

# INTUITION Tutorial for BioGateway

1. Introduction .....	1
2. Design.....	1
3. Variables and properties.....	2
4. How to build a query in 6 steps .....	4
5. Data filtering and other possible operations .....	7
5.1. Filtering entities by their relations and attributes.....	7
5.2. Count and display unique results .....	14
5.3. Optional relations .....	18
5.4. Multiple values.....	19
5.5. Creating and filtering variables .....	21
5.6. Union of queries .....	23
6. Use Cases.....	25

## 1. Introduction

INTUITION (<http://semantics.inf.um.es/intuition>) is a web application for user-friendly SPARQL query building. In this way, users can exploit RDF knowledge graphs without knowledge in SPARQL query language.

INTUITION analyses the knowledge network of an accessible endpoint, in this case, the current instance of BioGateway (<http://ssb4.nt.ntnu.no:23122/sparql>), and allows building biological queries graphically by defining search patterns: biological entities (nodes) that can be specified in detail through variables and are related through properties (edges).

In a graph, nodes represent different types of biological entities, such as genes, proteins or CRMs, and edges (or properties) are used to specify different types of relations that exist between two nodes (for example, <Gene> <encoding> <Protein>). Some properties are also used to add attributes to entities.

## 2. Design

In INTUITION we distinguish different sections:

- A- Query building canvas.
- B- Union builder (Used for queries that use union clauses).
- C- Variable browser (Main types of entities in the network).
- D- Pattern designer (nodes and links):

- Set attributes: Allows a user to edit the intrinsic properties of a variable, that distinguish one entity from another. For example, the name or description of a gene. Therefore, its inclusion acts as a filter because it permits to specify the entity or entities with a certain pattern of characteristics.

- Add relations: Allows a user to include relations between entities to build queries of greater biological relevance. The inclusion of relations implies the insertion of a search pattern, so its use also selects/filters the knowledge network.

- Add optional relations: Allows a user to include optional relations between entities. Their use does not act as a filter, but adds information when the pattern is met.

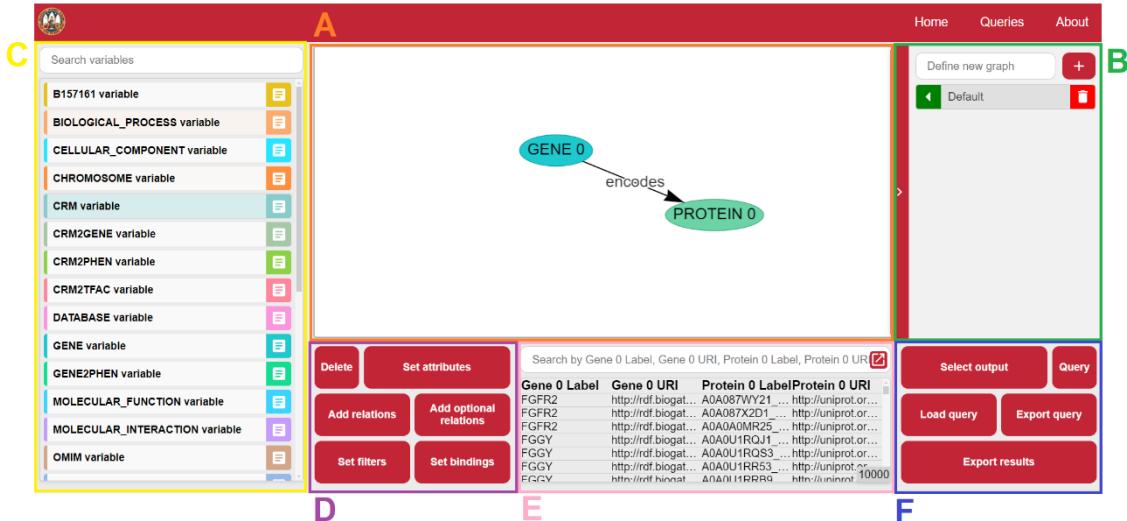
- Set bindings: Enables the creation of custom variables using attributes included in the search pattern, renamed, or selected for inclusion in the output.

- Set filters. Button to specify filters on attributes used in the search pattern, renamed, or selected to be included in the output. It can also be used to apply filters on new variables (bindings).

E- Output display.

F- Query builder:

- Select output: output selector. Allow to indicate which variables are shown in the output screen.
- Query: runs the query.
- Load query: to load a previously designed query.
- Export query: export the designed query.
- Export results: exports the query results.



### 3. Variables and properties

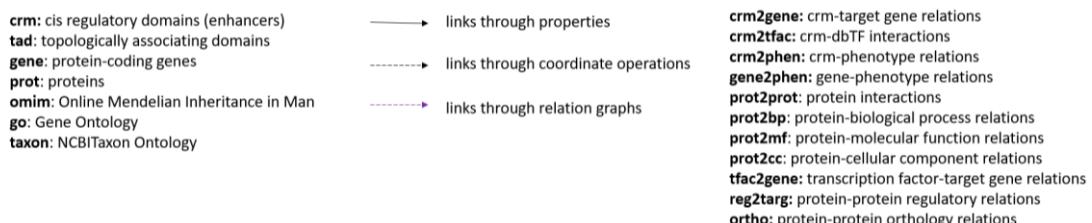
Variables correspond to biological entities that we can use to develop query patterns. We can group these variables into 3 groups:

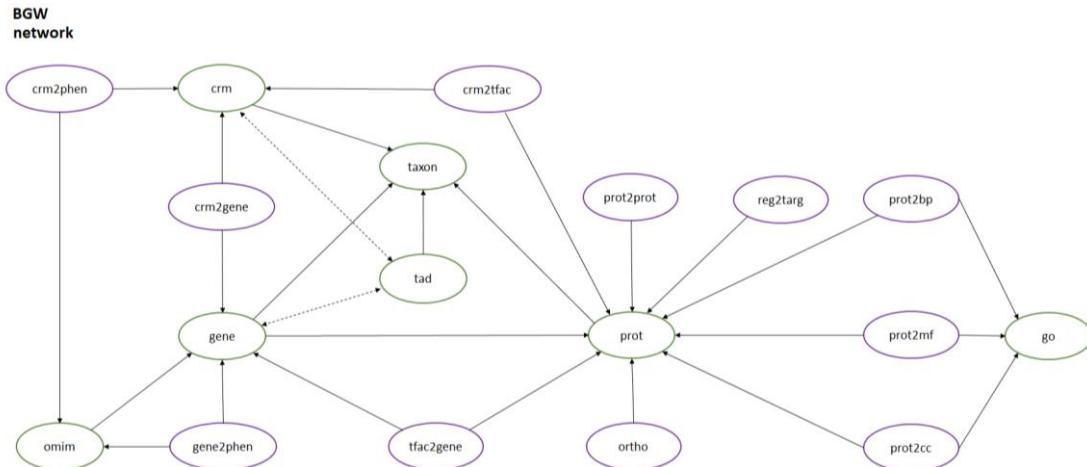
1- Variables corresponding to biological entities modelled by BioGateway:

- Gene variable: protein-coding genes.
- Protein variable: proteins.
- CRM variable: cis-regulatory module (currently only enhancers).

- TAD variable: topologically associated domain.
  - Database variable: databases.
  - Chromosome variable: chromosomes.
  - Reference\_genome variable: genome assembly.
  - Transcription factor variable: transcription factors (currently only proteins that interact with CRM).
- 2- Variables corresponding to biological entities imported from external ontologies:
- OMIM variable: entities from OMIM ontology (mainly phenotypes).
  - Molecular\_interaction: entities from Molecular Interactions ontology (MI).
  - Cellular\_component variable: cellular components from Gene Ontology (GO).
  - Molecular\_function variable: molecular functions from GO.
  - Biological\_process variable: biological processes from GO.
  - Root variable: top hierarchically class of NCBI Taxon Ontology.
  - Taxonomic\_rank variable: top hierarchically class of NCBI Taxon Ontology.
- 3- Variables, modelled by BioGateway, corresponding to relations between biological entities:
- crm2gene variable: relation between CRM and gene.
  - crm2phen variable: relation between CRM and phenotype.
  - crm2tfac variable: relation between CRM and protein (transcription factor).
  - gene2phen variable: relation between gene and phenotype.
  - tfac2gene variable: relation between gene and protein.
  - prot2prot: molecular interaction relation between proteins.
  - reg2targ variable: regulatory relation between proteins.
  - Ortho variable: orthology relation between proteins.
  - prot2cc variable: relation between protein and its cellular components.
  - prot2mf: relation between protein and its molecular functions.
  - prot2bp variable: relation between protein and its biological processes.

Properties are used to semantically relate different biological entities, and to provide attributes to these entities. These entities are detailed with examples and their domains [here](#). It also includes information about the vocabularies used.





#### 4. How to build a query in 6 steps

The query building process involves linking entities (nodes) with their attributes and/or other entities through properties (edges). We take as an example the previous case, the query: *Which proteins do the different genes encode?* (<Gene> <encodes> <Protein>).

1. Select the first entity (subject node), in this case, Gene, in the Variable browser.
2. Select the second entity (object node), in this case, Protein, in the Variable browser.
3. Select the relation between both entities in the Properties selectors, in this case, "encodes".
4. Select in "Nodes shown" the data you want to show in the output (click on "+").
5. Click on "Query" to launch the query.
6. Click on "Export as" to download the data. Click on "Export query" to save the query.

1.

2.

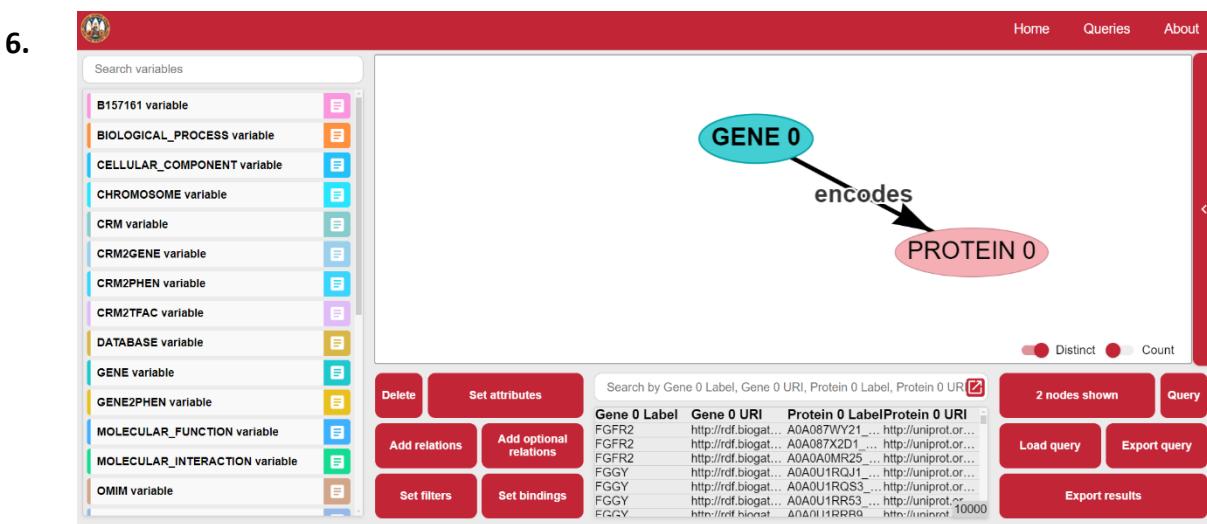
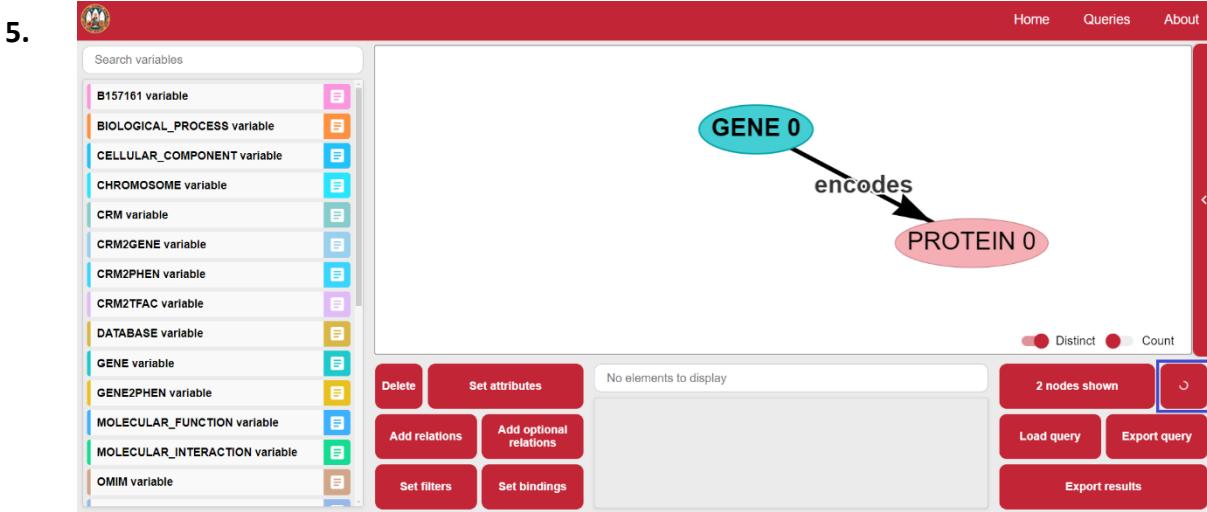
The screenshot shows the Bio2rdf SPARQL query builder interface. The search bar at the top contains the query: `SELECT ?s WHERE { ?s a gene }`. The results pane on the right shows a single result: `GENE 0`. Below the results is a context menu with the following options: `is instance of`, `encodes`, `has version`, `part of`, `is subclass of`, `involved in`, `on strand`, and `has close match`. The `encodes` option is highlighted with a blue box. The bottom right corner of the interface has buttons for `Select output`, `Query`, `Load query`, `Export query`, and `Export results`. The interface is styled with a red header and a light gray background.

