Contents

GeneWeaver Documentation

GeneWeaver is a web application for the integrated cross-species analysis of functional genomics data from heterogeneous sources. The application consists of a large database of gene sets curated from multiple public data resources and curated submissions, along with a suite of analysis tools designed to allow flexible, customized workflows through web-based interactive analysis or scripted API driven analysis. Gene sets come from multiple widely studied species and include ontology annotations, brain gene expression atlases, systems genetic study results, gene regulatory information, pathway databases, drug interaction databases and many other sources. Users can retrieve, store, analyze and share gene sets through a graded access system. Gene sets and analysis results can be stored, shared and compared privately, among user defined groups of investigators, and across all users. Analysis tools are based on combinatorics and statistical methods for comparing, contrasting and classifying gene sets based on their members.

Each **"gene set"** contains a list of genomic features, free text descriptive content, ontology annotations and gene association scores. Genomic features are mapped within and across multiple species. Currently 10 species are supported, *Mus musculus, Homo sapiens, Rattus norvegicus, Danio rerio, Drosophilia melanogaster, Macaca mulatta, Caenorhabditis elegans, Saccharomyces cervisiae, Gallus gallus, Canis familiaris.* Additional species are added in response to community request.

GeneWeaver allows users to integrate these diverse functional genomics data across species, tissue and experimental platform to address questions about the relations among genes and biological functions. Applications include the prioritization of gene-disease associations from multiple evidence sources, the classification and comparison of biological functions based on biological substrates, and the identification of similar genes based on function. Cross species analysis enables the discovery of conserved mechanisms of biological functions, and the discovery of divergent functions served by conserved biological mechanisms.

PDF Version of the GeneWeaver 2.0 documentation.

Table of Contents

  - Clustering
  - DBScan Gene Clustering
  - MSET
  - Anchored Biclique of Biomolecular Associations (ABBA)
  - Boolean Algebra
- API
- FAQ
- Searching
  - General Search
  - Advanced Search
- Gene Set Utilities
  - View My Genesets
  - Geneset Details Pages
  - Edit Geneset Page
  - Similar Genesets
- Software and Installation
- Data Availability and Exporting
- Other Important Links
- Publications
- Data Sharing and Privacy Policies
- Contact Us
- Acknowledgements

Getting Started

- The **Quick Start Guide** is designed to help get new users up and running quickly with a basic two page GeneWeaver Guide.
- The **GeneWeaver Tutorial** provides a guided tutorial exercise to get you familiar with using and interpreting basic GeneWeaver analyses and tools through applied examples. The exercise is suitable for use in demonstrations, workshops and courses.
- For more details on **how to get started** by searching for some specific genes of interest and analyzing them, read the **General Search** section.
- To learn more about how to work with other researchers and their data within GeneWeaver check out **Users and Groups**.
- **GeneWeaver Movie** provides video clips featuring examples of the use of GeneWeaver tools.
- **FAQ**

Uploading Gene Sets

To compare individual user generated gene sets among a collection or to the large database of publicly available sets of genes, gene sets must first be uploaded and added to analysis projects. Registered users can log in and **upload single gene sets** or make use of the **Batch Gene Set Upload** process. If you have questions about what metadata to enter, see the **General Definitions** page and the **Standards for Common Gene Set Types**.

Upload a Single Gene Set

1. On the Navigation bar, open the Manage GeneSets Menu and select “Upload GeneSet”.
2. Fill in the descriptive metacontent fields with a GeneSet name that would be interpretable to a general user of GeneWeaver, following our curation standards and suggestions. A short figure label is used to

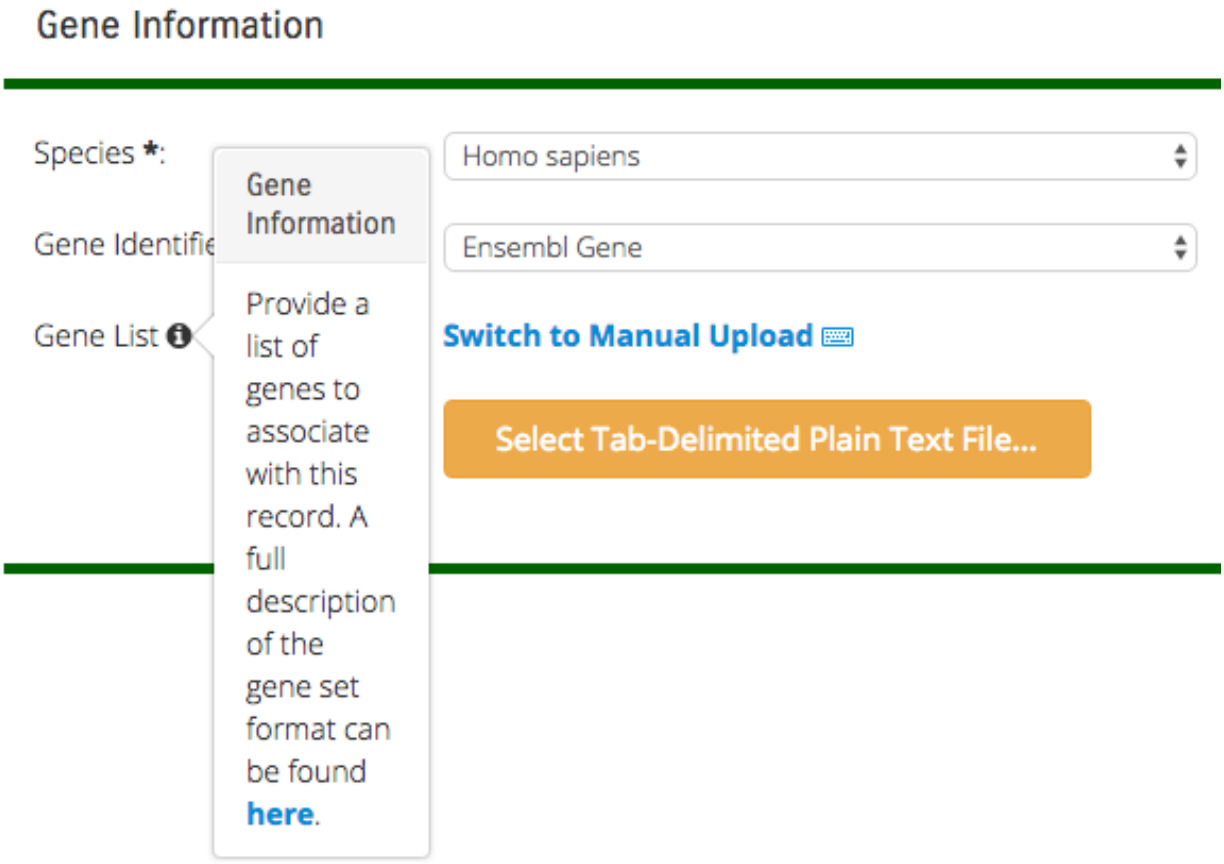
readily identify this gene set in visualization. Select a score type used to associate genes with the list, e.g. a p-value, q-value, correlation coefficient, effect size or binary association. The GeneSet description field should be used to provide detailed information about how the genes were associated with the list, including experimental and analysis information, rules for inclusion, and source information if the gene set comes from a publication or other data resource.

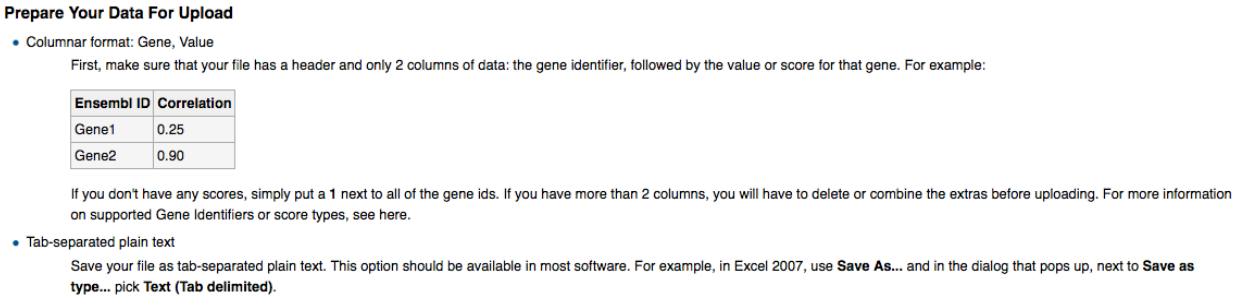
3. Choose Access permissions for your gene set. First, use the pulldown menu to select whether this is a public gene set available to any GeneWeaver user, or a private gene set available only to you or your groups. Next, if the set is private, use the list to select the groups that may access the gene set in database searches and analyses.

4. Provide publication information. If a PubMed ID is available, enter it and click the arrows. The publication information will be automatically imported. If the publication is pending, or the gene set is not associated with a publication you may enter a working title, authors and abstract information.

5. Choose the species and identifier used in your gene list. It is beneficial to use an identifier that best reflects the measured genomic feature on your list. For example, in a microarray experiment, use the specific probe ids from the microarray, and in a transcriptome alignment from RNA seq, use the transcript ID. Gene symbols are frequently updated and are sometimes not unique. Once the gene set is in GeneWeaver it is straightforward to display the gene symbols that best match the ids used in the upload step.

6. Type values, paste them in or select a file containing your gene list and scores by clicking on "Switch to File Upload". Format your input as two column tab-delimited text. For short gene sets, you may copy and paste a selection into the upload form. For larger gene sets, you can prepare a separate text file for upload.





Prepare Your Data For Upload

- Columnar format: Gene, Value

  First, make sure that your file has a header and only 2 columns of data: the gene identifier, followed by the value or score for that gene. For example:

Ensembl ID	Correlation
Gene1	0.25
Gene2	0.90

  If you don't have any scores, simply put a **1** next to all of the gene ids. If you have more than 2 columns, you will have to delete or combine the extras before uploading. For more information on supported Gene Identifiers or score types, see here.

- Tab-separated plain text

  Save your file as tab-separated plain text. This option should be available in most software. For example, in Excel 2007, use **Save As...** and in the dialog that pops up, next to **Save as type...** pick **Text (Tab delimited)**.

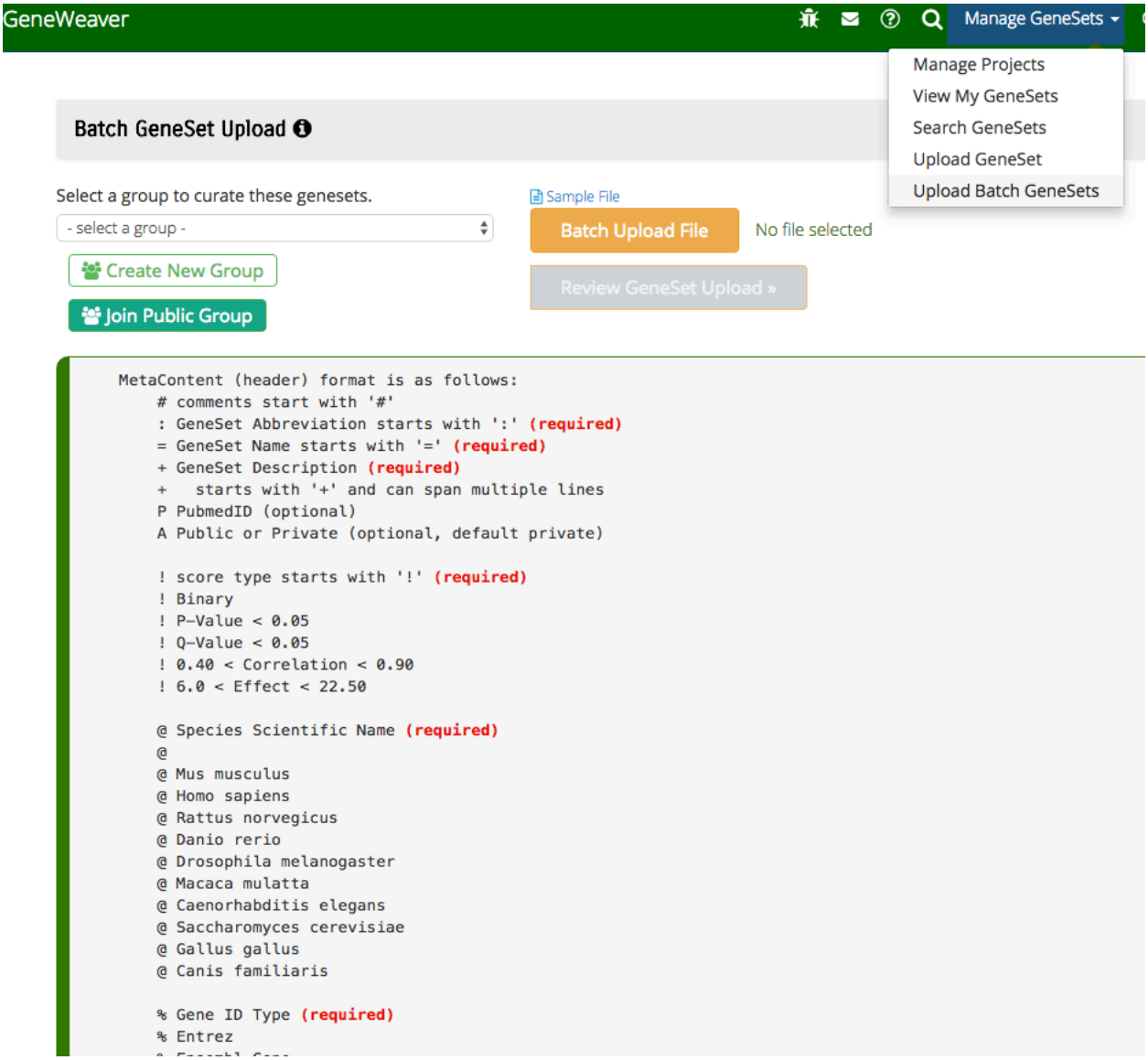
7. Click "**Review GeneSet Upload**".
8. Note that if any genes are entered incorrectly they will not be added, only those that use a valid gene identifier will be included.
9. Review results of the upload and add annotations. See gene set details. To use your new gene set in analyses, you must add it to projects.

Top ↑

Batch Gene Set Upload

If you have many gene sets to upload, for example, the results of a clustering analysis, use the bulk upload form. An example of a bulk upload file is provided. Contact the GeneWeaver team for assistance with very large batch submissions and integration of large scale data resources.

On the navigation bar, under "Manage GeneSets" select "Upload Batch GeneSets".



This page requires that a group is selected to curate the genesets. A private group can be used if the data will not become public. To learn why curation is necessary and how to curate go here.

A sample upload file that includes the formatting rules is displayed on the page and a sample file may also be opened by clicking on the "Sample File" link.

When your file is prepared, click on "Batch Upload File" to select it. Then click on "Review GeneSet Upload" to start the upload process.

When completec, review the results of the upload and add annotations. See gene set details. To use your new gene set in analyses, you must add it to projects.

Top ↑

Users and Groups

GeneWeaver is available without registration to enable all users to search the database and analyze gene sets. Registered users can access several additional features including long-term storage of gene sets, projects and

results. Registered users can also form groups, designate administrators and share gene sets, projects and results to the members of their user group.

Guest User

If you prefer to not register, you will become a guest user by doing a search, selecting some gene sets and adding them to a new project. This project can be used by the analysis tools but will not persist beyond 24 hours.

Registration

All pages contain a navigation bar at the top. In the right hand corner, click on "Welcome Guest" and select "Create Account". The only information needed is your name, email and a password.

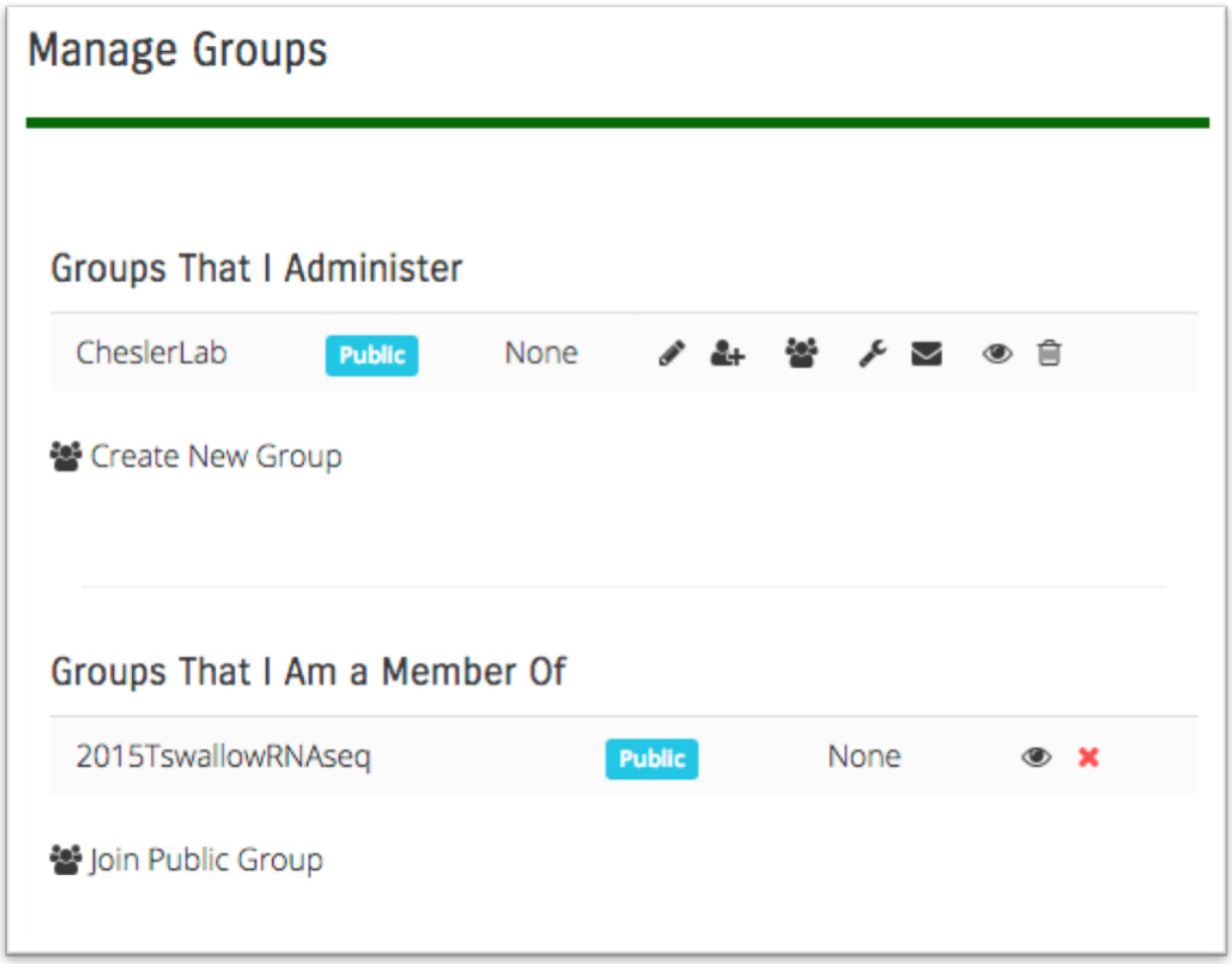
Accounts Page

Once registered, use the Welcome drop-down to log in. A logged in user will see Welcome and their name on the navigation bar. Click there and select "Account Settings". There also is a link to the Account Settings page on the page footer.

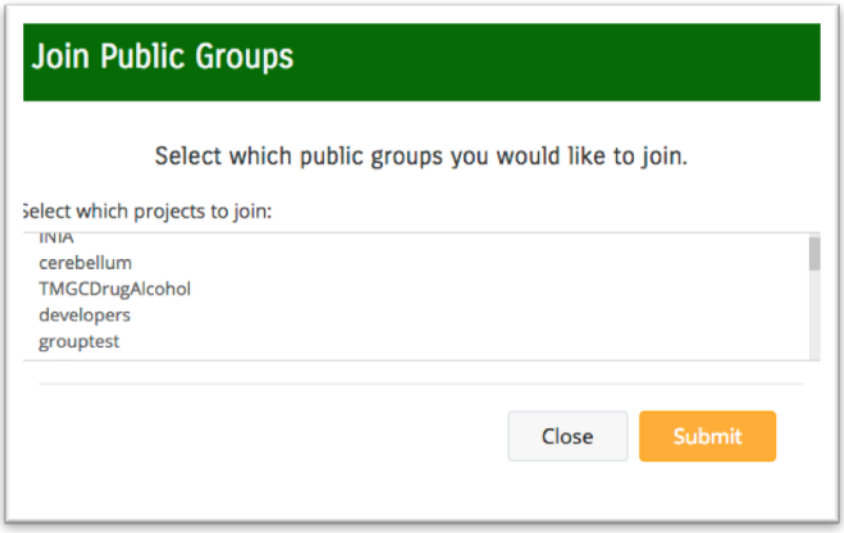
On your accounts page find the **Manage Groups** section. Here you can select the appropriate icons to:

- Create/Delete Group
- Edit / Add / Delete Members
- Email Members
- View Members
- View Curation Tasks
- Set Group Administration Options

Your group can be private, only the members you choose can use it, or public to all.



By selecting the **Join Public Group** icon at the bottom of this section, a modal will be displayed allowing you to join one of many publically available groups.



The accounts page is where you can:

- Change your Password

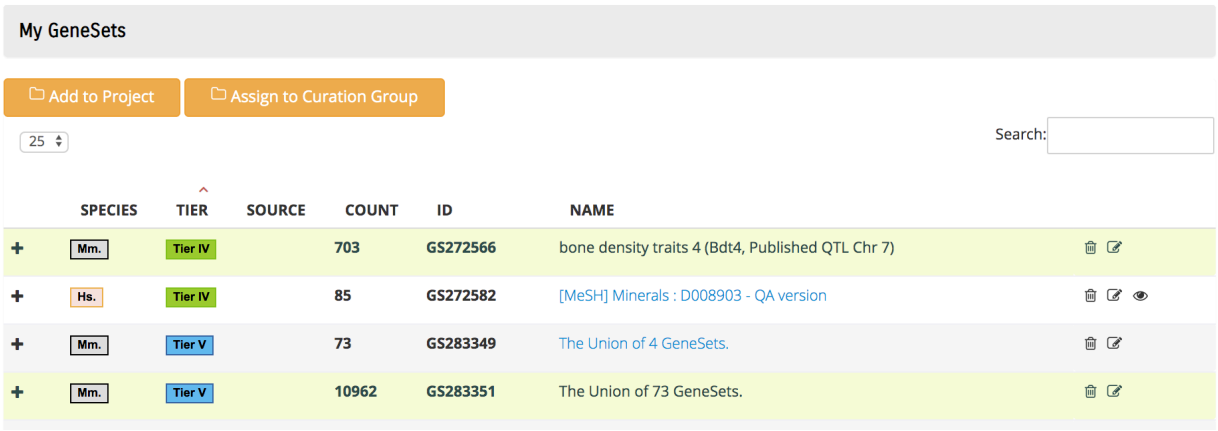
- Choose to receive email notifications
- Generate an API Key
- Elect a Text Annotator (Monarch, NCBO or Both)
- Edit your name or email address

Projects

You must add a gene set to a project in order to be able to select it for analysis. Projects consist of one or more gene sets. A gene set may be in many projects. A project may be associated with private and/or public groups.

Projects can be created or selected in several places:

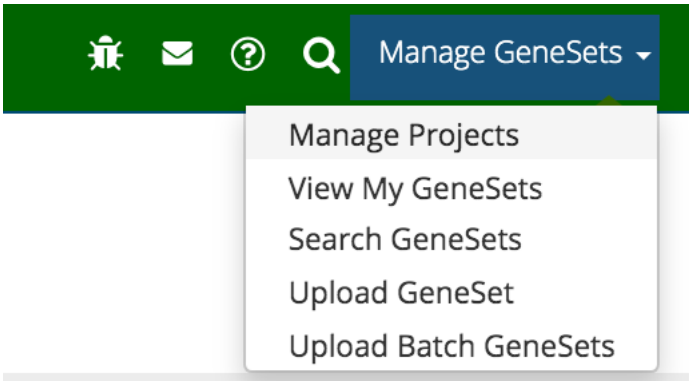
- Immediately after uploading a gene set
- Search results page
- My Projects page
- My GeneSets page



When using the My GeneSets page, click on each desired GeneSet, which is then highlighted. Then click the Add to Project button.

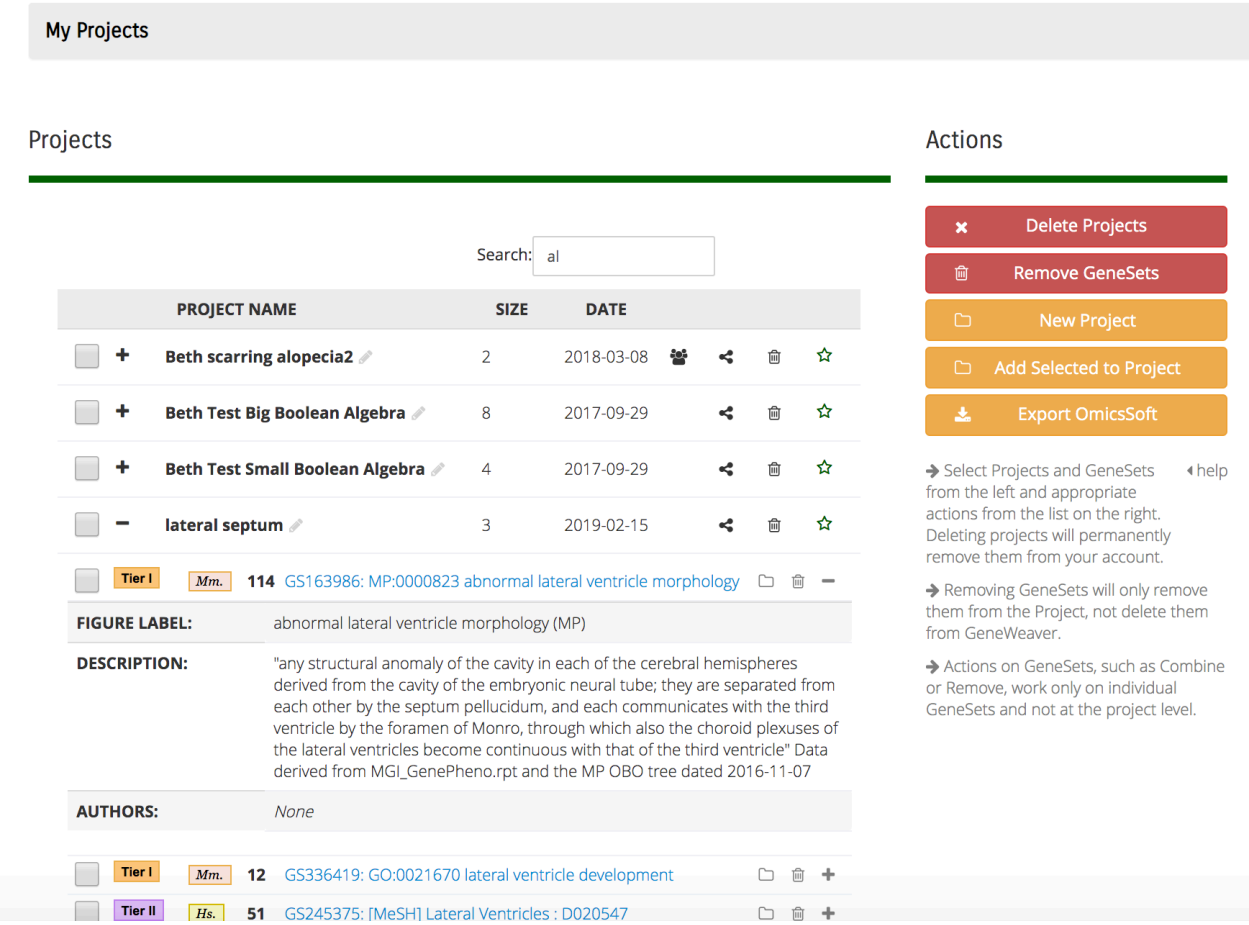
My Projects Page

You can get to the My Projects Page from the navigation bar or footer (under Manage GeneSets) or from the icon in the center of the home page.



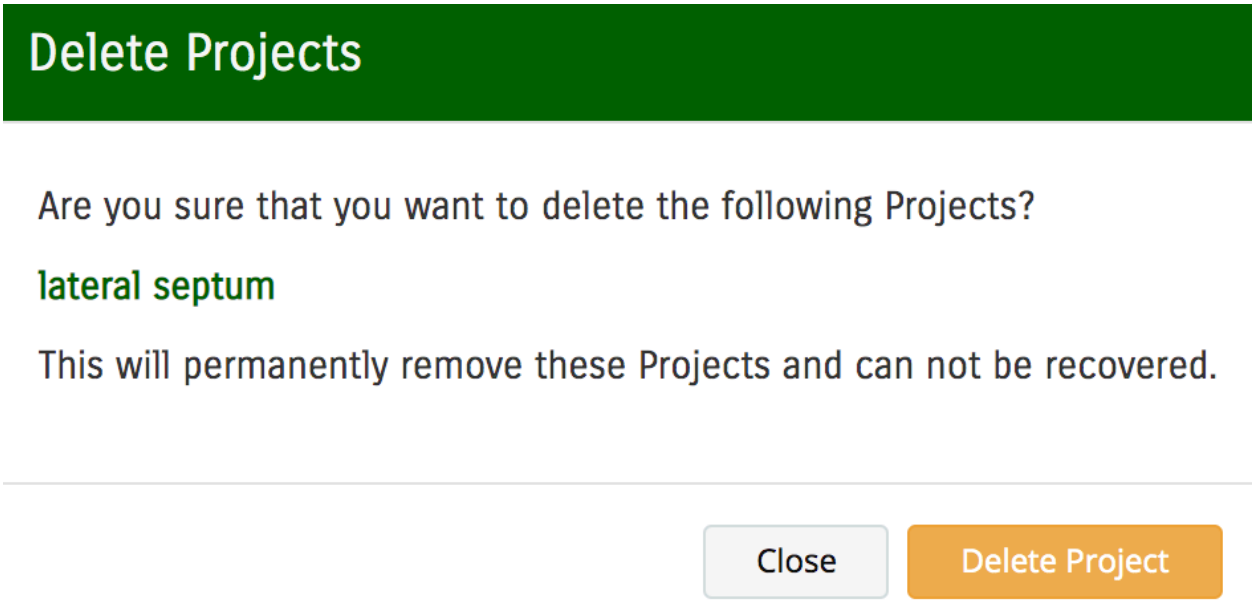
Use the search box to limit the list of projects by entering text that is included in the project name(s).

Use the + on the left side of the project to show all its Gene Sets. Use the + on the right side of the Gene Set to see its figure label, description, and authors.

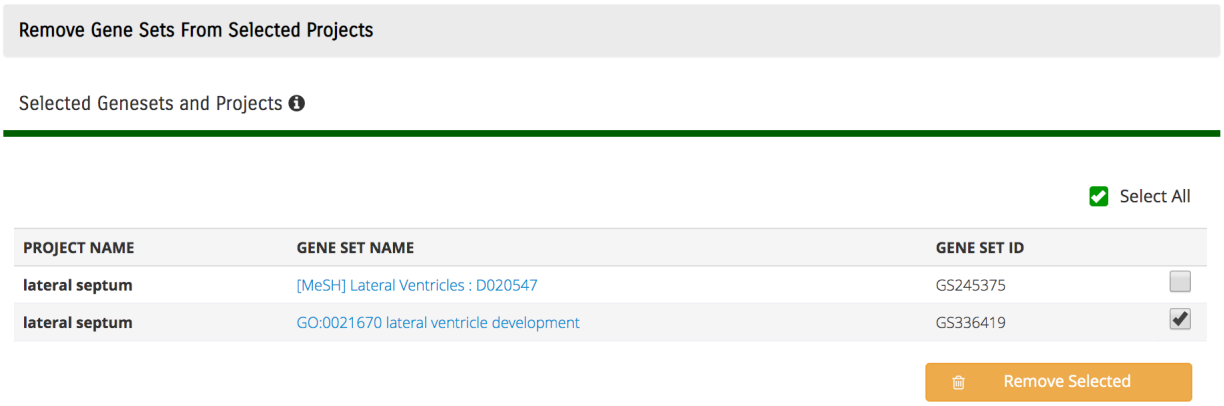


On this page you can use the action buttons on the right side of the page to:

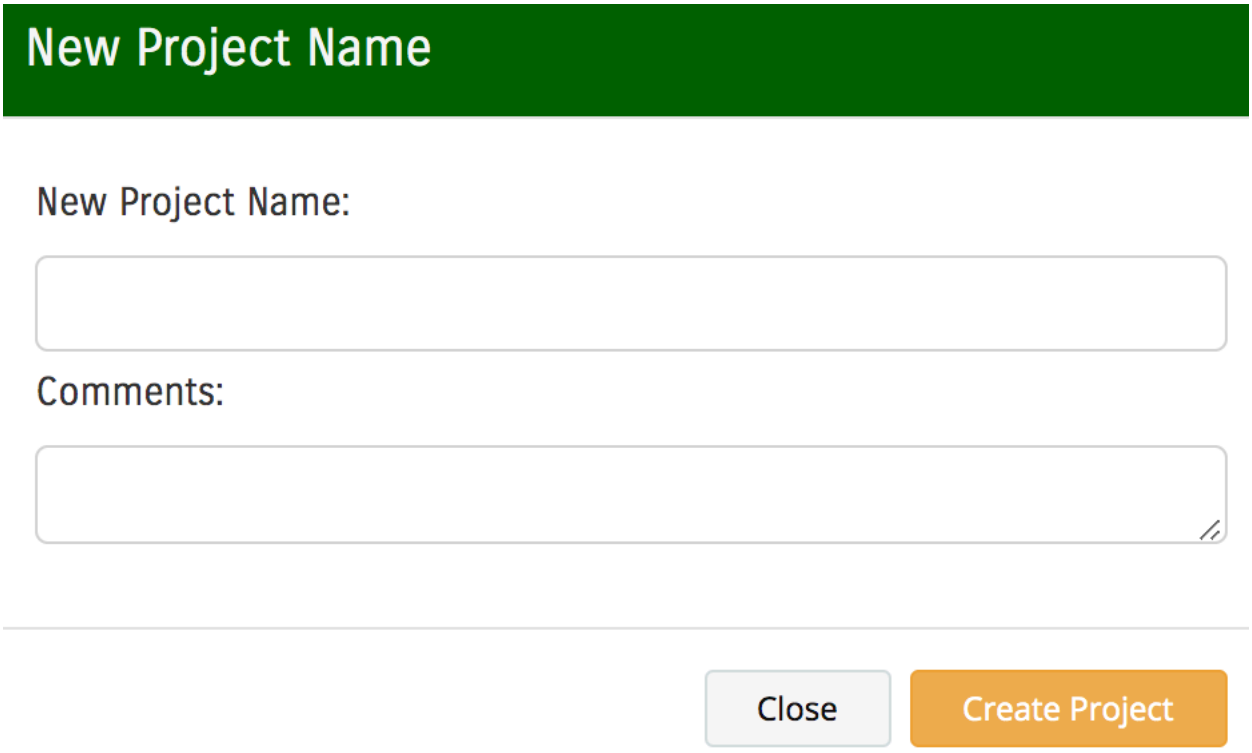
- Delete Projects



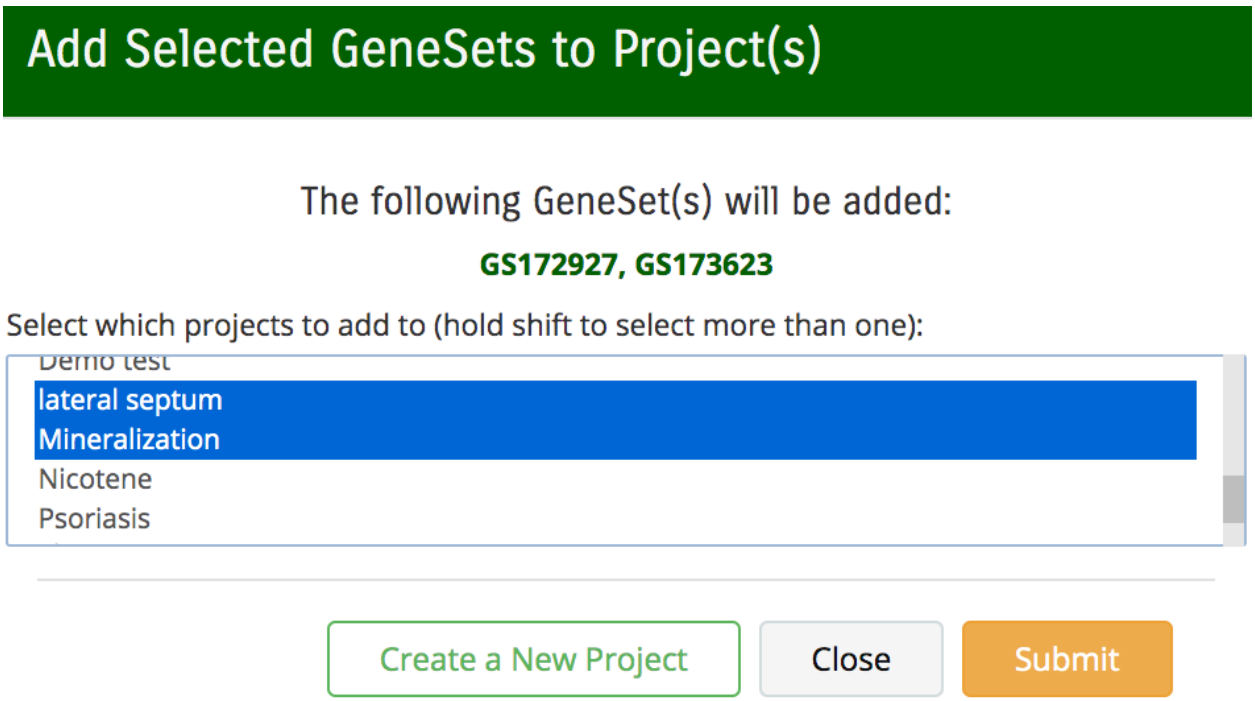
- Remove Gene Sets from a Project



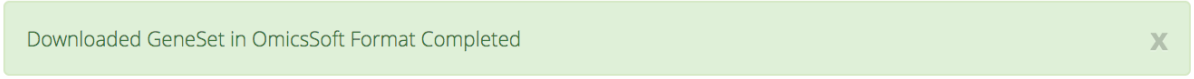
- Create a new Project



- Add Gene Sets to Another Project



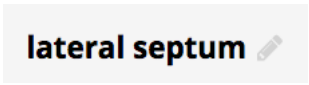
- Export to OmicsSoft



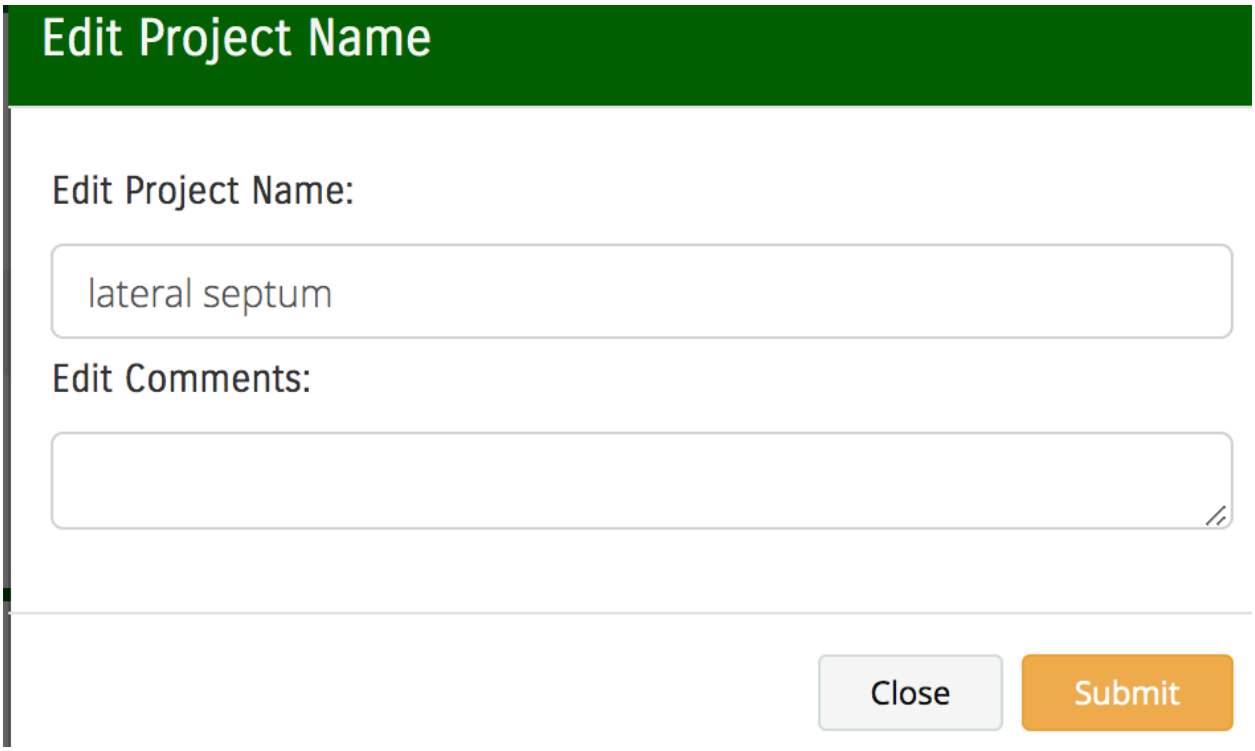
The projects are listed in table rows that show each project's name, size (number of Gene Sets), and creation date. There are several icons that can be used for specific functions:

- Expand/Contract the Project's List of Gene Sets

- Edit the Project Name

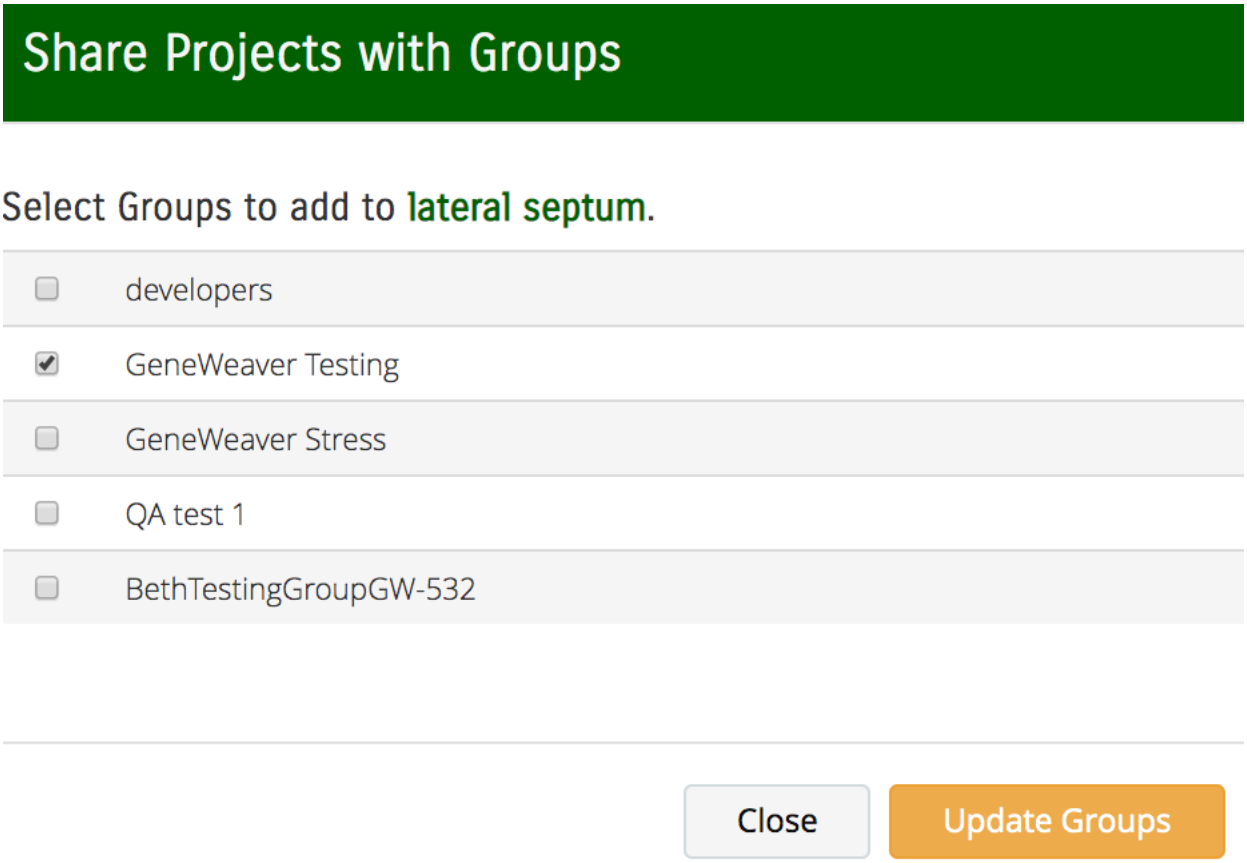


Clicking on the pencil icon will open the Edit Project Name dialog box.

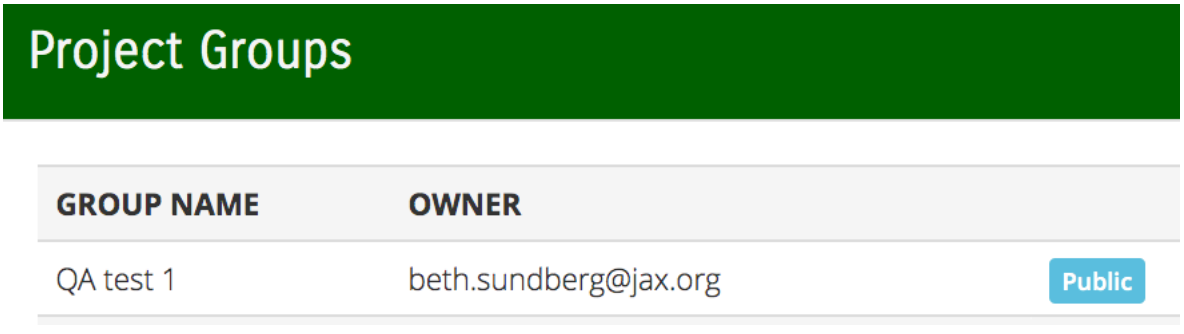


- Share a Project with a Group

Clicking this icon will open the Share dialog box where you can select multiple groups.



- View Groups the Project is Shared With



- The Star Icon is used to Mark Rows of Special Interest

Each Gene Set row includes icons for these functions:

- Add Gene Set to Projects (folder icon)
- Remove Gene Set from Project (trash icon)
- More Information (+ / -)

Clicking the link on the Gene Set name will take you to the **Gene Set Details** page.

Next Step:

Once you have your project(s) in order, go to the **Analysis** page.

Top ↑

Curation

Controlling the quality and validity of the large-scale analysis of secondary data requires the enforcement of interpretable standards for gene set construction and description. GeneWeaver's use of discrete analysis eliminates many barriers to the integration of heterogeneous data sets across species and experiments. However, it is important for users to be able to rapidly interpret the nature of gene sets retrieved from the site, requiring a minimal standard for metadata associated with secondary data. For this purpose, both unstructured textual descriptions of the data and structured ontology annotations to the terms in these descriptions are used to define gene sets.

Our **Curation Standards** provide detailed guidance to GeneWeaver curation policies and sample curation types. We have also included a brief explanation of the **Curation Process**, which includes a guide to our *new* curation interface.

Curation Standards Documentation

Secondary functional genomics data consists of the results of analyzed experiments in functional genomics. In contrast to primary data stores such as Gene Expression Omnibus (GEO) in which raw experimental data are stored, a secondary data store attempts to collect the results of experimental design and decision making process of the researcher so that one may interpret and integrate the gene set centered outcomes of the studies. Controlling the quality and validity of the large-scale analysis of secondary data requires the enforcement of interpretable standards for gene set construction and description. GeneWeaver's use of discrete analysis eliminates many barriers to the integration of heterogeneous data sets across species and experiments. However, it is important for users to be able to rapidly interpret the nature of gene sets retrieved from the site, requiring a minimal standard for metadata associated with secondary data. For this purpose, both unstructured textual descriptions of the data and structured ontology annotations to the terms in these descriptions are used to define gene sets. In the interest of encouraging submission we are cautious not to be too prescriptive or burdensome to users, but rather to provide guidelines on standards used by internal curators to assess data quality and clarity to enable rapid acceptance of community submissions to the data repository.

Curation Tiers

Tier Name	Curator Description
Tier I	**Public Resource Grade** Resource GeneWeaver Large data sets primarily curated by their parent resource. GeneWeaver ensures consistency of metadata (gene annotations to KEGG, MP and GO, curated functional associations in the Neuroinformatics Framework, Comparative Toxicogenomics Database)
Tier II	**Machine-Generated from public sources** GeneWeaver Gene sets resulting from genome analysis, not otherwise published in total, e.g. gene co-expression to behavior from GeneNetwork.org, QTL positional candidates from MGI. GeneWeaver curators examine data and metadata.

Tier Name	Curator Description
Tier III	**Human-Curated** GeneWeaver Curated user-deposited data and publication supplements in domains of interest.
Tier IV	**Submitted to Public- Provisional User** User-deposited data made available to the public. All Tier IV is examined for promotion to Tier III
Tier V	**Private User and Group data- Uncurated** User Data sets deposited for private or group-only analysis

Tier Name	Tier Description
Tier I Public Resource Data	Tier I data are professionally curated into another major database and are imported into GeneWeaver,which ensures consistency of metadata. Resource grade data is updated on a six-month cycle. These include: gene annotations to KEGG, MP and GO, curated functional associations in Neuroinformatics Framework, and Comparative Toxicogenomics Database.
Tier II Machine-Generated from public sources	Tier II data are computationally generated from data in public sources. These include empirical data obtained from public sources and their associated analytical tools, e.g. bulk analysis of gene co-expression to phenotypes across mouse strains from GeneNetwork.org, or QTL positional candidates from MGI. In contrast to Tier I in which the individual gene annotations to function are manually curated, Tier II includes machine generated gene annotations to functions from curated experimental data. GeneWeaver curators examine data and metadata.
Tier III Human-Curated Data	Tier III data are directly entered or reviewed by a professional curator for redundancy with existing records and adherence to documentation standards. Users who submit data under Tier IV have the option of sharing their data to the public. These data will be marked provisional until reviewed by the curator for data entry errors, compliance to metadata standards and redundancy with existing data. The submitter of the data will have the opportunity to approve the curators modifications to them prior to upgrade to Tier III status. For some research areas, a professional curator has identified and entered gene expression, quantitative trait locus and genomewide association studies (GWAS). Where possible, the curator has obtained results directly from the study authors, supplements or data repositories such as GEO, in addition to the often highly-filtered set of results reported in publications.
Tier IV Submitted to Public-Provisional	Tier IV consists of user submitted data that has been shared to the public prior to review. This data is indicated as provisional, but can be used in all analyses. Curatorial review is required to remove the provisional label.
Tier V Private User and Group Data, Uncurated	Data in user accounts that is assigned private or group level access is confidential, is not exposed to analyses by users outside of the group to whom it is shared, and is therefore not reviewed by the professional curator.

General Definitions

Gene Set Name: A brief title for the gene set, approximately sentence length, that should provide a clear and concise description of the contents of a gene set interpretable to most users of GeneWeaver, but with sufficient detail to satisfy a domain expert. This is the major gene set name that is displayed in all search results, project directory and table views of analysis results. Standards for specific gene set types are given in the following section.

Gene Set Figure Label: A brief 23 character abbreviation to facilitate recognition of the gene set in a graph or other display.

Gene Set Description: A detailed description of the gene set, including rules for its construction, experimental methods and analyses used to generate data, anatomical terms, and traceable references to source data including accession information and date. Abbreviations should be avoided.

Ontology Annotations: Relevant terms from Disease Ontology, Mammalian Ontology and other OBO ontologies supplied by curators or identified through the application of the NCBO Annotator to textual descriptions including publication abstracts.

Publication Information: PubMed ID, title, authors, publication information and full-text of the abstract.

Standards for Common Gene Set Types

Type of Data: Differential Expression Profiling

Gene Set Name: Genes [upregulated/downregulated/differentially expressed] in [tissue] [comparison]. ***Example***: Genes differentially expressed in striatum of C57Bl/6J compared to C57Bl/6C. Note: spell out anatomical terms as nouns, e.g. striatum, not striatal. Include complete strain names, e.g. C57BL/6J not B6.

Gene Set Figure Label: B6JvsB6CStriatum

Gene Set Description: Indicate which samples were compared. What experimental manipulations or tissue differences are being examined? Indicate statistical methodology, significance thresholds and which changes are reported here. Indicate if uploaded p-value, q-value, effect size or fold change and fold change reference. ***Example***: Striatum gene expression differences between naive C57BL/6J and C57BL/6C substrains corresponding to a 5% FDR. A small number of genes are highly differentially expressed between B6 substrains, C57BL/6J (high alcohol consumption preference) and C57BL/6C (low alcohol consumption preference). Fold expression change are relative to B6/J.

Gene Set Contents: Gene identifier and statistical score for differential expression, e.g. p-value, q-value, correlation coefficient, binary score, effect size or fold change.

Type of Data: Published QTL Candidate Gene List

Gene Set Name: Description (name, Published QT Chr # MGI:#). ***Example***: cocaine related behavior 10 (Cocrb10, Published QTL Chr #)

Gene Set Figure Label: (QTL-name-Organism-Chr #). ***Example***: QTL-Cocrb10-Mouse-Chr 9

Gene Set Description: QTL Name Definition, candidate gene selection method (e.g. 1.5 LOD drop; inter-marker interval). Exact description of phenotype. Strains used for mapping should be included. ***Example***: Rats were subjected to a forced swim test (FST) procedure in which they are placed in water for 5 min, and their behavior was scored every 5 sec as immobility, climbing, or swimming. Data were analyzed for each activity with consideration given to their non-independence. p-value:0.0002, Variance: 3.6, Peak Marker: D5Rat40 (BLAT 16538053) Spans 1-41538053. This interval was obtained by using a fixed interval width of 25 Mbp around the peak marker. Strains were WKY/NHsd and F344/NHsd. Also defined as Imm3.

Gene Set Contents: Gene identifier and binary score.

Type of Data: Co-Expression to Phenotype

Gene Set Name: Describe tissue and phenotype correlated. ***Example***: Cerebellum gene expression correlates of acetic acid writhing behavior in BXD recombinant inbred mice.

Gene Set Figure Label: Co-expression writhing

Gene Set Description: Indicate what the comparison was that was made and any statistical cut-offs that were used. ***Example***: Cerebellum gene co-expression with acetic acid writhing in BXD RI mice. Gene expression data was obtained from genenetwork.org SJUT Cerebellum mRNA M430 (Mar05) RMA data set. Behavioral phenotype data was collected by RMQ and consisted of the number of writhes in response to 0.6% acetic acid i.p.

Gene Set Contents: Gene identifier and statistical score for co-expression. e.g. R-squared, p-value, q-value, binary threshold.

Type of Data: Reference Ontology

Gene Set Name: Term # and name. ***Example***: MP:XXXXXXX Abnormal.

Gene Set Figure Label: Term #. ***Example***: Term #

Gene Set Description: Term Definition. ***Example***: "Increase in the dose or concentration of a foreign compound required to induce a specific level of response" www.informatics.jax.org, 2010-12-01

Gene Set Contents: All gene sets include genes, mutant alleles or gene products annotated to an ontology term by a professional curator. Each gene directly annotated to the term is given a score of 1, each gene connected to a term through annotations to its higher order parents is given a score of 2. To use only direct annotations in an analysis assign a threshold of < 2 to each Gene Set.

