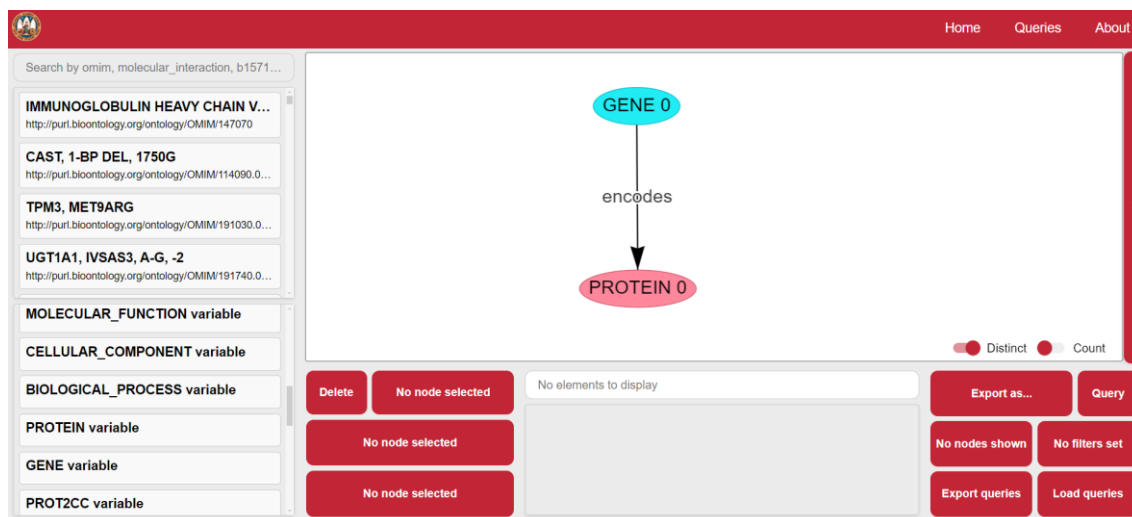


## 1. Introduction

INTUITION is a web application for user friendly SPARQL query building. In this way, users can exploit RDF knowledge graphs without advanced knowledge in SPARQL query language.

INTUITION analyses the knowledge network of an accessible endpoint, in this case BioGateway (<http://ssb4.nt.ntnu.no:23122/sparql>), and allows building queries graphically by representing nodes (entities) and edges (properties).

RDF knowledge graphs represent entities through uniform resource identifiers (URIs), and information through triples or statements that represent a directional relationship between two entities, similar to a sentence: <subject> <predicate> <object>. For example: <Gene> <encodes> <Protein>. The SPARQL query language also uses this pattern to build queries. These queries can be complex by linking multiple triples and including operations. A tutorial for building SPARQL queries in BioGateway (BGW) is available in this repository (<https://github.com/juan-mulero/cisreg>). INTUITION uses this same design for query development (Figure 1).



## 2. Design

In INTUITION we distinguish different sections (Figure 2):

- A- Query building screen.
- B- Entity browser (Currently deactivated).
- C- Graph builder (Used for queries that use union clauses).
- D- Variable selection (Main types of entities in the network).
- E- Properties selection:
  - Object properties: relations between entities. Their inclusion acts as a filter.
  - Data properties: attributes of the entities. Their inclusion acts as a filter.
  - Optional properties: relations between entities. Their inclusion does not act as a filter (optional).
- F- Output screen.
- G- Query builder:

- Query: runs the query.
- Export as: exports the query results.
- Nodes shown: output editor. Allow to indicate which variables are shown in the output screen, and to create bindings, i.e., to define variables.
- Filters set: allow adding filters.
- Export query: export the designed query.
- Load query: to load a previously designed query.

