

INTUITION Tutorial for BioGateway

1. Introduction.....	1
2. Design	1
3. Variables and properties.....	2
4. How to build a query	4
5. Data filtering and other possible operations	5
5.1. Filters of properties.....	5
5.2. Count and display unique results.....	9
5.3. Multiple values	10
5.4. Creating and filtering variables	10
5.5. Union of queries.....	11
6. Use Cases	13
7. Useful notes.....	13
7.1. Numbering of variables.....	13
7.2. Selection of object properties	14

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 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 biological queries graphically by representing biological entities (nodes) related through properties (edges).

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.

2. Design

In INTUITION we distinguish different sections:

A- Query building canvas.

- B- Entity browser (Currently deactivated).
- C- Union buider (Used for queries that use union clauses).
- D- Variable browser (Main types of entities in the network).
- E- Properties selectors:
 - 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 display.
- 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 displays a complex web interface for biological data querying. It features a top navigation bar with 'Home', 'Queries', and 'About' links. The main workspace is divided into several functional areas:

- Panel B (Entity browser):** A search bar and a list of entities from the OMIM ontology, including 'IMMUNOGLOBULIN HEAVY CHAIN V...', 'CAST, 1-BP DEL, 1750G', 'TPM3, MET9ARG', and 'UGT1A1, IVSAS3, A-G, -2'.
- Panel A (Query graph):** A central area showing a query graph with a node 'GENE 0' connected to 'PROTEIN 0' via an 'encodes' relationship.
- Panel C (Union builder):** A sidebar for defining new graphs, currently showing a 'Default' graph.
- Panel D (Variable browser):** A list of variables for selection, including 'MOLECULAR_FUNCTION variable', 'CELLULAR_COMPONENT variable', 'BIOLOGICAL_PROCESS variable', 'PROTEIN variable', 'GENE variable', and 'PROT2CC variable'.
- Panel E (Properties selectors):** A set of buttons for managing query properties: 'Delete', 'Set 'gene' properties...', 'Set 'gene' optional properties...', and 'Set 'gene' data properties...'.
- Panel F (Output display):** A table showing query results with columns for 'Gene 0 Label', 'Gene 0 URI', and 'Protein 0 Label/URI'. It lists various gene and protein identifiers and their corresponding URIs.
- Panel G (Query builder):** A set of buttons for managing queries: 'Export as...', 'Query', '2 nodes shown', 'No filters set', 'Export queries', and 'Load queries'.

3. Variables and properties

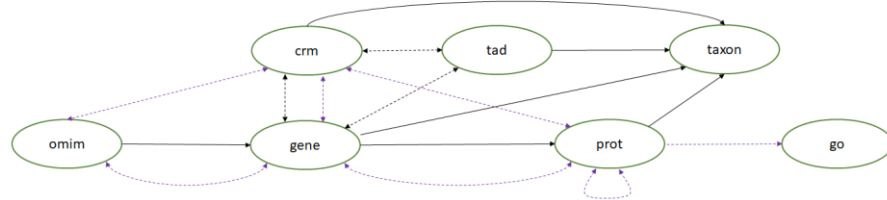
Variables:

- CRM variable: cis regulatory module.
- Gene variable: genes.
- Protein variable: proteins.
- OMIM variable: entities from OMIM ontology (mainly phenotypes).
- Molecular_interaction: entities from Molecular Interactions ontology (MI).
- crm2gene variable: relation between CRM and gene.
- gene2phen variable: relation between gene and phenotype.
- crm2phen variable: relation between CRM and phenotype.
- crm2tfac variable: relation between CRM and protein (transcription factor).
- Transcription factor variable: transcription factors (currently only proteins that interact with CRM).
- reg2targ variable: regulatory relation between proteins.

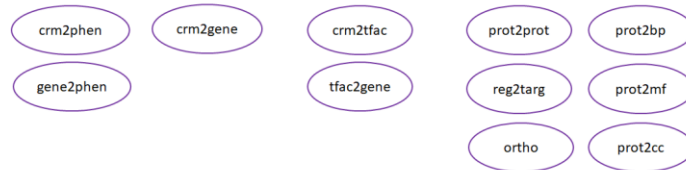
- prot2prot: molecular interaction relation between proteins.
- tfac2gene variable: relation between gene and protein.
- Database variable: databases.
- Chromosome variable: chromosomes.
- Reference_genome variable: genome assembly.
- TAD variable: topologically associated domain.
- Cellular_component variable: cellular components from Gene Ontology (GO).
- prot2cc variable: relation between protein and its cellular components.
- Molecular_function variable: molecular functions from GO.
- prot2mf: relation between protein and its molecular functions.
- Biological_process variable: biological processes from GO.
- prot2bp variable: relation between protein and its biological processes.
- Ortho variable: orthology relation between proteins.
- Root variable: top hierarchically class of NCBITaxon Ontology.
- Taxonomic_rank variable: top hierarchically class of NCBITaxon Ontology.