Type of Data: Co-Expression Clusters

Gene Set Name: Co-Expression clusters. ***Example***: Co-expression cluster of nicotine Dependence genes significantly expressed in the adolescent PFC, VS and Hippocampus.

Gene Set Figure Label: Abbreviated description. ***Example***: Adolesc Rat Nic Dependence

Gene Set Description: Indicate what samples were compared and what was clustered. ***Example***: Studies analyzing brain samples from female rats that had been injected with nicotine at four different ages show that nicotine exerts the greatest influence during adolescence. Using DNA microarrays, gene expression correlates were obtained from the prefrontal cortex (PFC), ventral striatum (VS), and hippocampus. Principal cluster analysis was then used to identify 76 genes that changed significantly in at least one of these three brain regions during the experiment.

Gene Set Contents: Gene identifier and statistical score for cluster analysis or binary threshold.

Type of Data: Genome Wide Association Study Gene Set Name: GWAS of … ***Example***: GWAS of Alcohol and Nicotine Dependence in Australian DNA-Pools.

Gene Set Figure Label: Abbreviated description. ***Example***: GWAS Alcohol Nicotine

Gene Set Description: List of positional candidate genes after correcting for multiple testing and controlling the false discovery rate from genome wide association study. Represents genes associated with a linked cytological region or genes 'near' an associated SNP. ***Example***: Genome-wide association study identifies a locus at 7p15.2 associated with endometriosis.

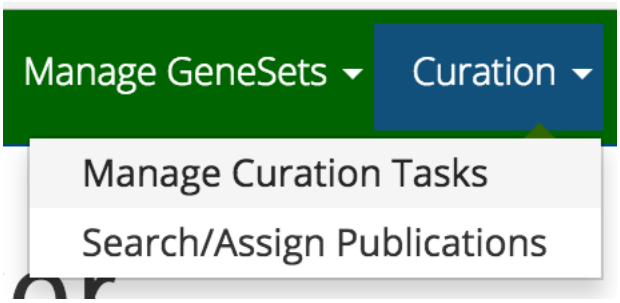
Gene Set Contents: Gene identifier and binary threshold.

ToDo

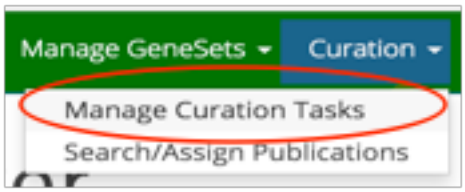
Top ↑

Curation Guide

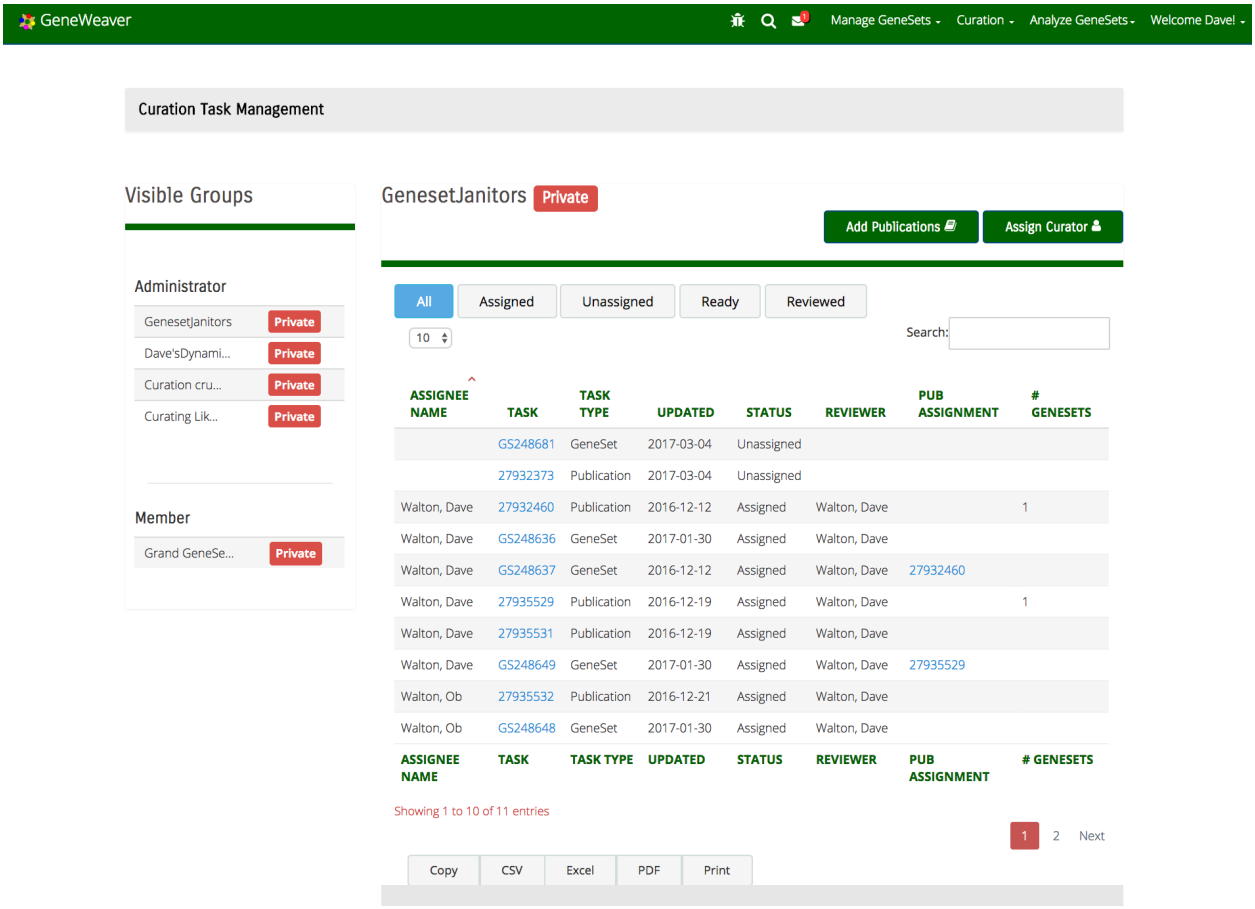
The Curation menu in GeneWeaver provides options for managing curation tasks and searching and assigning publications



Managing Curation Tasks

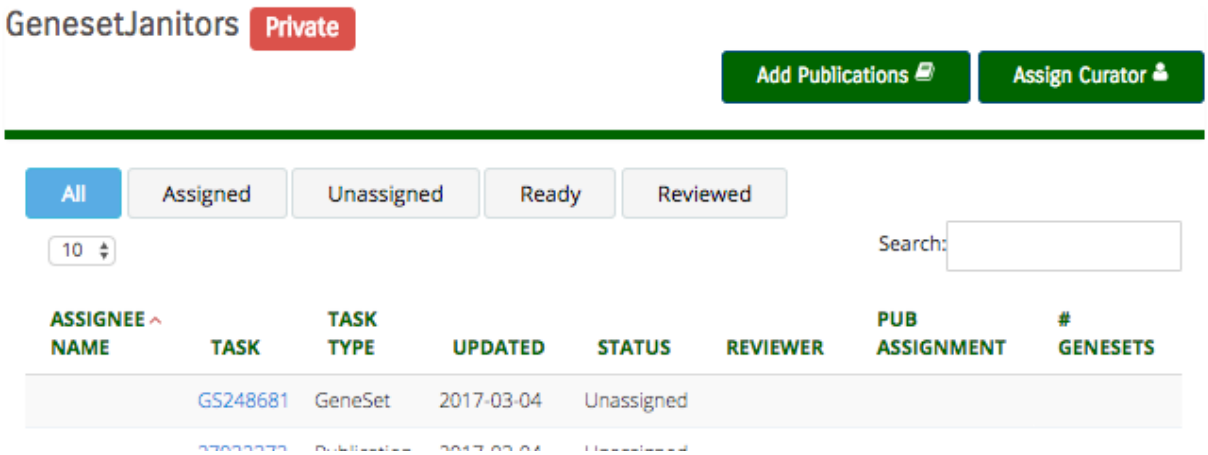


When selecting "Manage Curation Tasks" from the navigation menu you'll be presented with a page containing in the side bar, all of the curation groups you belong to separated by groups you administer and groups of which you are just a member. The main body of the page will contain the list of curation tasks for the selected group in the side bar. The curation tasks are a mix of publications and genesets, which have been assigned to this group, with the tasks, which have not yet been assigned to a curator, appearing at the top of the table.



You can change the selected group in the main part of the page just by clicking on the group of interest in the side bar.

Immediately above the table, there are buttons which will allow you to filter the contents of the table to contain: **All** results, **Assigned** tasks, **Unassigned** tasks, tasks which are **Ready** for review and tasks which have been **Reviewed**. In this context *Assigned* and *Unassigned* are referring to curator assignment.



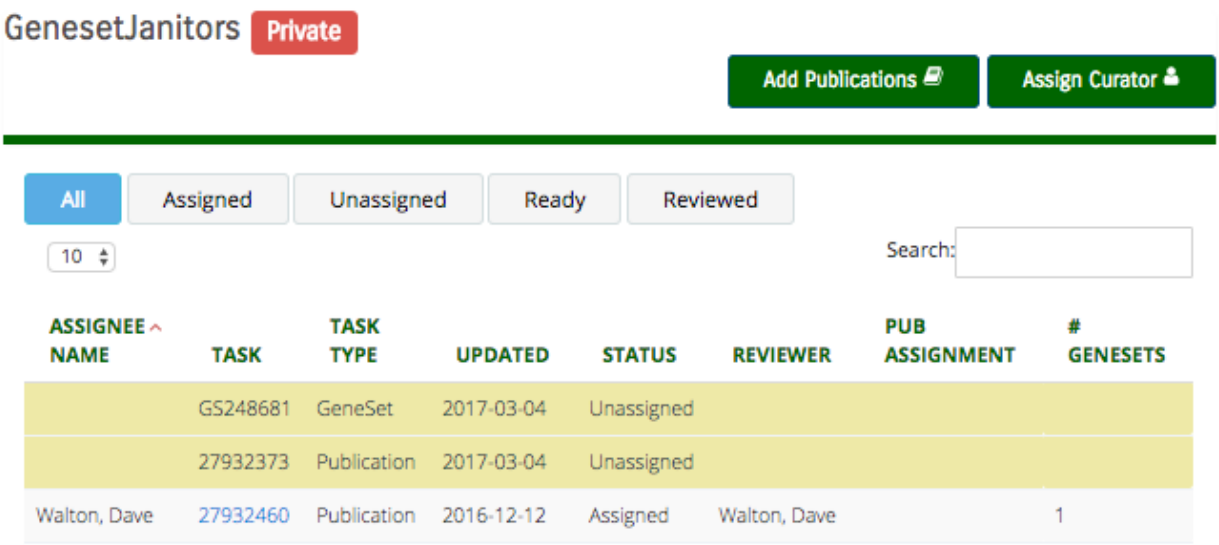
The columns of the table are mostly self-explanatory, however it's worth explaining PUB ASSIGNMENT and # GENESETS.

The PUB ASSIGNMENT column will display the associated PubMed ID for a geneset task, when it was

entered via an association when a **Publication Assignment**. The link on the PubMed ID will take you to the publication assignments page.

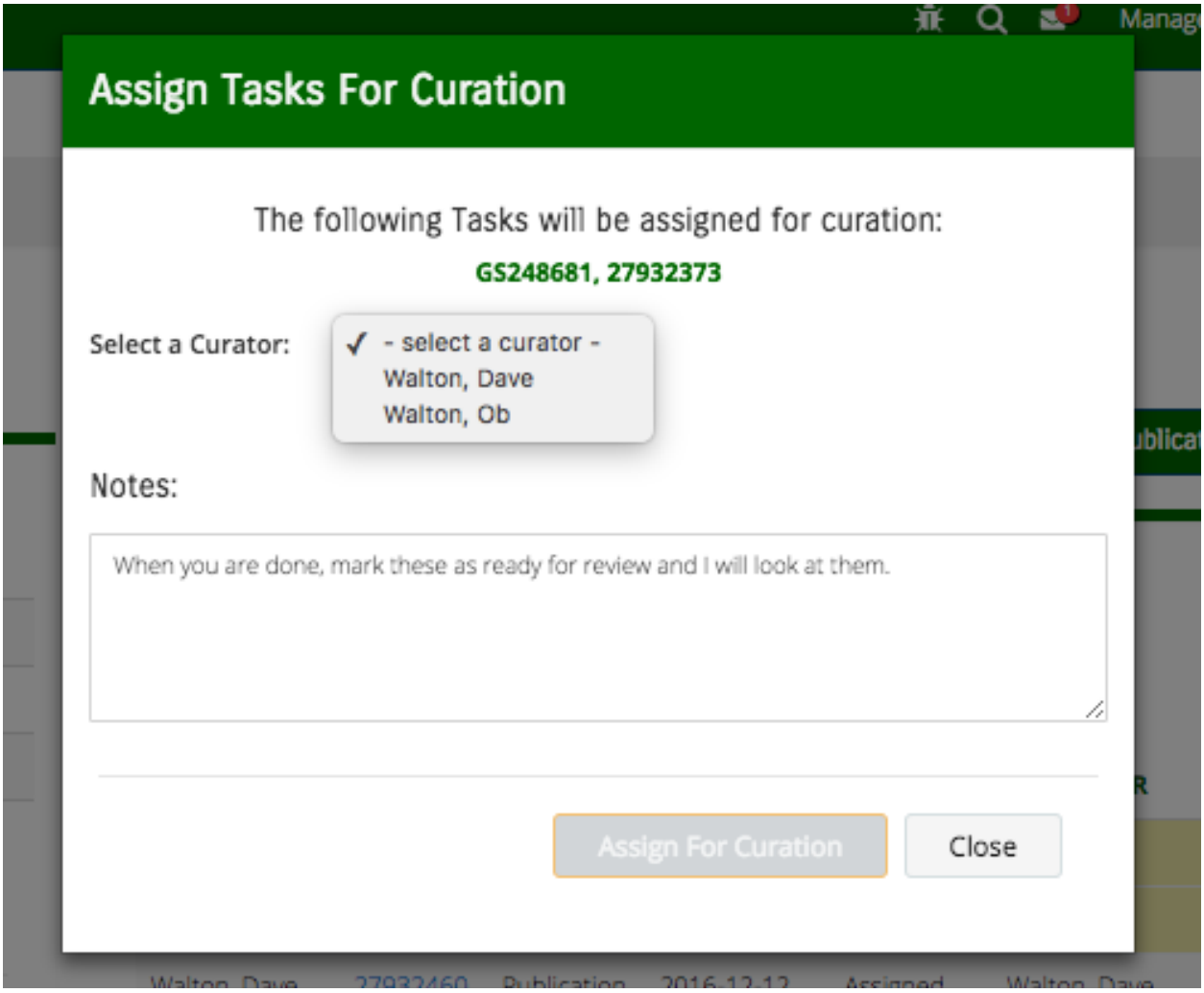
The # GENESETS column indicates for a publication, how many genesets are associated with it as part of this specific publication assignment. If this publication is assigned to another curation group as well, genesets as part of that publication Assignment will not be part of this number.

If you are an administrator of the curation group for which you are managing tasks, there should also be an **Assign Curator** button at the top right of the page. You are able to select one or more task rows in the table, at which point they should be highlighted yellow.



One note about how row selection works: There are no ***Shift*** or ***Control*** operations for selecting multiple rows. Rows are selected one at a time, and remain selected until you click on the row again, when it becomes deselected. Also, selections do not persist when you move to the next page of results. This latter issue is something we intend to address in a future release. However, for the time being it's recommended you select the visible rows you would like to assign, assign them, and then move onto the next page of results.

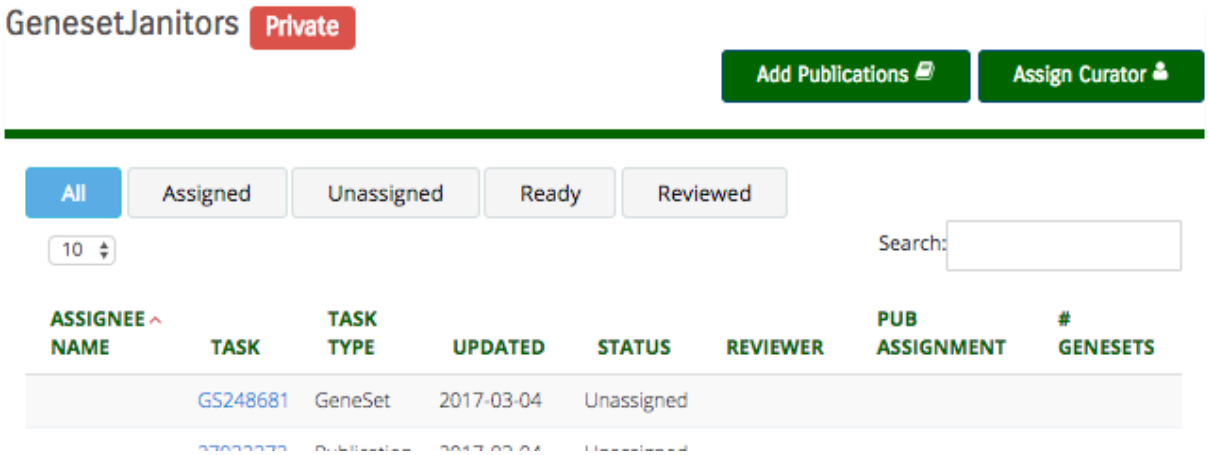
Once you've chosen the tasks you want to assign (or reassign), you will select the Assign Curator button.



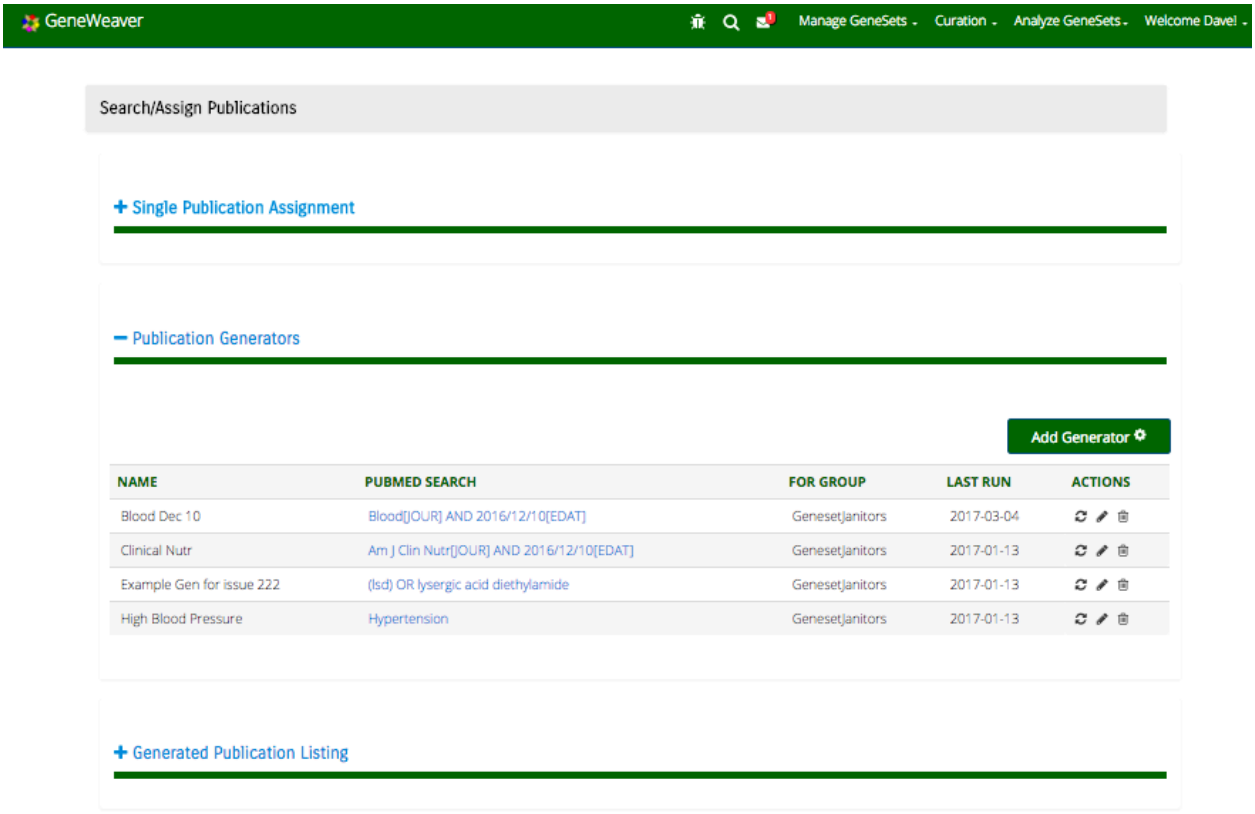
You will then be presented with a modal dialog box, where you can select the individual you wish to curate the tasks, and include a note regarding the curation assignment.

Once a curator has been selected, click the **Assign For Curation** button. If you select **Close** instead no assignment will be made.

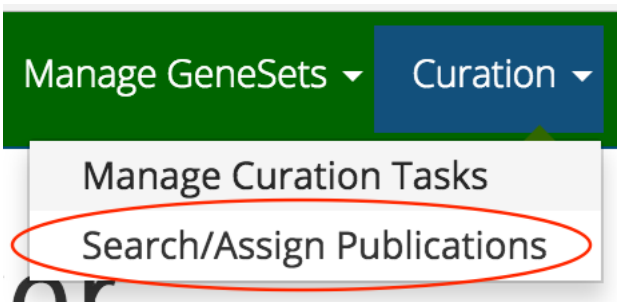
For your convenience, if you realized while in the Curation Task Management page that you want to assign a publication to this group, so that you can subsequently assign it to a curator, there is also an ***Add Publication*** button at the top of the page.



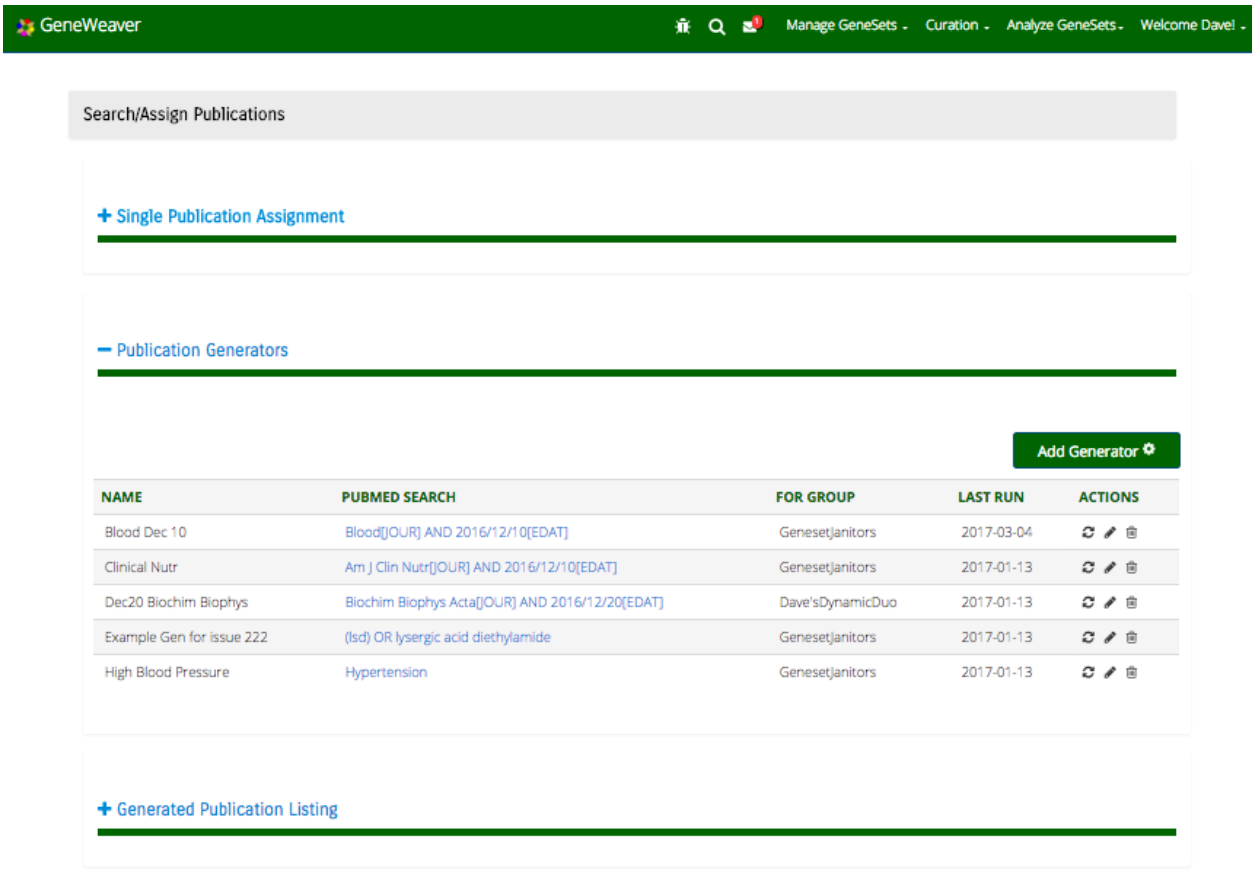
This button will take you to the **Search/Assign Publications** page with only publication generators listed that were created for the curation group.



Search/Assign Publications



When selecting "Search/Assign Publications" from the page menu you'll be presented with a page containing an "accordion" display, with the middle section opened by default. The assumption is that most times the user will be interested in generating a list of publications from which to make assignments.



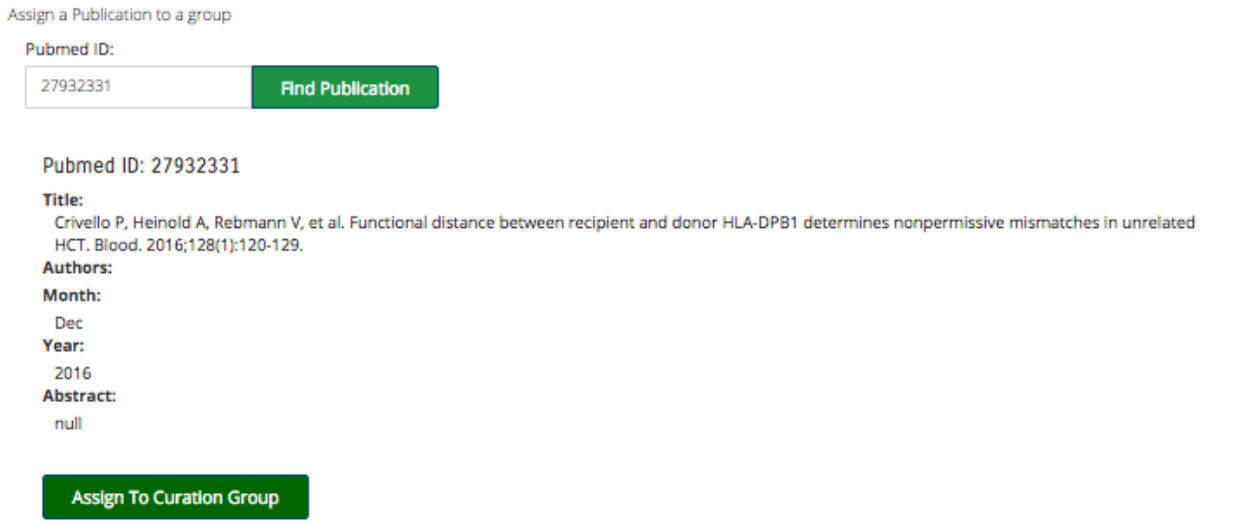
The section is broken into 3 parts:

1. Single Publication Assignment
2. Publication Generators
3. Generated Publication Listing

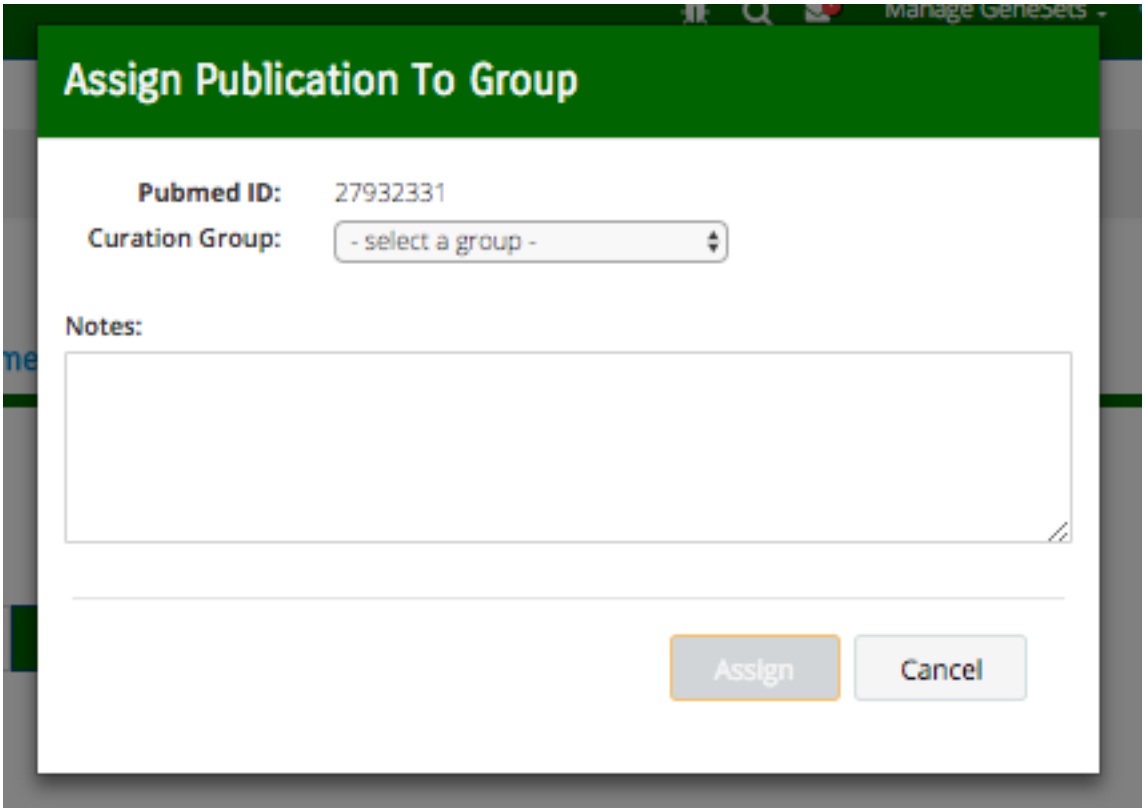
Single Publication Assignment

If you select the + symbol next to *Single Publication Assignment* you will be presented with a simple search box. This would be used in the case where you have a specific PubMed ID that you know and want to assign for curation. You simply enter the PubMed ID and select the **Find Publication** button.

Assuming you've entered a valid PubMed ID, the citation will be returned so that you can confirm that this is indeed your publication of interest.

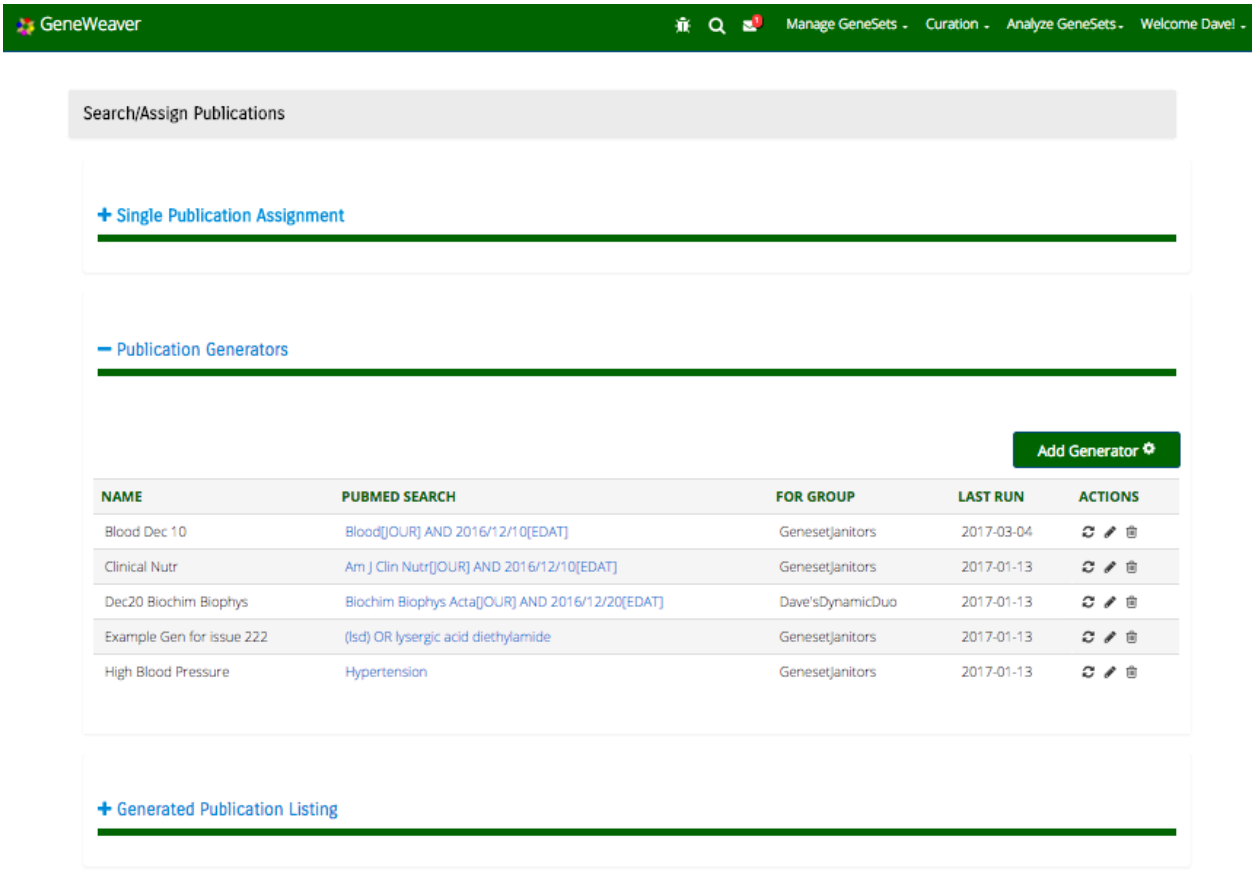


To assign the publication to a curation group to work on, just select the Assign To Curation Group button and you will be presented with the following modal dialog box displaying a drop down so you can select the curation group and a text box so that you can enter any curation notes you might have.



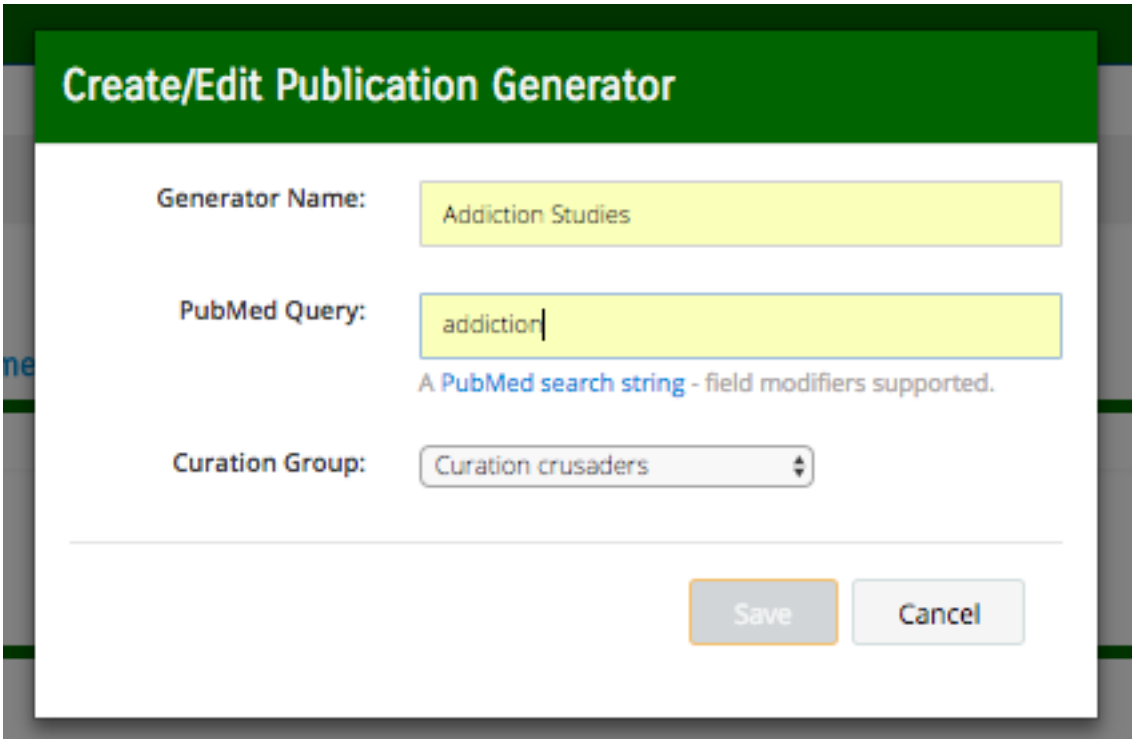
Publication Generation

If you select the + symbol next to *Publication Generators* you will be presented with a table of generators that have been created for groups of which you are a member, and an **Add Generator** button.

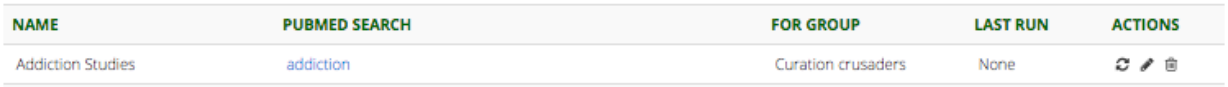


The columns of the table represent: the NAME that was assigned to the generator when it was queried; the PUBMED SEARCH term that is used to search PubMed and bring back a list of publications; FOR GROUP which is the curation group for which the generator was created; the date the generator was LAST RUN; and a series of ACTIONS which can be executed on a generator (will discuss these later).

In the case where there are no generators already created for any of the groups to which you belong, the first step would be to click **Add Generator**. This will bring up a modal dialog box



You will be presented with three fields, which are all mandatory in order to have the **Save** button enabled. Generator Name is a self selected name to represent your generator. PubMed Query must be a valid PubMed search term. You can learn more about valid PubMed terms using the following YouTube video (<https://www.youtube.com/watch?v=dncRQ1cobdc&feature=relmfu>). There is also a link to the PubMed search string builder (<https://www.ncbi.nlm.nih.gov/pubmed/advanced>) directly in the dialog box.



Once created the generator becomes available in the table of generators.

Generator Actions

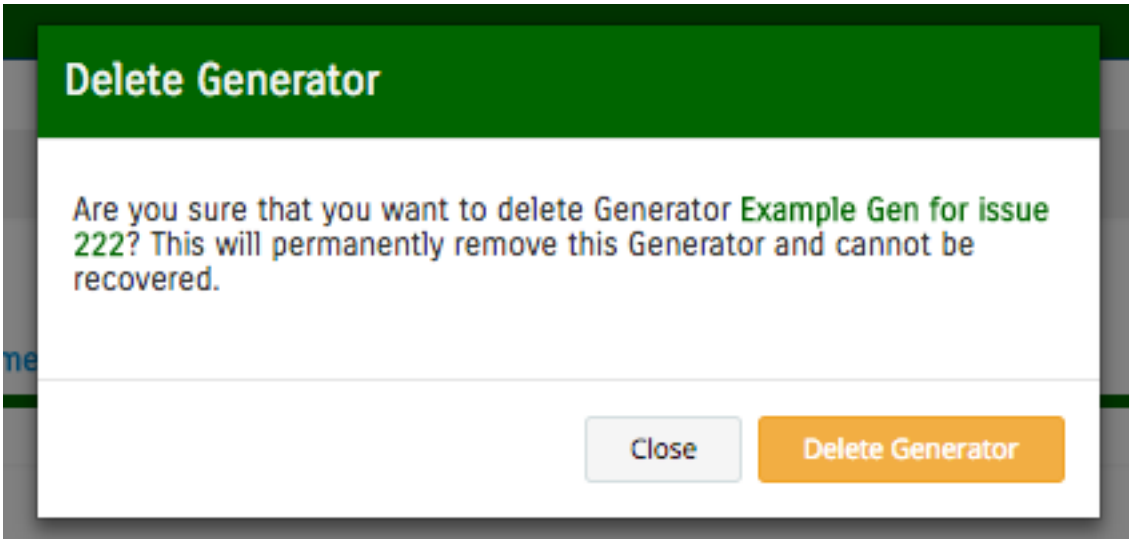
There are three actions available to be used with generators:

- Run
- Edit
- Delete

We'll discuss Run last as it's most involved and leads to the next section.

Edit is fairly straight forward. It presents you with a modal dialog identical to the one you get when creating a new generator. You are able to update any of name, search term or group association.

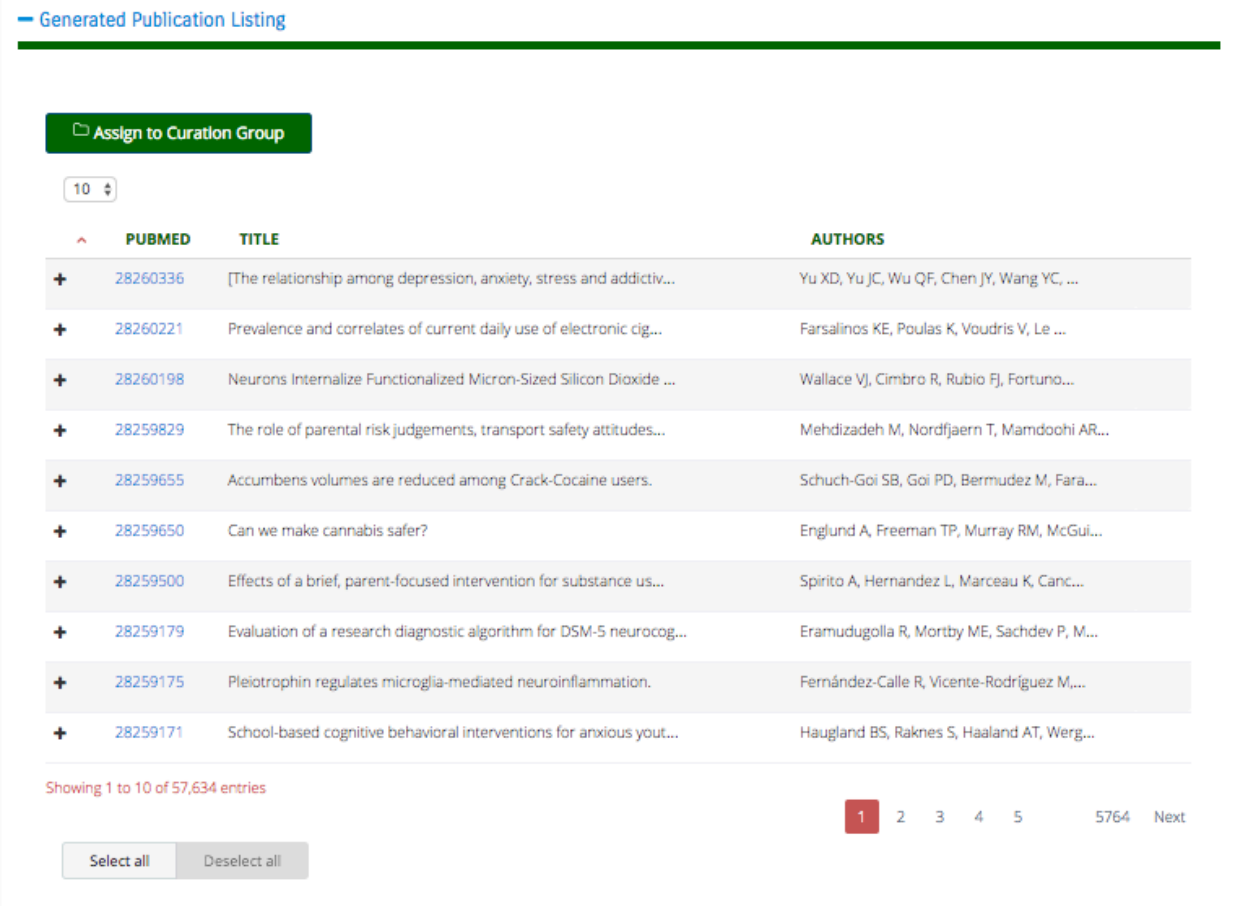
Delete will simply bring up a confirmation dialog box.



Lastly the Run option will cause the generator to run against PubMed, automatically collapse the **Publication Generators** accordion section and will expand the **Generated Publication Listing** section, with the results of the generator displayed.

Generated Publication Listing

If you select the + symbol next to *Generated Publication Listing* you will be presented with a table of publications that have been pulled from PubMed and are the result of the PubMed search term associated with a given generator. This section is populated by selecting the Run icon in the generator table.

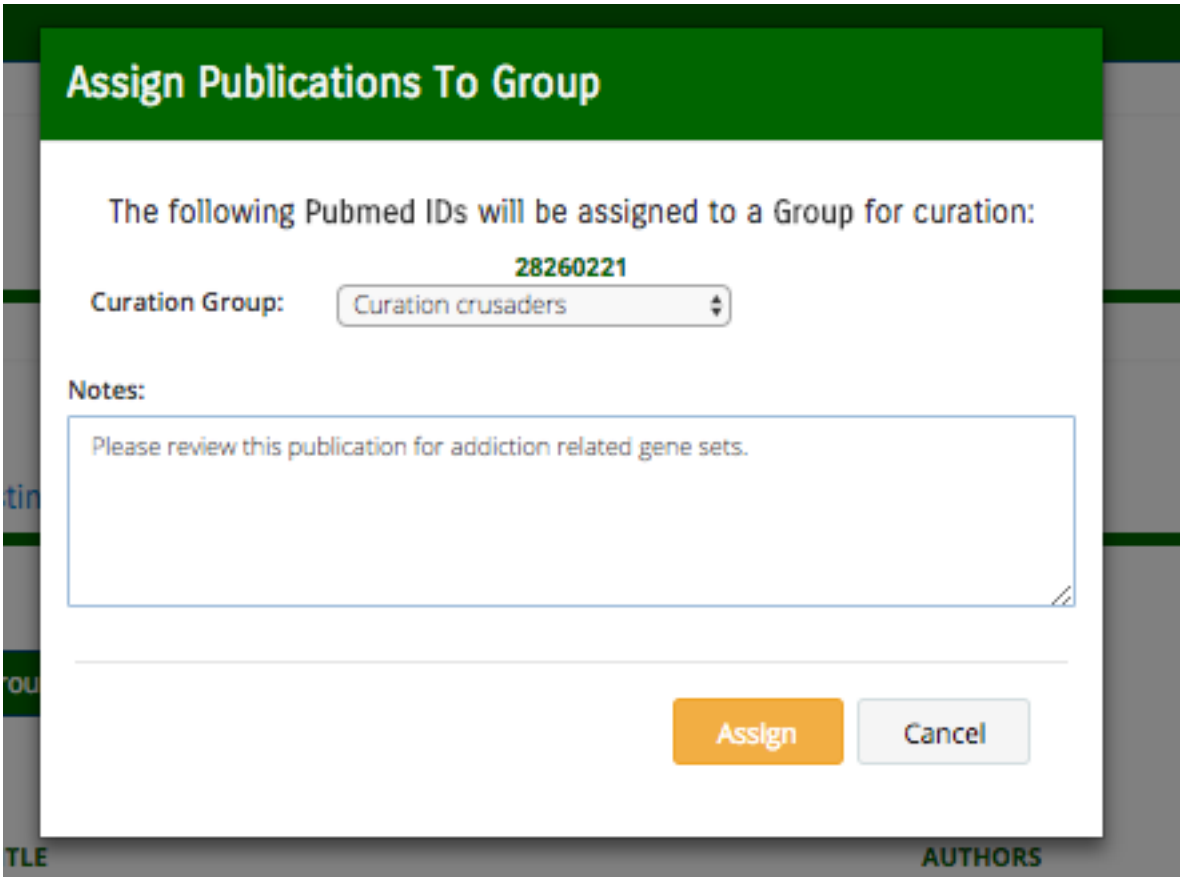


Publications that are pulled by a publication generator are not persisted in the GeneWeaver database. At least, not until the time they are assigned to a curation group. Instead the publications that are not already assigned to a group are pulled directly from PubMed at the time of generation. Some of these queries can result in a very large number of publications (hundreds of thousands). Therefore we only display a slice of the publications at a time. We do keep track of the total number that match the search term, and allow you to page through the results, each time going back out to PubMed to pull in the next set.

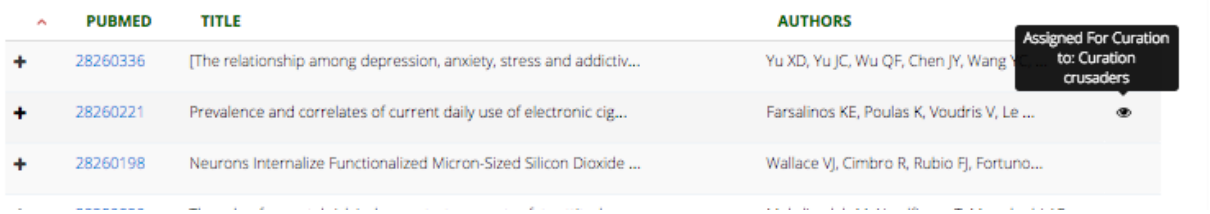
Similar to the Curation Task Management page, you can select multiple rows to be assigned to a curation group all at once. This is done by individually selecting each publication of interest. There are no features for multi select all at once using either the control or shift keys. The only way you can de-select a row, is by clicking the row again.

You can get more detail about a publication by clicking the + symbol at the beginning of the row. This will display the title, authors, journal and publication date, a link to the full text of the publication and the abstract.

Once you've selected the publication or publications that you would like to assign to a curation group, you select the **Assign to Curation Group** button. This will bring up a modal dialog box where you will select a curation group, and optionally type in a note regarding the curation that is to be done.

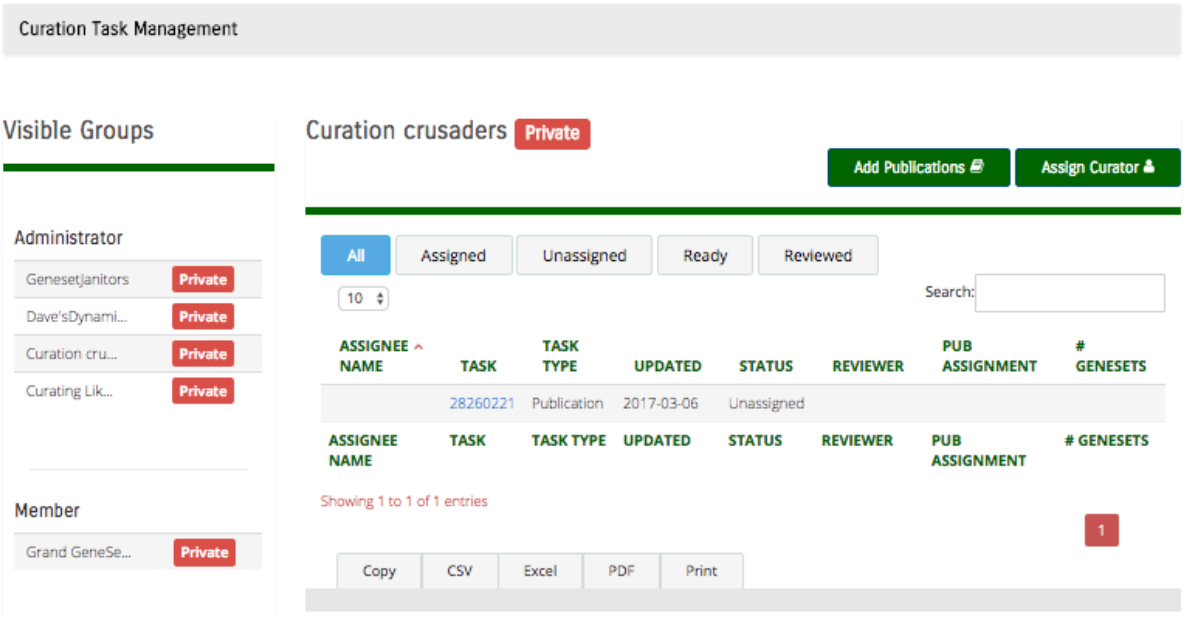


Once assigned the publications that have been assigned to a curation group should now have a View icon appearing at the end of the row, and if you cursor over the icon you will see a tool tip telling you what group or groups are curating this publication.



Also, if you select the + symbol at the beginning of the row now, the groups will be listed under **Assigned to Curation Groups** under the expanded details.

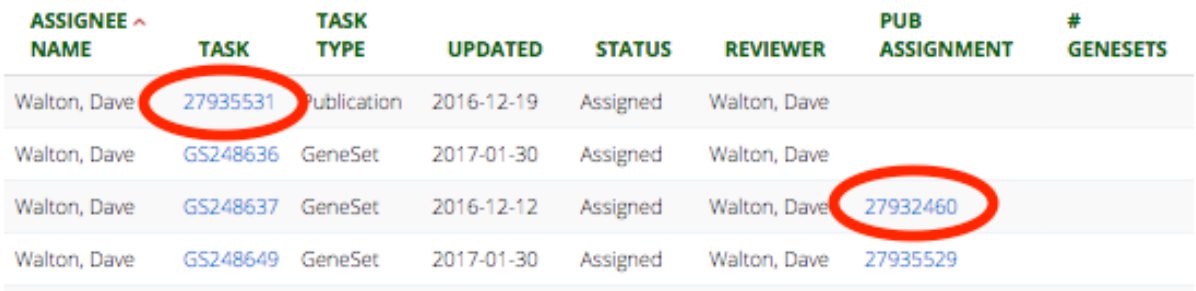
Once an assignment has been done a notification will be sent to the administrator of the curation group so they know that there is a new publication that needs to be assigned to a curator. Notifications will be discussed in another section. If you now return the the **Manage Curation Tasks** page for the curation group to which the publication has been assigned, you should now see the publication listed at the top of the tasks table.



