AquaMine

Release 1.1

Elsik Lab

Jul 26, 2023

TABLE OF CONTENTS

1 Overview of AquaMine **3**

2 Navigation and Searching in AquaMine **5**
- 2.1 Quick Search . . . 5
- 2.2 Templates . . . 7
  - 2.2.1 Generate query code . . . 9
  - 2.2.2 Download results . . . 9
  - 2.2.3 Customize output and manage columns . . . 10
  - 2.2.4 Optional filters . . . 11
- 2.3 QueryBuilder . . . 11
  - 2.3.1 Model browser . . . 13
  - 2.3.2 QueryBuilder Examples . . . 13
    - 2.3.2.1 Example 1: Querying for protein coding genes . . . 13
    - 2.3.2.2 Example 2: Querying for protein coding genes on a particuler chromosome . . . 15
    - 2.3.2.3 Example 3: Querying for Protein Coding genes on a particular chromosome and their exons . . . 17

3 Report Pages **21**
- 3.1 Summary . . . 22
- 3.2 Transcripts . . . 22
- 3.3 Proteins . . . 23
- 3.4 Function . . . 23
- 3.5 Homology . . . 24
- 3.6 Publications . . . 24
- 3.7 Other . . . 24

4 Genomic Regions Search **27**

5 Lists **31**
- 5.1 Creating Lists . . . 31
- 5.2 Saving Lists . . . 33

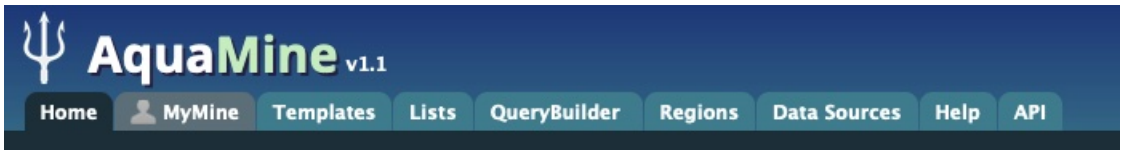
6 MyMine **37**

7 API **39**

8 Data Sources **41**

9 How to cite **43**

AquaMine is a data mining resource that integrates genomic data of importance to U.S. aquaculture. It combines data generated by the aquaculture research community with data from external public databases. Many customized bioinformatics tools are available that researchers with or without bioinformatic programming skills can use to create their own crafted datasets for use in their research projects. AquaMine is powered by InterMine, an open-source data warehousing system that provides access to a wide variety of datasets.



Main site: https://aquamine.rnet.missouri.edu/aquamine/begin.do

Link to the datasets used in AquaMine can be found here: https://aquamine.rnet.missouri.edu/aquamine/dataCategories.do

AquaMine.org is based upon work supported by the National Science Foundation under Award Number 0054449. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. AquaMine is developed and hosted at the University of Missouri. If you have comments or if you wish to report a problem, please contact the Database Administrator.

CHAPTER

ONE

OVERVIEW OF AQUAMINE

This section provides a brief overview of the layout for AquaMine.

The navigation panel highlights different functionalities of AquaMine.

Home - The home page for AquaMine

MyMine - The MyMine serves as a portal for account management. When logged in to AquaMine Users can access their saved templates, most recent queries and saved lists.

Templates - List of templates that users can select from based on the nature of their query. Each template is a predefined query with a simple form containing a description of what input is expected and the type of output that will be generated.

Lists - Allows users to upload lists of genes on which they can perform enrichment analyses and export the results. Users that log in to AquaMine can save their lists for future use.

QueryBuilder - A flexible interface that allows users to create their own custom query template while browing the AquaMine data models. Queries can be exported in a variety of formats to share with other users.

Regions - The Genomic Region Search tool where users can enter a series of genomic coordinates, specify flanking regions and fetch all features that fall within the given interval. The result can be exported or saved as a list for further analyses.

Data Sources - Provides a summary of all the data loaded into AquaMine including their sources, associated publications and links to source sites.

Help - Links to the AquaMine help docs and tutorials

API - Describes the InterMine API that allows users to programmatically access AquaMine.

CHAPTER

TWO

NAVIGATION AND SEARCHING IN AQUAMINE

There are several ways for users to query AquaMine.

2.1 Quick Search

Quick Search allows users to search keywords from any of the AquaMine datasets. There is a Quick Search box on the AquaMine home page or a smaller search box in the upper right corner of all pages.

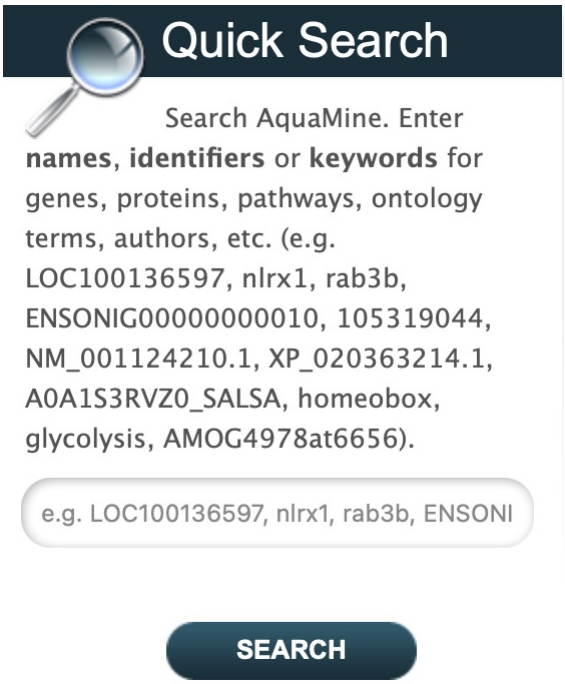


Fig. 1: Quick Search from home page

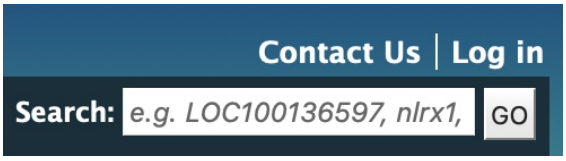


Fig. 2: Quick Search from any page

Quick Searches can be conducted with a number of identifiers including gene names, transcripts, pathways, gene identifiers or organisms. The wildcard character * can be used to retrieve all results that match a particular search query.

For an example, we will use the lysozyme gene *COX1* as an example. Enter COX1 in the search box and click **Search**. The results page is tabulated and displays a summary about your query, as shown below.

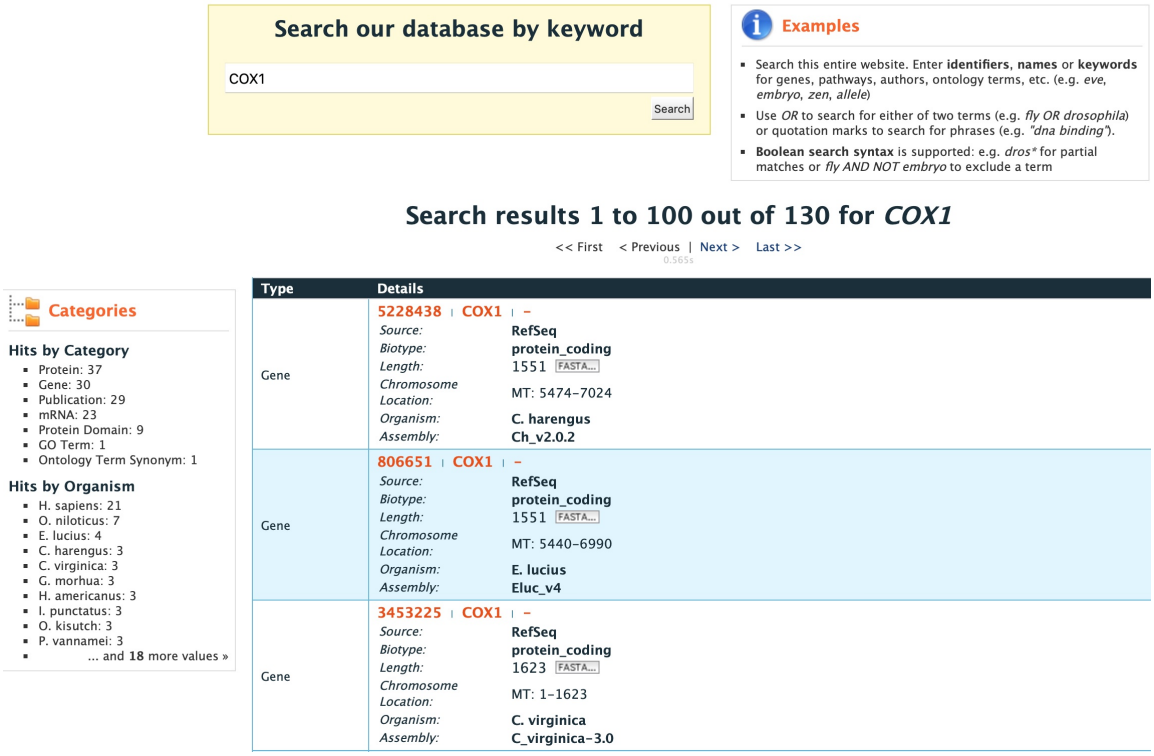


Fig. 3: Results table for Quick Search with the gene *COX1*

Hits are summarized in the box to the left of the results tables and can be filtered based on **Category** and **Organism**. Clicking on any of them will filter by the selected category. Note that for results with sequence data available, the sequence can be downloaded in FASTA format by clicking on the FASTA box within the hit Details box. The score column in the result table indicates the similarity of your query to each of the hits. The results page can also be converted to a list (and saved if users are logged in). To enable this feature click on **Gene** in the **Hits by Category** then click on *O. niloticus* in **Hits by Organism**.

After the table has been filtered for gene and organism, checkboxes will be available for users to select genes they would like to add to their list. Once the genes are selected, click on **CREATE LIST**. See the lists section for more detail on creating and saving lists.

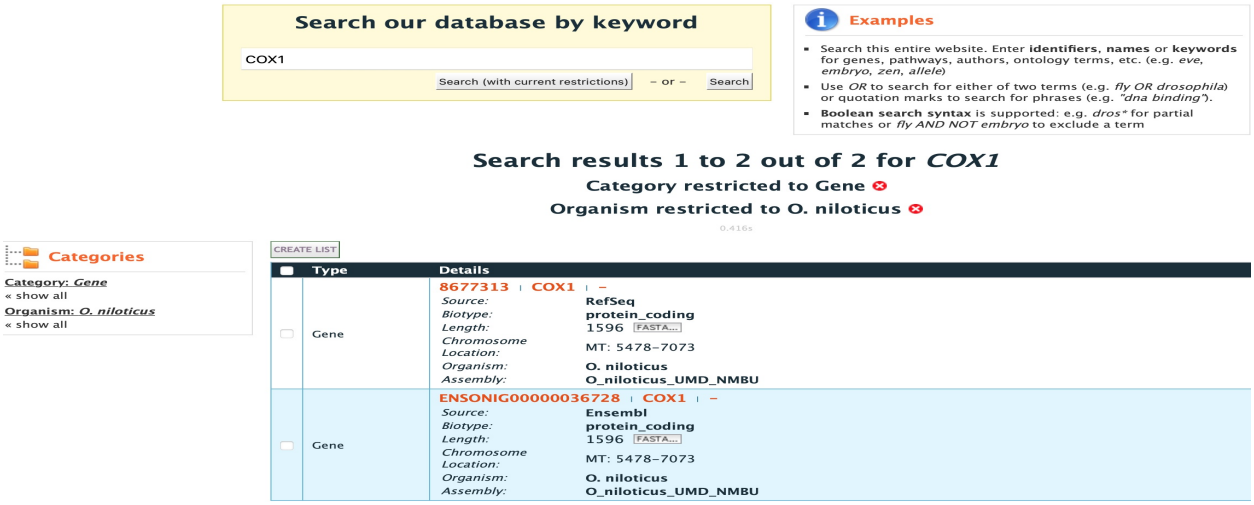


Fig. 4: Results table for Quick Search with the gene *COX1* filtered by gene then by organism ..

2.2 Templates

Templates or predefined queries are another search method within AquaMine. Popular templates are displayed on the home page, grouped by category (e.g., Genes, Proteins, Interactions) and the complete list can be seen by clicking the **Templates** menu tab.

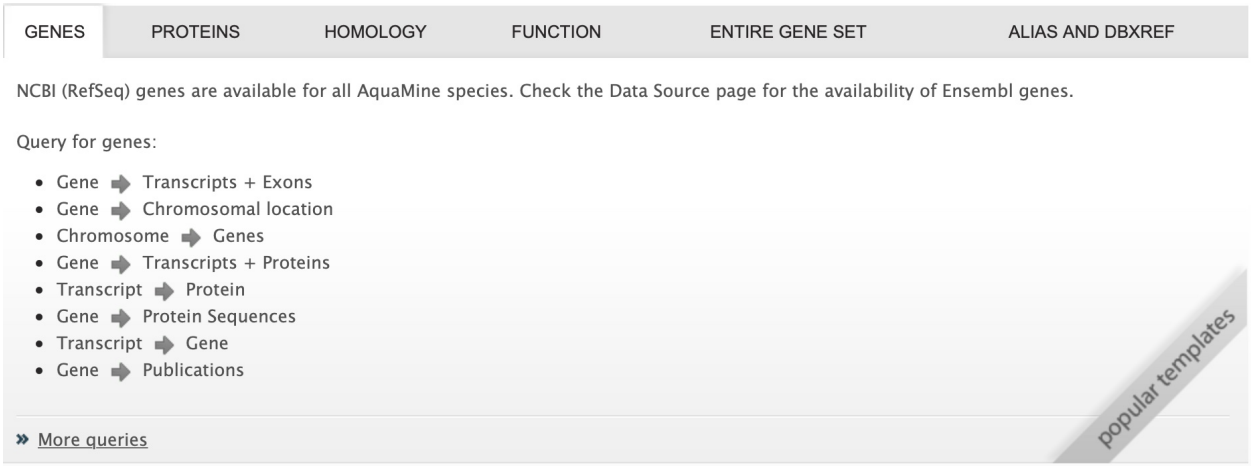


Fig. 5: Popular templates

As an example, the **Gene -> Protein Sequnces** template queries AquaMine to retrieves all homologues for a given gene. Here, we will do a search with the gene ID *110523811*.

The results page displays all of the proteins for that query gene. When logged in to AquaMine, users can save their results as a list for further analyses by clicking on the **Save as List** button above the results table then choosing columns to save in their list. See the lists section for more detail on creating and saving lists. Note the "Trail: Query" text at the upper left of the results table. Clicking on the "Query" link will bring you back to the query that generated the table to allow for edits without having to start with a new template.

