CandidaMine Documentation

Release 1.0

CandidaMine Documentation

Jun 21, 2023

CandidaMine is an integrative data warehouse for Candida Species genomes and transcriptomes. Powered by Inter-Mine, it provides a user-friendly way to access genomic, proteomic, interaction and literature data.

This user guide is aimed at giving users an introduction to the different parts of CandidaMine and how users can make the most of CandidaMine.

Main site: http://candidamine.org/

CandidaMine Overview

1.1 Home Page

Figure Fig. 1.1 summarize of the CandidaMine Home page layout and the top menu items:
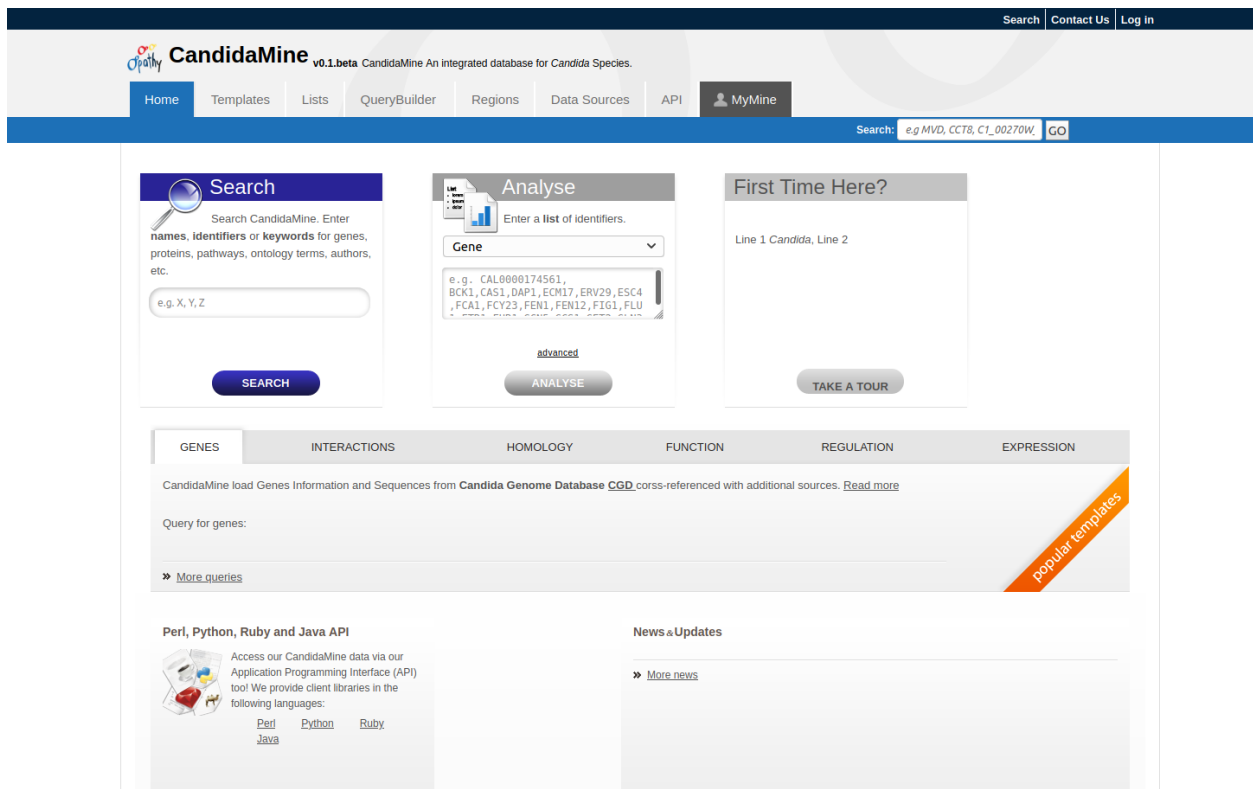

Fig. 1.1: CandidaMine main page.

The top menu items are as the following : *Home* – The home page for CandidaMine.

Templates – List of templates that users may select from based on the nature of their query.

Lists – Allows users to upload lists of genes and perform enrichment analyses. Logged-in users may save their lists for future use.

QueryBuilder – Allows users to build custom queries by browsing the HymenopteraMine data model and customize their results. The queries may be exported to a number of formats including XML.

Regions – Genomic Region Search page where users may enter genomic coordinates and fetch features that fall within the interval. The interval may be extended to increase the range of search.

Data sources – Table of all data sources with their links, date of download, and related publication(s).

API – Describes the InterMine API that allows users to programmatically access CandidaMine.

MyMine – Once users are logged in, MyMine serves as portal for accessing saved lists and saved templates. Users may also check their account details and manage their account using MyMine.

1.2 Searching CandidaMine

1.2.1 Keywords search

The Search box enables users to search keywords from any of the datasets on **CandidaMine**. The search box is located on the main page and in the upper-right corner of each page.

1.2.2 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. A form will appear asking for the search parameters (organism, feature types, genomic coordinates, etc.)

Report Page

Every object (e.g., Gene, Protein, Exon) in *CandidaMine* has a detailed report page. The layout of the report page depends on the data available for the object. Report pages may be accessed by clicking on an object name in the results table after running a query.

Example by keyword search -> search for ASH1. Clicking on an item in the result table will bring up its report page. For example clickng on ASH1 in Candida albicans with show its report page.

