Assignment 5 SPARQL queries

I would like you to create the SPARQL query that will answer each of these questions.  Please submit the queries simply as a text document (NO programming is required!) - submit to GitHub as usual.

*For many of these you will need to look-up how to use the SPARQL functions ‘COUNT’ and ‘DISTINCT’ (we used ‘distinct’ in class), and probably a few others...*

-----------------------------------------

**UniProt SPARQL Endpoint:  http://sparql.uniprot.org/sparql/**

PREFIX up:<http://purl.uniprot.org/core/>

PREFIX keywords:<http://purl.uniprot.org/keywords/>

PREFIX uniprotkb:<http://purl.uniprot.org/uniprot/>

PREFIX taxon:<http://purl.uniprot.org/taxonomy/>

PREFIX ec:<http://purl.uniprot.org/enzyme/>

PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>

PREFIX skos:<http://www.w3.org/2004/02/skos/core#>

PREFIX owl:<http://www.w3.org/2002/07/owl#>

PREFIX bibo:<http://purl.org/ontology/bibo/>

PREFIX dc:<http://purl.org/dc/terms/>

PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>

PREFIX faldo:<http://biohackathon.org/resource/faldo#>

PREFIX GO:<http://purl.obolibrary.org/obo/GO\_>

PREFIX allie:<http://allie.dbcls.jp/>

PREFIX CHEBI:<http://purl.obolibrary.org/obo/CHEBI\_>

PREFIX chebihash:<http://purl.obolibrary.org/obo/chebi#>

PREFIX cco:<http://rdf.ebi.ac.uk/terms/chembl#>

PREFIX codoa:<http://purl.glycoinfo.org/ontology/codao#>

PREFIX ensembl:<http://rdf.ebi.ac.uk/resource/ensembl/>

PREFIX ensemblexon:<http://rdf.ebi.ac.uk/resource/ensembl.exon/>

PREFIX ensemblprotein:<http://rdf.ebi.ac.uk/resource/ensembl.protein/>

PREFIX ensemblterms:<http://rdf.ebi.ac.uk/terms/ensembl/>

PREFIX ensembltranscript:<http://rdf.ebi.ac.uk/resource/ensembl.transcript/>

PREFIX glycan:<http://purl.jp/bio/12/glyco/glycan#>

PREFIX glyconnect:<https://purl.org/glyconnect/>

PREFIX identifiers:<http://identifiers.org/>

PREFIX mesh:<http://id.nlm.nih.gov/mesh/>

PREFIX mnet:<https://rdf.metanetx.org/mnet/>

PREFIX mnx:<https://rdf.metanetx.org/schema/>

PREFIX orthodb:<http://purl.orthodb.org/>

PREFIX orthodbGroup:<http://purl.orthodb.org/odbgroup/>

PREFIX patent:<http://data.epo.org/linked-data/def/patent/>

PREFIX pubmed:<http://rdf.ncbi.nlm.nih.gov/pubmed/>

PREFIX rh:<http://rdf.rhea-db.org/>

PREFIX schema:<http://schema.org/>

PREFIX sh:<http://www.w3.org/ns/shacl#>

PREFIX sio:<http://semanticscience.org/resource/>

PREFIX slm:<https://swisslipids.org/rdf/>

PREFIX sp:<http://spinrdf.org/sp#>

PREFIX uberon:<http://purl.obolibrary.org/obo/uo#>

PREFIX uniprot:<http://purl.uniprot.org/uniprot/>

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

**1 POINT**  How many protein records are in UniProt?

Answer: 281303435

SELECT (STR(COUNT(?protein)) as ?count)

WHERE{?protein rdf:type up:Protein}

**1 POINT** How many Arabidopsis thaliana protein records are in UniProt?

Answer: 89182

SELECT ?taxon

WHERE{

?taxon a up:Taxon .