The properties are detailed with examples and their domains [here](#).

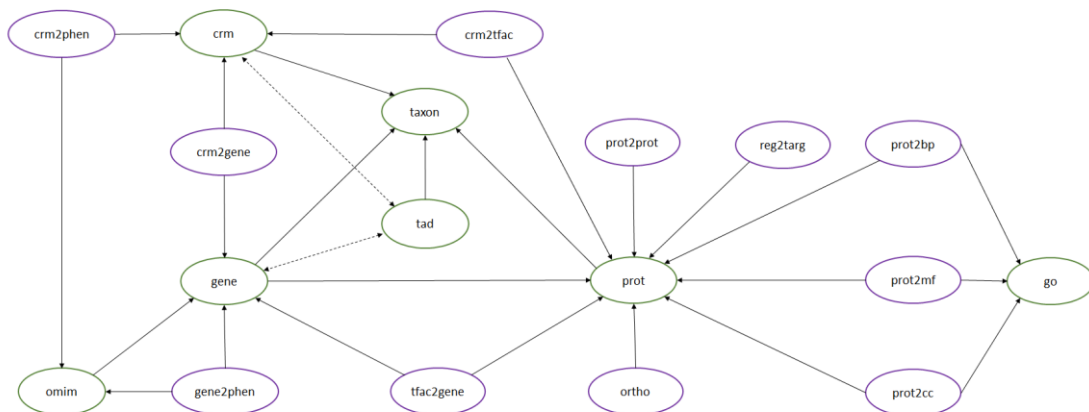
A-type grahs
(entities)



B-type graphs
(relations)



BGW
network



crm: cis regulatory domains (enhancers)
tad: topologically associating domains
gene: protein-coding genes
prot: proteins
omim: Online Mendelian Inheritance in Man
go: Gene Ontology
taxon: NCBITaxon Ontology

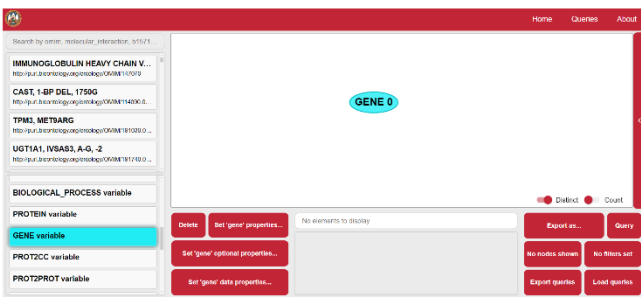
—→ links through properties
 - - - - - links through coordinate operations
 links through relation graphs

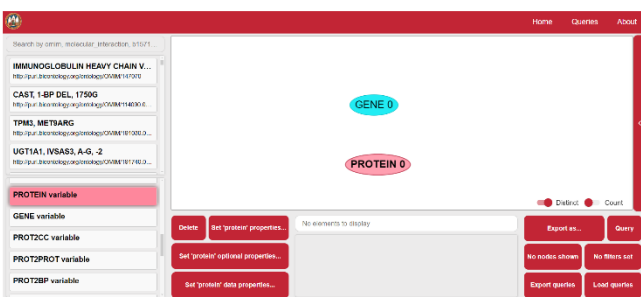
crm2gene: crm-target gene relations
crm2tfac: crm-dbTF interactions
crm2phen: crm-phenotype relations
gene2phen: gene-phenotype relations
prot2prot: protein interactions
prot2bp: protein-biological process relations
prot2mf: protein-molecular function relations
prot2cc: protein-cellular component relations
tfac2gene: transcription factor-target gene relations
reg2targ: protein-protein regulatory relations
ortho: protein-protein orthology relations

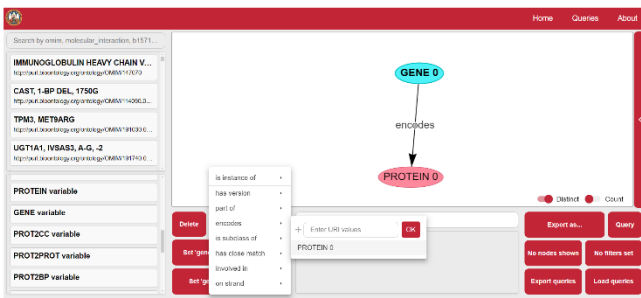
4. How to build a query

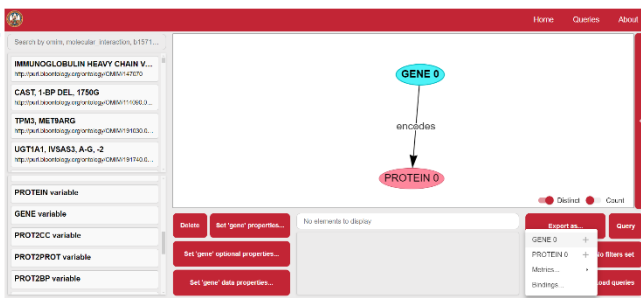
The query building process involves linking entities (nodes) through properties (edges). We take as an example the previous case: <Gene> <encodes> <Protein>.

