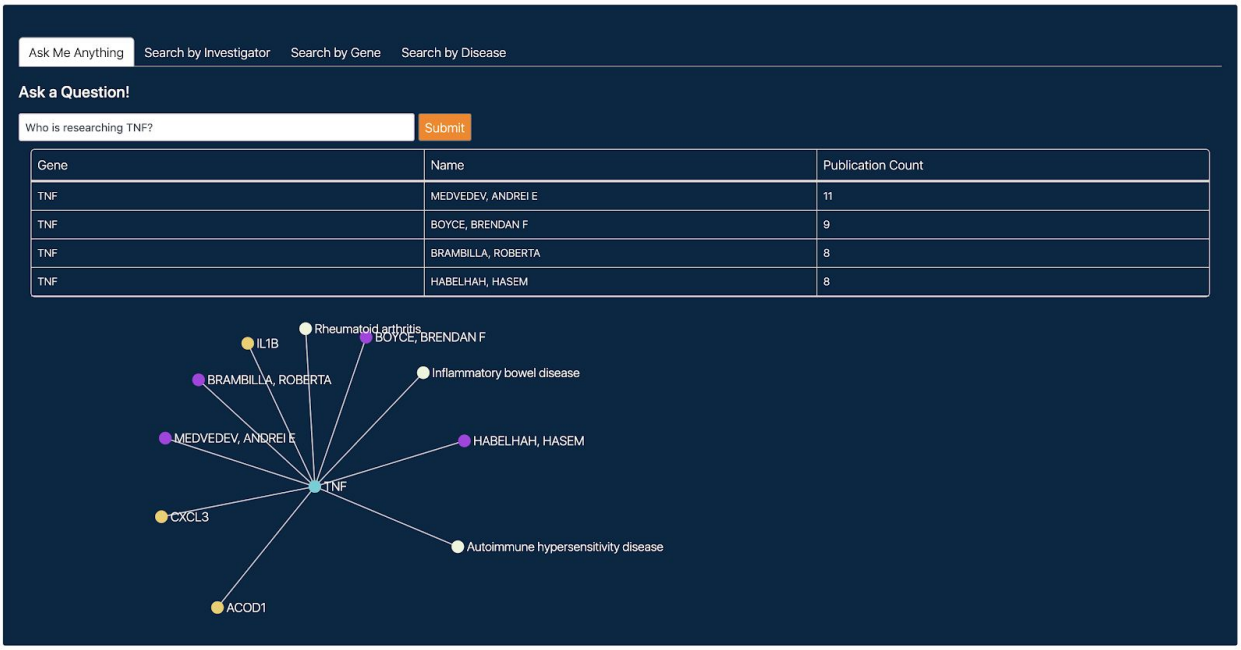

Figure 6. Natural language input example for TNF gene query

2. Searching by Investigator, Gene, and Disease

The second, third, and fourth tabs allow the user to generate a network based on an investigator, gene, or disease.

If the user would like to generate a network based on an investigator, the name of the investigator can be used as the search term. GADDNET will then generate the network with the genes the investigator has published on and the related data to those genes (Figure 7). If the user would like to generate a network based on a gene, the symbol for the gene (e.g. BRCA1, TNF) can be used as the search term. GADDNET will then generate a network with the related data (Figure 8 & 9). For searching based on a disease, the name of the disease can be used (e.g. Lyme disease, Alzheimer's disease) (Figure 10 & 11).