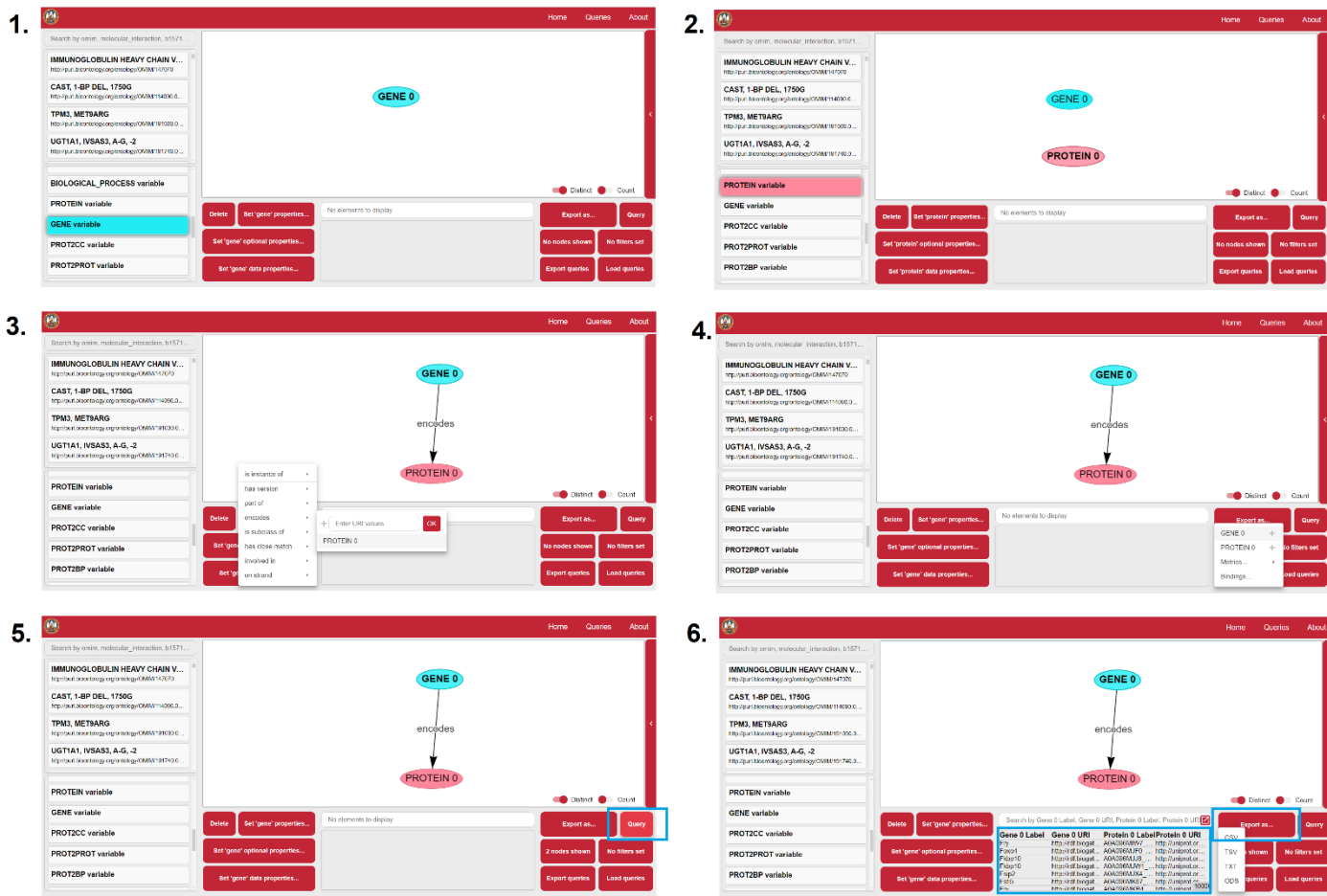
The screenshot shows a query builder interface with several panels:

- Panel B:** Search results for 'IMMUNOGLOBULIN HEAVY CHAIN V...', 'CAST, 1-BP DEL, 1750G', 'TPM3, MET9ARG', and 'UGT1A1, IVSAS3, A-G, -2'.
- Panel A:** A graph showing 'GENE 0' encoding 'PROTEIN 0'.
- Panel C:** A sidebar with 'Define new graph' and a 'Default' button.
- Panel D:** A list of variables: 'MOLECULAR\_FUNCTION variable', 'CELLULAR\_COMPONENT variable', 'BIOLOGICAL\_PROCESS variable', 'PROTEIN variable', 'GENE variable', and 'PROT2CC variable'.
- Panel E:** Action buttons: 'Delete', 'Set 'gene' properties...', 'Set 'gene' optional properties...', and 'Set 'gene' data properties...'.
- Panel F:** A table of results with columns: 'Gene 0 Label', 'Gene 0 URI', 'Protein 0 Label', and 'Protein 0 URI'. The table contains 10 rows of data.
- Panel G:** Action buttons: 'Export as...', 'Query', '2 nodes shown', 'No filters set', 'Export queries', and 'Load queries'.