?taxon up:scientificName "Arabidopsis thaliana"

}

LIMIT 30

<http://purl.uniprot.org/taxonomy/3702>

SELECT (STR(COUNT(?protein)) as ?count)

WHERE{

?protein a up:Protein;

up:organism taxon:3702

}

LIMIT 30

**1 POINT**:  What is the description of the enzyme activity of UniProt Protein Q9SZZ8

Answer: Beta-carotene + 4 reduced ferredoxin [iron-sulfur] cluster + 2 H(+) + 2 O(2) = zeaxanthin + 4 oxidized ferredoxin [iron-sulfur] cluster + 2 H(2)O.

Process

SELECT distinct ?pred

WHERE

{ uniprot:Q9SZZ8 a up:Protein;

?pred ?o

}

LIMIT 30

<http://purl.uniprot.org/core/enzyme>

(I change ?pred to up:enzyme)

SELECT distinct ?pred

WHERE

{ uniprot:Q9SZZ8 a up:Protein;

up:enzyme ?enzyme.

?enzyme ?pred ?o

}

LIMIT 30

<http://purl.uniprot.org/core/activity>

(I change ?pred to up:activity)

SELECT ?o

WHERE

{ uniprot:Q9SZZ8 a up:Protein;

up:enzyme ?enzyme.

?enzyme up:activity ?o # I’ll change ?o 🡪 ?activity

}

http://purl.uniprot.org/enzyme/1.14.15.24#SIPF8A63F68B2741FFE

SELECT ?pred

WHERE

{ uniprot:Q9SZZ8 a up:Protein;

up:enzyme enz:1.14.15.24.

enz:1.14.15.24 up:activity ?activity.

?activity ?pred ?a # I’ll change ?a 🡪 ?activity\_label

}

<http://www.w3.org/2000/01/rdf-schema#label>

(I change ?pred to rdfs:label)

SELECT ?activity\_label

WHERE

{

uniprot:Q9SZZ8 a up:Protein ;

up:enzyme ?enzyme.

?enzyme up:activity ?activity.

?activity rdfs:label ?activity\_label

}

**1 POINT**:  Retrieve the proteins ids, and date of submission, for proteins that have been added to UniProt this year   (HINT Google for “SPARQL FILTER by date”)

Answer: 58202805

SELECT ?id (COUNT(?date) AS ?date\_count)

WHERE {

?protein a up:Protein ;

up:created ?date

FILTER ( ?date >= "2019-01-01"^^xsd:date)

}

**1 POINT** How  many species are in the UniProt taxonomy?

Answer: 1766921

SELECT distinct ?pred

FROM <http://sparql.uniprot.org/taxonomy>

WHERE{

?taxon a up:Taxon;

?pred ?o

}

LIMIT 30

<http://purl.uniprot.org/core/rank> (I change ?pred to up:rank)

SELECT distinct ?o

FROM <http://sparql.uniprot.org/taxonomy>

WHERE{

?taxon a up:Taxon;

up:rank ?o

}

LIMIT 30

<http://purl.uniprot.org/core/Species> (I change ?o to up:species)

SELECT (STR(COUNT(DISTINCT ?taxon)) AS ?count) #Here I do the count

FROM <http://sparql.uniprot.org/taxonomy>

WHERE{

?taxon a up:Taxon ;

up:rank up:Species

}

**1 POINT**  How many species have at least one protein record?

Answer: 984622

SELECT (STR(COUNT(DISTINCT ?taxon)) AS ?count) #No intermediate steps cause

WHERE I reuse the same prefixes that I

{ already know

?protein a up:Protein ;

up:organism ?taxon .

?taxon up:rank up:Species

}

**From the Atlas gene expression database SPARQL Endpoint: http://www.ebi.ac.uk/rdf/services/atlas/sparql**

IM JUST ADDING THIS PREFIX TO THE ONES I HAD, BECAUSE I HAD TO USE SOME OF THE UNIPROT PREFIXES TOO

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX owl: <http://www.w3.org/2002/07/owl#>

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

PREFIX dc: <http://purl.org/dc/elements/1.1/>

PREFIX dcterms: <http://purl.org/dc/terms/>

PREFIX dbpedia2: <http://dbpedia.org/property/>

PREFIX dbpedia: <http://dbpedia.org/>

PREFIX foaf: <http://xmlns.com/foaf/0.1/>

PREFIX skos: [http://www.w3.org/2004/02/skos/core#](http://www.w3.org/2004/02/skos/core)