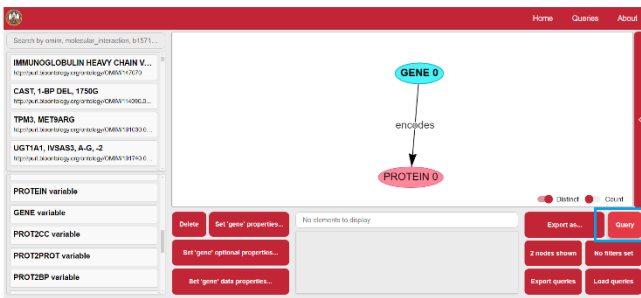
1. Select the first entity (subject node), in this case, Gene, in the Variable browser.
2. Select the second entity (object node), in this case, Protein, in the Variable browser.
3. Select the relation between both entities in the Properties selectors, in this case, "encodes".
4. Select in "Nodes shown" the data you want to show in the output.
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.

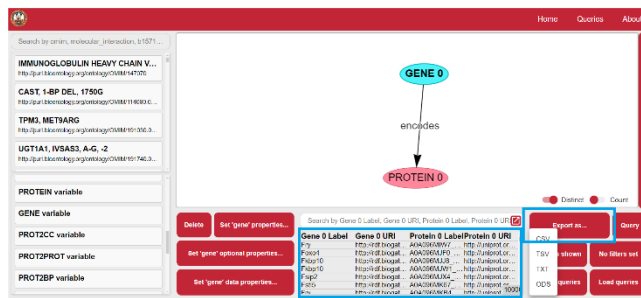
1. 

2. 

3. 

4. 

5. 

6. 

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

The screenshot shows the INTUITION web interface. On the left, there's a sidebar with search filters like 'IMMUNOGLOBULIN ...', 'CAST, 1-BP DEL, 17...', 'TPM3, MET9ARG', 'UGT1A1, IVSAS3, A-...', 'BIOLOGICAL_PROC...', 'PROTEIN variable', 'GENE variable', 'PROT2CC variable', and 'PROT2PROT variable'. The main area displays a graph with a blue node 'GENE 0' and a red node 'PROTEIN 0' connected by an 'encodes' relationship. Below the graph, there's a table with columns 'Gene 0 Label', 'Gene 0 URI', 'Protein 0 Label', and 'Protein 0 URI'. The table lists several genes and proteins, including Fry, Foxo4, Fkbp10, Fkbp10, Fsp12, and Fstl5. On the right, there's a console window showing a query and its results, including a response time of 33455 ms.

5. Data filtering and other possible operations

5.1. Filters of properties

Linking two biological entities or variables by their relation (object property) is the simplest filtering. However, any biological entity can also be filtered by its characteristics or attributes (data properties). The section "Property selectors" enables this task.

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 " \subseteq " to indicate a substring contained in the string. In addition, to make filtering easier for the user, the default configuration of INTUITION is not case-sensitive. 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.

The screenshot shows the INTUITION web interface. On the left, there's a sidebar with search filters like 'IMMUNOGLOBULIN HEAVY CHAIN V...', 'CAST, 1-BP DEL, 1750G', 'TPM3, MET9ARG', 'UGT1A1, IVSAS3, A-G, -2', 'DATABASE variable', 'CHROMOSOME variable', 'REFERENCE_GENOME variable', 'TAD variable', and 'CRM2PHEN variable'. The main area displays a graph with a yellow node 'CHROMOSOME 0' and a blue node 'CRM 0' connected by a 'part of' relationship. Below the graph, there's a table with columns 'Crm 0 Label' and 'Crm 0 URI'. The table lists several CRM labels and their corresponding URIs. On the right, there's a console window showing a query and its results, including a response time of 729 ms.