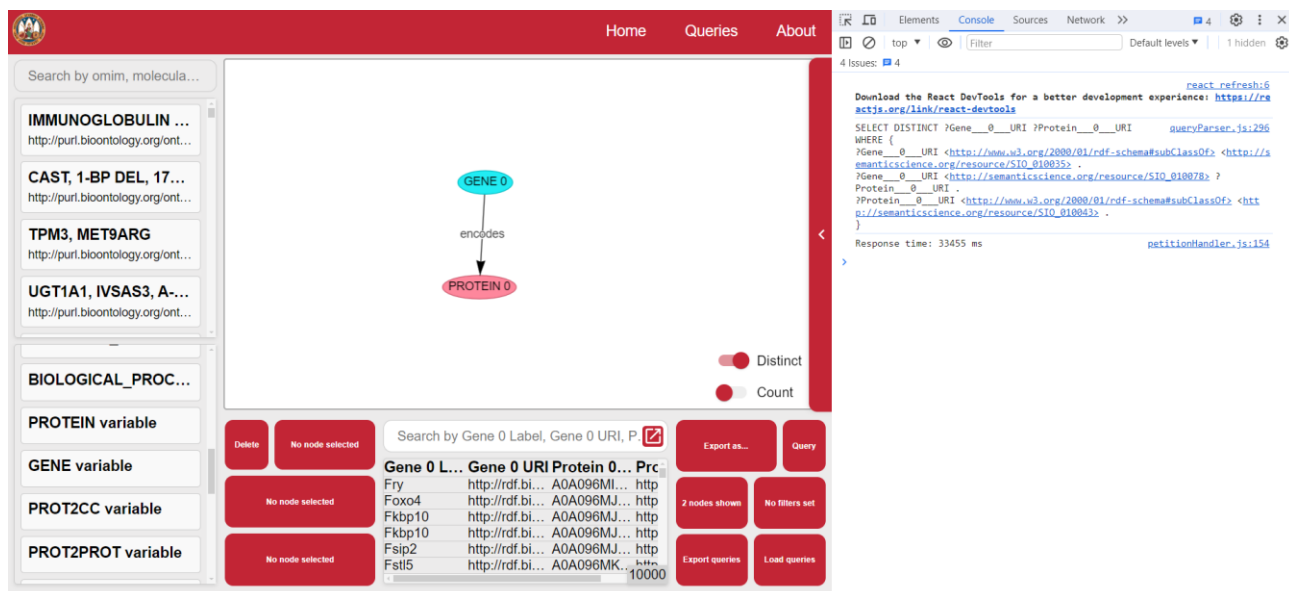
### 3. How to build a query

We take as an example the previous case: <Gene> <encodes> <Protein> (Figure 3).

1. Select the first node or subject, in this case Gene, in the variables section.
2. Select the second node or object, in this case Protein, in the variables section.
3. Select the object property that relates both entities, in this case "encodes".
4. Select in "Nodes shown" the data we want to show in the output, in this case Gene and Protein.
5. Click on "Query" to launch the query.
6. Click on "Export as" to download the data obtained. Click on "Export query" if you want to save the query as well.



The generated SPARQL query can also be found by accessing the Console (Figure 4).



## 4. Filters and other clauses

### 4.1. DISTINCT and COUNT

By default, the queries include the DISTINCT clause to avoid duplicate results. This can be deactivated clicking on the Distinct button, in the Query building screen. You can also activate the Count button if you want to include this clause in the query (Figure 5).

The screenshot shows the query building interface. On the left, there is a search bar and a list of variables. The main area displays a query graph with 'GENE 0' encoding 'PROTEIN 0'. Below the graph, there is a table with columns 'Gene 0 URI count' and 'Protein 0 URI count'. The 'Distinct' button is highlighted, and the 'Count' button is also visible. The interface includes a search bar, a list of variables, and a table of results.

Gene 0 URI count	Protein 0 URI count
255461	525012

### 4.2. VALUES

This clause permits the assignment of one or more values to a specific variable. For example, cis regulatory modules identified in two tissues of interest. First we indicate the subject node (CRM variable), then we include in the corresponding object property (observed in) the values to be included. Click on "+" to include values and click on OK when all values are listed (Figure 6).

The screenshot shows the query building interface. On the left, there is a search bar and a list of variables. The main area displays a query graph with 'CRM 0' observed in. A dropdown menu is open, showing the 'observed in' property. The 'Values' button is highlighted, and the 'OK' button is also visible. The interface includes a search bar, a list of variables, and a table of results.