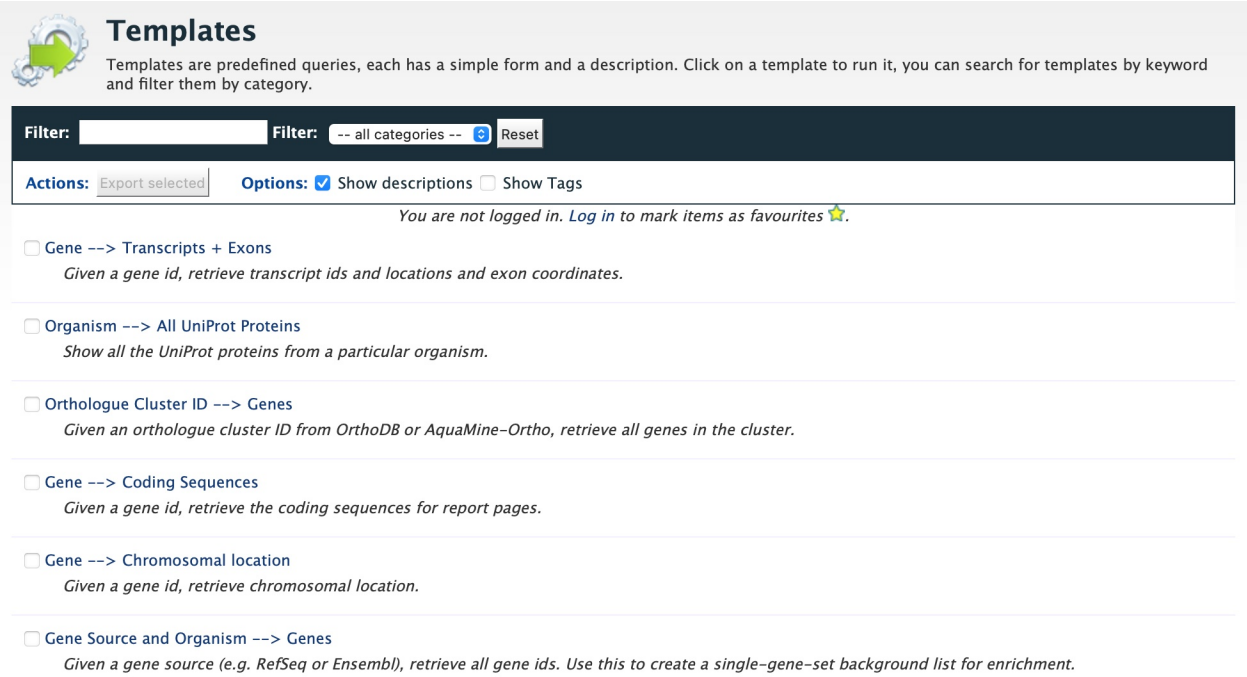
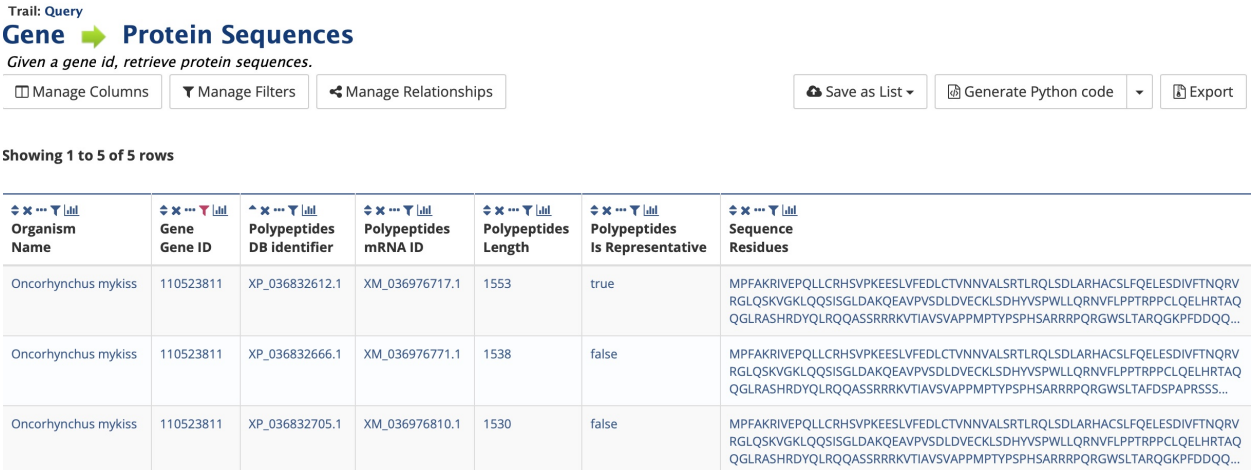


Fig. 6: Full list of templates on Templates page

Fig. 7: Example: Gene –> Protein Sequences



Example: Gene –> Protein template search results.

2.2.1 Generate query code

The code for each template query can be retrieved by clicking on the arrow next to **Generate Python Code** and choosing the desired language from the pull-down menu. The language options are Python, Perl, Java, Ruby, JavaScript, and XML.

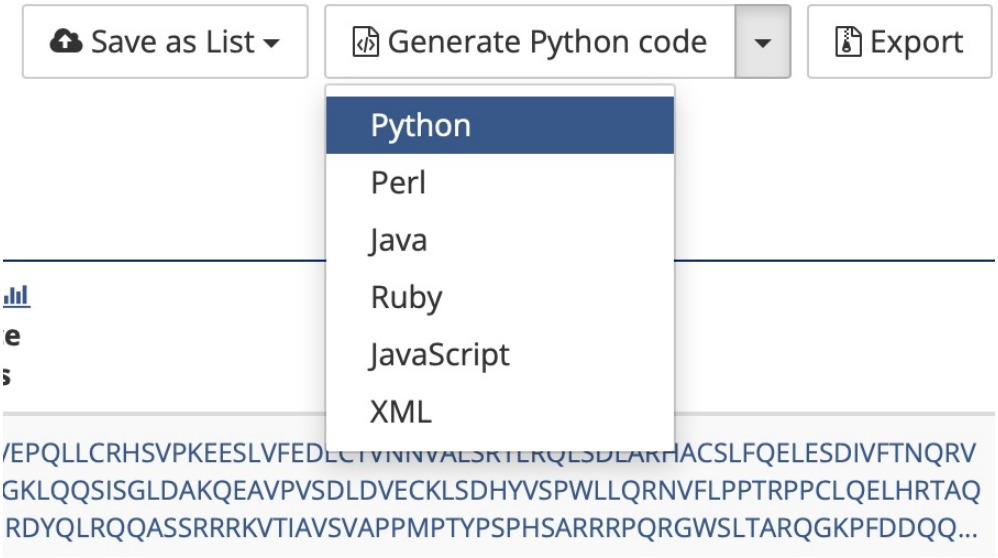


Fig. 8: Options for generating code from template query

2.2.2 Download results

The search results from a template query can be downloaded by clicking the **Export** button above the table and choosing the desired format from the pull-down menu to the right of the File name field. Available formats are tab-separated values, comma-separated values, XML, and JSON. When the results contain genomic features, they may also be downloaded in FASTA, GFF3, or BED format. Other options may be specified in the submenu to the left of the download box. By default, all rows and all columns are downloaded, but individual columns may be included or excluded by clicking on the toggles next to the column headers in the **All Columns** submenu. The number of rows and row offset are set in the **All Rows** submenu. Download the results as a compressed file by choosing GZIP or ZIP format in the **Compression** submenu (default is **No Compression**). Column headers are not added by default but may be included under the **Column Headers** submenu. Finally, the **Preview** submenu displays the first three rows of the file to be downloaded so that the desired format and options may be finalized before beginning the download. When ready, click the **Download file** button to download the results.

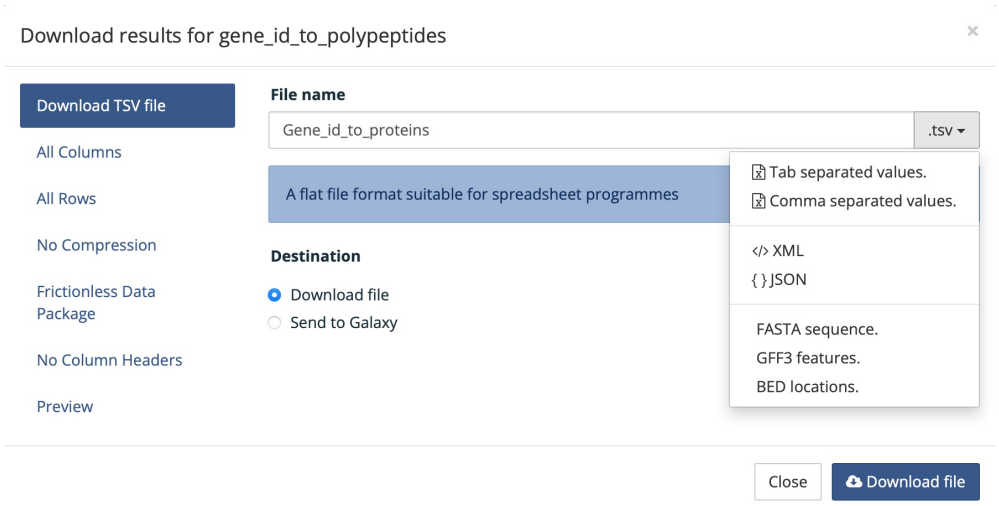


Fig. 9: Options for downloading results from template query

2.2.3 Customize output and manage columns

To customize the results table layout, click the **Manage Columns** button. This allows users to rearrange, remove or order columns. Filters can be edited by clicking the **Manage Filters** button. To specify the entity relationships within the query and change the way the results are presented in the table, click **Manage Relationships**. Clicking on the **Manage Relationships** option bring also brings up a blue information panel ("What does this do?") that provides more detailed information.

Fig. 10: Options for customizing the results of a template query.

To further manage column data, each column has a set of icons in its header. Mousing over any icon will reveal what it does. The sort icon resembles two triangles and allows you to sort a column in either direction (ascending/descending; a->z, z->a). To delete any column from your table, click on the "x" icon. Alternatively, the three dots "…" can be selected to "Toggle column visibility" and hide the column. Once hidden, the icon turns into a double arrow that can be selected to expand the column into its original form. The next icon resembling a funnel can be selected to edit or remove any currently active filters. A summary of the data within a column can be viewed by clicking on the icon that resembles a graph. The data within each summary can be selected for further filtering or downloading.

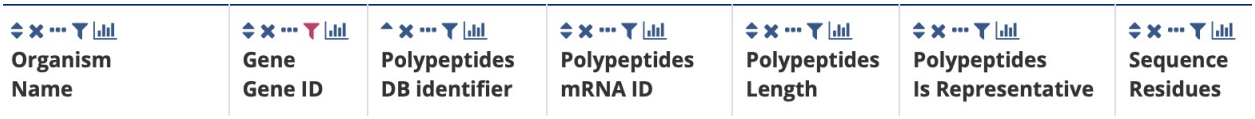


Fig. 11: Column header icons available to edit the results of a template query.

2.2.4 Optional filters

****If you are using a template with an optional filter and want to edit the template, the "Query" link at the top left of the results page will bring you back to the template that you used to create those results. Additionally, if you click on "Edit Query" in a template, the optional settings becomes non-optional and the default constraint is applied. To edit the template query, navigate back to the template by using the "Templates" tab on the home page menu bar.**

Some templates have optional filters that are disabled by default. For the Gene ID –> Homologues query, there is an optional filter to specify the Last Common Ancestor used in the query. To enable the filter, click **ON** next to the **Homologue > Last Common Ancestor** label to select an ancestor of choice.

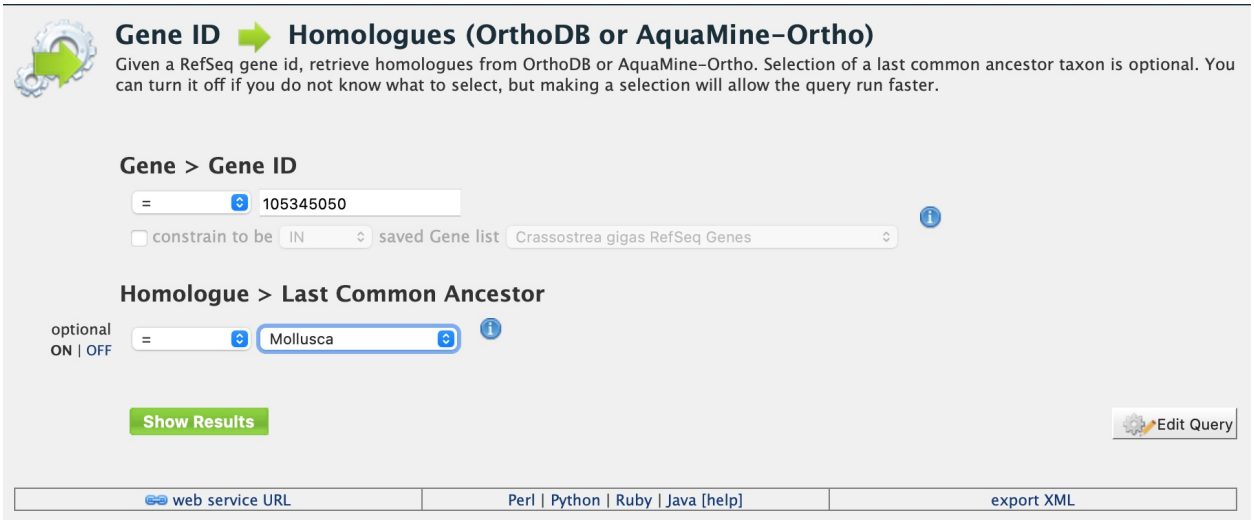
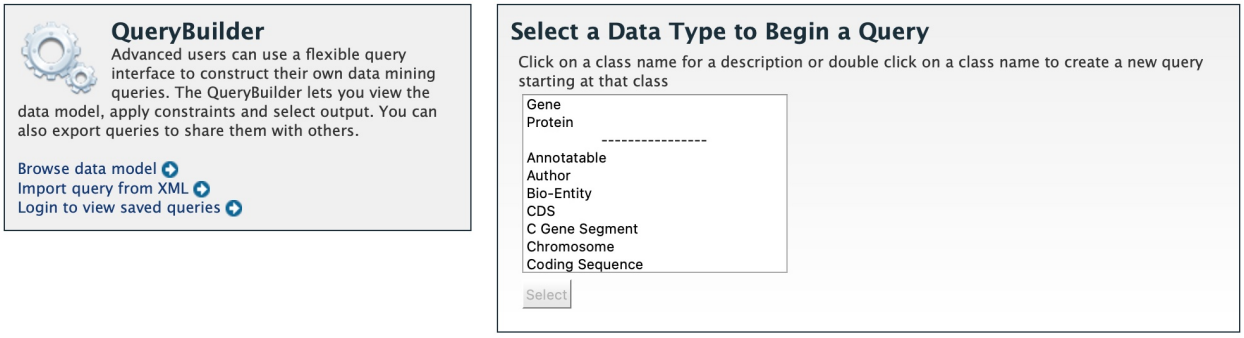


Fig. 12: Using the optional organism filter in the Gene ID –> Homologues query template.

2.3 QueryBuilder

The provided templates are suitable for many different types of searches, new queries may be built from scratch using the **QueryBuilder**. The possibilities of queries using the QueryBuilder are endless. The output may be formatted exactly as desired, and the query constraints may be chosen to perform complex search operations.

To begin, select a **Data Type**. For example, select **Gene** as a Data Type then click the **Select** button to be taken to the Model browser.



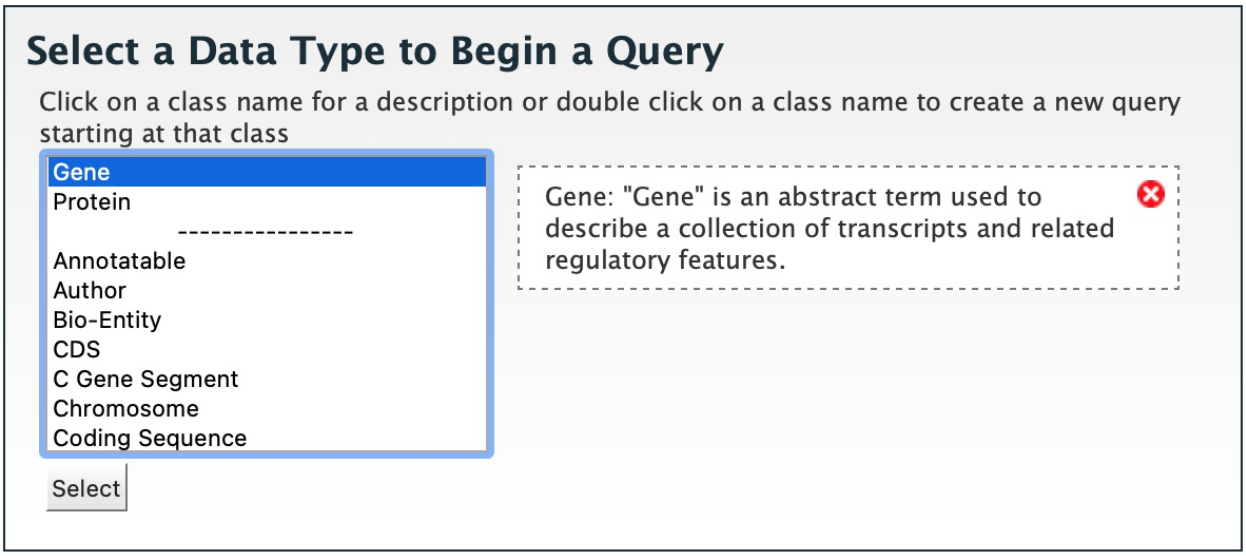


Fig. 13: Selecting data type as **Gene** in QueryBuilder.