**1 POINT** What is the Affymetrix probe ID for the Arabiodopsis Apetala3 gene?  (HINT - you cannot answer this directly from Atlas - you will first have to look at what kinds of database cross-references are in Atlas, and then construct the appropriate URI for the Apetala3 gene based on its ID number in \*that\* database)

#First, I tried in the uniport database

Answer: NO ANSWER (email)

SELECT distinct ?pred

WHERE {

?protein a up:Protein ;

up:organism taxon:3702;

?pred ?o

}

LIMIT 30

<http://purl.uniprot.org/core/recommendedName> (change ?pred to up:recommendedName and ?o to ?name) ?name gives a list of names when I ask for it

SELECT distinct ?pred

WHERE {

?protein a up:Protein ;

up:organism taxon:3702;

up:recommendedName ?name .

?name ?pred ?o

}

<http://purl.uniprot.org/core/fullName> (change ?pred to up:fullName and ?o to ?full) ?full gives a list of names such as “Putative acyl-activating enzyme 19”, “ADP,ATP carrier protein 3”, “mitochondrial”, “Probable acyl-activating enzyme 1”, “peroxisomal”,Serine/threonine protein phosphatase 2A 55 kDa regulatory subunit B alpha isoform” or “Aldehyde oxidase 4”.

SELECT distinct ?full

WHERE {

?protein a up:Protein ;

up:organism taxon:3702;

up:recommendedName ?name .

?name up:fullName ?full

FILTER CONTAINS( ?full, 'APETALA 3') .

}

LIMIT 30

Floral homeotic protein APETALA 3

Now I know that APETALA3 is in uniport, I change the endpoint to http://www.ebi.ac.uk/rdf/services/atlas/sparql and I add the service uniprot

SELECT ?full

WHERE

{

SERVICE<http://sparql.uniprot.org/sparql>

{

?protein a up:Protein ;

up:organism taxon:3702 ;

up:recommendedName ?name .

?name up:fullName ?full .

FILTER CONTAINS( ?full, 'APETALA 3') .

}

}

I got the same response than before: Floral homeotic protein APETALA 3, so I proceed to set the conditions for atlas 3 for the probe ID

SELECT distinct ?pred

WHERE

{

SERVICE<http://sparql.uniprot.org/sparql>

{

?protein a up:Protein ;

up:organism taxon:3702 ;

up:recommendedName ?name .

?name up:fullName ?full .

FILTER CONTAINS( ?full, 'APETALA 3') .

}

?probe ?pred ?o

}

I asked for all the predicates and got a list of more than 200, but none of them had a result when I asked for the predicates.

**3 POINTS** - get the experimental description for all experiments where the Arabidopsis Apetala3 gene is DOWN regulated

SELECT ?description

WHERE

{

?experiment atlasterms:hasAnalysis ?analysis ;

dcterms:description ?description .

?analysis atlasterms:hasExpressionValue ?expression .

?expression atlasterms:isMeasurementOf ?probe ;

sio:SIO\_000300 ?value .

FILTER REGEX(?value, 'DOWN') .

?probe atlasterms:dbXref ?protein .

SERVICE <http://sparql.uniprot.org/sparql>

{

?protein a up:Protein ;

up:organism taxon:3702 ;

up:recommendedName ?name .

?name up:fullName ?full .

FILTER REGEX( ?full, 'APETALA 3') .

}

}

**From the REACTOME database SPARQL endpoint:  http://www.ebi.ac.uk/rdf/services/reactome/sparql**

I ALSO ADDED THIS TWO TO THE ONES I HAD

PREFIX biopax3: <http://www.biopax.org/release/biopax-level3.owl#>

PREFIX tax:<http://identifiers.org/taxonomy/>

**2 POINTS**:  How many REACTOME pathways are assigned to Arabidopsis (taxon 3702)? (note that REACTOME uses different URLs to define their taxonomy compared to UniProt, so you will first have to learn how to structure those URLs….)

<https://www.ebi.ac.uk/rdf/services/sparql>

I saw SPARQL examples and I clicked on REACTOME (query 1), and it gave me:

SELECT DISTINCT ?pathway ?pathwayname

WHERE

{

?pathway rdf:type biopax3:Pathway .