Node 'CRM 0' data properties

in taxon	=	show as...	Show in results:	<input type="checkbox"/>	Make transitive:	<input type="checkbox"/>	
is instance of	=	show as...	Show in results:	<input type="checkbox"/>	Make transitive:	<input type="checkbox"/>	
involved in pos...	=	show as...	Show in results:	<input type="checkbox"/>	Make transitive:	<input type="checkbox"/>	
has definition	=	show as...	Show in results:	<input type="checkbox"/>	Make transitive:	<input type="checkbox"/>	
start position	142353	>=	show as...	Show in results:	<input type="checkbox"/>	Make transitive:	<input type="checkbox"/>
has name	=	show as...	Show in results:	<input type="checkbox"/>	Make transitive:	<input type="checkbox"/>	
end position	230005	<=	show as...	Show in results:	<input type="checkbox"/>	Make transitive:	<input type="checkbox"/>
is defined by	=	show as...	Show in results:	<input type="checkbox"/>	Make transitive:	<input type="checkbox"/>	

Set properties

The node definition by the subClassOf property will be included

Cancel

Node 'CHROMOSOME 0' data properties

is instance of	=	show as...	Show in results:	<input type="checkbox"/>	Make transitive:	<input type="checkbox"/>	
has name	chr-16	=	show as...	Show in results:	<input type="checkbox"/>	Make transitive:	<input type="checkbox"/>
category	=	show as...	Show in results:	<input type="checkbox"/>	Make transitive:	<input type="checkbox"/>	

Set properties

The node definition by the subClassOf property will be included

Cancel

Variables can also be filtered clicking on “Filters set” button.

Some additional examples are given below:

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

The resources in RDF are represented by URIs, so they must be used when the filtering involves a resource. This is the case for filtering by taxon (*in taxon* property). Since operating with URIs can be tedious, e.g. http://purl.obolibrary.org/obo/NCBITaxon_9606, the "content in" or " \subseteq " operator makes it easier to work with identifiers. The following example illustrates the filtering of human genes.

The screenshot shows the INTUITION web interface. At the top, there is a navigation bar with 'Home', 'Queries', and 'About' links. Below the navigation bar is a search bar containing the text 'Search by omim, molecular_interaction, b1571...'. On the left side, there is a sidebar with a list of variables: 'IMMUNOGLOBULIN HEAVY CHAIN V...', 'CAST, 1-BP DEL, 1750G', 'TPM3, MET9ARG', 'UGT1A1, IVSAS3, A-G, -2', 'BIOLOGICAL_PROCESS variable', 'PROTEIN variable', 'GENE variable', 'PROT2CC variable', and 'PROT2PROT variable'. The main area displays a single node labeled 'GENE 0' in a light blue oval. Below the node, there is a message 'No elements to display'. At the bottom, there are several buttons: 'Delete', 'Set 'gene' properties...', 'Set 'gene' optional properties...', 'Set 'gene' data properties...', 'Export as...', 'Query', 'No nodes shown', 'No filters set', 'Export queries', and 'Load queries'. There are also radio buttons for 'Distinct' and 'Count'.

The screenshot shows a dialog box titled 'Node 'GENE 0' data properties'. The dialog box has a table with columns for property name, value, operator, show as..., show in results, and make transitive. The properties listed are: 'is instance of', 'in taxon', 'part of', 'has synonym', 'start position', 'end position', 'has name', and 'has definition'. The 'in taxon' property has a value of '9606'. The 'show in results' and 'make transitive' checkboxes are all unchecked. At the bottom, there is a 'Set properties' button, a 'Cancel' button, and a note: 'The node definition by the subClassOf property will be included'.

Property	Value	Operator	Show as...	Show in results	Make transitive
is instance of		=	show as...	<input type="checkbox"/>	<input type="checkbox"/>
in taxon	9606	⊆	show as...	<input type="checkbox"/>	<input type="checkbox"/>
part of		=	show as...	<input type="checkbox"/>	<input type="checkbox"/>
has synonym		=	show as...	<input type="checkbox"/>	<input type="checkbox"/>
start position		=	show as...	<input type="checkbox"/>	<input type="checkbox"/>
end position		=	show as...	<input type="checkbox"/>	<input type="checkbox"/>
has name		=	show as...	<input type="checkbox"/>	<input type="checkbox"/>
has definition		=	show as...	<input type="checkbox"/>	<input type="checkbox"/>

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

Chromosomes 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.

The screenshot shows the OMIM browser interface. On the left, there is a sidebar with a search bar and a list of variables: IMMUNOGLOBULIN HEAVY CHAIN V..., CAST, 1-BP DEL, 1750G, TPM3, MET9ARG, UGT1A1, IVSAS3, A-G, -2, DATABASE variable, CHROMOSOME variable, REFERENCE_GENOME variable, TAD variable, and CRM2PHEN variable. The main area displays a query graph with two nodes: 'CHROMOSOME 0' (yellow oval) and 'GENE 0' (blue oval). An arrow labeled 'part of' points from 'GENE 0' to 'CHROMOSOME 0'. Below the graph, there are buttons for 'Delete', 'No node selected', 'No elements to display', 'Export as...', 'Query', 'No nodes shown', 'No filters set', 'Export queries', and 'Load queries'. The interface also includes a 'Distinct' and 'Count' toggle.