2.3.1 Model browser

After selecting a data type, the **Model builder** appears displaying the attributes for the chosen feature class **Gene**.

First lets select `Gene` as a Data Type in the QueryBuilder. Then click on `Select`. This will take you to a Model browser where you can select the attributes for the feature class 'Gene', which you would want to be shown in your results.

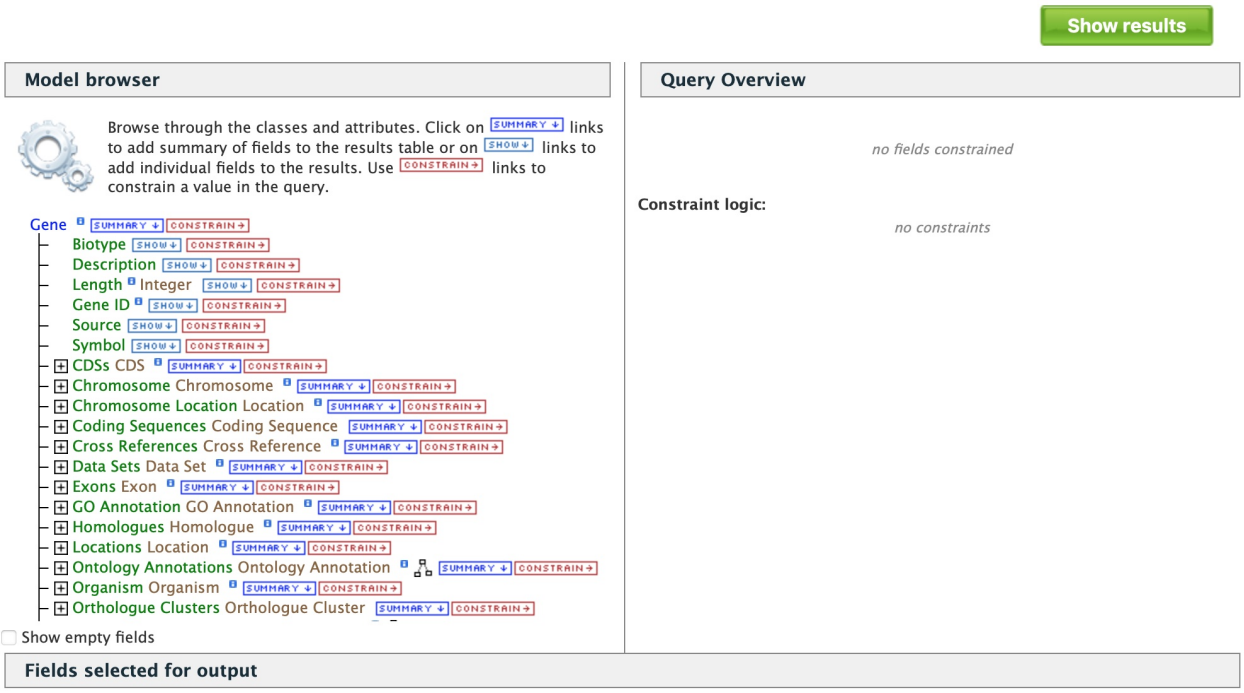


Fig. 14: Model browser with "Gene" selected as the data type.

2.3.2 QueryBuilder Examples

The following three examples provide details as to how to use the QueryBuilder using "Gene" as the selected data type.

2.3.2.1 Example 1: Querying for protein coding genes

In the Model browser, click **Show** next to **Biotype**, **Gene ID** and **Symbol**, which will add these fields to the query. Notice that these two fields appear below the data type **Gene** in the Query Overview section.

Then click **Constrain** next to **Biotype**. The first drop-down menu defaults to = (equals sign). In the second drop-down menu, select **Protein Coding**, then click the **Add to query** button. This adds a constraint to the query to search only for protein coding genes. Notice that the Query Overview section now shows "Biotype = Protein Coding". Also, two types of icons appear next to the attributes. Clicking on the red "X" icon next to an attribute will remove that field or constraint from the query. Clicking on the blue pencil icon next to a constraint brings up the constraint editing window where changes may be made to the query filters.

Lastly, click on **Show Results** above the Model Browser. The resulting table contains all protein coding genes in the database, with Gene ID, Gene Symbol and Gene Biotype as the table columns. Because the Biotypes should all be the same (protein coding), that column can be deleted by clicking the "x" above it. Alternatively, the three dots "…" can

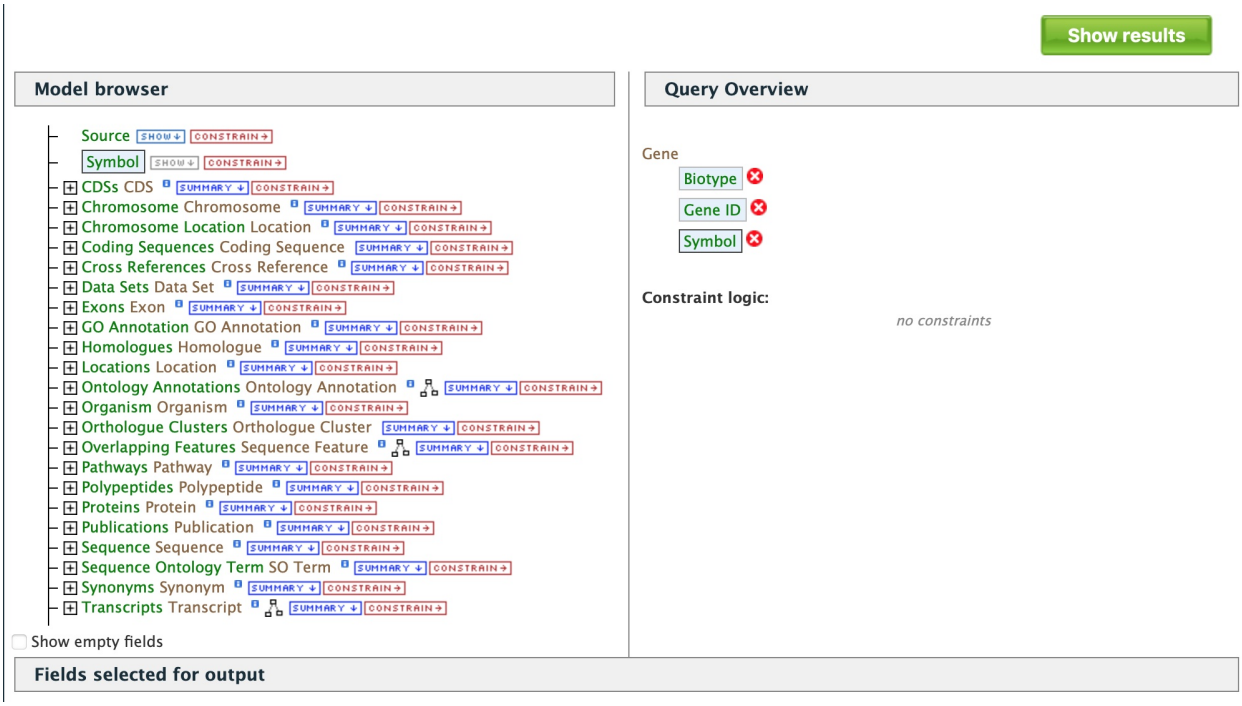


Fig. 15: Model browser with "Gene" selected as the data type.
Example 1, Step 1: Select fileds to be added to the query

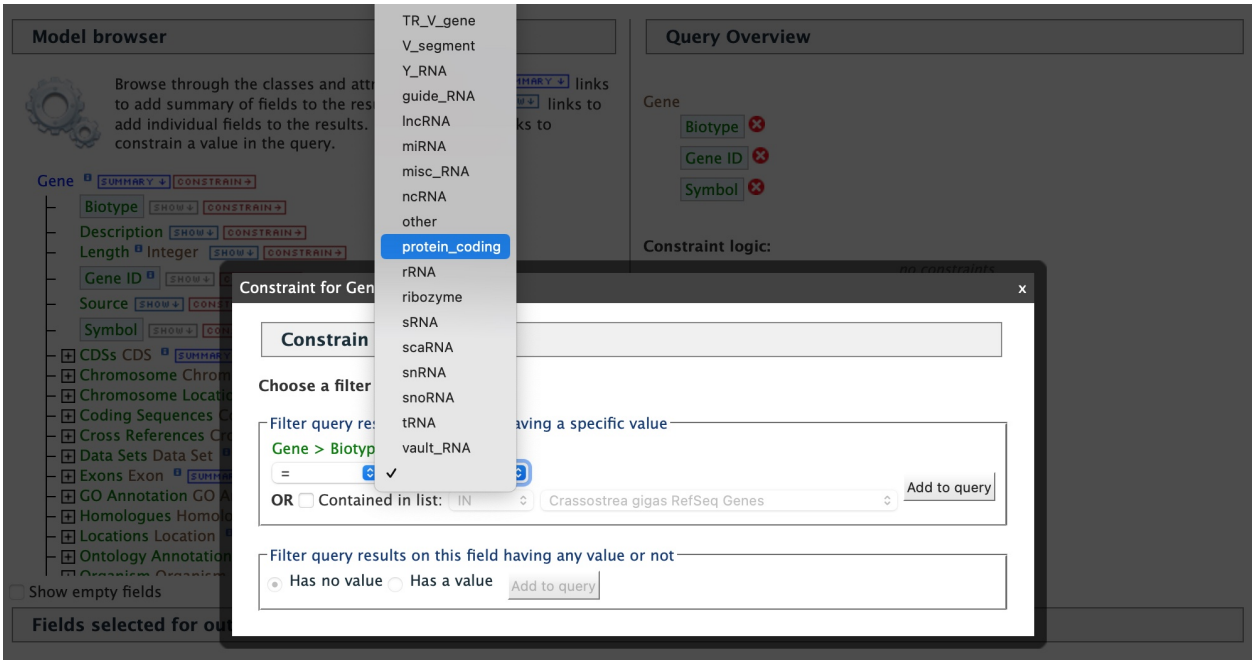


Fig. 16: Example 1, Step 2 Adding a constraint to the query on Biotype.

be selected to "Toggle column visibility" and hide the column. See the *Customize output and manage columns* section to review ways to manage column data. Clicking on "Query" in the "Trail: Query > Results" link at the upper left of the results table will bring you back to the query that generated the table to allow for edits without having to build a new query. For example, the column order can be changed by sliding the blue squares under the "Columns to Display" section at the bottom of the page.

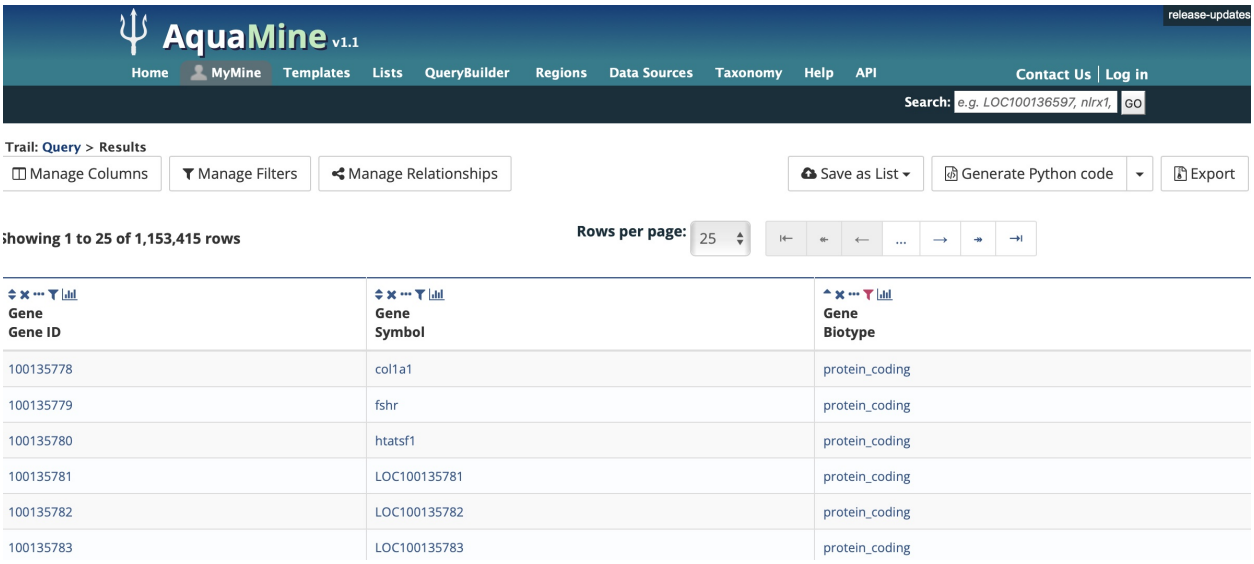


Fig. 17: Example 1, Step 3 Display query results.

2.3.2.2 Example 2: Querying for protein coding genes on a particuler chromosome

Users can customize the previously run query by adding another constraint for **Chromosome**. On every query results page, note the "Trail: Query" text at the upper left of the results table. Clicking on the "Query" link will bring you back to the query that generated the table to allow for edits without having to build a new query. In the Model browser where you began Example 1 after clicking on **Show** next to **Biotype**, **Gene ID** and **Symbol**, click on the "+" (plus sign) next to the Chromosome feature class to display its attributes.

Next click on **Constrain** next to the **Chromosome ID** attribute and in the text box of the pop-up window select "=" (equals). The chromosome ID format is specific to an organism. For this example, we will use the chromosome IDs for Exsox lucius. Enter **LG14** for chromosome number 14 then click on **Add to Query**, which adds the additional constraint to the query. We will add another constraint for the organism. Click on the **Organism** to reveal its attributes then click on **constrain** next to Name. In the resulting pop-up window, slect "=" (equals) then "Esox lucius", the Northern pike..

Click on **Show results** and the query will result in all protein-coding genes on the Chromosome with constrained ID and organism. Note that the number of results has been reduced with the addition of the constraints.