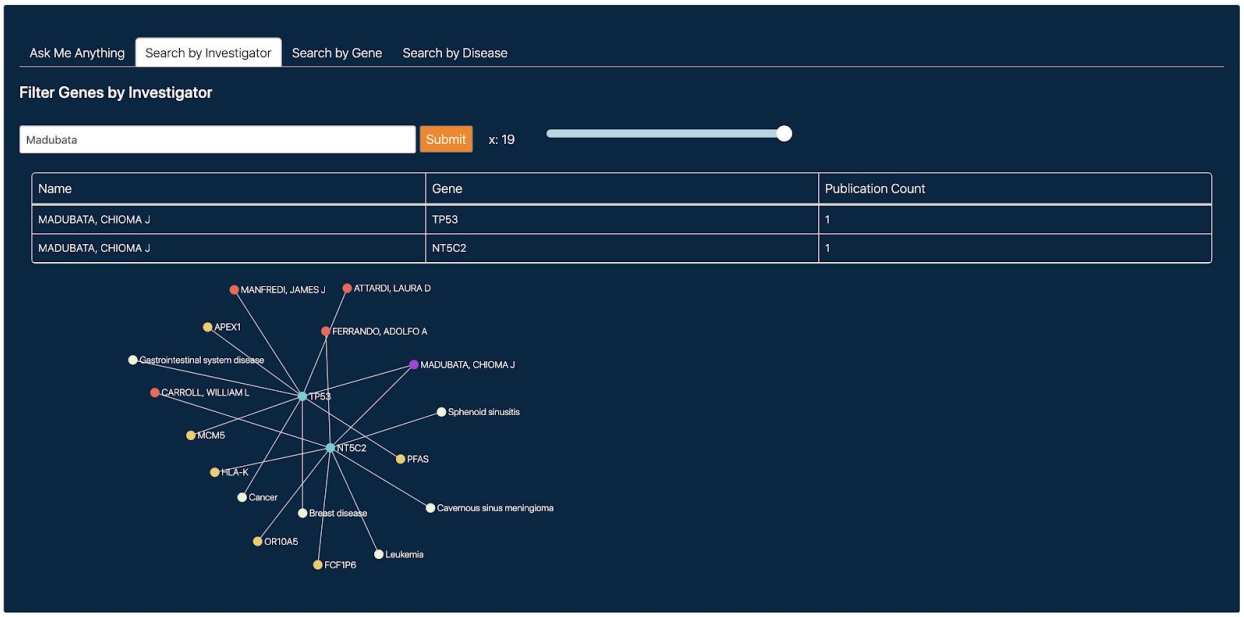

Figure 7. Investigator-based query: network for investigator named "Madubata"