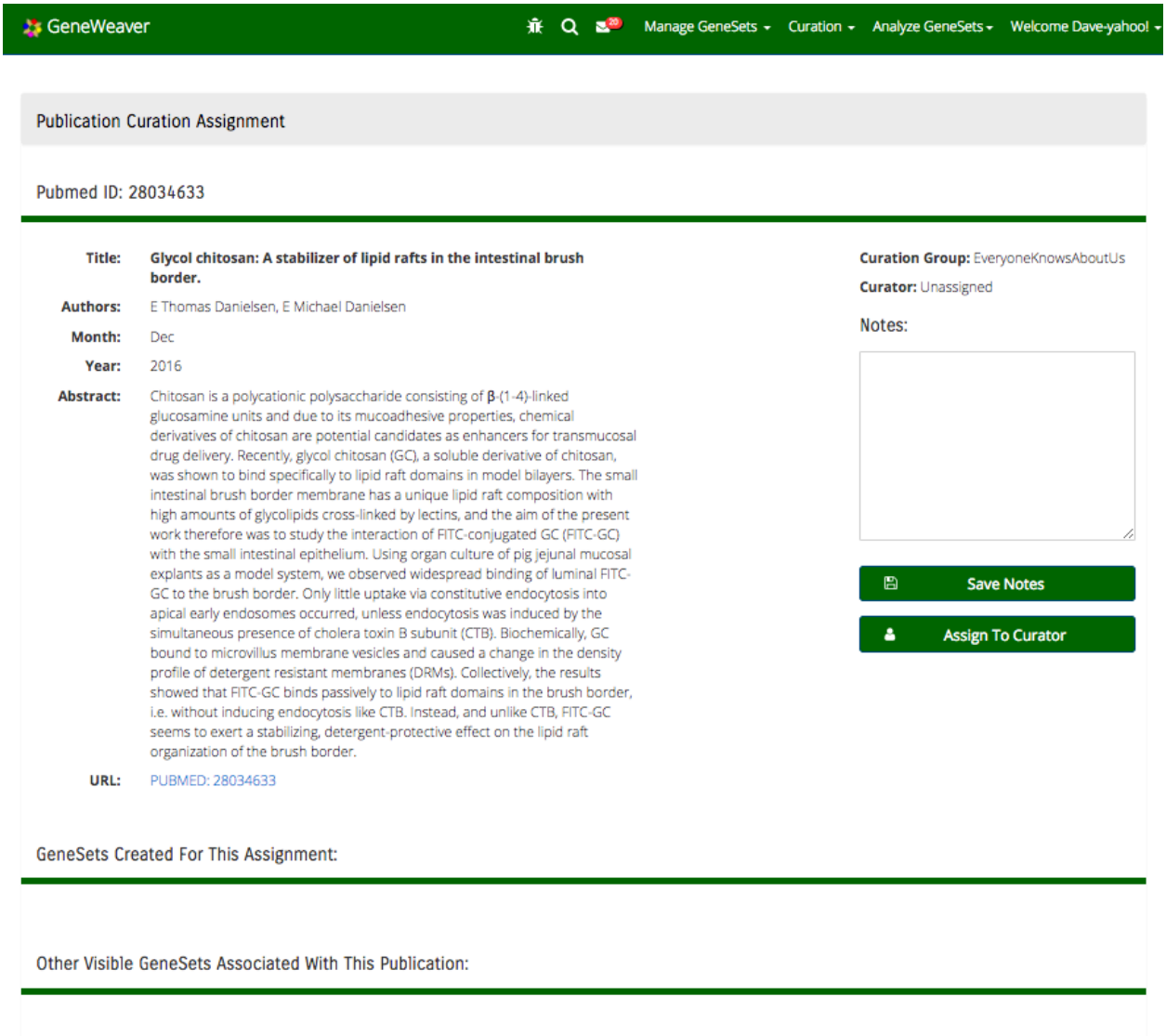
Publication Curation Assignment

You can get to the ***Publication Curation Assignment*** page from the **Curation Task Management** page in one of two ways.

- Click on the PubMed ID in the TASK column of a publication row of the task table.
- Click on the PubMed ID in the PUB ASSIGNMENT column of a geneset row of the task table.



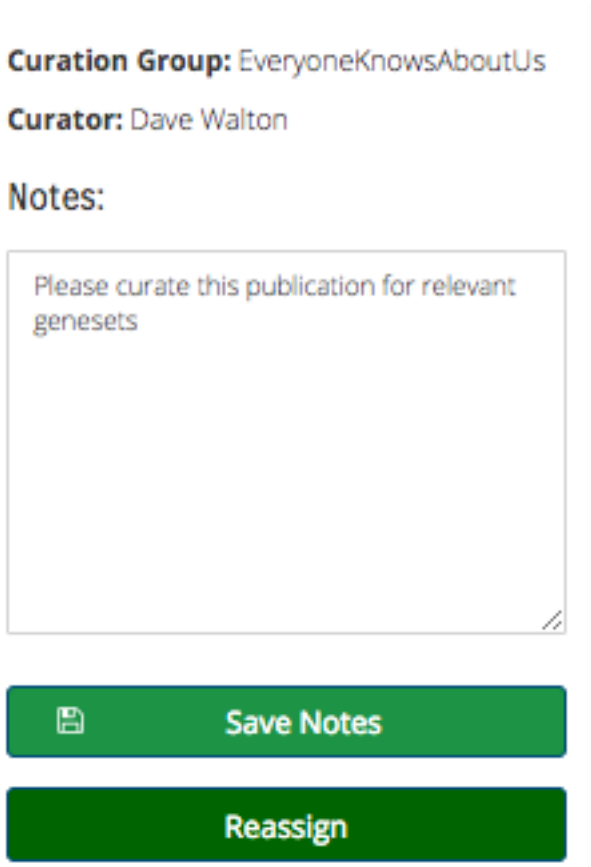
If you select a publication that has not been assigned to a curator yet, you'll get to a page that looks something like this:



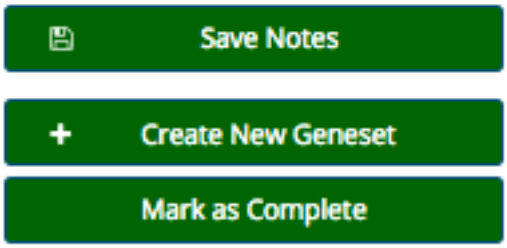
The citation information is present, and the curation group is identified, but there is no curator assigned and no associated genesets.

Assignment to a curator could have been done via the **Curation Task Management** page as detailed previously, or by using the **Assign To Curator** button on this page. The functionality of that button is essentially the same as on the other page, with an option to select a curator, and include a curation note.

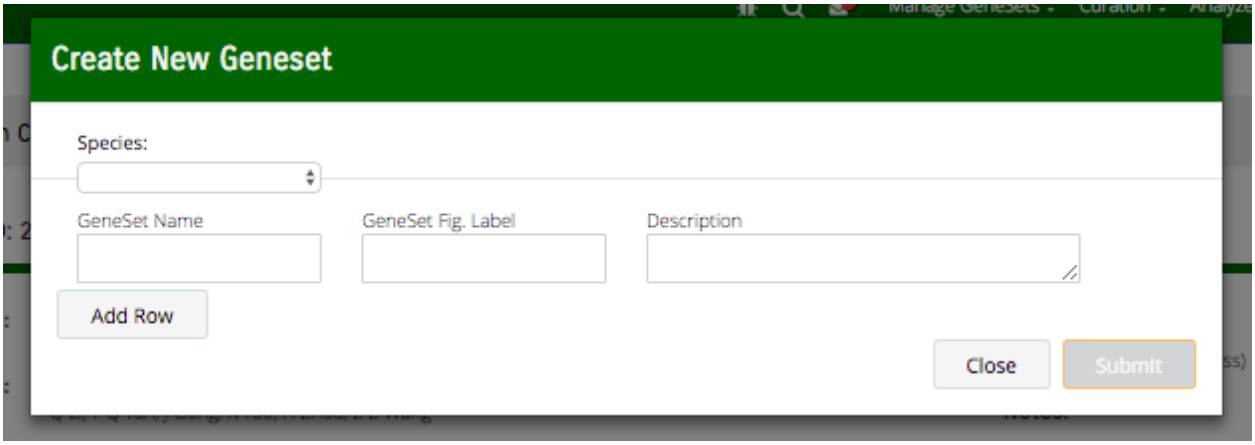
Once the curator is assigned, the curator's name and any notes that have been entered will appear in the upper right hand side of the page.



As the assignee of a publication, you will be presented with an additional button below **Save Notes** to be used to **Create New Geneset**. The **Reassign** button that was visible to the administrator now becomes a **Mark as Complete** button.



Clicking on the Create New Geneset button brings up a dialog that allows you to enter a "stub" for one or more new genesets. A stub is essentially a placeholder for a geneset that will be more completely populated at a later time. This gives a curator the ability to quickly create a bunch of stubs while reviewing an article without having to enter the full information for each.

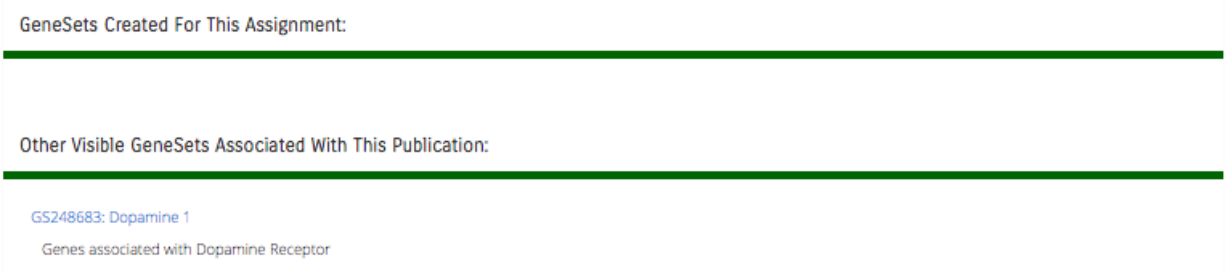


The curator can select the species of interest and then just enter the name, the label to be used in figures and a description. They can add multiple for this species by selecting **Add Row**, and when they've entered the information for all the geneset stubs associated with this species, they hit **Submit**.

When you've hit Submit, some automatic annotation of the geneset happens in the background. Your geneset stub will not immediately become visible under **GeneSets Created For This Assignment**. Instead you will see "loading…". Once the geneset stubs are created the page will display the new geneset stubs.

Once it's loaded the geneset stub will appear under **GeneSets Created For This Assignment**. It might take a while for the new geneset stub(s) to appear in the list of genesets associated with the publication assignment, since GeneWeaver is calling out to an external text annotator to annotate the geneset description and publication abstract.

If there are other genesets visible to the user that are associated with this publication, but were not created through this publication assignment, then they will show up under **Other Visible GeneSets Associated With This Publication**.

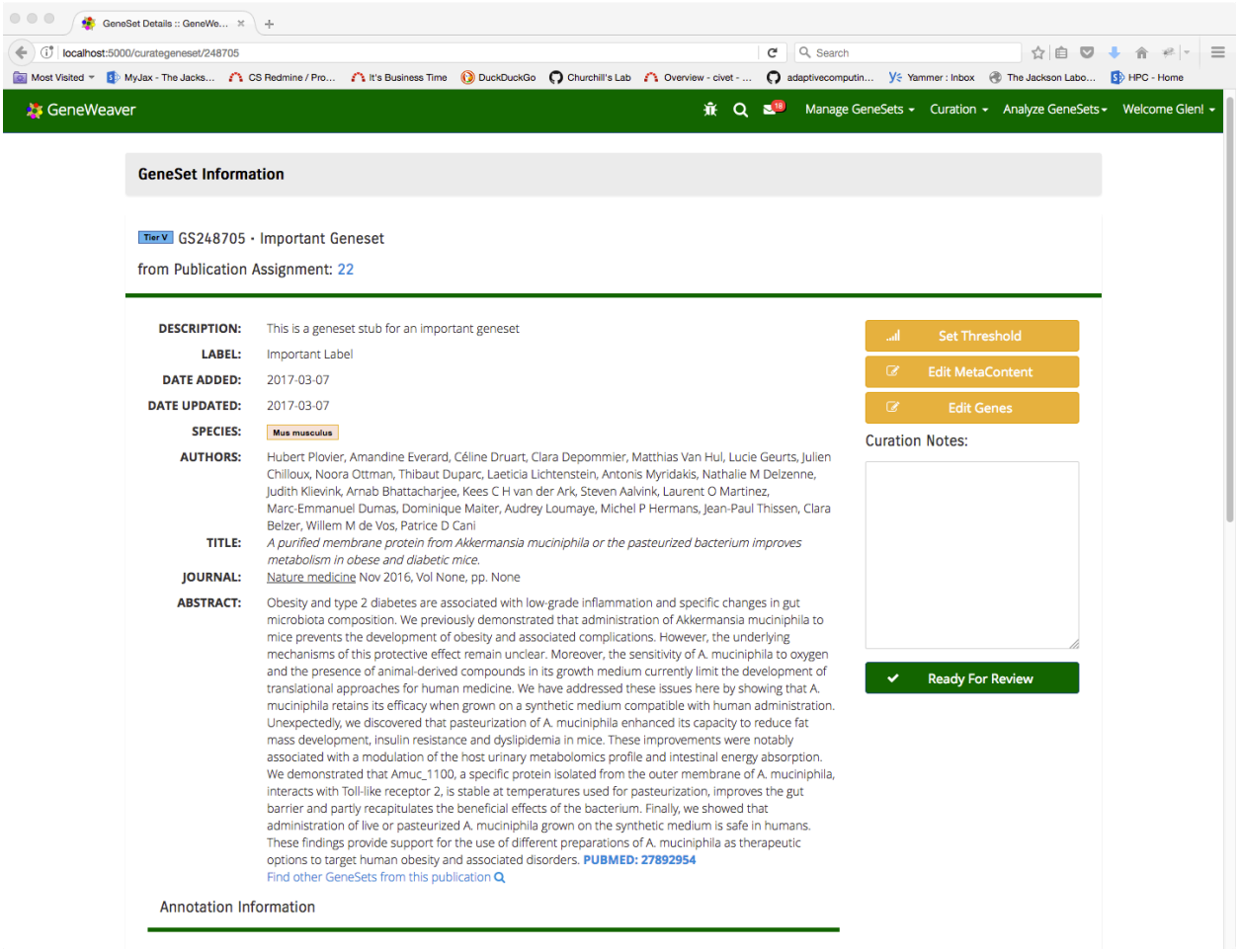


Once the geneset stubs have been created, the curator can click on the link for any one of the genesets, and begin curation of an actual geneset.

When curation of all of the associated genesets for this publication are complete, the curator should click the **Mark as Complete** button on the **Publication Curation Assignment** page.

Curation Page The geneset curation page is essentially the standard ***view geneset details*** page with some of the features turned off. On this page the curator can add or remove genes from the geneset, set a threshold, edit meta content, or update the curation notes. Once the curator has finished editing the geneset they can mark is **Ready for Review**, which will send the geneset back to the group administrator

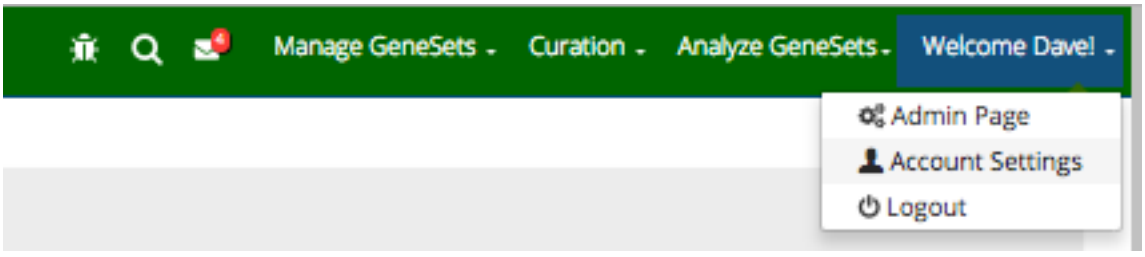
for review. If the group has multiple administrators then the geneset will be sent to the administrator that assigned the curation task to the curator.

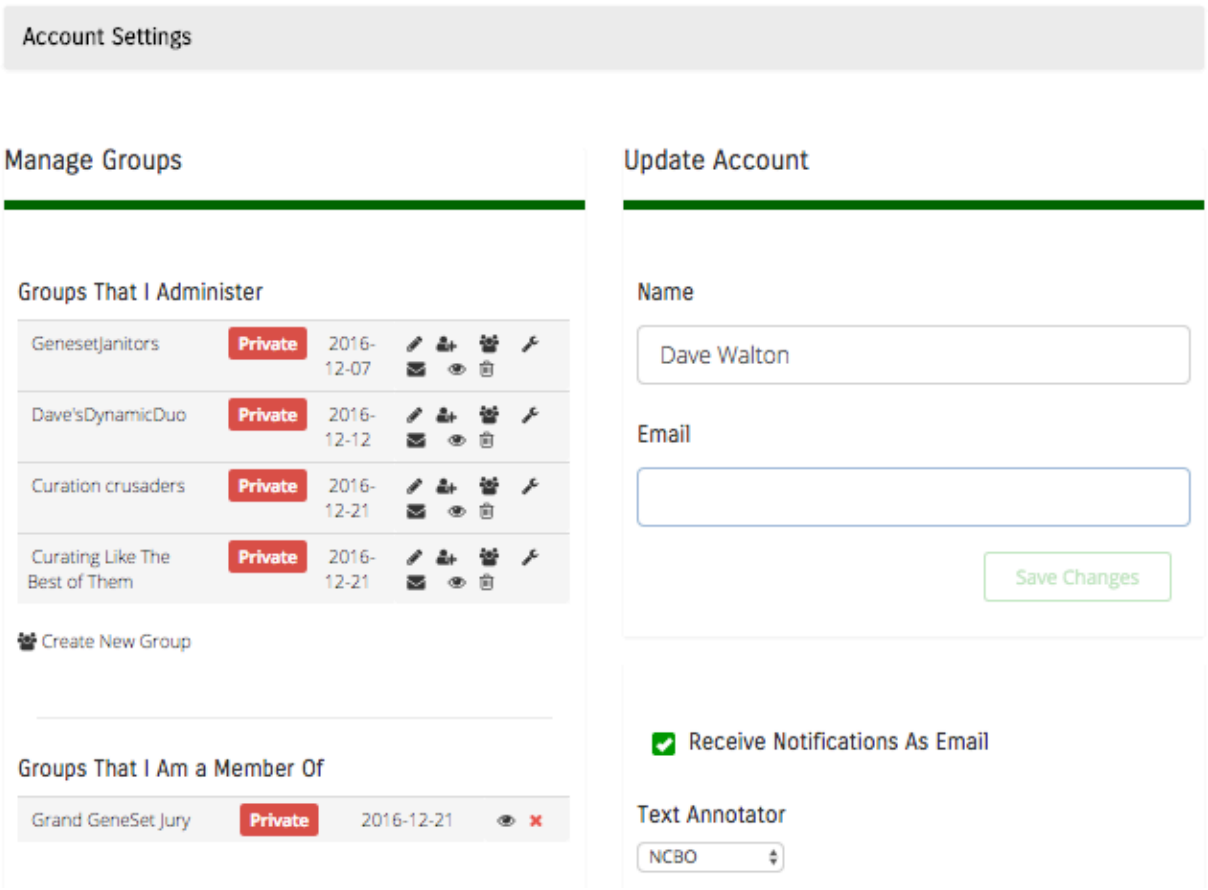


Top ↑

Notifications

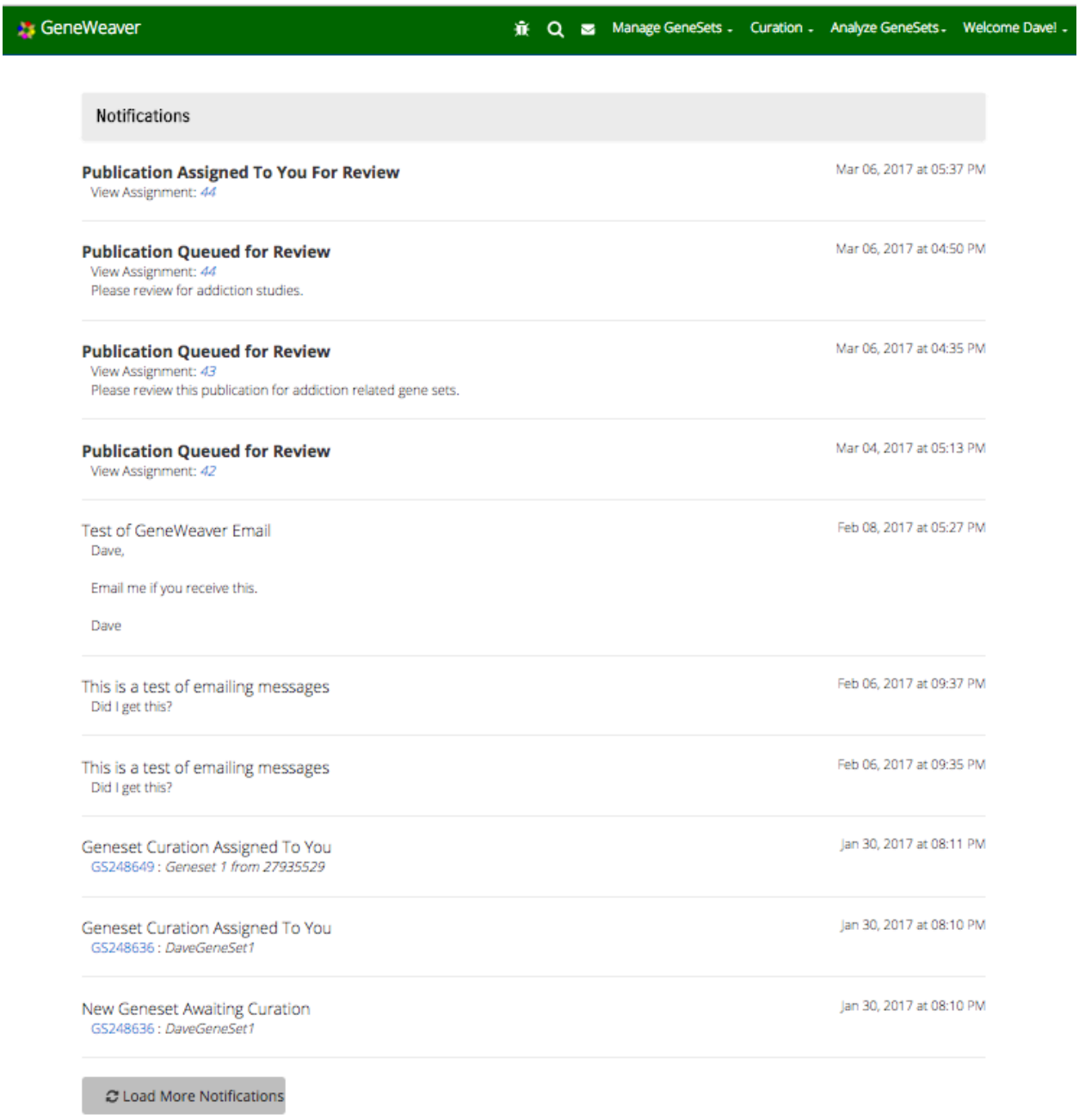
Notifications are the mechanism GeneWeaver uses to send messages within the application. There is also an option to receive email for notifications, which can be controlled from the **Account Settings** page.





Regardless of whether or not a user has been configured to receive emails, they will always receive messages through the **Notifications** page. The fact that you have pending notifications will be noted in the menu bar by a red indicator over the envelope icon.

The **Notifications** page itself is fairly straight forward listing the notifications that have not yet been seen in bold, and the rest of the notifications in normal font. There is a button at the bottom of the page that allows you to **Load More Notifications** so that you can get your full history of notifications.



Top ↑

Analysis Tools

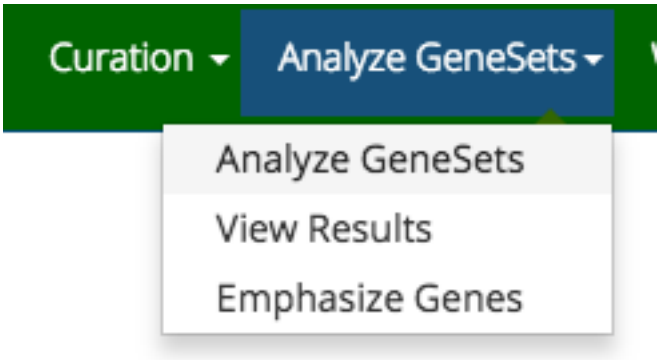
GeneWeaver uses a set of analysis tools to operate on genes and gene sets. These tools evaluate a range of data inputs for the purposes of elucidating hierarchical relationships among a set of gene sets of interest. They can be used to visualize bipartite clusters, **HiSim Graph** or visualize genes with the more common intersections, **GeneSet Graph**.

Generation and visualization of a maximal triclique using the intersection of gene sets with the **Triclique Viewer Tool** can allow users to discover novel relationships between gene ontology terms. The over-

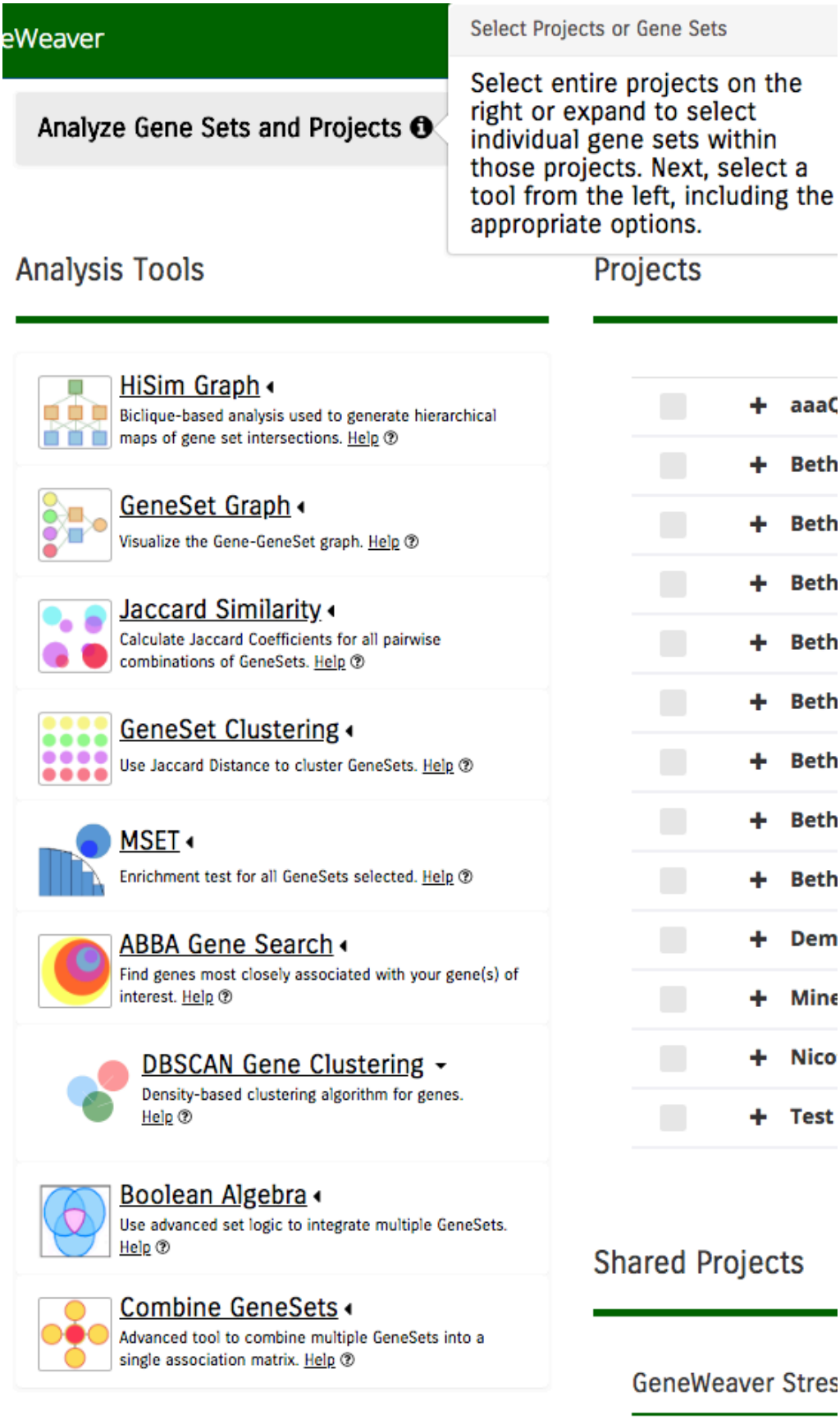
lap/similarity of gene sets themselves can be visualized with **Jaccard Similarity** plots. These set overlaps are also available for **Clustering**, while component gene intersections can be found on our **Gene Intersection Lists**. The **Boolean Algebra** tool uses advanced set logic to integrate multiple genesets. For each tool, GeneWeaver allows users to expand their search beyond a single species using **Homology Mapping**.

Analyze Gene Sets Tab

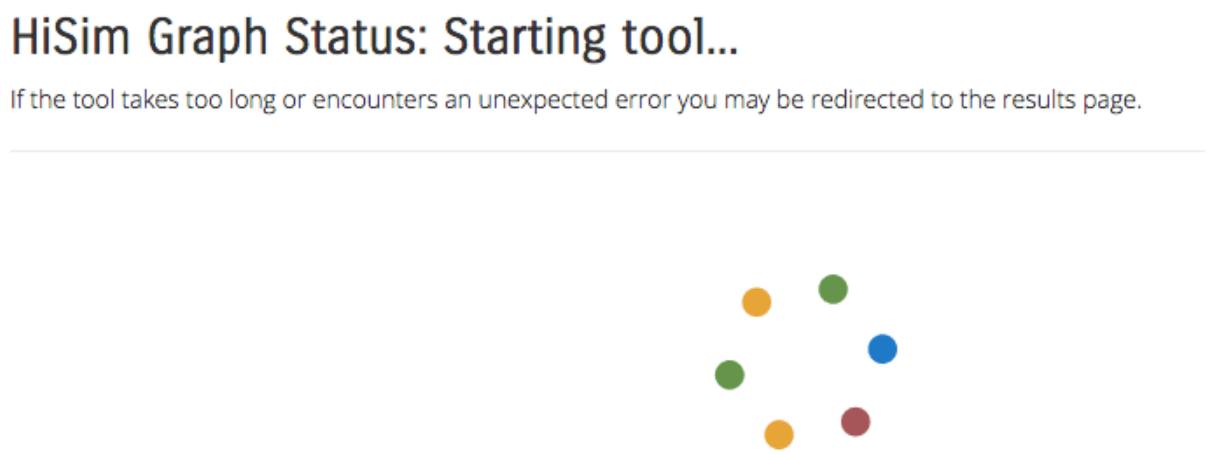
Use the analyze gene sets tab on the navigation bar to move to the analysis tools.



A registered user or guest user who has a temporary project will see the Analyze page. Down the left side are all the tools. Select one or more projects or gene sets and click on the desired tool. Options will then be displayed below the tool. Select the desired options and click the Run button.

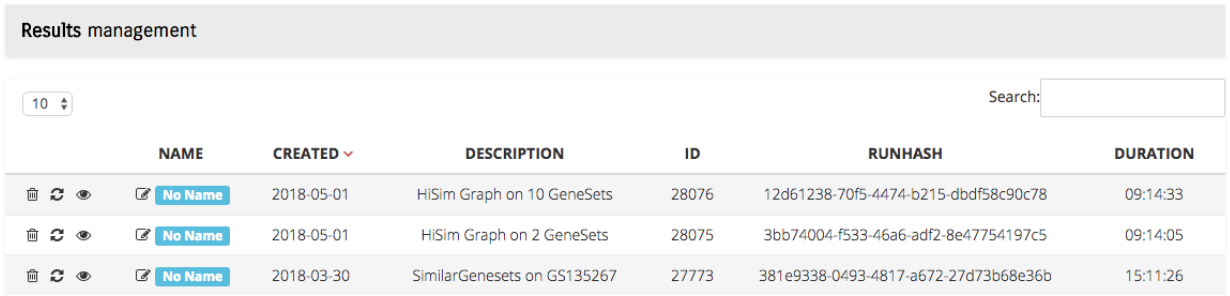

eWeaver
Select Projects or Gene Sets
Analyze Gene Sets and Projects
Select entire projects on the right or expand to select individual gene sets within those projects. Next, select a tool from the left, including the appropriate options.
Analysis Tools
Projects
HiSim Graph
Biclique-based analysis used to generate hierarchical maps of gene set intersections. Help
GeneSet Graph
Visualize the Gene-GeneSet graph. Help
Jaccard Similarity
Calculate Jaccard Coefficients for all pairwise combinations of GeneSets. Help
GeneSet Clustering
Use Jaccard Distance to cluster GeneSets. Help
MSET
Enrichment test for all GeneSets selected. Help
ABBA Gene Search
Find genes most closely associated with your gene(s) of interest. Help
DBSCAN Gene Clustering
Density-based clustering algorithm for genes. Help
Boolean Algebra
Use advanced set logic to integrate multiple GeneSets. Help
Combine GeneSets
Advanced tool to combine multiple GeneSets into a single association matrix. Help
aaaC
Beth
Beth
Beth
Beth
Beth
Beth
Beth
Beth
Dem
Mine
Nico
Test
Shared Projects
GeneWeaver Stres

A tool can take a long time, depending on the size and complexity of the selected gene sets. A message will be displayed showing the progress of the tool. You can now navigate away from this page and later return to the results page.



View Results

The link to the results page is on the analyze gene sets tab.



Your tool has completed once the duration column has a time listed. From this page you can:

- Delete a test that you are no longer interested in
- Re-run a test
- View the test results
- Edit the test name
- Use the Search box to display test name matches
- Sort the columns by clicking on the header
- Select up to 100 results to display per page

Top ↑

HiSim Graph

About the HiSim Graph Tool

The HiSim Graph, short for Hierarchical Similarity Graph, is a tool for grouping functional genomic datasets based on the genes they contain. *For example*: The user may want to determine what a set of experiments on alcohol preference have in common, and what makes various experiments unique from one another. Alternatively, one may wish to take a large set of studies of related phenomena and identify their shared or distinct substrates. In this situation one may want to know whether there is a shared biological basis for addiction and learning, and if so, what the substrate is. The user might also want to examine studies of a large number of related disorders and determine whether a more appropriate biologically-based classification can be constructed.

The HiSim Graph Tool is designed to address these goals; it presents a tree of hierarchical relationships for a set of input GeneSets. The structure is determined solely from the gene overlaps of every combination of GeneSets.

Understanding the Results of the HiSim Graph

It's best to use the HiSim Graph Tool with a knowledge on what set intersections are: If GeneSet A contains Gene A, Gene B, and Gene C, and also GeneSet B contains Gene A, Gene B, and Gene D. Then the intersection of GeneSet A and GeneSet B will contain Gene A and Gene B, because an intersection of sets are whatever is contained in all sets intersected.

In terms of GeneSets, the smallest intersections (fewest GeneSets, most genes) are towards the right, and the largest intersections (most GeneSets, fewest genes) are on the left. When thinking about the genes in all the GeneSets, the roles are reversed (smallest number of genes on the left, largest number of genes on the right).

Figure 1: Relation of GeneSets to the HiSim Graph

HiSim Graphs must be interpreted in the context of the input GeneSets. The above example represents differentially expressed genes in multiple brain regions of alcohol preferring rats from a single study. The highest intersection represents a gene differentially expressed in all 5 brain regions. In this case, the highest intersection represents the highest amount of correspondence between data sets. As you move to the right, genes become more specific to the brain regions tested. Each solid node has children and can be collapsed by clicking on it. Leaf nodes are empty and colored by species, which is identified in a legend at the bottom of the screen.

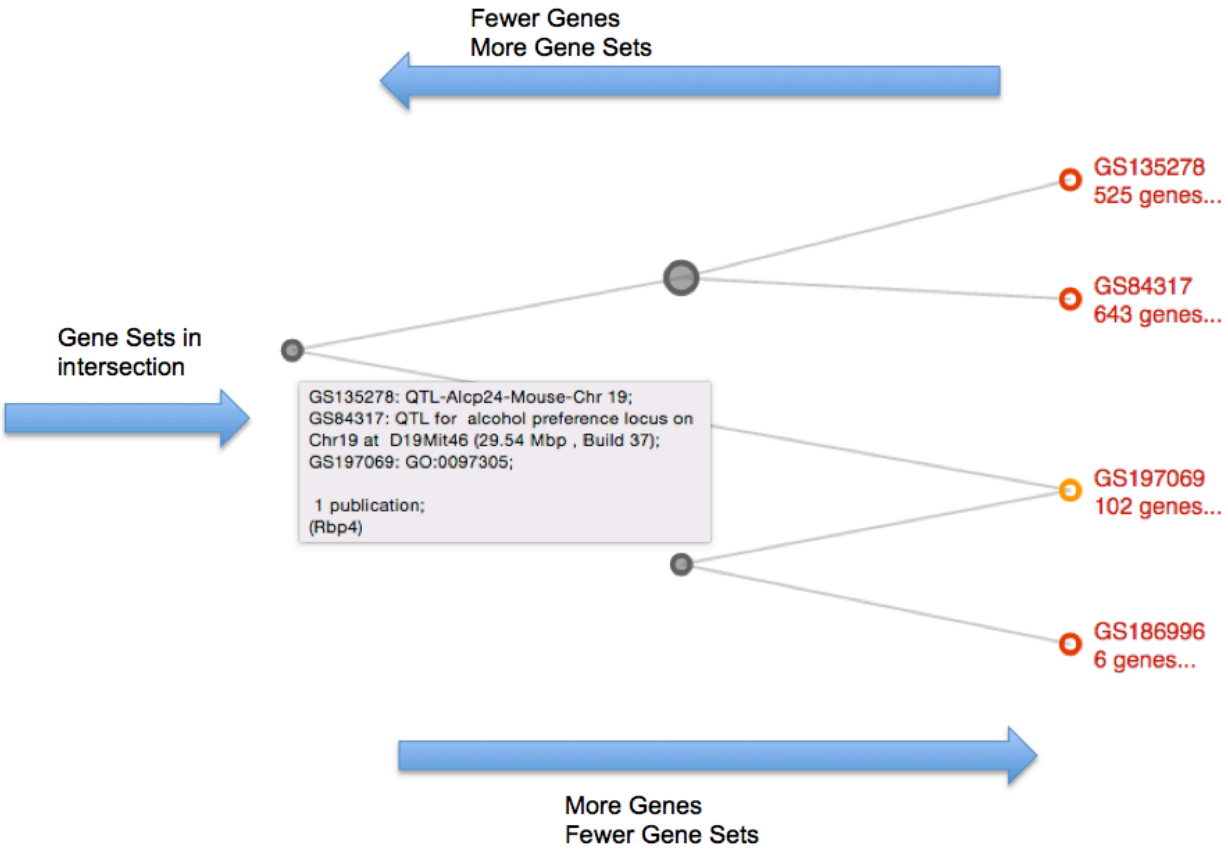

Fewer Genes
More Gene Sets
GS135278
525 genes...
GS84317
643 genes...
Gene Sets in
intersection
GS135278: QTL-Alcp24-Mouse-Chr 19;
GS84317: QTL for alcohol preference locus on
Chr19 at D19Mit46 (29.54 Mbp , Build 37);
GS197069: GO:0097305;
1 publication;
(Rbp4)
GS197069
102 genes...
GS186996
6 genes...
More Genes
Fewer Gene Sets

Figure 1:

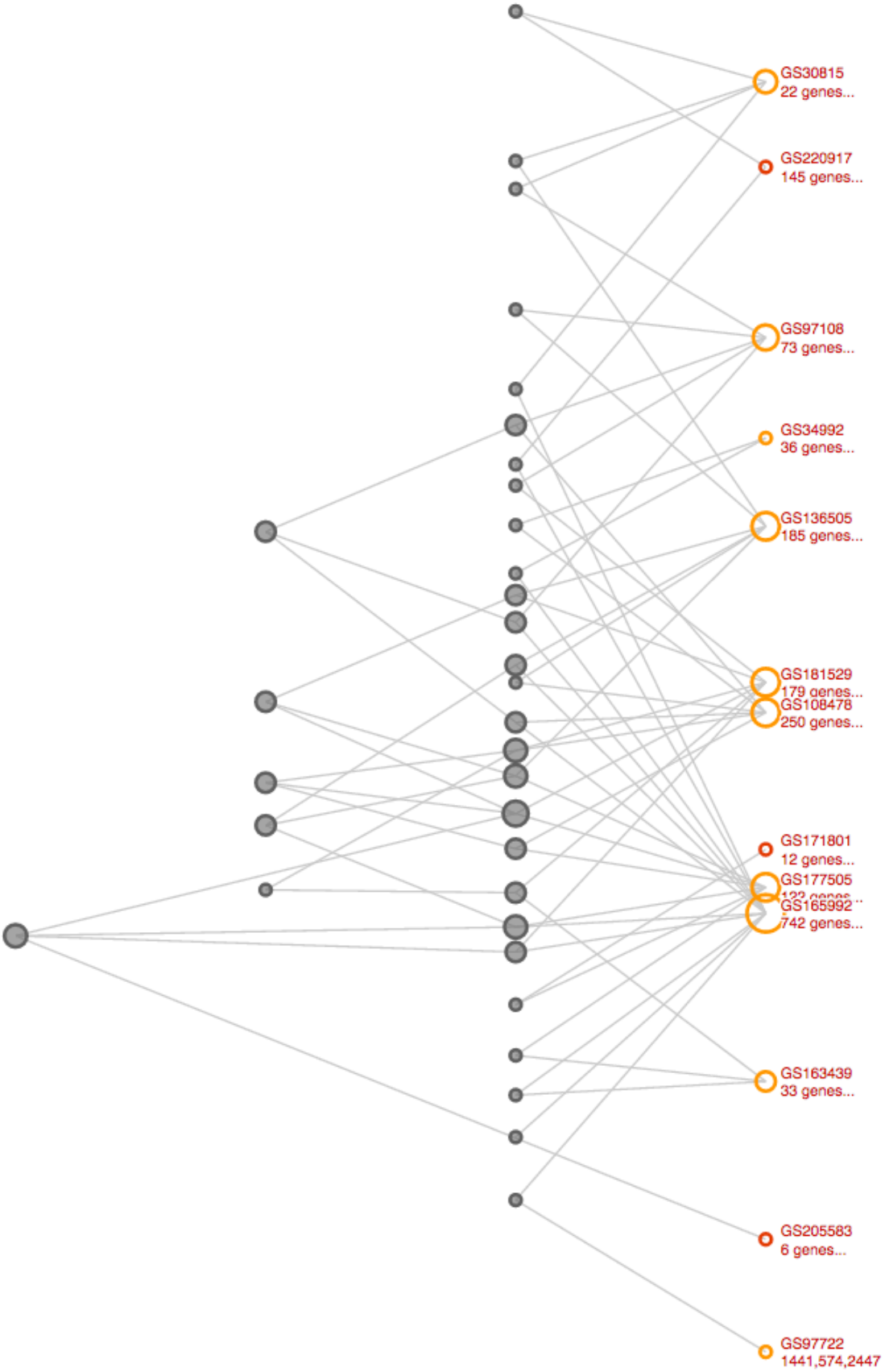

GS30815
22 genes...
GS220917
145 genes...
GS97108
73 genes...
GS34992
36 genes...
GS136505
185 genes...
GS181529
179 genes...
GS108478
250 genes...
GS171801
12 genes...
GS177505
GS165992
742 genes...
GS163439
33 genes...
GS205583
6 genes...
GS97722
1441,574,2447

Figure 2: A HiSIm Graph for diverse functions

If one were to start with multiple alcohol preference measures from different studies, the top of the HiSim Graph represents the correspondence between the experiments (such as well-characterized alcohol preference genes), and as you descend the graph the intersections describe more specific features shared between experiments (such as stress response or tissue source).

When starting with more loosely related inputs, interpretation becomes more difficult. If one started with alcohol preference, nicotine dependence, and traumatic brain injury data (Figure 2), the top of the HiSim Graph would represent more generic processes such as neural plasticity in this case.

Using the HiSim Graph Tool

Access the HiSim Graph Tool through the Analyze Genesets tab.

To generate a HiSim Graph, you must first select gene sets from a project. Projects may be created and updated by uploading GeneSets, searching the GeneWeaver database, or through the use of other tools in the GeneWeaver system. See the documentation for uploading GeneSets, Search, or Manage GeneSets to learn more about these functions. To select an entire project or multiple projects for analysis, check the box next to the project name. To select individual GeneSets within a project, click on the + beside the project name and check individual GeneSets using the check boxes. Next, click on the HiSim Graph icon in the Analysis tools box to the left of the project list. Select the options you would like for the tool to run on, and click Run.

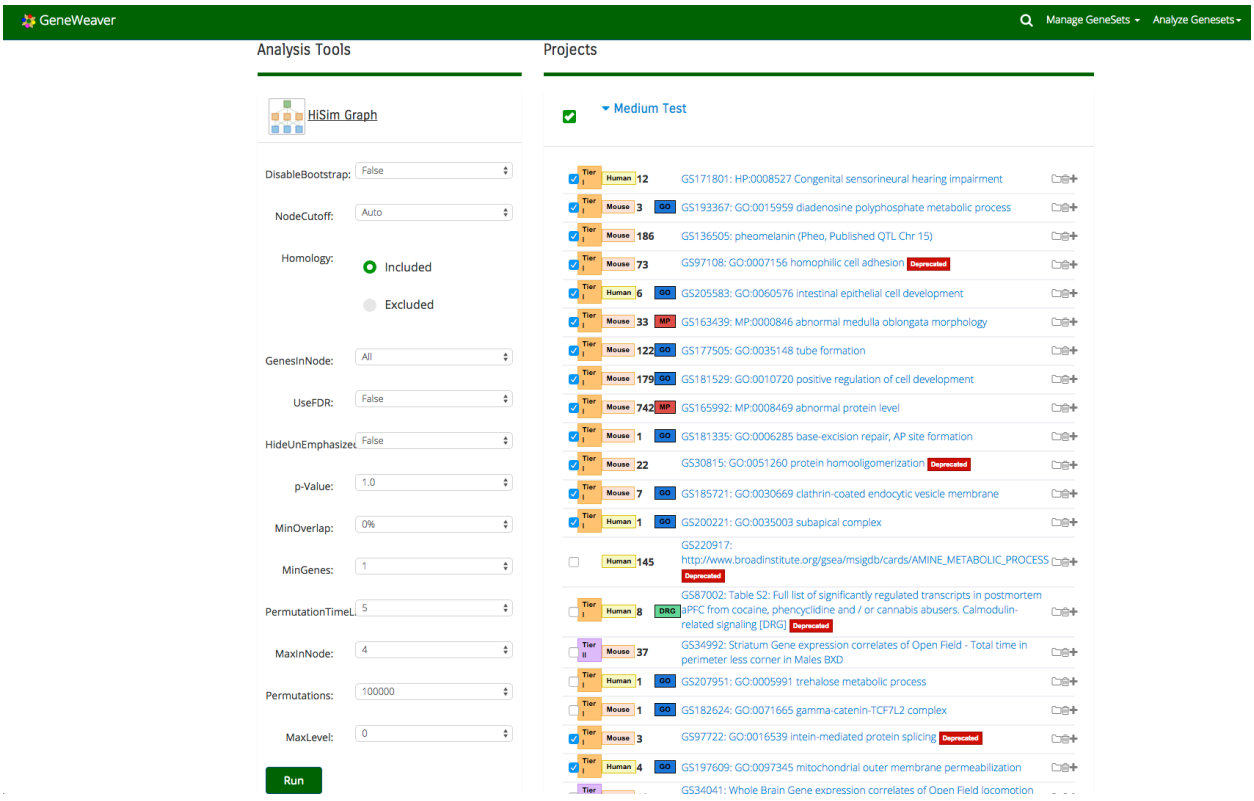


Figure 3: Selecting gene sets and executing an analysis from the Analyze GeneSets page

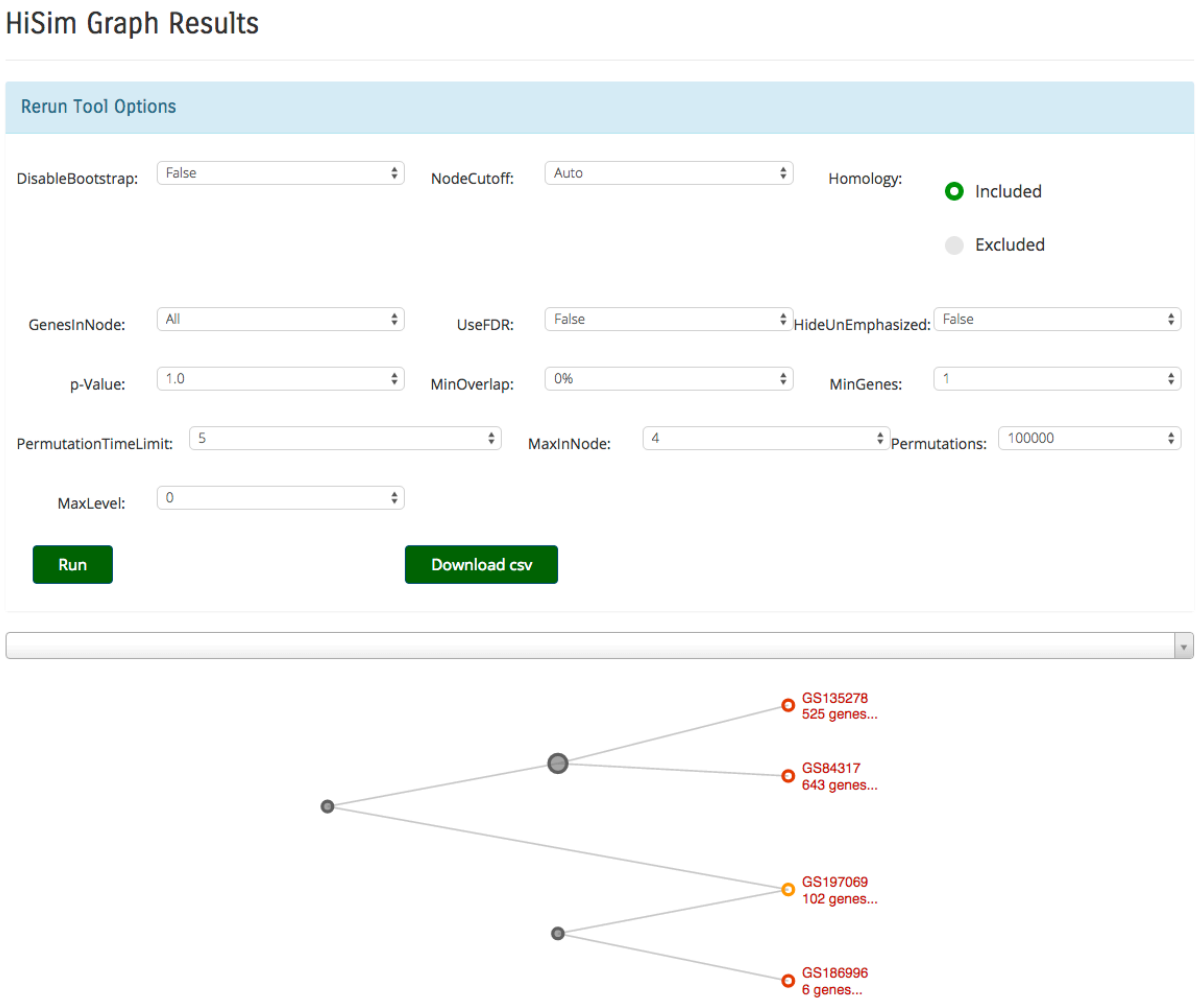


Figure 4: The results page for the HiSim Graph.

Most genes are connected to two of the input GeneSets. One gene is connected to three of the input sets. (Inset)

The GeneSet Intersection page

GeneSet intersection data can be downloaded as a csv file for subsequent analyses. The GeneSets giving rise to each node can be stored in a separate project.

The HiSim Graph opens and the nodes can be selected to expand the graph. More details of each intersection can be viewed by clicking on the individual nodes in the tree. A link at the bottom of the frame allows download of the csv.

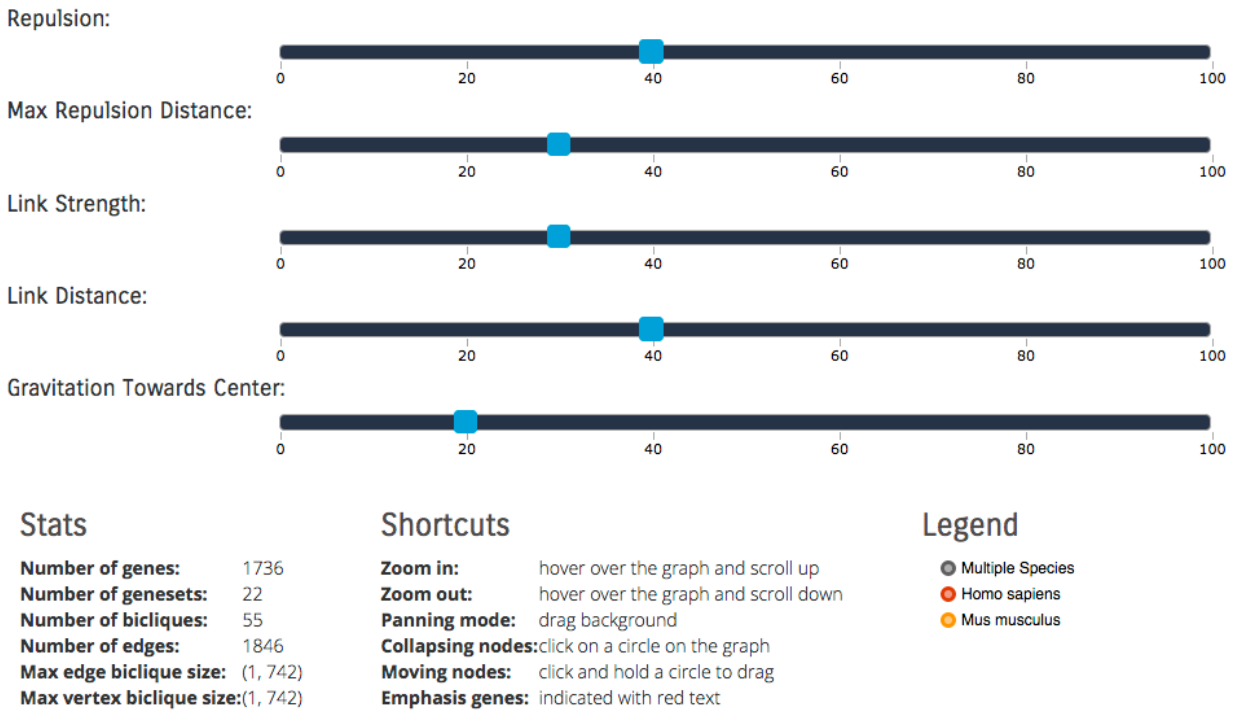


Figure 5: These options are available for the HiSim Graph, to change the way nodes interact with each other. The stats of the graph, as well as shortcuts and the legend identifying each species in the graph, are also displayed.

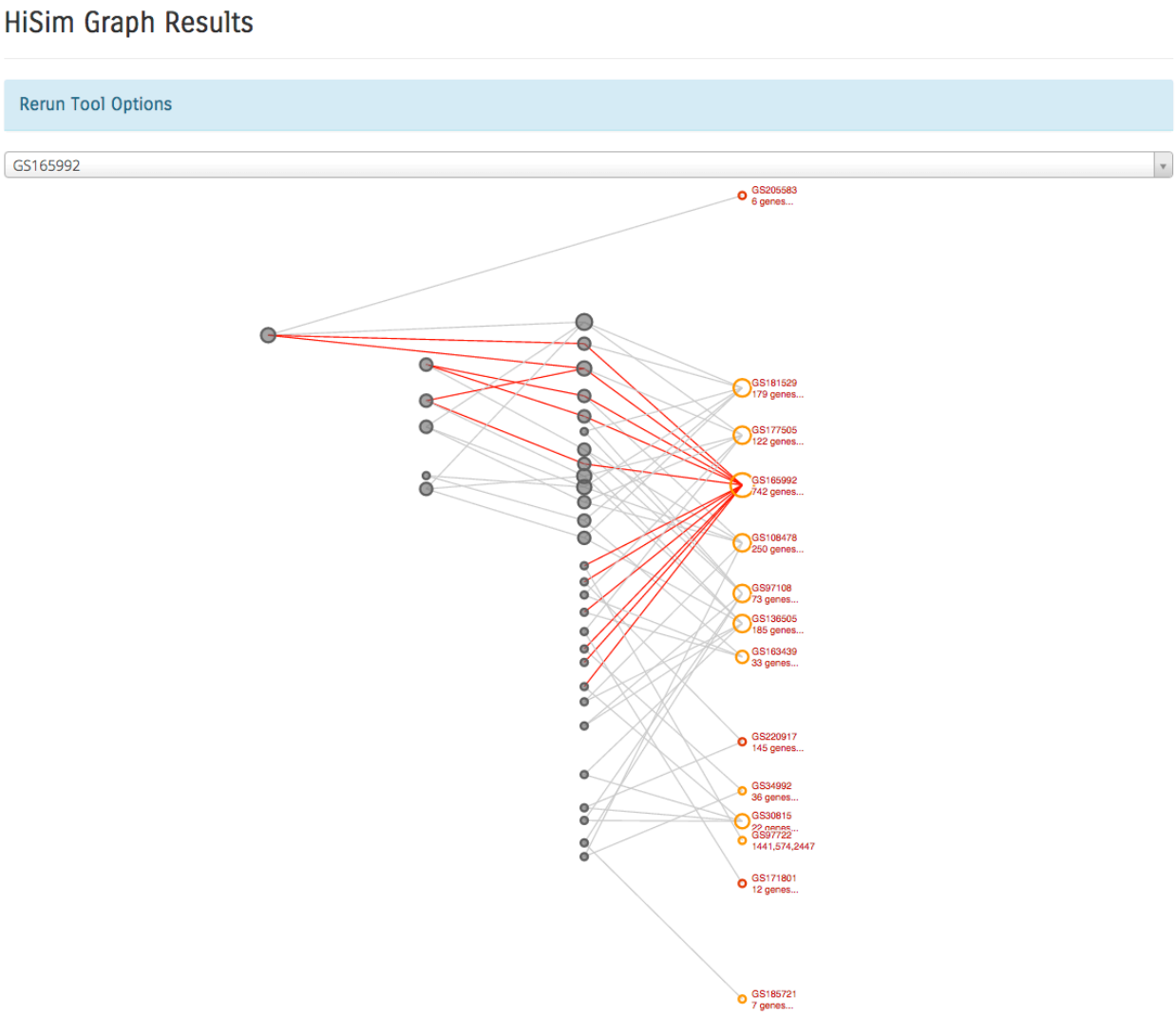


Figure 6. This shows the search function, which highlights paths between nodes containing the item searched for, whether it be gene, geneset, or species.