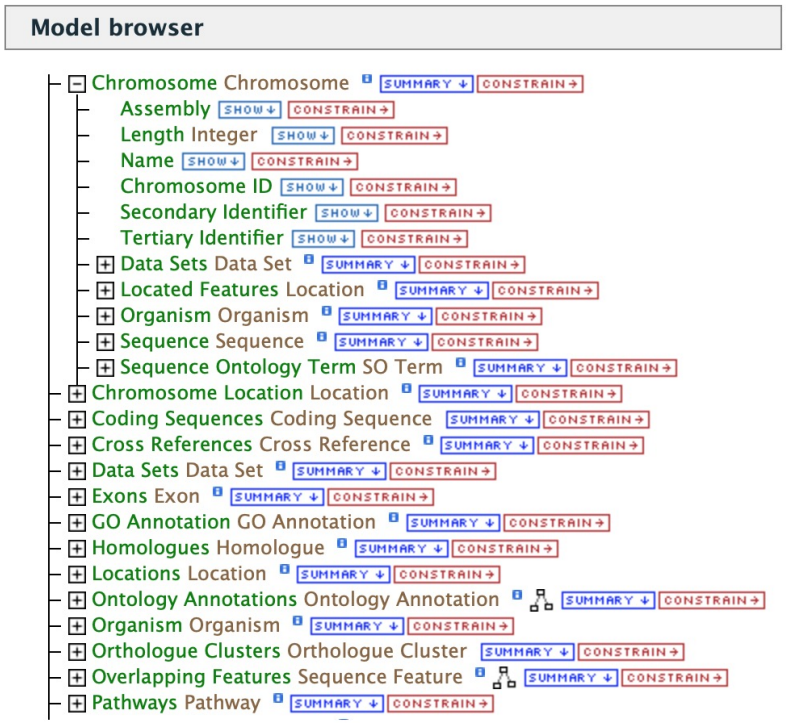


Fig. 18: Example 2, Step 1 View attributes of Chromosome feature class

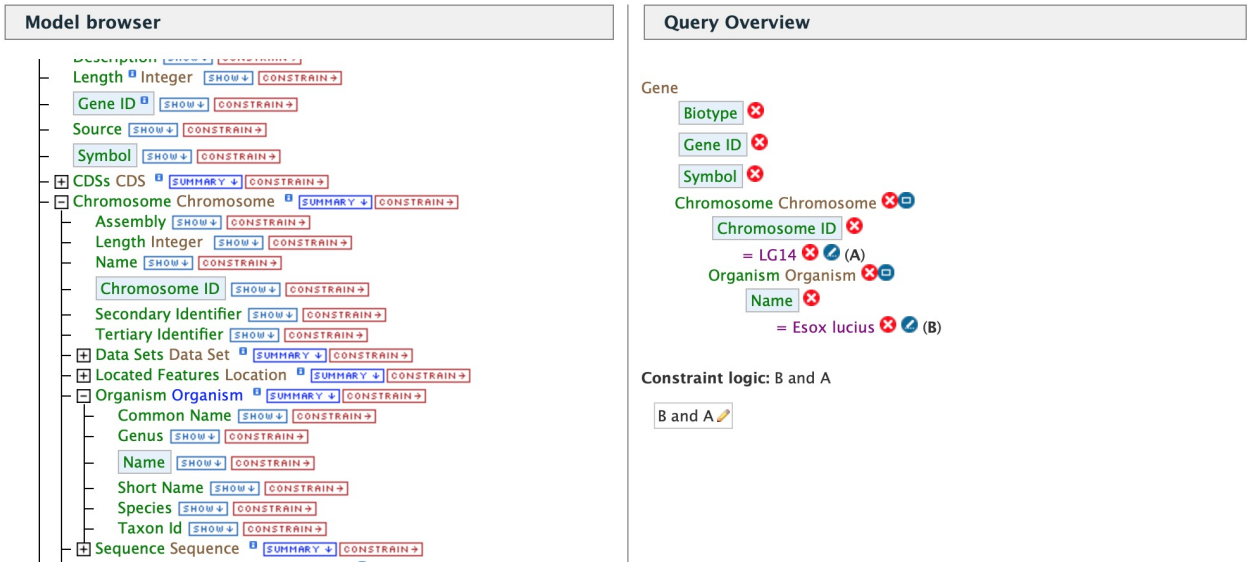
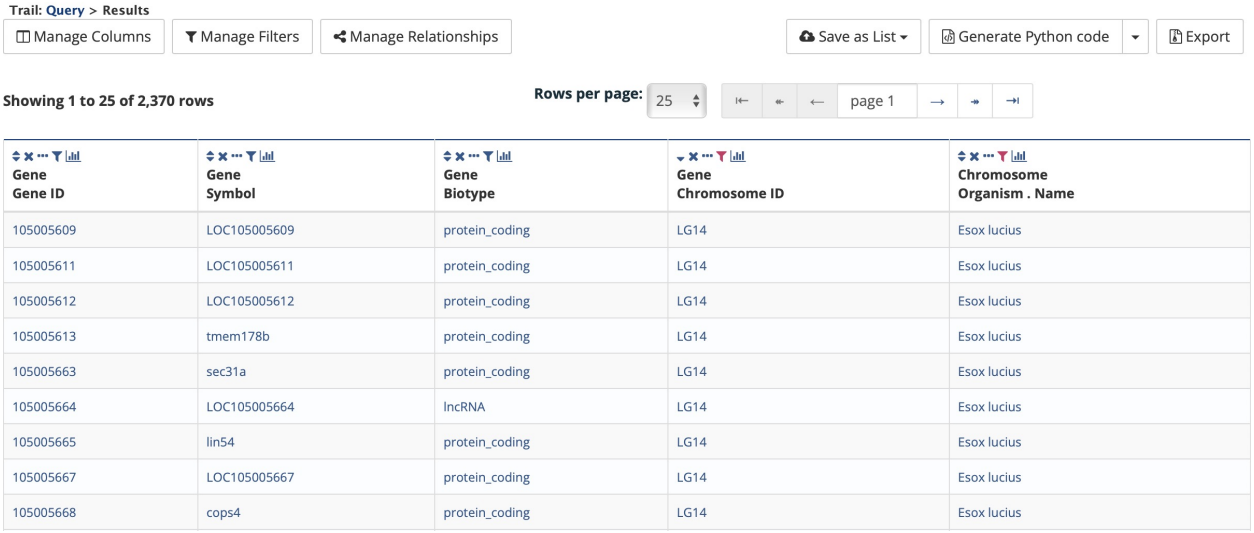


Fig. 19: Example 2, Step 2 Adding constraints to the Chromosome ID and Organism



Gene Gene ID	Gene Symbol	Gene Biotype	Gene Chromosome ID	Chromosome Organism . Name
105005609	LOC105005609	protein_coding	LG14	Esox lucius
105005611	LOC105005611	protein_coding	LG14	Esox lucius
105005612	LOC105005612	protein_coding	LG14	Esox lucius
105005613	tmem178b	protein_coding	LG14	Esox lucius
105005663	sec31a	protein_coding	LG14	Esox lucius
105005664	LOC105005664	lncRNA	LG14	Esox lucius
105005665	lin54	protein_coding	LG14	Esox lucius
105005667	LOC105005667	protein_coding	LG14	Esox lucius
105005668	cops4	protein_coding	LG14	Esox lucius

Fig. 20: Example 2, Step 3 Result table after constraining by Chromosome ID and Organism

2.3.2.3 Example 3: Querying for Protein Coding genes on a particular chromosome and their exons

This last example builds upon the previous queries to display all exons for each gene on a particular chromosome. Again, note the "Trail: Query" text at the upper left of the results table. Clicking on the "Query" link will bring you back to the query that generated the table to allow for edits without having to build a new query. Scroll down to **Exon** feature class and select the "+" (plus sign) next to Exons to display its attributes. Click **Show** next to **Length** and **Exon Identifier**.

The Query Overview shows the query in progress with the selected fields. Also notice that a third type of icon, a blue square, appears next to some attributes. Clicking on a blue square icon brings up a window where the query Join Style may be modified. When adding a constraint, you can decide whether you want to show only those results with the information (genes with exons) or all results and the constrained feature if it exists (e.g., all genes and indicate exons if they exist). Click on the blue square icon next to **Exon collection** to bring up the Switch Join Style window. The default option is to show only Genes if they have a exon (inner join). Change this to **Show all Genes and show Exons if they are present** (outer join) then click **Add to query**.

Then click "Show results" to run the new query.

The results table now lists a new column **Gene Exons**, which we added to the query. If we look at the third row in the table, it lists 14 exons. Click on the **14 exons** link to expand that entry. That column now has additional rows containing the **Exon identifier** and **Length** for each of the 14 exons.

In changing the join style to an outer join, the exons have been grouped together by gene making it easier to determine how many exons are contained in each gene. If the same query is run with the default join (outer join) of **Show only Genes if they have an Exon** , the results table adds a new row for each new exon rather than grouping exons by gene. To change the join, click on the blue square icon next to **Exon collection** to bring up the Switch Join Style window as done above..

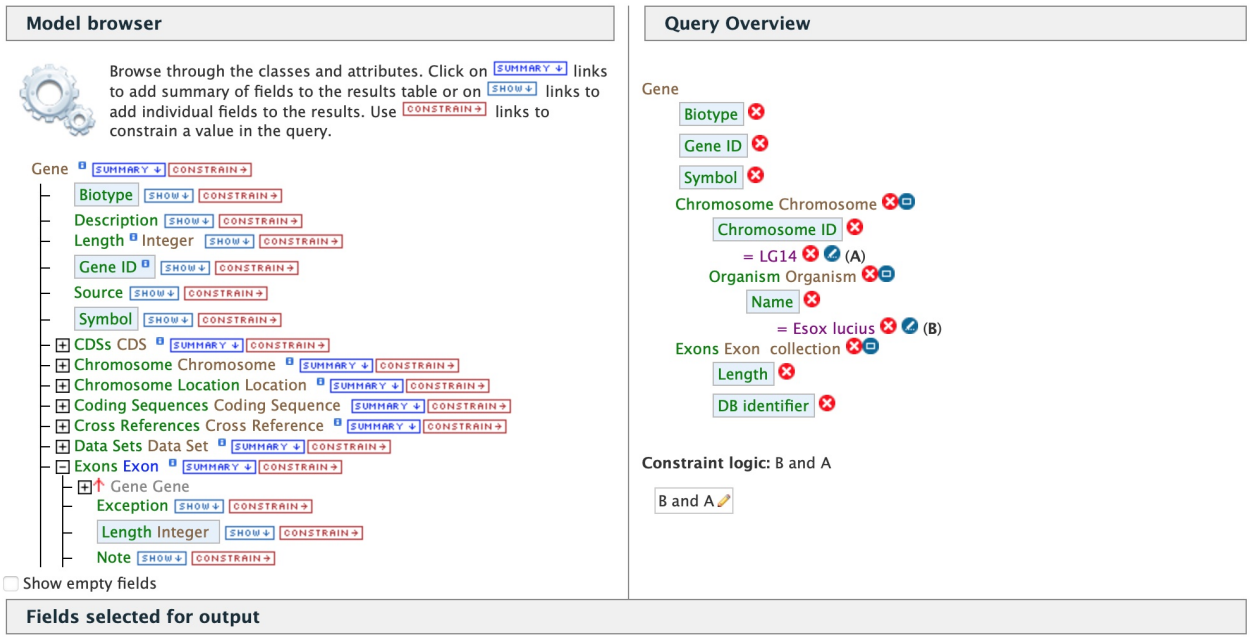


Fig. 21: Example 3, Step 1 Expand exon attributes and add fields to the query

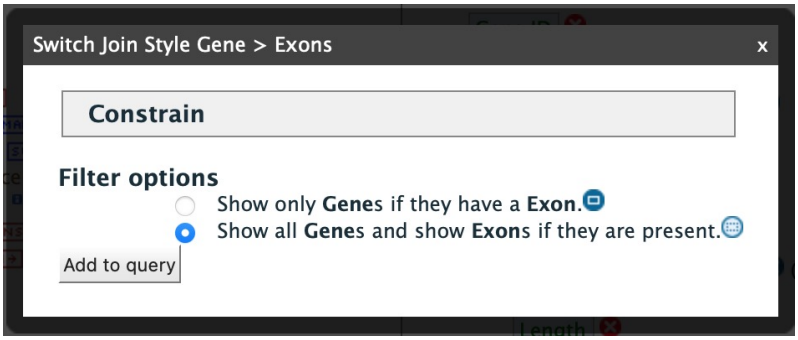
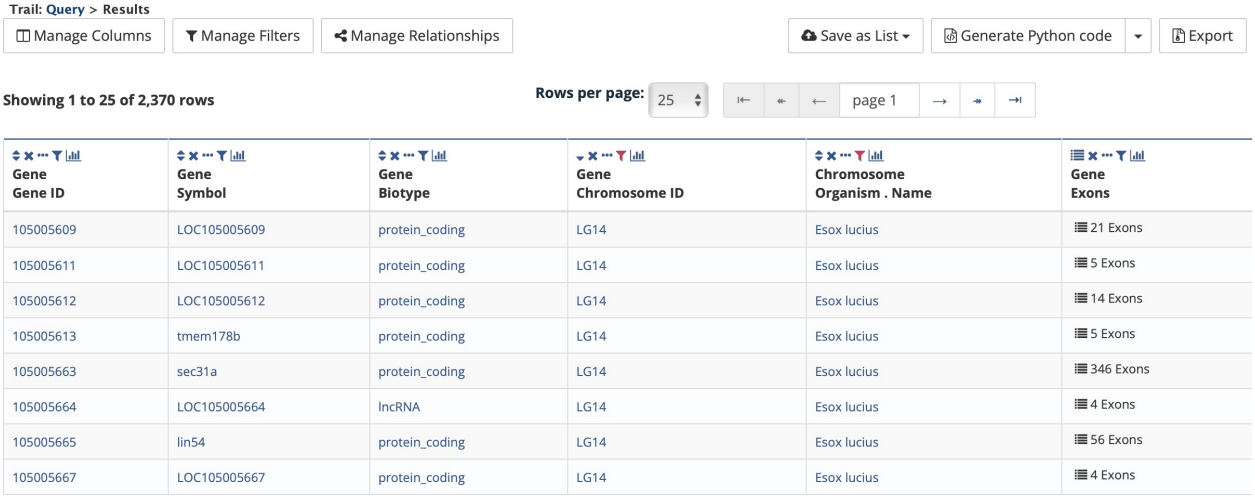


Fig. 22: Example 3, Step 2 Select join style for exons



Gene Gene ID	Gene Gene Symbol	Gene Gene Biotype	Gene Chromosome ID	Chromosome Organism . Name	Gene Exons
105005609	LOC105005609	protein_coding	LG14	Esox lucius	21 Exons
105005611	LOC105005611	protein_coding	LG14	Esox lucius	5 Exons
105005612	LOC105005612	protein_coding	LG14	Esox lucius	14 Exons
105005613	tmem178b	protein_coding	LG14	Esox lucius	5 Exons
105005663	sec31a	protein_coding	LG14	Esox lucius	346 Exons
105005664	LOC105005664	lncRNA	LG14	Esox lucius	4 Exons
105005665	lin54	protein_coding	LG14	Esox lucius	56 Exons
105005667	LOC105005667	protein_coding	LG14	Esox lucius	4 Exons