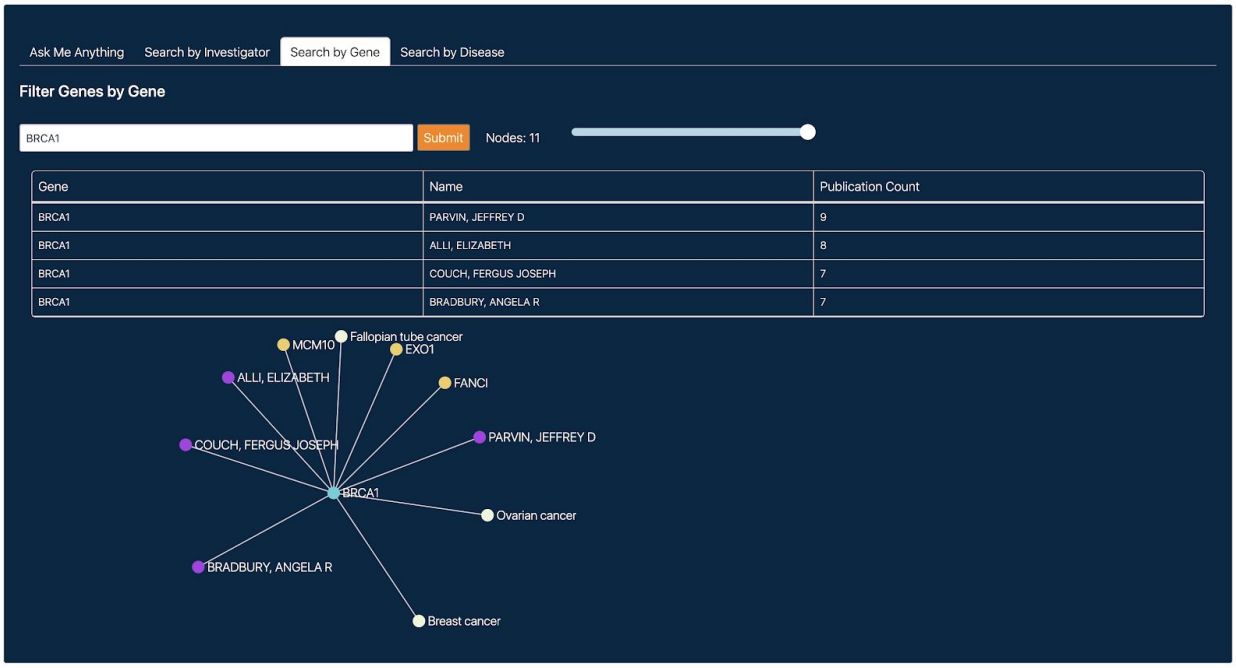

Figure 8. Gene-based query: network for BRCA1 gene

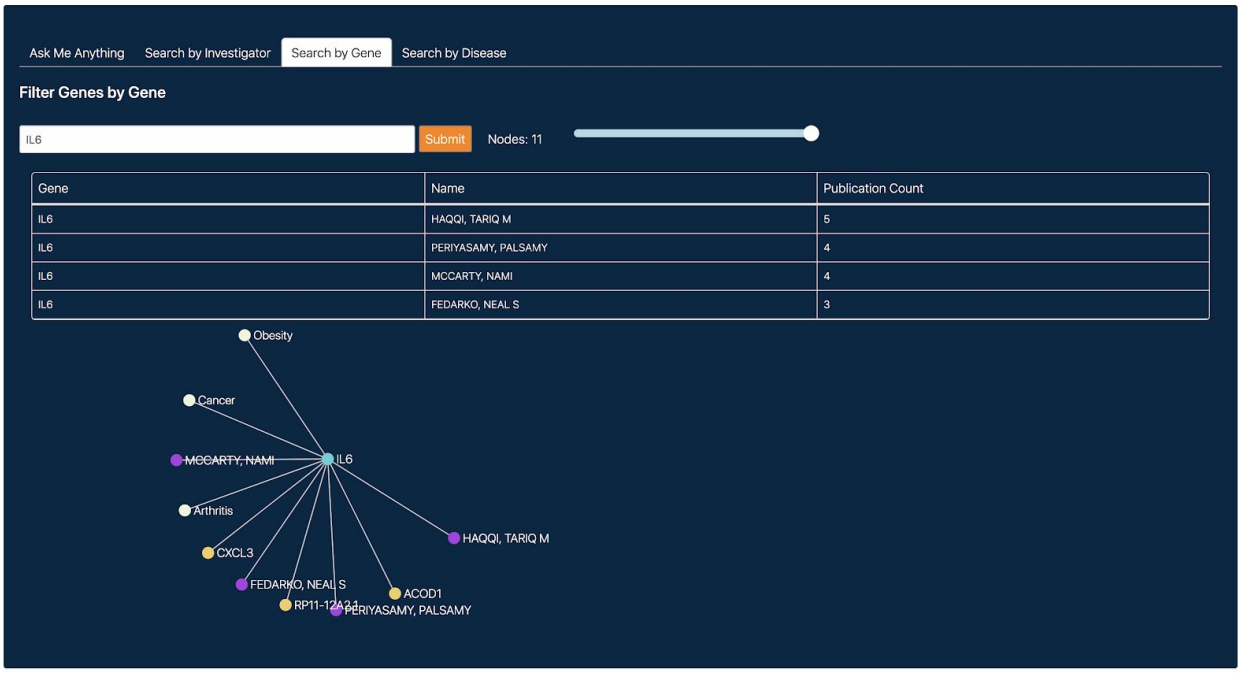

Figure 9. Gene-based query: network for IL6 gene

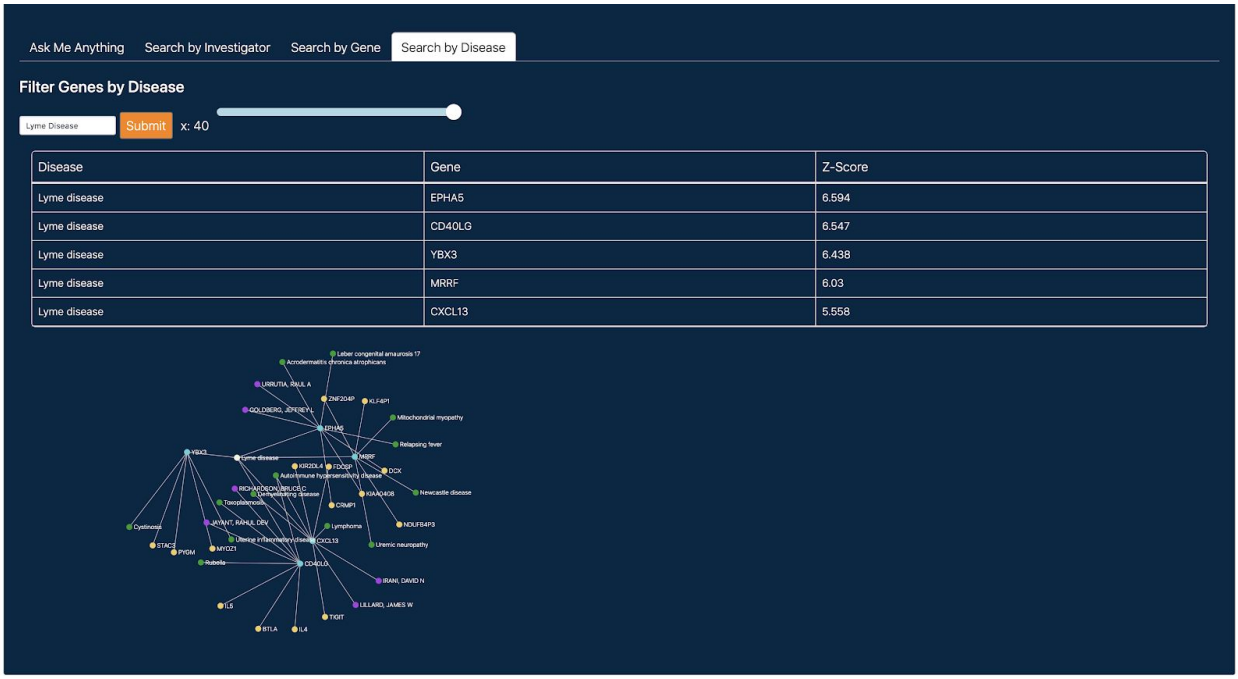

Figure 10. Disease-based query: network for Lyme Disease

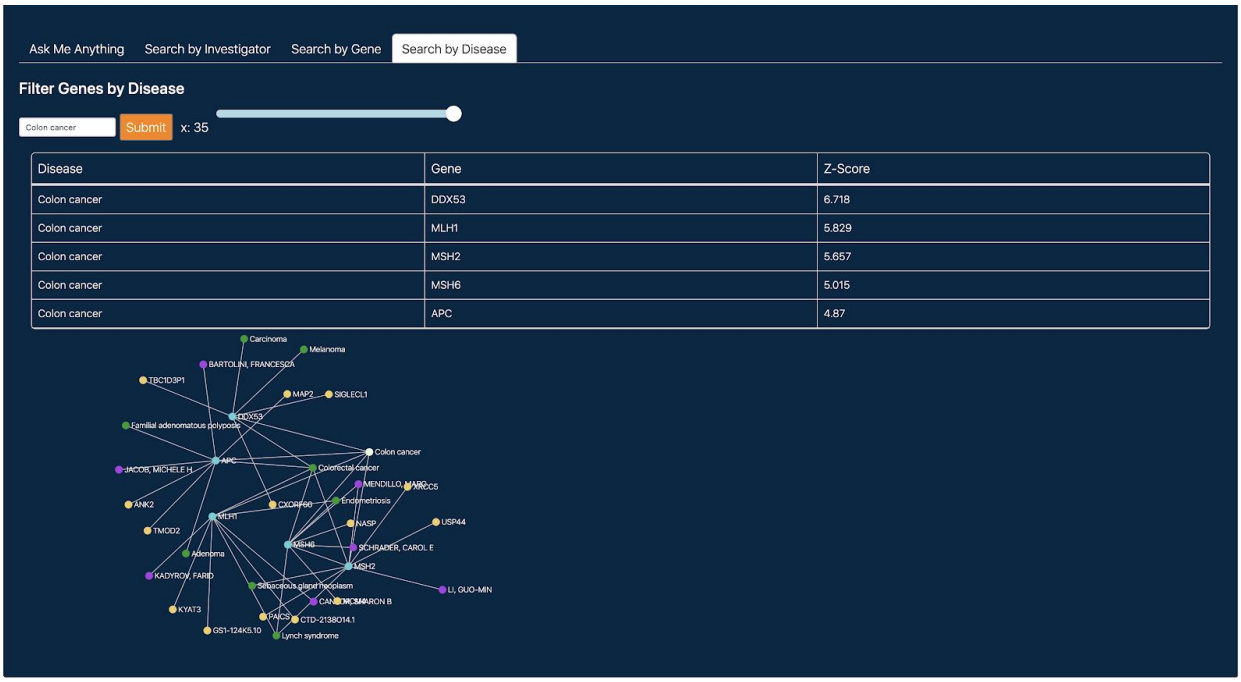

Figure 11. Disease-based query: network for Colon Cancer

Manipulating the Network Generated by GADDNET

The network that GADDNET generates using the d3 library is interactive. The user can drag nodes around the screen and reconfigure the position of the network. When the user mouses over a node, the link and the connected node will be outlined in blue.