CRM 0 URI count
No targets available

### 4.3. Filters of properties

Clicking on "data properties" we can also filter attributes. If the value is a string we can use the operator "=" to indicate an exact value, or the operator "⊆" to indicate a substring contained in the string. If the value is numeric we can use the operators =, >, ≥, <, ≤. For example, we filter the variables chromosome and CRM to obtain the enhancers that overlap at certain coordinates (Figure 7).

The screenshot displays the INTUITION web interface. At the top, there's a navigation bar with 'Home', 'Queries', and 'About' links. Below it, a search bar contains the text 'Search by omim, molecular\_interaction, b1571...'. A list of search results is shown on the left, including 'IMMUNOGLOBULIN HEAVY CHAIN V...', 'CAST, 1-BP DEL, 1750G', 'TPM3, MET9ARG', and 'UGT1A1, IVSAS3, A-G, -2'. The main area shows a graph with two nodes: 'CHROMOSOME 0' (yellow) and 'CRM 0' (blue), connected by a 'part of' relationship. Below the graph, there are buttons for 'Delete', 'No node selected', and 'Search by Crm 0 Label, Crm 0 URI'. A table lists 'Crm 0 Label' and 'Crm 0 URI' with various identifiers. On the right, there are buttons for 'Export as...', 'Query', '1 nodes shown', 'No filters set', 'Export queries', and 'Load queries'. Below the main interface, two dialog boxes are shown. The first is titled 'Node 'CHROMOSOME 0' data properties' and contains fields for 'is instance of', 'has name', and 'category', each with a dropdown menu and a 'show as...' button. The second dialog box is titled 'Node 'CRM 0' data properties' and contains fields for 'in taxon', 'is instance of', 'involved in pos...', 'has definition', 'start position', 'has name', 'end position', and 'is defined by', each with a dropdown menu and a 'show as...' button. Both dialog boxes have 'Set properties' and 'Cancel' buttons.

Variables can also be filtered clicking on "Filters set" button.

### 4.4. Creating and filtering variables

INTUITION supports the generation and filtering of new variables. This functionality is implemented in "Nodes shown" > "Bindings" button. For example, by subtracting the end and start positions of the CRMs we obtain the length of the sequences in a new variable. Then, we can filter this new variable in the "Filters set" button (Figure 8).

Search by omim, molecular\_interaction, b1571...

IMMUNOGLOBULIN HEAVY CHAIN V...  
http://purl.bioontology.org/ontology/OMIM/147070

CAST, 1-BP DEL, 1750G  
http://purl.bioontology.org/ontology/OMIM/114090.0...

TPM3, MET9ARG  
http://purl.bioontology.org/ontology/OMIM/191030.0...

UGT1A1, IVSAS3, A-G, -2  
http://purl.bioontology.org/ontology/OMIM/191740.0...

DATABASE variable

CHROMOSOME variable

REFERENCE\_GENOME variable

TAD variable

CRM2PHEN variable

CHROMOSOME 0

part of

CRM 0

Distinct Count

Delete Set 'crm' properties...

Search by Crm 0 Label, Crm 0 URI, Chromosome 0 Label, Chromo...

Export as... Query

Chromos...	Chromos...	Start pos...	End posit...	Length
chr-16	https://www...	143029	143282	253
chr-16	https://www...	149322	149455	133
chr-16	https://www...	142866	147016	4150
chr-16	https://www...	158710	160710	2000
chr-16	https://www...	162658	163864	1206
chr-16	https://www...	145000	145400	400

2 nodes shown 1 filter set

Export queries Load queries

Bindings and Variables

length End position CRM 0 - Start position CRM 0 Show in results Absolute

Variable length results from Start position CRM 0 Start position CRM 0 Show in results Absolute Add binding