3.

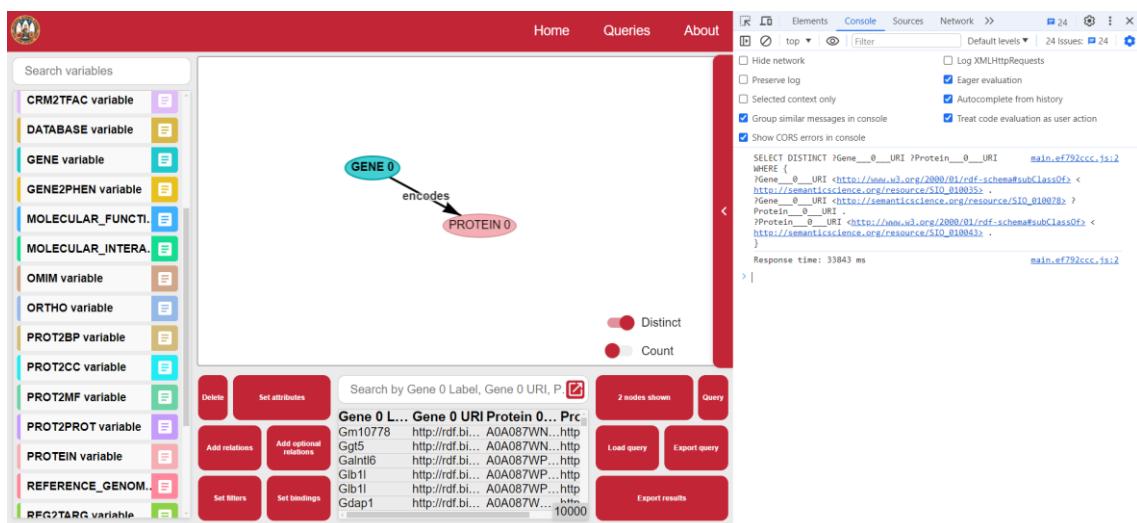
The screenshot shows the Bio2rdf SPARQL query builder interface. The search bar at the top contains the query: `SELECT ?s WHERE { ?s a gene }`. The results pane on the right shows a single result: `GENE 0`. Below the results is a context menu with the following options: `is instance of`, `encodes`, `has version`, `part of`, `is subclass of`, `involved in`, `on strand`, and `has close match`. The `encodes` option is highlighted with a blue box. A modal dialog box is open, prompting for a URI value, with the text `PROTEIN 0` entered. The bottom right corner of the interface has buttons for `Select output`, `Query`, `Load query`, `Export query`, and `Export results`. The interface is styled with a red header and a light gray background.

4.

The screenshot shows the Bio2rdf SPARQL query builder interface. The search bar at the top contains the query: `SELECT ?s WHERE { ?s a gene }`. The results pane on the right shows a single result: `GENE 0`. Below the results is a context menu with the following options: `Delete`, `Set attributes`, `Add relations`, `Add optional relations`, `Set filters`, and `Set bindings`. The `Add relations` button is highlighted with a blue box. The bottom right corner of the interface has buttons for `Select output`, `Query`, `Load query`, `Export query`, and `Export results`. The interface is styled with a red header and a light gray background.



The generated SPARQL query can also be found by accessing the Console.



## 5. Data filtering and other possible operations

### 5.1. Filtering entities by their relations and attributes

Linking two biological entities (or variables) by their relation (properties) is the simplest way to create a search pattern. A search pattern selects the desired information from the knowledge network. That is, an information filter is applied. However, any biological entity can also be selected by its characteristics or attributes. For example, genes can be selected by their names. Below we illustrate a use case that extends the previous query to: *Which proteins are encoded by the TOX3 gene?* To do this, we include the name of the gene in the attributes of the 'Gene' node (click on the corresponding node and then on the 'Set attributes' button).

1.

Search variables

- CRM2TFAC variable
- DATABASE variable
- GENE variable
- GENE2PHEN variable
- MOLECULAR\_FUNCTION variable
- MOLECULAR\_INTERACTION variable
- OMIM variable
- ORTHO variable
- PROT2BP variable
- PROT2CC variable
- PROT2MF variable
- PROT2PROT variable
- PROTEIN variable
- REFERENCE\_GENOME variable
- REFG2TARG variable

Home    Queries    About

Distinct    Count

Set attributes

No elements to display

Select output    Query

Add relations    Add optional relations

Set filters    Set bindings

Load query    Export query

Export results

2.

Node 'GENE 0' data properties

is instance of show as... Show in results:  Make transitive:

part of show as... Show in results:  Make transitive:

in taxon show as... Show in results:  Make transitive:

end position show as... Show in results:  Make transitive:

start position show as... Show in results:  Make transitive:

has definition show as... Show in results:  Make transitive:

has synonym show as... Show in results:  Make transitive:

has name TOX3 show as... Show in results:  Make transitive:

Set properties    Cancel

The node definition by the subClassOf property will be included

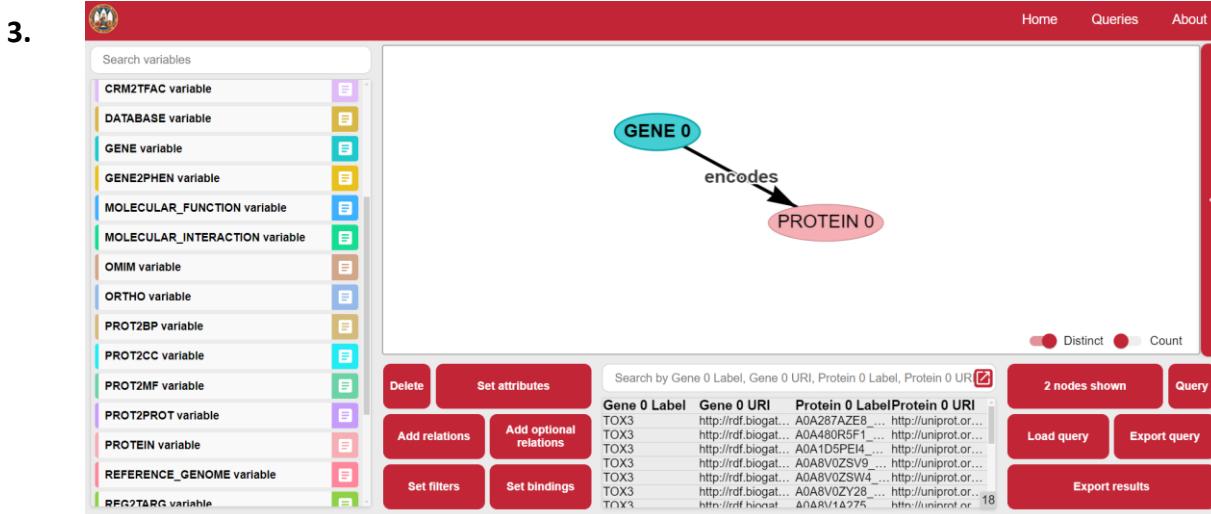
Home    Queries    About

Distinct    Count

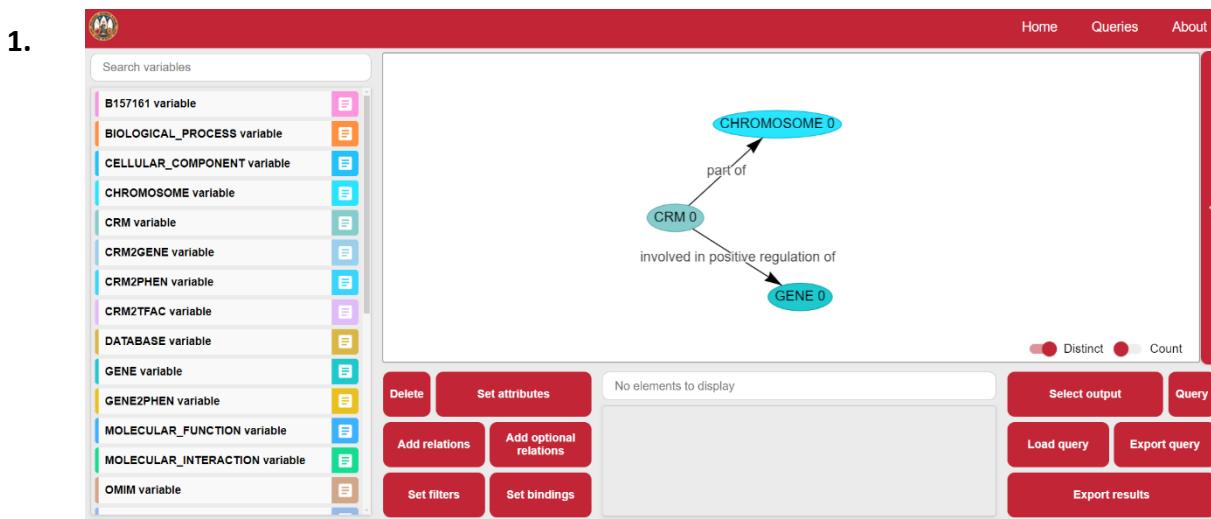
Select output    Query

Load query    Export query

Export results



By defining the desired characteristics of biological entities (by clicking on "Set attributes") we can select entities based on these characteristics (attributes). If the character is defined as "string" composed of letters and/or numbers we can use the operator '=' to find only exact strings, or the operator ' $\subseteq$ ' to find substrings contained in a larger string. If the character is only numeric, we can find results equal to, larger, or smaller than, using the operators '=', '>', '>=', '<', '<=''. To change the operators just click on the default operator. For example, we can query: *Which genes are regulated by enhancers that overlap with the chr16:52565276 mutation?* i.e. CRM sequences that positively regulate gene expression. To do this, we create the relations: <CRM> <part of> <Chromosome>, and <CRM> <involved in positive regulation of> <Gene>. Then we add attributes to the Chromosome (chromosome name) and CRM (sequence coordinates) nodes. Then we select the data output (Genes) and run the query.



2.

3.

4.

Variables can also be filtered clicking on “Set filters” button.

Some additional examples are given below:

- **Example 1:** Filtering by taxon.