The network is constructed as a force graph, meaning that it dynamically reconfigures itself every time a new network is generated. The force graph has a set link length, strength, and graph gravity that controls how spread out the nodes are and the strength of the "pull" towards the center of the graph.

The user can also mouse over the links to see the weight of the connection between two nodes. For links between investigators and genes, mousing over the link will reveal the number of publications the investigator has on PubMed about that gene (Figure 12). For links between genes and predicted genes, the tooltip text shows the similarity score for the two genes (Figure 13). For links between genes and diseases, the tooltip shows the z-score of the gene-disease connection (Figure 14).

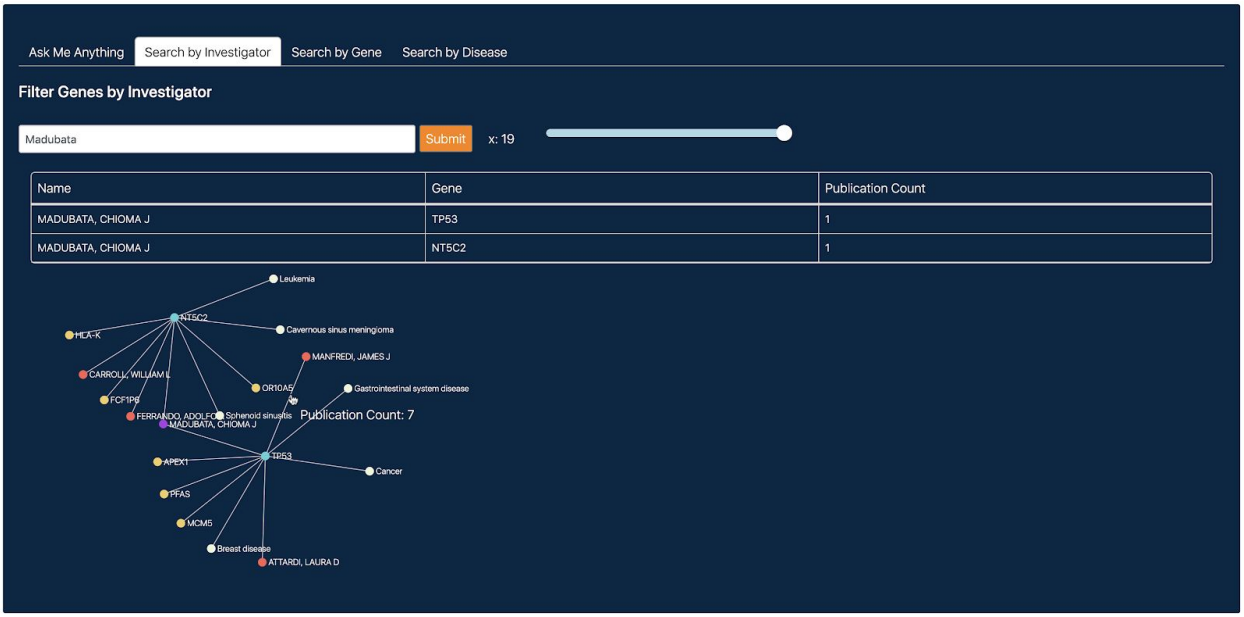

Figure 12. Tooltip showing publication count for a gene the investigator has worked on