The report page Fig. 2.1 provides a complete description for this gene. The header displays the database identifier, followed by the information from the summary window for the gene (organism, symbol, source, etc.) **Biotype** indicates the type of gene; in this case the type is protein coding.

The contents of the report page are divided into categories based on the type of information provided.

2.1 Summary

A **Summary** section near the top of the report provides information on the gene such as its length, chromosome location, and strand information as shown in Fig. 2.2.

2.2 Genomics

2.3 Proteins

The **Proteins** section provides information about the protein product of gene. The comments section gives a brief description about the protein along with the UniProt accession.

2.4 Homology

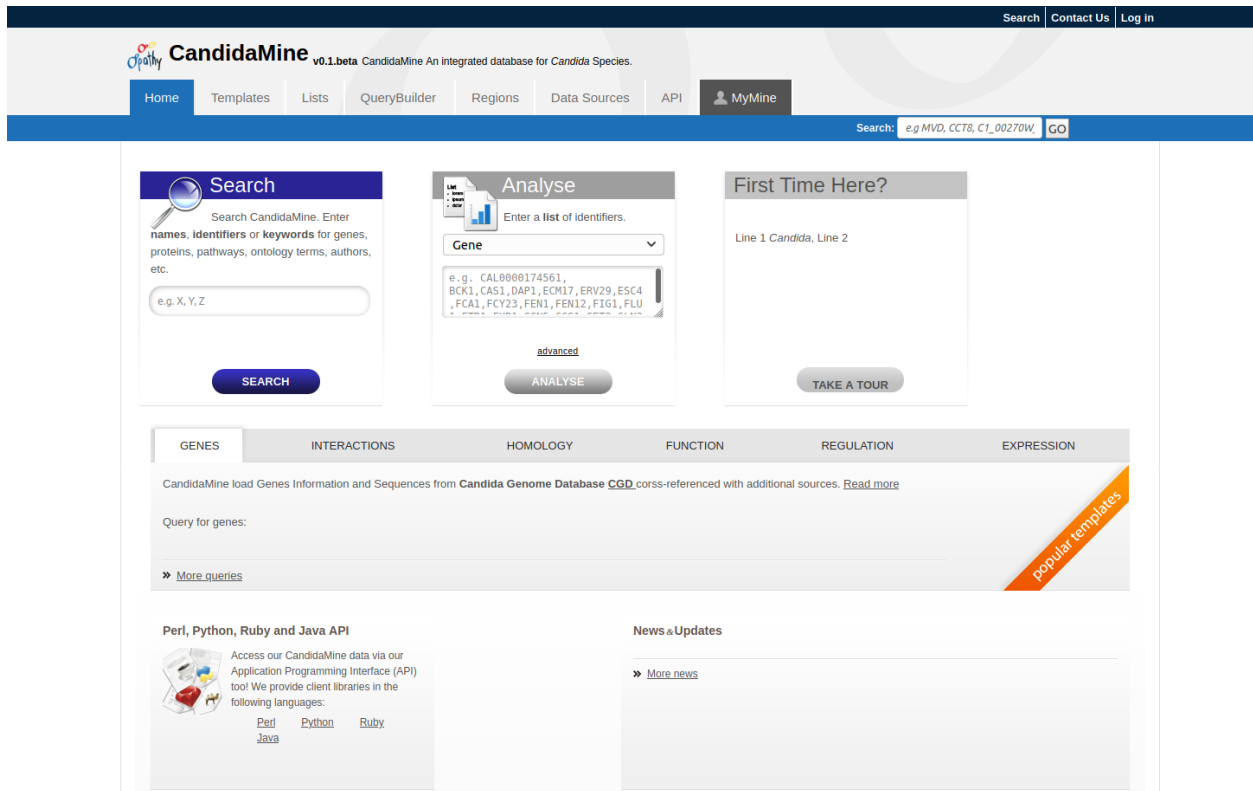The **Homology** section includes information on homologues for the gene.

Fig. 2.1: **ASH1** Report page.

2.5 Expression

2.6 Interactions

2.7 Other

This last section provides miscellaneous information that doesn't fit into any of the above categories, e.g., data sets including a gene, protein domain regions for a protein, etc.

user_guide/images/report_page_summary.png

Fig. 2.2: Summary section of report page.

Template Queries

Another method of searching **CandidaMine** is through the use of **templates** (predefined queries). Popular templates are displayed on the home page, grouped by category (Genes, Protein, Homology, etc.) see Fig. 3.1. The full list of templates may be viewed by clicking the **Templates** menu tab. Fig. 3.2.

3.1 Generate query code

The code for each query may be obtained 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.

3.2 Download results

The search results may also 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 (blue box in the figure below). 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 (orange box in the figure below). 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.