The resources in RDF are represented by Uniform Resource Identifiers (URIs), so these must be used when filtering a resource. This is the case for taxon (*in taxon* property, in "Set attributes"). Because URIs can be tedious to work with, e.g. "[http://purl.obolibrary.org/obo/NCBITaxon\\_9606](http://purl.obolibrary.org/obo/NCBITaxon_9606)", the "content in" or " $\subseteq$ " operator makes it easier to work with identifiers only. Below we include a table with the taxonomic IDs of the most relevant species included in BioGateway:

label	Taxon ID	URI_taxon
<i>Mus musculus</i>	10090	<a href="http://purl.obolibrary.org/obo/NCBITaxon_10090">http://purl.obolibrary.org/obo/NCBITaxon_10090</a>
<i>Arabidopsis thaliana</i>	3702	<a href="http://purl.obolibrary.org/obo/NCBITaxon_3702">http://purl.obolibrary.org/obo/NCBITaxon_3702</a>
<i>Oryza sativa Japonica Group</i>	39947	<a href="http://purl.obolibrary.org/obo/NCBITaxon_39947">http://purl.obolibrary.org/obo/NCBITaxon_39947</a>
<i>Dictyostelium discoideum</i>	44689	<a href="http://purl.obolibrary.org/obo/NCBITaxon_44689">http://purl.obolibrary.org/obo/NCBITaxon_44689</a>
<i>Zea mays</i>	4577	<a href="http://purl.obolibrary.org/obo/NCBITaxon_4577">http://purl.obolibrary.org/obo/NCBITaxon_4577</a>
<i>Caenorhabditis elegans</i>	6239	<a href="http://purl.obolibrary.org/obo/NCBITaxon_6239">http://purl.obolibrary.org/obo/NCBITaxon_6239</a>
<i>Danio rerio</i>	7955	<a href="http://purl.obolibrary.org/obo/NCBITaxon_7955">http://purl.obolibrary.org/obo/NCBITaxon_7955</a>
<i>Gallus gallus</i>	9031	<a href="http://purl.obolibrary.org/obo/NCBITaxon_9031">http://purl.obolibrary.org/obo/NCBITaxon_9031</a>
<i>Sus scrofa</i>	9823	<a href="http://purl.obolibrary.org/obo/NCBITaxon_9823">http://purl.obolibrary.org/obo/NCBITaxon_9823</a>
<i>Bos taurus</i>	9913	<a href="http://purl.obolibrary.org/obo/NCBITaxon_9913">http://purl.obolibrary.org/obo/NCBITaxon_9913</a>
<i>Homo sapiens</i>	9606	<a href="http://purl.obolibrary.org/obo/NCBITaxon_9606">http://purl.obolibrary.org/obo/NCBITaxon_9606</a>
<i>Drosophila melanogaster</i>	7227	<a href="http://purl.obolibrary.org/obo/NCBITaxon_7227">http://purl.obolibrary.org/obo/NCBITaxon_7227</a>
<i>Oryctolagus cuniculus</i>	9986	<a href="http://purl.obolibrary.org/obo/NCBITaxon_9986">http://purl.obolibrary.org/obo/NCBITaxon_9986</a>
<i>Rattus norvegicus</i>	10116	<a href="http://purl.obolibrary.org/obo/NCBITaxon_10116">http://purl.obolibrary.org/obo/NCBITaxon_10116</a>
<i>Saccharomyces cerevisiae</i> S288C	559292	<a href="http://purl.obolibrary.org/obo/NCBITaxon_559292">http://purl.obolibrary.org/obo/NCBITaxon_559292</a>
<i>Schizosaccharomyces pombe</i> 972h-	284812	<a href="http://purl.obolibrary.org/obo/NCBITaxon_284812">http://purl.obolibrary.org/obo/NCBITaxon_284812</a>
<i>Chlamydomonas reinhardtii</i>	3055	<a href="http://purl.obolibrary.org/obo/NCBITaxon_3055">http://purl.obolibrary.org/obo/NCBITaxon_3055</a>
<i>Plasmodium falciparum</i> 3D7	36329	<a href="http://purl.obolibrary.org/obo/NCBITaxon_36329">http://purl.obolibrary.org/obo/NCBITaxon_36329</a>
<i>Neurospora crassa</i> OR74A	367110	<a href="http://purl.obolibrary.org/obo/NCBITaxon_367110">http://purl.obolibrary.org/obo/NCBITaxon_367110</a>
<i>Canis lupus familiaris</i>	9615	<a href="http://purl.obolibrary.org/obo/NCBITaxon_9615">http://purl.obolibrary.org/obo/NCBITaxon_9615</a>

The following example illustrates the filtering of human genes using the taxon ID (9606):  
*What human genes does the network contain?*

1.

The screenshot shows the BioGateway interface. At the top, there is a navigation bar with 'Home', 'Queries', and 'About' buttons. Below the navigation bar, there is a sidebar on the left containing a 'Search variables' section and a list of variable types. The 'GENE variable' is selected. The main area of the interface shows a network graph with a central node labeled 'GENE 0'. Below the graph, there are several buttons: 'Delete', 'Set attributes' (which is highlighted in red), 'Add relations', 'Add optional relations', 'Set filters', and 'Set bindings'. To the right of the graph, there is a message 'No elements to display'. At the bottom right, there are buttons for 'Select output', 'Query', 'Load query', 'Export query', and 'Export results'. There are also 'Distinct' and 'Count' radio buttons.

2.

Search variables

B151616 variable

**BIOLOGICAL\_PROCESS** variable

CELLULAR\_COMPONENT variable

CHROMOSOME variable

CRM variable

CRM2GENE variable

CRM2PHEN variable

CRM2TFAC variable

**DATABASE** variable

GENE variable

GENE2PHEN variable

MOLECULAR\_FUNCTION variable

MOLECULAR\_INTERACTION variable

OMIM variable

Node 'GENE 0' data properties

is instance of  show as... Show in results:  Make transitive:

part of  show as... Show in results:  Make transitive:

**in taxon** 9606  show as... Show in results:  Make transitive:

end position  show as... Show in results:  Make transitive:

start position  show as... Show in results:  Make transitive:

has definition  show as... Show in results:  Make transitive:

has synonym  show as... Show in results:  Make transitive:

has name  show as... Show in results:  Make transitive:

**Set properties**   The node definition by the subClassOf property will be included

Distinct Count

Select output Query

Load query Export query

Export results

### - **Example 2:** Filtering by chromosome.

Chromosomes are entities available as variables, and are resources that have labels. Therefore, chromosomes can be filtered through their labels. Strings can be filtered in INTUITION using the "`=`" (exact value) or "`⊆`" (contained in) operators. The default configuration of string filtering is not case-sensitive. The following example filters human genes on chromosome 1 (*What human genes are located on chr-1?*).

1.

```
graph TD; CHROMOSOME0((CHROMOSOME 0)) -- part of --> GENE0((GENE 0))
```

Search variables

- B157161 variable
- BILOGICAL\_PROCESS variable
- CELLULAR\_COMPONENT variable
- CHROMOSOME variable
- CRM variable
- CRM2GENE variable
- CRM2PHEN variable
- CRM2TFAC variable
- DATABASE variable
- GENE variable
- GENE2PHEN variable
- MOLECULAR\_FUNCTION variable
- MOLECULAR\_INTERACTION variable
- OMIM variable

Distinct Count

Delete Set attributes

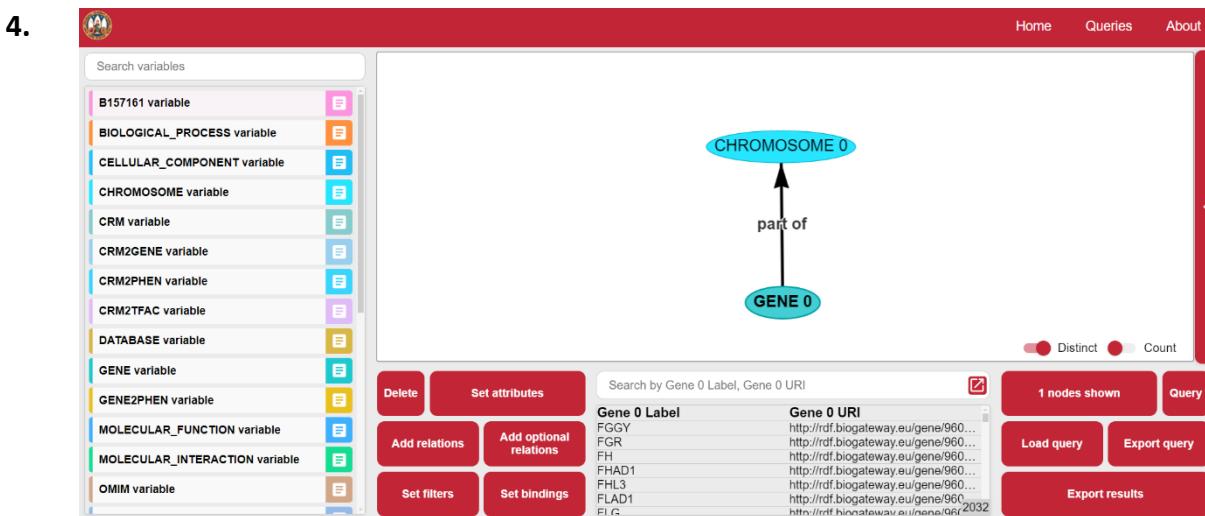
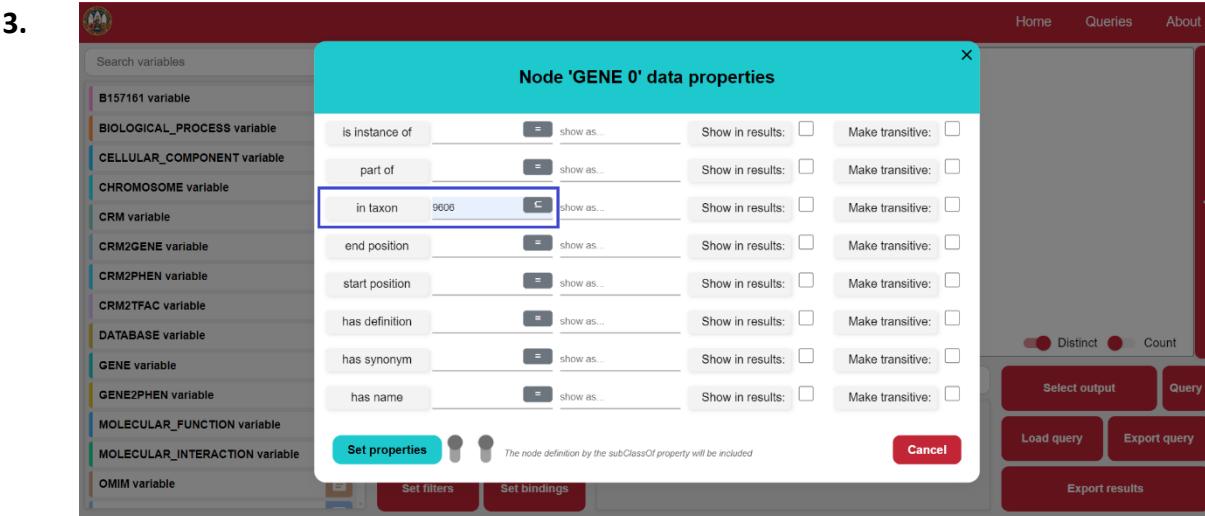
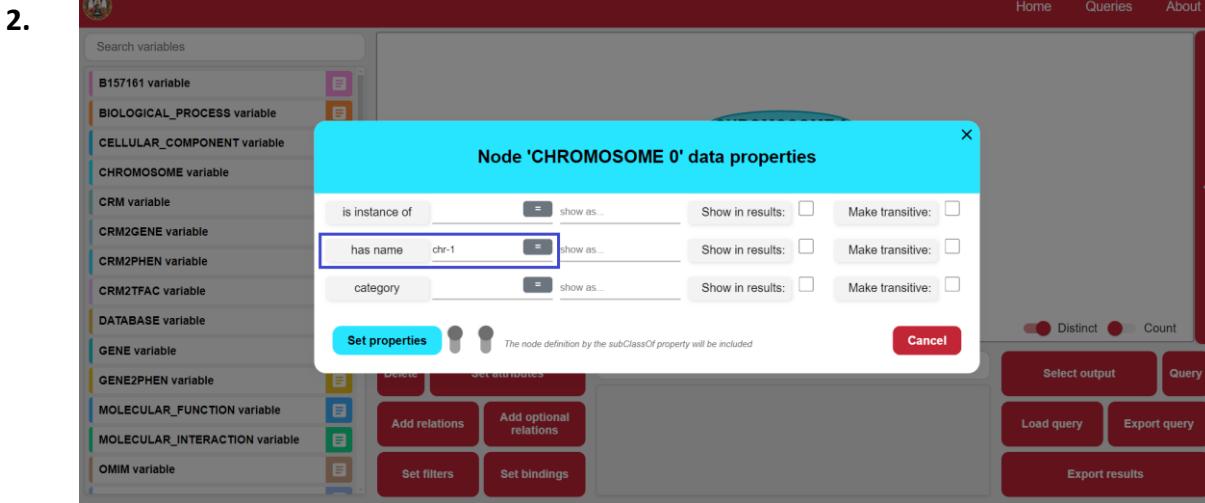
Add relations Add optional relations