Node 'CHROMOSOME 0' data properties

is instance of	=	show as...	Show in results:	<input type="checkbox"/>	Make transitive:	<input type="checkbox"/>	
has name	chr-1	=	show as...	Show in results:	<input type="checkbox"/>	Make transitive:	<input type="checkbox"/>
category		=	show as...	Show in results:	<input type="checkbox"/>	Make transitive:	<input type="checkbox"/>

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

Node 'GENE 0' data properties

is instance of	=	show as...	Show in results:	<input type="checkbox"/>	Make transitive:	<input type="checkbox"/>	
in taxon	9606	⊆	show as...	Show in results:	<input type="checkbox"/>	Make transitive:	<input type="checkbox"/>
part of		=	show as...	Show in results:	<input type="checkbox"/>	Make transitive:	<input type="checkbox"/>
has synonym		=	show as...	Show in results:	<input type="checkbox"/>	Make transitive:	<input type="checkbox"/>
start position		=	show as...	Show in results:	<input type="checkbox"/>	Make transitive:	<input type="checkbox"/>
end position		=	show as...	Show in results:	<input type="checkbox"/>	Make transitive:	<input type="checkbox"/>
has name		=	show as...	Show in results:	<input type="checkbox"/>	Make transitive:	<input type="checkbox"/>
has definition		=	show as...	Show in results:	<input type="checkbox"/>	Make transitive:	<input type="checkbox"/>

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

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

For this example we illustrate the query building to obtain the proteins encoded by the human TOX3 gene.

Node search is disabled temporally

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...

BIOLOGICAL_PROCESS variable

PROTEIN variable

GENE variable

PROT2CC variable

PROT2PROT variable

Delete Set 'gene' object properties

Set 'gene' optional properties

Set 'gene' data properties

Search by Gene 0 Label, Gene 0 URI, Protein 0 Label, Protein 0 URI

Gene 0 Label	Gene 0 URI	Protein 0 Label	Protein 0 URI
TOX3	http://rdf.biogate...J3QQQ6_HUMAN	http://uniprot.org...	
TOX3	http://rdf.biogate...H3BTZ9_HUMAN	http://uniprot.org...	
TOX3	http://rdf.biogate...TOX3_HUMAN	http://uniprot.org...	

Export as... Query

2 nodes shown No filters set

Export query Load query

Distinct Count

Node 'GENE 0' data properties

is instance of	=	show as...	Show in results:	<input type="checkbox"/>	Make transitive:	<input type="checkbox"/>	
in taxon	9606	=	show as...	Show in results:	<input type="checkbox"/>	Make transitive:	<input type="checkbox"/>
part of	=	show as...	Show in results:	<input type="checkbox"/>	Make transitive:	<input type="checkbox"/>	
has synonym	=	show as...	Show in results:	<input type="checkbox"/>	Make transitive:	<input type="checkbox"/>	
start position	=	show as...	Show in results:	<input type="checkbox"/>	Make transitive:	<input type="checkbox"/>	
end position	=	show as...	Show in results:	<input type="checkbox"/>	Make transitive:	<input type="checkbox"/>	
has name	TOX3	=	show as...	Show in results:	<input type="checkbox"/>	Make transitive:	<input type="checkbox"/>
has definition	=	show as...	Show in results:	<input type="checkbox"/>	Make transitive:	<input type="checkbox"/>	

Set properties

The node definition by the subClassOf property will be included

Cancel

5.2. Count and display unique results

The output table shows the biological entities consulted. Since the user has the freedom to choose which entities want to include in the output ("Nodes shown" button), the result can include duplicate entities. The Distinct button, activated by default, removes these duplicates. The Count button allows you to count the results obtained.

The screenshot shows the INTUITION web interface. On the left, there's a sidebar with a search bar and a list of variables: IMMUNOGLOBULIN HEAVY CHAIN V..., CAST, 1-BP DEL, 1750G, TPM3, MET9ARG, UGT1A1, IVSAS3, A-G, -2, BIOLOGICAL_PROCESS variable, PROTEIN variable, GENE variable, PROT2CC variable, and PROT2PROT variable. The main area displays a query result for 'GENE 0' encoding 'PROTEIN 0'. Below the diagram, there's a table with columns 'Gene 0 URI count' and 'Protein 0 URI count'. The table shows 255461 for Gene 0 and 525012 for Protein 0. On the right, there are buttons for 'Export as...', 'Query', '2 nodes shown', 'No filters set', 'Export queries', and 'Load queries'. A red box highlights the 'Distinct' and 'Count' buttons.

5.3. Multiple values

INTUITION also 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.

Since RDF resources are represented through URIs, the input values for filtering must be URIs. We are working on the use of labels for the next release version.