3.3 Customize output

Click the **Manage Columns** button to customize the results table layout. Edit or remove active filters by clicking the **Manage Filters** button. Click **Manage Relationships** to specify the entity relationships within the query.
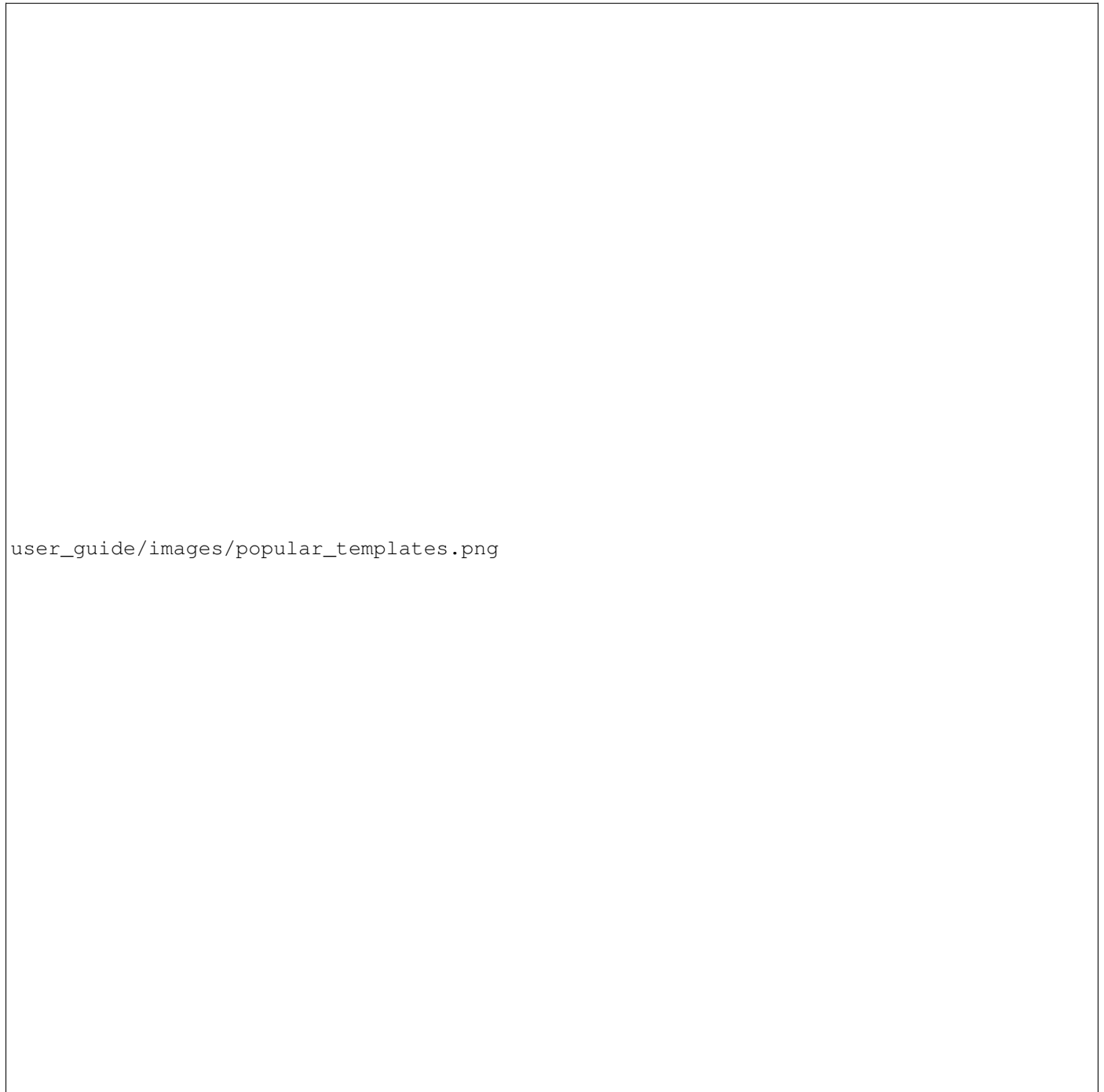
user_guide/images/popular_templates.png

Fig. 3.1: Popular templates on the home page. Templates are grouped by category.