Fig. 23: Example 3, Step 3 Query results with exon constraint

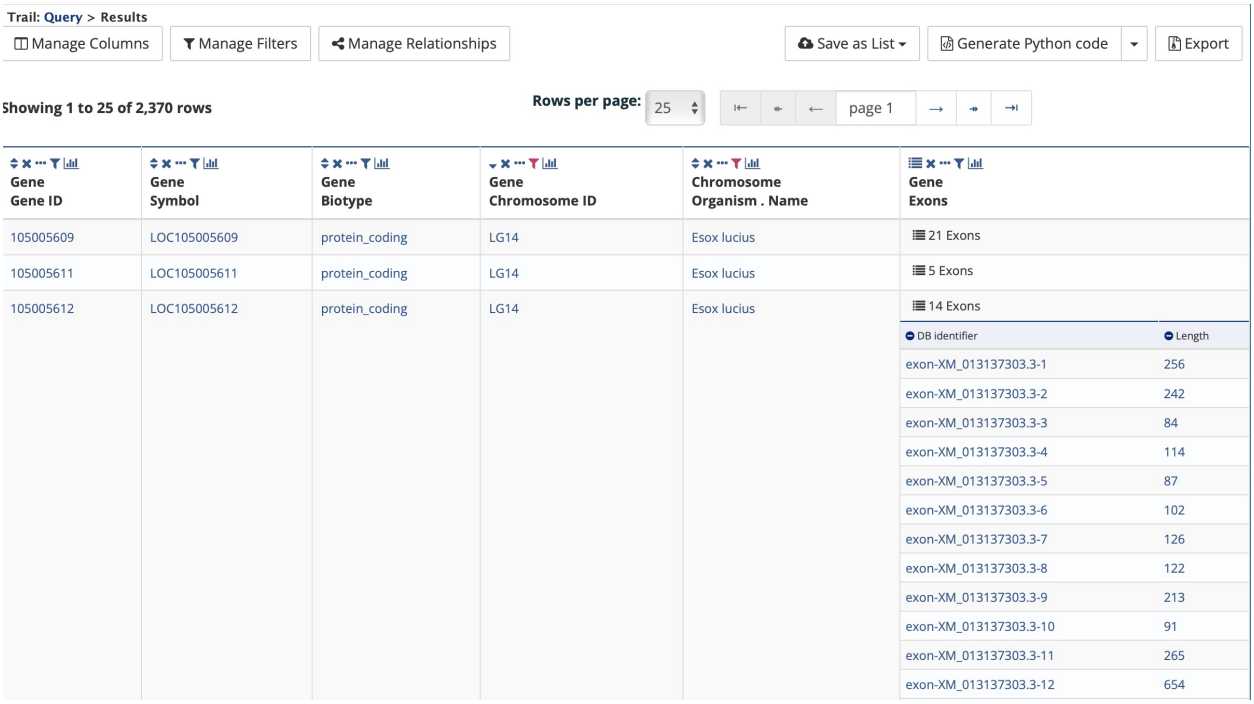


Fig. 24: Example 3, Step 4 Query results with exon column expanded

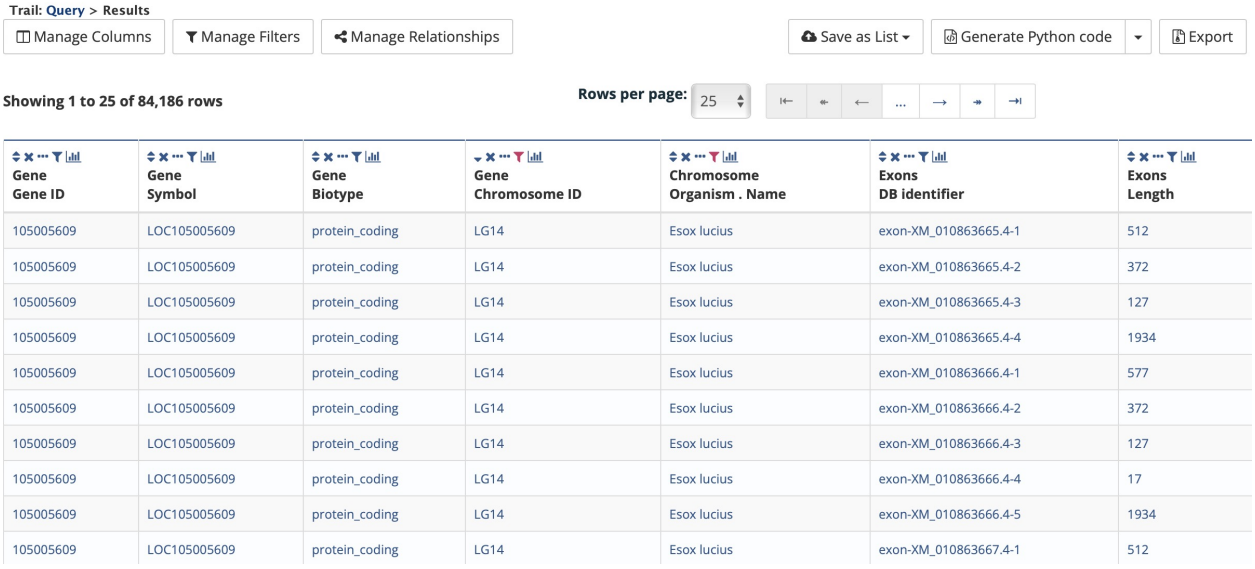


Fig. 25: Example 3, Step 5 Query results with default join style run for genes containing exons

CHAPTER

THREE

REPORT PAGES

All objects in AquaMine (e.g., gene, protein, transcript, publication) have report pages that can viewed after running a query. It allows users to view all available information for that object while providing links to related objects. As an example, we can revisit the templates example. In the list of templates under the **Templates** tab on the AquaMine home page, select Gene -> Protein Sequences to query AquaMine to retrieve all protein sequencess for a given gene. Enter the Gene ID "ENSELUG00000001587" into the the LOOKUP search box then click **Show Results**. In the results table, note that every entry is contains a link. You can mouse over any link to bring up a summary of that object. If we hover over the first Gene ID, we can see a summary box that includes information about that particular gene.

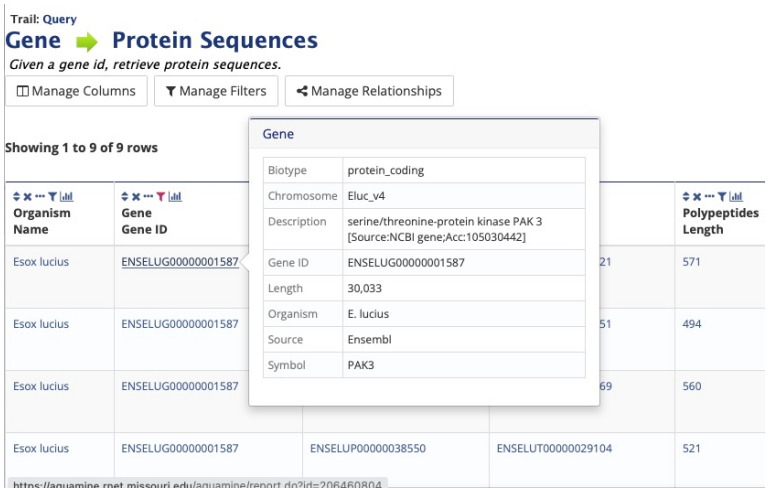


Fig. 1: Summary for gene entry in query results table

Clicking on that same link will bring up its report page that includes a comprehensive overview for that gene. The report page header shows the Gene ID and its Biotype, for this example, protein coding. The tabs at the top of the page in the Quick Links menu bar quickly bring you to the data listed. The column on the right side of the report page displays external links to other Mines and databases.

The content of the report page is divided into categories based on the type of information provided for that particular object. Clicking on links within each category bring up more details about the objects of interest.

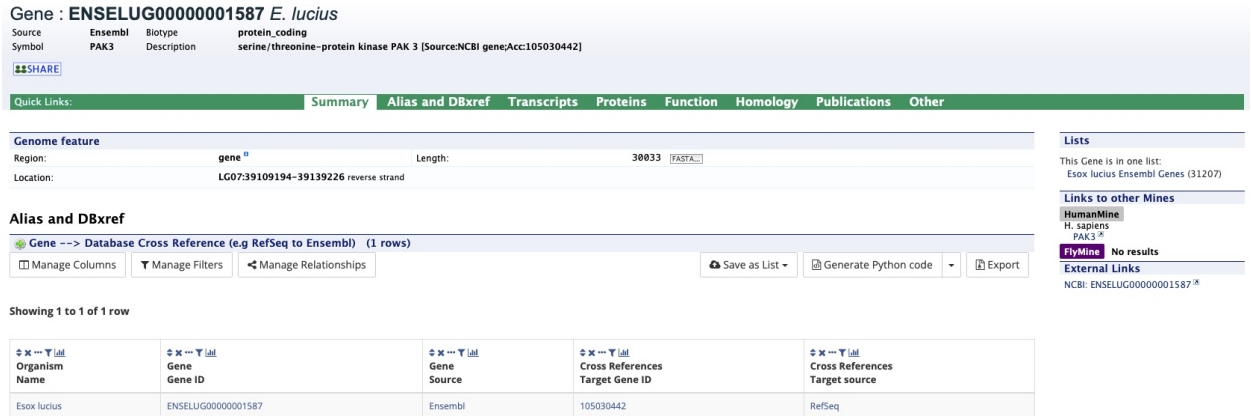


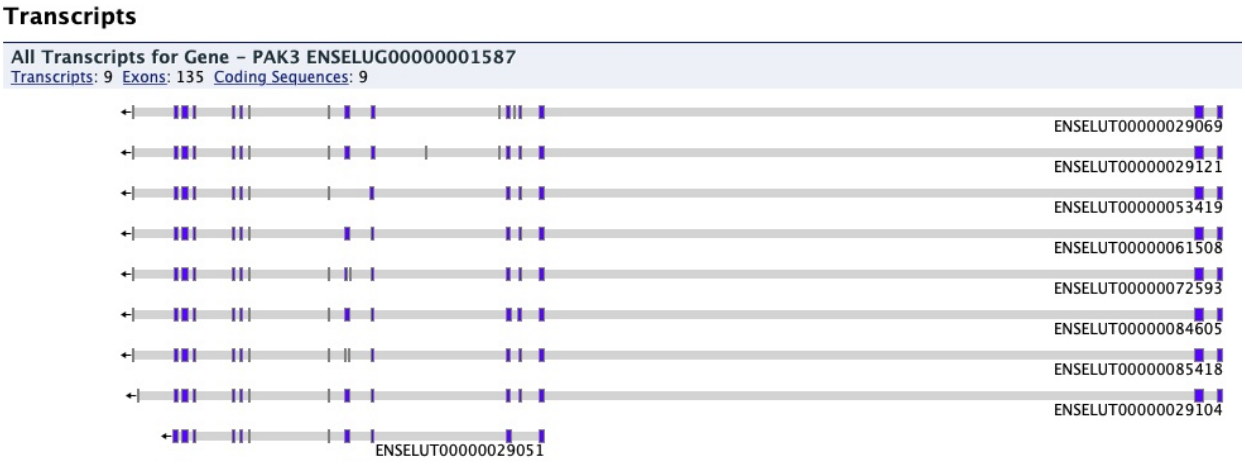
Fig. 2: Report page for protein-coding gene

3.1 Summary

The **Summary** section near the top of the report provides information on the gene such as its length, chromosome location, and strand information. Users can also get the complete FASTA sequence of the gene by clicking on the FASTA tab.

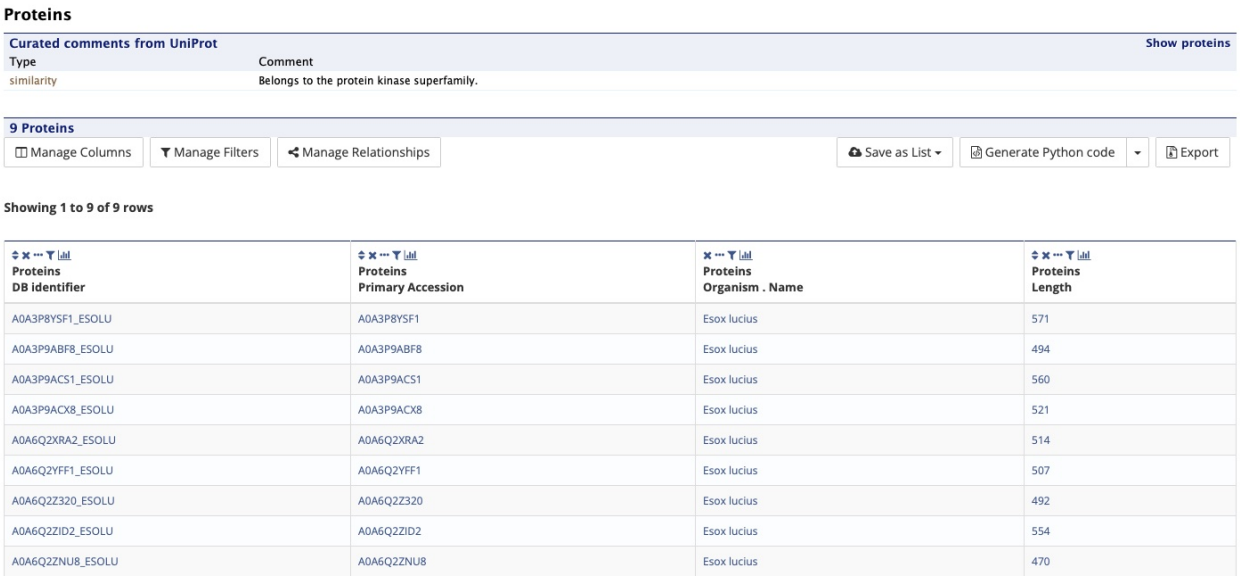
3.2 Transcripts

The **Transcripts** section contains information about the gene model, such as transcripts and exons. Links to FASTA files are included where applicable.



3.3 Proteins

The **Proteins** section provides information about the protein product of the gene. The comments section gives a brief description about the protein along with the UniProt accession and links to any outside data sets.



Proteins

Curated comments from UniProt — Show proteins

Type	Comment
similarity	Belongs to the protein kinase superfamily.

9 Proteins

Manage Columns | Manage Filters | Manage Relationships | Save as List | Generate Python code | Export

Showing 1 to 9 of 9 rows

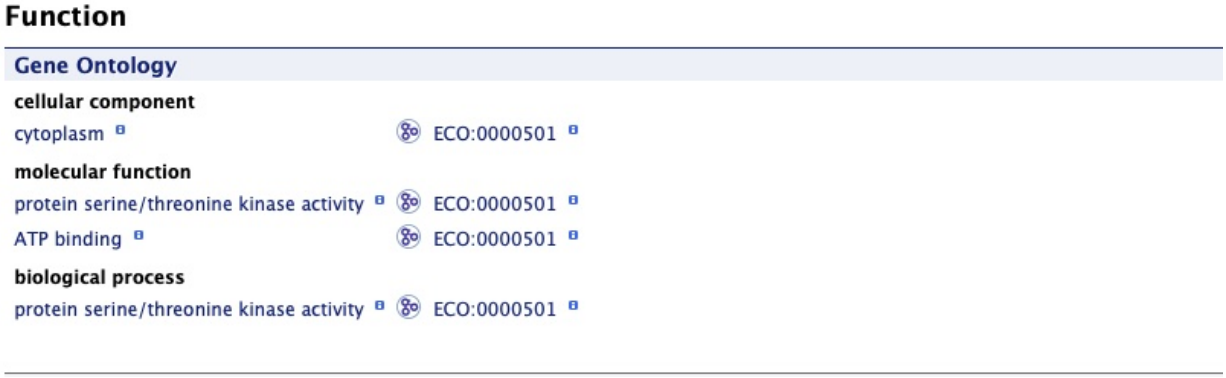
Proteins DB identifier	Proteins Primary Accession	Proteins Organism . Name	Proteins Length
A0A3P8YSF1_ESOLU	A0A3P8YSF1	Esox lucius	571
A0A3P9ABF8_ESOLU	A0A3P9ABF8	Esox lucius	494
A0A3P9ACS1_ESOLU	A0A3P9ACS1	Esox lucius	560
A0A3P9ACX8_ESOLU	A0A3P9ACX8	Esox lucius	521
A0A6Q2XRA2_ESOLU	A0A6Q2XRA2	Esox lucius	514
A0A6Q2YFF1_ESOLU	A0A6Q2YFF1	Esox lucius	507
A0A6Q2Z320_ESOLU	A0A6Q2Z320	Esox lucius	492
A0A6Q2ZID2_ESOLU	A0A6Q2ZID2	Esox lucius	554
A0A6Q2ZNU8_ESOLU	A0A6Q2ZNU8	Esox lucius	470

3.4 Function

The **Function** section displays Gene Ontology annotations for a gene. Annotations are divided into three categories:

- Cellular Component
- Molecular Function
- Biological Process

The GO terms are displayed along with the evidence code indicating how the annotations were derived. A results with Pathway information is also displayed if applicable.