The screenshot shows the INTUITION web interface. On the left, there's a sidebar with a search bar and a list of variables: IMMUNOGLOBULIN HEAVY CHAIN V..., CAST, 1-BP DEL, 1750G, TPM3, MET9ARG, UGT1A1, IVSAS3, A-G, -2, ROOT variable, TAXONOMIC_RANK variable, TFAC2GENE variable, CRM2GENE variable, and CRM variable. The main area displays a query result for 'CRM 0' observed in two tissues: 'http://purl.obolibrary.org/obo/UBERON_0002107' and 'http://purl.obolibrary.org/obo/UBERON_0000948'. Below the diagram, there's a table with columns 'Gene 0 URI count' and 'Protein 0 URI count'. The table shows 255461 for Gene 0 and 525012 for Protein 0. On the right, there are buttons for 'Export as...', 'Query', '2 nodes shown', 'No filters set', 'Export queries', and 'Load queries'. A red box highlights the 'Distinct' and 'Count' buttons.

5.4. 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.

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...

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

Export as... Query

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 Count

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

Set bindings Cancel

Filters

length <= 5000 Filter length <= 5000 Add

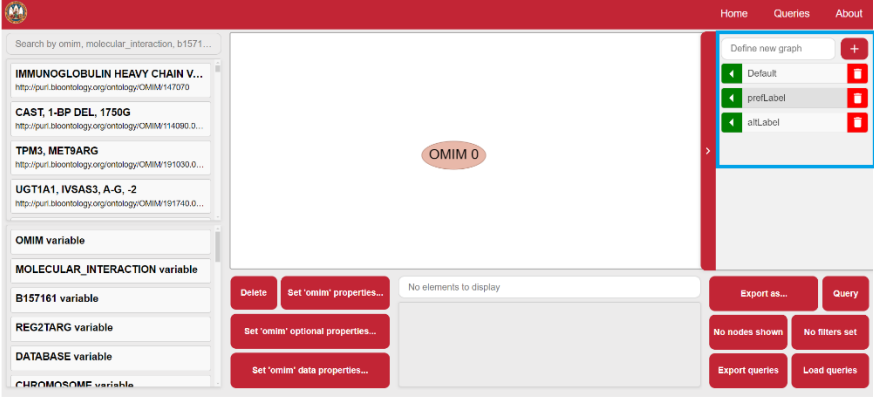
Set Filters Cancel

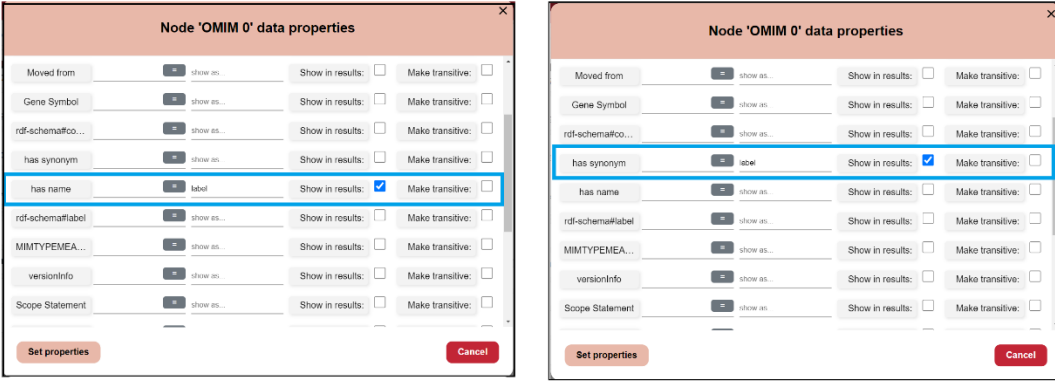
5.5. Union of queries

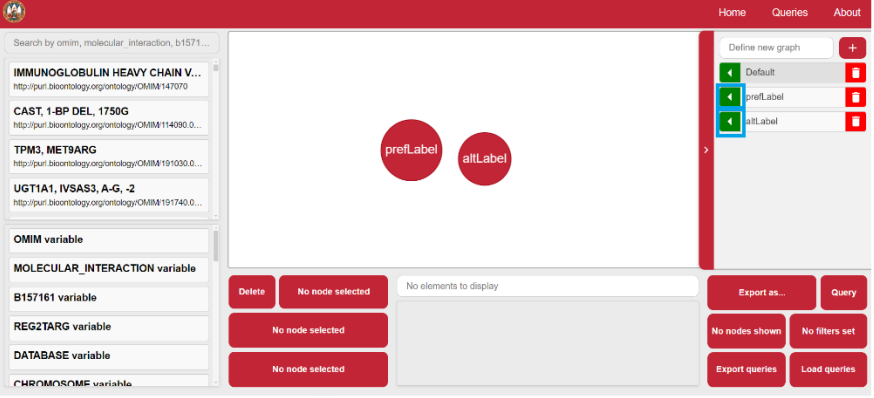
INTUITION also allows the use of the UNION clause of SPARQL. UNION merges subqueries through common variables in both queries. We illustrate its use through a use case. For example, we retrieve the OMIM entities that contain the string "breast cancer" as a name or synonym. To do that:

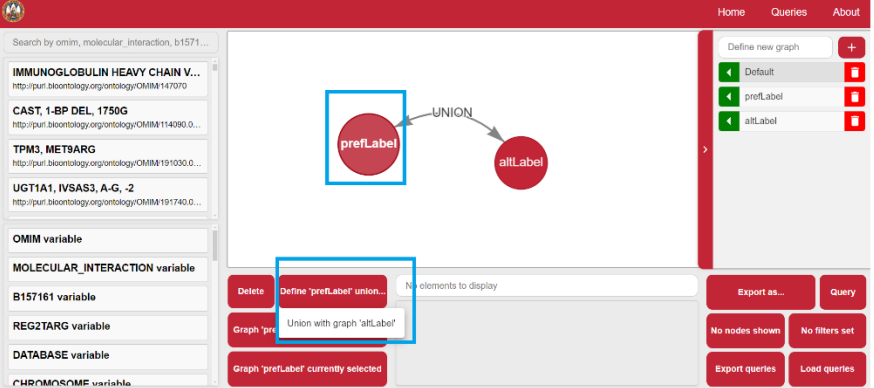
1. In the Graph builder section, we create the graphs belonging to each of the subqueries, and we include an OMIM node in each of them. In the example, the graphs are "prefLabel" for the main label query and "altLabel" for the query of the synonym.
2. In each of the graphs we define the variable "label"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

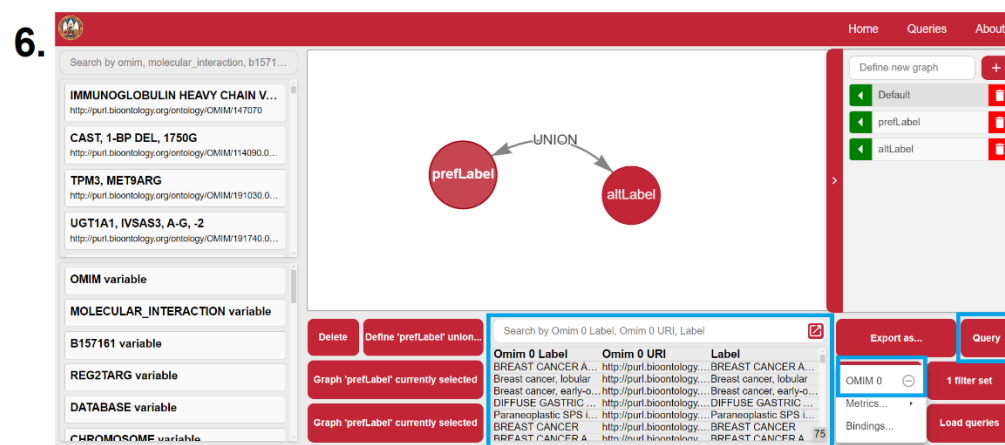
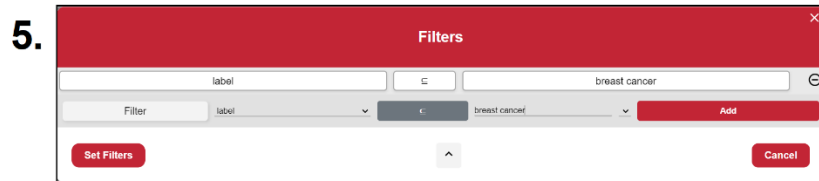
5. In "Node shown" we select the variables to be shown and in "Filters set" we filter the variable "label".
6. Run the query (Query).

1. 

2. 

3. 

4. 