Set filters Set bindings

No elements to display

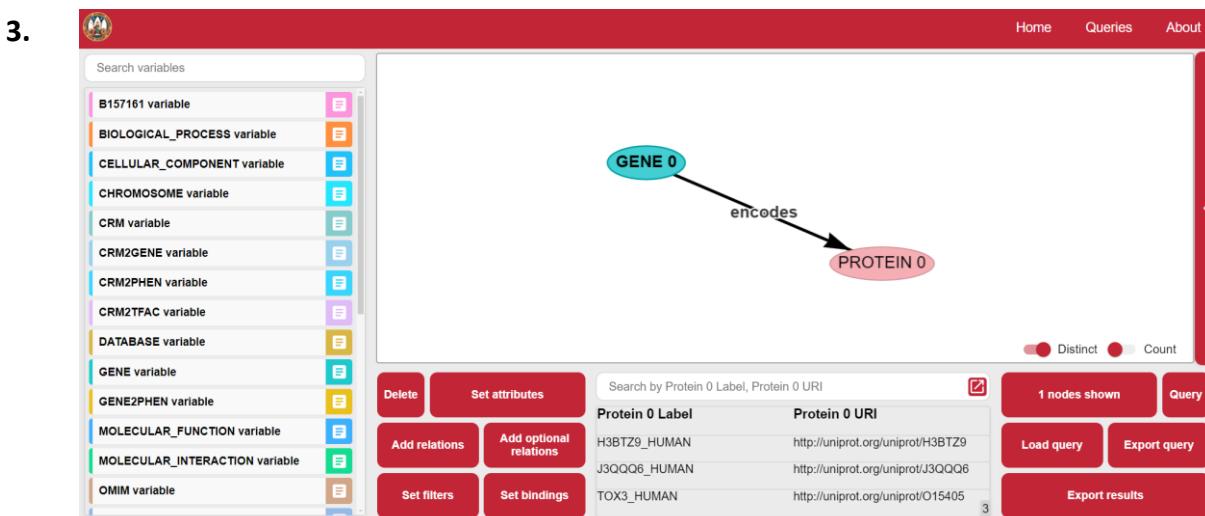
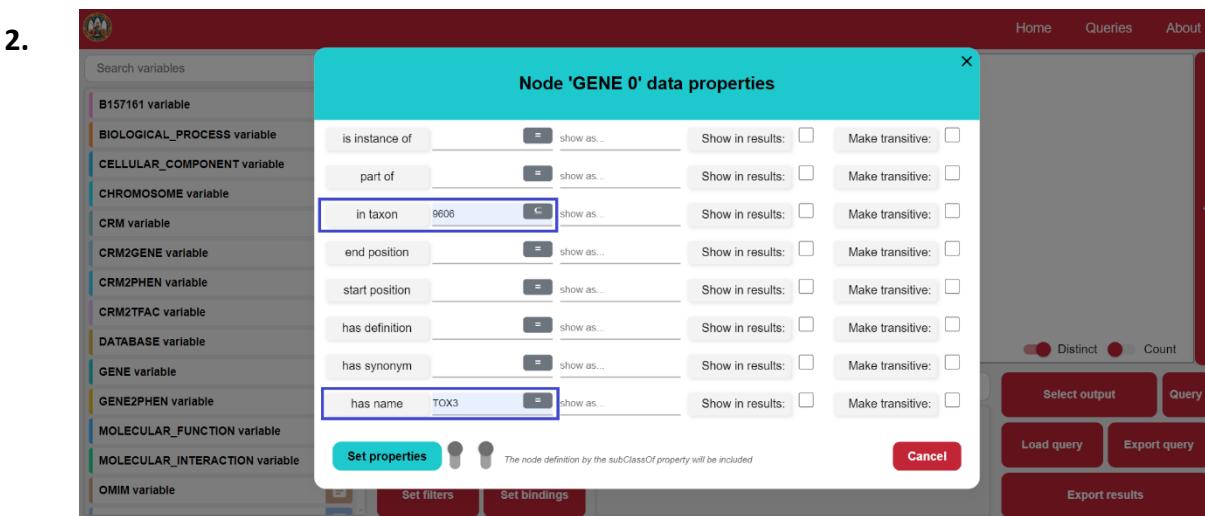
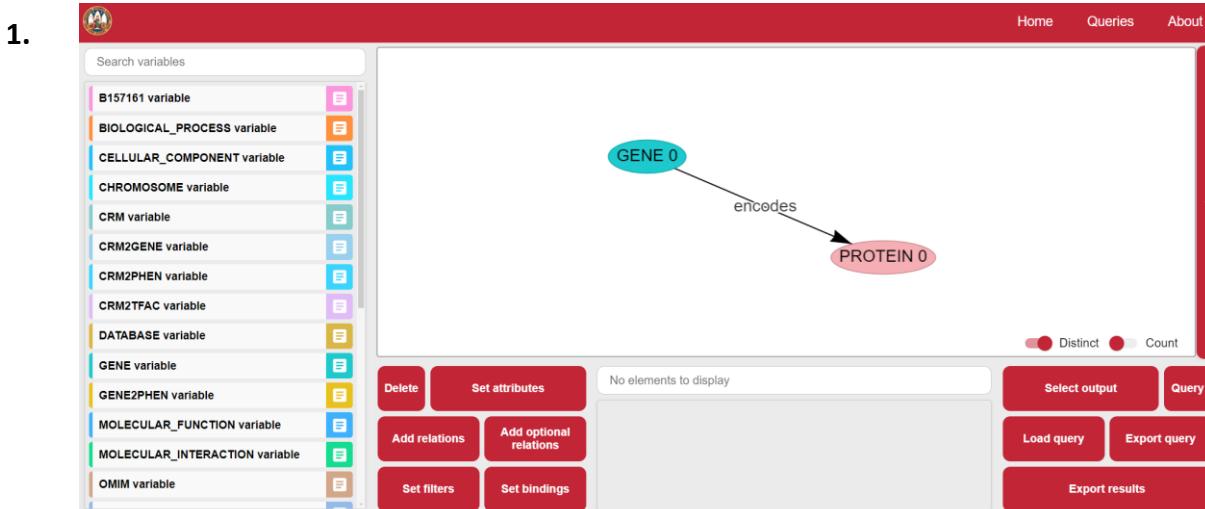
Select output Export query

Load query Export results



- **Example 3: Filtering by name.**

For this example we illustrate the query building to obtain the proteins encoded by the human TOX3 gene: *What proteins are encoded by the human TOX3 gene?* To do this, after including the link <Gene> <encodes> <Protein>, we modify the attributes of Gene node to indicate the name (TOX3) and the human taxon (9606).



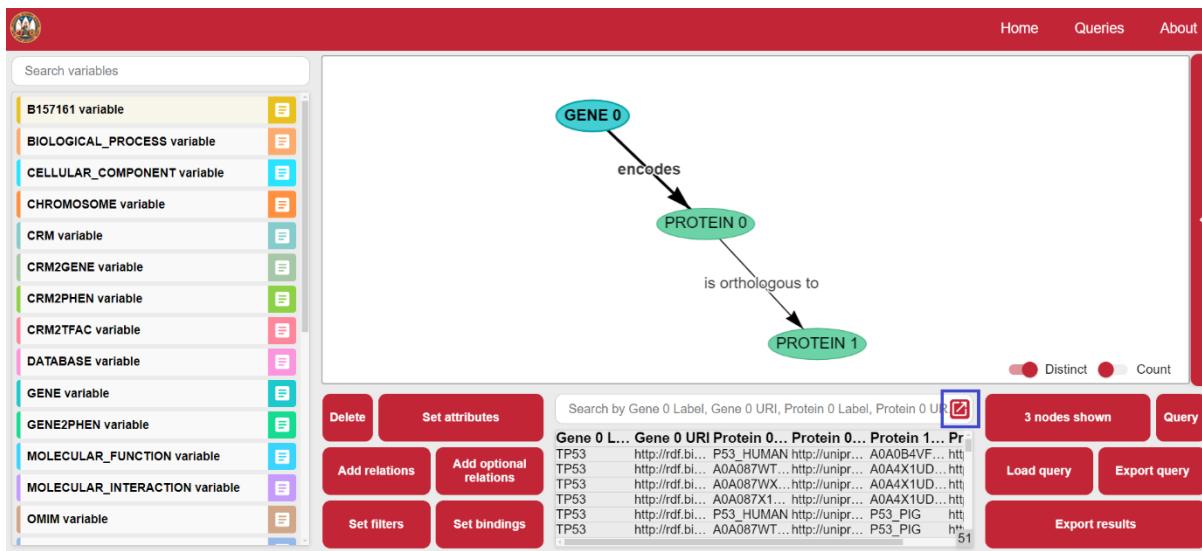
## 5.2. Count and display unique results

The output table shows the biological entities that meet the biological selection criteria. Since a user can design complex query patterns and has the freedom to choose which entities they want to include in the output ("Select output" button), duplicate entities might appear in the result. For this reason, the "Distinct" button is activated for automatic filtering. For example, we can query: *What are the orthologous proteins of the human TP53 gene?* To do this, we generate the relations <Gene> <encodes> <Protein>, and <Protein> <is orthologous to> <Protein>. Then we modify Gene's attributes to indicate the name and taxon.

1.

2.

3.

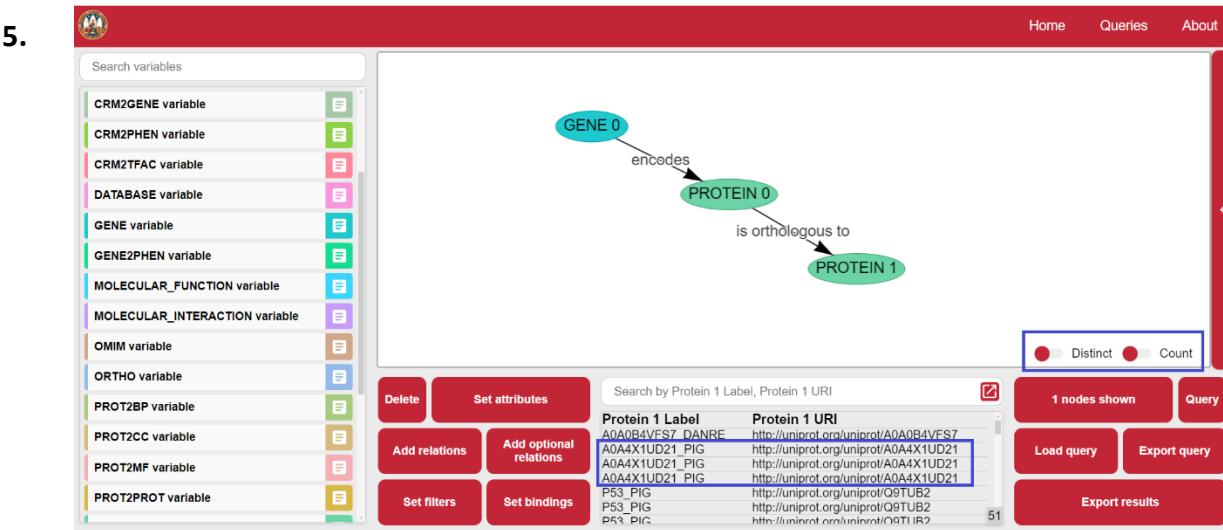


If we include all the entities in the output, and examine the extended table of results, we can see the relation between the human TP53 gene, its protein products and orthologous proteins in other organisms.

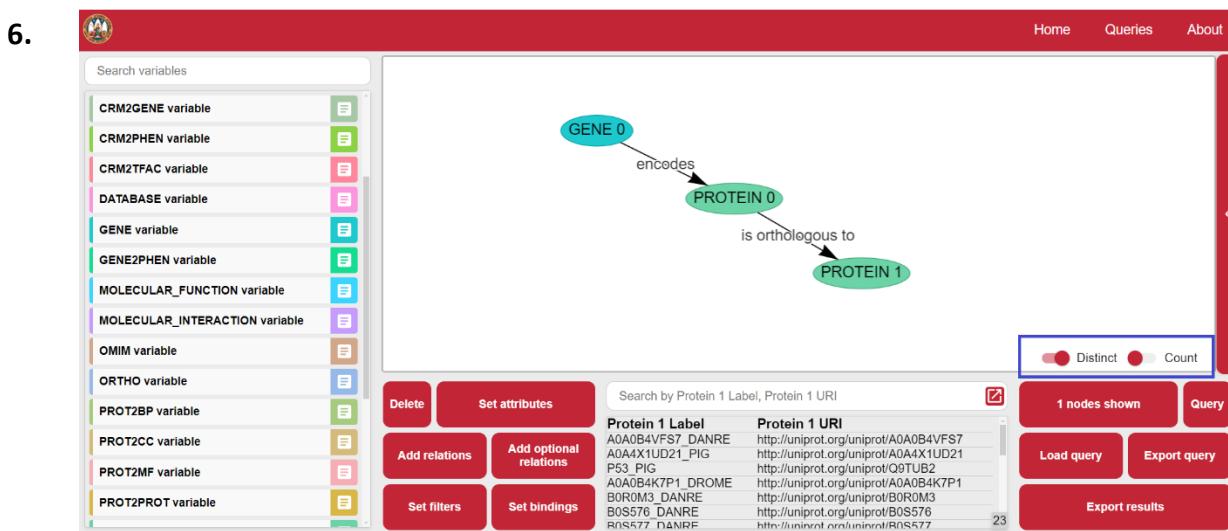
4.