Options

There are a number of options available to optimize the HiSim Graph analyses. You may access the following options on the Analyze GeneSets page by clicking on the HiSim Graph Tool.

DisableBootstrap

When the resulting HiSim Graph is unimaginably large, a bootstrapping filter is applied to reduce the output size. This step removes edges that are weakly supported by the underlying data, for example, those partitions of GeneSet subgroups that are driven by a single gene difference between the groups. If you would like the large, unfiltered graph instead, set this option to True to disable bootstrapping. Be warned this may stretch the graph's size.

Figure 6: A HiSim Graph with DisableBootstrap turned on (True).

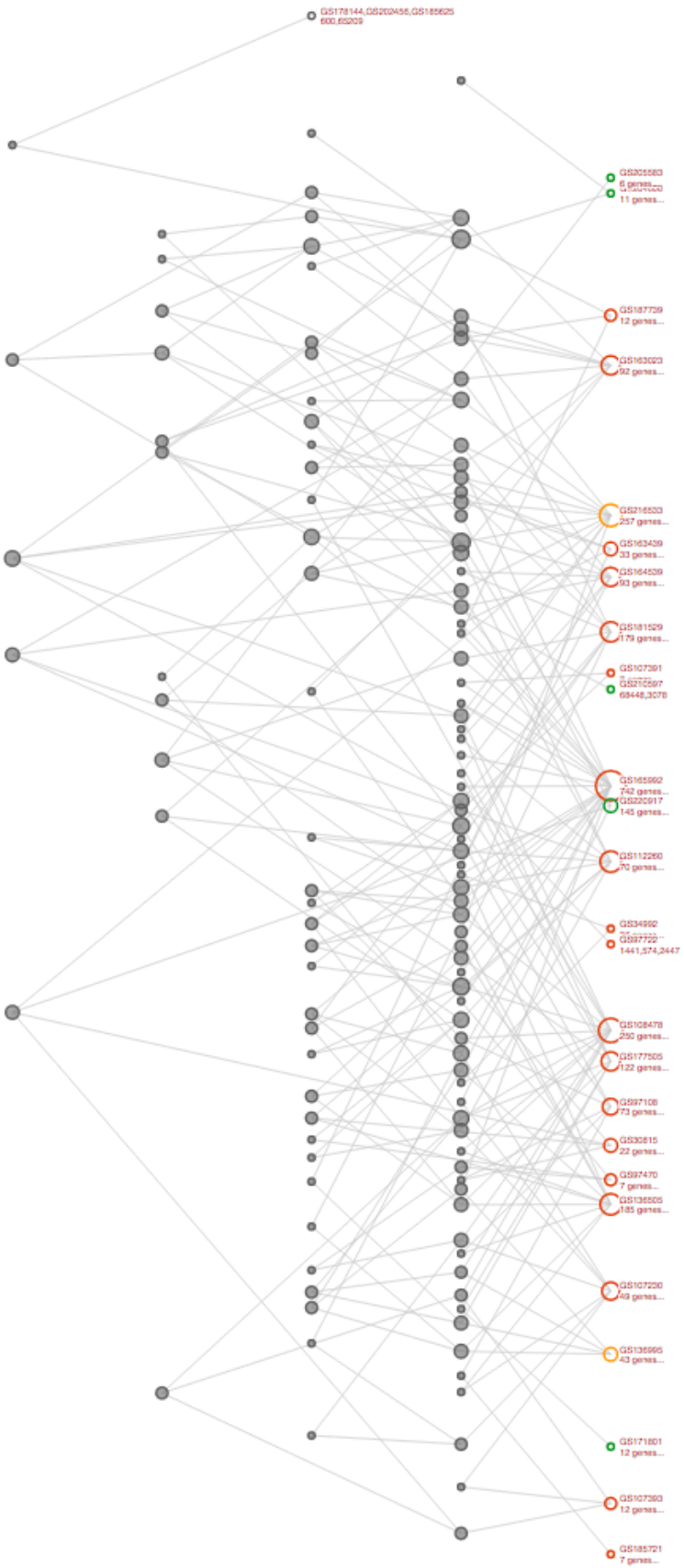

GS178144,GS202456,GS185625
600,66209
GS205583
6 genes...
11 genes...
GS187729
12 genes...
GS163023
92 genes...
GS216533
257 genes...
GS163439
33 genes...
GS164539
93 genes...
GS181529
179 genes...
GS107391
GS210597
68448,3078
GS165992
742 genes...
GS222917
145 genes...
GS112260
70 genes...
GS34992
GS97722
1441,574,2447
GS108478
250 genes...
GS177505
122 genes...
GS97108
73 genes...
GS30815
22 genes...
GS97470
7 genes...
GS136505
185 genes...
GS107220
49 genes...
GS136995
43 genes...
GS171801
12 genes...
GS107393
12 genes...
GS185721
7 genes...

Figure 2:

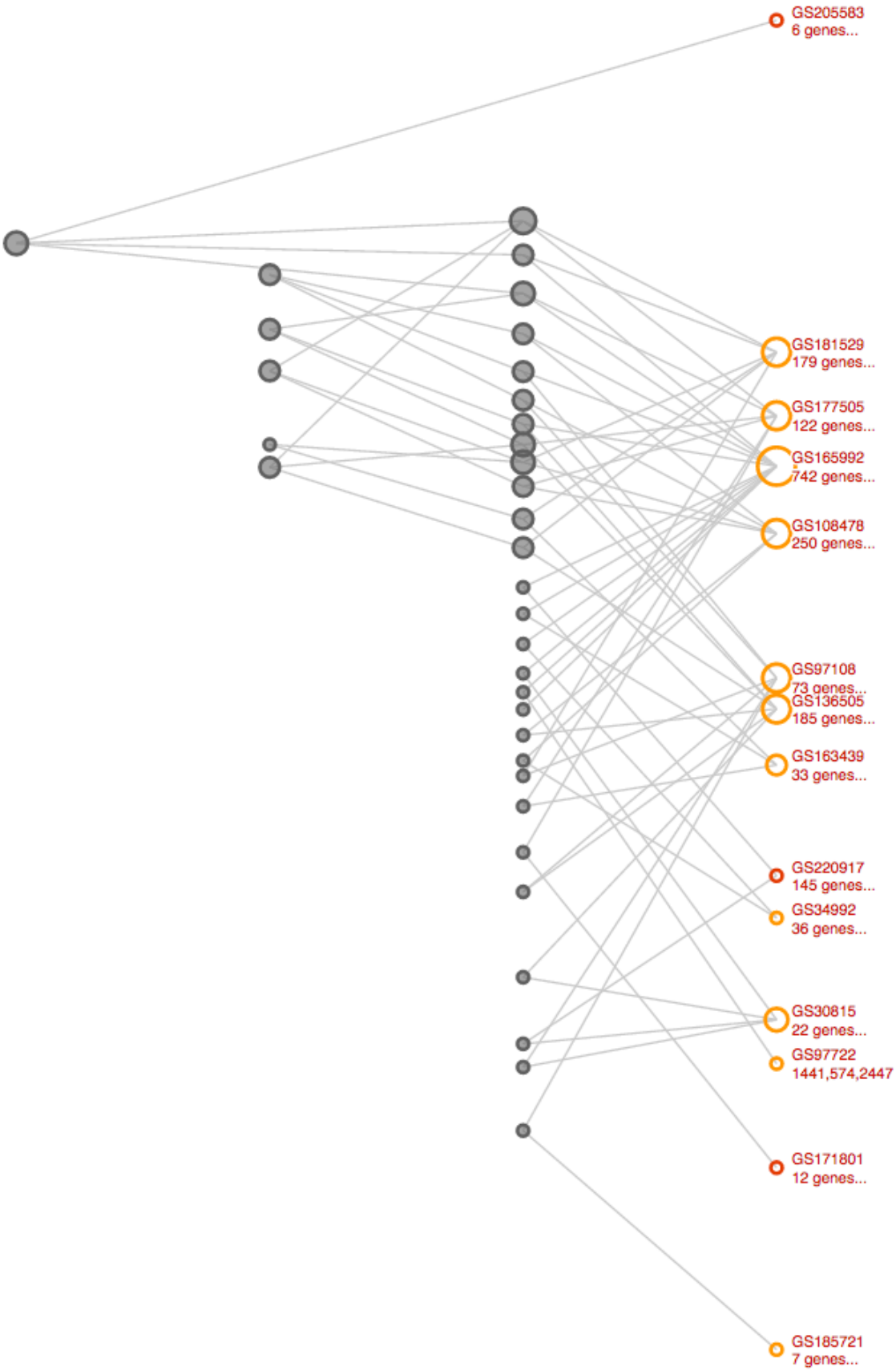

GS205583
6 genes...
GS181529
179 genes...
GS177505
122 genes...
GS165992
742 genes...
GS108478
250 genes...
GS97108
73 genes...
GS136505
185 genes...
GS163439
33 genes...
GS220917
145 genes...
GS34992
36 genes...
GS30815
22 genes...
GS97722
1441,574,2447
GS171801
12 genes...
GS185721
7 genes...

Figure 7: A HiSim Graph with DisableBootstrap turned off (False).

Homology

Include homology to integrate multi-species data. This is done by using homologene mappings to relate identifiers across species. If homology is excluded, data from multiple species will be segregated into separate trees.

Figure 8: Homology excluded. A separate map is drawn for mouse, no overlap with human is allowed.

Figure 9: Homology included. GeneSets from mouse and human are allowed to be mixed and are intertwined as one.

MinGenes

The minimum number of genes for an intersection. The default of 1 means that all intersections will be displayed. Increasing the value means that intersections with fewer genes will not be displayed in the output, decreasing noise and displaying more robust correspondence between GeneSets. This generally has the effect of removing the topmost nodes.

Figure 10: As shown above, the left tree is with the default MinGenes = 1, the right tree is with the default MinGenes = 5.

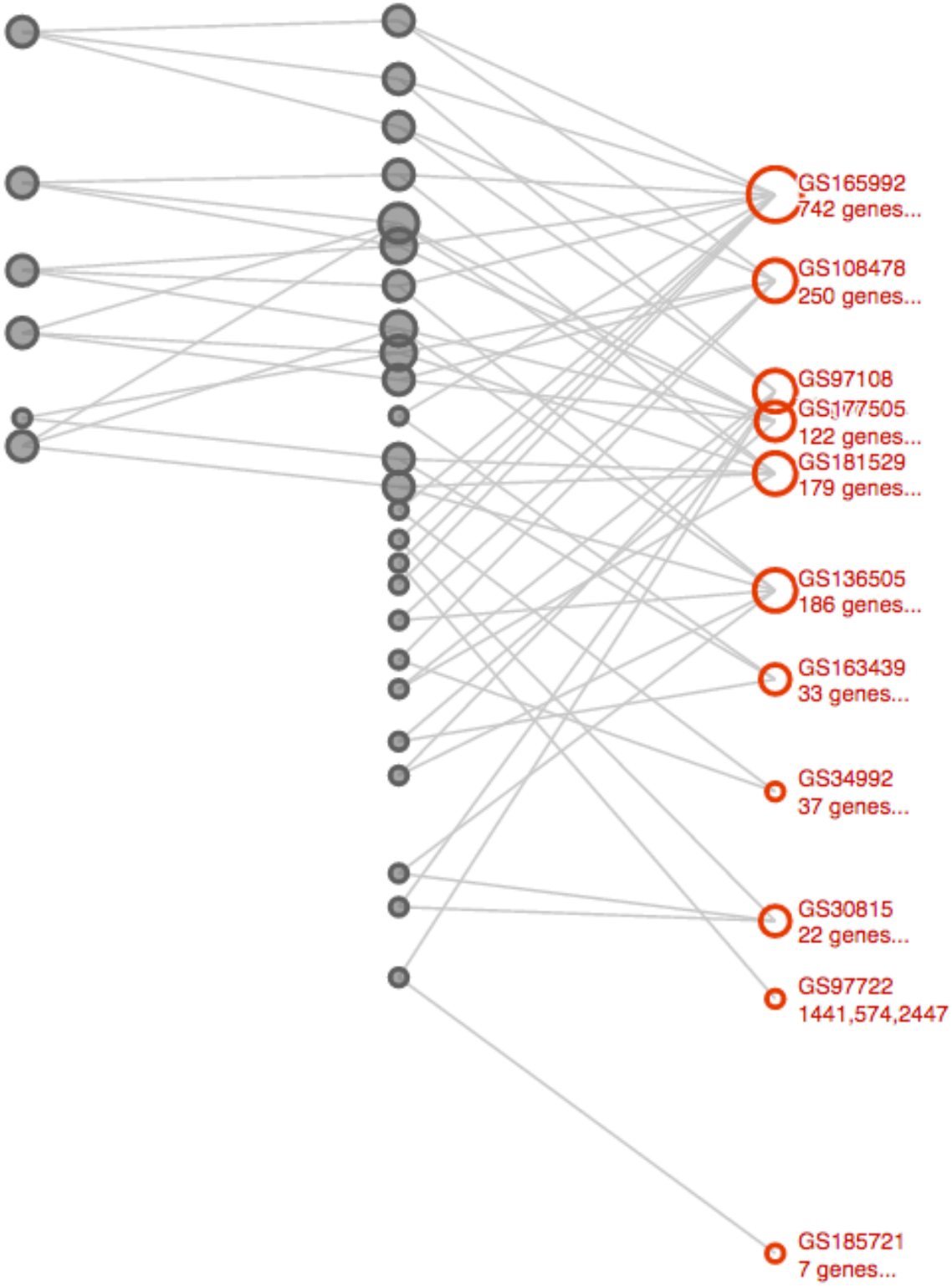

GS165992
742 genes...
GS108478
250 genes...
GS97108
GS177505
122 genes...
GS181529
179 genes...
GS136505
186 genes...
GS163439
33 genes...
GS34992
37 genes...
GS30815
22 genes...
GS97722
1441,574,2447
GS185721
7 genes...

Figure 3:

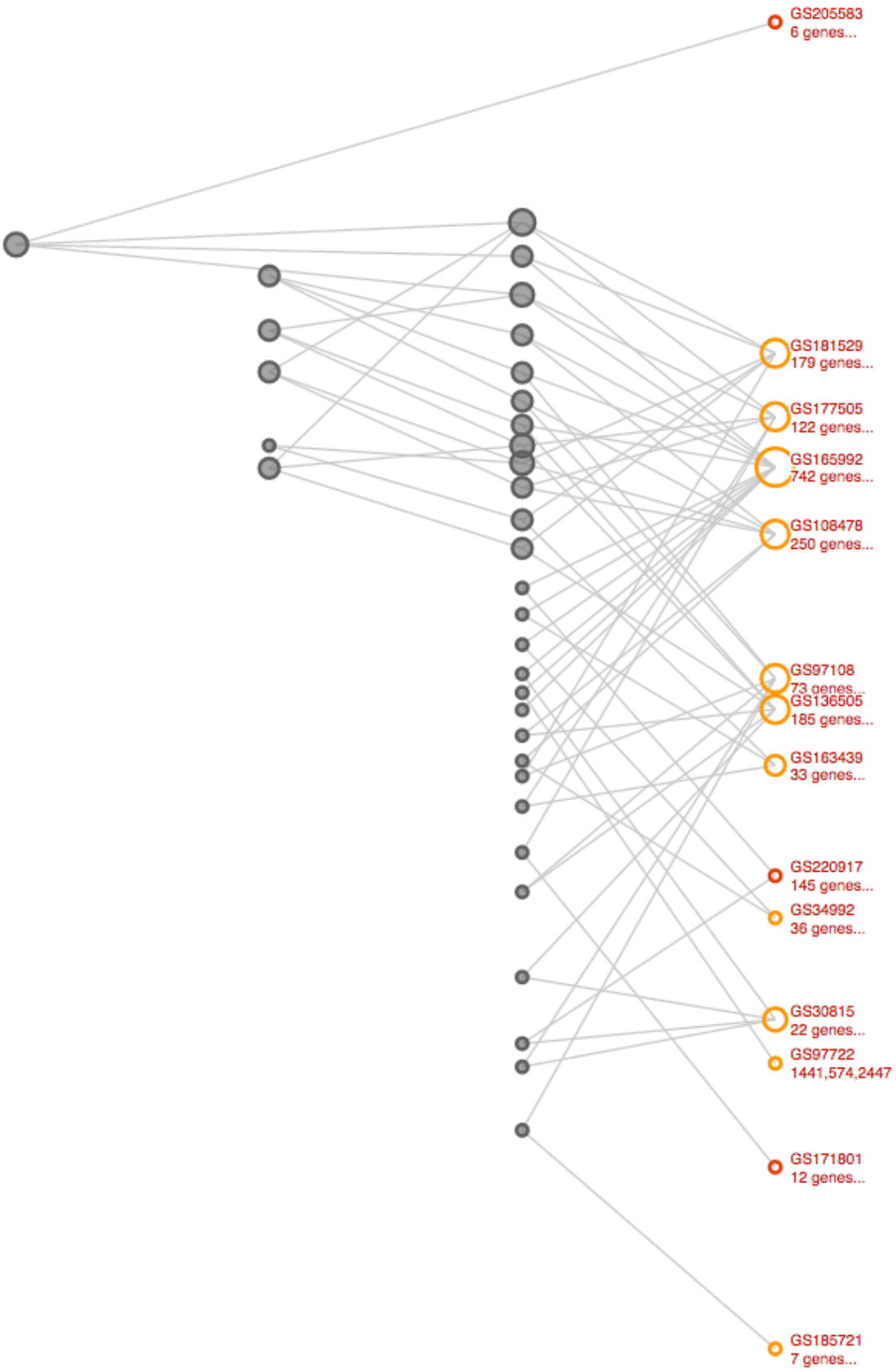

GS205583
6 genes...
GS181529
179 genes...
GS177505
122 genes...
GS165992
742 genes...
GS108478
250 genes...
GS97108
73 genes...
GS136505
185 genes...
GS163439
33 genes...
GS220917
145 genes...
GS34992
36 genes...
GS30815
22 genes...
GS97722
1441,574,2447
GS171801
12 genes...
GS185721
7 genes...

Figure 4:

Permutations

The HiSim Graph can ultimately address questions among highly curated data such as how much dimension reduction does gene overlap provide. For example, one may take a large set of gene sets associated with mood disorders and ask whether the data are similar enough to group together, i.e., of all possible subset intersections, how many are populated, and is this result better than chance?

The maximum number of permutations to run is set to 0 by default since it can take a long time to run for large input sets. The genes contained in each GeneSet are permuted over the union of all genes in the input sets, controlling for the size of each GeneSet. The permutation tests measure the likelihood of getting a similar tree structure (Parsimony) or of getting a similar aggregation of genes in each intersection (Gene Aggregation). Note that this is a maximum value since the actual results may be fewer due to the time limit.

Parsimony is a simple measure of the percentage of observed intersections out of all possible intersections. This is mathematically defined as:

$$n = \left| input\ sets \right|$$

$$Int_{observed} = \left| nodes \in PhenomeMap \right|$$

$$Int_{possible} = \sum_{k=1}^{n} \binom{n}{k}$$

$$Parsimony = \frac{Int_{observed}}{Int_{possible}}$$

Figure 5:

Figure 11: For those that aren't aware of the mathematical implications of parsimony, think of it as one of the many measures of accuracy for a map. You want more parsimony, but you can't always get full parsimony.

Gene Aggregation is a measure of the total node/tree probability. Each node is scored based on the intersection of genes and gene sets. Then the product of these scores is used to assign an overall tree aggregation probability:

$$Score_{node} = \left(\frac{\left| genes \in node \right|}{\left| \bigcup_{child \in children_{node}} genes \in child \right|} \right)^{\left| genesets \in node \right|}$$

$$GeneAggregation = \prod_{n \in PhenomeMap} Score_n$$

Figure 6:

Figure 12: Aggregation is another measure of accuracy that balances with parsimony in this tool, neither are ever fully accurate alone, but together they are more fine-tuned.

Permutation Time Limit

The maximum amount of time to spend doing permutations. For example, if Permutations is set to 100,000 and this value is 5 minutes, the result with either have 100,000 permutations (if they finished within 5 minutes), or will be truncated to the number of permutations which were able to finish within 5 minutes. The more time you give to Permutation Time Limit, the more accurate your results will be.

Top ↑

GeneSet Graph

Why Use the GeneSet Graph Tool

The GeneSet Graph is designed for the user in need of a partitioned display to illustrate just how tied genes are to one another. For example: a user in need of a GeneSet Graph would look for visual references more than chemical references or references by utility. A GeneSet Graph can also help pick apart the most valuable or most occurring genes depending on the user's preference.

Understanding the GeneSet Graph Tool

The GeneSet Graph Tool presents a partitioned display of genes and GeneSets. Genes are represented by elliptical nodes, and GeneSets are represented by boxes. The least-connected genes are displayed on the left, followed by the GeneSets, then the more-connected genes in increasing order to the right. Genes and GeneSets are connected by colored lines to show what genes are in which GeneSets. In this way, the GeneSet Graph displays the bipartite graph of the genes and GeneSets, but modifies the display of the gene partition to make it easier to visually interpret.

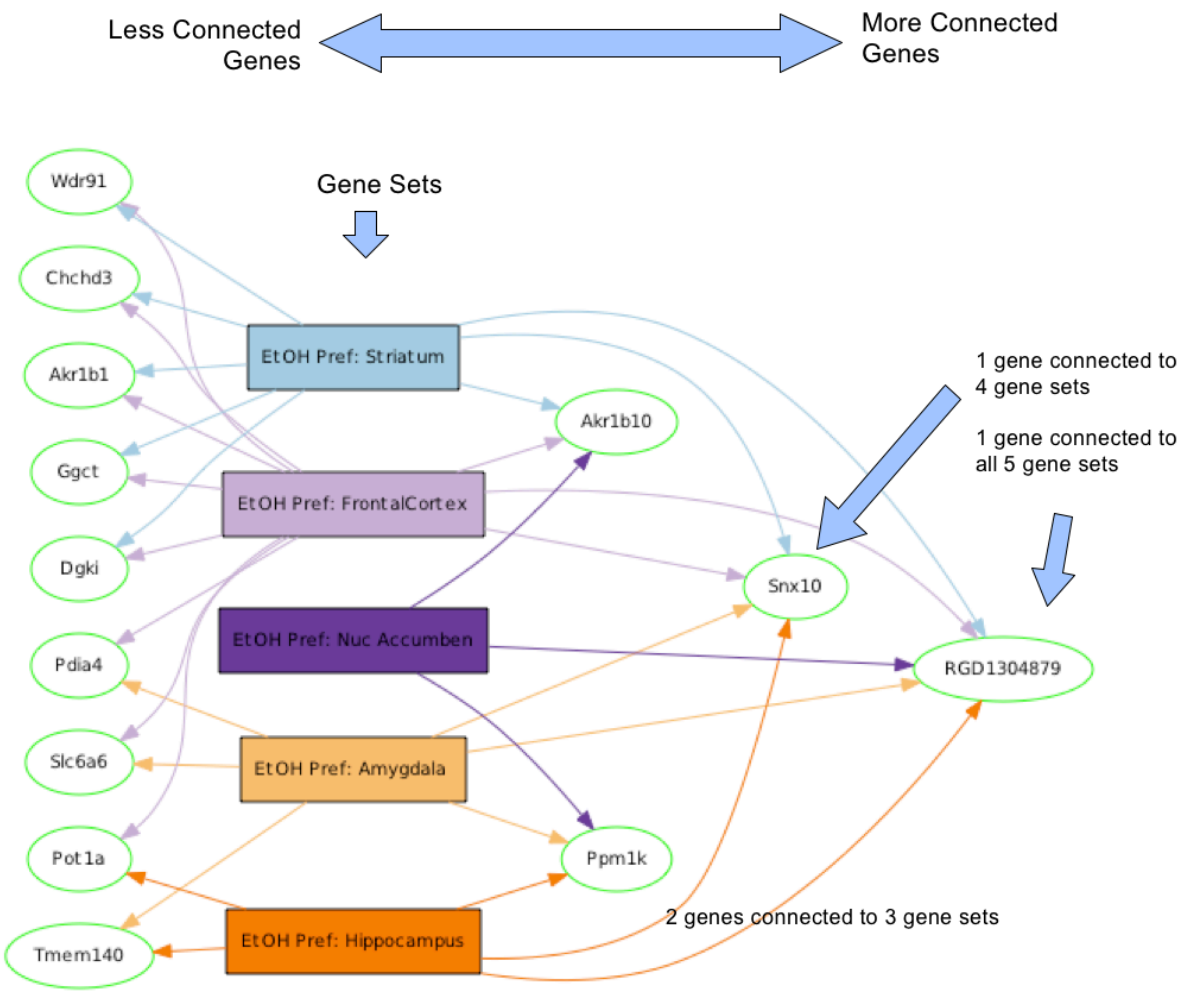


Figure 1: Least connected genes to the left, GeneSets in the middle, most connected genes on the right.

Using the GeneSet Graph Tool

Access the GeneSet Graph Tool through the Analyze Genesets tab.

To generate a GeneSet Graph, you must first select gene sets from a project. Projects may be created and updated by uploading GeneSets, searching the GeneWeaver database, or through the use of other tools in the GeneWeaver system. See the documentation for uploading GeneSets, Search, or Manage GeneSets to learn more about these functions. To select an entire project or multiple projects for analysis, check the box next to the project name. To select individual GeneSets within a project, click on the + beside the project name and check individual GeneSets using the checkboxes. Next, click on the GeneSet Graph icon in the Analysis tools box to the left of the project list. (For users that want to change options, press the green + sign before they start the tool).

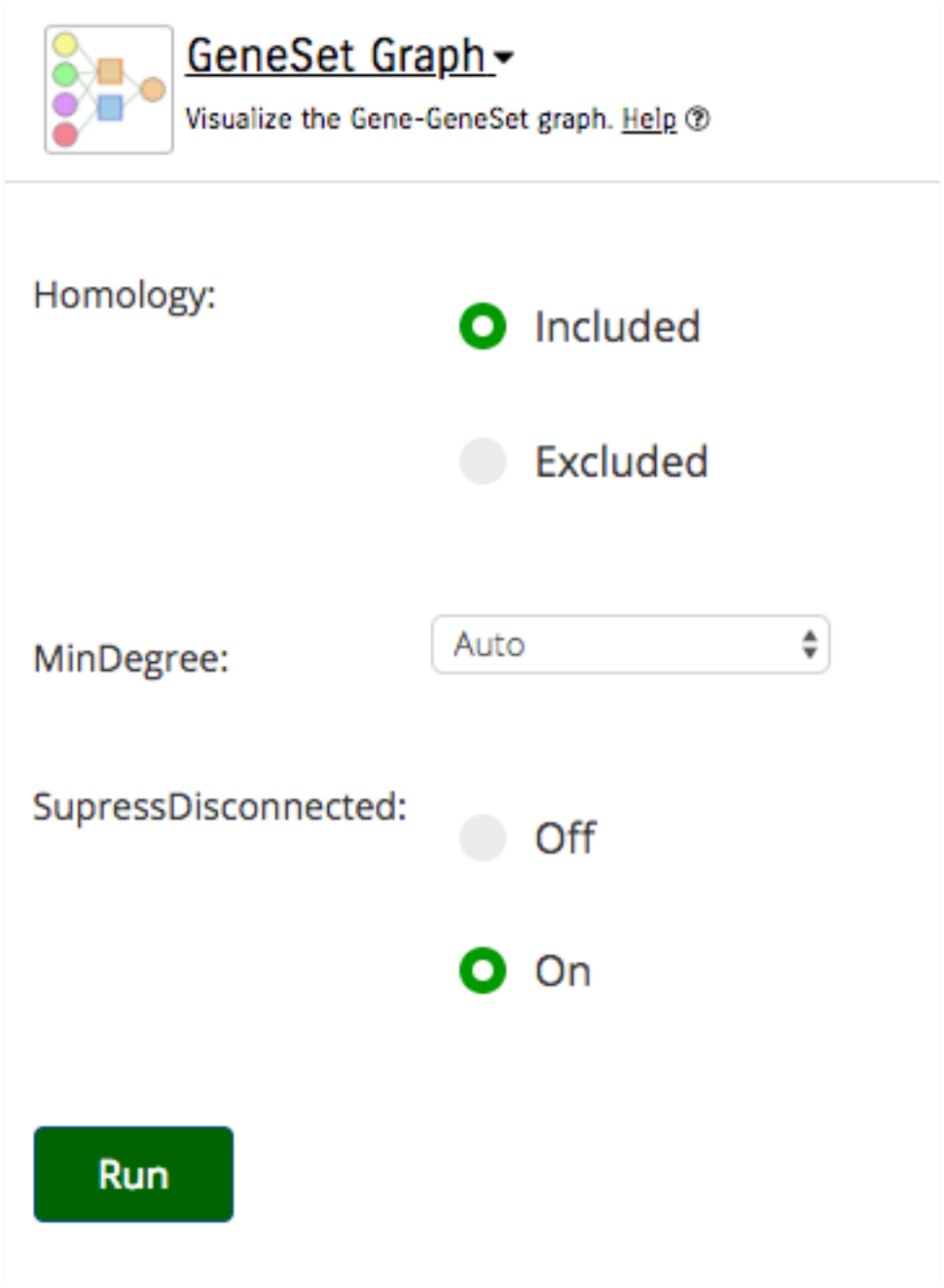


Figure 2: GeneSet Graph Selectino Icon.

The GeneSet Graph can be interactively panned and zoomed with the mouse, and more details of each gene or GeneSet can be viewed by clicking on the individual nodes in the display. In addition to these interactive features, there are also a few options available to optimize the display.

Clicking on a gene node executes a search for other GeneSets containing the gene of interest or its homologues. Clicking on a GeneSet node reveals full publication and annotation information, including the GeneSet description.

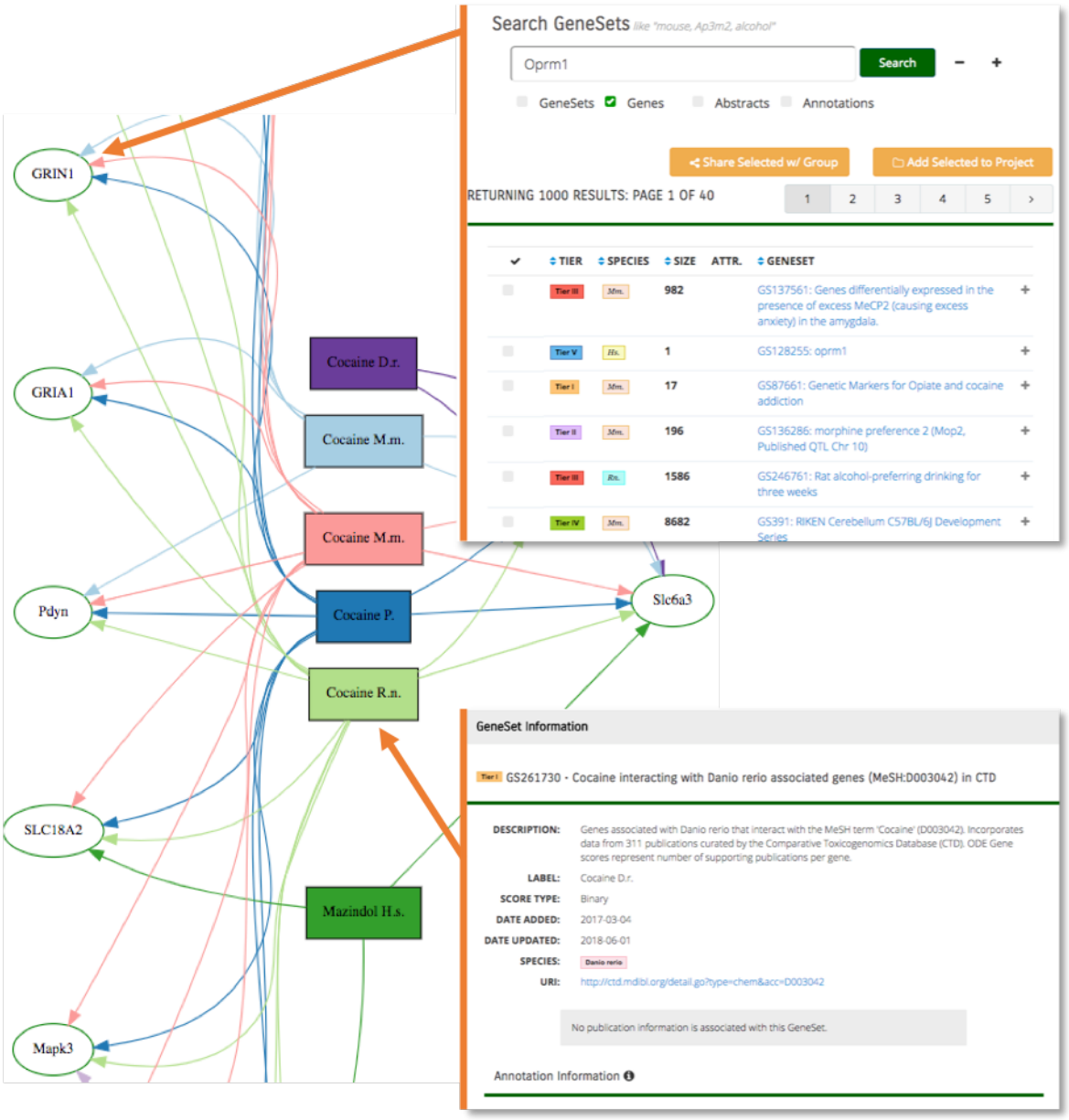


Figure 3: Selecting GeneSets will navigate users to the GeneSet page; selecting the gene will initiate a search of that gene.

Options

Suppress Disconnected

When enabled, this option will suppress the display of GeneSets which are not connected to any displayed genes. This helps remove unnecessary information for users that only want relations. This is only relevant when MinDegree is greater than 1.

Homology

Include homology to integrate multi-species data. If excluded, data from multiple species will be segregated into distinctly separate graphs.

Homology Excluded

Homology Included

Figure 4: 2 GeneSets each

from mouse and rat.

MinDegree

The minimum number of connections for a displayed gene. A value of 2 means that any displayed genes must be found in at least two of the input gene sets. Increasing this value will basically shift the resulting gene display left. Since lower-order overlaps are generally more likely and more numerous than higher-order intersections, this can quickly reduce the number of genes displayed and make the result more manageable.

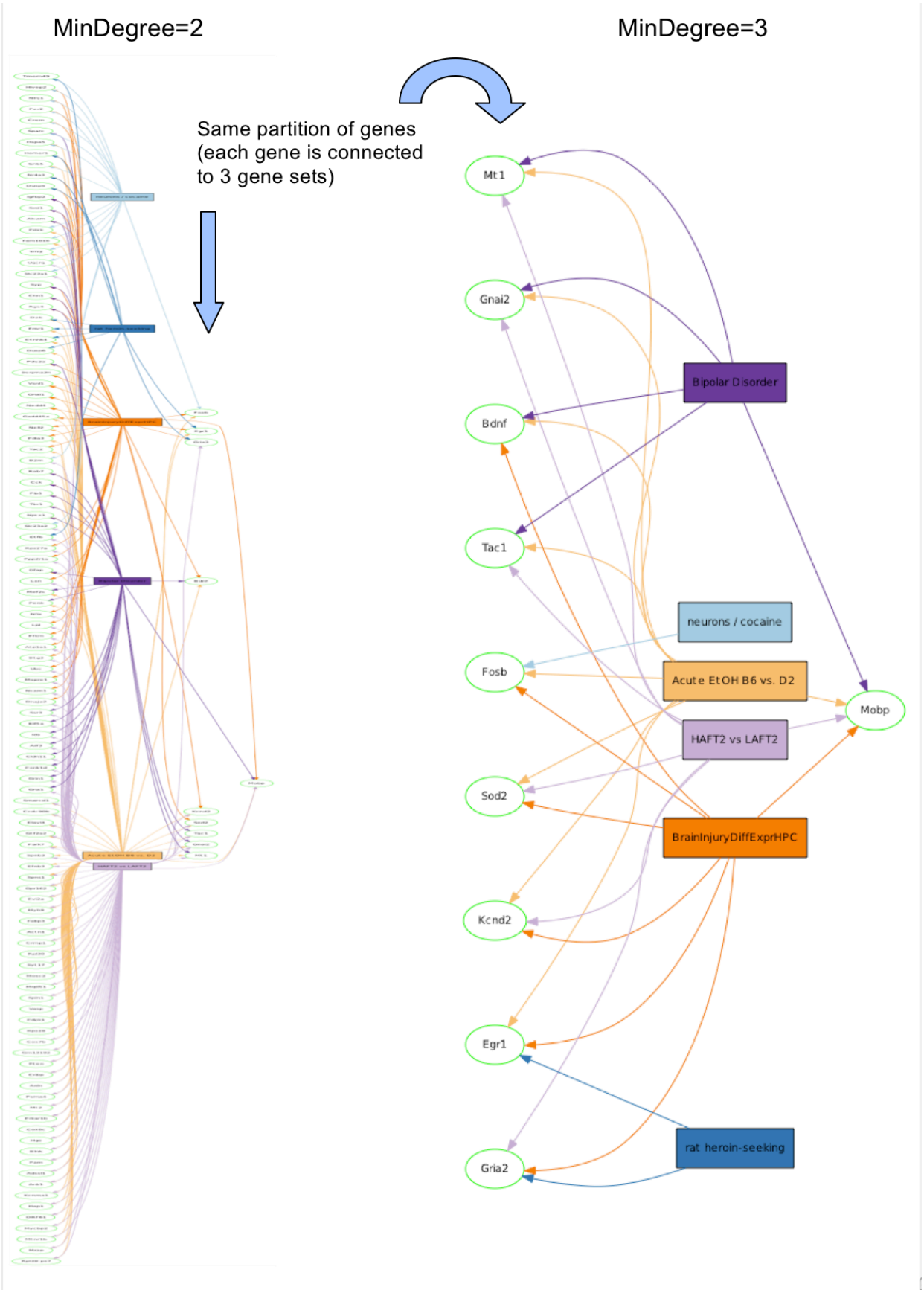

MinDegree=2
MinDegree=3
Same partition of genes
(each gene is connected
to 3 gene sets)
Mt1
Gnai2
Bipolar Disorder
Bdnf
Tac1
neurons / cocaine
Fosb
Acute EtOH B6 vs. D2
Mobp
HAFT2 vs LAFT2
Sod2
BrainInjuryDiffExprHPC
Kcnd2
Egr1
rat heroin-seeking
Gria2

Figure 5

Top ↑

Jaccard Similarity

Why Use the Jaccard Similarity Tool

The Jaccard Similarity Tool displays a matrix of Venn diagrams, which can be very useful for quickly finding overlapping GeneSets and evaluating the similarity of results across a collection of experiments. This snapshot may enable you to determine which can be removed or kept for more complex comparison analysis (such as the HiSim Graph).

Understanding the Jaccard Similarity Tool

Each Venn Diagram represents the pairwise gene overlap between the two GeneSets depicted for each row and column. Text overlays show the exact gene counts, Jaccard Similarity coefficient and p-value for every pair. The p-value is calculated based on the cumulative probability of obtaining a Jaccard coefficient greater than or equal to the observed value, using formula [17] in Real and Vargas, 1996.

For those less knowledgeable of Jaccard Similarity, it's the ratio of elements in both sets over the elements only found in separate sets. If your matrix produces two separate blue and red circles, rather than a touching Venn Diagram, it means nothing is alike in either of those two GeneSets.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$

Figure 7: Jaccard Similarity Equation - source

Background Processes

The Jaccard Similarity Tool now implements the calculation of the p-value for the Jaccard Similarity score based on an empirical sampling distribution. The distribution is approximated for each unique gene set cardinality (gene set size) pair. Each unique pair of cardinalities are randomly sampled (10,000 samples) from the actual gene list of the geneweaver database and plotted based on the frequency of Jaccard Similarity. The result is a Frequency versus Jaccard Similarity histogram that is used as the distribution for the calculation of the p-value. To calculate the p-value, the tool will simply compare the Jaccard Similarity of the user-selected gene set and grade it based on the curve stored in the database.

If the Jaccard Similarity does not exist in the curve - that is, if the Similarity is too high to occur *randomly* - the p-value is simply zero. If the Jaccard Similarity were to have a value of 1, this would indicate that either one is a subset or both are identical. In this case, we assign a special p-value of 1* since we agree that the probability of a set matching itself (and not some other set which contains other genes) will always occur.

The implementation of this process is coded and optimized for C++ which runs in the background as your results are loading onto the next page.

Using the Jaccard Similarity Tool

Access the Jaccard Similarity Tool through the Analyze Genesets tab.

To generate a Jaccard Similarity Matrix, you must first select gene sets from a project. Projects may be created and updated by uploading Gene Sets, searching the GeneWeaver database, or through the use of other tools in the GeneWeaver system. See the documentation for uploading GeneSets, Search, or Manage GeneSets to learn more about these functions. To select an entire project or multiple projects for analysis, check the box next to the project name. To select individual GeneSets within a project, click on the + beside the project name and check individual gene sets using the check boxes. Next, click on the Jaccard Similarity icon in the Analysis tools box to the left of the project list.

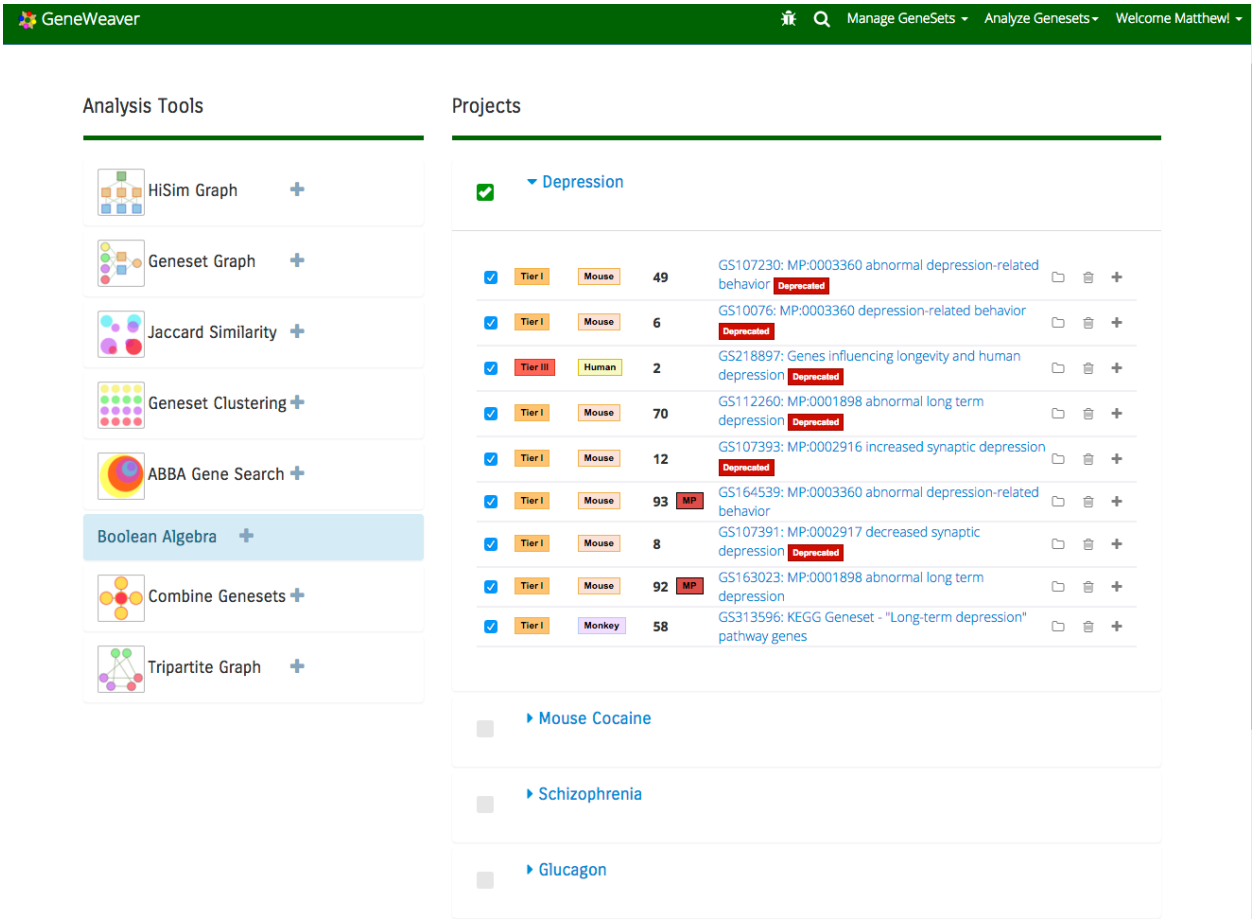


Figure 1: Once you have selected GeneSets from a project, select the **Jaccard Similarity** icon from the Analysis Tools box, to the left of your GeneSets.

Tool results are displayed as a grid of proportional overlaps. The grid, itself, is written in d3 for dynamic user interaction.

Figure 3: Venn diagram for 9 GeneSets. The detail below highlights Column 3, Row 2.

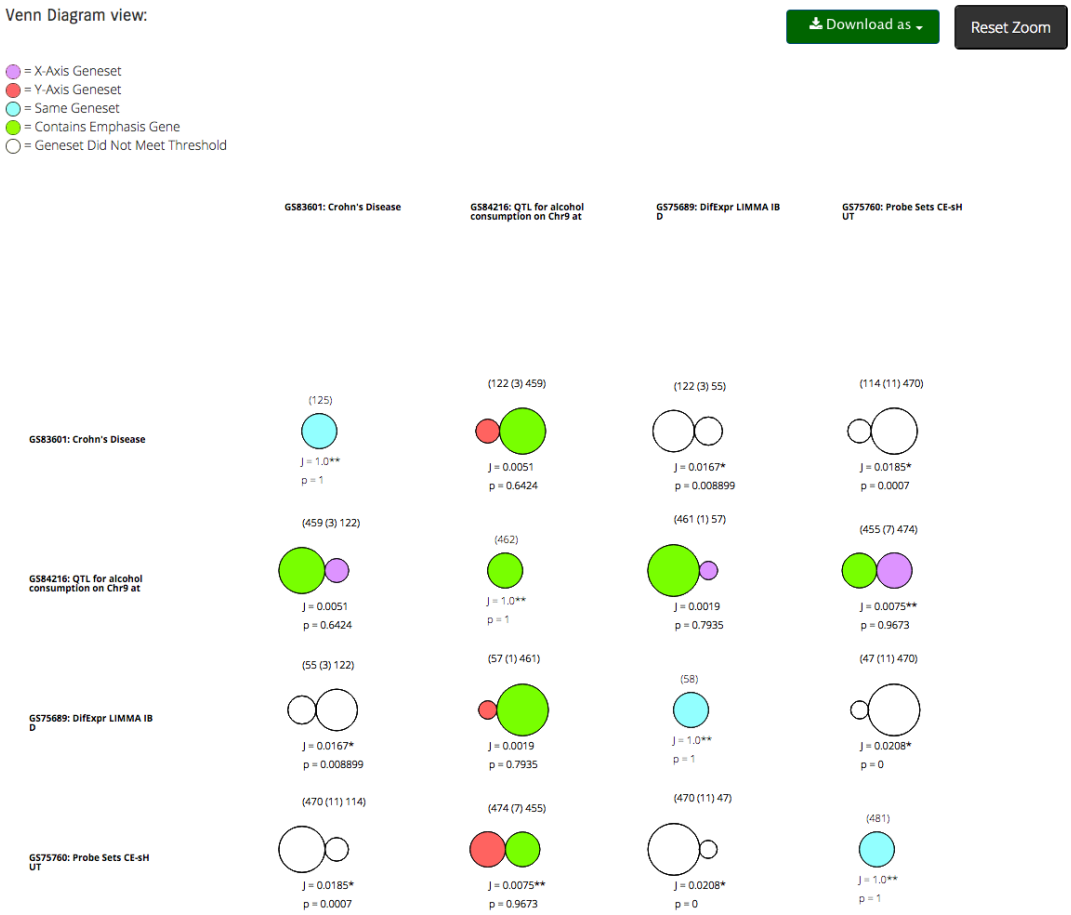

Venn Diagram view:
Download as
Reset Zoom
= X-Axis Geneset
= Y-Axis Geneset
= Same Geneset
= Contains Emphasis Gene
= Geneset Did Not Meet Threshold
GS83601: Crohn's Disease
GS84216: QTL for alcohol consumption on Chr9 at
GS75689: DifExpr LIMMA IB D
GS75760: Probe Sets CE-sH UT
(125)
(122 (3) 459)
(122 (3) 55)
(114 (11) 470)
J = 1.0**
p = 1
J = 0.0051
p = 0.6424
J = 0.0167*
p = 0.008899
J = 0.0185*
p = 0.0007
(459 (3) 122)
(462)
(461 (1) 57)
(455 (7) 474)
J = 0.0051
p = 0.6424
J = 1.0**
p = 1
J = 0.0019
p = 0.7935
J = 0.0075**
p = 0.9673
(55 (3) 122)
(57 (1) 461)
(58)
(47 (11) 470)
J = 0.0167*
p = 0.008899
J = 0.0019
p = 0.7935
J = 1.0**
p = 1
J = 0.0208*
p = 0
(470 (11) 114)
(474 (7) 455)
(470 (11) 47)
(481)
J = 0.0185*
p = 0.0007
J = 0.0075**
p = 0.9673
J = 0.0208*
p = 0
J = 1.0**
p = 1

Figure 8:

Jaccard Overlap

GS row = pink circle (left) GS column = green circle (right)J = Jaccard coefficientp = *p*-valueGreen circles show emphasis genes

The resulting matrix can be zoomed in and out by scrolling the mouse up and down. There is a reset zoom button just in case the user's place is lost in the matrix of venn diagrams. The user can also click and, in addition to these interactive features, the gene sets can be highlighted by row and column by *shift+clicking* on the intersection of two gene sets.

The gene sets can be deselected by alt+clicking on any highlighted gene set.

Rerun Option

The user is able to rerun the tool with different parameters with the rerun tool options.

Figure 7: Rerun tool option

This option is expandable/collapsable by simply clicking on the Rerun Tool Options text.

Geneset Panel

The geneset panel shows the Jaccard coefficients and the p-values for every geneset pair for the project the user has chosen. The geneset panel does not recieve the same reduction as the venn diagram as it would be helpful to still view every geneset pairing for convenience.

The user may also click the checkboxes located next to the geneset names for them to add those selected genesets to a project or to export the genes.

Figure 2: Click *Run* to produce Jaccard Similarity Results for your selected GeneSets. Text overlays show the exact gene counts, Jaccard Similarity coefficient and p-value for every pair.

Options

Homology

Include homology in order to integrate multi-species data. If excluded, homologous genes from different species will not be counted as intersecting. Data from separate species will never show an overlap without homology.