?pathway biopax3:displayName ?pathwayname

}

Answer: 809

SELECT DISTINCT ?pred

WHERE

{

?pathway a biopax3:Pathway .

?pathway biopax3:displayName ?pathwayname .

?pathway ?pred ?o

}

http://www.biopax.org/release/biopax-level3.owl#organism

(now I change ?pred to biopax3:organism and I asked for ?o, and this were the results

<http://identifiers.org/taxonomy/3702>

<http://identifiers.org/taxonomy/9913>

<http://identifiers.org/taxonomy/6239>

http://identifiers.org/taxonomy/9615

As is indicated in the enunciated, I chose the taxon 3702 that corresponds to Arabidopsis, and I proceed to count the results

SELECT (COUNT (DISTINCT ?pathway) AS ?count)

WHERE

{

?pathway a biopax3:Pathway ;

biopax3:organism tax:3702

}

**3 POINTS**:  get all PubMed references for the pathway with the name “Degradation of the extracellular matrix”

Answer: all references in a list

SELECT DISTINCT ?pred

WHERE

{

?pathway a biopax3:Pathway .

?pathway biopax3:displayName ?name .

?pathway ?pred ?o

}

http://www.biopax.org/release/biopax-level3.owl#xref

(I change ?pred to biopax3:xref and ?o to ?ref)

SELECT DISTINCT ?pred

WHERE

{

?pathway a biopax3:Pathway .

?pathway biopax3:displayName ?name .

?pathway biopax3:xref ?ref .

?ref ?pred ?o

}

http://www.biopax.org/release/biopax-level3.owl#db (To the the pubmed database. Now I change ?pred to biopax3:db and ?o to ?db)

When I look at the different databases, these were the results:

* "GENE ONTOLOGY"^^xsd:string
* "Reactome Database ID Release 61"^^xsd:string
* “Reactome"^^xsd:string
* "Pubmed"^^xsd:string
* "ISBN"^^xsd:string

Now I add the ?predicate id

http://www.biopax.org/release/biopax-level3.owl#id

(I also asked for the ids and I found many different ids, numbers and others like “R-BTA-975956"^^xsd:string, "R-ATH-983168"^^xsd:string, "GO:0000209"^^xsd:string or "9853369"^^xsd:string)

All are strings, that’s why I use the str() while filtering in the next query. I filter the db and I reveive ids like: "16807135"^^xsd:string or "12692549"^^xsd:string. Among all of them, I filter by ?name and I put the string that is in the enunciate of the exercise “Degradation of the extracellular matrix” giving me the final query.

SELECT DISTINCT (str(?Id) AS ?pubmed\_ID)

WHERE

{

?pathway a biopax3:Pathway ;

biopax3:displayName ?name ;

biopax3:xref ?ref .

?red biopax3:db ?db ;

biopax3:id ?Id .

FILTER(str(?db) ='Pubmed') .

FILTER(str(?name) = 'Degradation of the extracellular matrix')