user_guide/images/templates.png

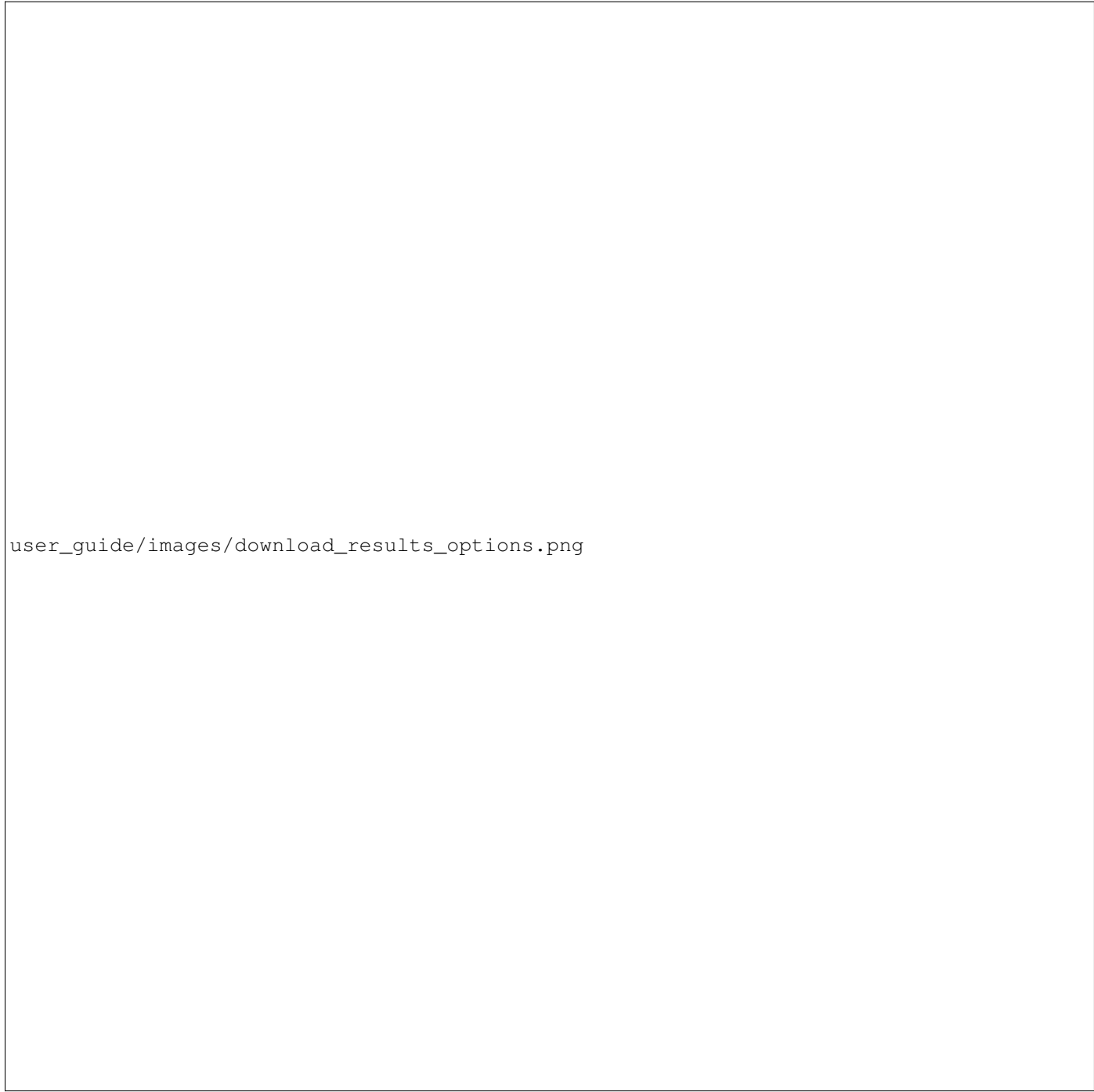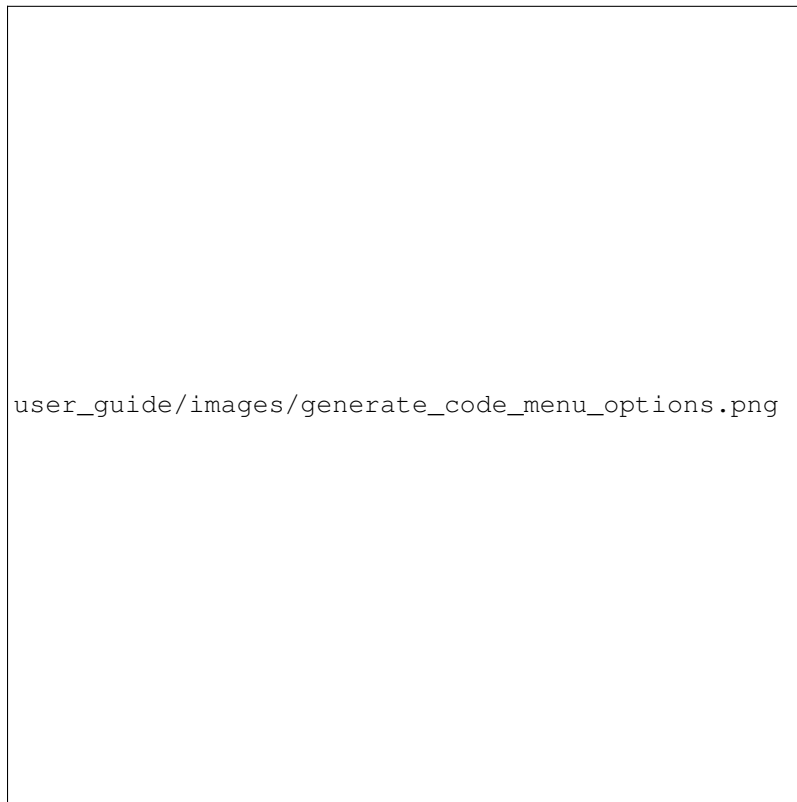Fig. 3.2: List of all available templates on the Templates page.