Set bindings Cancel

Filters

length <= 5000

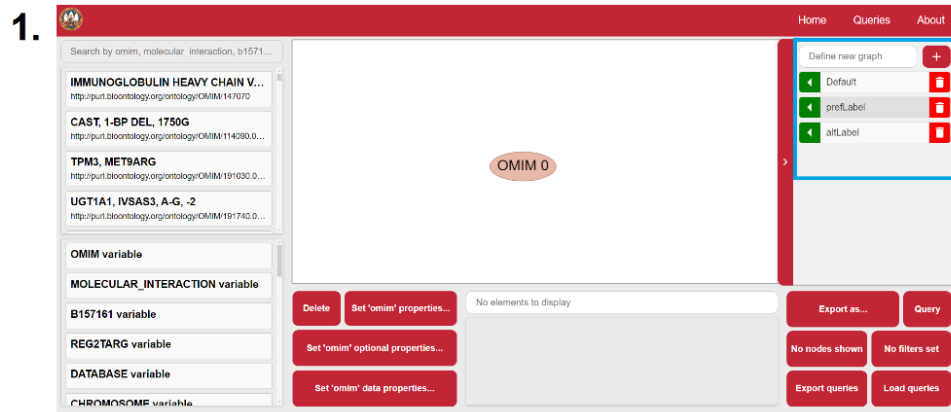
Filter length <= 5000 Add

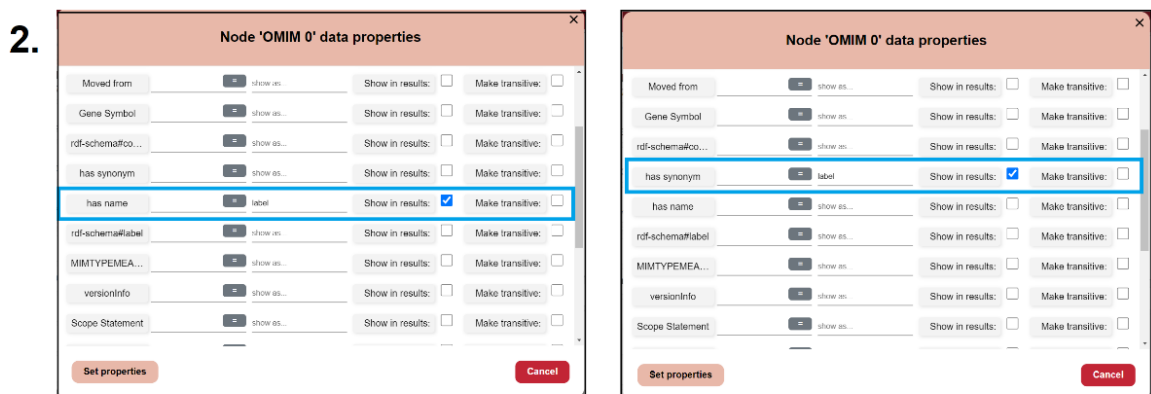
Set Filters Cancel

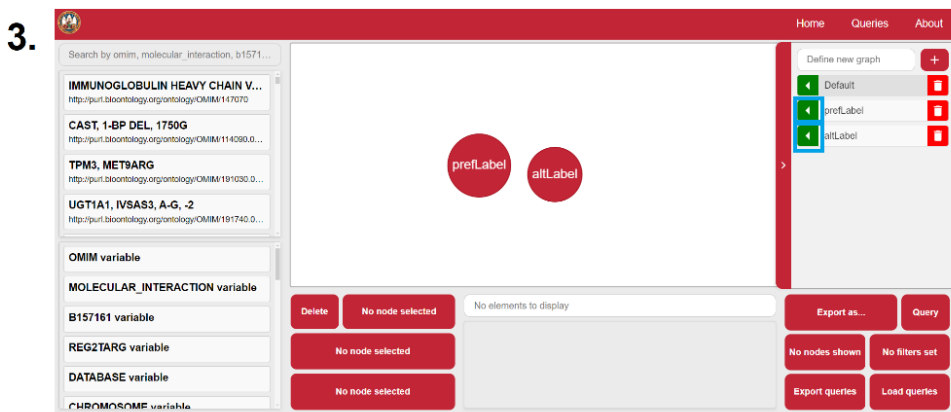
## 4.5. UNION

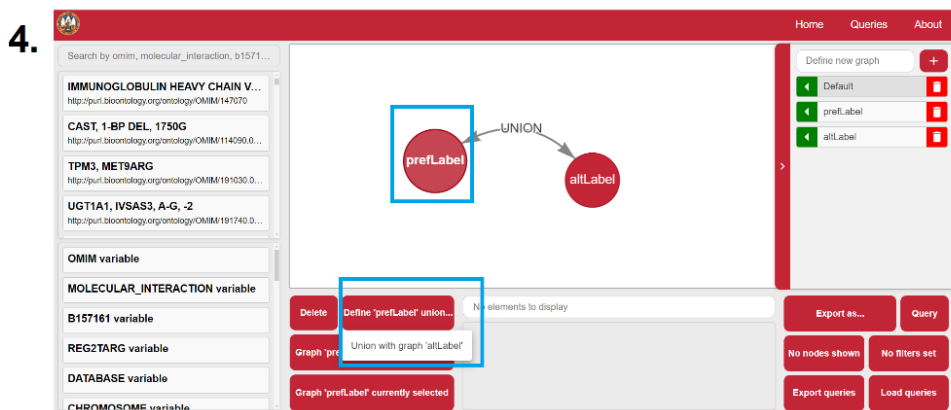
INTUITION also allows the use of the UNION clause of SPARQL. UNION merges subqueries through common variables in both queries. We illustrate its use through a use case. For example, we retrieve the OMIM entities that contain the string "breast cancer" as a name or synonym. To do that:

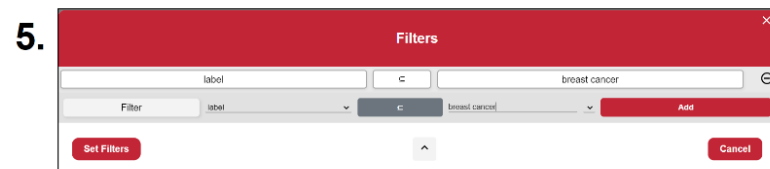
1. In the Graph builder section, we create the graphs belonging to each of the subqueries, and we include an OMIM node in each of them. In the example, the graphs are "prefLabel" for the main label query and "altLabel" for the query of the synonym.
2. In each of the graphs we define the variable "label" according to the appropriate dataproperties ("has name" and "has synonym" properties, respectively). For this, we use the "show as" functionality, which enables to rename the variables, in this case under the common variable called "label".
3. We return to the main graph where we will join the two subqueries. For this, we include the subgraphs clicking on the green flap of each of the subgraphs.
4. Select one of the subgraphs represented as nodes and click on "Define union".
5. In "Node shown" we select the variables to be shown and in "Filters set" we filter the variable "label".
6. Run the query (Query).

1. 

2. 

3. 

4. 

5. 

6.

Search by omim, molecular\_interaction, b1571...

IMMUNOGLOBULIN HEAVY CHAIN V...  