Gene 0 Label	Gene 0 URI	Protein 0 Label	Protein 0 URI	Protein 1 Label	Protein 1 URI
TP53	<a href="http://rdf.biogateway.eu/gene/9606/TP53">http://rdf.biogateway.eu/gene/9606/TP53</a>	P53_HUMAN	<a href="http://uniprot.org/uniprot/P04637">http://uniprot.org/uniprot/P04637</a>	A0A0B4VFS7_DANRE	<a href="http://uniprot.org/uniprot/A0A0B4VFS7">http://uniprot.org/uniprot/A0A0B4VFS7</a>
TP53	<a href="http://rdf.biogateway.eu/gene/9606/TP53">http://rdf.biogateway.eu/gene/9606/TP53</a>	A0A087WT22_HUMAN	<a href="http://uniprot.org/uniprot/A0A087WT22">http://uniprot.org/uniprot/A0A087WT22</a>	A0A4X1UD21_PIG	<a href="http://uniprot.org/uniprot/A0A4X1UD21">http://uniprot.org/uniprot/A0A4X1UD21</a>
TP53	<a href="http://rdf.biogateway.eu/gene/9606/TP53">http://rdf.biogateway.eu/gene/9606/TP53</a>	A0A087WXZ1_HUMAN	<a href="http://uniprot.org/uniprot/A0A087WXZ1">http://uniprot.org/uniprot/A0A087WXZ1</a>	A0A4X1UD21_PIG	<a href="http://uniprot.org/uniprot/A0A4X1UD21">http://uniprot.org/uniprot/A0A4X1UD21</a>
TP53	<a href="http://rdf.biogateway.eu/gene/9606/TP53">http://rdf.biogateway.eu/gene/9606/TP53</a>	A0A087X1Q1_HUMAN	<a href="http://uniprot.org/uniprot/A0A087X1Q1">http://uniprot.org/uniprot/A0A087X1Q1</a>	A0A4X1UD21_PIG	<a href="http://uniprot.org/uniprot/A0A4X1UD21">http://uniprot.org/uniprot/A0A4X1UD21</a>
TP53	<a href="http://rdf.biogateway.eu/gene/9606/TP53">http://rdf.biogateway.eu/gene/9606/TP53</a>	P53_HUMAN	<a href="http://uniprot.org/uniprot/P04637">http://uniprot.org/uniprot/P04637</a>	P53_PIG	<a href="http://uniprot.org/uniprot/Q9TUB2">http://uniprot.org/uniprot/Q9TUB2</a>
TP53	<a href="http://rdf.biogateway.eu/gene/9606/TP53">http://rdf.biogateway.eu/gene/9606/TP53</a>	A0A087WT22_HUMAN	<a href="http://uniprot.org/uniprot/A0A087WT22">http://uniprot.org/uniprot/A0A087WT22</a>	P53_PIG	<a href="http://uniprot.org/uniprot/Q9TUB2">http://uniprot.org/uniprot/Q9TUB2</a>
TP53	<a href="http://rdf.biogateway.eu/gene/9606/TP53">http://rdf.biogateway.eu/gene/9606/TP53</a>	A0A087WXZ1_HUMAN	<a href="http://uniprot.org/uniprot/A0A087WXZ1">http://uniprot.org/uniprot/A0A087WXZ1</a>	P53_PIG	<a href="http://uniprot.org/uniprot/Q9TUB2">http://uniprot.org/uniprot/Q9TUB2</a>
TP53	<a href="http://rdf.biogateway.eu/gene/9606/TP53">http://rdf.biogateway.eu/gene/9606/TP53</a>	A0A087X1Q1_HUMAN	<a href="http://uniprot.org/uniprot/A0A087X1Q1">http://uniprot.org/uniprot/A0A087X1Q1</a>	P53_PIG	<a href="http://uniprot.org/uniprot/Q9TUB2">http://uniprot.org/uniprot/Q9TUB2</a>
TP53	<a href="http://rdf.biogateway.eu/gene/9606/TP53">http://rdf.biogateway.eu/gene/9606/TP53</a>	P53_HUMAN	<a href="http://uniprot.org/uniprot/P04637">http://uniprot.org/uniprot/P04637</a>	A0A0B4K7P1_DROME	<a href="http://uniprot.org/uniprot/A0A0B4K7P1">http://uniprot.org/uniprot/A0A0B4K7P1</a>
TP53	<a href="http://rdf.biogateway.eu/gene/9606/TP53">http://rdf.biogateway.eu/gene/9606/TP53</a>	P53_HUMAN	<a href="http://uniprot.org/uniprot/P04637">http://uniprot.org/uniprot/P04637</a>	B0R0M3_DANRE	<a href="http://uniprot.org/uniprot/B0R0M3">http://uniprot.org/uniprot/B0R0M3</a>
TP53	<a href="http://rdf.biogateway.eu/gene/9606/TP53">http://rdf.biogateway.eu/gene/9606/TP53</a>	P53_HUMAN	<a href="http://uniprot.org/uniprot/P04637">http://uniprot.org/uniprot/P04637</a>	B0S576_DANRE	<a href="http://uniprot.org/uniprot/B0S576">http://uniprot.org/uniprot/B0S576</a>
TP53	<a href="http://rdf.biogateway.eu/gene/9606/TP53">http://rdf.biogateway.eu/gene/9606/TP53</a>	P53_HUMAN	<a href="http://uniprot.org/uniprot/P04637">http://uniprot.org/uniprot/P04637</a>	B0S577_DANRE	<a href="http://uniprot.org/uniprot/B0S577">http://uniprot.org/uniprot/B0S577</a>
TP53	<a href="http://rdf.biogateway.eu/gene/9606/TP53">http://rdf.biogateway.eu/gene/9606/TP53</a>	P53_HUMAN	<a href="http://uniprot.org/uniprot/P04637">http://uniprot.org/uniprot/P04637</a>	G1K2L5_DANRE	<a href="http://uniprot.org/uniprot/G1K2L5">http://uniprot.org/uniprot/G1K2L5</a>
TP53	<a href="http://rdf.biogateway.eu/gene/9606/TP53">http://rdf.biogateway.eu/gene/9606/TP53</a>	P53_HUMAN	<a href="http://uniprot.org/uniprot/P04637">http://uniprot.org/uniprot/P04637</a>	A0A167VDT2_CHICK	<a href="http://uniprot.org/uniprot/A0A167VDT2">http://uniprot.org/uniprot/A0A167VDT2</a>
TP53	<a href="http://rdf.biogateway.eu/gene/9606/TP53">http://rdf.biogateway.eu/gene/9606/TP53</a>	P53_HUMAN	<a href="http://uniprot.org/uniprot/P04637">http://uniprot.org/uniprot/P04637</a>	F1P1U2_CHICK	<a href="http://uniprot.org/uniprot/F1P1U2">http://uniprot.org/uniprot/F1P1U2</a>
TP53	<a href="http://rdf.biogateway.eu/gene/9606/TP53">http://rdf.biogateway.eu/gene/9606/TP53</a>	P53_HUMAN	<a href="http://uniprot.org/uniprot/P04637">http://uniprot.org/uniprot/P04637</a>	D4AA88_RAT	<a href="http://uniprot.org/uniprot/D4AA88">http://uniprot.org/uniprot/D4AA88</a>

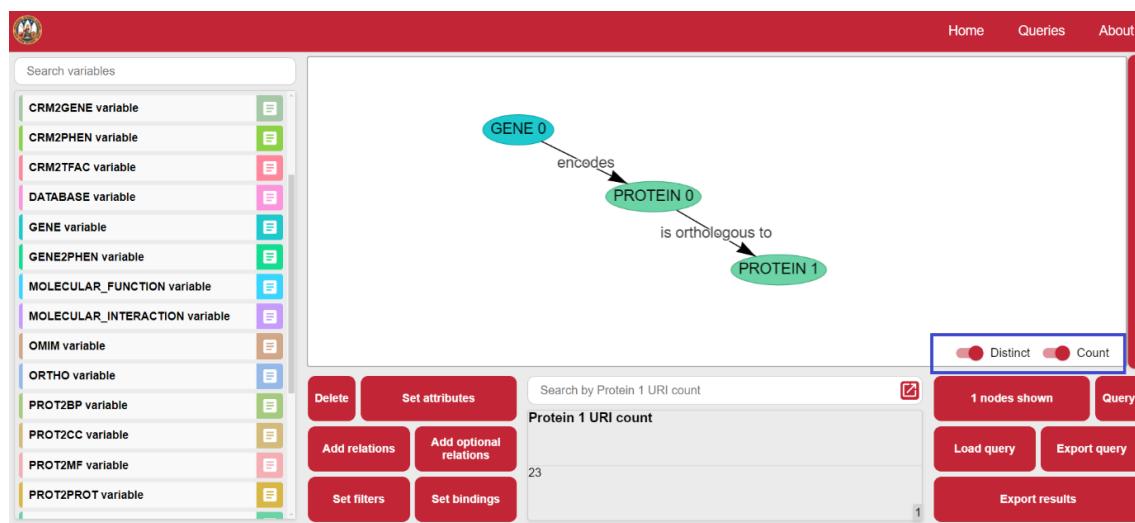
The table contains a total of 51 entries/rows, corresponding to the results that satisfy the search pattern. However, we can see that several proteins encoded by the human TP53 gene are related to the same orthologous protein. Therefore, if we now only select as output the orthologous proteins, and we deactivate the "Distinct" button, we obtain the same 51 results that satisfy the search pattern, but with repeated values, because we are only selecting the orthologous proteins.



On the contrary, with the "Distinct" functionality activated (activated by default), we only obtain the unique results, which in this case are 23 (23 orthologous proteins).

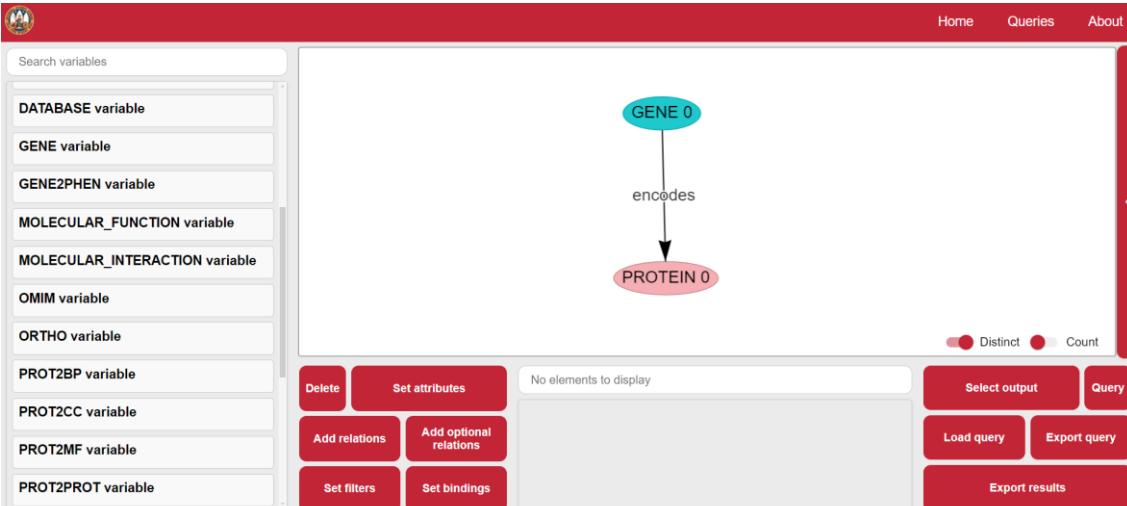


On the other hand, activating the "Count" button displays the number of entities that fit the search pattern of the query.

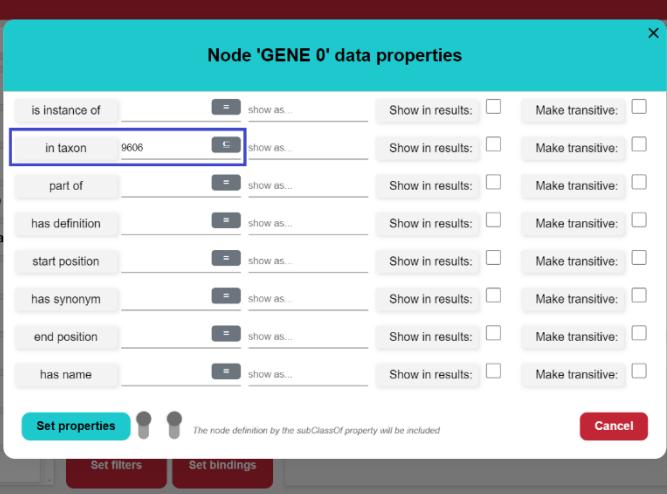


Below we include another example. We can build the query: *How many human genes encode proteins, and how many proteins are there?*

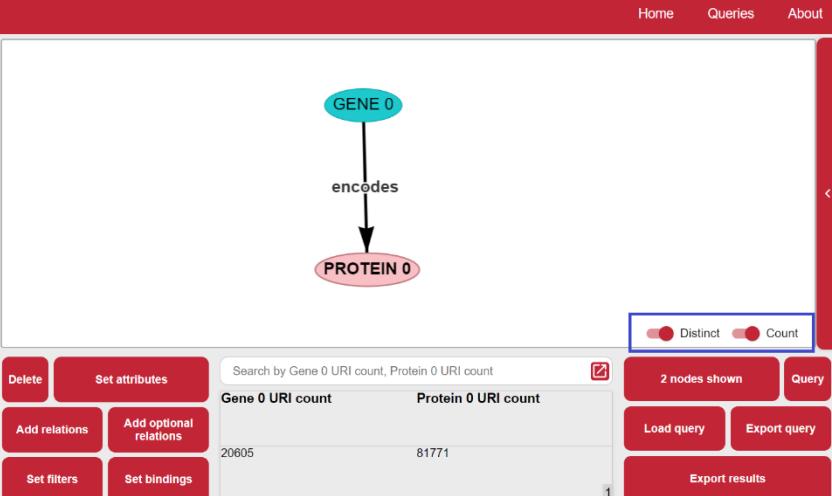
1.