user_guide/images/generate_code_menu_options.png

Fig. 3.3: Generate code options

user_guide/images/download_results_options.png

Fig. 3.4: Download results options
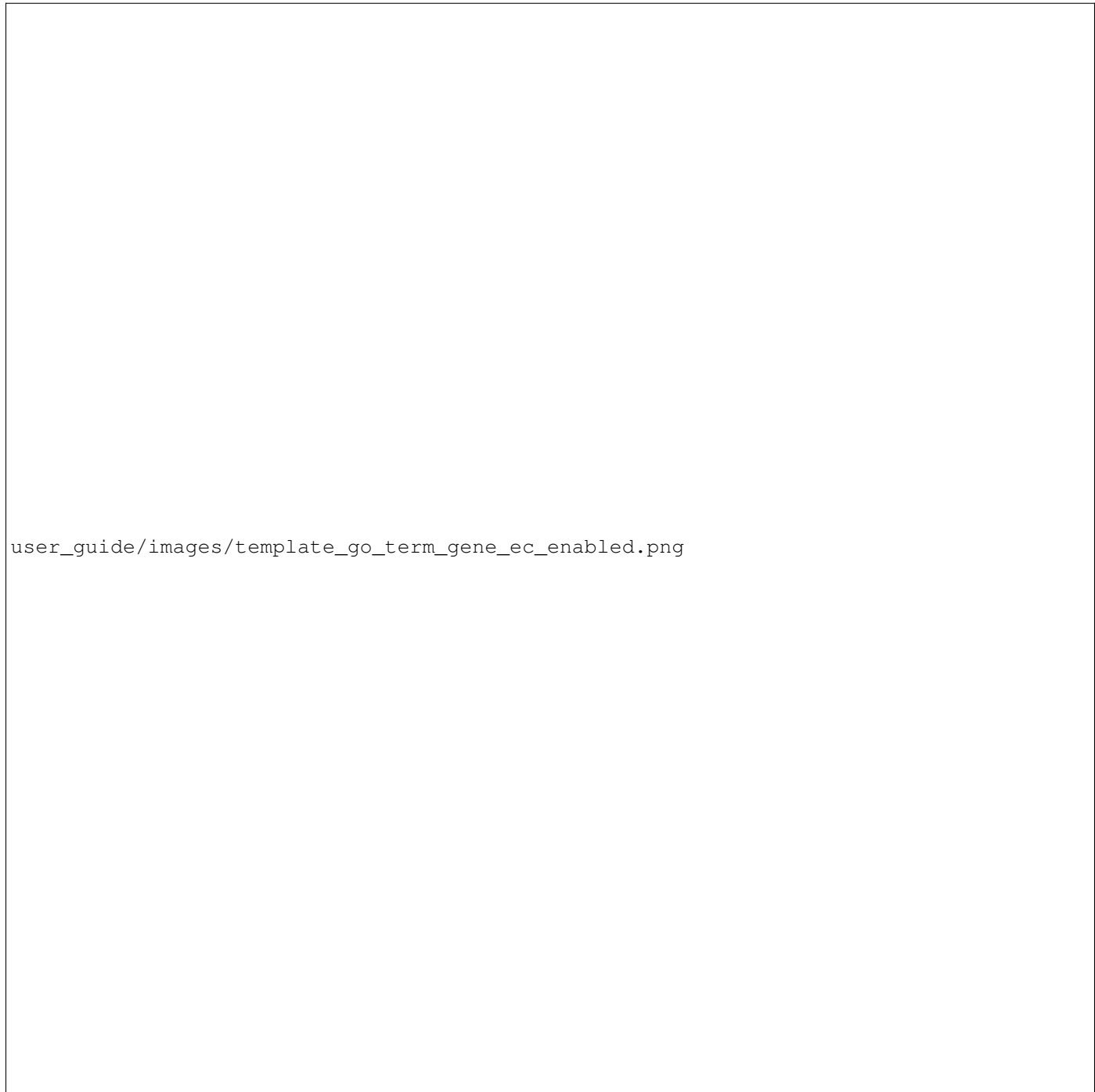
3.4 Optional filters

Some templates have optional filters that are disabled by default. For example, the GO Term –> Gene template has an additional filter for specifying a GO evidence code. To enable this filter, click **ON** below **GO Evidence Code > Code**.

3.5 Examples

3.5.1 Genes to Proteins

3.5.2 INDELS in coding regions

To get all insertion and deletions in coding regions you can run Insertions/Deletions in CDS region template. The templates has some filters to constrains the search for organism of interset , specfic gene, and optionals strains and study PMID as shown in Fig. 3.6.

user_guide/images/template_go_term_gene_ec_enabled.png

Fig. 3.5: Example: GO Term –> Gene template with GO evidence code filter enabled

Fig. 3.6: Insertions/Deletions in CDS region Template Query

Query Buidler

While the templates provided 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. Query builder provides an easy way to create new search queries. Query builder has a fast learning curve and provides flexible tools to design complex queries that could target all stored information in CandidaMine. For more detail documantion about Query Buidler; readers are encouraged to see https://flymine.readthedocs.io/en/latest/query-builder/index.html

4.1 Model browser

After choosing a data type, the **Model browser** appears displaying the attributes for the selected feature class.

4.2 Examples

The following examples will provided details steps on how to use Query Buidler to build your own custom queries.

4.2.1 Example : Querying for INDELS in coding regions

Building a new query starts by choosing a data type of interest e.g gene or transcript based on the required result. After choosing a data type, the Model browser appears displaying the attributes for the selected feature class. Figure shows an example of building a new query to select all insertions and deletions with coding regions of a specific gene of interest filtered by some strains similar to template query shown in Fig. 4.1. In this case Sequence Alteration data type (based on SO terms) was selected Fig. 4.1 A. Then desired attributes that would be retrieved in the result table are selected. To restrict the retrieved sequence alterations to be of Insertion or Deletion, a constraint is added to the query by selecting constain button then configure the filter as shown in Fig. 4.1 B. Sequence Alteration data type is a sequences feature that overlap with other genomic sequence features, we can selected to retrieve all overlapping feature with the result Sequences alteration, however to select only those within coding region we constrain overlapping feature to be of only Exon data type as shown in Fig. 4.1 C. Once Overlapping features are constrained as Exons, more