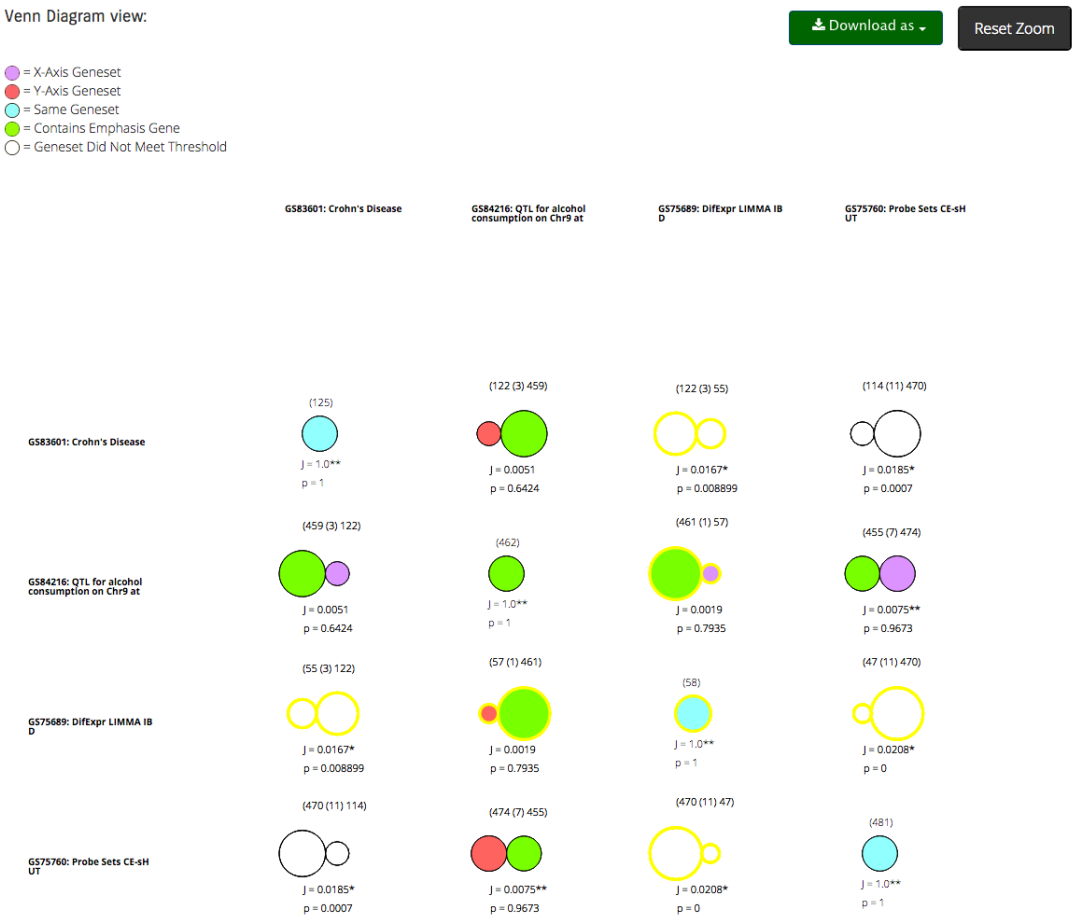


Figure 9: Figure 6: Highlight of row 2, column 3

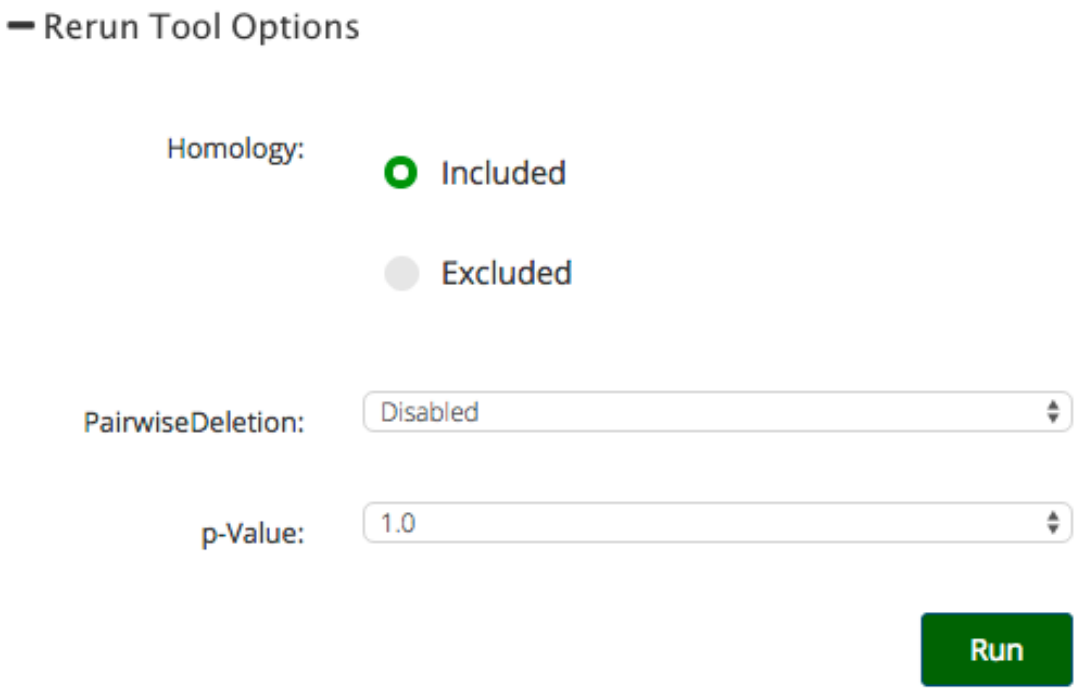

Rerun Tool Options
Homology:
Included
Excluded
PairwiseDeletion:
Disabled
p-Value:
1.0
Run

Figure 10:

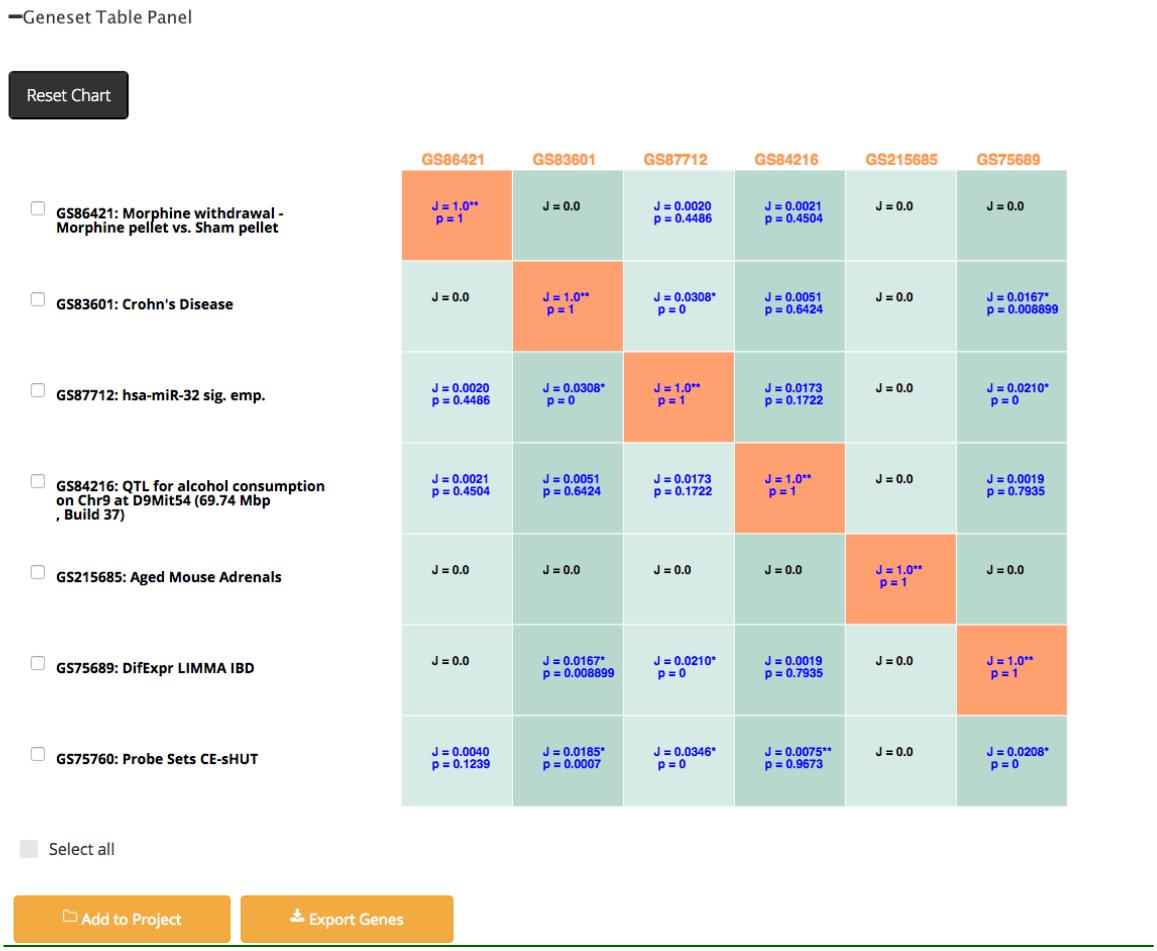

Geneset Table Panel
Reset Chart
GS86421
GS83601
GS87712
GS84216
GS215685
GS75689
GS86421: Morphine withdrawal - Morphine pellet vs. Sham pellet
J = 1.0** p = 1
J = 0.0
J = 0.0020 p = 0.4486
J = 0.0021 p = 0.4504
J = 0.0
J = 0.0
GS83601: Crohn's Disease
J = 0.0
J = 1.0** p = 1
J = 0.0308* p = 0
J = 0.0051 p = 0.6424
J = 0.0
J = 0.0167* p = 0.008899
GS87712: hsa-miR-32 sig. emp.
J = 0.0020 p = 0.4486
J = 0.0308* p = 0
J = 1.0** p = 1
J = 0.0173 p = 0.1722
J = 0.0
J = 0.0210* p = 0
GS84216: QTL for alcohol consumption on Chr9 at D9Mit54 (69.74 Mbp , Build 37)
J = 0.0021 p = 0.4504
J = 0.0051 p = 0.6424
J = 0.0173 p = 0.1722
J = 1.0** p = 1
J = 0.0
J = 0.0019 p = 0.7935
GS215685: Aged Mouse Adrenals
J = 0.0
J = 0.0
J = 0.0
J = 0.0
J = 1.0** p = 1
J = 0.0
GS75689: DifExpr LIMMA IBD
J = 0.0
J = 0.0167* p = 0.008899
J = 0.0210* p = 0
J = 0.0019 p = 0.7935
J = 0.0
J = 1.0** p = 1
GS75760: Probe Sets CE-sHUT
J = 0.0040 p = 0.1239
J = 0.0185* p = 0.0007
J = 0.0346* p = 0
J = 0.0075** p = 0.9673
J = 0.0
J = 0.0208* p = 0
Select all
Add to Project
Export Genes

Figure 11:

PairwiseDeletion

Pairwise Deletion is used to pick off problematic missing values from data while still aiming to get the remaining values for comparison-based use:

Values	Obj1	Obj2	Obj3
Length	23	N/A	13
Width	21	22	14
Depth	N/A	20	11

Figure 7: In Pairwise Deletion, when comparing length, only Obj1 and Obj3 will be compared. When comparing width, all will be compared, and when comparing depth, only Obj2 and Obj3 can be looked at. This prevents missing data from being assigned a default value such as 0 in the system.

Top ↑

Clustering

Why Use the Clustering Tool

Clustering is one of the most powerful tools in bioinformatics, where classifications are too strict for data distinction, clustering helps give the user an evaluation that is not so distinct.

Using the Clustering Tool

1. Select the gene sets from your list of projects that you would like to analyze.
   - You need a minimum of 3 gene sets in total to run the tool.
2. Select if homology is to be included or excluded.
   - Homology is included by default.
3. Select the method of clustering.
   - Average is the default method of clustering.
   - There are five methods of clustering. They are listed in the methods section.

Understanding your Results

Visualization Types There are two methods for visualizing your clustering results.

Force Directed Graph

- Tree representation of each cluster.
- Clear depiction of hierarchy.
- The most opaque node of a tree represents the clusters root.
- Each node is classified as one of the following:
  - **Cluster** - Grouping of gene sets
    - The opacity of the nodes is based on the Jaccard Similarity of its children. The more similar the gene sets, the darker the cluster.
    - On Hover: Reveals Jaccard Similarity of its child nodes. Reveals set notation of the containing hierarchy.

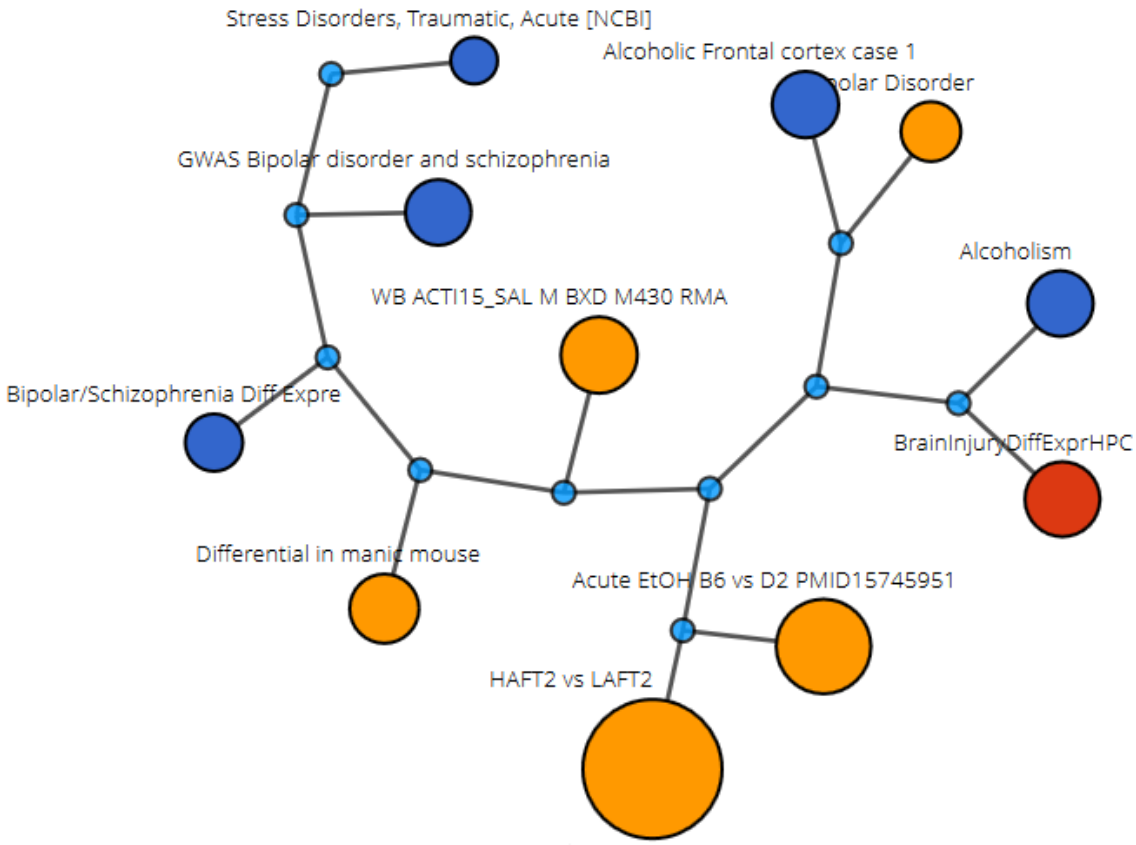

Stress Disorders, Traumatic, Acute [NCBI]
Alcoholic Frontal cortex case 1
GWAS Bipolar disorder and schizophrenia
Alcoholism
WB ACTI15_SAL M BXD M430 RMA
Bipolar/Schizophrenia Diff Expre
BrainInjuryDiffExprHPC
Differential in manic mouse
Acute EtOH B6 vs D2 PMID15745951
HAFT2 vs LAFT2

Figure 12:

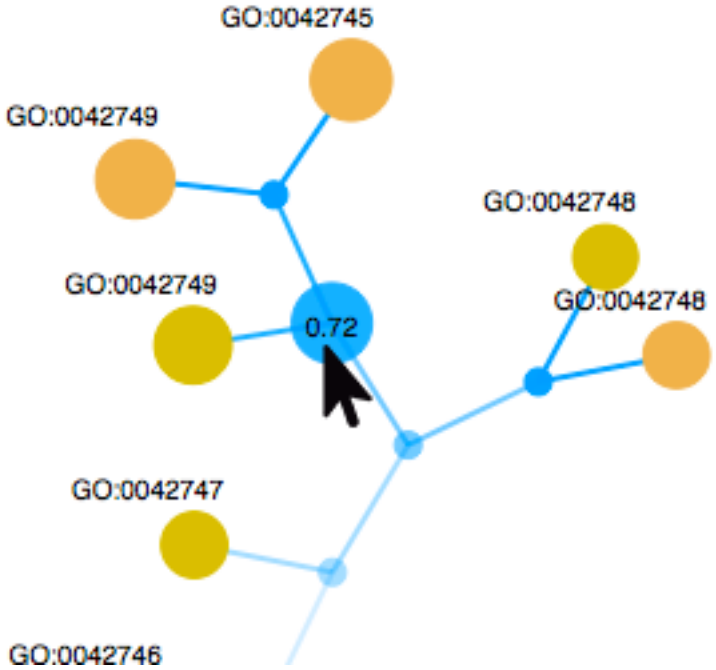


Figure 13:

Figure 14:

- `On Click: Collapses (absorbs its children).`
  - **Gene Set** - A set of genes
    - Colored based on the species contained in the gene set study.
    - Sized based on the relative size of the gene set.
    - On Hover: Reveals abbreviated gene set information.
    - On Click: Reveals and cycles through genes in groups of ten.
    - On Double Click: Opens a new page containing extensive gene set information.
  - **Gene**
    - On Hover: Reveals the name of the gene.
  - **Edges**
    - Connects nodes to its children.
    - The opacity of edges leading from cluster nodes is based on the cluster nodes Jaccard Similarity, following the same scale as above.

Partitioned Sunburst

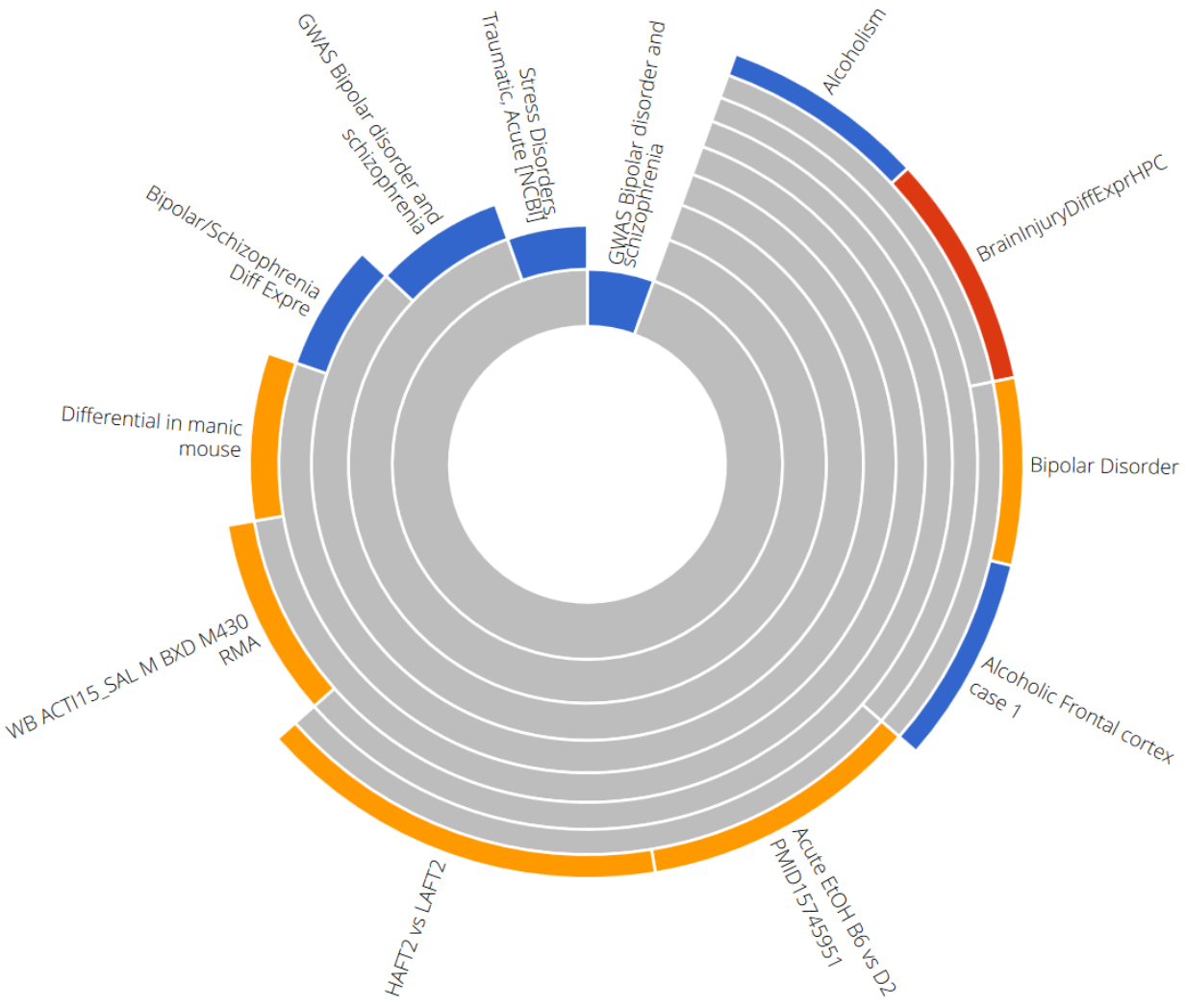


Figure 15:

- Top-down view of each tree.

- Center represents the root.
- Partitioned sub-circles represent clusters, gene set or gene.
- **Partition**
  - Partitions are the equivalent to nodes in a tree
  - Each parition is classified as one of the following:
    - **Cluster** - Grouping of gene sets
      - On Hover: Reveals Jaccard Similarity of its child partition and highlights all nodes within the cluster.
      - On Right Click: Opens a new "View GeneSet Overlap" page using all gene sets in the cluster as input.
    - **Gene Set** - A set of genes
      - Colored based on the species contained in the gene set study.
      - Drawn arc sizes are based on the relative size of the gene set.
      - On Right Click: Opens a new "View GeneSet Details" page for the given gene set.
- **Rings**
  - Each Ring represents a level in the tree.
  - The outer most levels are gene sets.
  - The levels leading up to a gene set represents the hierarchy of the cluster.

Clustering Methods

Listed below are the six different methods that the user can choose from while running the tool. The first five are different clustering methods that will run on the selected genesets and display a force directed tree and a partitioned sunburst based on the clustered genesets.

All five of the given clustering methods are agglomerative hierarchical clustering methods that start with each geneset belonging to its own cluster. They then combine the clusters at each iteration based off of a described linkage method that determines how the distance between two clusters is defined. The clusters are combined until there are no more clusters that are similar to each other (the distance between them is too large).

McQuitty

The McQuitty clustering method uses a linkage method where distance depends on the combination of clusters instead of the individual genesets within each cluster. When two clusters are joined together, the distance of the new cluster to any other cluster is calculated as the average distance between the two clusters that are being joined and the other cluster. For example, if clusters 2 and 4 have the greatest similarity and we are going to combine them into a new cluster called 2+4, then the distance from 2+4 to 1 is the average of the distances from 2 to 1 and 4 to 1.

- **Algorithm**
  - Each gene set is initialized as its own cluster.
  - The initial similarity between each cluster is the Jaccard Similarity of the two genesets.
  - While we still have similar clusters:
    - Clusters with highest similarity are clustered together.
    - Calculates the similarity between the new cluster and all the rest based on the McQuitty linkage method
- **Time Complexity**
  - $O(n^2 \log n)$
  - This method is the most time efficient.

Ward

The Ward clustering method uses a linkage method where the distance between two clusters is based off of the Jaccard Similarity score between them. When two clusters are joined together, the new cluster will take the union of the genesets in the two clusters that are being joined and set that as its geneset. It will then calculate the new geneset's similarity score against all the other cluster's genesets and that will be set as the distance between the new cluster and all the other clusters.

- **Algorithm**
  - Each gene set is initialized as its own cluster
  - The initial distance between clusters is the Jaccard Similarity score between each of the cluster's genesets
  - While we have clusters that are similar to each other:
    - Clusters with highest similarity are clustered together.
    - The new cluster contains a geneset which is the union of its children's genesets
    - Recalculates the Jaccard Similarity score between the new cluster and all the other clusters
- **Time Complexity**
  - $O(n^3)$

Complete

The Complete clustering method uses a linkage method where the distance between two clusters is the lowest similarity score between any of the genesets in one cluster compared to any of the genesets in the other cluster. When two clusters are combined, the genesets within each of the clusters are put into a new cluster. No new calculations are needed at each iteration because we are simply reusing the similarity scores of all the genesets compared to each other.

- **Algorithm**
  - Each gene set is initialized as its own cluster.
  - The similarity scores off all the genesets compared to each other are saved in a matrix
  - While we still have clusters that are similar:
    - Determine which two clusters to join:
      - The distance between two clusters is the lowest similarity score between a geneset in one cluster and a geneset in the other cluster
      - The highest of these distances determines which two clusters will be joined
    - Combines the two clusters to create a new cluster that has all the genesets that were present in the two children clusters
- **Time Complexity**
  - $O(n^3)$

Average

The Average clustering method uses a linkage method where the distance between two clusters is the average similarity score between all of the genesets in one cluster compared to all of the genesets in the other cluster. When two clusters are combined, the genesets within each of the clusters are put into a new cluster. No new calculations are needed at each iteration because we are simply reusing the similarity scores of all the genesets compared to each other.

- **Algorithm**
  - Each gene set is initialized as its own cluster.
  - The similarity scores off all the genesets compared to each other are saved in a matrix
  - While we still have clusters that are similar:
    - Determine which two clusters to join:

            - The distance between two clusters is the average similarity score between every geneset in one cluster and every geneset in the other cluster
            - The highest of these distances determines which two clusters will be joined
        * Combines the two clusters to create a new cluster that has all the genesets that were present in the two children clusters
- **Time Complexity**
    - $O(n^3)$

Single

The Single clustering method uses a linkage method where the distance between two clusters is the highest similarity score between any of the genesets in one cluster compared to any of the genesets in the other cluster. When two clusters are combined, the genesets within each of the clusters are put into a new cluster. No new calculations are needed at each iteration because we are simply reusing the similarity scores of all the genesets compared to each other.

- **Algorithm**
    - Each gene set is initialized as its own cluster.
    - The similarity scores off all the genesets compared to each other are saved in a matrix
    - While we still have clusters that are similar:
        * Determine which two clusters to join:
            - The distance between two clusters is the highest similarity score between any geneset in one cluster and any geneset in the other cluster
            - The highest of these distances determines which two clusters will be joined
        * Combines the two clusters to create a new cluster that has all the genesets that were present in the two children clusters
- **Time Complexity**
    - $O(n^3)$

Top ↑

DBSCAN Gene Clustering

What is DBSCAN?

DBSCAN (Density-Based Spatial Clustering of Application with Noise) is a clustering algorithm that groups genes into clusters based on how closely related the genes are.

Why Use the DBSCAN Tool?

In general, clustering is used to find patterns or outliers within data sets. In this implementation of DBSCAN, genes in the same cluster would be considered similar, while genes in different clusters would be less similar. An explanation of DBSCAN can be found here. Within Geneweaver, this tool can be used to infer relationships between genes. For example, if clusters with similar genes continue to appear in tests across multiple data sets, one could say that these genes are closely related.

DBSCAN Parameters

DBSCAN takes in 2 parameters, epsilon and minPoints.

The Epsilon Parameter

Epsilon determines how close the genes need to be in order to be considered in the same cluster. For example, an epsilon of 1 means that genes need to share at least 1 gene set. Another way of describing epsilon would be the "radius of the neighborhood". A larger epsilon will have a farther reach when finding clusters.

The minPoints Parameter

The minPoints parameter determines the minimum number of points required to form a cluster. A cluster can have more than the minPoints number of genes, but cannot be less than minPoints. If a cluster has less than minPoints number of genes, it is considered noise.

The DBSCAN Algorithm

Before the DBSCAN algorithm executes, it must determine how closely related each gene is to the other genes. A bipartite graph is used to show how the genes connect to each gene set. First, all closest paths between genes are found. Following that, the DBSCAN algorithm is run. You can find an example of DBSCAN here.

Run Times of DBSCAN

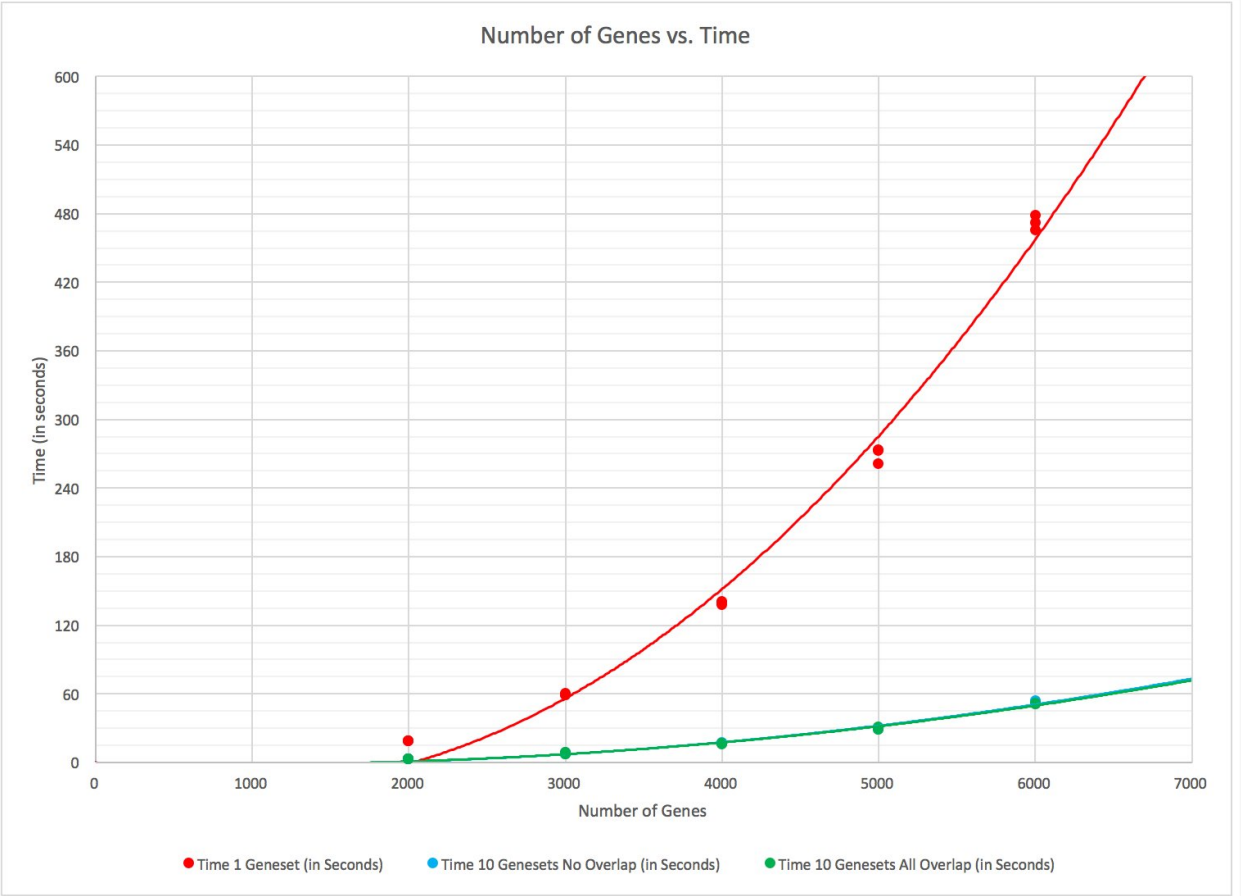
On average, the worst-case time complexity of DBSCAN is $O(n^2)$. However, due to the sheer variability of data sets and epsilon and minPoints combinations, it is difficult to accurately predict the run time of this implementation. There are some factors that will typically increase the run time. These include:

- Number of Genes: If more genes are tested, the run time is longer
- Epsilon Value: A larger epsilon will typically give a longer run time
- The size of gene sets: Gene sets with more genes in them will take longer to explore
- The density of genes: If the data set is denser (more connections), the run time is longer

Note: Even if no clusters are found, the algorithm may still take time to execute.

Below is a graph that shows the run times of the algorithm. The red line shows the run time if all genes are in the same gene set. The blue line shows the genes divided into 10 gene sets, with no overlap. The green line is similar to the blue line, but here the gene sets share one gene in common with one other gene set. This results in one giant cluster with all of the genes.

Note: Since the blue line and green line overlap, you may not be able to see the blue line.



Below is a table that estimates the run time of the red, blue, and green cases based on number of genes. Note that run times will change based on density of the gene sets and epsilon.

Number of Genes	1 Gene Set	10 Gene Sets, No Overlap	10 Gene Sets, Overlap
100	3	3	3
200	3	3	3
500	5	3	3
1,000	10	3	3
1,500	12	3	3
2,000	15	3	3
2,500	28	5	5
3,000	63	8	8
3,500	110	12	12
4,000	160	17	18
4,500	230	24	25
5,000	306	32	33
6,000	487	50	51
7,000	708	72	75
8,000	969	98	100
9,000	1270	129	131
10,000	1612	163	165

Approximate DBSCAN Run Times with Epsilon = 1 and Min Points = 1 (in seconds)

Visualization

Once DBSCAN is completed, results can be visualized in two ways. However, there is a possibility that visualization may not occur. If a data set is too large, the results will not be visualized and a message will be displayed.

> Note: Due to the rendering of the Cluster / Gene Table, run times may appear longer than estimated in here.

Circles

The default visualization on the tool is circle packing. This represents the clusters and the genes within them. The outermost circle is the entire data set. The darker blue circles within represent the different clusters. The circles within the clusters represent the genes that belong to the cluster. The color of each gene denotes the species.

To see more information about the cluster, you can click on the cluster. This will zoom in on the cluster and display gene IDs. Clicking on a gene ID will redirect to a search for that gene within the GeneWeaver database.

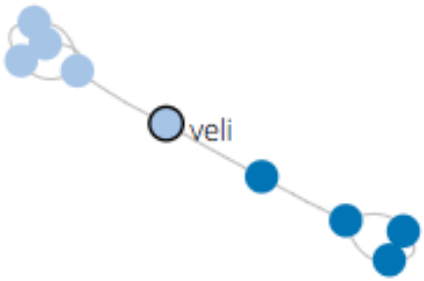
Below is an example of the circle packing visualization with zoom functionality.

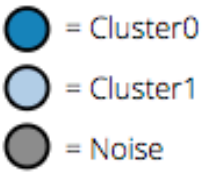
Wires

The other visualization is a wire representation. This shows the connections between all genes in the same gene set. The color of each gene shows which cluster the gene is in. If a gene is grey, it is considered noise. Mousing over a circle will highlight it and show the gene ID. By clicking and and holding a gene, you can drag the gene around the screen.

> Note: This visualization will only be drawn with small data sets due to the complexity of drawing all lines between genes.

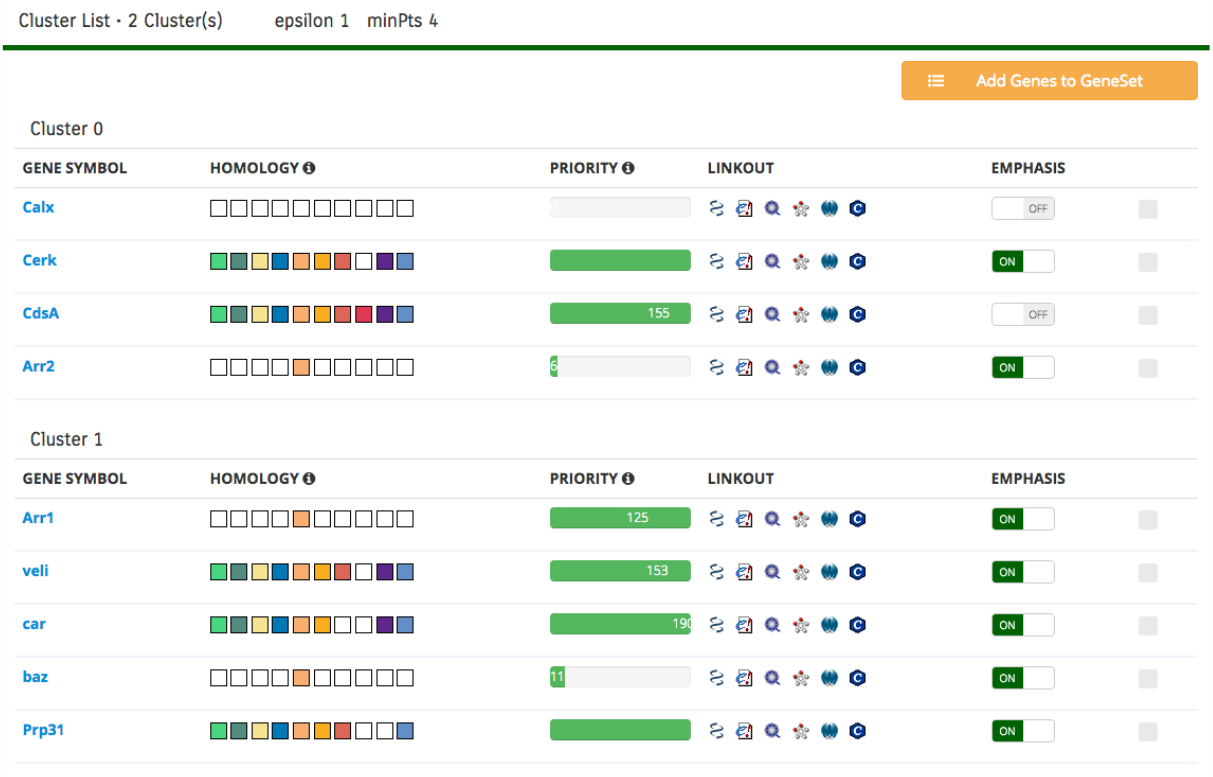
Below is an example of the wires visualization.





Cluster / Gene Table

Below the visualizations is a table. This table is split up into clusters, which contains all the genes within that specific cluster. Information about each gene can be seen here as well. This table is similar to the one on the **GeneSet Details** page.



If the data set becomes sufficiently large, a minimized table will be shown on screen. An example of the minimized table is below.

Cluster 0

GENE SYMBOL	LINKOUT	GENE SYMBOL	LINKOUT	GENE SYMBOL	LINKOUT
GFRA1		WDR75		TRIM38	
IL22RA1		TRIB3		IL11	
FBLN2		CTSK		MRPL32	
H2AFY2		SENP6		ZNF404	
CDKN2B		TMEM190		IFNLR1	
GALNT7		PLCB4		LTBP3	
GTF2H2C		CYP8B1		ZNF80	
EPGN		MALL		LRRC37A2	
ZNF45		SFT2D1		DBR1	
NFE2L1		SCNN1D		CSNK1A1L	
SLC27A6		PLEKHG3		ATP6AP1L	
SRP9P1		SEC14L4		RBCK1	
TRIM24		HIST1H1A		NUMBL	
GDF9		PAK1IP1		ELAVL2	

DBSCAN Example

Below is an example of the DBSCAN algorithm. For this example, epsilon is set to 1 and min-points is set to 4. Figure 1 shows the gene-to-gene set bipartite graph.

Figure 1: The gene-to-gene set bipartite graph

Finding Shortest Paths Between Genes

Starting at "Test Set 0" Prp31, Arr1, baz, and car are all in the same gene set. This means that when building the gene-to-gene graph, all of those genes will be connected to each other. "Test Set 1" shows that Arr1 and veli are connected. "Test Set 2"has veli and Arr2 connected. "Test Set 3" has Arr2 connected to CalX. Finally, "Test Set 4" has CalX, CdsA, and Cerk connected. Now that the connections between genes are determined, a map can be drawn showing these connections (Figure 2).

Figure 2: The gene-to-gene graph denoting shortest paths

Using this graph, the shortest path from a gene to any other gene can be determined. For example, the distance between Arr1 and baz is 1. The distance between Prp31 and CalX is 4. This is important when applying epsilon to the algorithm.

Running the DBSCAN Algorithm

This is the pseudocode for the algorithm.

Starting in the DBSCAN function, the cluster is first initialized to 0. Next, each point is visited only once. For this example, baz will be the first gene visited. baz will be first be marked as visited, then the neighbors of baz will be found by regionQuery. The regionQuery function will return all points within radius epsilon, including the point itself. Calling regionQuery on baz with epsilon will return all genes that are one away from baz. In this example baz, car, Prp31, and Arr1 are returned and listed as baz's neighbors.

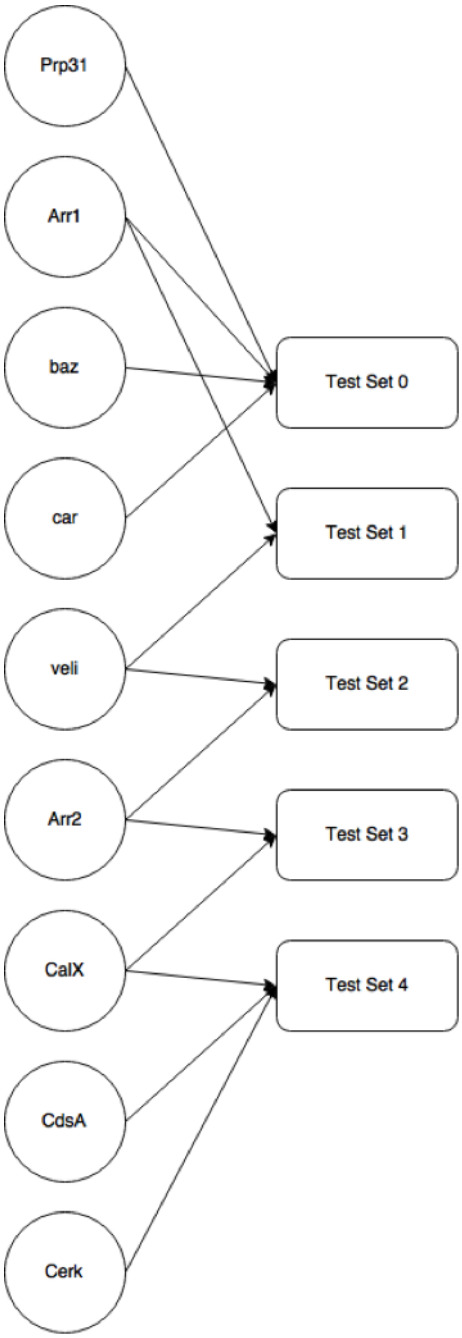

Prp31
Arr1
baz
car
veli
Arr2
CalX
CdsA
Cerk
Test Set 0
Test Set 1
Test Set 2
Test Set 3
Test Set 4

Figure 16:

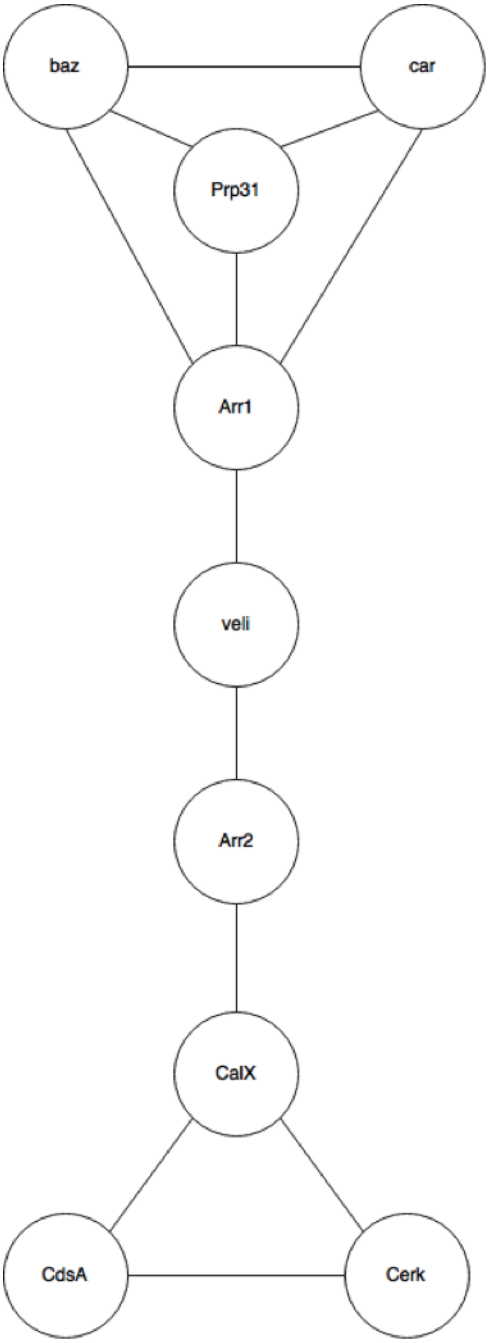


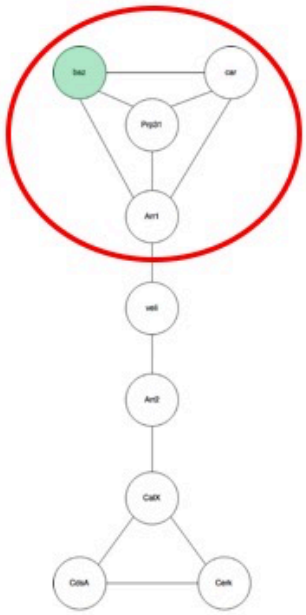
Figure 17:

```
DBSCAN(D, eps, minPts) {
        Clusters = [ ]
        for each point P in dataset D {
                if P is visited
                        continue next point
                mark P as visited
                NeighborPts = regionQuery(P, eps)
                if sizeof(NeighborPts) < minPts
                        mark P as NOISE
                else {
                        C = new cluster
                        expandCluster(P, NeighborPts, C, eps, minPts)
                        Add to C to list of Clusters
                }
        }
}

expandCluster(P, NeighborPts, C, eps, minPts) {
        add P to cluster C
        for each point P' in NeighborPts {
                if P' is not visited {
                        mark P' as visited
                        NeighborPts' = regionQuery(P', eps)
                        if sizeof(NeighborPts') >= minPts
                                NeighborPts = NeighborPts joined with NeighborPts'
                }
                if P' is not yet member of any cluster
                        add P' to cluster C
        }
}

regionQuery(P, eps)
        return all points within P's eps-neighborhood (including P)
```

Figure 18: DBSCAN Pseudocode



Epsilon = 1
Min Points = 4

Cluster(s):

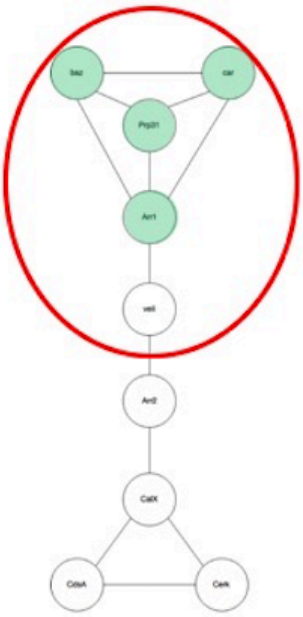
Neighbors List:
Baz
Car
Prp31
Arr1

The list of [baz, car, Prp31, Arr1] are returned. Now the amount of items in the list is checked with the minPoints parameter. If it is greater than or equal to minPoints, a cluster is formed. Otherwise, the point is labelled as noise. In this example, baz has 4 neighbors, which is equal to the number of points. The "C = next cluster" statement means that C is a valid cluster. Next, the expandCluster function is called.

The expandCluster will continue to expand the cluster until the edge of the cluster is reached. The edge of a cluster is reached when a point has a list of neighbors that is less than the number of minPoints. When entering the expandCluster function, the point P will be added to the cluster. The cluster is currently [baz]. Next, the algorithm runs through all of the neighbors to see if the cluster can be expanded. The list of neighbor points is now [baz, car, Prp31, Arr1]. First baz is looked at, but because it has already been visited, it is not going to be checked again. Next, car is checked. Car will then return a list of all its neighbors, which are [car, baz, Prp31, Arr1]. Then that list is checked against the number of minPoints. Since it is greater than or equal to minPoints, that list is added to the original list of neighbors. So the original neighbors list of [baz, car, Prp31, Arr1] and the new neighbors list of [car, baz, Prp31, Arr1] are added together. However, the algorithm does not add duplicate genes to the list. Therefore, nothing is added to the list and the neighbors list is [baz, car, Prp31, Arr1]. Then, the gene is added to the current cluster if it is not already part of a cluster. car is not a part of any other cluster so it is added to the current cluster. Now the cluster contains [baz, car].

Next, Prp31 is looked at. Its neighbors are [baz, car, Prp31, Arr1]. This list is equal to minPoints, but once again, the list of Prp31's neighbors are already in the list of baz's neighbors. So nothing is added to new neighbors, and since Prp31 is not a part of any other cluster, it is added to the current cluster, which is now [baz, car, Prp31].

Now, Arr1 is looked at. Its neighbors are [Arr1, baz, car, Prp31, veli]. Notice that a new gene appeared in Arr1's neighbors (veli). This gene is now added to the list of baz's neighbors. Arr1 is added to the current cluster, so the cluster now holds [baz, car, Prp31, Arr1]. Now there is still one gene left to check in baz's neighbors, which is veli.

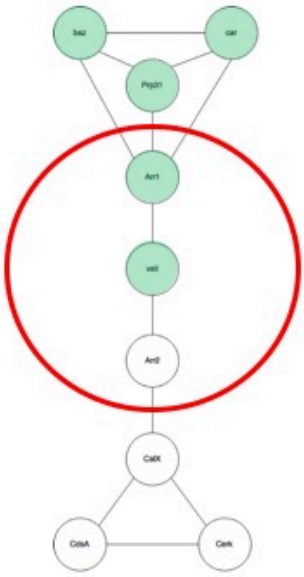


Epsilon = 1
Min Points = 4

Cluster(s):

Neighbors List:
~~Baz~~
~~Car~~
~~Prp31~~
~~Arr1~~
veli

veli is checked and it's neighbors are [veli, Arr1, Arr2]. The list is less than the number of minPoints, which means the cluster cannot be expanded past veli.

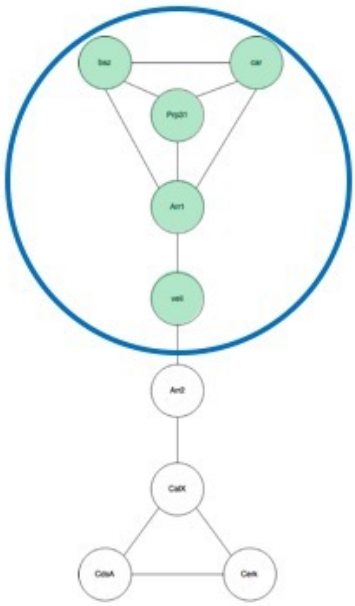


Epsilon = 1
Min Points = 4

Cluster(s):

Neighbors List:
~~Baz~~
~~Car~~
~~Prp31~~
~~Arr1~~
~~veli~~

However, veli is still part of the current cluster. The current cluster is now [baz, car, Prp31, Arr1, veli]. Since the list of baz's neighbors have all been checked, the cluster is finished.

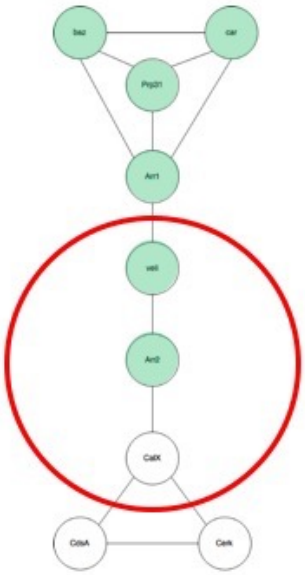


Epsilon = 1
Min Points = 4

Cluster(s):
[baz, car, Prp31, Arr1, veli]

Neighbors List:

Now that baz has been checked, it is time to check other genes. Next, car is checked. However, it was already visited when handling baz's neighbors, so nothing needs to be checked. The same applies for Prp31, Arr1, and veli. The next gene to check is Arr2. Arr2's neighbors are [veli, Arr2, CalX]. This is less than minPoints, so it is marked as noise.

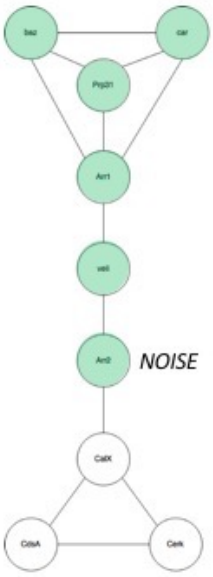


Epsilon = 1
Min Points = 4

Cluster(s):
[baz, car, Prp31, Arr1, veli]

Neighbors List:

However, just because a gene is marked is noise, does not guarantee it is noise when the algorithm is finished. Later in the algorithm, it can be added to a cluster.

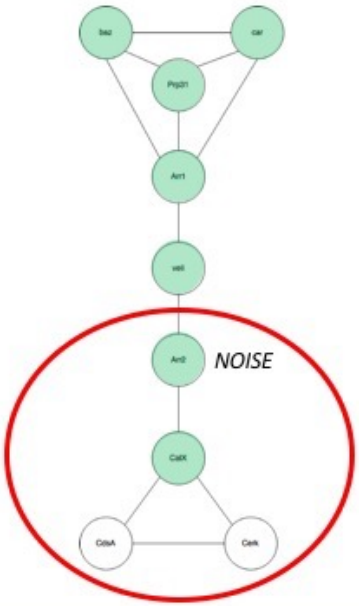


Epsilon = 1
Min Points = 4

Cluster(s):
[baz, car, Prp31, Arr1, veli]

Neighbors List:

Next, CalX is checked. It's neighbors are [CalX, Arr2, CdsA, Cerk]. This list is equal to minPoints, so the cluster needs to be expanded.



Epsilon = 1
Min Points = 4

Cluster(s):
[baz, car, Prp31, Arr1, veli]

Neighbors List:
Arr2
CalX
CdsA
Cerk

CalX is checked, but it is already visited, and it is not a part of any cluster, so it is added to the 2nd cluster. The 2nd cluster currently holds [CalX]. Next, Arr2 is checked, but it was already visited and marked as noise. However, it is not in any cluster, so it is added to the 2nd cluster. The 2nd cluster now contains [CalX, Arr2]. Next, CdsA is checked. Its neighbors are [CdsA, Cerk, CalX]. This list is not greater than minPoints so nothing is added. CdsA is not added to the 2nd cluster because it is not part of the first cluster. The 2nd cluster is now [CalX, Arr2, CdsA]. Finally, Cerk is checked. Its neighbors are [CdsA, CalX]. The list is smaller than minPoints, so they are not added to Calx's neighbors. Cerk is not a part of any cluster, so it is added to the 2nd cluster. The 2nd cluster is now complete. It contains [CalX, Arr2, CdsA, Cerk].

Now that CalX is checked, CdsA is checked. It was already visited in the expandCluster function so nothing

needs to be done. The same applies for Cerk. The algorithm is now complete.

Two clusters were produced: [baz, car, Prp31, Arr1, Veli] and [Arr2, CalX, CdsA, Cerk]

Figure 3 shows the gene-to-gene map visualized in clusters.

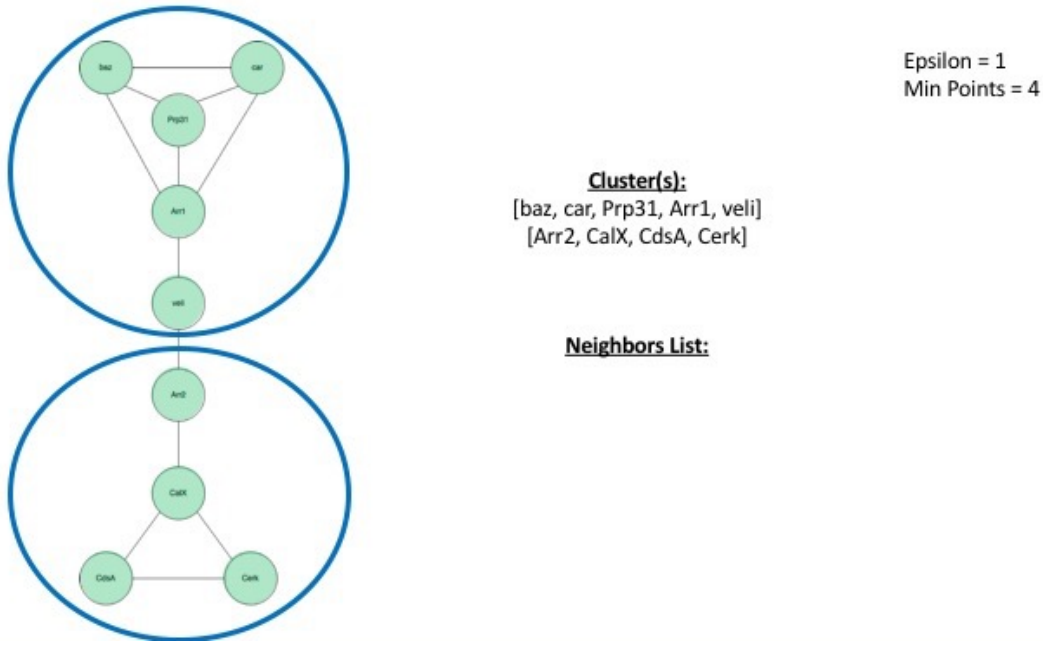


Figure 3: The result of the DBSCAN clustering

Top ↑

MSET

Modular single-set enrichment tool (MSET): randomization-based test for list over- or under-representation

About MSET

MSET was developed to compare gene lists. From four character lists (gene_list1, gene_list2, background1, background2), it computes a randomization-based p-value describing the likelihood that the intersect of **gene_list1** and **gene_list2** is underexpressed or overexpressed relative to randomness alone.

MSET is based on work from Eisinger et al., 2013, "Development of a versatile enrichment analysis tool reveals associations between the maternal brain and mental health disorders, including autism." BMC Neuroscience.

Why MSET?

MSET permits the selection, or customization, of the genes against which enrichment is performed. This yields the ability to perform more focused hypothesis testing relative to other enrichment tests. For example, genes specific to Alzheimer's may be selected to serve as the genes of interest against which enrichment testing is performed.

How Does MSET Work?