2.



3.



### 5.3. Optional relations

INTUITION also allows to include optional relations (“Add optional relations” button). As this is an optional pattern, the information is added if it exists, so it does not work as a filter. In this way, INTUITION allows queries like: *What proteins are encoded by the human TOX3 gene? Do these protein products interact with any other proteins? Is there information about proteins orthologous to those encoded by the human TOX3 gene?*

1.

Home    Queries    About

Search variables

- B157161 variable
- BIOLOGICAL\_PROCESS variable
- CELLULAR\_COMPONENT variable
- CHROMOSOME variable
- CRM variable
- CRM2GENE variable
- CRM2PHEN variable
- CRM2TFAC variable
- DATABASE variable
- GENE variable
- GENE2PHEN variable
- MOLECULAR\_FUNCTION variable
- MOLECULAR\_INTERACTION variable
- OMIM variable

Distinct    Count

Delete    Set attributes    No elements to display    Select output    Query

Add relations    Add optional relations    Load query    Export query

Set filters    Set bindings    Export results

2.

Node 'GENE 0' data properties

is instance of    show as...    Show in results:     Make transitive:   
 part of    show as...    Show in results:     Make transitive:   
 in taxon 9606    show as...    Show in results:     Make transitive:   
 end position    show as...    Show in results:     Make transitive:   
 start position    show as...    Show in results:     Make transitive:   
 has definition    show as...    Show in results:     Make transitive:   
 has synonym    show as...    Show in results:     Make transitive:   
 has name TOX3    show as...    Show in results:     Make transitive:   
 Set properties    The node definition by the subClassOf property will be included    Cancel

Set filters    Set bindings    ZNF217    http://rdf.bi... ZNF217\_HU... http://unipr... PAX7\_HU... 10000    4 nodes shown    Load query    Export query    Export results

3.

Home    Queries    About

Search variables

- B157161 variable
- BIOLOGICAL\_PROCESS variable
- CELLULAR\_COMPONENT variable
- CHROMOSOME variable
- CRM variable
- CRM2GENE variable
- CRM2PHEN variable
- CRM2TFAC variable
- DATABASE variable
- GENE variable
- GENE2PHEN variable
- MOLECULAR\_FUNCTION variable
- MOLECULAR\_INTERACTION variable
- OMIM variable

Distinct    Count

Delete    Set attributes    Search by Gene 0 Label, Gene 0 URI, Protein 0 Label, Protein 0 URI     4 nodes shown    Query

Add relations    Add optional relations    Load query    Export query

Set filters    Set bindings    Export results

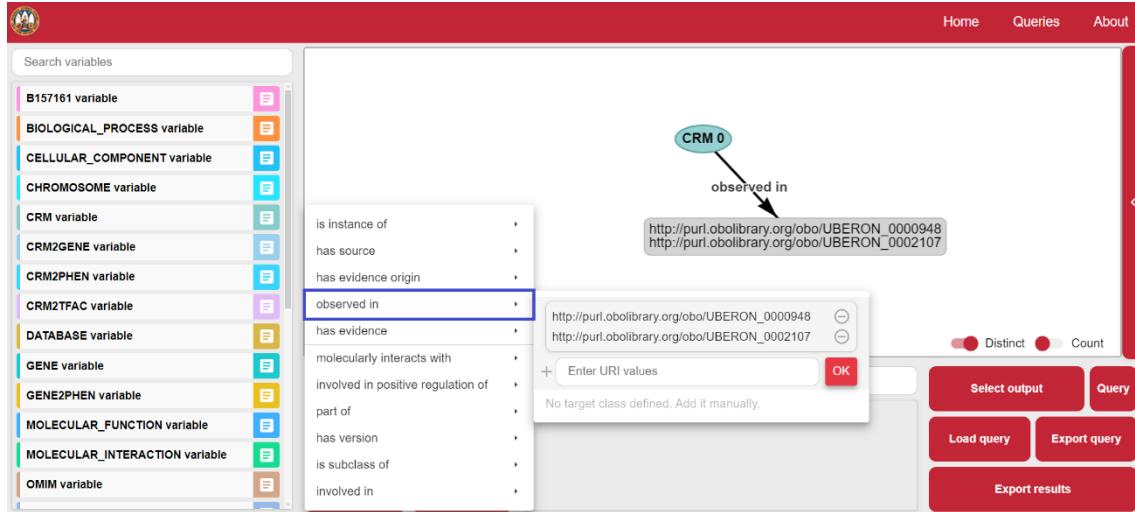
**Note:** Queries can involve different variables of the same entity type. Queries may require handling the same entity as different variables, e.g. protein-protein interaction involves two different proteins, and therefore two different variables. INTUITION allows the inclusion of the same variable more than once. These nodes are numbered starting with 0. The counter is reset to zero when the application is updated, not when the node is deleted.

#### 5.4. Multiple values

To avoid creating repetitive queries when the general structure of the query is the same, but the characteristics of the entities are different, INTUITION allows a user to assign different values to the variables. For example, if we are interested in searching for cis-regulatory modules (CRM) identified in two or more tissues of interest, we do not need to repeat the same query for each tissue. As shown in the example (*Which CRMs have been identified in heart (UBERON\_0000948) and liver (UBERON\_0002107)?*), we can specify different tissues in the "Enter URI values" cell of "observed in" property, in "Add realtions". Click on "+" to include values and click on "OK" when all values are listed. As BioGateway uses semantic resources to identify entities, the values entered must be Uniform Resource Identifiers (URIs) corresponding to these resources.

*Which CRMs have been identified in heart (UBERON\_0000948) and liver (UBERON\_0002107)?*

1.



The screenshot shows the INTUITION query builder interface. On the left, a sidebar lists various variable types with icons: B157161 variable (pink), BIOLOGICAL\_PROCESS variable (orange), CELLULAR\_COMPONENT variable (blue), CHROMOSOME variable (teal), CRM variable (purple), CRM2GENE variable (light blue), CRM2PHEN variable (light teal), CRM2TFAC variable (light orange), DATABASE variable (yellow), GENE variable (green), GENE2PHEN variable (light green), MOLECULAR\_FUNCTION variable (light teal), MOLECULAR\_INTERACTION variable (light green), and OMIM variable (brown). The 'CRM variable' section is currently selected. The main panel shows a list of properties: is instance of, has source, has evidence origin, observed in (selected and highlighted in blue), has evidence, molecularly interacts with, involved in positive regulation of, part of, has version, is subclass of, and involved in. Below this list, a tooltip for 'CRM 0' shows an 'observed in' relationship with two URIs: http://purl.obolibrary.org/obo/UBERON\_0000948 and http://purl.obolibrary.org/obo/UBERON\_0002107. A modal dialog box is open, prompting the user to 'Enter URI values' with an 'OK' button. At the bottom right of the interface, there are buttons for 'Select output', 'Query', 'Load query', 'Export query', and 'Export results'.

2.

Search variables

- B157161 variable
- BIOLOGICAL\_PROCESS variable
- CELLULAR\_COMPONENT variable
- CHROMOSOME variable
- CRM variable
- CRM2GENE variable
- CRM2PHEN variable
- CRM2TFAC variable
- DATABASE variable
- GENE variable
- GENE2PHEN variable
- MOLECULAR\_FUNCTION variable
- MOLECULAR\_INTERACTION variable
- OMIM variable

CRM 0

observed in

http://purl.obolibrary.org/obo/UBERON\_0000948  
http://purl.obolibrary.org/obo/UBERON\_0002107

Search by Crm 0 Label, Crm 0 URI

Crm 0 Label

Crm 0 URI

1 nodes shown

Distinct Count

Delete Set attributes

Add relations Add optional relations

Set filters Set bindings

Load query Export query

Export results

If we want to count the number of CRMs, we click on the 'Count' button to activate this counting functionality: *How many CRMs have been identified in heart (UBERON\_0000948) and liver (UBERON\_0002107)?*

Search variables

- B157161 variable
- BIOLOGICAL\_PROCESS variable
- CELLULAR\_COMPONENT variable
- CHROMOSOME variable
- CRM variable
- CRM2GENE variable
- CRM2PHEN variable
- CRM2TFAC variable
- DATABASE variable
- GENE variable
- GENE2PHEN variable
- MOLECULAR\_FUNCTION variable
- MOLECULAR\_INTERACTION variable
- OMIM variable

CRM 0

observed in

http://purl.obolibrary.org/obo/UBERON\_0000948  
http://purl.obolibrary.org/obo/UBERON\_0002107

Search by Crm 0 URI count

Crm 0 URI count

1536465

1 nodes shown

Distinct Count

Delete Set attributes

Add relations Add optional relations

Set filters Set bindings

Load query Export query

Export results

We can also add multiple values to the node that acts as the subject of the triplet:

Search variables

- B157161 variable
- BIOLOGICAL\_PROCESS variable
- CELLULAR\_COMPONENT variable
- CHROMOSOME variable
- CRM variable
- CRM2GENE variable
- CRM2PHEN variable
- CRM2TFAC variable
- DATABASE variable
- GENE variable
- GENE2PHEN variable
- MOLECULAR\_FUNCTION variable
- MOLECULAR\_INTERACTION variable
- OMIM variable

CRM 0

observed in

Enter values as a gene OK

No elements to display

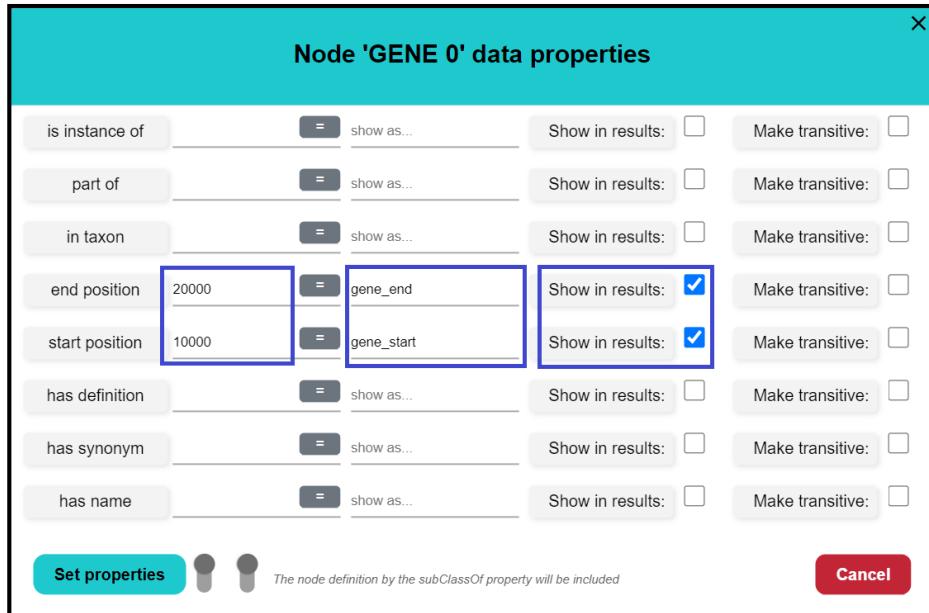
Select output Query

Load query Export query

Export results

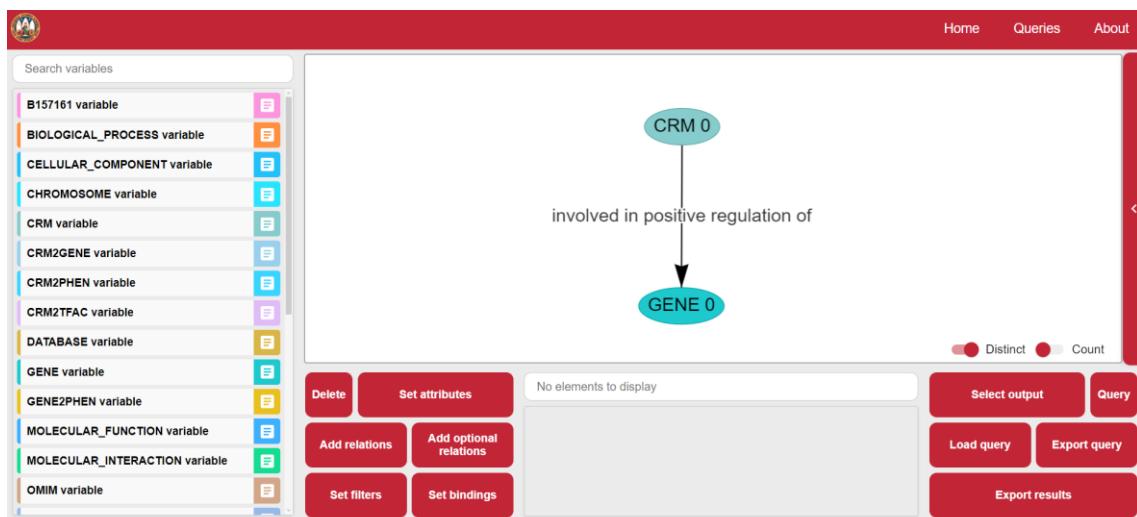
## 5.5. Creating and filtering variables

INTUITION allow a user to create their own selection variables. This functionality is implemented in "Set bindings" button, in the "Pattern designer", and can be applied to attributes used in the query, renamed, or selected for output. To use an attribute in the search pattern, a value must be entered to act as a filter. To rename an attribute, simply change its name in "show as". To mark it for data output, check "Shown in results".