http://purl.bioontology.org/ontology/OMIM/147370

CAST, 1-BP DEL, 1750G  
http://purl.bioontology.org/ontology/OMIM/114090.0...

TPM3, MET9ARG  
http://purl.bioontology.org/ontology/OMIM/191030.0...

UGT1A1, IVSAS3, A-G, -2  
http://purl.bioontology.org/ontology/OMIM/191740.0...

OMIM variable

MOLECULAR\_INTERACTION variable

B157161 variable

REG2TARG variable

DATABASE variable

CHROMOSOME variable

Define new graph +

Default

prefLabel

altLabel

Delete Define 'prefLabel' union...

Graph 'prefLabel' currently selected

Graph 'prefLabel' currently selected

Search by Omim 0 Label, Omim 0 URI, Label

Omim 0 Label	Omim 0 URI	Label
BREAST CANCER A...	http://purl.bioontology.org/ontology/OMIM/147370	BREAST CANCER A...
Breast cancer, lobular	http://purl.bioontology.org/ontology/OMIM/147370	Breast cancer, lobular
Breast cancer, early-onset	http://purl.bioontology.org/ontology/OMIM/147370	Breast cancer, early-onset
DIFFUSE GASTRIC ...	http://purl.bioontology.org/ontology/OMIM/147370	DIFFUSE GASTRIC ...
Paraneoplastic SPS I...	http://purl.bioontology.org/ontology/OMIM/147370	Paraneoplastic SPS I...
BREAST CANCER	http://purl.bioontology.org/ontology/OMIM/147370	BREAST CANCER
BREAST CANCER	http://purl.bioontology.org/ontology/OMIM/147370	BREAST CANCER

Export as... Query

OMIM 0 1 filter set

Metrics...

Load queries

## 5. Use Cases

The following Use Cases were developed in the paper "*Analysis of the landscape of human enhancer sequences in biological databases*". The corresponding queries are attached for reproducibility and as examples of use.

1. Use case 1: [Attach json files]
2. Use case 2: [Attach json files]
3. Use case 3: [Attach json files]