MSET performs enrichment testing using several entities:

User Selected:

- **Gene Set 1**: The first set of genes to perform MSET on
- **Gene Set 2**: The second set of genes to perform MSET on
- **Number of Trials**: The number of simulated sets to create

MSET Computed:

- **Gene Set 1 Background**: Determined from Gene ID Type and Species of Gene Set 1
- **Gene Set 2 Background**: Determined from Gene ID Type and Species of Gene Set 2
- **The Universe**: The intersection of **Gene Set 1 Background** and **Gene Set 2 Background**
- **Gene Set 1-U**: Genes in **Gene Set 1** that are also contained in the **The Universe**
- **Gene Set 2-U**: Genes in **Gene Set 2** that are also contained in the **The Universe**

MSET then takes the following steps:

1. First, the computed inputs are calculated,
2. Then, MSET calculates the v (said another way, it counts the number of shared genes)
3. For the **Number of Trials**, MSET then samples randomly without replacement from **The Universe** to generate two simulated gene sets of sizes **Gene Set 1-U** and **Gene Set 2-U** respectively,
   - For each trial, the intersection of the two simulated gene sets is recorded
4. MSET then calculates the p-value as:

$$\text{p-value} = \frac{\left(\begin{array}{c}\text{Number of Trials w/ Intersect of Simulations} \\ \text{Greater Than Intersect of Observed}\end{array}\right)}{\text{Total Number of Trials}}$$

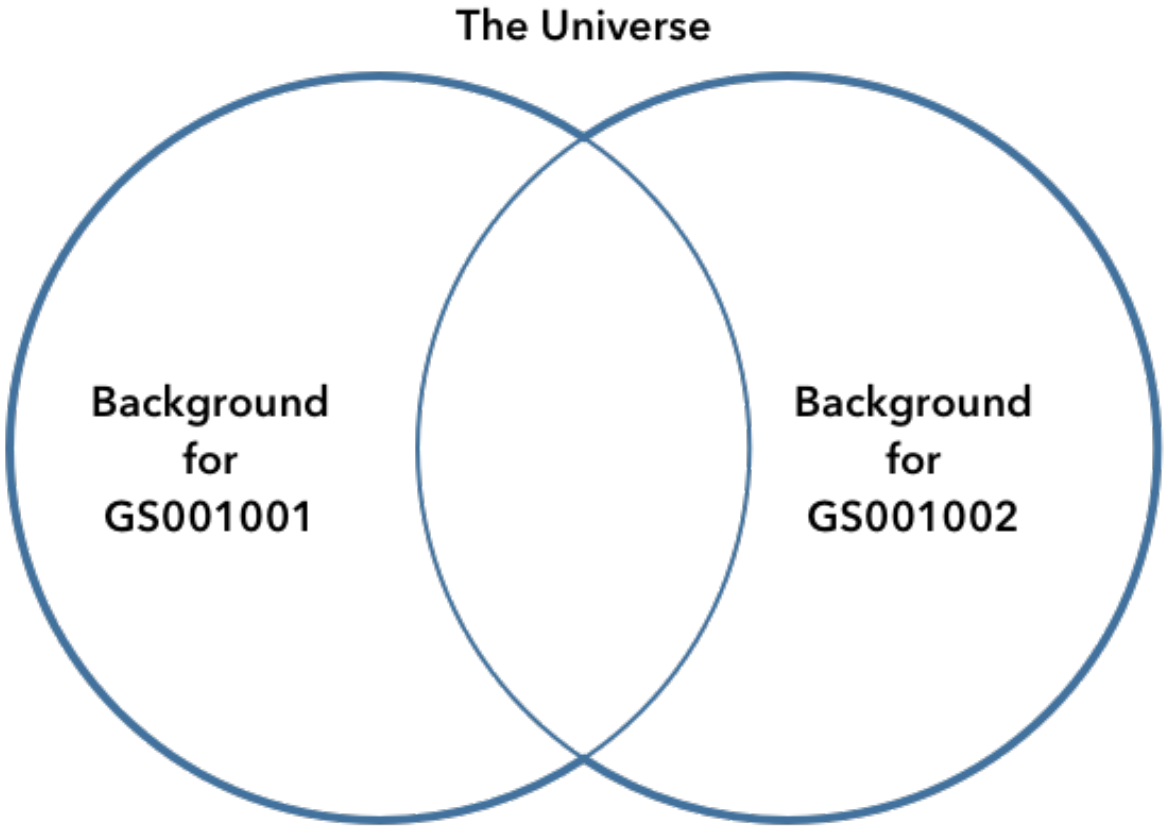
An Example

The example below illustrates MSET with four trials.

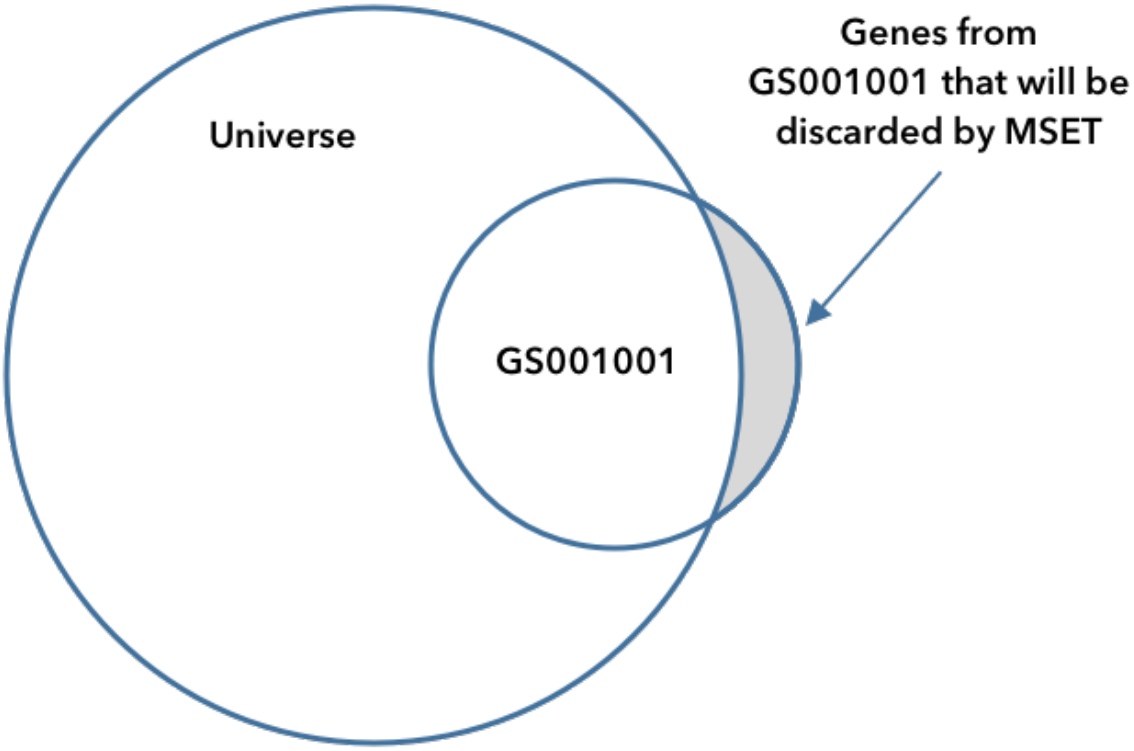
Given the following:

- Two gene sets, e.g. GS001001 and GS001002
- A background for both GS001001 and GS001002 (we can call them B001001 and B001002, respectively)
  - Geneweaver determines this automatically by inspecting the gene ID type and species of each gene set
- The number of trials MSET should perform (in this case, four)

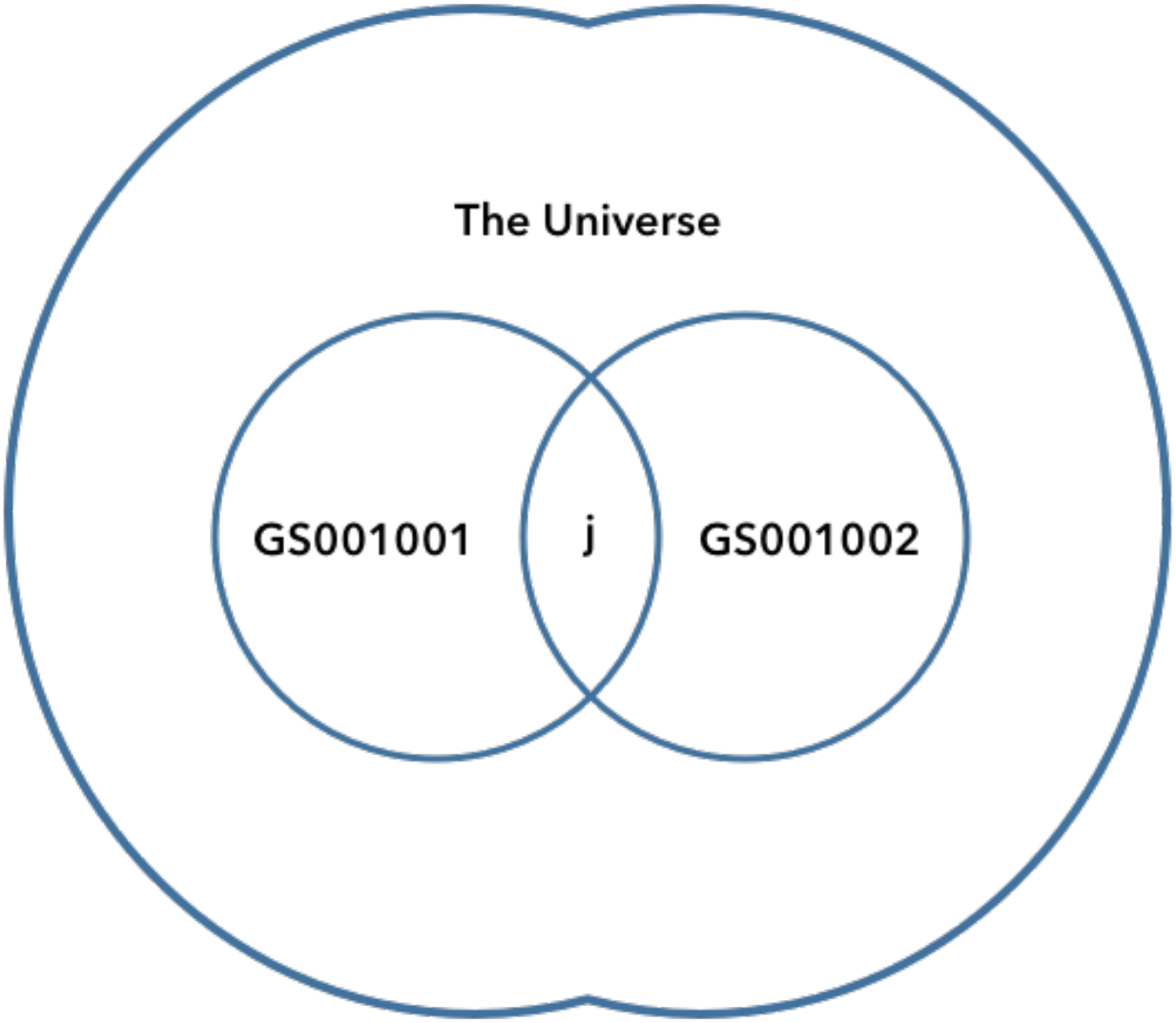
1. First, MSET defines *The Universe* as the intersection of B001001 and B001002.



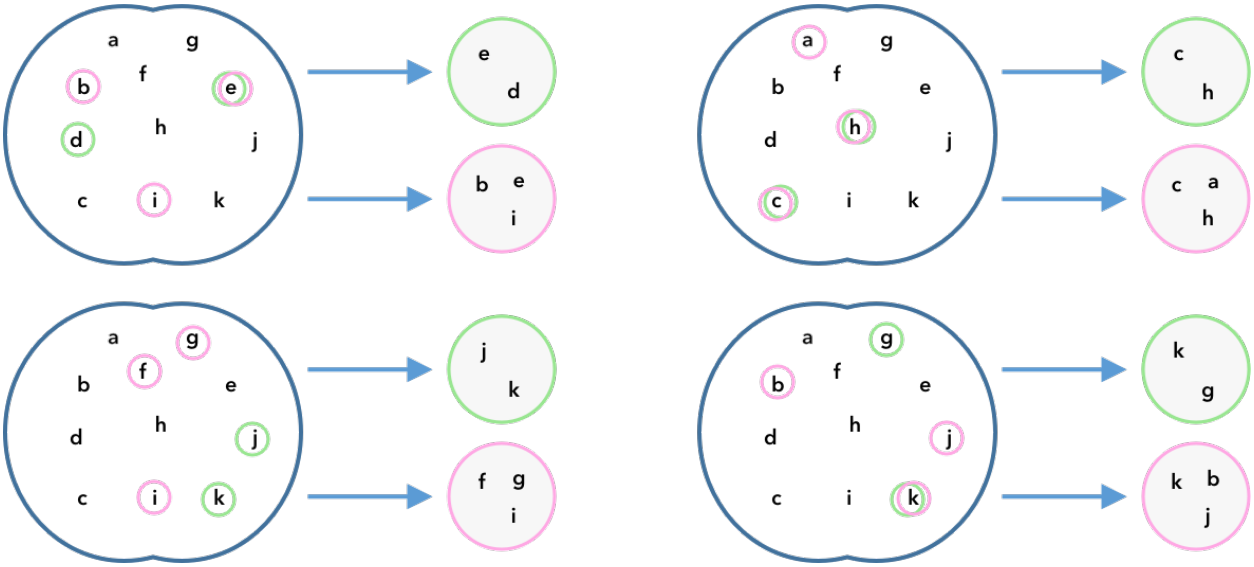
2. Any genes in GS001001 or GS001002 that aren't in *The Universe* are discarded from the analysis. GS001001 and GS001002 now only contain those genes that also exist in *The Universe*.



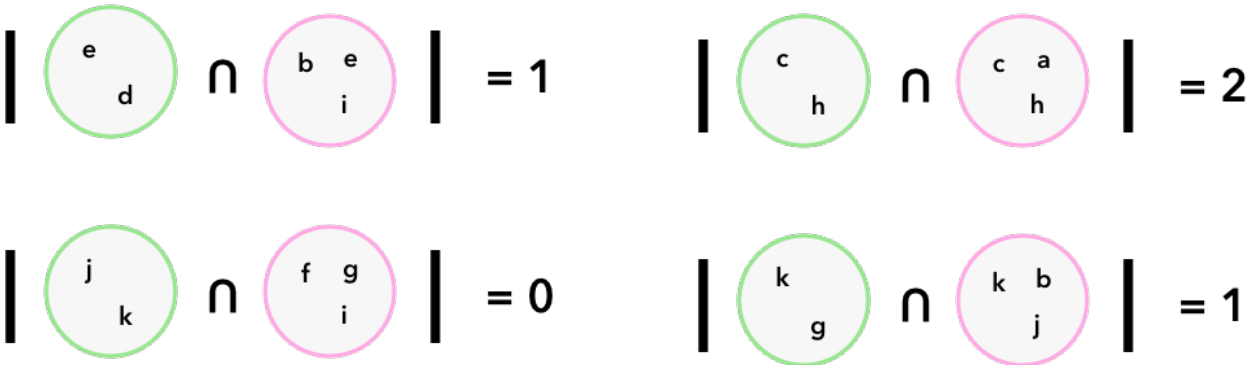
3. MSET then calculates the cardinality of the intersection of GS001001 and GS001002. Let's assume that GS001001 and GS001002 only share the gene j, then the intersect size is determined to be 1. Here we show simulated GS001001 set in green, and simulated GS001002 sets in pink. Genes which have been selected for either simulated set are circled in their set's color.



4. MSET then samples randomly without replacement to create four simulated sets each of GS001001 and GS001002. Here we assume that GS001001 has size 2, and GS001002 has size 3.



5. From the simulated gene sets above, MSET calculates the size of the intersect of each simulated set of GS001001 and GS001002.



6. MSET calculates a p-value using this formula:

$$\text{p-value} = \frac{\left(\begin{array}{c}\text{Number of Trials w/ Intersect of Simulations} \\ \text{Greater Than Intersect of Observed}\end{array}\right)}{\text{Total Number of Trials}}$$

We performed four trials, three of which had samples with an intersection at least as large as our observed gene sets. So MSET would return a p-value of 3/4, or 0.75.

Using MSET

Access the MSET Tool through the Analyze Genesets tab.

To analyze your genes, select two gene sets. You will often have organized these sets into a project relevant to your work. Projects may be created and updated by uploading GeneSets, searching the GeneWeaver database, or through the use of other tools in the GeneWeaver system. See the documentation for uploading

GeneSets, Search, or Manage GeneSets to learn more about these functions. MSET can only accept two gene sets as input, so you can only use the whole-project select box if your project only contains two sets.

Next, click on the MSET icon in the Analysis tools box to the left of the project list and specify how many trials you'd like MSET to perform. Once you're ready click the run button.

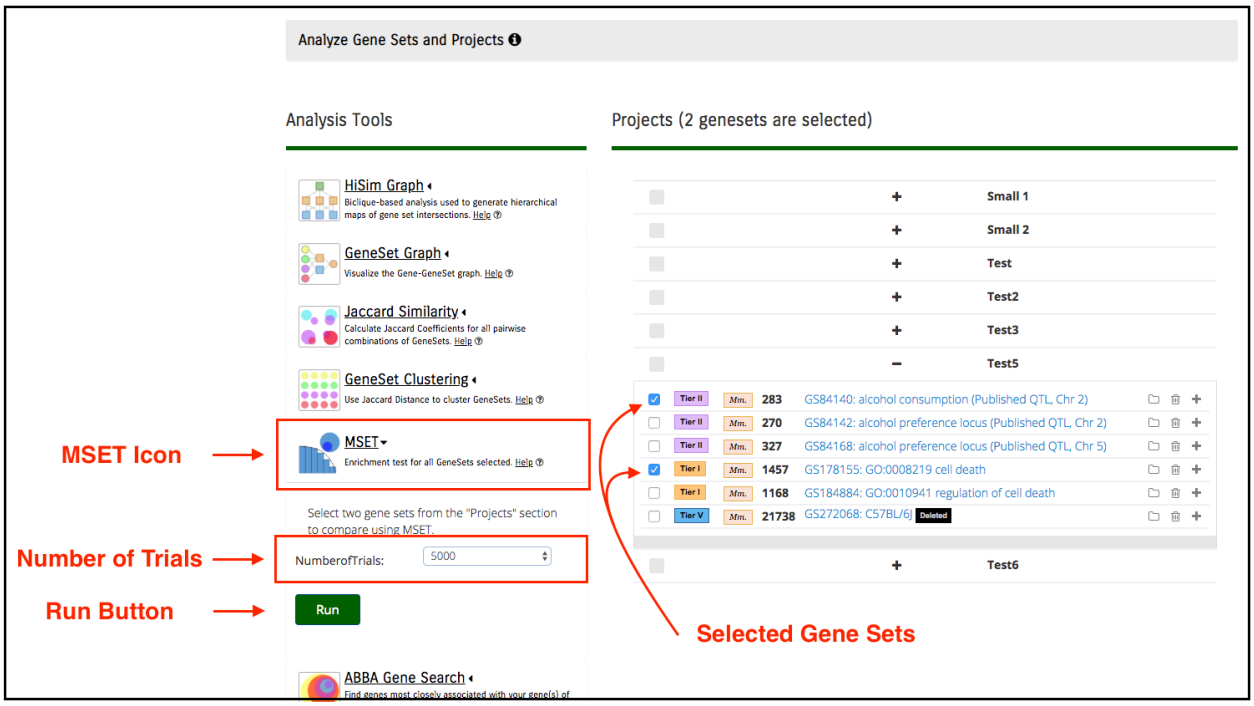


Figure 19:

Once the tool has completed the analysis you will be directed to the results page. There you can view the distribution graph of all simulated intersect sizes, an accurate size comparison graph of the selected sets and the background, and the genes shared by the two input sets. You can download both graphs for later use, and you can also create a new gene set from the genes shared by your two input sets.

Top ↑

ABBA

Given a set of interesting genes, do other genes have similar relationships to known sets of genes? For example, given a set of genes known to be related to drug abuse, what other genes share similar expression patterns in drug abuse gene sets? By answering this question, it becomes possible to elucidate under-studied or obfuscated genes that may play a role in complex phenotypes.

We have developed a new GeneWeaver tool to address this question, which we call **Anchored Biclique of Biomolecular Associations (ABBA)**. This tool takes advantage of the large number of collected data and cross-species integration to find new genes for investigation.

The search begins with a user-provided list of genes of interest, such as highly-studied genes with known pathways and relationships. The database then finds any gene sets that contain at least N of the genes in the provided list. From the resulting list of gene sets, ABBA then isolates any genes that occur in at least M GeneSets but not in the initial list. These resulting genes share similar gene set overlap with the original input set, but may not have been previously considered in relation to the gene set of interest.

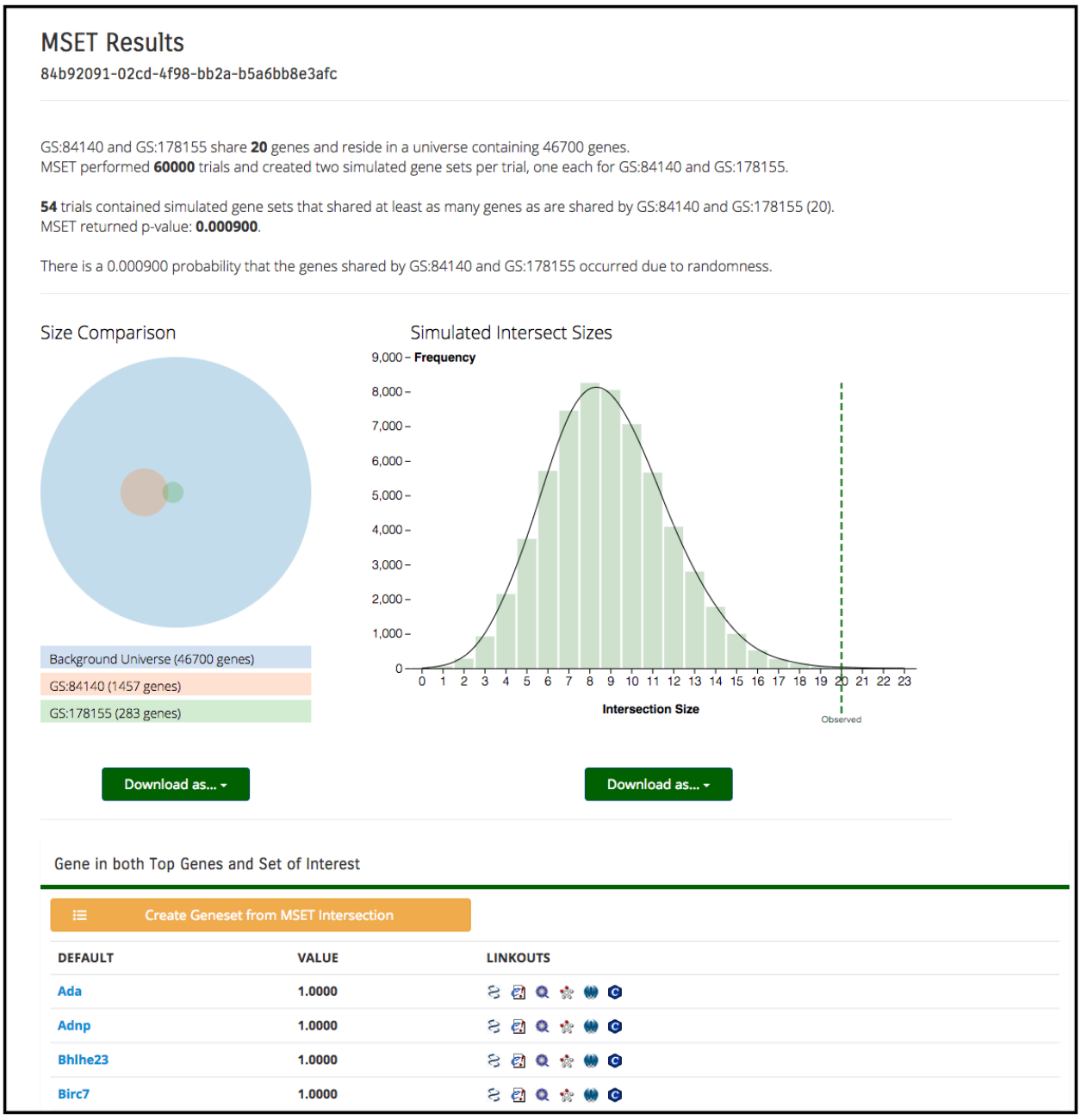

MSET Results
84b92091-02cd-4f98-bb2a-b5a6bb8e3afc
GS:84140 and GS:178155 share 20 genes and reside in a universe containing 46700 genes.
MSET performed 60000 trials and created two simulated gene sets per trial, one each for GS:84140 and GS:178155.
54 trials contained simulated gene sets that shared at least as many genes as are shared by GS:84140 and GS:178155 (20).
MSET returned p-value: 0.000900.
There is a 0.000900 probability that the genes shared by GS:84140 and GS:178155 occurred due to randomness.
Size Comparison
Background Universe (46700 genes)
GS:84140 (1457 genes)
GS:178155 (283 genes)
Simulated Intersect Sizes
Frequency
Intersection Size
Observed
Download as...
Download as...
Gene in both Top Genes and Set of Interest
Create Geneset from MSET Intersection
DEFAULT
VALUE
LINKOUTS
Ada 1.0000
Adnp 1.0000
Bhlhe23 1.0000
Birc7 1.0000

Figure 20:

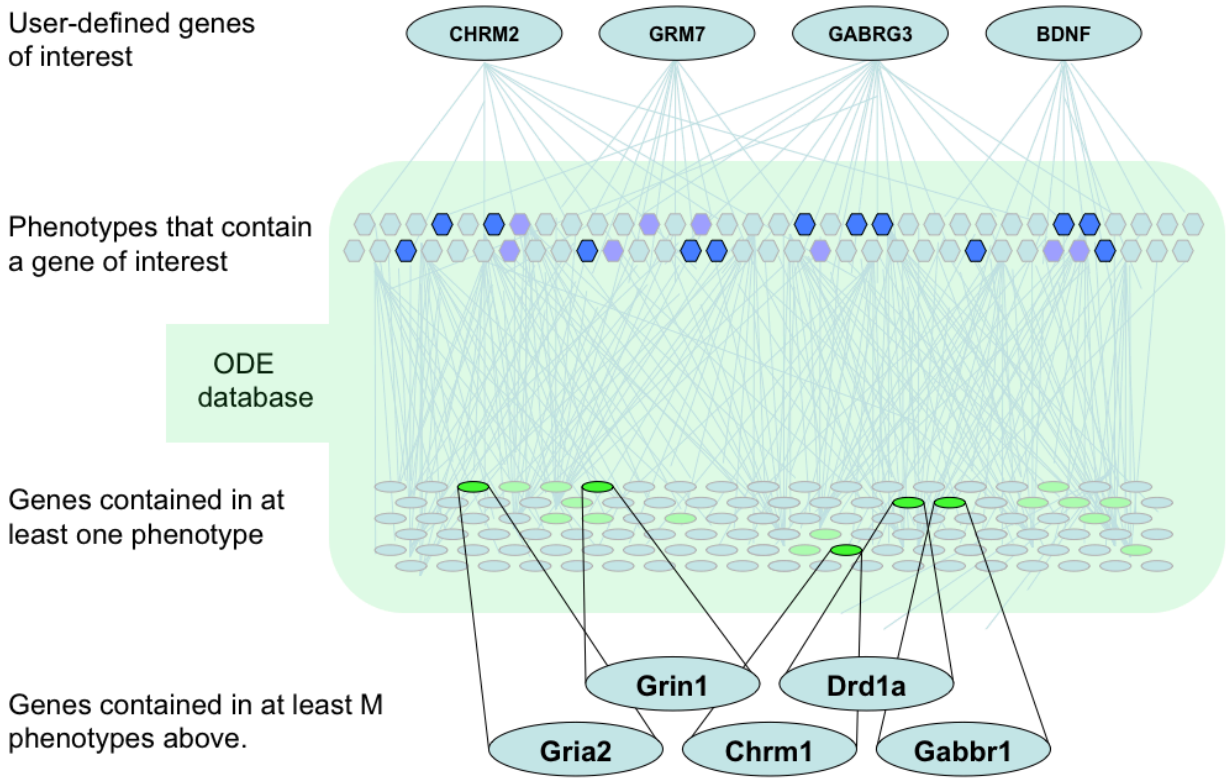


Figure 21: "ABBA applied to a set of 4 genes of interest"

In the above figure, the lighter nodes indicate less overlap. Using N=2 produces a collection of 37 GeneSets as of 7 July 2010. For brevity, only the top 5 results are shown above. With M=15, the following table lists genes in the result having similar relationships to the input set.

ABBA Result Genes (Number of Phenotypes)				
Grin1 (21)	Drd1a (20)	Gria2 (19)	Chrm1 (19)	Gabbr1 (18)
Adcyap1r1 (18)	Chrm4 (17)	Drd2 (17)	Grik3 (17)	Grm8 (17)
Gabrg2 (16)	Gabra1 (16)	Chrm3 (16)	Agtrl1 (16)	Grik1 (16)
Ntsr1 (16)	Grm2 (15)	Celsr3 (15)	F2r (15)	Chrna3 (15)
Hcrtr2 (!5)	Sstr2 (15)	Npy2r (15)	Edg2 (15)	Gabrb3 (15)
Gabrd (15)	Adrb2 (15)	Chrm5 (15)	Gria3 (15)	

Without reasonable thresholds, the results quickly become overwhelming. As of this writing, a simple set of 4 genes of interest results in 555 GeneSets and over 38,000 genes in the candidate list. Increasing the input set to 7 genes of interest results in 983 GeneSets and almost 40,000 genes. Simply requiring gene sets to contain at least 3 genes significantly reduces the search space to 11 and 37 GeneSets, respectively.

Genes of Interest	GeneSets to Search	Genes to Examine
CHRM2, GABRG3, BDNF, GRM7	w/ N=1, 555 GeneSets	38302 Genes
	w/ N=3, 11 GeneSets	8983 Genes
CHRM2, GABRG3, BDNF, GRM7, SOD2, MOBP, NPY	w/ N=1, 983 GeneSets	39313 Genes
	w/ N=3, 37 GeneSets	36076 Genes
	w/ N=5, 6 GeneSets	11630 Genes

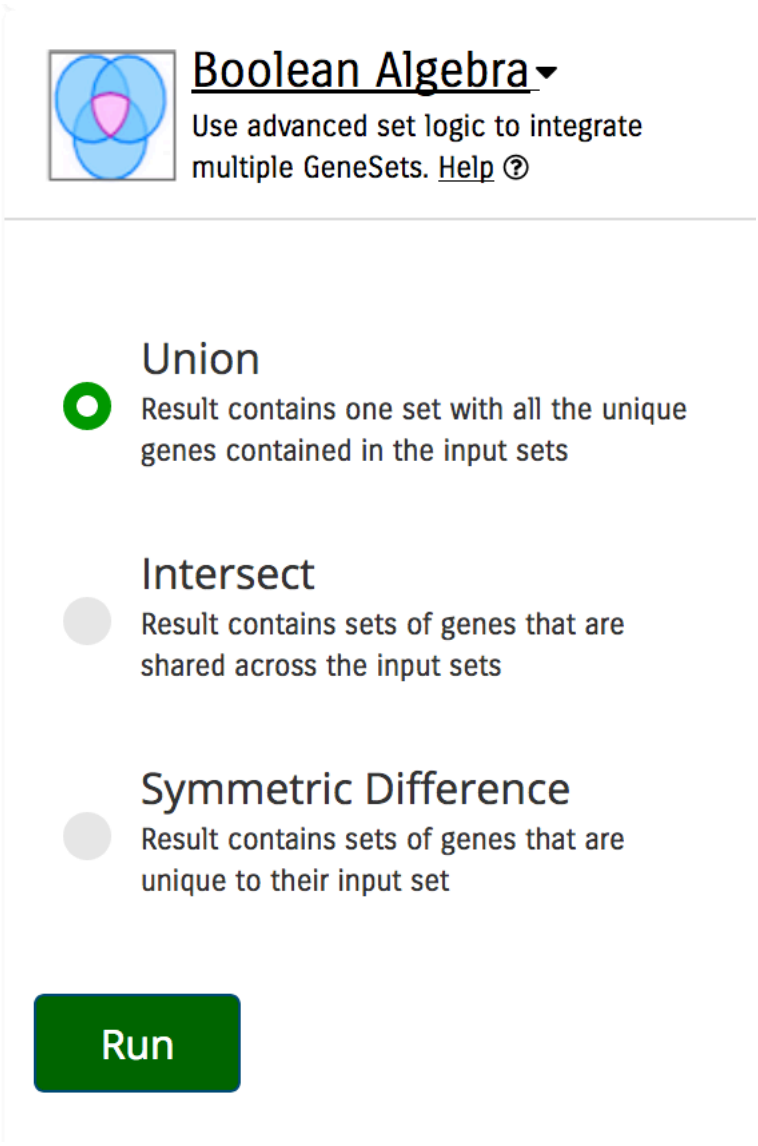
Top ↑

Boolean Algebra

The Boolean Algebra Tool performs basic set operations on at least two Gene Sets. Results are displayed as lists of genes beloging to one of the three different types of set operations: Union, Intersect, and Symmetric Difference. Furthermore, results allow users to quickly determine new relationships between Gene Sets and create a new Gene Set based on set-derived findings.

Using the Boolean Algebra Tool

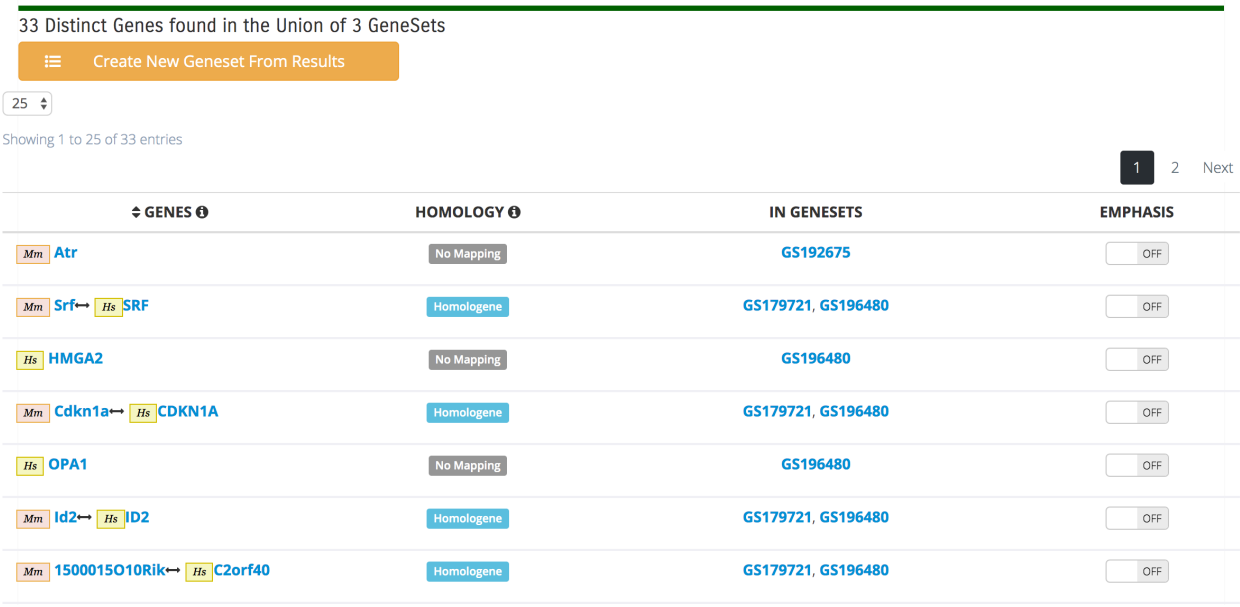
Access the Boolean Algebra Tool through the Analyze Genesets tab, located in the left-hand column and distinguished by the Venn diagram icon.



To generate Boolean Algebra results, select either a Project of two or more Gene Sets or at least two individual Gene Sets from a project. Next, select the appropriate Boolean Algebra function. These functions are based on basic *Set Algebra*: **Union**, **Intersection**, **Symmetric Difference**.

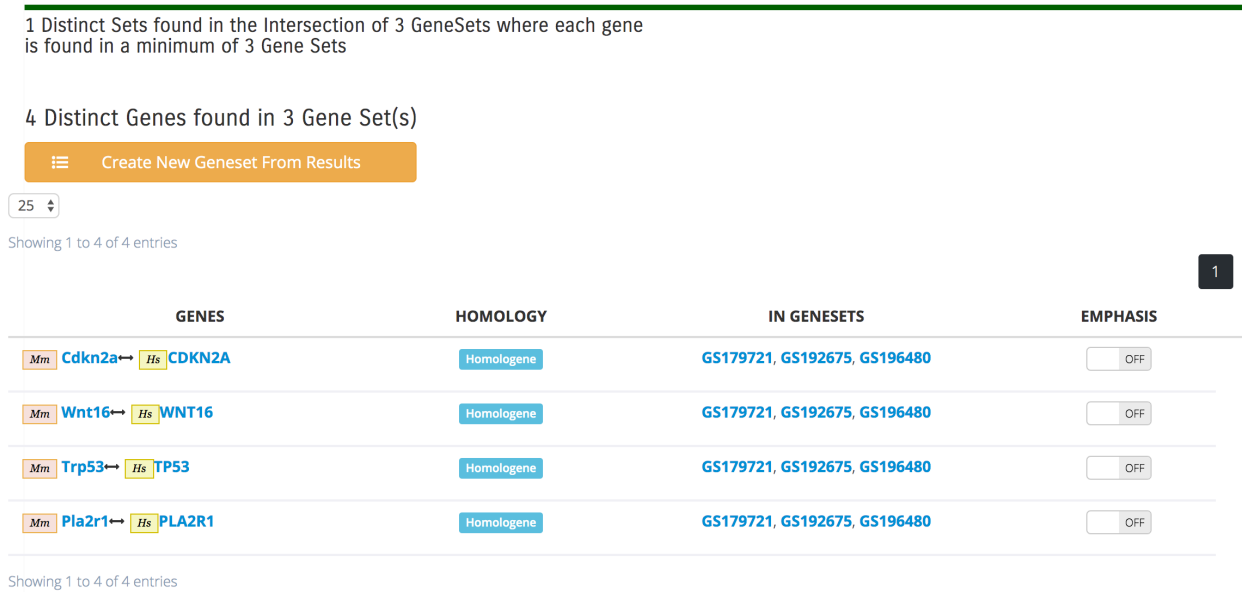
- **Union**: This tool generates a set of all genes located in all sets. It removes duplicates by default. The results will display what homology mapping was used to generate a gene entry.

This result shows the union of three Gene Sets, two mouse and one human.

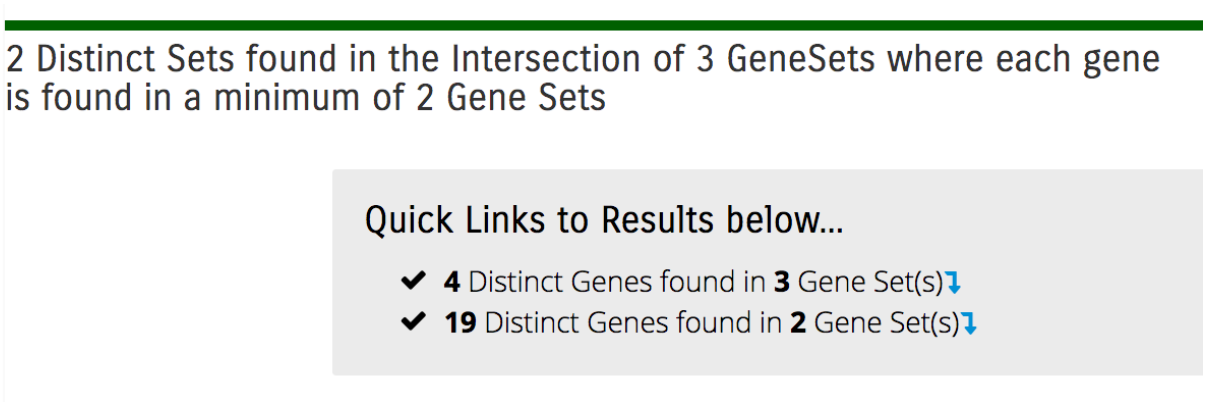


- **Intersection**: This option will cause the Boolean tool to return all genes in common with the selected Gene Set inputs. It has an additional option (*"Genes must intersect in at least X"*) that specifies the minimal amount of overlaps required to return a result. If a minimal overlap is set to *3*, for example, only Gene Sets that intersect with 3 or more genes will be evaluated, and only the intersecting genes will be returned. In addition, results are divided into separate groups based on the number of genes in their intersections.

These three Gene Sets have 4 genes in common. All of them are homologs between mouse and human.

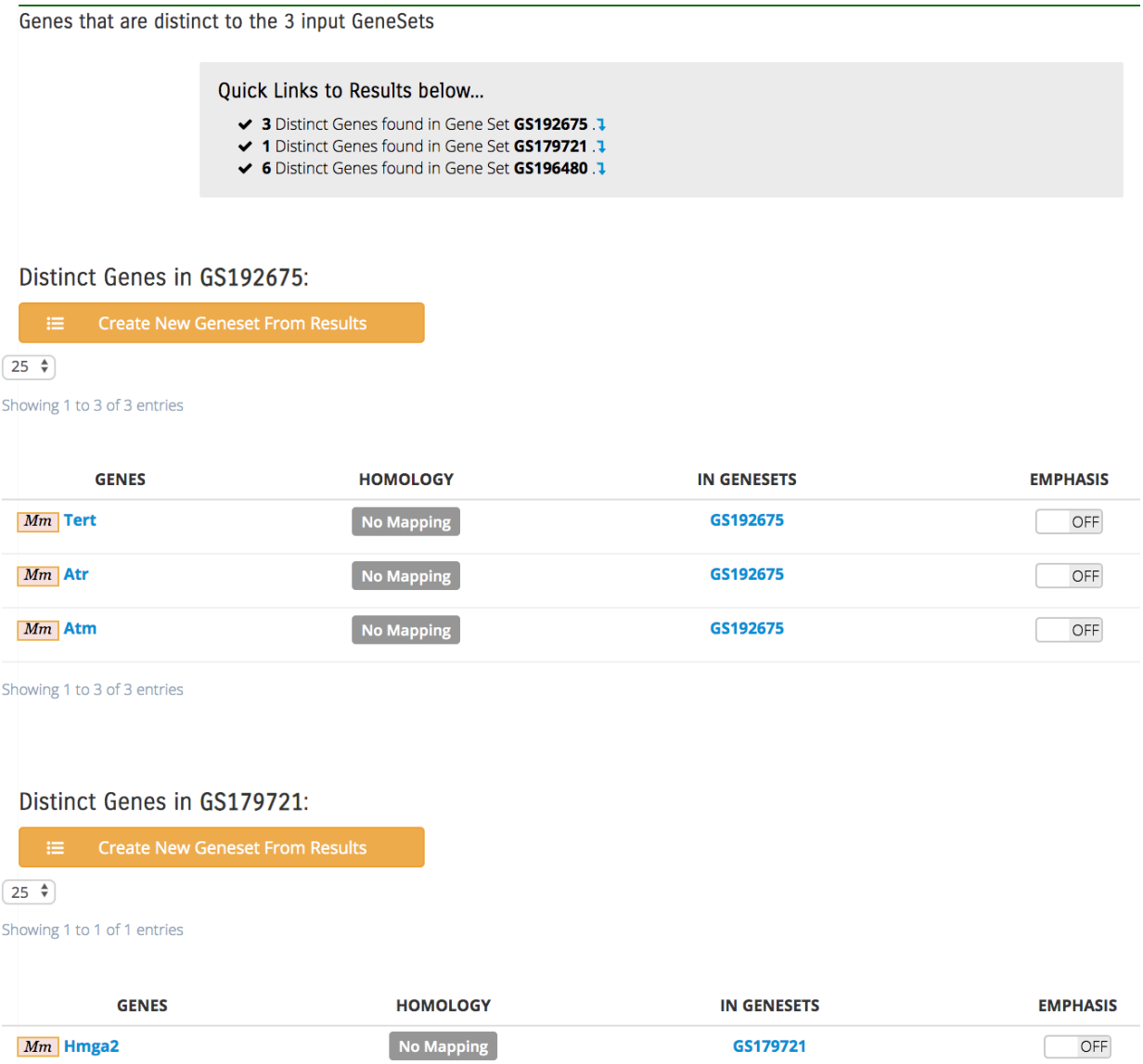


Changing the overlap to 2 created two sets of results, those in all 3 Gene Sets and those in only 2 of the Gene Sets.



- **Symmetric Difference**: This tool will create a set of genes that are unique to the Gene Sets selected as input. It effectively finds the Union of all Gene Sets minus the intersection of those Gene Sets.

In this example, there is a result set of unique genes for each input Gene Set.



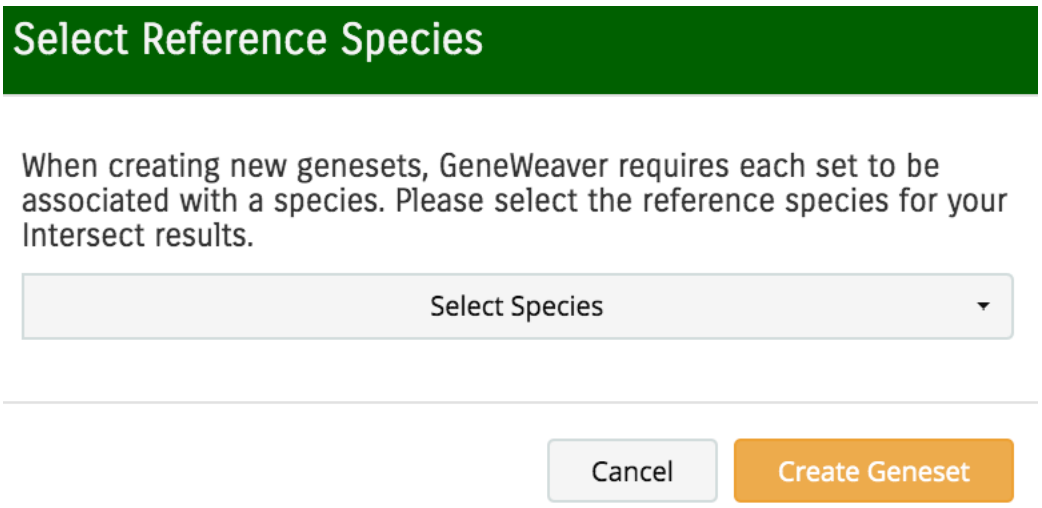
Managing Results

A table located just below the circle overlap diagram and above the results is intended to display a broad survey of genes included in the input Gene Sets, categorized by species. It lists: *Genes Specific to Species*, *Genes In Common with at Least One Other Species*, and *Total Number of Genes*. These values are based on the total number of genes in the input sets, and may not specifically represent results. The table is intended to help aid in the selection of which species to map the results in cases where new Gene Sets are created.

	MUS MUSCULUS	HOMO SAPIENS
Genes Specific to Species	4	6
Genes In Common with at Least One Other Species	27	23
Total Number of Genes	31	29

Genes returned by the Boolean Algebra tool can be added to new Gene Sets. To do this, click on the **Create New Gene Set From Results** button for the group you want.

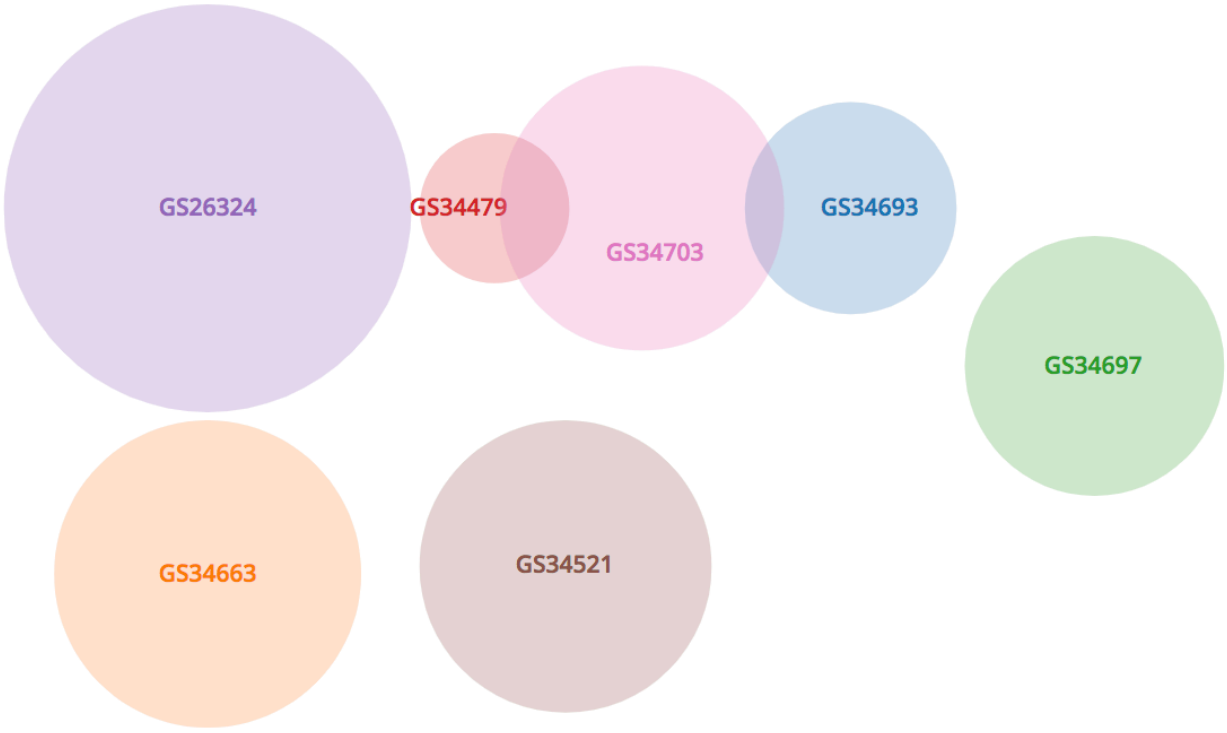
Since results can contain genes from a mixed set of species, a species must be selected for mapping the genes in the new Gene Set.



The standard Upload GeneSet page will open. The genes will be listed in the gene information section. If no species is selected, no genes will be listed. You can now edit any of the fields to change the Gene Set name, description, etc. Follow the Upload GeneSet procedure. It is also important to note that very large gene lists may take a few moments to load, during which time the user may experience a dimmed 'Loading' screen.

Circle Overlap Diagram

If the user selects 10 or fewer Gene Sets, a gene overlap diagram will appear near the top of the results page. The **Circle Overlap** representation is an approximation of Euler fractional overlaps. It represents how the input genesets relate to each other. It uses the same homology mapping as the Boolean Algebra tool to render the approximate fractional overlap of the genes shared between each set.



Top ↑

API

Geneweaver API Request Formats

The Geneweaver API can be accessed through the following address: https://geneweaver.org/api/

Definitions

Term	Definition
Output	The output of every call to the Geneweaver API will be in JSON format. An example of JSON can be viewed here: http://json.org/example.
API Key	The Geneweaver API makes use of api keys to identify users and determine permissions they have when executing api calls. For example, to determine if a user has permission to view a private gene set they must identify themselves via their unique api key. In place of an api key, the "guest" key may be used instead; however, this will limit the user to public data only. A user may request an API key by creating an account on Geneweaver and asking for an API key on the account management page.
apiKey	A unique identifier for a user (see API Key)
ReferenceID	A string representing the gene ID
GeneDatabase	A string with the Database Name corresponding to the gene ID
homology	Optional addition that will return homologous genes
<GeneSetID>	A positive integer value representing a gene set ID
<GeneID>	A positive integer value representing a gene ID

Term	Definition
<ProjectID>	A positive integer value representing a project ID
<PlatformID>	A positive integer value representing a platform ID
<PublicationID>	A positive integer value representing a publication ID
<SpeciesID>	A positive integer value representing a species ID
<DatabaseID>	A positive integer value representing a gene database ID
<Project_Name>	A string representing the name of an existing or new project
<TaskID>	A unique identifier for a task returned by a tool
<FileType>	The file type you wish to get (see specific tool for available file types)

Data Calls

This section outlines the individual calls that are available from the Geneweaver API.

Get Gene Sets by Gene Reference ID: This call returns all gene sets that contain the specified gene. The added homology parameter will return all gene sets that contain homologous genes as well.

`/api/get/geneset/bygeneid/<apiKey>/<ReferenceID>/<GeneDatabase>/homology`

Sample Call: https://geneweaver.org/api/get/geneset/bygeneid/Fw7J4GeAXE8CMVvLTKyrtBDk/RGD 2561/RGD/homology

Get Gene Set by Gene Set ID: This call returns all information about a specified gene set given that gene set ID.

`/api/get/geneset/byid/<GeneSetID>/`

Sample Call: https://geneweaver.org/api/get/geneset/byid/220592/

Get Gene Set by User: This call returns all gene sets owned by the specified user

`/api/get/geneset/byuser/<apikey>/`

Sample Call: https://geneweaver.org/api/get/geneset/byuser/Fw7J4GeAXE8CMVvLTKyrtBDk/

Get Genes by Gene Set ID: This call returns all genes belonging to a given gene set.

`/api/get/genes/bygenesetid/<GeneSetID>/`

Sample Call: https://geneweaver.org/api/get/genes/bygenesetid/220592/

Get Gene by Gene ID: This call returns all information about a specified gene given a ODE gene ID.

`/api/get/gene/bygeneid/<GeneID>/`

Sample Call: https://geneweaver.org/api/get/gene/bygeneid/8/

Get Geneset by Project ID: This call returns all genesets associated with a project given a project ID.

`/api/get/geneset/byprojectid/<apikey>/<ProjectID>/`

Sample Call: https://geneweaver.org/api/get/geneset/byprojectid/Fw7J4GeAXE8CMVvLTKyrtBDk/2404 /

Get Geneset by Geneset ID: This call returns all the information about a given geneset given its geneset ID

`/api/get/geneset/bygenesetid/<GeneSetID>/`

Sample Call: https://geneweaver.org/api/get/geneset/bygenesetid/8/

Get Projects by User: Returns all the projects that are owned by a given user.

`/api/get/project/byuser/<apikey>/`

Sample Call: https://geneweaver.org/api/get/project/byuser/Fw7J4GeAXE8CMVvLTKyrtBDk/

Get Ontologies by Geneset ID: Returns all the Ontology annotations associated with a geneset.

`/api/get/ontologies/bygeneset/<apikey>/<GeneSetID>/`

Sample Call: https://geneweaver.org/api/get/ontologies/bygeneset/Fw7J4GeAXE8CMVvLTKyrtBDk/8/

Get Probes by Gene ID: Returns all the probes associated with a gene.

`/api/get/probes/bygeneid/<apikey>/<ReferenceID>/`

Sample Call: https://geneweaver.org/api/get/probes/bygeneid/Fw7J4GeAXE8CMVvLTKyrtBDk/RGD2561/

Get Platform by Platform ID: Returns the platform associated with a platform ID.

`/api/get/platform/byid/<apikey>/<PlatformID>/`

Sample Call: https://geneweaver.org/api/get/platform/byid/Fw7J4GeAXE8CMVvLTKyrtBDk/3/

Get SNP by Gene ID: Returns all the SNPs associated with a gene (provided SNPs are loaded in the GW DB).

`/api/get/snp/bygeneid/<apikey>/<ReferenceID>/`

Sample Call: https://geneweaver.org/api/get/snp/bygeneid/Fw7J4GeAXE8CMVvLTKyrtBDk/RGD2561/

Get Publication by Publication ID: Returns all the publication data for given publication ID.

`/api/get/publication/byid/<apikey>/<PublicationID>/`

Sample Call: https://geneweaver.org/api/get/publication/byid/Fw7J4GeAXE8CMVvLTKyrtBDk/26/

Get Species by Species ID: Returns all the species information given a species ID.

`/api/get/species/byid/<apikey>/<SpeciesID>/`

Sample Call: https://geneweaver.org/api/get/species/byid/Fw7J4GeAXE8CMVvLTKyrtBDk/4/

Get Gene Database by Database ID: Returns information on a gene database given a database ID.