6. Use Cases

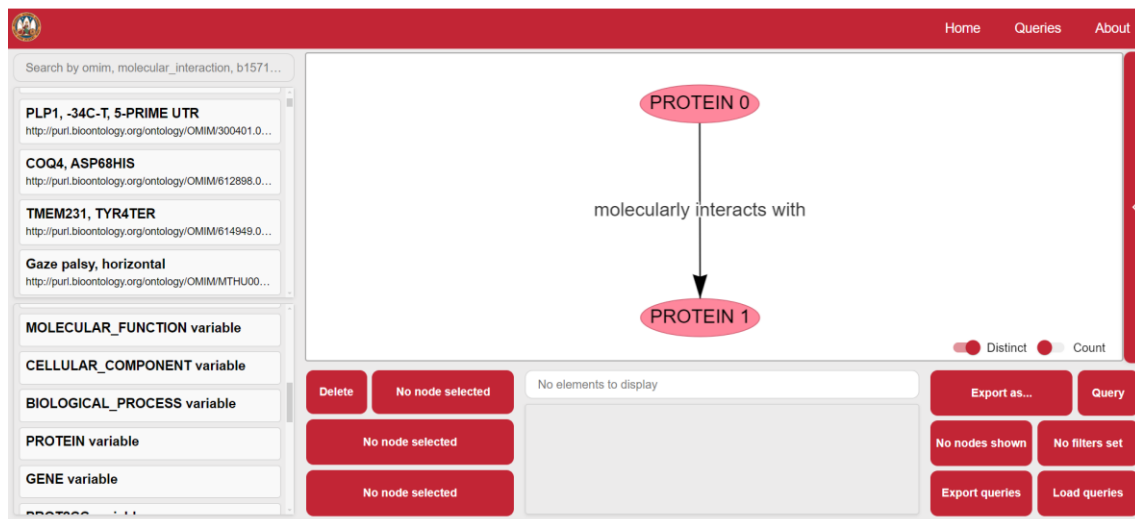
The following Use Cases were developed in the paper "*Analysis of the landscape of human enhancer sequences in biological databases*". The corresponding queries are attached for reproducibility and as examples of use. These use cases include complex queries that connect multiple nodes, use different 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](#).

7. Useful notes

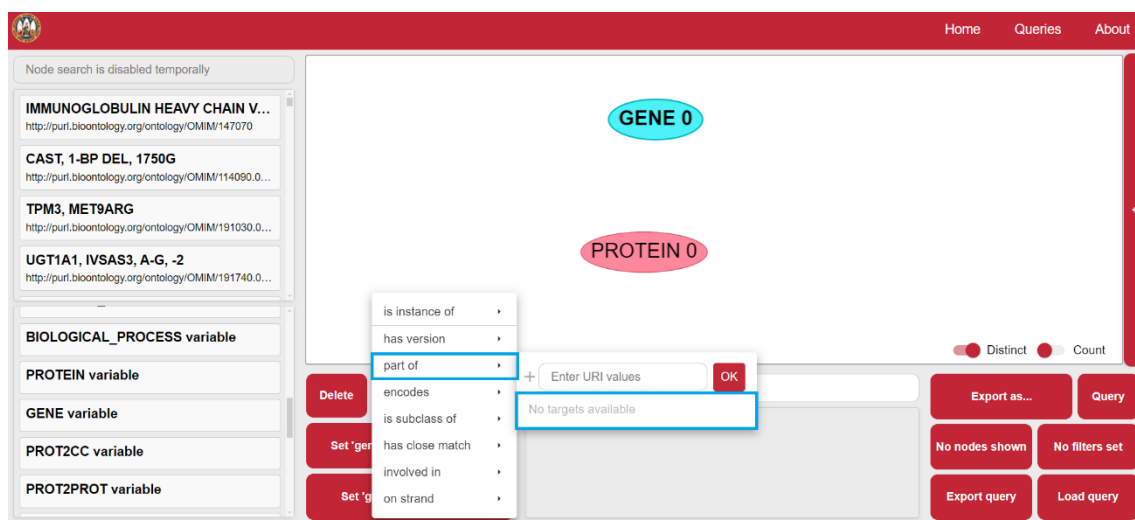
7.1. Numbering of variables

Queries can involve different variables. These queries can require the use of an entity as different variables, for example the protein-protein interaction involves two different proteins, and therefore two different variables. INTUITION permits to include the same variable more than once. These nodes are listed starting from 0. The counter is reset when the application is refreshed, not when the query is deleted.



7.2. Selection of object properties

In order to help the user select appropriate links between biological entities (nodes introduced by variables), INTUITION only allows to connect two nodes according to their appropriate property (object property). That is, INTUITION does not allow you to link two entities through an incorrect property. For example, if the variables Gene and Protein are introduced, INTUITION only allows connecting both variables through the "encodes" property.



[Home](#)
[Queries](#)
[About](#)

Node search is disabled temporarily

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...>

BIOLOGICAL_PROCESS variable

PROTEIN variable

GENE variable

PROT2CC variable

PROT2PROT variable

```

graph TD
    GENE0((GENE 0)) -- encodes --> PROTEIN0((PROTEIN 0))

```

☐ Distinct
☐ Count

Delete
Set 'gen
Set 'g

is instance of
has version
part of
encodes
is subclass of
has close match
involved in
on strand

+ Enter URI values
PROTEIN 0
OK

Export as...
Query

No nodes shown
No filters set

Export query
Load query