Figure 13. Tooltip showing similarity score for predicted gene co-expression

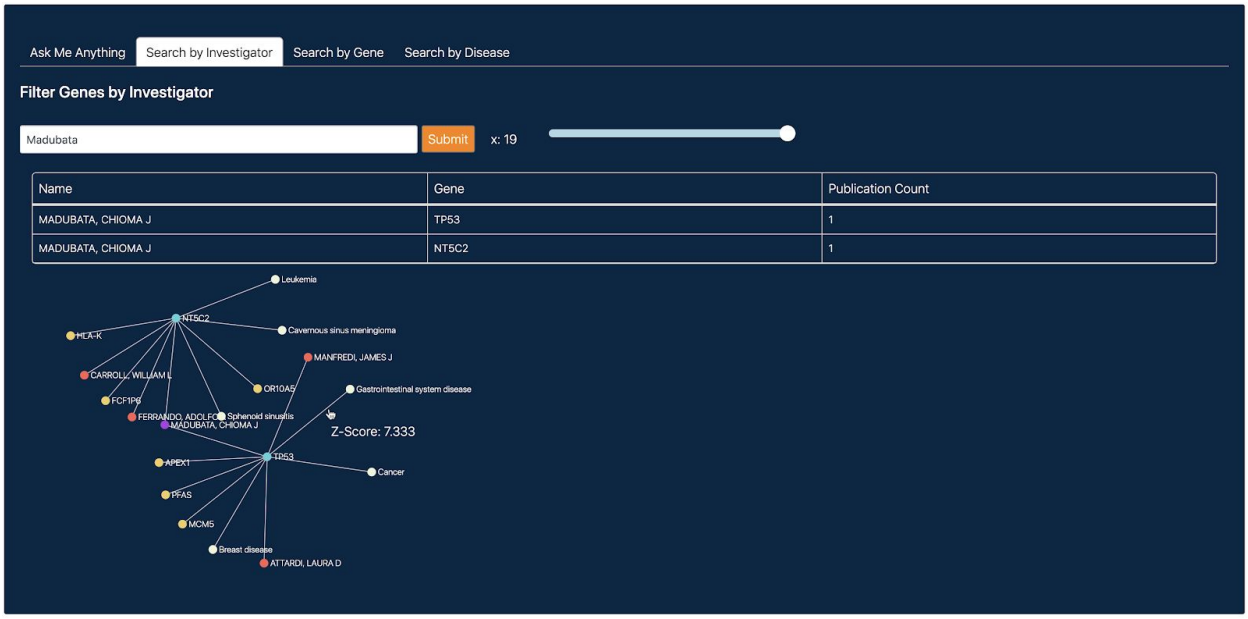

Figure 14. Tooltip showing z-score for a disease that a gene is connected to

When the network is generated, a slider appears adjacent to the search bar. The slider allows the user to control the number of visible nodes in the graph. The network dynamically updates to remove and add nodes and the links connecting them. The outermost nodes, representing the most indirect connections, are removed first. This allows the user to decrease the number of nodes if there are a large number of connections or if the investigator the user searcher for has published papers on many genes and therefore has many connecting nodes (Figure 15).