`/api/get/genedatabase/byid/<apikey>/<DatabaseID>/`

Sample Call: https://geneweaver.org/api/get/genedatabase/byid/Fw7J4GeAXE8CMVvLTKyrtBDk/7/

Create Project: Creates a project for the user and returns the project id that was just created.

`/api/add/project/byuser/<apikey>/<Project_Name>/`

Sample Call: https://geneweaver.org/api/add/project/byuser/Fw7J4GeAXE8CMVvLTKyrtBDk/myNewProject/

Add GeneSet To Project: Adds an existing gene set to a project you own

`/api/add/geneset/toproject/<apikey>/<ProjectID>/<GeneSetID>/`

Sample Call: https://geneweaver.org/api/add/geneset/toproject/Fw7J4GeAXE8CMVvLTKyrtBDk/3323/86676/

Remove GeneSet From Project: Removes a gene set from a project you own.

`/api/Delete/geneset/fromproject/<apikey>/<ProjectID>/<GeneSetID>/`

Sample Call: https://geneweaver.org/api/delete/geneset/fromproject/Fw7J4GeAXE8CMVvLTKyrtBDk/3323/86676/

Tool Output Calls

This section is dedicated to calling the GeneWeaver tools via the api. Tools are called by their separate api URLs. This will initiate the tool to run. The tools will return a task ID. Then the getStatus api call may be made to determine if the tool has finished processing your request given a task id. Once complete the finished data may be retrieved via the getFile api call using a task id.

For **ALL** tools, any of the parameters may be substituted with `Default` to use the default values.

Get Status of Tool Job: This api call will return the status of a job given its unique task ID.

`/api/tool/get/status/<TaskID>/`

This will return one of the following:

- **PENDING:** The task is in the queue to be run
- **STATUS INFO:** The task is currently being run
- **FAILURE:** The task has failed, contact an administrator
- **SUCCESS:** The task has completed successfully

Sample Call: https://geneweaver.org/api/tool/get/status/c0bdc0e4-3e23-4273-aeeb-21539e60c53d/

Get Results Link: This api call will return a url that can be called to access a file requested by the user if the user has permission to access that file. This is useful if you wish to store a quicker method of repeat access to a file.

`/api/tool/get/link/<apikey>/<TaskID>/<FileType>/`

Sample Call: https://geneweaver.org/api/tool/get/link/Fw7J4GeAXE8CMVvLTKyrtBDk/c0bdc0e4-3e23-4273-aeeb-21539e60c53d/pdf/

Get Results File: This api call will return the file requested by the user if the user has permission to access that file.

`/api/tool/get/file/<apikey>/<TaskID>/<FileType>/`

Sample Call: https://geneweaver.org/api/tool/get/file/Fw7J4GeAXE8CMVvLTKyrtBDk/c0bdc0e4-3e23-4273-aeeb-21539e60c53d/pdf/

Get Results by User: Returns all the tasks ids run by the user

`/api/get/results/byuser/<apikey>/`

Sample Call: https://geneweaver.org/api/get/results/byuser/Fw7J4GeAXE8CMVvLTKyrtBDk/

Get Results by Task ID: Returns all the information about a given tool run given a task ID

`/api/get/result/bytaskid/<apikey>/<TaskID>/`

Sample Call: https://geneweaver.org/api/get/result/bytaskid/Fw7J4GeAXE8CMVvLTKyrtBDk/c0bdc0e4-3e23-4273-aeeb-21539e60c53d/

Run Tool Calls

GeneSet Viewer: This tool visualizes the gene-geneset graph. This tool requires at least 2 genesets.

`/api/tool/genesetviewer/<apikey>/<homology>/<supressDisconnected>/<minDegree>/<genesets>/`

`/api/tool/genesetviewer/byprojects/<apikey>/<homology>/<supressDisconnected>/<minDegree>/<projects>/`

Variables:

- homology: [“Included”,”Excluded”] default “Included”
- supressDisconnected: [“On”,”Off”] default “On”
- minDegree: [“Auto”, “1”,”2”,”3”,”4”,”5”,”10”,”20”] default “Auto”
- genesets: A list of GeneSet IDs separated by colons. [“:”]
- projects: A list of Project IDs separated by colons. [“:”]

Expected Returns: [“pdf”, ”dot”, ”svg”]

Sample Call: https://geneweaver.org/api/tool/genesetviewer/Fw7J4GeAXE8CMVvLTKyrtBDk/Included/On/Auto/391:394:395/

https://geneweaver.org/api/tool/genesetviewer/byprojects/Fw7J4GeAXE8CMVvLTKyrtBDk/Included/On/Auto/3323:2404/

Jaccard Clustering: This tool displays the Jaccard Distance (a measure of dissimilarity) and is used to cluster genesets. This tool requires at least 3 genesets.

`/api/tool/jaccardclustering/<apikey>/<homology>/<method>/<genesets>/`

`/api/tool/jaccardclustering/byprojects/<apikey>/<homology>/<method>/<projects>/`

Variables:

- homology: [“Included”, ”Excluded”] default “Included”
- methods: [“Ward”, ”Single”, ”Centroid”, ”McQuitty”, ”Average”, ”Complete”, ”Median”]
- genesets: A list of GeneSet IDs separated by colons. [“:”]
- projects: A list of Project IDs separated by colons. [“:”]

Expected Returns: [“pdf”, ”png”, ”jac”]

Sample Call: https://geneweaver.org/api/tool/jaccardclustering/Fw7J4GeAXE8CMVvLTKyrtBDk/Included/Ward/391:394:395/

https://geneweaver.org/api/tool/jaccardclustering/byprojects/Fw7J4GeAXE8CMVvLTKyrtBDk/Included/Ward/3323:2404/

Jaccard Similarity: This tool computes the Jaccard coefficient, a measure of similarity, for multiple genesets. This tool requires at least 2 genesets.

`/api/tool/jaccardsimilarity/<apikey>/<homology>/<pairwiseDeletion>/<genesets>/`

`/api/tool/jaccardsimilarity/byprojects/<apikey>/<homology>/<pairwiseDeletion>/<projects>/`

Variables:

- homology: [“Included”,”Excluded”] default “Included”
- pairwiseDeletion: [“Enabled”,”Disabled”] default “Disabled”
- genesets: A list of GeneSet IDs separated by colons. [“:”]
- projects: A list of Project IDs separated by colons. [“:”]

Expected Returns: [“svg”, ”png”, “txt”*]

> *the txt follows this format. Rows are separated by newlines, columns by tabs. The first row character is a 0, then tab separated geneset names on the first row. Every following row begins with a geneset name to create the matrix. The values in the corresponding areas are the “jaccardValue:pValue” of those two genes.

Sample Call: http://geneweaver.org/api/tool/jaccardsimilarity/Fw7J4GeAXE8CMVvLTKyrtBDk/Included/Disabled/391:394:395//

http://geneweaver.org/api/tool/jaccardsimilarity/byprojects/Fw7J4GeAXE8CMVvLTKyrtBDk/Included/Disabled/3323:2404/

Combine: This tool creates a geneset-gene matrix of the combined genesets. This tool requires at least 2 genesets.

`/api/tool/combine/<apikey>/<homology>/<genesets>/`

`/api/tool/combine/byprojects/<apikey>/<homology>/<projects>/`

Variables:

- homology: [“Included”,”Excluded”] default “Included”
- genesets: A list of GeneSet IDs separated by colons. [“:”]
- projects: A list of Project IDs separated by colons. [“:”]

Expected Returns: [“odemat”]

Sample Call: https://geneweaver.org/api/tool/combine/Fw7J4GeAXE8CMVvLTKyrtBDk/Included/391:394:395/

https://geneweaver.org/api/tool/combine/byprojects/Fw7J4GeAXE8CMVvLTKyrtBDk/Included/3323:2404/

Phenome Map: This tool uses biclique-based analysis to generate hierarchical maps of gene set interactions.

`/api/tool/phenomemap/<apikey>/<homology>/<minGenes>/<permutationTimeLimit>/<maxInNode>/<permutations>/<disableBootstrap>/<minOverlap>/<nodeCutoff>/<geneIsNode>/<useFDR>/<hideUnEmphasized>/<p\_Value>/<maxLevel>/<genesets>/`

`/api/tool/phenomemap/byprojects/<apikey>/<homology>/<minGenes>/<permutationTimeLimit>/<maxInNode>/<permutations>/<disableBootstrap>/<minOverlap>/<nodeCutoff>/<geneIsNode>/<useFDR>/<hideUnEmphasized>/<p\_Value>/<maxLevel>/<projects>/`

Variables:

- homology: [“Included”,”Excluded”] default “Included”
- minGenes: [“1”,”2”,”3”,”4”,”5”,”6”,”7”,”8”,”9”,”10”,”15”,”20”,”25”] default “1”
- permutationTimeLimit: [“5”,”10”,”15”,”20”] default “5”
- maxInNode: [“4”,”8”,”12”,”16”,”20”,”24”,”28”,”32”] default “4”
- permutations: [“100000”,”50000”,”25000”,”5000”,”1000”,”500”,”100”,”0”] default “100000”
- disableBootstrap: [“False”,”True”] default “False”
- minOverlap: [“0%”,”5%”,”10%”,”15%”,”20%”,”25%”,”50%”,”75%”] default “0%”
- nodeCutoff: [“Auto”,”1.0”,”0.1”,”0.01”,”0.001”,”0.0001”,”0.00001”] default “Auto”
- geneIsNode: [“All”, “Exclusive”] default “All”
- useFDR: [“False”,”True”] default “False”
- hideUnEmphasized: [“False”,”True”] default “False”
- p_Value: [“1.0”,”0.5”,”0.10”,”0.05”,”0.01”] default “1.0”
- maxLevel: [“0”,”10”,”20”,”40”,”60”,”80”,”100”] default “0”

- genesets: A list geneset IDs separated by colons. [":"]
- projects: A list of project IDs separated by colons. [":"]

Expected Returns: ["dot", "el.profile", "el", "graphml", "odemat", "svg"]

Sample Call: https://geneweaver.org/api/tool/phenomemap/Fw7J4GeAXE8CMVvLTKyrtBDk/Included/1/5/4/100000/False/0%/Auto/All/False/False/1.0/0/391:394:395/

https://geneweaver.org/api/tool/phenomemap/byprojects/Fw7J4GeAXE8CMVvLTKyrtBDk/Included/1/5/4/100000/False/0%/Auto/All/False/False/1.0/0/3323:2404/

Boolean Algebra: This tool searches for genes across genesets.

`/api/tool/booleanalgebra/<apikey>/<relation>/<genesets>/`

`/api/tool/booleanalgebra/byprojects/<apikey>/<relation>/<projects>/`

Variables:

- homology: ["Included","Excluded"] default "Included"
- relation: ["Intersect","Union", "Except" ,"Intersect :#"] where # is an number, default option "Union", default value for # is 2
- genesets: A list of GeneSet IDs separated by colons. [":"]
- projects: A list of Project IDs separated by colons. [":"]

Expected Returns: ["txt"]

> This file has four sections of raw data separated by newlines. The first section has the method used (Union or Intersect At Least 2). The second section has the resulting genes' names. The third has the result genes' ids. The fourth is a 2d array print out of the genesets used to run the tool with all their genes' ids.

Sample Call: https://geneweaver.org/api/tool/combine/Fw7J4GeAXE8CMVvLTKyrtBDk/Included/391:394:395/

https://geneweaver.org/api/tool/combine/byprojects/Fw7J4GeAXE8CMVvLTKyrtBDk/Included/3323:2404/

Wrapping RestFUl API Code

There are numerous ways to wrap function URLs to ensure that return values are processed. Below is an example of a python method, adapted from http://stackoverflow.com/questions/17301938/making-a-request-to-a-restful-api-using-python

```
#Python 2.7.6
#RestfulClient.py

import requests
import json

# Replace with the correct URL
url = "http://api_url"

# retrieve API URL
myResponse = requests.get(url)
#print (myResponse.status_code)
```

```
# For successful API call, response code will be 200 (OK)
if(myResponse.ok):

    # Loading the response data into a dict variable
    # json.loads takes in only binary or string variables so using
    # content to fetch binary content
    # Loads (Load String) takes a Json file and converts into python data
    # structure (dict or list, depending on JSON)
    jData = json.loads(myResponse.content)

    print("The response contains {0} properties".format(len(jData)))
    print("\n")
    for key in jData:
        print key + " : " + jData[key]
else:
  # If response code is not ok (200), print the resulting http error code with description
    myResponse.raise_for_status()
```

Example Script

Below is an example of a Python script that makes various Geneweaver API calls. The script will:

1. Print the information about an example gene set
2. Print the information about the genes in the example gene set
3. Create a new project called "Nicotine Studies"
4. Add the example gene set to the new project
    - Afterwards, 9 other gene sets will also be added to the new project
5. Print the information about all the gene sets owned by the user
6. Print the information about all the projects owned by the user
7. Run the GeneSet Viewer tool on the 10 example gene sets
    - The script will wait 10 seconds for the tool job to finish
8. Print the status of the tool job
9. Print a link to the result of the tool job
10. Run the Jaccard Clustering tool on the 10 example gene sets
    - The script will wait 10 seconds for the tool job to finish
    - Afterwards, the script will also print the status of the tool job and then print a link to the result of the tool job
11. Run the Combine tool on two example projects
    - The script will wait 10 seconds for the tool job to finish
    - Afterwards, the script will also print the status of the tool job and then print a link to the result of the tool job
12. Print all of the tasks that the user ran.

```
# Python 2.7.13
# tutorial-api.py

import httplib
import json
import urllib
import time

# Replace with the correct API key
apikey = "Fw7J4GeAXE8CMVvLTKyrtBDk"
```

```
# Prepare the connection to Geneweaver
host = "geneweaver.org"
method = "GET"
connection = httplib.HTTPConnection(host)

# This function takes a GeneWeaver API URL and loads the result
# in a Python object.
def retrieveApiUrl(url):
    url = urllib.quote(url)
    connection.request(method, url)
    response = connection.getresponse()
    is_successful = response.status == 200 and response.reason == "OK"
    data = response.read() if is_successful else None
    jData = json.loads(data) if is_successful else None
    return jData

# This function waits 10 seconds so that a tool has enough time to run.
def waitForToolToFinish():
    print("Waiting 10 seconds for the task to complete...")
    time.sleep(10)
    print("10 seconds has elapsed, resuming...")
    print("")

"""
Get Gene Set by Gene Set ID:
"""

# Replace with the desired parameters.
GeneSetID = "14888"

# Call the API
url = "/api/get/geneset/byid/{}/".format(GeneSetID)
jData = retrieveApiUrl(url)

# Print the results if successful
if jData is not None:
    for key in jData[0][0]:
        print(key + " : " + str(jData[0][0][key]))
    print("")

"""
Get Genes by Gene Set ID:
"""

# Replace with the desired parameters.
GeneSetID = "14888"

# Call the API
url = "/api/get/genes/bygenesetid/{}/".format(GeneSetID)
jData = retrieveApiUrl(url)

# Print the results if successful
if jData is not None:
    for gene in jData:
```

```
        for key in gene[0]:
            print(key + " : " + str(gene[0][key]))
        print("")

"""
Create Project:
"""

# Replace with the desired parameters.
Project_Name = "Nicotine Studies"

# Call the API
url = "/api/add/project/byuser/{}/{}/"
    .format(apikey, Project_Name)
jData = retrieveApiUrl(url)

# Print the results if successful
if jData is not None:
    print("ProjectID = " + str(jData[0]))
    print("")

    # Save the ProjectID for later
    ProjectID = jData[0]

"""
Add GeneSet To Project:
"""

# Replace with the desired parameters.
GeneSetID = "14888"

# Call the API
url = "/api/add/geneset/toproject/{}/{}/{}/"
.format(apikey, ProjectID, GeneSetID)
jData = retrieveApiUrl(url)

# Print the results if successful
if jData is not None:
    print("ProjectID = " + str(jData[0][0]))
    print("GeneSetID = " + str(jData[0][1]))
    print("")

# Add 9 more gene sets
for GeneSetID in ["14889", "14890", "14891", "14892", "14887", "14893",
"14885", "86761", "86791"]:
    # Call the API
    url = "/api/add/geneset/toproject/{}/{}/{}/".format(apikey, ProjectID,
    GeneSetID)
    jData = retrieveApiUrl(url)

    # Print the results if successful
    if jData is not None:
        print("ProjectID = " + str(jData[0][0]))
        print("GeneSetID = " + str(jData[0][1]))
```

```
        print("")

"""
Get Gene Set by User:
"""

# Call the API
url = "/api/get/geneset/byuser/{}/".format(apikey)
jData = retrieveApiUrl(url)

# Print the results if successful
if jData is not None:
    for gene_set in jData:
        for key in gene_set[0]:
            print(key + " : " + str(gene_set[0][key]))
        print("")

"""
Get Projects by User:
"""

# Call the API
url = "/api/get/project/byuser/{}/".format(apikey)
jData = retrieveApiUrl(url)

# Print the results if successful
if jData is not None:
    for gene_set in jData:
        for key in gene_set[0]:
            print(key + " : " + str(gene_set[0][key]))
        print("")

"""
GeneSet Viewer:
"""

# Replace with the desired parameters.
homology = "Included" # ["Default", "Included","Excluded"]
supressDisconnected = "On" # ["Default", "On","Off"]
minDegree = "Auto" # ["Default", "Auto", "1","2","3","4","5","10","20"]
genesets = "14888:14889:14890:14891:14892:14887:14893:14885:86761:86791"
FileType = "pdf" # GeneSet Viewer can get ["pdf", "dot", "svg"]

# Call the API
url = "/api/tool/genesetviewer/{}/{}/{}/{}/{}/"
.format(apikey, homology, supressDisconnected, minDegree, genesets)
jData = retrieveApiUrl(url)

# Print the results if successful
if jData is not None:
    print("TaskID = " + jData)
    print("")

    # Save the TaskID for later
```

```
    TaskID = jData

# Wait for the task to complete.
waitForToolToFinish()

"""
Get Status of Tool Job:
"""

# Call the API
url = "/api/tool/get/status/{}/".format(TaskID)
jData = retrieveApiUrl(url)

# Print the results if successful
if jData is not None:
    print("Status = " + jData)
    print("")

"""
Get Results Link:
"""

# Replace with the desired parameters.
FileType = "pdf" # See the specific API to check
# which FileTypes are available.

# Call the API
url = "/api/tool/get/link/{}/{}/{}/".format(apikey, TaskID, FileType)
jData = retrieveApiUrl(url)

# Print the results if successful
if jData is not None:
    result_link = "http://{}{}".format(host, jData)

    # Follow this link to get the result file.
    print("File is located at: " + result_link)
    print("")

"""
Jaccard Clustering:
Note: This section combines creating the task and
getting the link to the result
"""

# Replace with the desired parameters.
homology = "Included" # ["Default", "Included","Excluded"]
jc_method = "Ward" # ["Default", "Ward", "Single", "Centroid", "McQuitty",
    # "Average", "Complete", "Median"]
genesets = "14888:14889:14890:14891:14892:14887:14893:14885:86761:86791"
FileType = "jac" # Jaccard Clustering can get ["pdf", "png", "jac"]

# Call the API
url = "/api/tool/jaccardclustering/{}/{}/{}/{}/".format(apikey, homology,
    jc_method, genesets)
```

```
jData = retrieveApiUrl(url)

# Print the results if successful
if jData is not None:
    print("TaskID = " + jData)
    print("")

    # Save the TaskID for later
    TaskID = jData

# Wait for the task to complete.
waitForToolToFinish()

# Check to see if the task really has completed successfully.
url = "/api/tool/get/status/{}/".format(TaskID)
jData = retrieveApiUrl(url)
if jData is not None:
    print("Status = " + jData)
    print("")

# Call the API
url = "/api/tool/get/link/{}/{}/{}/"
    .format(apikey, TaskID, FileType)
jData = retrieveApiUrl(url)

# Print the results if successful
if jData is not None:
    result_link = "http://{}{}".format(host, jData)

    # Follow this link to get the result file.
    print("File is located at: " + result_link)
    print("")

"""
Combine:
"""

# Replace with the desired parameters.
homology = "Included" # ["Default", "Included","Excluded"]
projects = "3323:2404"
FileType = "odemat" # Combine can get ["odemat"]

# Call the API
url = "/api/tool/combine/byprojects/{}/{}/{}/"
    .format(apikey, homology, projects)
jData = retrieveApiUrl(url)

# Print the results if successful
if jData is not None:
    print("TaskID = " + jData)
    print("")

    # Save the TaskID for later
    TaskID = jData
```

```
# Wait for the task to complete.
waitForToolToFinish()

# Check to see if the task really has completed successfully.
url = "/api/tool/get/status/{}/".format(TaskID)
jData = retrieveApiUrl(url)
if jData is not None:
    print("Status = " + jData)
    print("")

# Call the API
url = "/api/tool/get/link/{}/{}/{}/"
    .format(apikey, TaskID, FileType)
jData = retrieveApiUrl(url)

# Print the results if successful
if jData is not None:
    result_link = "http://{}{}".format(host, jData)

    # Follow this link to get the result file.
    print("File is located at: " + result_link)
    print("")

"""
Get Results by User:
"""

# Call the API
url = "/api/get/results/byuser/{}/".format(apikey)
jData = retrieveApiUrl(url)

# Print the results if successful
if jData is not None:
    for task in jData:
        for key in task[0]:
            print(key + " : " + str(task[0][key]))
        print("")

"""
GeneSet Upload:
Note: This section assumes that the file "tutorial_example_data.txt"
exists in the current directory
"""

url = "/api/add/geneset/byuser/{}/".format(apikey)

# Replace with your actual file
file_path = "tutorial_example_data.txt"

jData = None
with open(file_path, 'r') as file:
    file_text = file.read()
```

```
    formData = json.dumps({ "gs_name": "Test 1",
        "gs_abbreviation": "rat heroin-seeking",
        "gs_description": "Test Description",
        "gs_threshold_type": "3",
        "permissions": "private",
        "pub_pubmed": "19664213",
        "sp_id": "3",
        "gene_identifier": "gene_7",
        "file_text": file_text })

    jData = postFormAndRetrieveApiUrl(url, formData)

# Print the results if successful
if jData is not None:
    print(jData)

"""
GeneSet URL Upload:
"""

url = "/api/add/geneset/byuser/{}/".format(apikey)

# Replace with your actual file url
file_url = "http://geneweaver.org/docs/tutorial_example_data.txt"

formData = json.dumps({ "gs_name": "Test 2",
    "gs_abbreviation": "rat heroin-seeking",
    "gs_description": "Test Description",
    "gs_threshold_type": "3",
    "permissions": "private",
    "pub_pubmed": "19664213",
    "sp_id": "3",
    "gene_identifier": "gene_7",
    "file_url": file_url })

jData = postFormAndRetrieveApiUrl(url, formData)

# Print the results if successful
if jData is not None:
    print(jData)
```

Top ↑

FAQ

FREQUENTLY ASKED QUESTIONS

Q: What is GeneWeaver? What happened to “The Ontological Discovery Environment”?

Q: How is GeneWeaver different from gene set enrichment or ontology over-representation tools?

Q: How do I add my own gene sets to GeneWeaver?

Q: I got great results, but how do I make a high resolution image for my presentation?

Q: How do I add Open Biological Ontology annotation to my gene set?

Q: How do I change the abbreviation, name etc. for my gene set?

Q: I set my threshold too high/low. How do I change it?

Q: I uploaded a file with 200 genes, but it says that my gene set is empty?

Q: A public gene set is improperly labeled. How do I report this?

Q: How are homologous genes identified?

Q: My gene sets are listed as 'deprecated'. What does this mean?

Q: How should I cite GeneWeaver in my research?

Q: What do all the acronyms on the site stand for?

ANSWERS

Q: What is GeneWeaver? What happened to "The Ontological Discovery Environment"? A. The Ontological Discovery Environment was conceived of as a tool for the integration of biological functions based on the molecular processes that subserved them. From these data, an empirically derived ontology may one day be inferred. Sounds like a mouthful? We think so, too. Moreover, our acronym, ODE, sounds like "ordinary differential equations", "open development environment", "Ohio Department of Education", and the airport in Odense, Denmark. Our users have found the system valuable for a wide range of applications in the arena of functional genomic data integration. While the underlying algorithms of The Ontological Discovery Environment can be extended to many contexts, we chose to rename the system "GeneWeaver" to reflect the emphasis on genes and genomes, allowing our users to weave together the many complex relations among processes, pathways and functions implicit in functional genomics experiments.

Q: How is GeneWeaver different from gene set enrichment or ontology over-representation tools? There are many statistical tools for the analysis of gene set overrepresentation, and it is indeed possible to perform similar analyses using some of the functions in GeneWeaver. However, GeneWeaver's primary focus and strength is in using gene sets to organize biological functions. GeneWeaver enables highly flexible set-set comparisons of both user submitted and curated gene sets. The suite of combinatorial tools enable large collections of user submitted tools to be compared to each other, and the hierarchical similarity tools enable classification and organization of gene sets based on the genes they contain. This allows discovery of hidden relations among common biological processes, even if those processes have been studied using highly diverse species, analytic methods and approaches. The GeneWeaver tools provide facile data integration and harmonization, and enable user directed integration of new and published results. Major incorporated data from other resources provides a wealth of other sources of contextual information which facilitate interpretation of these discoveries.

Q: How do I add my own gene sets to GeneWeaver? A: There are step-by-step instructions available in the Uploading Gene Sets section.

Q: How do I add Open Biological Ontology annotation to my gene set? A: Browse to your GeneSet and click the "edit" link. Scroll to the bottom of the page and use the Tree Browser to select entries for your GeneSet. To change the OBO source, use the drop box at the top of the tree display. Finally, to remove any extraneous entries, you can use the little red 'x' on the left side. After saving the changes, your new information will be displayed and your GeneSet will be searchable using any of the ontologies selected.

Q: How do I change the abbreviation, name etc. for my gene set? A: Go to the View My GeneSets page, the link is found in the Manage Gene Sets column. Scroll to your GeneSet and click the "edit" link. Then simply change the values and save your changes. The new text will be displayed immediately.

Q: I set my threshold too high/low. How do I change it? A: Go to the View My GeneSets page (the link is found in the Manage GeneSets drop-down on the navigation bar). Scroll to your GeneSet and click the link on the gene set name in order to see the GeneSet Information page. Click the Set Threshold button. Then simply fix the thresholds and save the changes. The new thresholds will be applied immediately.

Q: I've got great results, but how do I make a high resolution image for my presentation? A: Each tool has a link to export the result as a PDF. Save the file and open it in Adobe Acrobat, Inkscape or other software. Save as PNG. This PNG file can be easily inserted into MS Powerpoint presentations or Word documents.

Q: I uploaded a file with 200 genes, but it says that my gene set is empty? A: If there was no error reported, you probably set your threshold too high/low, see the previous question. If there was an error, your data probably uses a different microarray or gene ID type than what was provided on the upload page.

Q: A public gene set is improperly labeled. How do I report this? A: From the GeneSet's information page, click the bug icon on the top navigation bar and let us know what specifically needs updating. Include the "GS" number.

Q: How are homologous genes identified? A: We use homologene along with any information provided by the reference genome. ex: RGD provides MGI ids as well.

Q: My gene sets are listed as 'deprecated'. What does this mean? A. If a newer version of a Gene Set in one of your projects is available, the version you stored is marked "deprecated." Clicking on the provided icon will update your project with the latest version of this data. New versions are available when we update data from external sources, e.g. MP and GO annotations, or when the GeneSet Metadata has been updated.

Q: How should I cite GeneWeaver in my research? A: Please cite: 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. Please visit other relevant publications.

Q: What do all the acronyms on the site stand for? A: DRG (Drug-Related Genes), CTD (Comparative Toxicogenomics Database), MP (Mammalian Phenotype Ontology), HP (Human Phenotype Ontology), ABA (Allen Brain Atlas), GO (Gene Ontology), MeSH (Medical Subject Headings).

Top ↑

Searching GeneWeaver

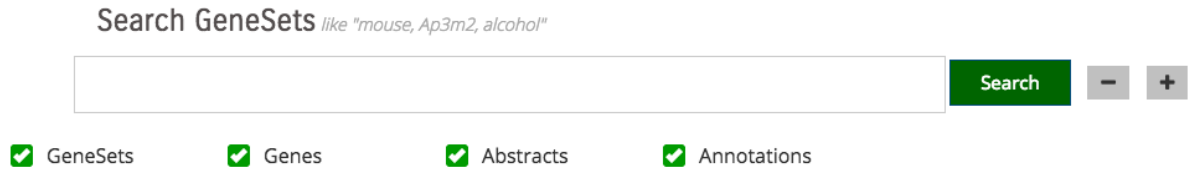
Our database includes data obtained from numerous **external data resources**. GeneWeaver allows users to conduct text searches on metadata and raw data stored in our database. These include searches by permission level, species, curation tier, gene set information or genes of interest. Occasionally it is useful

to search for gene sets anchored on genes or gene sets of interest based on their overlap with neighboring gene sets. **Anchored Biclique of Biomolecular Associations (ABBA)** is a tool that allows you to accomplish this task.

General Search

The general GeneWeaver database search is available from the main page or by following the **search** icon in the header (magnifying glass).

You can limit searches to ***GeneSets***, ***Genes***, ***Abstracts*** or ***Annotations***. By default, searches will be performed across each domain. In addition, by selecting the **-** or **+** icons next to the search bar, you can add additional parameters. Each parameter that is added will be evaluated as an ***and*** operator.

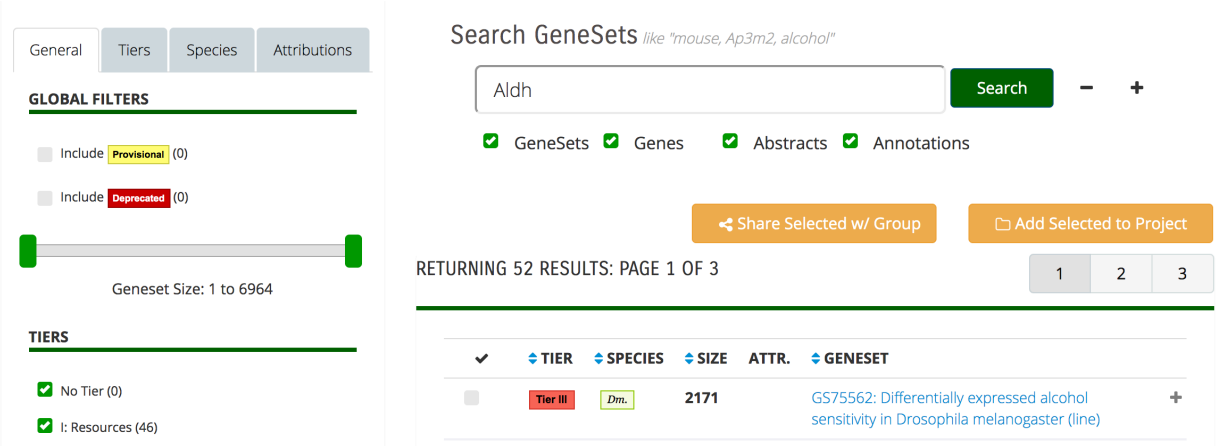


Search Results

Search results will be limited to your appropriate permission level, so private genesets will not be shown unless you are the owner, or they have been explicitly shared with you via group permissions. In addition, only the top **1000** matches will be displayed.

Step 1: Search

Here is an example of the results when searching for the gene "Aldh".



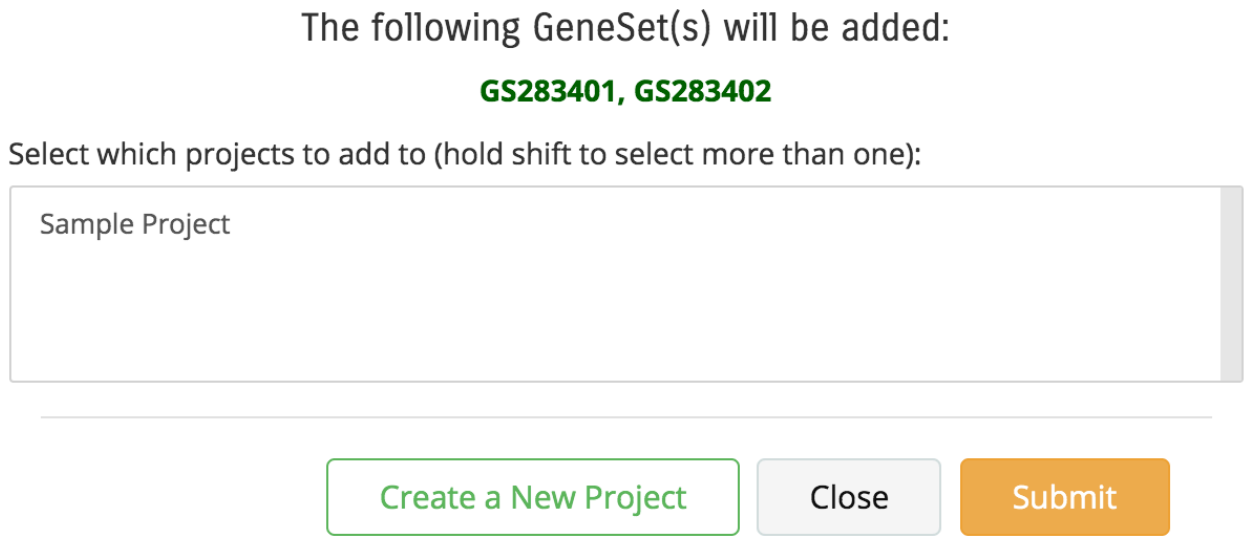
The resulting 52 GeneSets are displayed on 3 separate pages. Each page can be viewed by clicking on the number (1, 2, or 3) shown in the gray boxes.

Step 2: Put the Results into a Project

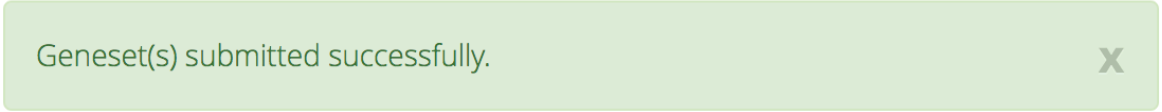
A project is needed in order to use any of the analysis tools. Select one or more GeneSets to add to the project by clicking in the checkbox on the left side of each GeneSet. Now click the Add Selected to Project button.

Note that a new project can be created or the selected results may be added to one of your existing projects. If you do not have any projects the only option will be to create a new project.

Once the project is created, you must select it (and/or any existing ones) from the list in the dislog box and submit.



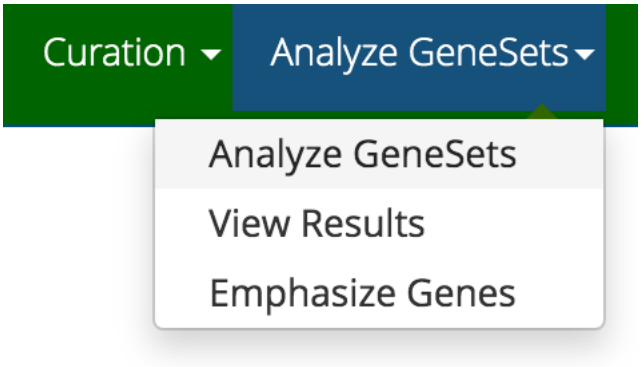
A message indicating success or an error will be displayed on the search page.



Note that selected GeneSets can also be shared with groups by using the Share selected w/Group button. Learn more about groups on the Users and Groups page.

Step 3: Analyze

Go to the Analyze page in order to use GeneWeaver's tools on your project(s) and gene sets. The link is on both the header and footer.



For details go to the Analysis Tools page.

Advanced Search

GeneWeaver's search is performed using the Sphinx search engine. As a result, the search box accepts Sphinx-based shortcuts.

The words **OR** and **NOT** are translated automatically into sphinx operators " | " and " - " respectively.

Currently defined field names are:

- GeneSet Metadata: *@name_*, @description_, *@label_*, @attribution_
- Publication Metadata: *@pub_authors_*, @pub_title_, *@pub_abstract_*, @pub_journal_, _@pmid_
- Group-Specific Access: _@group_
  - names of groups shared with (use quotes if necessary), or "Public"/"Private"
- All Gene Identifiers and Aliases: _@gene_
  - Does not currently map to microarray probes or homologous genes
- Ontology Terms: _@term_
  - includes unique ID, name, and term description
- Species: *@species_*, @taxid_
  - scientific name, or NCBI taxid (9606=human, 10090=mouse, etc)

Example Search Terms

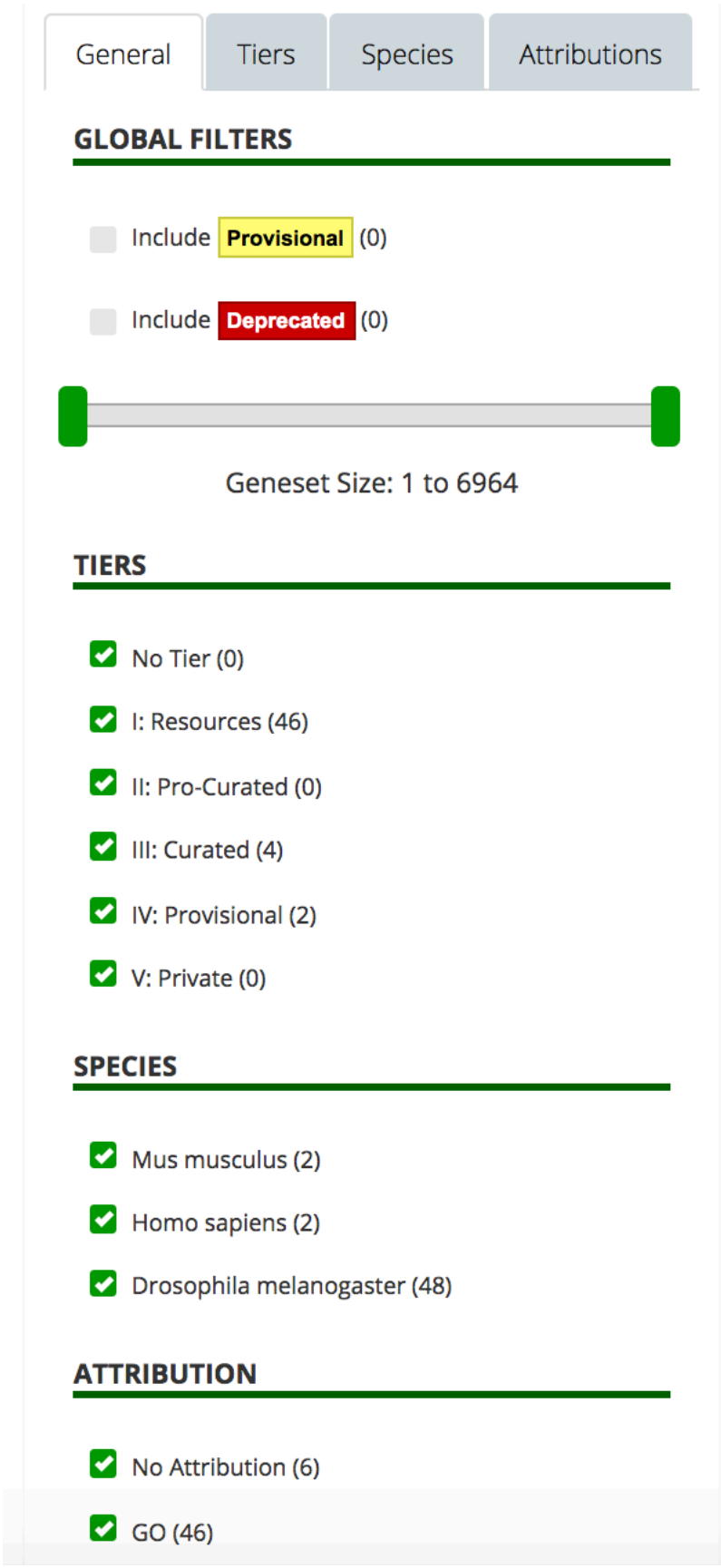
- `alcohol preference`
- or, more precise: `alcohol preference -QTL`
- or, even more precise: `alcohol preference -QTL @species rattus`

Search all fields for striatum and gene *Mobp*:

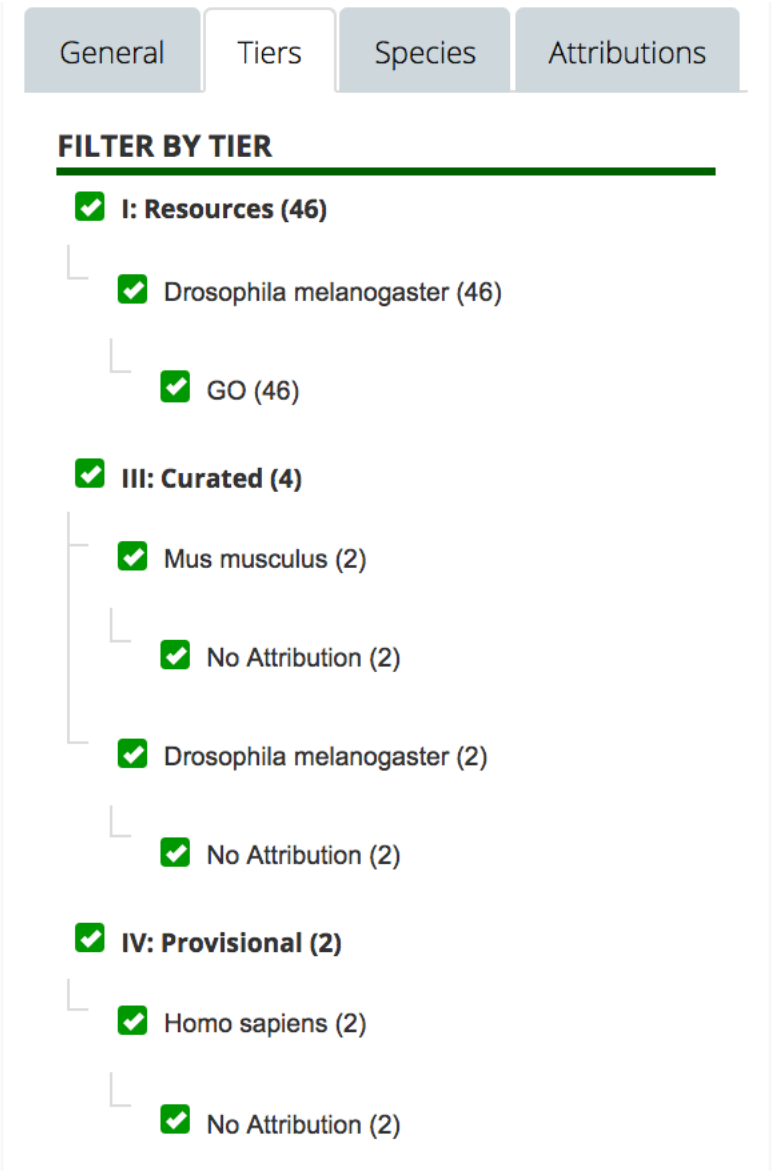
- `MA:0000891 Mobp`
- or, more precise: `@term MA:0000891 @gene Mobp`

Filtering Search Results

There is a filter section on the left side of the page.

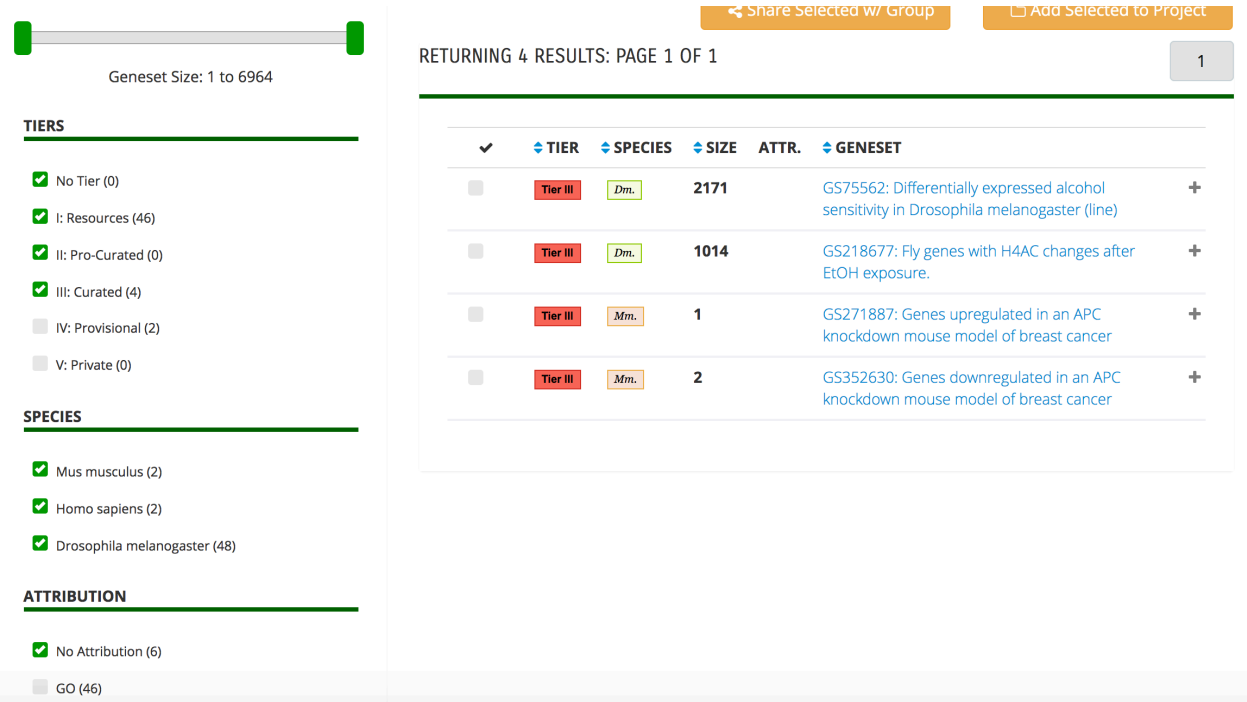

General
Tiers
Species
Attributions
GLOBAL FILTERS
Include Provisional (0)
Include Deprecated (0)
Geneset Size: 1 to 6964
TIERS
No Tier (0)
I: Resources (46)
II: Pro-Curated (0)
III: Curated (4)
IV: Provisional (2)
V: Private (0)
SPECIES
Mus musculus (2)
Homo sapiens (2)
Drosophila melanogaster (48)
ATTRIBUTION
No Attribution (6)
GO (46)

The tabs will allow you to reduce the scope of the results based on **Tier**, **Species**, and **Attributions**.



By managing these filters, you can easily reduce the results of complex queries.

The search filters are also shown on the general tab. Here is an example of how to reduce the 52 results found in the aldh gene search by de-selecting Tier IV, Tier V, and the GO attribution so that only a few genesets are shown.



You can limit the GeneSets to only show those under a specific **GeneSet Size** by moving the slider.

The results can be sorted using the header over the table of results.

For more information on the meaning of terms such as tier, check out these pages: Curation Tiers and General Definitions.

Top ↑

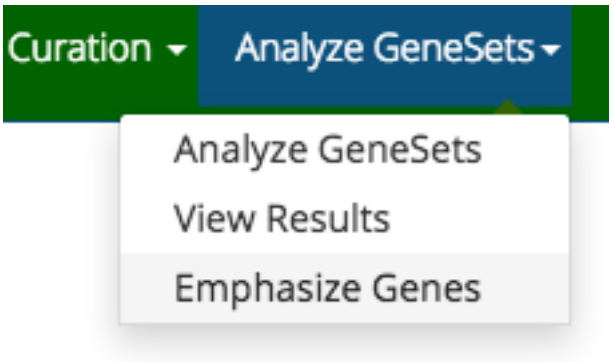
Gene Set Utilities

GeneSet Details Pages allow users to view vital information about gene sets of interest, including associated genes, homologs and references to external links. **Gene Intersection Lists** are useful for determining which information is shared between gene sets of interest. In addition, GeneWeaver tools allow users to **Combine** gene sets of interest or perform more complex set operations based on **Boolean Algebra**. Gene sets may also be annotated with information about **Emphasis Genes**, allowing users to augment GeneWeaver tools with gene-specific information.

Emphasis Genes

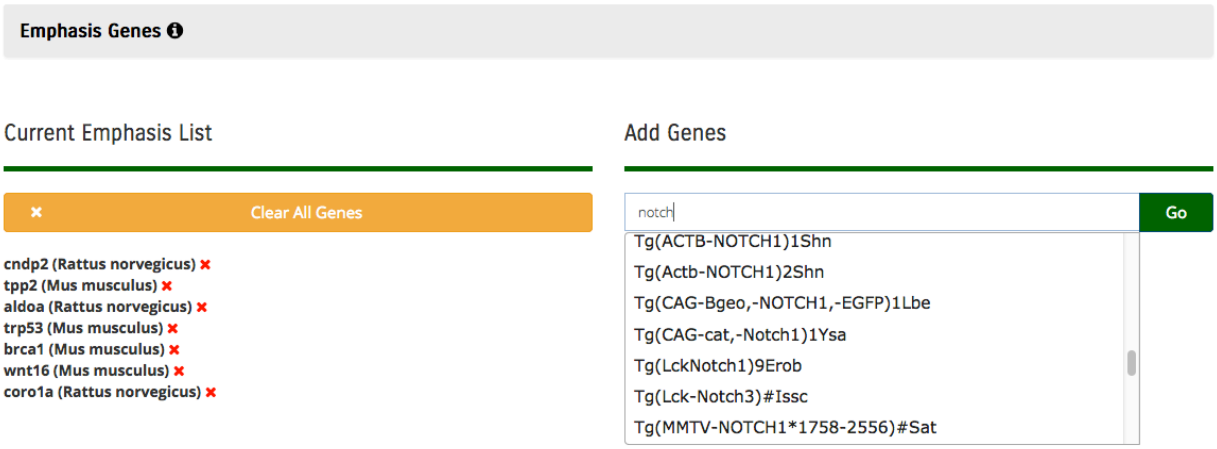
The Emphasis Genes utility enables users to select genes or an entire set of genes that may be highlighted in various analysis tools.

To set emphasis genes choose "Emphasize Genes" from the Analyze GeneSets drop-down on the navigation bar or from the footer.

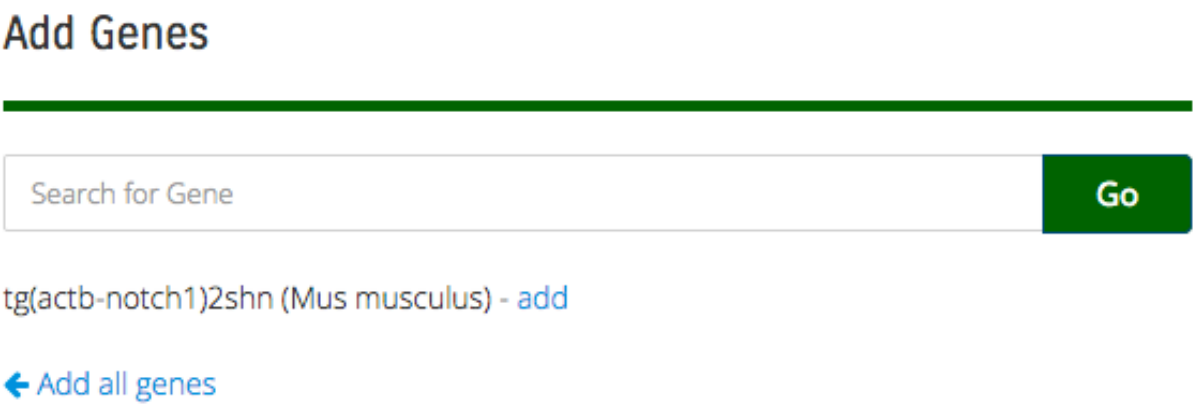


The current emphaisis genes are listed on the left side of the page.

To modify your emphasis genes, you can remove genes one at a time using the "x" icon next to each gene. To clear the entire list, click the "Clear all genes" button at the top of the page.



To add a gene, type the gene name or part of it in the box on the right side of the page. A list will appear based on the partial name. Select one and click the "Go" button.



The gene or genes if the selection included several, will be listed on the page. Use the "Add all genes" or "Add" link to select the desired gene(s).

Homology Mapping

GeneWeaver uses the concept of Homology Mapping to expand search and analysis capabilities beyond a single species. Currently, we rely on data provided by Homologene to assert homology between clustered sets of reference gene ids. That is, GeneWeaver creates a set of unique id clusters (representing Entrez, Ensembl, Gene Cards, etc.) representing specific genes, these clusters are connected across species using mappings established by Homologene.

Gene Intersection Lists

Gene Intersection Lists are useful for determining which information is shared between gene sets of interest.

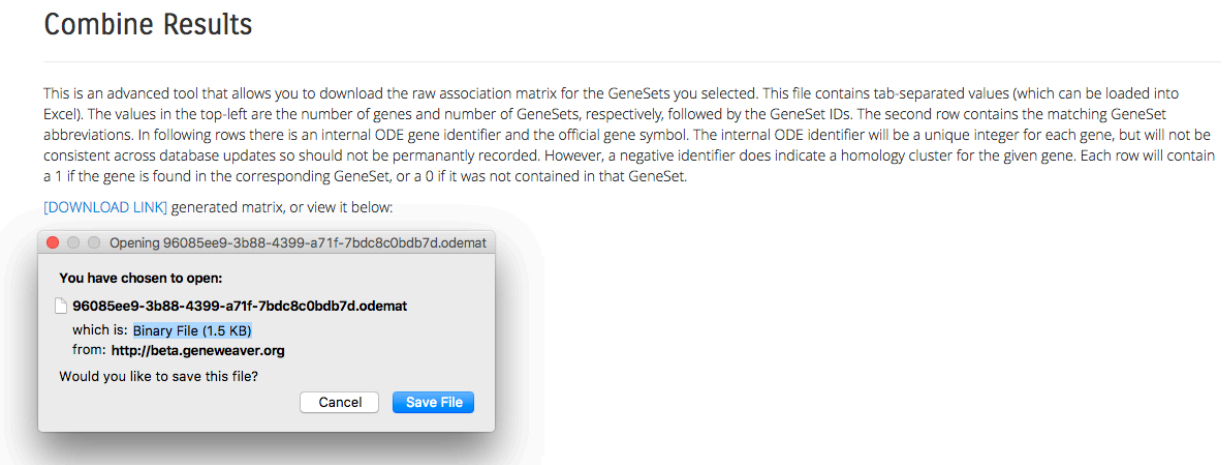
Gene intersection lists can be generated by clicking on the output of various tools including the Hypergeometric tests, Jaccard similarity matrix Venn diagrams and HiSim Graph nodes. A table of genes by GeneSets is displayed. Next to each gene symbol are links to gene specific queries of external resources. Each gene has links to associated databases, such as NCBI, Ensembl, STRING, MGI, GeneNetwork, etc. For users with the FireGoose GAGGLE extension installed, you will also find the genes on the page available for broadcast on the page. Filled circles indicate the presence of a gene in a GeneSet. Green (light) circles indicate that the exact gene is present in multiple gene sets. Dark (maroon) circles indicate a homologous gene is present in multiple gene sets. The table can be exported using the export .csv feature at the bottom of the window.

Combine

GeneWeaver tools allow users to combine gene sets of interest. GeneWeaver tools operate on a weighted bi-partite adjacency matrix, a table of Association Scores in a Gene (row) x GeneSet (col) tab delimited text format. For many GeneSets, the scores are binary.

To create sample GeneWeaver data for development or off-line analysis:

1. Perform a database query using the search field.
2. Add the GeneSets to a project.
3. Go to the "Analyze GeneSets" page.
4. Select the project or specific GeneSets from projects.
5. Select the "Combine GeneSets" tool, pick homology included or excluded and click run.
6. Save the file to your computer.