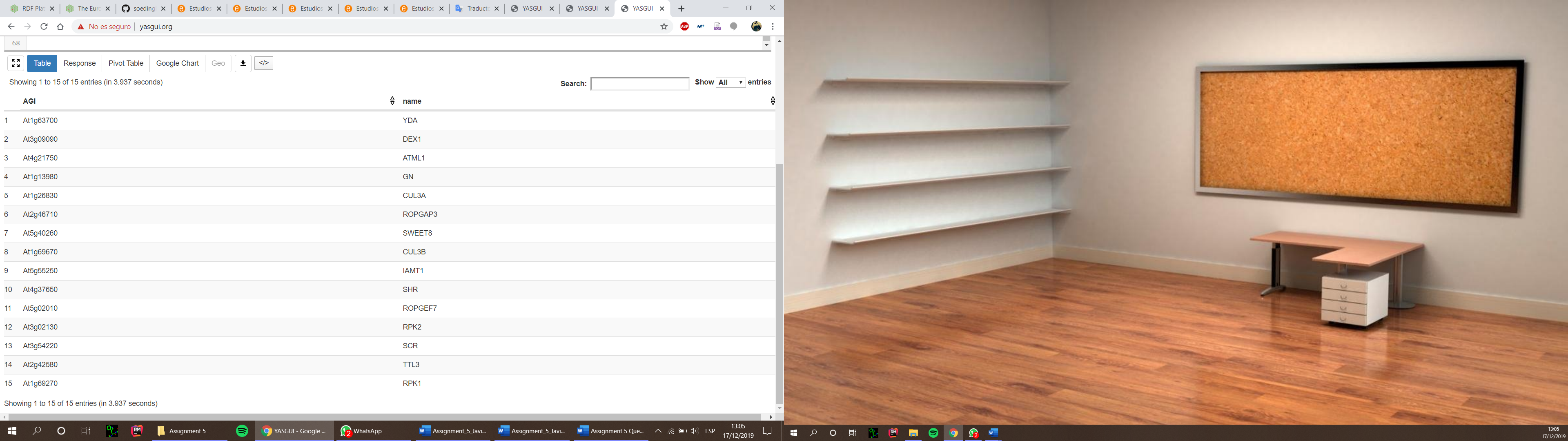
}

LIMIT 100 # I put this limit to ensure that I’m receiving the output correctly, in order to get all, don’t put the limit

# **BONUS QUERIES**

**UniProt BONUS 2 points**:  find the AGI codes and gene names for all Arabidopsis thaliana  proteins that have a protein function annotation description that mentions “pattern formation”

Answer: table with AGIs and the name of the gene



SELECT distinct ?pred

WHERE

{

?protein a up:Protein ;

up:organism taxon:3702 ;

?pred ?o

}

LIMIT 30

http://purl.uniprot.org/core/encodedBy (I change ?pred to up:encodedBy and ?o to ?gene)

Repeat for another predicate

http://purl.uniprot.org/core/annotation (I change ?pred to up:annotation and ?o to ?annotation)

SELECT distinct ?pred ?o

WHERE

{

?protein a up:Protein ;

up:organism taxon:3702 ;

up:encodedBy ?gene ;

up:annotation ?annotation .

?gene ?pred ?o

}

LIMIT 30

http://purl.uniprot.org/core/locusName (I change ?pred to up:locusName and ?o to ?AGI)

http://www.w3.org/2004/02/skos/core#prefLabel (I change ?pred to up:prefLabel and ?o to ?name (the label of the gene is its gene name)

SELECT ?o

WHERE

{

?protein a up:Protein ;

up:organism taxon:3702 ;

up:encodedBy ?gene ;

up:annotation ?annotation .

?gene up:locusName ?AGI ;

skos:prefLabel ?name .

?annotation a ?o

}

LIMIT 30

?o could be Transmembrane\_annotation, Subunit\_annotation, Binding\_site, Sequence, Region, Subcellular\_location, Motif, Domain, etc. But we need the function annotation:

http://purl.uniprot.org/core/Function\_Annotation (Change ?o to up:Function\_Annotation)

SELECT ?pred ?o

WHERE

{

?protein a up:Protein ;

up:organism taxon:3702 ;

up:encodedBy ?gene ;

up:annotation ?annotation .