Function

Gene Ontology

cellular component

cytoplasm — ECO:0000501

molecular function

protein serine/threonine kinase activity — ECO:0000501

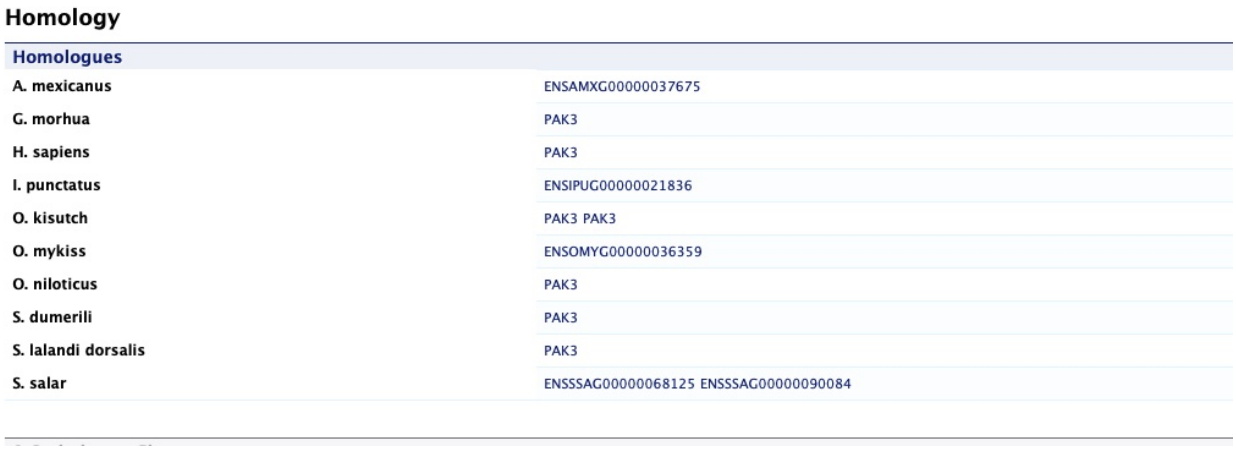
ATP binding — ECO:0000501

biological process

protein serine/threonine kinase activity — ECO:0000501

3.5 Homology

The **Homology** section provides information for all homologues and displays a summarized view of the homologues reported in different organisms.



Homology

Homologues	
A. mexicanus	ENSAMXG00000037675
G. morhua	PAK3
H. sapiens	PAK3
I. punctatus	ENSIPUG00000021836
O. kisutch	PAK3 PAK3
O. mykiss	ENSOMYG00000036359
O. niloticus	PAK3
S. dumerili	PAK3
S. lalandi dorsalis	PAK3
S. salar	ENSSSAG00000068125 ENSSSAG00000090084

3.6 Publications

The **Publications** section displays a table of publications related to the gene with links to full citations.

Publications

1 Publications

Manage Columns | Manage Filters | Manage Relationships | Save as List | Generate Python code | Export

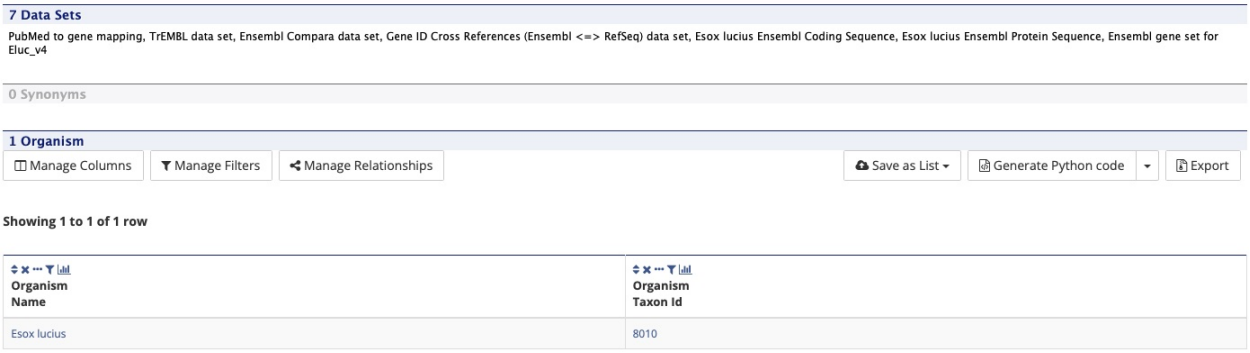
Showing 1 to 1 of 1 row

Publications First Author	Publications Title	Publications Year	Publications Journal	Publications Volume	Publications Pages	Publications PubMed ID
Rondeau Eric B	The genome and linkage map of the northern pike (Esox lucius): conserved synteny revealed between the salmonid sister group and the Neoteleostei.	2014	PLoS One	9	e102089	25069045

3.7 Other

This last section provides miscellaneous information that do not fit into any of the above categories. This example lists other data sets that the gene belongs to.

Other


7 Data Sets
PubMed to gene mapping, TrEMBL data set, Ensembl Compara data set, Gene ID Cross References (Ensembl <=> RefSeq) data set, Esox lucius Ensembl Coding Sequence, Esox lucius Ensembl Protein Sequence, Ensembl gene set for Eluc_v4
0 Synonyms
1 Organism
Manage Columns
Manage Filters
Manage Relationships
Save as List
Generate Python code
Export
Showing 1 to 1 of 1 row
Organism Name
Organism Taxon Id
Esox lucius
8010

CHAPTER

FOUR

GENOMIC REGIONS SEARCH

The **Genomic Regions Search** is a tool to fetch features that are within a given set of genomic coordinates or are within a given number of bases flanking the coordinates.

To begin this type of search, click the **Regions** tab on the menu bar. A form will appear asking for the search parameters (organism, feature types, genomic coordinates, etc.)

The coordinates must have one of three formats:

1. chromosome_number:start..end
2. chromosome_number:start-end
3. chromosome_number start end (tab delimited)

Click on the input examples above the text input box (number 4) to view a representative set of coordinates in each format. Click the **Genome coordinates help** link near the top of the form for more detailed information on the input format requirements.

During a search, regions may be extended on either side of the genomic coordinates using the slider or by entering text in the field to the left of the slide bar. There is also the option to perform a strand-specific region search using the checkbox at the bottom of the form (number 6)

As an example, select *Ictalurus punctatus* from the Select Organism drop-down, and notice that the associated Assembly appears in the next drop-down menu, *IpCoco_1.2*. Under the Select Feature Types section, check the box next to Gene, and enter the following coordinates into the genomic regions search text field:

14:6000000..6800000

Click the search box to conduct the genomic regions search. If there are no overlaps within your search coordinates, the search can be done again with the search region extended using the slide bar or entering text into the search box (e.g., 10k).

The search results page presents a list of features present within the genomic interval that was searched. In this case, the feature type was limited to Gene. The results may be exported as tab-separated or comma-separated values. If they contain genomic features, there is also the option to saved the results in GFF3 or BED format. The FASTA sequences of the features may also be downloaded. Links within the features provide detailed reports. If users are interested in creating a list of particular features from the result page then they can filter based on feature type (if applicable), shown in red box, and click on **Go**.

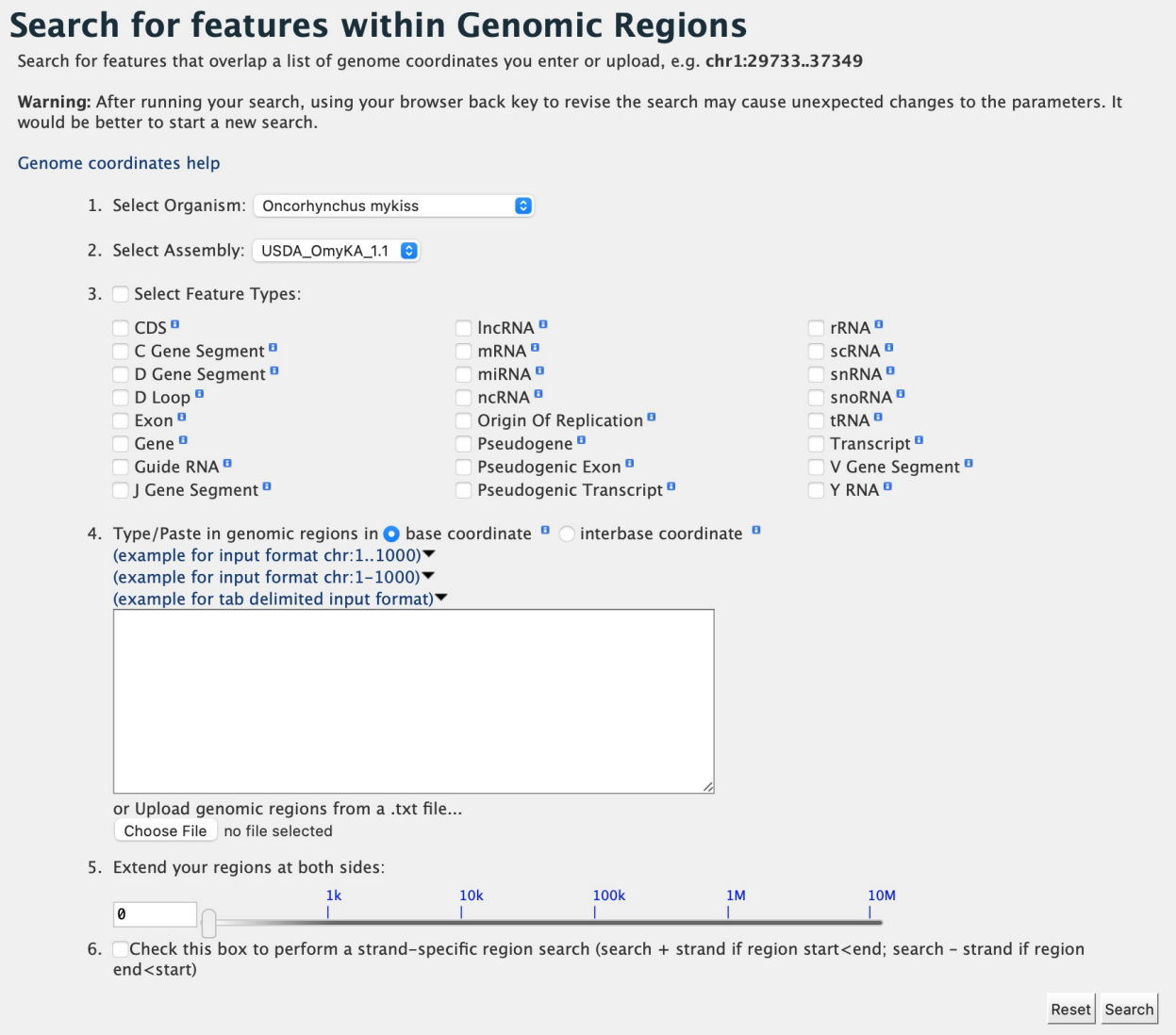


Fig. 1: Genomic Regions search form

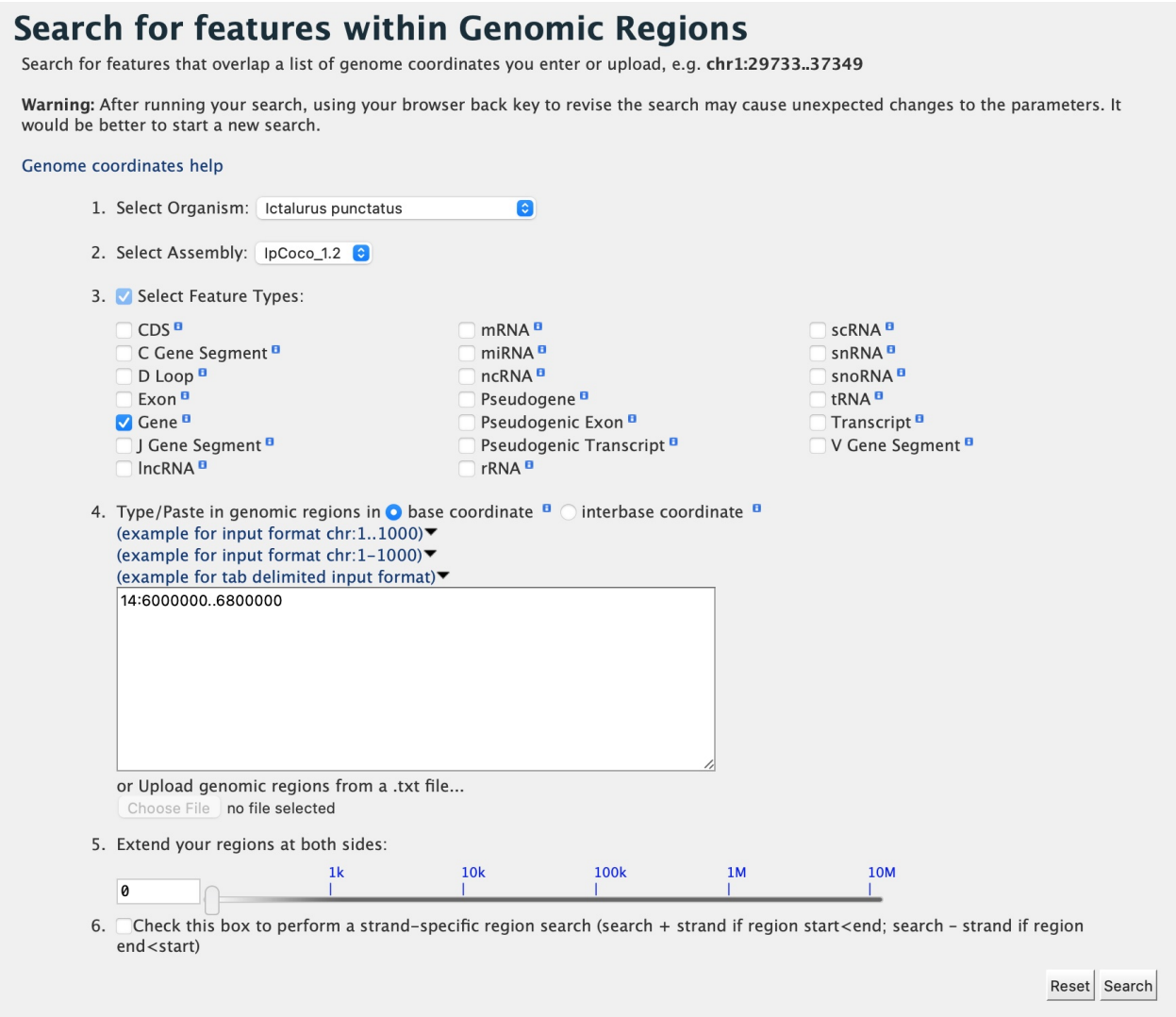


Fig. 2: Genomic Regions search example with *Bos taurus*

Fig. 3: Genomic Regions search results

CHAPTER

FIVE

LISTS

5.1 Creating Lists

Users may create and save lists of features, such as gene IDs, transcript IDs, gene symbols, etc. The list tool searches the database for the list items and attempts to convert each identifier to the selected type. Click on the Lists tab from the menu to access the full list upload form. A short version of the form is also in the Quick List box on the home page.