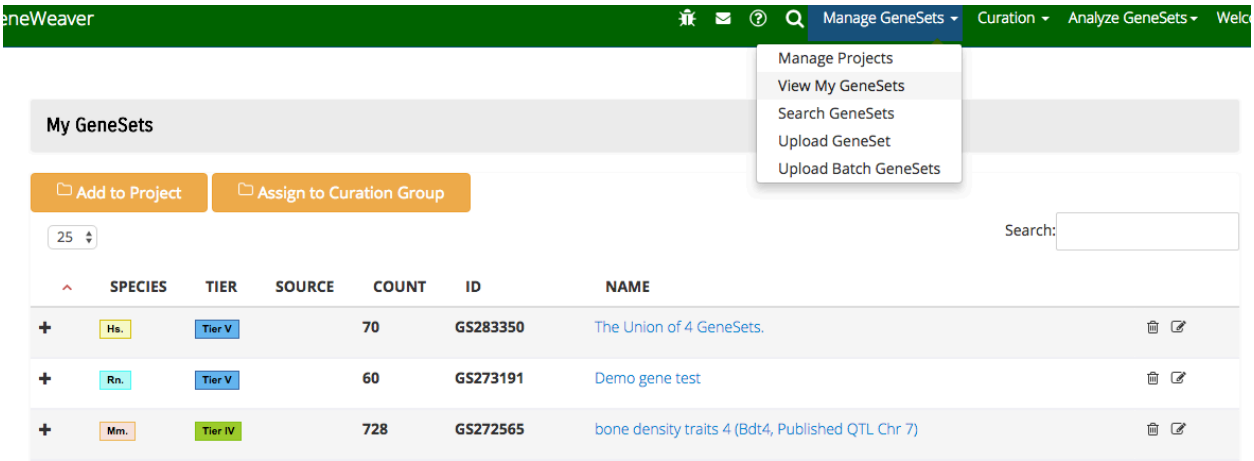
External Data Resources

GeneWeaver contains publically available sets of genes annotated to structured vocabularies and ontologies that are assigned Tier I, or public resource data. Other sets of genes, such as MeSH term-to-gene annotations, are derived from the processing of public sources and attributed to Tier II. In the case of MeSH, we take advantage of NCBI's gene-to-Pubmed and Pubmed-to-mesh files to produce sets of genes annotated through their transitive associations.

Top ↑

View My Genesets

Genesets that you added are listed on the View My Genesets page. They can be added by uploading or using some of the tools, such as the boolean algebra tool.

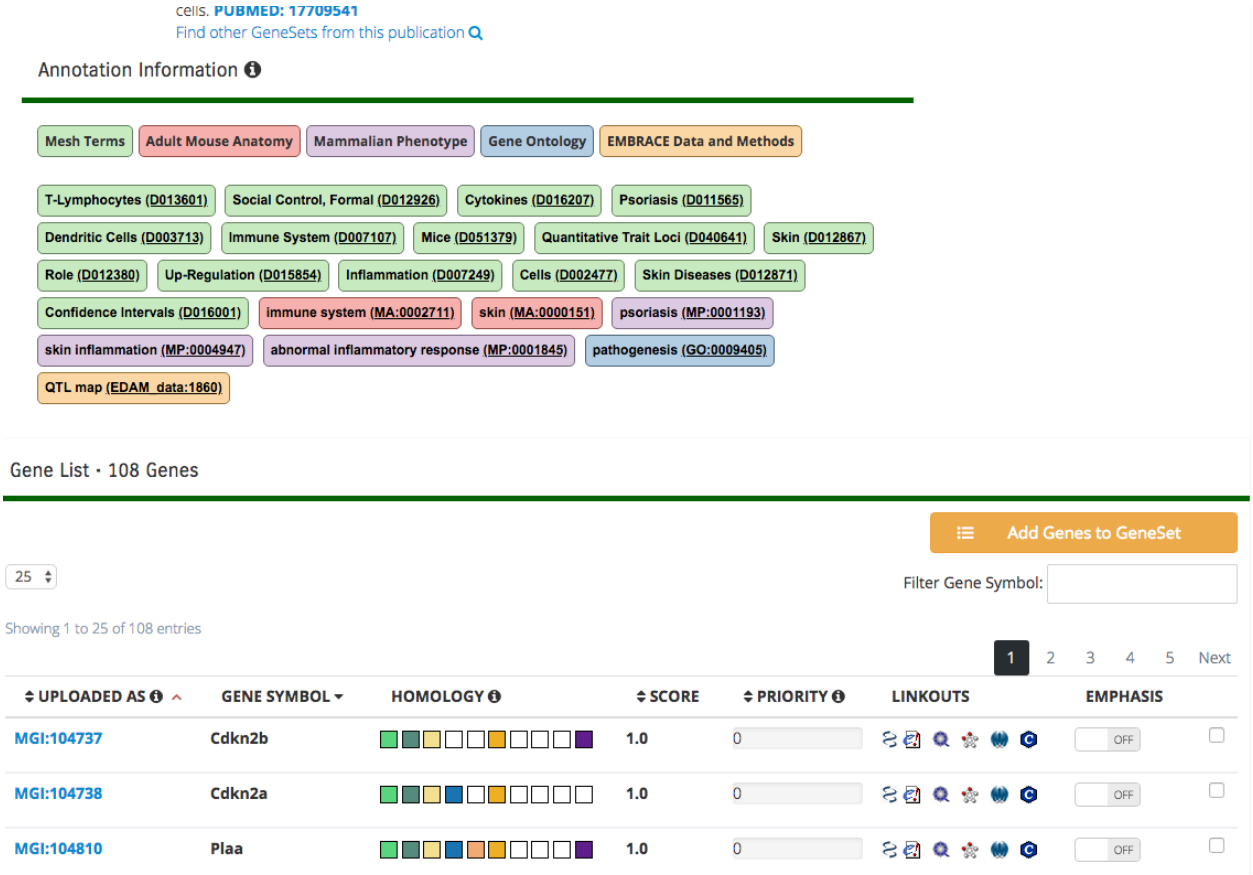


Clicking on a geneset on this page will highlight it in yellow. Several can be selected this way and then added to a project or assigned to a curation group. The list can be sorted by clicking on a column header. Typing in the Search box will filter the list of genesets. The filter is case sensitive. Clicking the link on the geneset name will open the geneset details page. Clicking the edit icon will open the edit geneset page.

Geneset Details Pages

This page provides a comprehensive look at all the information that has been entered about a geneset. You can get to this page by clicking a link on the geneset name from any page that lists genesets, such as the My Geneset page, search results and some of the tool results.

The basic information is displayed here in detail: geneset name, geneset id number, tier, description, figure label, score type, date added, date of the most recent update, species and the publication information: URI, authors, title, journal and abstract. Scroll the page down to see the color coded annotation information from several ontology databases. Click on a link for any term to open the ontology webpage describing the term.



Further down on the page is a list of all the genes. If the list is long, it will be displayed using several pages. The "uploaded as" column shows the identifier used when this gene was uploaded. Select a choice in the "gene symbol" column to show the corresponding identifier in various other formats. Mouse over the "homology" boxes to see homology mappings to other species in GeneWeaver. The "linkouts" column contains icons allowing you to link to other websites, including Entrez, Ensembl, Gene Network, String, Allen Brain Atlas and Comparative Toxicogenomics Database. Other columns include the score, priority and emphasis.

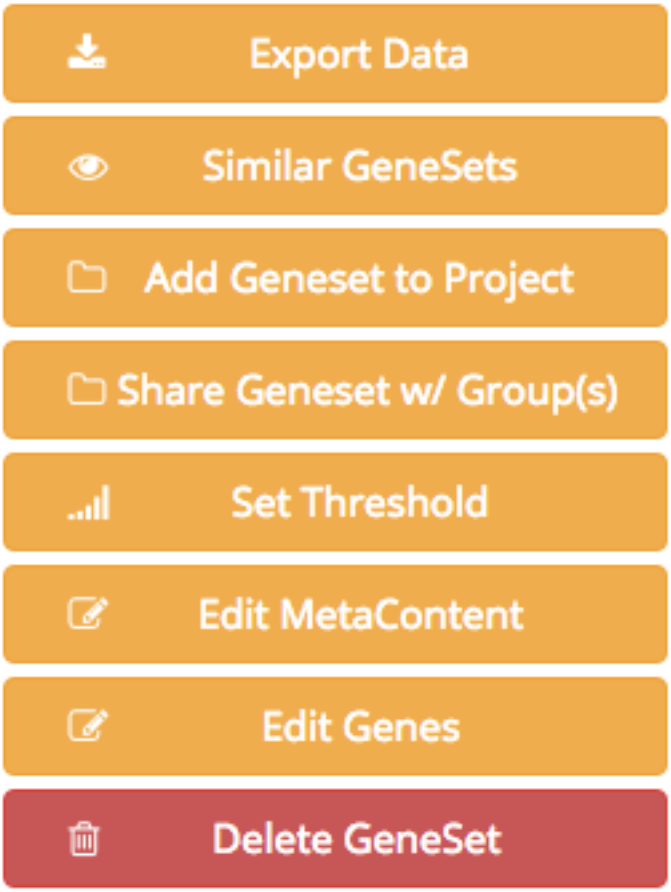
Check the box in the final column to select that gene to be added to another geneset by using the "Add Genes to GeneSet" button.

The sort order of the columns can be changed by clicking on the uploaded as, score or priority columns. The genes listed can be limited by entering a gene in the "Filter Gene Symbol" box.

At the top right of the page are several buttons:

- Export Data has 3 formats:
  1. Export Data: a text file containing all the gene symbols for each gene
  2. Export OmicsSoft
  3. Export GeneSet Complete: a text file that can be used by GeneWeaver's batch upload function.
- Similar Genesets will open a page showing the top 1,000 GeneWeaver genesets that are similar to this one.
- Request Curation
- Add Geneset to a Project
- Share Geneset w/ Group(s)

If you originally created the geneset that is displayed, then there are more functional buttons present that allow you to make changes.



The "Set Threshold" button opens a new page where you can change the significance threshold.

The "Delete GeneSet" button will ask you to confirm that you want the geneset removed.

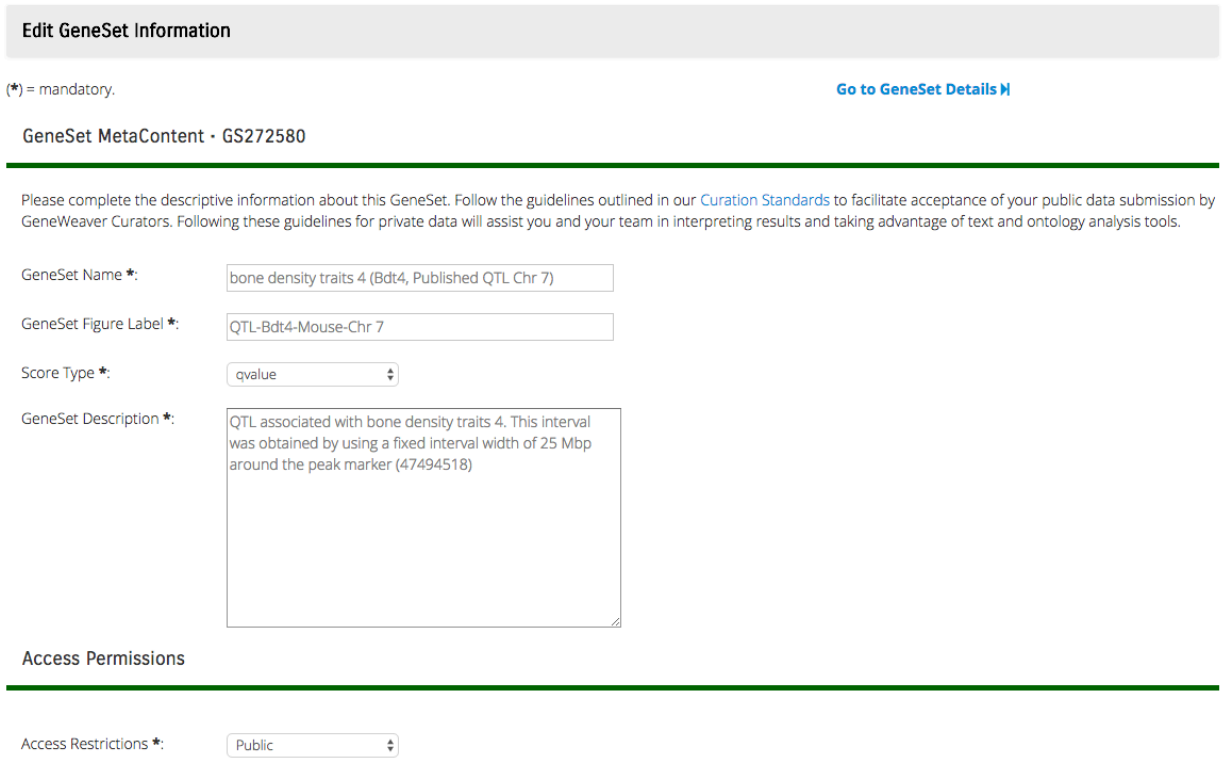
Using the "Edit MetaContent" and "Edit Genes" buttons will open the edit geneset page.

Edit Geneset Page

You get to the edit geneset page from the geneset details page or from the upload geneset page. On this page is both a link and a button you can use to go to the geneset details page. Be sure to click on **"Save Updates"** before leaving the page if you have made any changes.

Edit MetaContent

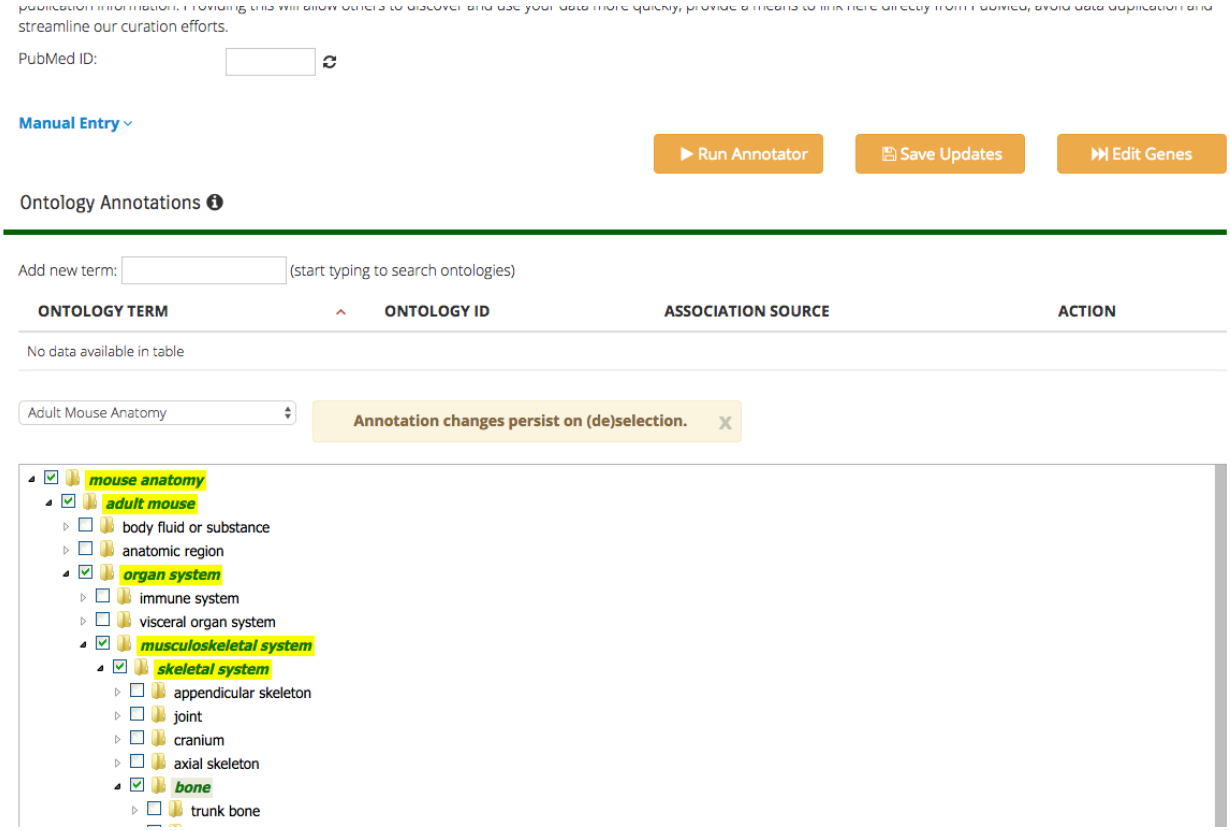
Click the "Edit MetaContent" button and the top portion of the page changes to a format that allows editing.



Here you can change the name, figure label, score type, description and access restrictions. If you know the **PubMed ID**, enter it and click the link next to the box for it to be looked up. Alternatively, click on "Manual Entry" and fill in the information.

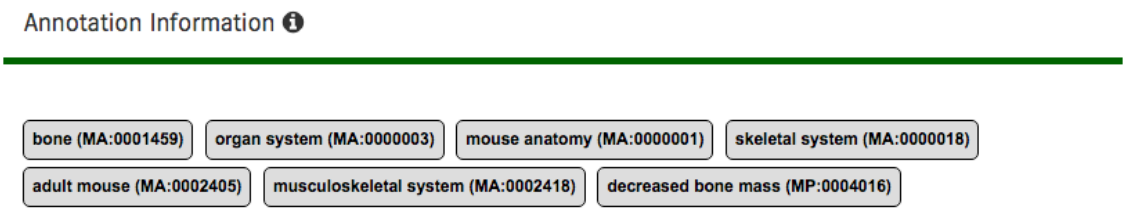
Ontology Annotations

Scroll below the publication area to see the ontology annotations.



You can enter a term in the box to search the ontologies for it. Click to select the desired one. Or select an ontology from the selection box. Click to expand the hierarchy and check the desired term(s).

Click the "Save Updates" button.

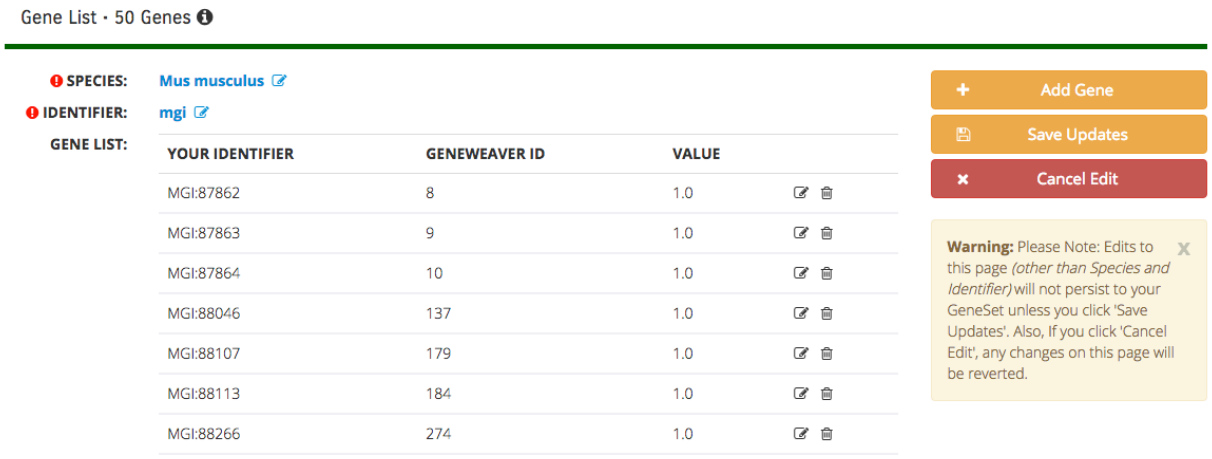


In the Edit Metatdata mode, the ontology terms are displayed in a fashion that allows removal. The ontology columns can be sorted by clicking on the header.

ONTOLOGY TERM	ONTOLOGY ID	ASSOCIATION SOURCE	ACTION
adult mouse	MA:0002405	Manual Association	Remove
bone	MA:0001459	Manual Association	Remove
decreased bone mass	MP:0004016	Manual Association	Remove
mouse anatomy	MA:0000001	Manual Association	Remove
musculoskeletal system	MA:0002418	Manual Association	Remove
organ system	MA:0000003	Manual Association	Remove
skeletal system	MA:0000018	Manual Association	Remove

Edit Genes

Click the "Edit Genes" button to see an editable list of all the genes in the geneset. They will be displayed on the screen below the annotations.



In the editing mode, you can change the species or identifier. Click on the edit icon for a gene and a form will open so you can edit the identifier or score. Click the trash icon to remove a gene from the geneset. Click on the "Add Gene" button to add another gene to the geneset. Make sure to click on "Save Updates" when you are done.

Similar Genesets

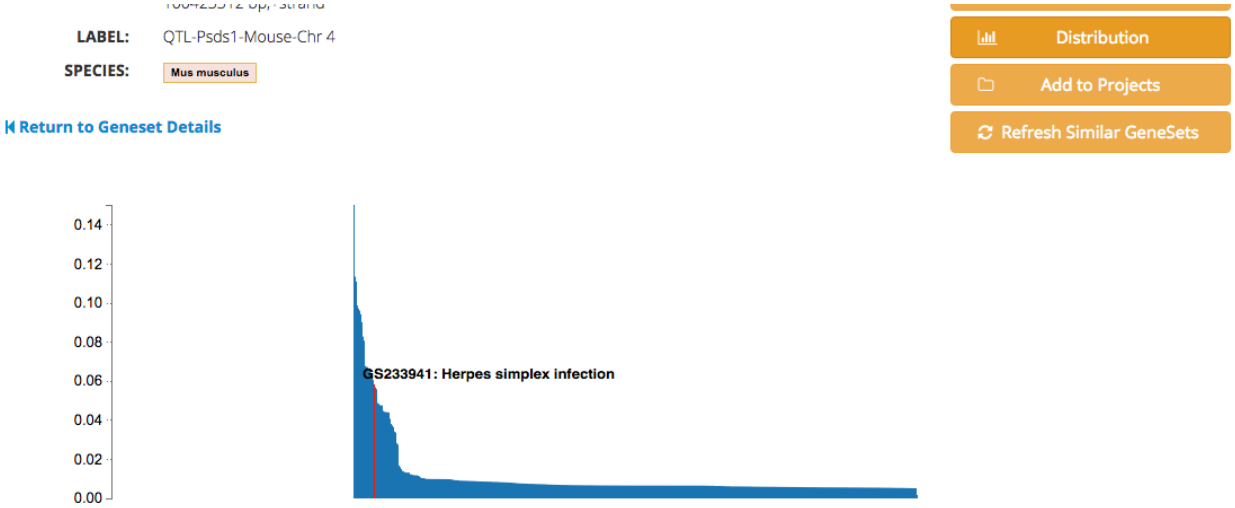
The view geneset details page has a button linking to this page. A message will be displayed if a similarity analysis needs to be run on the geneset with an option to "Click here to start now". There also is a button on the page that allows you to "Refresh Similar GeneSets" if the analysis is old.

The "Export GeneSets" button will create a "csv" file of all the similar GeneSets. The columns include: geneset id, name, number of genes, and Jaccard Similarity score.

Scroll down to see the list of similar genesets. You may select between 10 and 100 to display per page. This list will be sorted by the Jaccard Similarity. Click on any column to change the sort order. The tier, species and attribution columns allow selecting a filter in order to limit the number of genesets. You may also enter a string of characters into the "Search" box to filter the list by the geneset name.

Check the box to the right of any genesets and use the "Add to Projects" button if you desire to keep a selection of these genesets for use later.

Click on the "Distribution" button to add a distribution graph to the page.



Hover your mouse over the graph to see where each geneset is plotted.

Software

Public access to the GeneWeaver **analysis codebase** along with appropriate **schema build scripts** is available.

Please **contact** the GeneWeaver Team for information on how a new module may be incorporated into the GeneWeaver environment.

Top ↑

Installation

The GeneWeaver interface is open source and freely available from our git repository hosted by Bitbucket. *Although, due to security, Bitbucket is password protected. Please contact us for appropriate permissions.*

Top ↑

Data

Available Data

Many of the publications referenced by GeneWeaver GeneSets have been collected into a version X8 EndNote formatted library (zip file) that can be downloaded by clicking here.

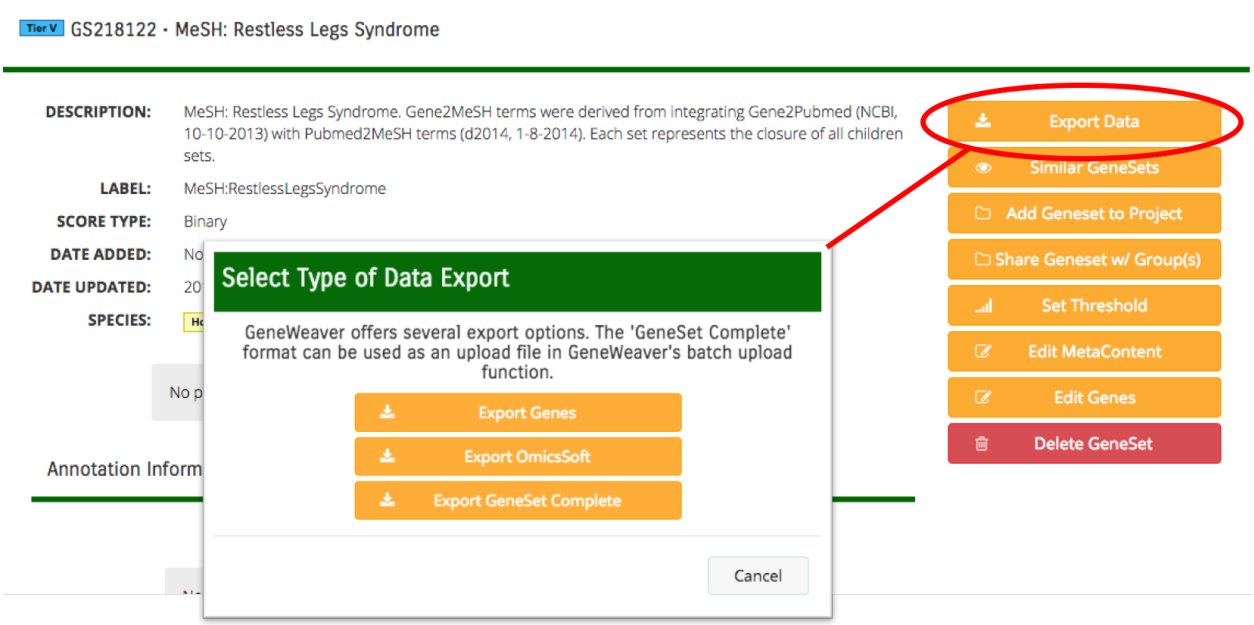
In addition, GeneWeaver creates a daily summary of data sources and related information, which can be found here.

Data Export

GeneWeaver also allows users to download genesets that they have permission to access. Available formats are:

- **Genes**: This will produce a tab-deliminated file of all genes in the geneset, along with each gene's identifier for each database that GeneWeaver tracks.
- **OmicsSoft**: This option produces an appropriate text file for inport into the OmicsSoft software suite. Visit OmicsSoft for more information about their acceptable format. Since *Omicsoft* fields are slightly different than GeneWeaver data fields, not all information is mapped in a one-to-one fashion. The available *Omicsoft* mappings are as follows:
  - *Source* == GeneWeaver Generated
  - *Name* == GeneWeaver GeneSet Name
  - *Description* == GeneWeaver GeneSet Description
  - *Tag* == GeneWeaver
  - *Type* == This field is only populated if the set is originally uploaded via the OmicsSoft upload tool.
  - *Project* == This field is only populated if the set is originally uploaded via the OmicsSoft upload tool.
- **Export Complete GeneSet**: This option will produce an export file formated for *Batch Upload* back into GeneWeaver.

In order to use the export function, visit an available GeneSet page, and select ***Export Data*** from the right hand column (see the Figure below). This will bring up a modal where you can select the appropriate format. Depending on your browser settings, the download should start automatically.



Top ↑

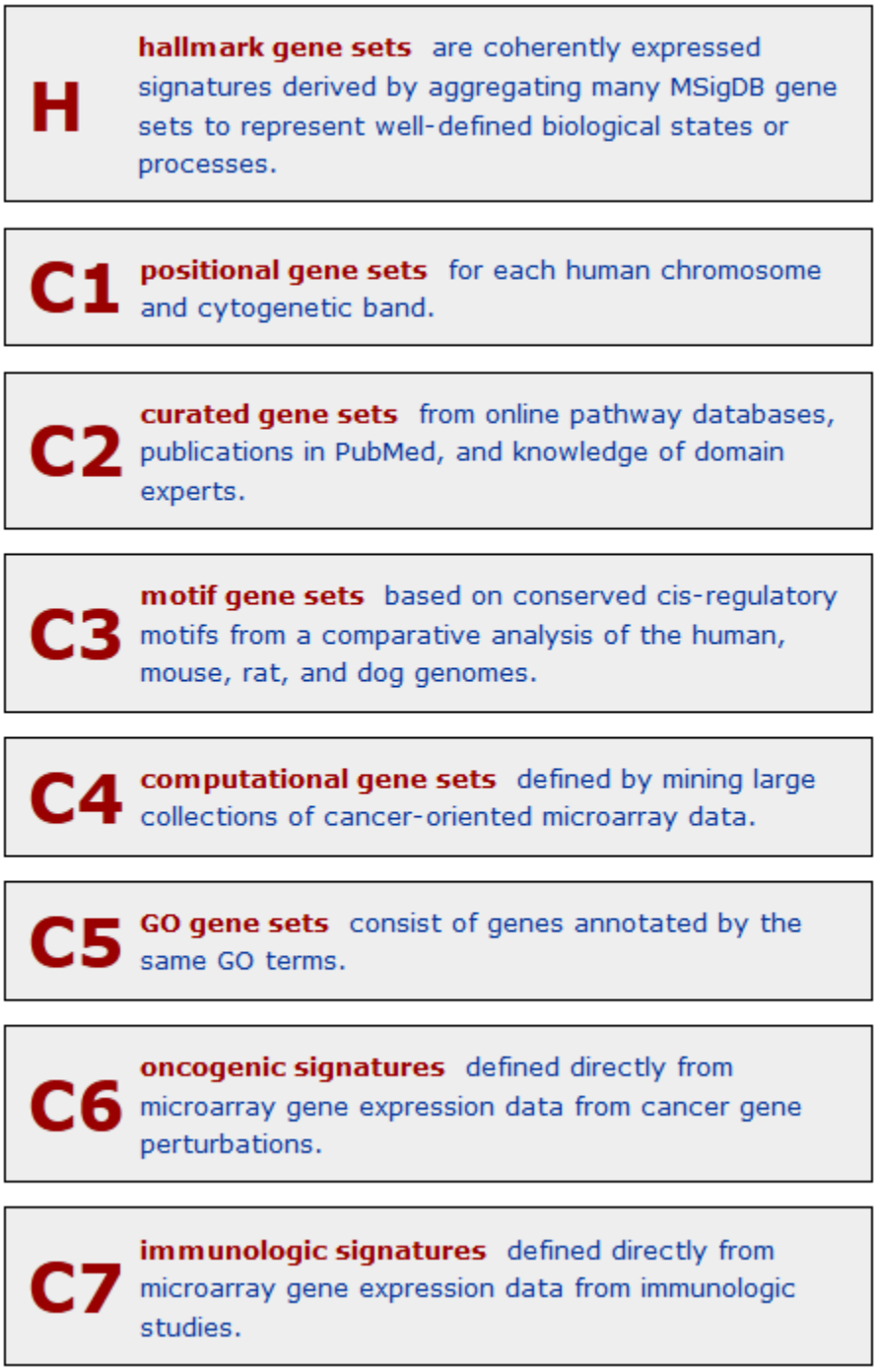
External Data Resources

GeneWeaver contains publically available sets of genes annotated to structured vocabularies and ontologies that are assigned Tier I, or public resource data. Other sets of genes, such as MeSH term-to-gene annotations, are derived from the processing of public sources and attributed to Tier II. In the case of MeSH, we take advantage of NCBI's gene-to-Pubmed and Pubmed-to-mesh files to produce sets of genes annotated through their transitive associations.

Tier	Resource	Description	Number of Gene Sets (2012)	Number of Gene Sets (2015)	Number of Gene Sets (2018)
1	Allen Brain Atlas (ABA)	Sets containing upregulated genes found within mouse brain regions and structures. These genes exhibit a >= 2.0 fold change in expression energies compared to all other basic cell groups and brain regions (ABA refers to this area as 'grey' contrast structures). These sets are generated using the ABA API and its differential gene search pipeline.	785	740	785
1	Comparative Toxicogenomics Database (CTD)	Sets of genes associated with CTD chemical-gene interactions are obtained via CTD flat files.	6266	6177	21630
1	Drug Related Gene Database (DRG)	Drug Related Gene Database, compiled bt the Neuroscience Informatics Framework (NIF) contains gene expression data related to drug abuse research.	1208	253	238
1	Human and Mouse Gene Ontology (GO)	Sets of genes from human and mouse annotated to the Gene Ontology (GO), obtained from the Gene Ontology Consortium and MGI.	33668	33668	85573
1	Human Phenotype Ontology Annotations (HP)	Gene sets derived from annotations of genes to HPO.	6276	4011	6276
1	Kyoto Encyylopedia of Genes and Genomes (KEGG)	Pathways derived from the KEGG API are directly parsed for identifiers that map to GeneWeaver. Pathway data for humans, mice, rats, and rhesus monkeys is currently included.	0	1172	1339
1	Mammalian Phenotype Annotations (MP)	Gene sets derived from annotations of mutant mice to MP terms in MGI, with transitive closure.	7966	7966	7931
2	Medical Subject headings (MeSH)	Genes annotated to MeSH terms were aggregated with gene2publication associations from PubMed. Associations must appear in a minimum of two publications. Genes associated with the closure of each set were obtained.	0	12069	12069
1	Molecular Signature Database (MSigDB)	Sets of genes annotated to disease for use with Gene Set Enrichment Analysis (GSEA) downloaded from MSigDB v.5.0. Only sets derived from hallmark, C1, C3, C4, C6, and C7 collections are incorporated*. MSigDB genesets that are curated from other resources (e.g. KEGG or GO) are ignored to eliminate data redundancy.	0	3738	3738

Tier	Resource	Description	Number of Gene Sets (2012)	Number of Gene Sets (2015)	Number of Gene Sets (2018)
1	MouseQTLs from MGI	Sets of positional candidate genes for the confidence interval around all the QTLs within MGD.	0	5050	3405
1	Online Mendelian Inheritance in Man (OMIM)	Gene-disease phenotype data is retrieved from OMIM's Morbid Map and Phenotype Series list. Unconfirmed and spurious mappings are ignored.	0	738	738
1	Pathway Commons (PC)	Sets of genes derived from the "top" pathways: those that are neither controlled nor a pathway component of another biological process. KEGG pathways are removed from this data set to prevent duplicate genesets.	0	1036	1149
1	Rat QTLs from RGD	Sets of positional candidate genes for the confidence interval around all the QTLs within the RGD.	0	2048	2064
1	Genome Wide Association Studies (GWAS)	Catalog of Published Genome-Wide Association Studies	0	0	3389

*Information on the MSigDB file types included in GenWeaver (H, C1, C3, C4, C6 and C7)



Other Important Links

- You can find a detailed description of GeneWeaver's usage, data sharing and privacy policies **here**.
- These pages also include information on GeneWeaver related **Publications** and **How To Cite** GeneWeaver. In addition, our **QR Code** allows you to rapidly connect viewers of your printed work to GeneWeaver.
- Please feel free to **Contact the GeneWeaver Team** with any questions you might have about the

data or tools.

- The GeneWeaver Team would also like to **Acknowledge** the many folks who contribute to and support our efforts. Importantly, the GeneWeaver/The Ontological Discovery Environment was *initiated as a project of the NIAAA Integrative Neuroscience Initiative on Alcoholism (**U01AA13499**, **U24AA13513**), and is supported by **R01 02-AA18776 NIAAA/NIDA**.*

Top ↑

Publications

How to cite GeneWeaver

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**. **Nucleic Acids Research**; (2012) 40(D1): D1067-D1076

Publications Describing GeneWeaver

- 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**. **Nucleic Acids Research**; (2012) 40(D1): D1067-D1076.
- Jason A. Bubier, Michael A. Langston, Erich J. Baker, and Elissa J. Chesler. **Integrative functional genomics for systems genetics in GeneWeaver.org**. **Methods Mol Biol.**; (2017). 1488:131-152.
- Jeremy J. Jay; **Cross species integration of functional genomics experiments**. **International Review of Neurobiology**; (2012) 104:1-24.
- Jeremy J. Jay, John D. Eblen, Yun Zhang, Mikael Benson, Andy D. Perkins, Arnold M. Saxton, Brynn H. Voy and Elissa J. Chesler and Michael A. Langston; **A systematic comparison of genome-scale clustering algorithms**. **BMC Bioinformatics**; (2012) Jun 25;13 Supplement 10:S7.
- Erich J. Baker, Jeremy J. Jay, Vivek M. Philip, Yun Zhang, Zuopan Li, Roumyana Kirova, Michael A. Langston and Elissa J. Chesler; **Ontological Discovery Environment: a system for integrating gene-phenotype associations**. **Genomics**; (2009) Dec;94(6):377-87.

Other Relevant GeneWeaver Citations

- Ahmadi Adl, A., & Qian, X. (2015). **Tumor stratification by a novel graph-regularized bi-clique finding algorithm**. **Computational Biology and Chemistry**, 57, 3-11. 10.1016/j.compbiolchem.2015.02.010
- Baker, E.J. (2012). **Biological databases for behavioral neurobiology. International Review of Neurobiology**, 103:19-38.
- Baker, E., Culpepper, C., Philips, C., Bubier, J., Langston, M., & Chesler, E. (2014). **Identifying common components across biological network graphs using a bipartite data model.- BMC Proceedings**; Oct 13; 8.
- Baker E., Bubier J.A., Reynolds T., Langston M.A., & Chesler E.J. (2016). **GeneWeaver: data driven alignment of cross-species genomics in biology and disease**. **Nucleic Acids Res.**, 44(D1):D555-9.
- Bettembourg, C., Diot, C., & Dameron, O. (2014). **Semantic particularity measure for functional characterization of gene sets using gene ontology**. **PLoS ONE**, 9:e86525.
- Bhandari, P., Hill, J.S., Farris ,S.P., Costin, B., Martin, I., Chan, C-L., Alaimo, J.T., Bettinger, J.C., Davies, A.G., Miles, M.F., & Grotewiel, M. (2012). **Chloride intracellular channels modulate**

acute ethanol behaviors in Drosophila, Caenorhabditis elegans and mice. **Genes, Brain, and Behavior**; Jun; 11(4):387-97.

- Bhattacharyya, T., Gregorova, S., Mihola, O., Anger, M., Sebestova, J., Denny, P., Simecek, P., & Forejt, J. (2013). **Mechanistic basis of infertility of mouse intersubspecific hybrids**. **PNAS**; 2013 110 (6) E468-E477.
- Bubier, J.A., & Chesler, E.J. (2012). **Accelerating discovery for complex neurological and behavioral disorders through systems genetics and integrative genomics in the laboratory mouse**. **Neurotherapeutics**; Apr; 9(2):338-48.
- Bubier, J.A., Jay, J.J., Baker, C.L., Bergeson , S.E., Ohno, H., Metten, P., Crabbe, J.C., & Chesler, E.J. (2014). **Identification of a QTL in Mus musculus for Alcohol Preference, Withdrawal, and Ap3m2 Expression Using Integrative Functional Genomics and Precision Genetics- Genetics**; Jun 11;197(4):1377-1393.
- Bubier, J.A., & Chesler, E.J. (2015). **GeneWeaver: finding consilience in heterogeneous cross-species functional genomics data.** -*' Mammalian Genome*'; Oct 26.
- Cesar, A.S., Regitano, L.C., Mourão, G.B., Tullio, R.R., Lanna, D.P., Nassu, R.T., Mudado, M.A., Oliveira, P.S., do Nascimento, M.L., Chaves, A.S., Alencar, M.M., Sonstegard, T.S., Garrick, D.J., Reecy, J.M., & Coutinho, L.L. (2014). **Genome-wide association study for intramuscular fat deposition and composition in Nellore cattle.- BMC Genetics**; Mar 25; 15:39.
- Chan, R.F. (2013). Genetic analysis of ethanol sensitivity and tolerance in drosophila. **Master's Thesis**.
- Chen, H.C., Zou, W., Lu, T.P., & Chen, J.J. (2014). **A composite model for subgroup identification and prediction via bicluster analysis**. **PLoS ONE** , 9 (e111318). 10.1371/journal.pone.0111318
- Chesler, E.J., Plitt, A., Fisher, D., Hurd, B., Lederle, L., Bubier, J.A., Kiselycznyk, C., & Holmes, A. (2012). **Quantitative trait loci for sensitivity to ethanol intoxication in a C57BL/6J×129S1/SvImJ inbred mouse cross**. **Mammalian Genome**; Jun;23(5-6):305-21.
- Chesler, E.J., Logan, R.W. (2013). **Opportunities for bioinformatics in the classification of behavior and psychiatric disorders- International Review of Neurobiology**; 104:183-211
- Devera, S.M., Costina, B.N., Xub, R., El-Hagea, N., Balinang, J., Samoshkin, A., … Hausera, K.F. (2014). **Differential expression of the alternatively spliced OPRM1 isoform Î¼ -opioid receptor-1K in HIV-infected individuals**'. **AIDS**, 28, 19-30. 10.1097/QAD.0000000000000113
- Eblen, J.D., Phillips, C.A., Rogers, G.L., & Langston, M.A. (2012). **The maximum clique enumeration problem: algorithms, applications, and implementations.- BMC Bioinformatics**; Jun 25;13 Suppl 10:S5.
- Farris, S.P. (2016).**Moving toward understanding the proteome involved in substance abuse**. **Biological Psychiatry**, 79 , 422-424. 10.1016/j.biopsych.2016.01.003
- Groshaus, M., & Montero, L. (2016). **Tight lower bounds on the number of bicliques in false-twin-free graphs**. **Theoretical Computer Science**, 636, 77-84. 10.1016/j.tcs.2016.05.027
- Grubb, S.C., Bult, C.J., & Bogue, M.A. (2014). **Mouse Phenome Database.Nucleic Acids Research**; 42:D825-D834.
- Ha, T.J., Swanson, D.J., Kirova, R., Yeung, J., Choi, K., Tong, Y., Chesler, E.J., & Goldowitz, D. (2012). **Genome-wide microarray comparison reveals downstream genes of Pax6 in the developing mouse cerebellum.- European Journal of Neuroscience**; Oct 36(7); 2888-98.
- Haendel, M.A., & Chesler, E.J. (2012). **Lost and found in behavioral informatics.- International Review of Neurobiology**; 103:1-18.

- Hait, N.C., Wise, L.E., Allegood, J.C., O'Brien, M., Avni, D., Reeves, T.M., Knapp, P.E., Lu, J., Luo, C., Miles, M.F., Milstien, S., Lichtman, A.H., & Spiegel, S. (2014). **Active, phosphorylated fingolimod inhibits histone deacetylases and facilitates fear extinction memory**.*' Nature Neuroscience.*'
- Han, B.W., & Chen, Y.Q. (2013). **Potential pathological and functional links between long noncoding RNAs and hematopoiesis**.*' Science Signaling*'; 6.
- Jay, J.J., & Chesler, E.J. (2014). **Performing integrative functional genomics analysis in GeneWeaver.org**.*' Methods in Molecular Biology*'; pp 13-29.
- Jegga, A.G. (2014). **Candidate gene discovery and prioritization in rare diseases**.**Methods in Molecular Biology**; 1168:295-312.
- Kang, H., Choi, I., Cho, S., Ryu, D., Lee, S., & Kim, W. (2014). **GsGator: An integrated web platform for cross-species gene set analysis**. **BMC Bioinformatics**, 15 (13). 10.1186/1471-2105-15-13
- Kershenbaum, A., Cutillo, A., Darabos, C., Murray, K., Schiaffino, R., & Moore, J.H. (2016). **Bicliques in graphs with correlated edges: From artificial to biological networks**. In **Lecture Notes in Computer Science** (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (138-155). 10.1007/978-3-319-31204-0_10
- Logan, R.W., Robledo, R.F., Recla, J.M., Philip, V.M., Bubier, J.A., Jay, J.J., Harwood, C., Wilcox, T., Gatti, D.M., Bult, C.J., Churchill, G.A., & Chesler, E.J. (2013). **High-precision genetic mapping of behavioral traits in the diversity outbred mouse population- Genes, Brain and Behavior**; Jun; 12(4):424-37.
- Louati, A., Haddad, J.E., & Pinson, S. (2015). **A multi-agent approach for trust-based service discovery and selection in social networks**. **Scalable Computing**, 16, 381-402. doi: 10.12694/scpe.v16i4.1129
- McWhite, C.D., Liebeskind, B.J., & Marcotte, E.M. (2015). **Applications of comparative evolution to human disease genetics**. **Current Opinion in Genetics and Development**, 35, 16-24. doi: 10.1016/j.gde.2015.08.004
- Meehan, T.F., Carr, C.J., Jay, J.J., Bult, C.J., Chesler, E.J., & Blake, J.C. (2011). **Autism candidate genes via mouse phenomics**. **Journal of Biomedical Informatics**; Dec; 44 Supplement 1:S5-11.
- Mooney, M., & McWeeney, S. (2014). **Data integration and reproducibility for high-throughput transcriptomics**. In **International Review of Neurobiology** (55-71). doi: 10.1016/B978-0-12-801105-8.00003-5
- Nestor, C.E., Barrenäs, F., Wang, H., Lentini, A., Zhang, H., Bruhn, S., Jörnsten, R., Langston, M.A., Rogers, G., Gustafsson, M., & Benson, M. (2014). **DNA methylation changes separate allergic patients from healthy controls and may reflect altered CD4+ T-cell population structure- PLoS Genetics**; Jan;10(1):e1004059.
- O'Brien, M.A., Costin, B.N., & Miles, M.F. (2012). **Using genome-wide expression profiling to define gene networks relevant to the study of complex traits: from RNA integrity to network topology.- International Review of Neurobiology**;104:91-133.
- Oren, Y., Nachshon, A., Frishberg, A., Wilentzik, R., & Gat-Viks, I. (2015). **Linking traits based on their shared molecular mechanisms**. **eLife** , 2015 (e04346). 10.7554/eLife.04346
- Padula, A.E., Griffin, W.C. 3rd, Lopez, M.F., Nimitvilai, S., Cannady, R., McGuier, N.S., Chesler, E.J., Miles, M.F., Williams, R.W., Randall, P.K., Woodward, J.J., Becker, H.C., & Mulholland, P.J. (2015). **KCNN Genes that Encode Small-Conductance Ca2+-Activated K+ Channels Influence Alcohol and Drug Addiction.- Neuropsychopharmacology**; Feb 9.

- Recla, J.M., Robledo, R.F., Gatti, D.M., Bult, C.J., Churchill, G.A., & Chesler, E.J. (2014). **Precise genetic mapping and integrative bioinformatics in Diversity Outbred mice reveals Hydin as a novel pain gene**.*' Mammalian Genome*'; 25:211-222.
- Rojas-Muñoz, A., Maurya, M.R., Lo, F., & Willems, E. (2014). **Integrating omics into the cardiac differentiation of human pluripotent stem cells Wiley Interdisciplinary Reviews: Systems Biology and Medicine**, 6, 247-264. doi: 10.1002/wsbm.1268
- Roth, A., Kyzar, E.J., Cachet, J., Stewart, A.M., Green, J., Gaikwad, S., O'Leary, T.P., Tabakoff, B., Brown, R.E., & Kalueff, A.V. (2013). **Potential translational targets revealed by linking mouse grooming behavioral phenotypes to gene expression using public databases**. **Progress in Neuro-Psychopharmacology & Biological Psychiatry**; 40:313-325.
- Saccone, S.F. (2012). **Applying in silico integrative genomics to genetic studies of human disease**. In **International Review of Neurobiology** (133-156). doi: 10.1016/B978-0-12-388408-4.00007-1
- Smith, C.L., & Eppig, J.T. (2012). **The Mammalian Phenotype Ontology as a unifying standard for experimental and high-throughput phenotyping data**. **Mammalian Genome**, 23, 653-668. doi: 10.1007/s00335-012-9421-3
- Talishinsky, A., & Rosen, G.D. (2012). **Systems Genetics of the Lateral Septal Nucleus in Mouse: Heritability, Genetic Control, and Covariation with Behavioral and Morphological Traits**. **PLoS ONE**; 7(8): e44236.
- Tsiliki, G., Tsaramirsis, K., & Kossida, S. (2014). **AmalgamScope: merging annotations data across the human genome.- BioMed Research International.**; Epub 2014 May 20.
- Van Steen, K., & Malats, N. (2014). **Perspectives on data integration in human complex disease analysis**. In **Big Data Analytics in Bioinformatics and Healthcare** (284-322). 10.4018/978-1-4666-6611-5.ch013
- Williams, R.W. (2011). **Genetic and genomic web resources for research on alcohol use and abuse**. **Alcohol Research and Health**, 34 , 378-380.
- Williams, R.W., & Mulligan, M.K. (2012). **Genetic and molecular network analysis of behavior**. **International Review of Neurobiology**; 104:135-57.
- Wiltshire, T., Ervin, R.B., Duan, H., Bogue, M.A., Zamboni, W.C., Cook, S., Chung, W., Zou, F., & Tarantino, L.M. (2015). **Initial locomotor sensitivity to cocaine varies widely among inbred mouse strains. - Genes, Brain and Behavior**, 14, 271-280. 10.1111/gbb.12209
- Wittkop, T., TerAvest, E., Evani, U.S., Fleisch, K.M., Berman, A.E., Powell, C., Shah, N.H., & Mooney, S.D. (2013). **STOP using just GO: A multi-ontology hypothesis generation tool for high throughput experimentation**. **BMC Bioinformatics**, 14.
- Zhang, Y., Phillips, C.A., Rogers, G.L., Baker, E.J., Chesler, E.J., & Langston, M.A. (2014). **On finding bicliques in bipartite graphs: a novel algorithm and its application to the integration of diverse biological data types. BMC Bioinformatics**, 15.

Related Resources and Publications GeneNetwork

QR Code

Top ↑

Figure 22: QR Code to Main Page

Policies

Usage Policy and Disclaimer

Data and web site providers make no guarantees or warranties as to the accuracy or completeness of results obtained from accessing and using information from GeneWeaver. We will not be liable to any user or anyone else for any inaccuracy, error or omission, regardless of cause, in the data contained in the GeneWeaver databases or any resulting damages. In addition, the data providers do not warrant that the databases will meet your requirements, be uninterrupted or error-free. Data providers expressly exclude and disclaim all expressed and implied warranties of merchantability and fitness for a particular purpose. Data providers shall not be responsible for any damage or loss of any kind arising out of or related to your use of the databases, including without limitation data loss or corruption, regardless of whether such liability is based in tort, contract or otherwise.

To report any errors found in the GeneWeaver database, please notify the appropriate person listed on our Contacts page.

Data Sharing Policy

Data sharing in GeneWeaver is as broad or restrictive as the investigator allows. When uploading data, it can be made private, public or accessible only to selected groups. Access restrictions can be changed at any time. All group members are also visible on the account setup page. The only people with access to your data are those who you personally allow, or those who your group administrator(s) allow. GeneWeaver will make no use of the data outside of normal metrics used to optimize algorithm or database efficiency, or in other internal use solely for the development of GeneWeaver, see Privacy Policy for more.

In addition, our directives to share data stem from the NIH Data Sharing Policy that states:

> Data should be made as widely and freely available as possible while safeguarding the privacy of participants, and protecting confidential and proprietary data.

Privacy Policy

- In order to integrate data from many users, while protecting private data, we must store data on your server. User contact information is collected for optional display with your gene sets to foster collaborative research.
- Entering user information is not mandatory.
- User information will not be sold or otherwise distributed.
- GeneWeaver records some information about how the site is used such as the IP address of machines accessing data sets. This information is used to monitor our system performance, to prevent abuse of the system, and to guide further development of the GeneWeaver. This information is stored on the server in files that are accessible to members of the development group. Specific information will not be released.
- When you visit GeneWeaver, your use of the site is recorded in two ways. First, your use is logged by the Web server in standard log files. The IP address of your machine, the date and time, and the name of the page you visit are recorded. Second, for each request from the SQL database, the GeneWeaver records your IP number, the time, and the data set from which you request information. This information is collected for statistical purposes. Our system uses a software program (Analog) to create summary statistics that we find helpful in assessing patterns of data use, in measuring system performance and in detecting problems. This information is used to provide you with better internet service.
- GeneWeaver also may request permission to place a so-called 'cookie' text file on your system to allow you to retain information on your set-up preferences.
- For site security purposes and to ensure that this server remains available to users, this computer system employs programs that monitor network traffic to identify unauthorized attempts to upload or change information, and to detect unusually high numbers of requests from single IP addresses. By accessing this site, you expressly consent to usage monitoring of this site for unauthorized or unusual activities. Unauthorized attempts to upload information and change information are prohibited.
- In some cases, personal identifier information such as name or e-mail is requested or required. This information may be posted for public access along with the submitted comments and messages that it accompanies. In all cases, participation is strictly voluntary and no other use is made of the information. User data can be labeled as private, group or public on submission. Permissions can be changed. Gene sets that are not marked as public will not be included in global analyses of the database contents.

Top ↑

Contacts

For questions about…

GeneWeaver vision and future direction:

- Project Lead **Elissa J. Chesler, Ph.D.**, Professor, The Jackson Laboratory, Bar Harbor, Maine (207)288-6000x6453

Software questions and reporting bugs:

- **Erich J. Baker, Ph.D.**, Professor, Baylor University School of Engineering and Computer Science, Waco, Texas (254)710-7304

Database design:

- **Erich J. Baker, Ph.D.**, Professor, Baylor University School of Engineering and Computer Science, Waco, Texas (254)710-7304

Graph algorithms:

- **Michael A. Langston, Ph.D.**, Professor, University of Tennessee College of Engineering, Department of Electrical Engineering and Computer Science, Knoxville, Tennessee, and Collaborating Scientist, Oak Ridge National Laboratory, Oak Ridge, Tennessee (865)974-3534

Data curation

- **Jason Bubier**, GeneWeaver Data Curator/Associate Research Scientist, Chesler Lab, The Jackson Laboratory, Bar Harbor, Maine (207)288-6000x6704

Top ↑

Acknowledgements

> "Making sense of genomics is risky, But with database builders so frisky Gene expression in brains
> May one day explain A mouse's obsession with whiskey"

-Poet Laureate of the Neuroscience Program, University of Illinois at Urbana-Champaign, November 27, 2006

Support

GeneWeaver / The Ontological Discovery Environment was initiated as a project of the NIAAA Integrative Neuroscience Initiative on Alcoholism (U01AA13499, U24AA13513), and is currently supported by R01 AA018776, jointly funded by NIDA and NIAAA. Additional support comes from the Center for Precision Genetics, NIH U54 OD020351.

When using GeneWeaver, please cite:

> 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**. **Nucleic Acids Research**; (2012) 40(D1): D1067-D1076

Current members of the GeneWeaver team:

- **Elissa J. Chesler**, Fearless Leader, Jive Translator
- **Erich J. Baker**, Database Design Leader
- **Michael A. Langston**, Graph Algorithms Team Leader
- **Jason A. Bubier**, Data Curator
- **Charles Phillips**, Graph Algorithms
- **Timothy Reynolds**, Lead Developer
- **Computational Sciences Group, The Jackson Laboratory, Development and QA team**

Former team members:

- **Roumyana Kirova**, Statistics and Data Mining
- **Vivek Philip**, Visualization
- **Zuopan Li**, Web Programmer
- **Yun Zhang**, Graph Algorithms
- **Michael Marion**, Data Curator
- **Jeremy Jay**, Lead Developer

Top ↑

These pages are maintained by the GeneWeaver team and the **Chesler Lab** at **The Jackson Laboratory** in Bar Harbor, Maine.