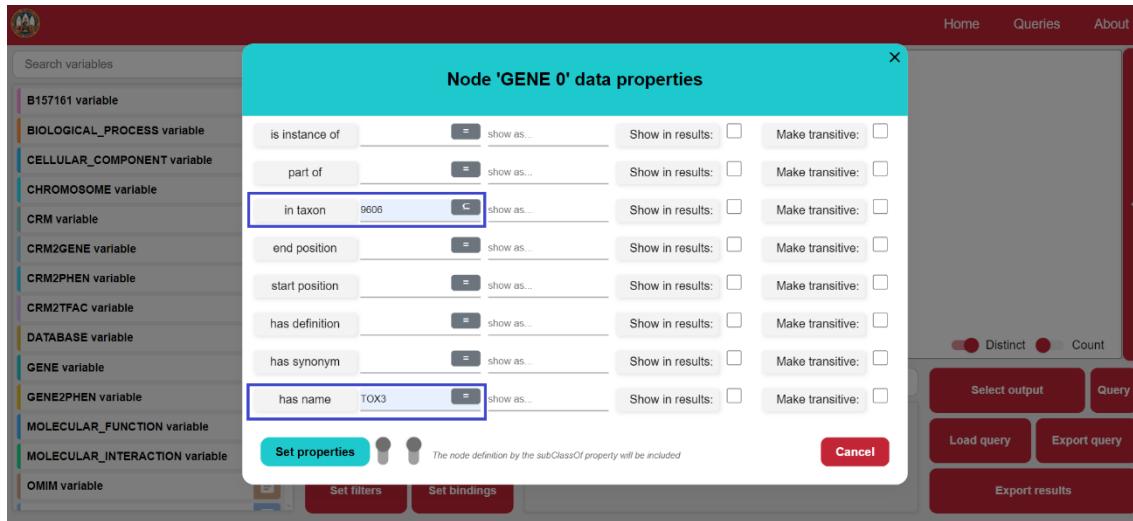


For example, by subtracting the end and start positions of the CRMs, and adding “1” to this number, we obtain the length of the sequences in a new variable. Then, we can filter this new variable in the “Set filters” button. Below we illustrate an example: *Which CRMs with a length less than or equal to 500 bp positively regulate the human TOX3 gene?* For this:

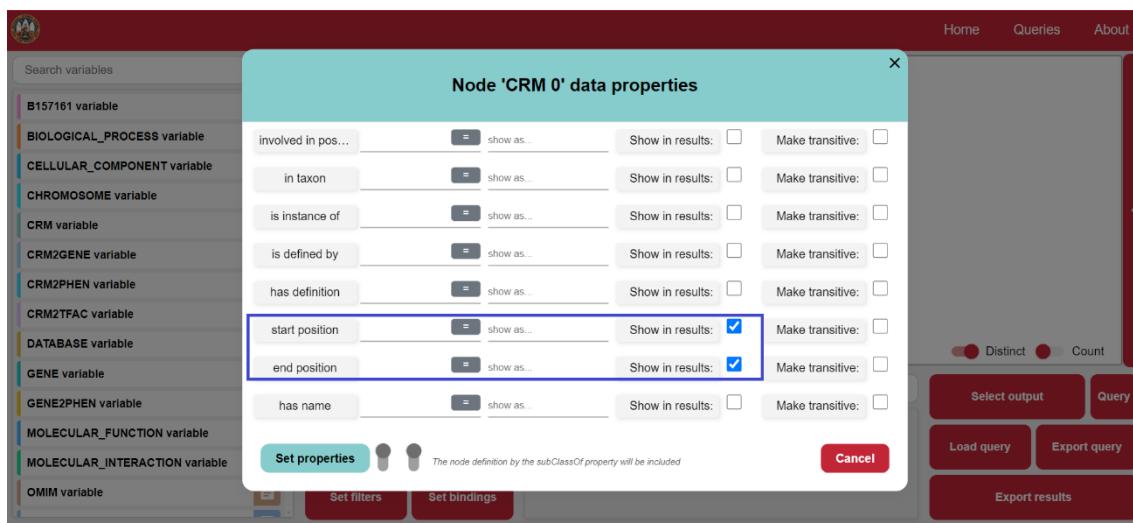
- First, we generate the relation <CRM> <involved in positive regulation of> <Gene>.



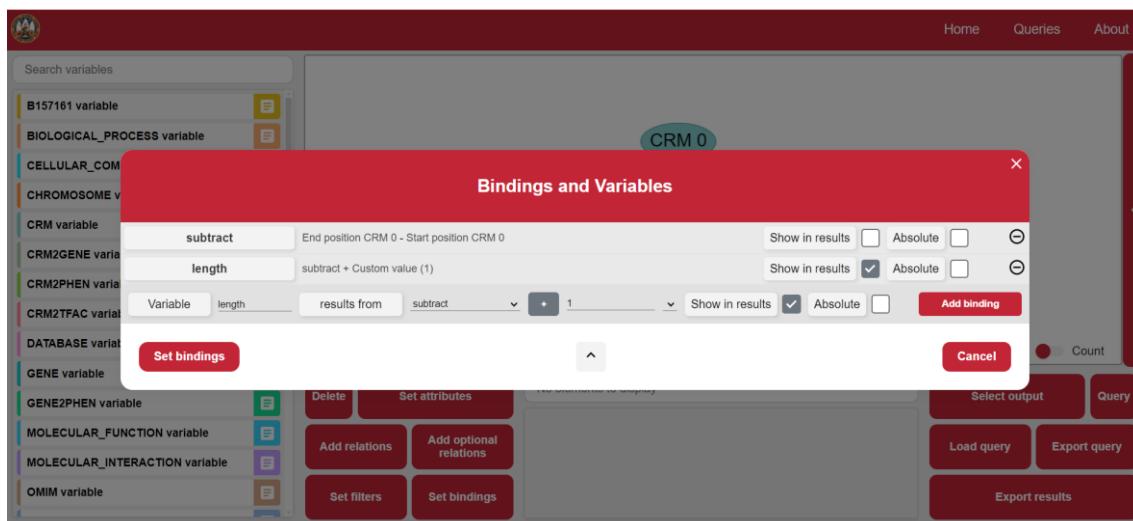
- Assign the attributes corresponding to the gene (name TOX3, and taxon).



- Select the CRM attributes that we are going to use to generate the new variables.



- Create the new variable in “Set bindings”.



- We filter in ‘Set filters’ the new variable.

The screenshot shows the INTUITION interface with a 'Filters' dialog box open. The dialog box has a red header and contains a filter for 'length' less than or equal to 500. It includes buttons for 'Set Filters', 'Cancel', and 'Add'. The background shows a search interface with various variables listed on the left and a graph visualization on the right.

- Select the output and run the query:

The screenshot shows the INTUITION interface displaying the results of a query. The results table shows 91 rows of data related to CRM 0 and GENE 0. The table includes columns for Crm 0 Label, Crm 0 URI, Start positi..., End positi..., and Length. Buttons for '1 nodes shown', 'Query', 'Load query', 'Export query', and 'Export results' are visible on the right.

## 5.6. Union of queries

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 (*Which OMIM entities contain 'breast cancer' in their preferred label or alternative label?*), i.e. "name" and "synonym" are different attributes, but we can unite their values in a common variable using the UNION clause and a new unified rename for the attributes we want to unify ("label" in this example). To do that:

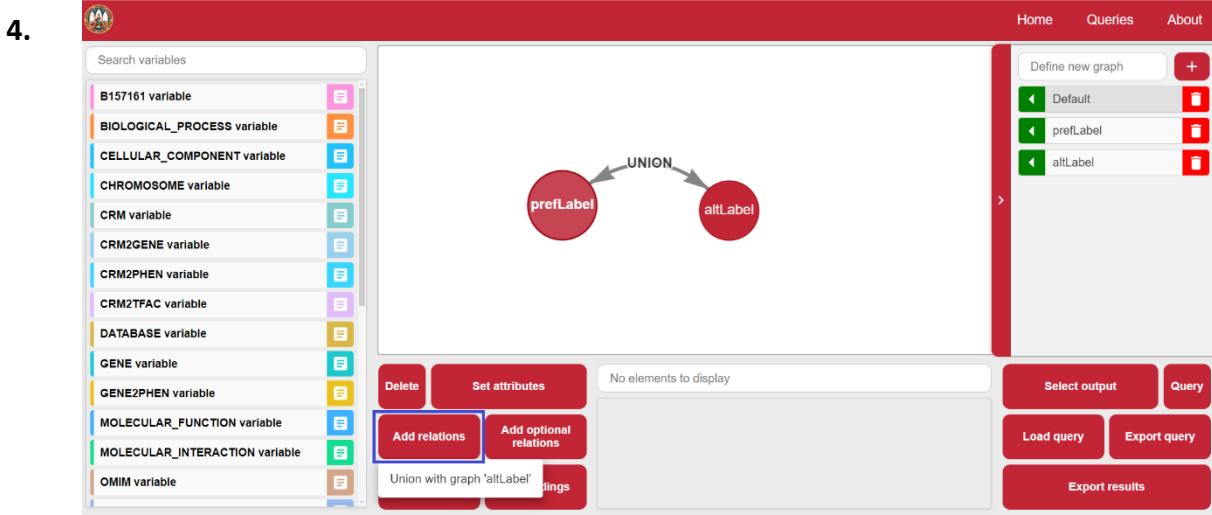
1. In the "Union builder" section, we create the graphs belonging to each of the subqueries, and we include an OMIM node in each of them. In the example, the graphs are "prefLabel" for the main label query and "altLabel" for the query of the synonym.
2. In each of the graphs we define the variable "label" 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.

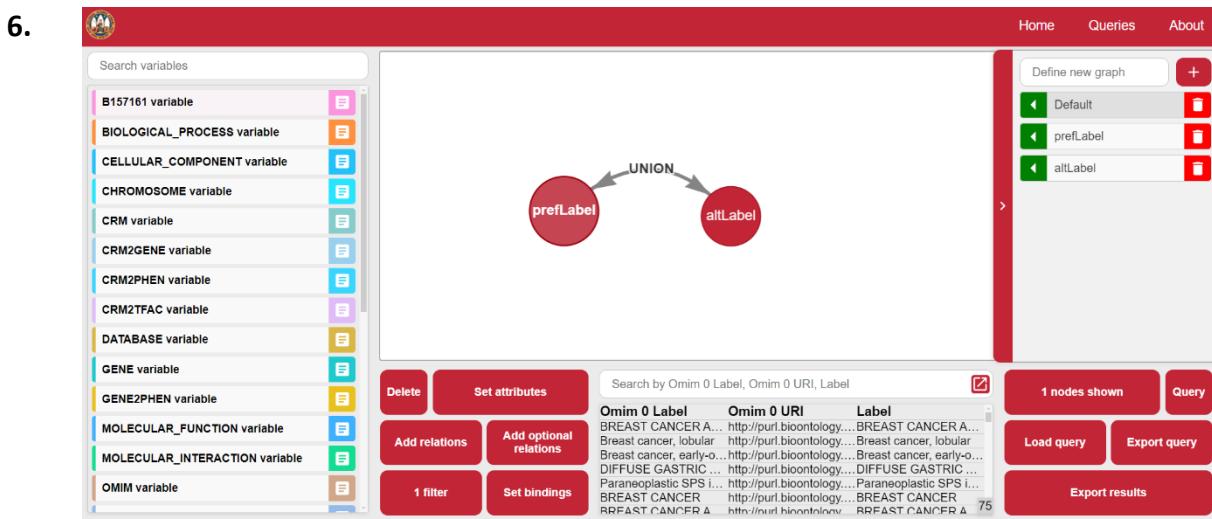
2.