attributes are shown in the model browser under it e.g parent Gene. Accordingly we can constrain the parent gene of the exons as shown in Fig. 4.1 D and constrain the strains as shown in Fig. 4.1 E.
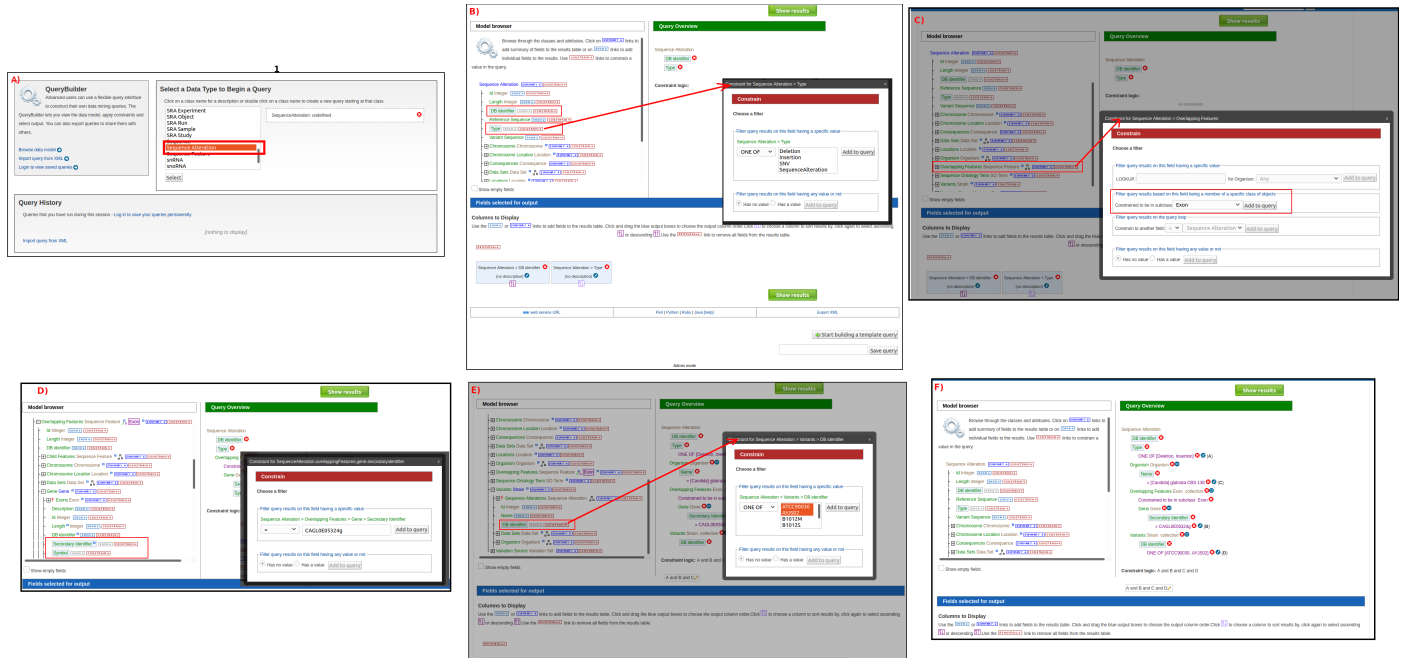

Fig. 4.1: A step by step example on how to build a custom query to retrieve all insertion and deletions within the coding region of a target gene fitler by some strains. A) Select Object of interest in this case is Sequences altarion to begin designing the query. B) add basic attribute to the query result and constraint type attribute to be Deletion and Insertion. C) Constrain overlapping features to be only of type Exons. D) Add basic attribute of the gene from the Exon object and constain Secondary Identifier to specific gene of interest. E) Constrain Variant strain identifier. F) Final layout of the template after specifying all attributes to show in the result and the contains to control the final output.

Lists

A powerful feature of the InterMine framework is the analysis of features lists e.g genes or proteins. Users can store gene lists for example and list of differentially expressed genes from a specific RNASeq experiment then performing GO-term enrichment analysis on such lists.

5.1 Creating Lists

The list tool searches the database for the list items and attempts to convert each identifier to the selected type. User can create list from **Quick List** box on the home page or by clicking on the **Lists** tab from the menu to access the full list upload as seen in Fig. 5.1.

5.1.1 Creating list example

As an example, enter the following identifiers (comma-separated):

```
ASH1, CAL0000174561, FTR1,CAS1,CR_08980C_A
```

Leave the **Select Type** as "Gene" and **Organism** drop-down as "Any". Then click **Create List**. A Summary table is displayed with the results of searching for each of the five identifiers in the list Fig. 5.2.

Next, click **Save a list of 5 Genes**. A **List Analysis** page is presented that contains widgets allowing users to perform analyses on the genes in the list.

The available widgets are:

- Chromosome Distribution.

- Gene Ontology Enrichment.