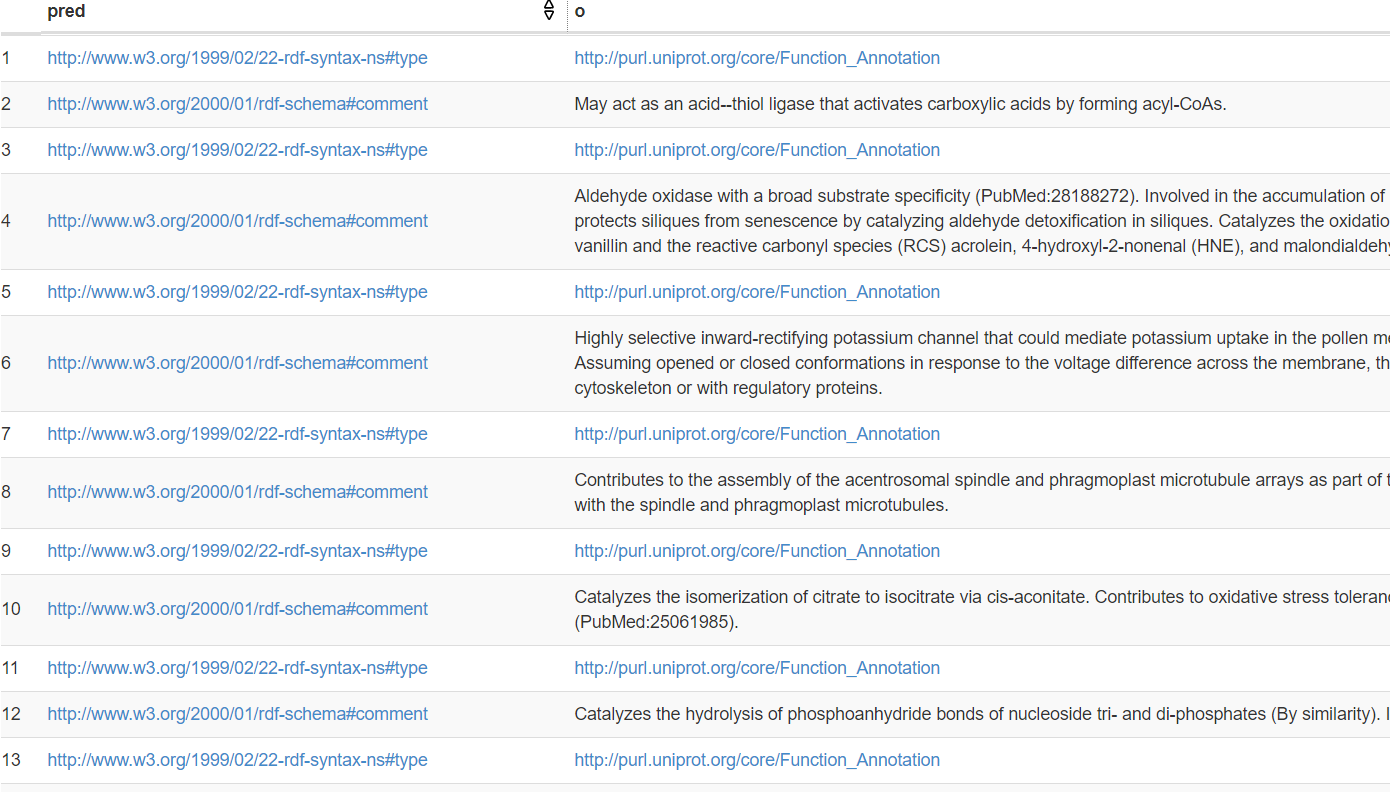
?gene up:locusName ?AGI ;

skos:prefLabel ?name .

?annotation a up:Function\_Annotation ;

?pred ?o

}



If I ask for the ?pred and the ?o, we can see how we have type and comments, we need the comments that have text to filter them.

SELECT ?AGI ?name

WHERE

{

?protein a up:Protein ;

up:organism taxon:3702 ;

up:encodedBy ?gene ;

up:annotation ?annotation .

?gene up:locusName ?AGI ;

skos:prefLabel ?name .

?annotation a up:Function\_Annotation ;

rdfs:comment ?Comment .

FILTER CONTAINS(?Comment, 'pattern formation')

}

This is the final query

**REACTOME BONUS 2** **points**:  write a query that proves that all Arabidopsis pathway annotations in Reactome are “inferred from electronic annotation” (evidence code)  (...and therefore are probably garbage!!!)

Answer: count1: 654481 // count2: 654481

THIS MEANS THAT THE 100% of the pathways annotations are “inferred from electronic annotation”

SELECT (count(?pathway1) as ?Count1) (count(?term) as ?count2)

WHERE

{

?pathway1 a biopax3:Pathway ;

biopax3:organism taxon:3702.

?pathway2 a biopax3:Pathway ;

biopax3:organism taxon:3702;

biopax3:evidence ?evidence .

?evidence biopax3:evidenceCode ?evidenceCode .

?evidenceCode biopax3:term ?term .

}