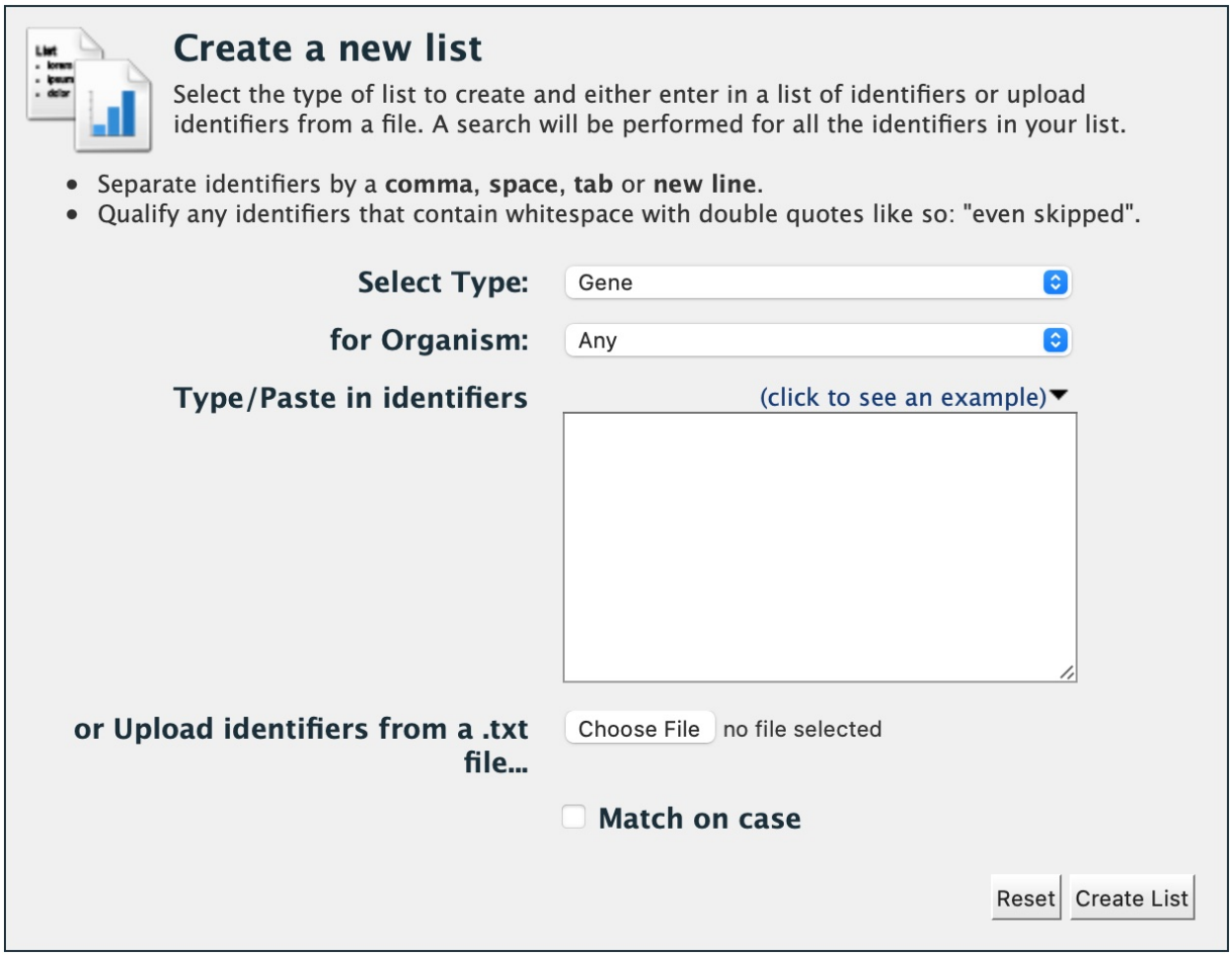


Fig. 1: List upload form

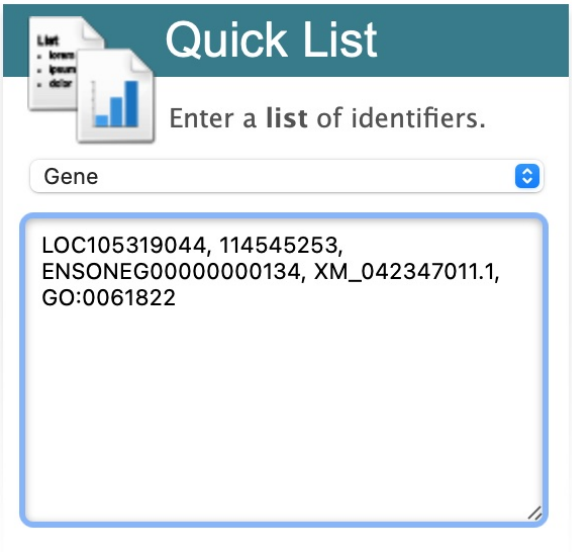


Fig. 2: Quick list from AquaMine home page

As an example, enter the following comma-separated identifiers into the Lists upload form under the **Lists** tab. Notice that they do not have to be in the same format. A Summary table is displayed with the results of searching for each of the five identifiers in the list.

LOC105319044, 114545253, wisp1, ENSONEG00000000134, gstk1

Leave the **Select Type** drop-down menu to **Gene** and the **Organism** drop-down to **Any**. Click on **Create List**. Note that you can also upload a list from a .txt file.

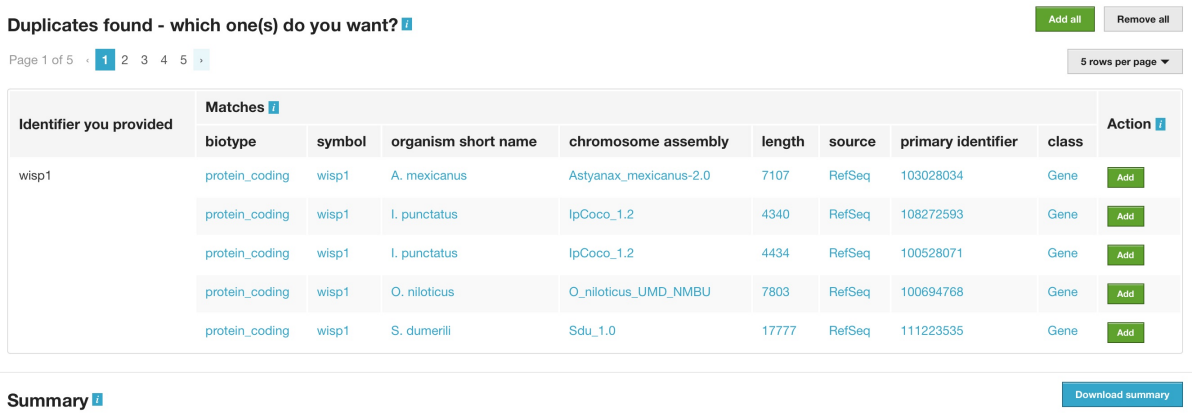


Fig. 3: List Example: Search results for list of identifiers

The summary table provides information regarding those identifiers that had a direct hit without any duplicates. If there are any duplicates, users can decide to add the relevant entries individually by clicking on the **Add** button under the **Action** column or choosing the **Add all** tab. Here we will click **Add all**. Once the selections have been added, the list can be saved by clicking the **Save a list of 27 Genes** button on the top of the summary table. Name the list by entering text into the **Choose a name for the list** box at the top of the results page.

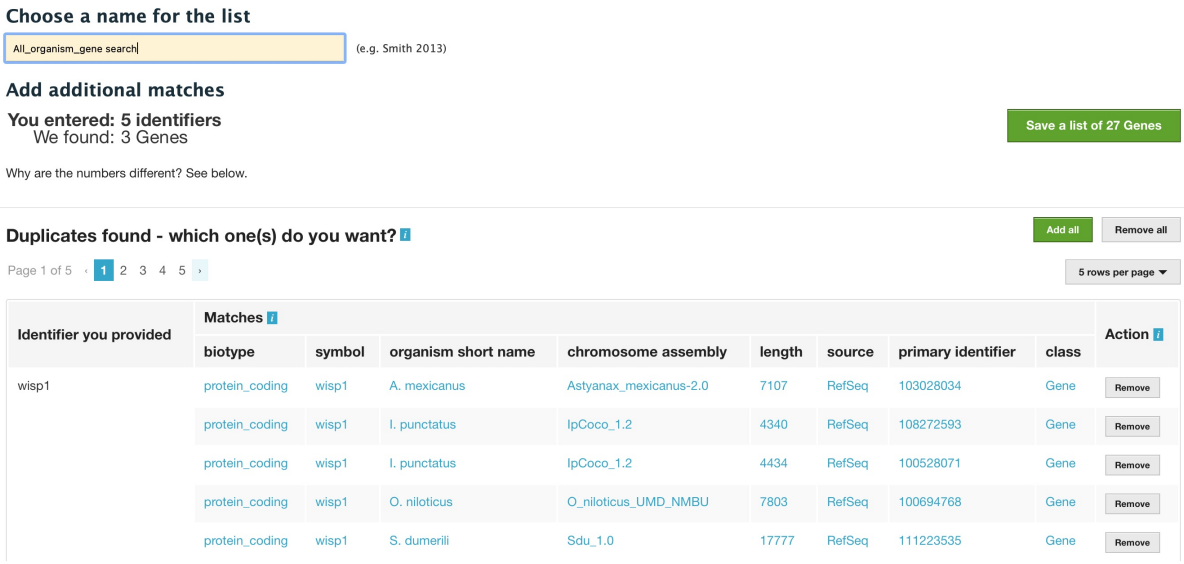


Fig. 4: List Example: Saving list of identifiers

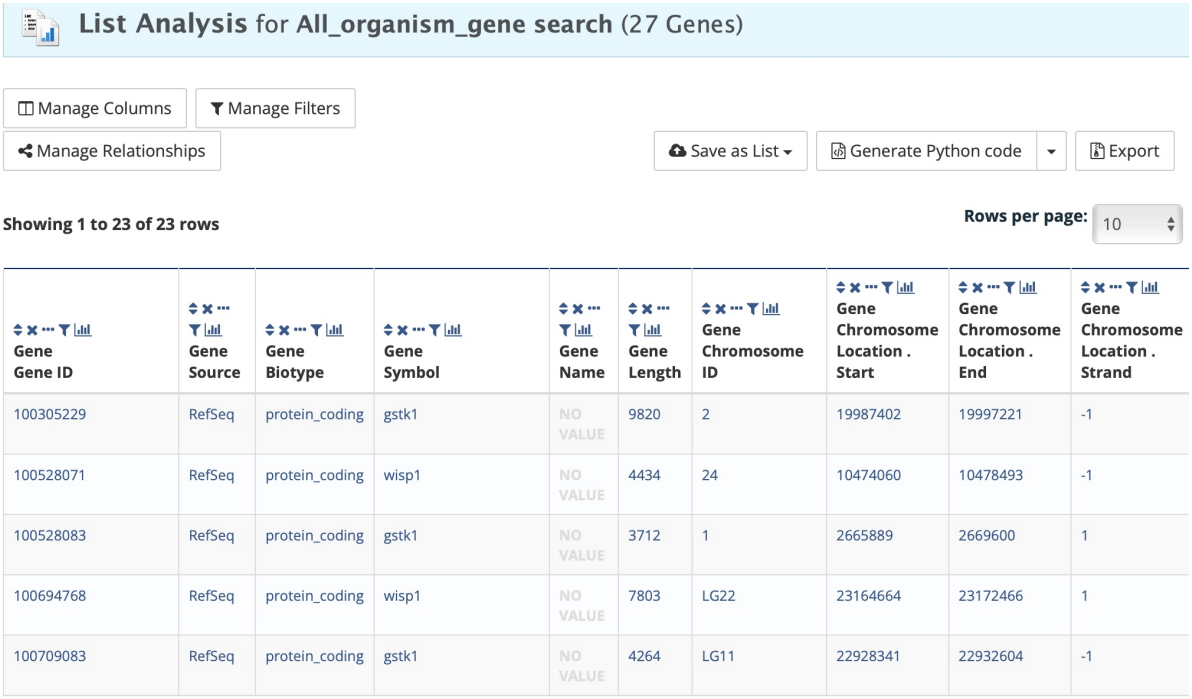
After the list is saved, users are presented with a **List Analysis** page. This page provides users with widgets to perform analyses on gene lists that they have created.

The selection of widgets provided on the List Analysis page depend on the contents of the list. The available widgets for this list example include:

1. Gene Ontology Enrichment
2. Publication Enrichment
3. Pathway Enrichment

5.2 Saving Lists

To see your saved lists, click the **View** tab on the **Lists** page. Note the saved list is highlighted in light purple and has a "MY" label in the corner. The remaining lists are gene lists for the organisms in AquaMine that can be used for future analyses. If not logged in, lists will be saved temporarily during your current session. However, you must be logged in to save your lists permanently. Further analyses of lists can be done with the **Actions** links at the top of the list. The links become active once lists are selected for analyses. Saved lists may also be accessed from the **MyMine** menu tab.



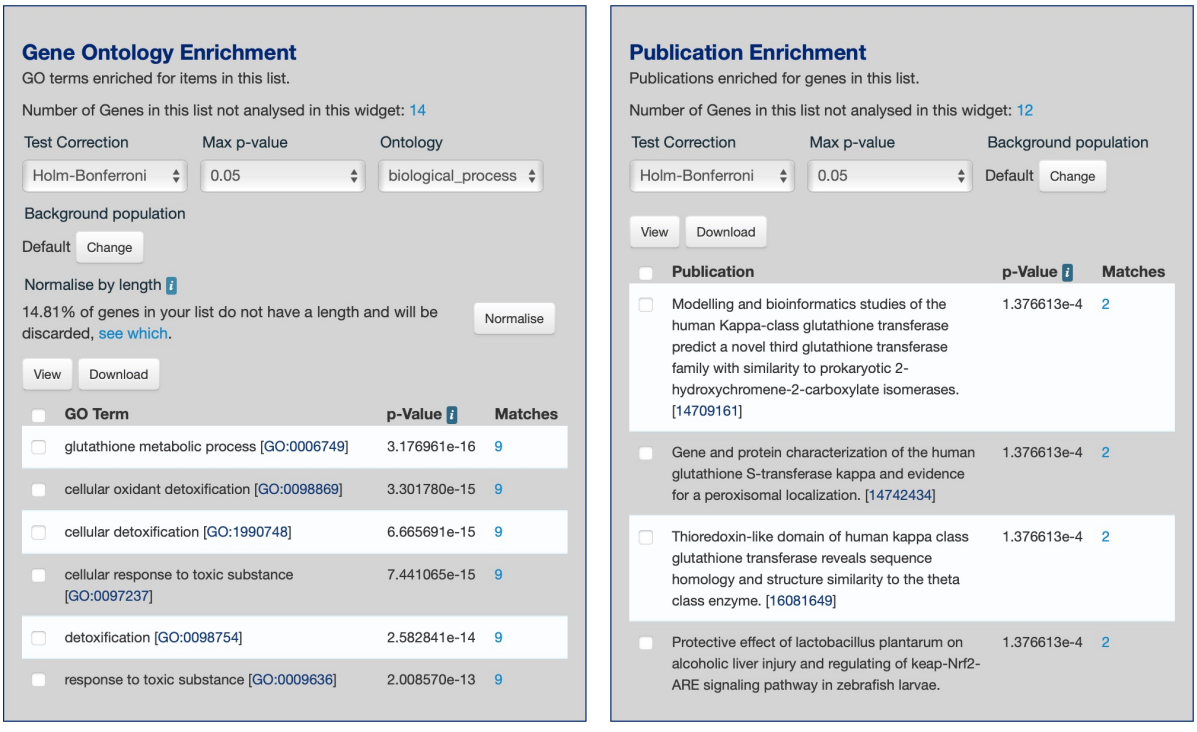
List Analysis for **All_organism_gene search** (27 Genes)

Manage Columns | Manage Filters | Manage Relationships | Save as List | Generate Python code | Export

Showing 1 to 23 of 23 rows — **Rows per page:** 10

Gene Gene ID	Gene Source	Gene Biotype	Gene Symbol	Gene Name	Gene Length	Gene Chromosome ID	Gene Chromosome Location . Start	Gene Chromosome Location . End	Gene Chromosome Location . Strand
100305229	RefSeq	protein_coding	gstk1	NO VALUE	9820	2	19987402	19997221	-1
100528071	RefSeq	protein_coding	wisp1	NO VALUE	4434	24	10474060	10478493	-1
100528083	RefSeq	protein_coding	gstk1	NO VALUE	3712	1	2665889	2669600	1
100694768	RefSeq	protein_coding	wisp1	NO VALUE	7803	LG22	23164664	23172466	1
100709083	RefSeq	protein_coding	gstk1	NO VALUE	4264	LG11	22928341	22932604	-1

Fig. 5: List Example: Analysis for gene list



Gene Ontology Enrichment

GO terms enriched for items in this list.