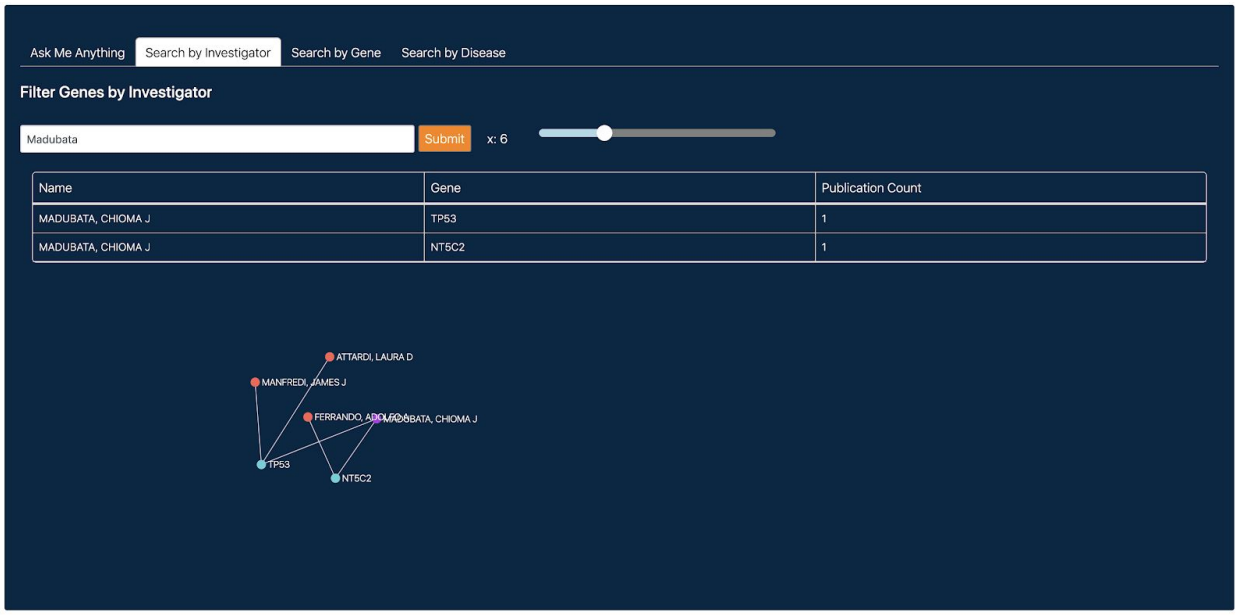

Figure 15. Slider dynamically manipulating the number of nodes appearing in the network

Discussion & Future Work

While GADDNET already covers multiple datasets and types of data, there is always room to expand. The investigator list that the platform's data collection was based around also includes the institutions and departments each investigator is a member of. This data has yet to be leveraged in the platform. Integrating research institution data into the platform is one of the next steps for GADDNET. This would allow researchers to limit their search to their own research institutions or to look for their counterparts in a specific institution.

Another direction for institutional data would be focusing on a specific institution to allow for richer and more complete and meaningful data to be collected and added to a institution-specific platform. Building out the platform in this way would allow there to be more productive

information available to researchers using the platform to discover more about their field of research, beyond their own focuses.

This information will also be integrated into the NLP part of the platform. This would allow users to ask questions such as "Who is working on TNF at Mount Sinai?" or "Which research institutes are specialized in Breast Cancer?."

The slider feature of the platform could also be expanded to not only decrease and add nodes that existed in the original network generated when the user searched for an investigator, gene, or disease, but also to add increasingly extended connections. This would allow the user to have more flexibility and control over how many connections they can see.

The GADDNET platform has yet to be deployed as it is still in a development stage. When it is released, it has the potential to help institutions and investigators broaden the scope of their research and help to bring under-researched genes into focus. It also serves to help patients search for a disease or a gene they are interested in and learn about related genes, diseases, and the investigators who are working on that kind of research. This will help patients find out about potential scientific advances that they could benefit from or opportunities to participate in drug research. A combination of these efforts, from institutions, researchers, and patients, will help to advance and accelerate drug discovery research.

Major References

Facebook Inc. (2018, June 26). Create a New React App. Retrieved November 12, 2019, from React website: https://reactjs.org/docs/create-a-new-react-app.html

Google Cloud. (2019, April 10). Natural Language. Retrieved December 12, 2019, from Google Cloud website: https://cloud.google.com/natural-language/

Kamenzky, N., & Abonil, T. (2019, November 4). Request-Promise. Retrieved December 5, 2019, from NPM website: https://www.npmjs.com/package/request-promise

Lachmann, A., Schilder, B. M., Wojciechowicz, M. L., Torre, D., Kuleshov, M. V., Keenan, A. B., & Ma'ayan, A. (2019). Geneshot: search engine for ranking genes from arbitrary text queries. *Nucleic Acids Research*, *47*(W1), W571-W577. https://doi.org/10.1093/nar/gkz393

Lerner, A. (2016, November). Introduction to Promises. Retrieved December 9, 2019, from FullStack React website: https://www.newline.co/fullstack-react/30-days-of-react/day-15/

National Center for Biotechnology Information. (2011, May 27). Retrieve PMC article identifiers (PMCIDs) from a search. Retrieved December 10, 2019, from PubMed Central website: https://www.ncbi.nlm.nih.gov/pmc/tools/get-pmcids/

Oprea, T. I., Bologa, C. G., & Brunak, S. (2018). Unexplored therapeutic opportunities in the human genome. *Nature Reviews Drug Discovery*, *17*, 317-332. https://doi.org/10.1038/nrd.2018.14

ORCiD. (2019, October 3). Basic Tutorial: Searching Data Using the ORCiD API (3.0+). Retrieved December 10, 2019, from Member Support Center website: https://members.orcid.org/api/basic-tutorial-searching-data-using-orcid-api-30