3.



5.

The screenshot shows a 'Filters' dialog box. It contains two input fields: 'label' with 'breast cancer' and 'Filter' with 'label'. Below these are 'Add' and 'Cancel' buttons. At the bottom are 'Set Filters' and 'Cancel' buttons.



## 6. Use Cases

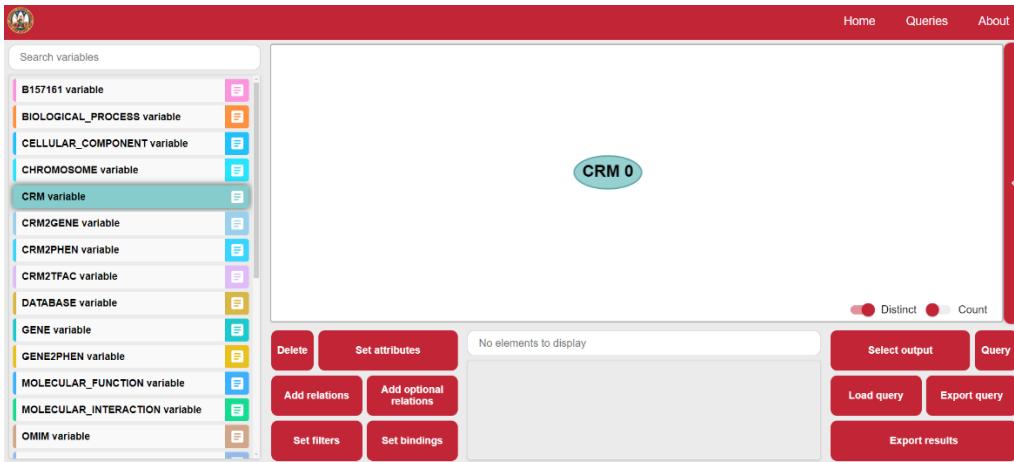
The following Use Cases were developed in the paper "*Integration of chromosome locations and functional aspects of enhancers and topologically associating domains in knowledge graphs enables versatile queries about gene regulation*". The corresponding queries are attached for reproducibility and as examples of use. These use cases include complex queries that connect multiple nodes, use different filters, create variables, and join queries, so we recommend their consultation for a deeper understanding of the concepts introduced here for the graphical query building.

### 1. Use case 1: json files to load [here](#).

2. Use case 2: json files to load [here](#).
3. Use case 3: json files to load [here](#).

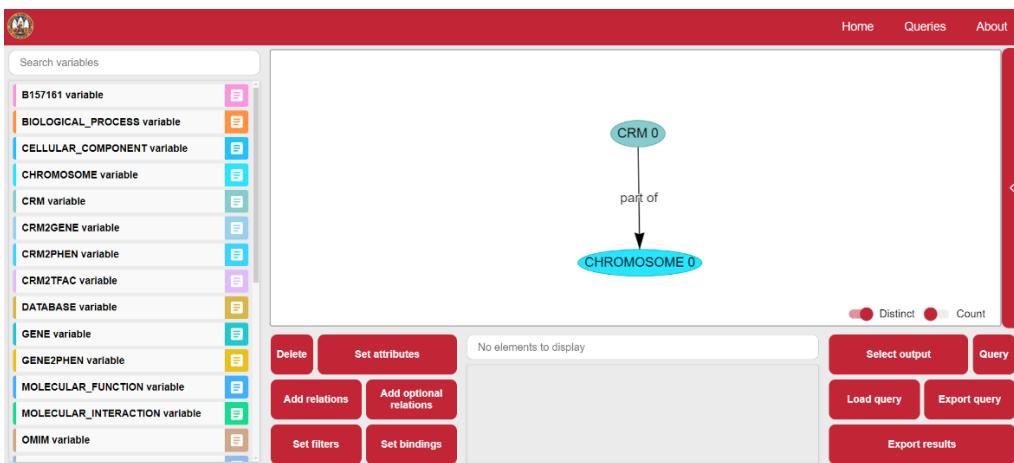
A guided step-by-step guide to building Use Case 1.1 is shown below: *Is the rs4784227 mutation (chr16:52565276) located in any enhancer sequence linked to target genes in the network? What databases support the sequence and what are their target genes? Is the enhancer related to any disease? Which proteins are encoded by the genes?*

- First, we insert the CRM node by clicking on the CRM variable (“Variable browser” section).



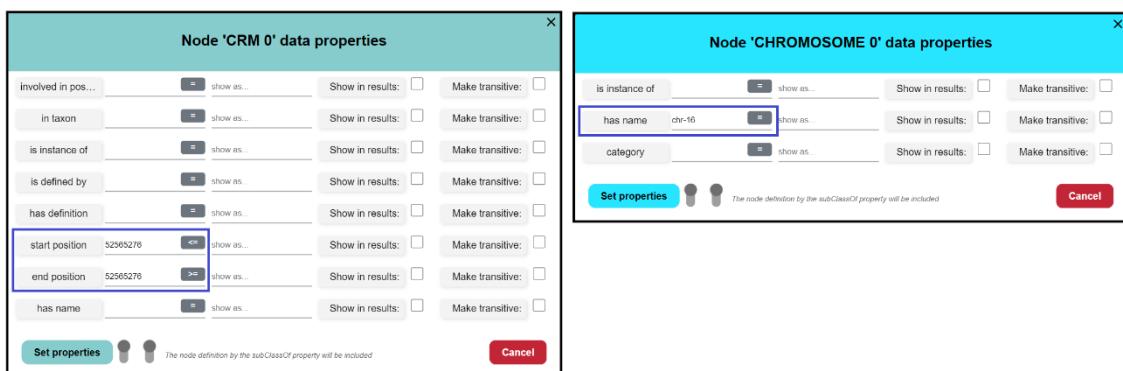
The screenshot shows the Variable browser interface. The sidebar on the left lists various variable types, with 'CRM variable' selected. The main panel displays a node labeled 'CRM 0' with a blue oval highlight. Below the node are buttons for 'Delete', 'Set attributes', 'Add relations', 'Add optional relations', 'Set filters', and 'Set bindings'. To the right of the node are buttons for 'Select output', 'Query', 'Load query', 'Export query', and 'Export results'. At the bottom of the main panel, it says 'No elements to display'.

- We link the CRM to the chromosome variable (“Add relations” button).



The screenshot shows the Variable browser interface after adding a relation. The main panel now displays a directed graph with an arrow pointing from 'CRM 0' to 'CHROMOSOME 0'. The arrow is labeled 'part of'. The node 'CRM 0' is highlighted with a blue oval. The node 'CHROMOSOME 0' is also highlighted with a blue oval. The rest of the interface is identical to the previous screenshot.

- And modify the attributes of both variables to select only those CRMs that overlap with the mutation (chr16:52565276) (“Set attributes” button).

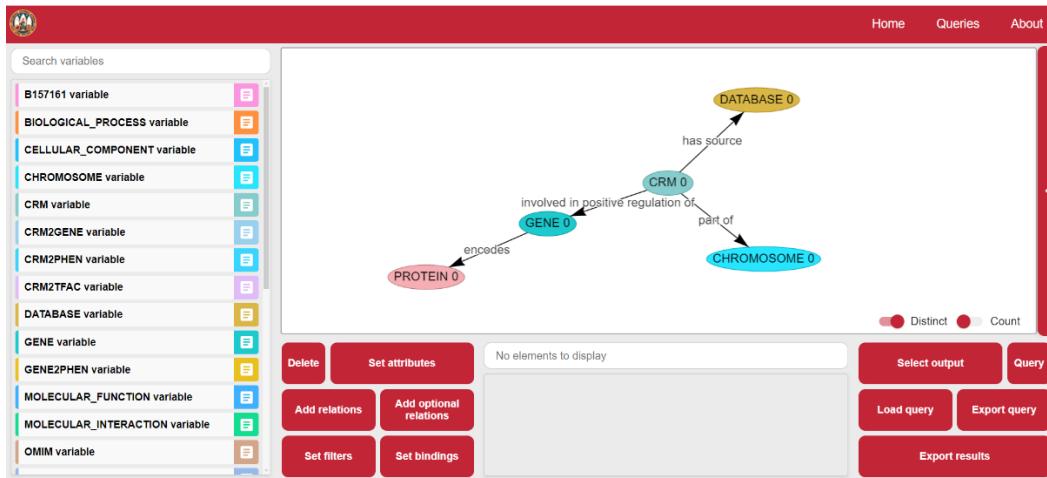


The screenshot shows two 'data properties' dialog boxes side-by-side.

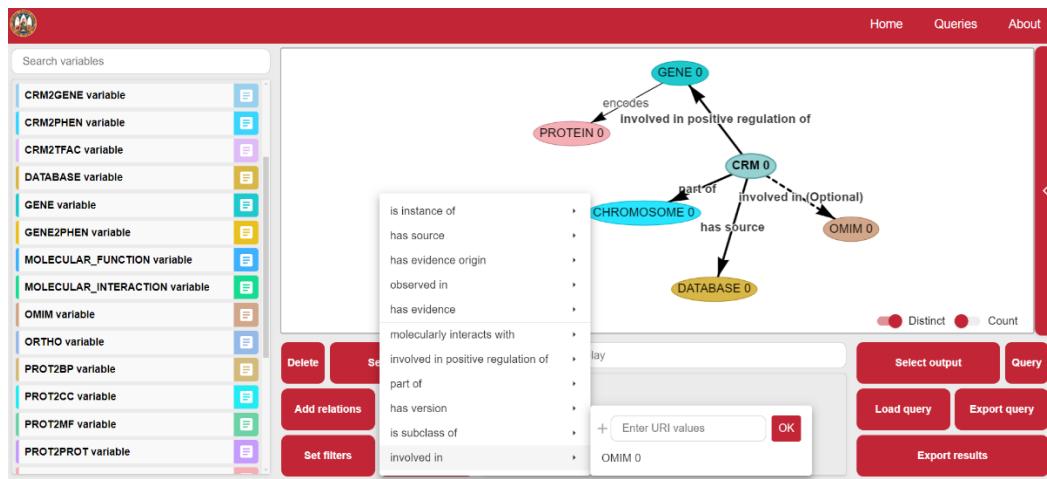
**Node 'CRM 0' data properties:** This dialog lists properties for the CRM node. The 'start position' and 'end position' fields are highlighted with a blue box. The 'start position' field contains '52565276' and the 'end position' field contains '52565276'. Other properties listed include 'involved in pos...', 'in taxon', 'is instance of', 'is defined by', 'has definition', 'has name', 'show as...', 'Show in results:', 'Make transitive:', and 'Set properties'.

**Node 'CHROMOSOME 0' data properties:** This dialog lists properties for the chromosome node. The 'has name' field is highlighted with a blue box and contains 'chr-16'. Other properties listed include 'is instance of', 'has name', 'category', 'show as...', 'Show in results:', 'Make transitive:', and 'Set properties'.

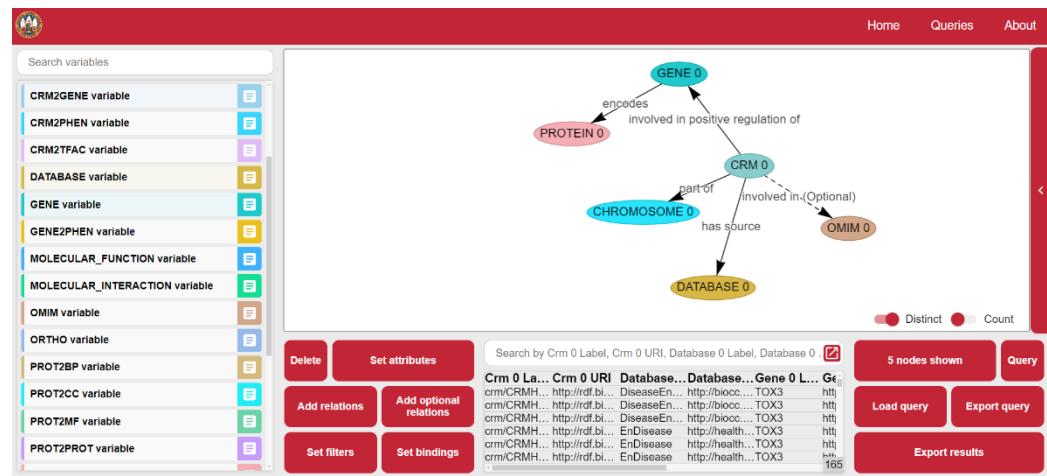
- We link the CRM entity with its database and target genes. We also link the genes to their encoded proteins (“Add relations” button).



- We include the relation between CRM and phenotype as an optional pattern (information that is included additionally and does not act as a filter) (“Add optional relations” button).



- Select the output data of interest (“Select output”) and run the query (“Query”).



- Finally, we can expand the results table, save the results table (“Export results”) and save the generated query (“Export query”).