Number of Genes in this list not analysed in this widget: 14

Test Correction: Holm-Bonferroni | Max p-value: 0.05 | Ontology: biological_process

Background population: Default Change

Normalise by length

14.81% of genes in your list do not have a length and will be discarded, see which. Normalise

View | Download

GO Term	p-Value	Matches
glutathione metabolic process [GO:0006749]	3.176961e-16	9
cellular oxidant detoxification [GO:0098869]	3.301780e-15	9
cellular detoxification [GO:1990748]	6.665691e-15	9
cellular response to toxic substance [GO:0097237]	7.441065e-15	9
detoxification [GO:0098754]	2.582841e-14	9
response to toxic substance [GO:0009636]	2.008570e-13	9

Publication Enrichment

Publications enriched for genes in this list.

Number of Genes in this list not analysed in this widget: 12

Test Correction: Holm-Bonferroni | Max p-value: 0.05 | Background population: Default Change

View | Download

Publication	p-Value	Matches
Modelling and bioinformatics studies of the human Kappa-class glutathione transferase predict a novel third glutathione transferase family with similarity to prokaryotic 2-hydroxychromene-2-carboxylate isomerases. [14709161]	1.376613e-4	2
Gene and protein characterization of the human glutathione S-transferase kappa and evidence for a peroxisomal localization. [14742434]	1.376613e-4	2
Thioredoxin-like domain of human kappa class glutathione transferase reveals sequence homology and structure similarity to the theta class enzyme. [16081649]	1.376613e-4	2
Protective effect of lactobacillus plantarum on alcoholic liver injury and regulating of keap-Nrf2-ARE signaling pathway in zebrafish larvae.	1.376613e-4	2

Fig. 6: List Example: Displayed widgets for list analysis

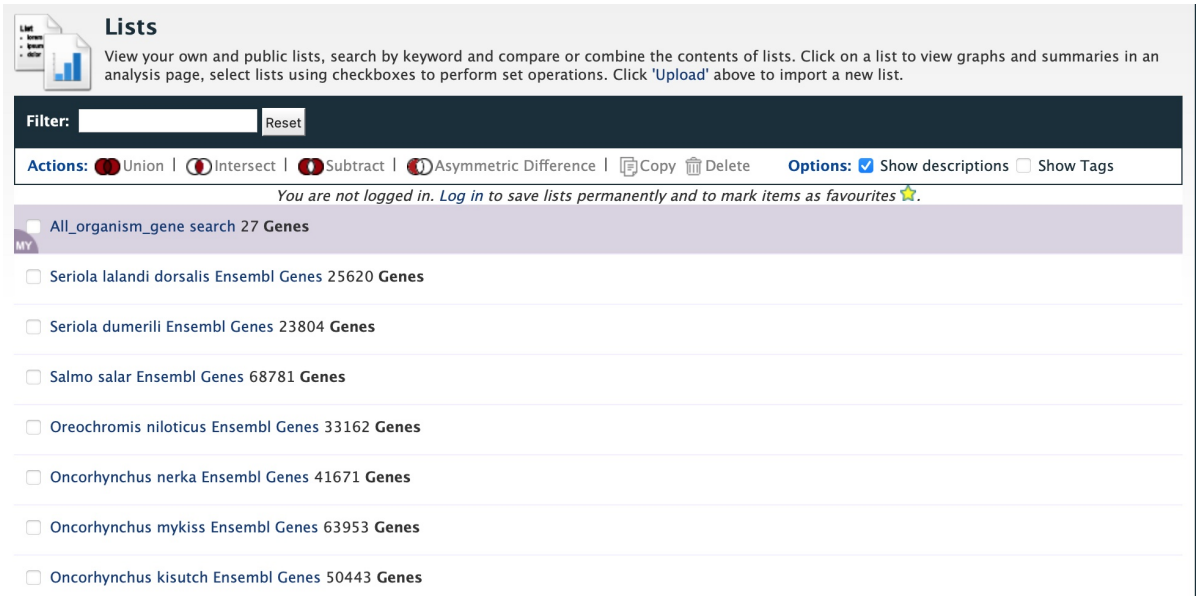


Fig. 7: List Example: Saved user lists

CHAPTER

SIX

MYMINE

MyMine serves as a portal where logged-in users may manage their lists, queries, templates, and account details.

To access MyMine, click on the MyMine menu tab. A submenu appears with six options:

Lists - Lists saved by the user when logged in.

History - List of most recently run queries.

Queries - List of saved queries.

Templates - Templates created or marked as "favorite" by the user.

Password - Password reset form.

Account Details - User preferences form.

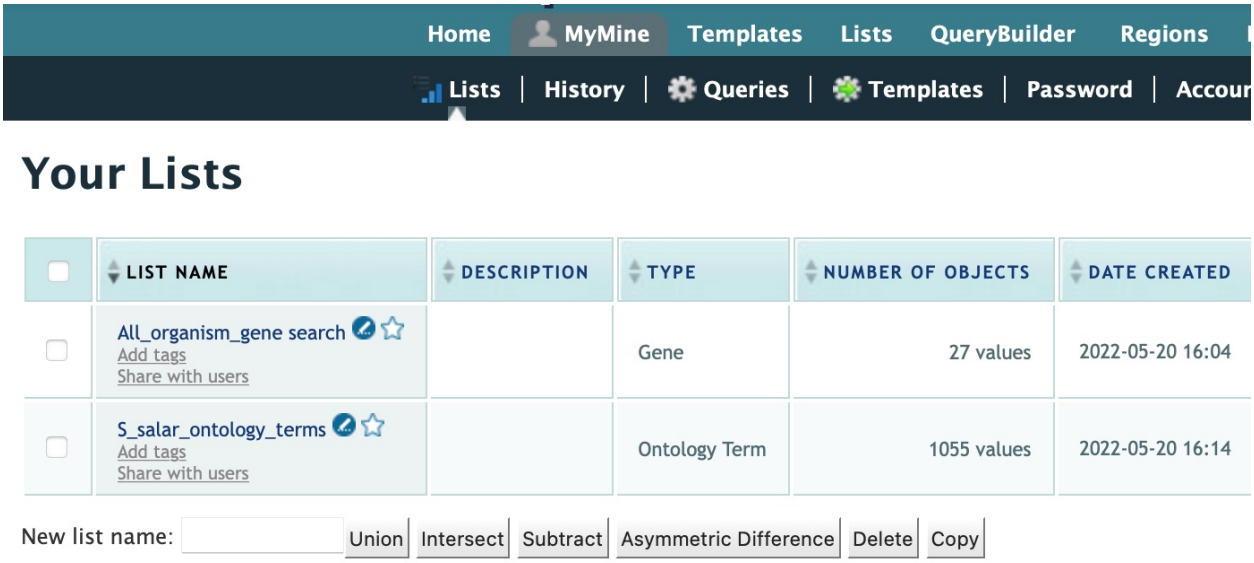


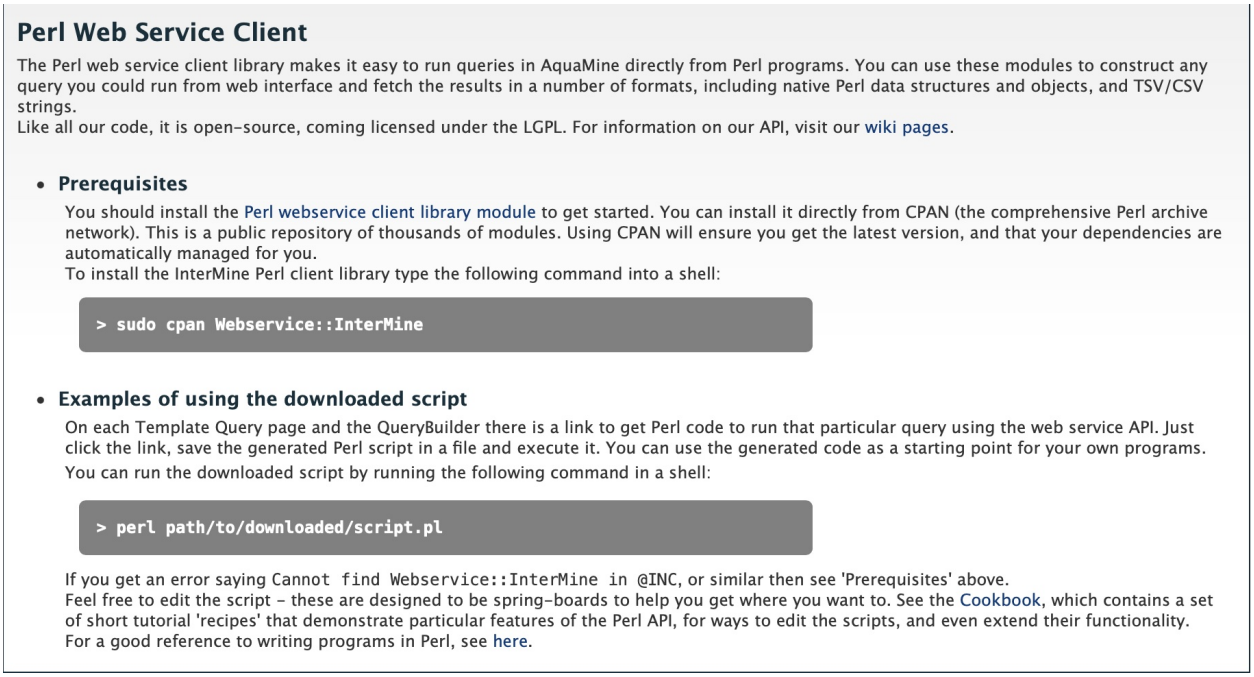
Fig. 1: Saved lists found under MyMine. Note that currently saved lists can be selected for analyses to contribute to new lists.

CHAPTER

SEVEN

API

An API is available for users who would like to programmatically access FAANGMine.



Perl Web Service Client

The Perl web service client library makes it easy to run queries in AquaMine directly from Perl programs. You can use these modules to construct any query you could run from web interface and fetch the results in a number of formats, including native Perl data structures and objects, and TSV/CSV strings.
Like all our code, it is open-source, coming licensed under the LGPL. For information on our API, visit our wiki pages.

- **Prerequisites**

  You should install the Perl webservice client library module to get started. You can install it directly from CPAN (the comprehensive Perl archive network). This is a public repository of thousands of modules. Using CPAN will ensure you get the latest version, and that your dependencies are automatically managed for you.
  To install the InterMine Perl client library type the following command into a shell:

  ```
  > sudo cpan Webservice::InterMine
  ```

- **Examples of using the downloaded script**

  On each Template Query page and the QueryBuilder there is a link to get Perl code to run that particular query using the web service API. Just click the link, save the generated Perl script in a file and execute it. You can use the generated code as a starting point for your own programs.
  You can run the downloaded script by running the following command in a shell:

  ```
  > perl path/to/downloaded/script.pl
  ```

  If you get an error saying `Cannot find Webservice::InterMine in @INC`, or similar then see 'Prerequisites' above.
  Feel free to edit the script - these are designed to be spring-boards to help you get where you want to. See the Cookbook, which contains a set of short tutorial 'recipes' that demonstrate particular features of the Perl API, for ways to edit the scripts, and even extend their functionality.
  For a good reference to writing programs in Perl, see here.

Perl, Python, Ruby, and Java are the languages supported by the InterMine API.

For more detailed information, view the InterMine documentation.

CHAPTER

EIGHT

DATA SOURCES

The Data Sources table provides a description of the datasets that are integrated into AquaMine, along with their download location, version or release, citations wherever applicable, and any additional comments. It is a useful place to begin to find out what data has been loaded for your organism of interest.

Data Category	Description	Organism (* US Aquaculture Species)	Source	PubMed	Link
Assembly	Genome Assembly	*Astyanax mexicanus* (Mexican tetra)	Astyanax_mexicanus-2.0	McGaugh et al. - PubMed 25329095	NCBI FTP
		Clupea harengus (Atlantic herring)	Ch_v2.0.2	Pettersson et al. - PubMed 31649060	NCBI FTP
		Crassostrea gigas (Pacific oyster)*	cgigas_uk_roslin_v1	Peñaloza et al. - PubMed 33764468	NCBI FTP
		Crassostrea virginica (Eastern oyster)*	C_virginica-3.0	Not published	NCBI FTP
		Esox lucius (Northern pike)	Eluc_v4	Rondeau et al. - PubMed 25069045	NCBI FTP
		Gadus morhua (Atlantic cod)	gadMor3.0	Tørresen et al. - PubMed 28100185	NCBI FTP
		Gasterosteus aculeatus aculeatus (Three-spined stickleback)	GAculeatus_UGA_version5	Nath et al. - PubMed 33598708	NCBI FTP
		Hippoglossus hippoglossus (Atlantic halibut)	fHipHip1.pri	Einfelt et al. - PubMed 33655659	NCBI FTP
		Homarus americanus (American lobster)	GMGI_Hamer_2.0	Polinski et al. - PubMed 34162536	NCBI FTP
		Ictalurus punctatus (Channel catfish)*	IpCoco_1.2	Liu et al. - PubMed 27249958	NCBI FTP
		Mercenaria mercenaria (Northern quahog)	ASM1480567v1.1	Song et al. - PubMed 33487168	NCBI FTP
		Micropterus salmoides (Largemouth bass)	ASM1485139v1	Sun et al. - PubMed 32985096	NCBI FTP
		Morone saxatilis (Striped sea bass)*	NCSU_SB_2.0	Not published	NCBI FTP
		Oncorhynchus kisutch (Coho salmon)*	Okis_V2	Kim et al. - PubMed 26614614	NCBI FTP
		Oncorhynchus mykiss (Rainbow trout)*	USDA_OmyKA_1.1	Gao et al. - PubMed 33616628	NCBI FTP
		Oncorhynchus nerka (Sockeye salmon)	Oner_1.0	Christensen et al. - PubMed 33119641	NCBI FTP
		Oncorhynchus tshawytscha (Chinook salmon)	Otsh_v2.0	Christensen et al. - PubMed 29621340	NCBI FTP
		Oreochromis niloticus (Nile tilapia)*	O_niloticus_UMD_NMBU	Conte et al. - PubMed 30942871	NCBI FTP
		Penaeus monodon (Giant tiger prawn)*	NSTDA_Pmon_1	Quyen et al. - PubMed 32033920	NCBI FTP
		Penaeus vannamei (Pacific white shrimp)*	ASM378908v1	Zhang et al. - PubMed 30664654	NCBI FTP
		Perca flavescens (Yellow perch)*	PFLA_1.0	Feron et al. - PubMed 31903688	NCBI FTP
		Salmo salar (Atlantic salmon)*	Ssal_v3.1	Not published	NCBI FTP

Fig. 1: AquaMine Data Sources table

CHAPTER

NINE

HOW TO CITE

AquaMine is a project supported by the National Science Foundation to address the need for a high performance data mining resource that enables fine-grained querying and integrating of the heterogeneous aquatic species data with existing information, such as functions of known genes and research datasets.

For more generic examples on how to use InterMine, click here.

These are tutorials created by FlyMine that showcase the different features of InterMine.