- **Protein Domain Enrichment.**

    - Domains from Proteins

    - Predicted Domains from genes.

```
user_guide/images/list_upload.png
```
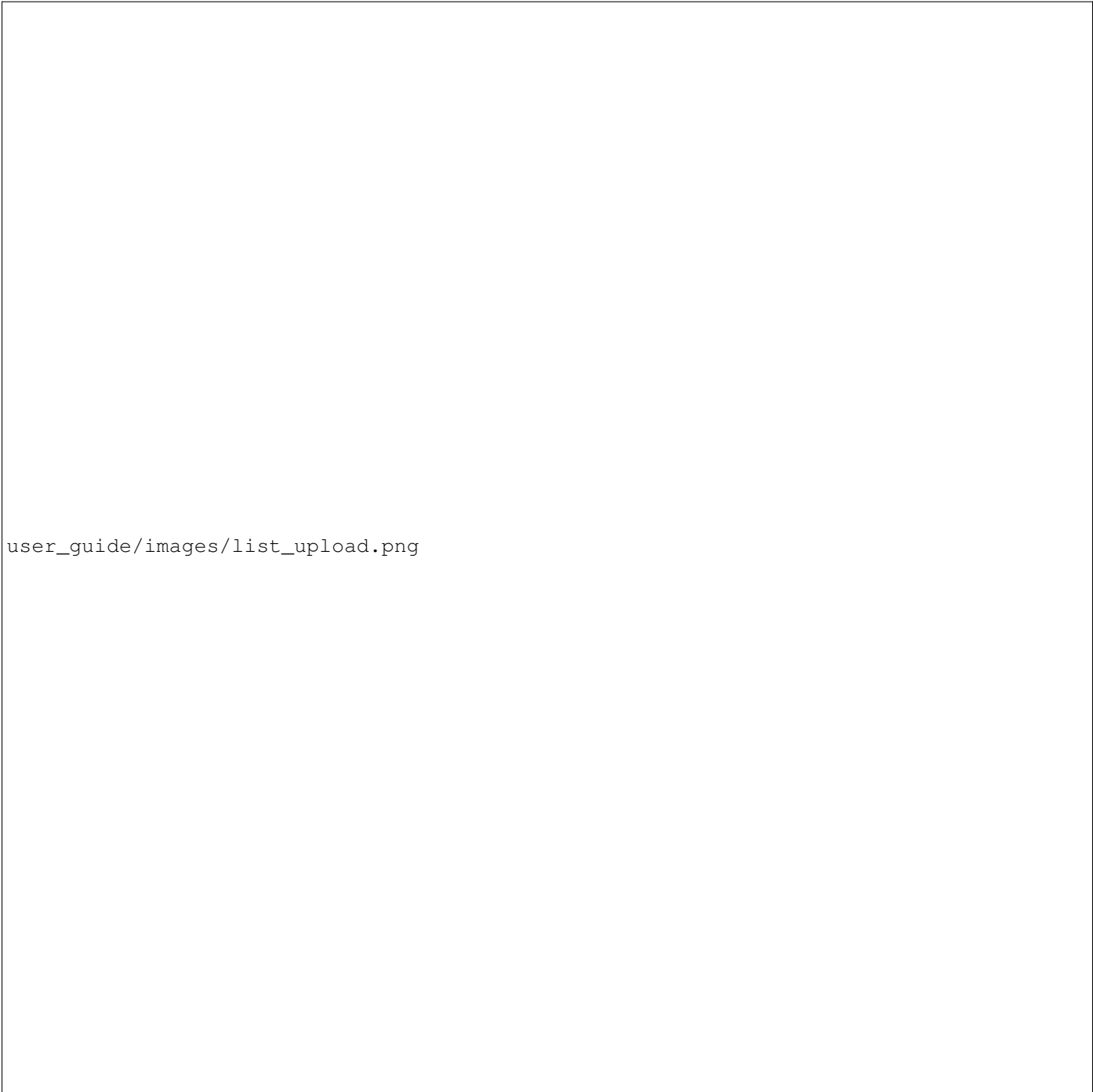
Fig. 5.1: List upload tool

user_guide/images/list_results.png

Fig. 5.2: Example: Search results for list of five identifiers
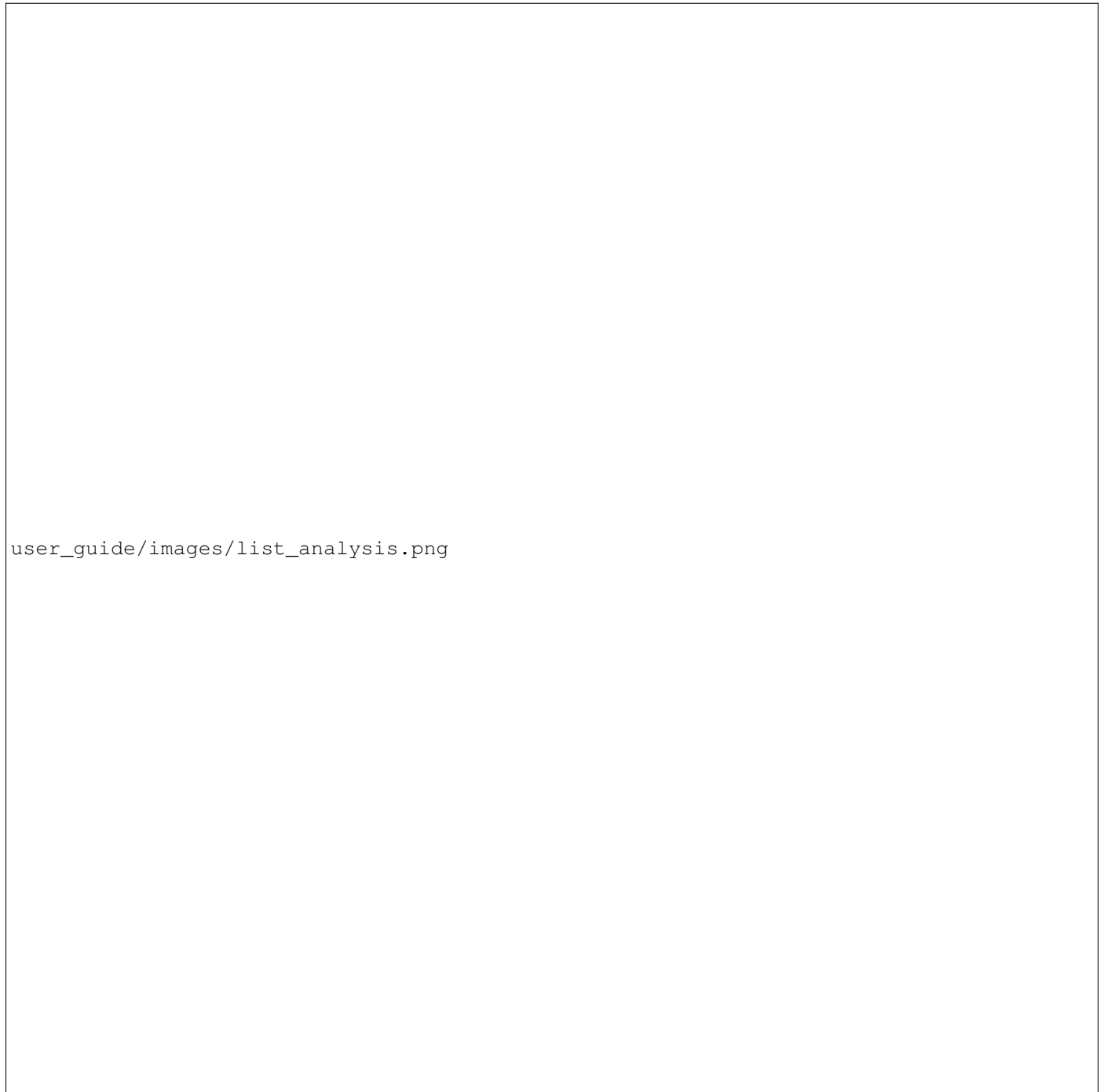
user_guide/images/list_analysis.png

Fig. 5.3: Example: List analysis for gene list

- Phenotypes (APO).

- Pathway Enrichment.

- Publication Enrichment.

The selection of widgets provided on the List Analysis page depend on the contents of the list.

5.2 Saving Lists

Saved lists appear under the **View** tab on the Lists page. For users who are not logged in, lists are saved temporarily; users must log in to save lists permanently. Saved lists may also be accessed from the **MyMine** menu item.

Predefined lists of all genes from different species are also available on the Lists page for all users.

```
user_guide/images/saved_lists.png
```

Fig. 5.4: Saved lists. Lists belonging to user are highlighted.

MyMine

MyMine is your personal InterMine account where you can manage your lists, queries, templates etc, share your lists with other users and create favourite templates and lists. You can access your MyMine account from the main tabs. The MyMine tab then has a series of subtabs for managing lists, templates, queries and your account details etc:

6.1 Create an account

Create an account through the Log in link:
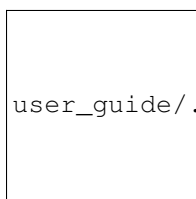
user_guide/../_images/createaccount.png

NOTE - any information saved in your account is private. It will not be accessible by other users and we will not inspect your saved data beyond automatic performance optimisation and updates.

6.2 Lists

The lists tab provides details of all the lists you have made. If you have lists that need upgrading, these will be shown first. Lists may need upgrading if some of the identifiers have become outdated between CandidaMine data releases. To upgrade a list click on the green arrow, this will take you to listconfirmationpage.

```
user_guide/../_images/myminelistupgrade.png
```

Lists are shown in alphabetical order and options are available to rename, mark as favourites, copy, delete and carry out set operations. .. image:: ../_images/myminelists.png

6.3 History

This tab displays any searches you have run during the current session. These are **not** saved permanently, but the history provides an option to permanently save a query - these will then be shown in the *Queries* tab of your MyMine account.

```
user_guide/../_images/myminehistory.png
```

6.4 Queries

Any queries you run or create can be saved permanently to your MyMine account. Queries can be saved from your *History* or from the query builder (see saveexport).

6.5 Templates

Any template searches you create yourself will be stored permanently here, with options to run, edit and export (as xml) as well as delete if the search is no longer required. Any query created using the query builder can be saved as a template as long as it has at least one constraint (see buildatemplate). You can also import template searches from xml - this is a useful way to share a template search you have created with colleagues. Template searches that you create yourself or share with colleagues are not made public. Templates that you have created yourself also appear under the main templates tab and are highlighted to indicate that they are your own rather than public templates (see buildatemplate).

6.6 Password

You can change your password here.

6.7 Account Details

The account details tab allows you set various aspects of your account as follows:

Inform me by email of newly shared lists: Do you want to receive an email if someone shares a list with you? (see listsshare for more details about list sharing).

Allow other users to share lists with me without confirmation: Do you want users to be able to share lists with you without asking first?

Display name: Set the name displayed in your InterMine interface.

Your preferred email address: Set the email address you prefer to use for correspondence - for example if someone shares a list with you. This could be different to the email you use to login to your account.

API access key:

API keys are used to access the features of the InterMine API without having to use your username or password.

For each new database release, all lists and queries are transferred to the new database release. Sometimes identifiers in lists become outdated and you will be asked to update your list (see listsupgrade). Occasionally we have to make changes to the underlying data model which make affect any queries you have saved. Please contact us (contact) if you would like any further information or help about such a query.

CHAPTER 7

Intermine API

CHAPTER 8

Technicals

CHAPTER 9

Data Sources

CHAPTER 10

Source code

CHAPTER 11

Issues