SADI

Find. Integrate. Analyze.

Semantic Automated Discovery and Integration

http://sadiframework.org

Summary

- SADI is a **set of conventions** for creating Semantic Web Services that can be **automatically discovered and orchestrated**.
- SADI does not create new technologies or message formats. It relies on well-established standards: RDF, OWL and HTTP.
- SADI service consumes an RDF graph with a designated node and produces an RDF graph about the same node with some **new properties attached**.
- Declaration of the new property predicates describes the semantics of the service and makes it *discoverable*.

Terminology

- XML and XML Schema
- Simple Object Access Protocol (SOAP)
- Resource Description Framework (RDF)
  - Universal Resource Identifiers (URIs)
- Web Ontology Language (OWL)
- HTTP GET and POST

Web Services

vs.

Semantic Web

Web Services

XML + XML Schema

Semantic Web

RDF + OWL

Web Services

POST of SOAP-XML

Semantic Web

GET of RDF-XML

Web Services

No (rigorous) semantics

Semantic Web

Rich, flexible semantics

Web Services

&

Semantic Web

Fundamentally different technologies!

By BIN HE, MITESH PATEL, ZHEN ZHANG, *and* KEVIN CHEN-CHUAN CHANG

ACCESSING THE DEEP WEB

Attempting to locate and quantify material on the Web that is hidden from typical search techniques.

The Web has been rapidly "deepened" by massive databases online and current search engines do not reach most of the data on the Internet [4]. While the surface Web has linked billions of static HTML pages, a far more significant amount of information is believed to be "hidden" in the deep Web, behind the query forms of searchable databases, as Figure 1(a) conceptually illustrates. Such information may not

>1000 X more data in the Deep Web than in Web pages

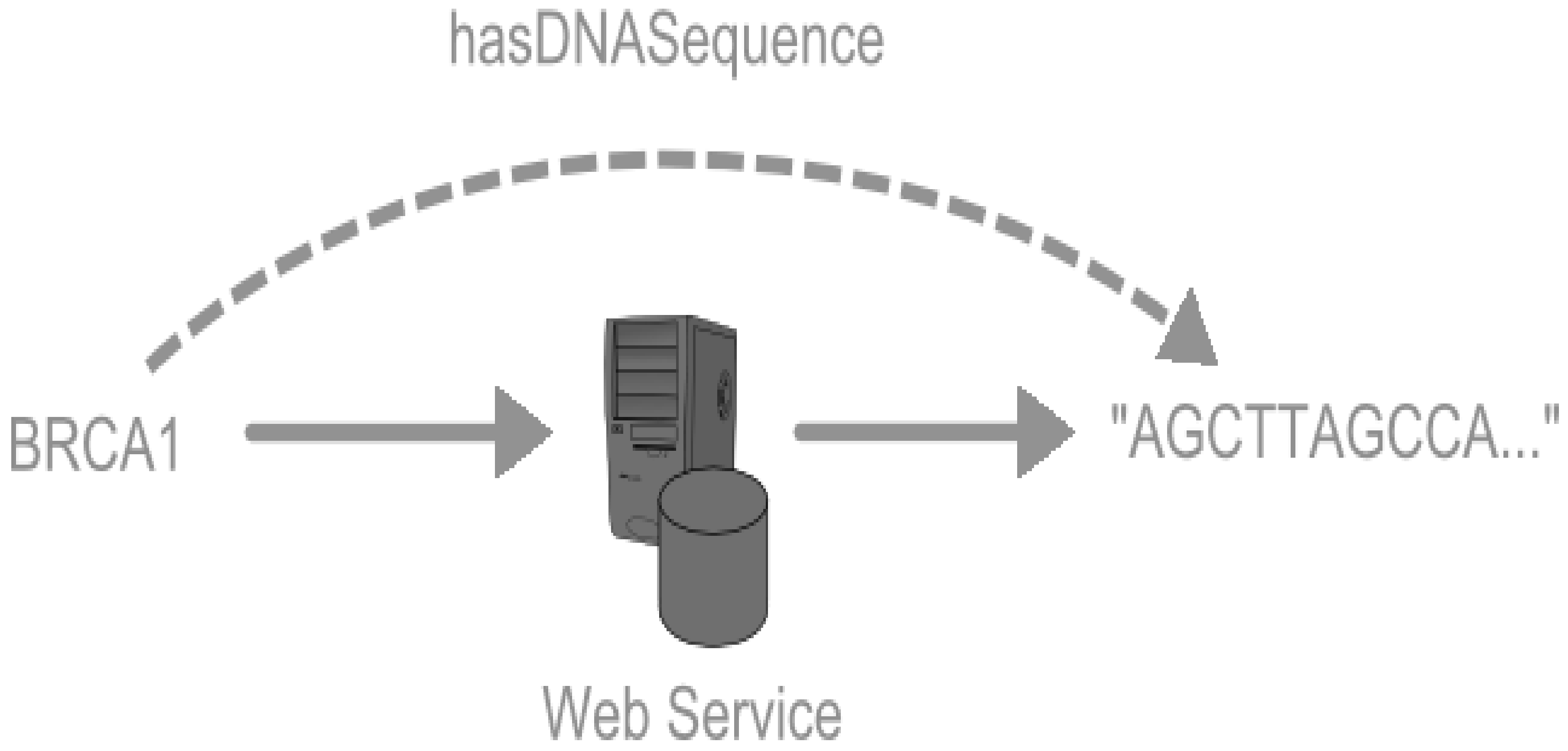
In bioinformatics this is primarily databases and analytical algorithms

Web Service output is ***critical to success*** for the Semantic Web!!

SADI

- Based on the observation of usage and behaviour of BioMoby Semantic Web Services Since 2002
- Standards-compliant
- Lightweight with only 2 “rules”

What [most] bioinformatics Web Services do



SADI “rules” a.k.a key practices

1. Make the implicit explicit.
   - All service input and output data are RDF instances of OWL classes
2. The URI of the input must be preserved in the output.
   - All URIs are “annotated” where the input becomes decorated by additional information instead of replaced

Consequence

“Semantics” of the interactions are now explicit

“Semantics” of HTTP POST are identical to the “Semantics” of HTTP GET

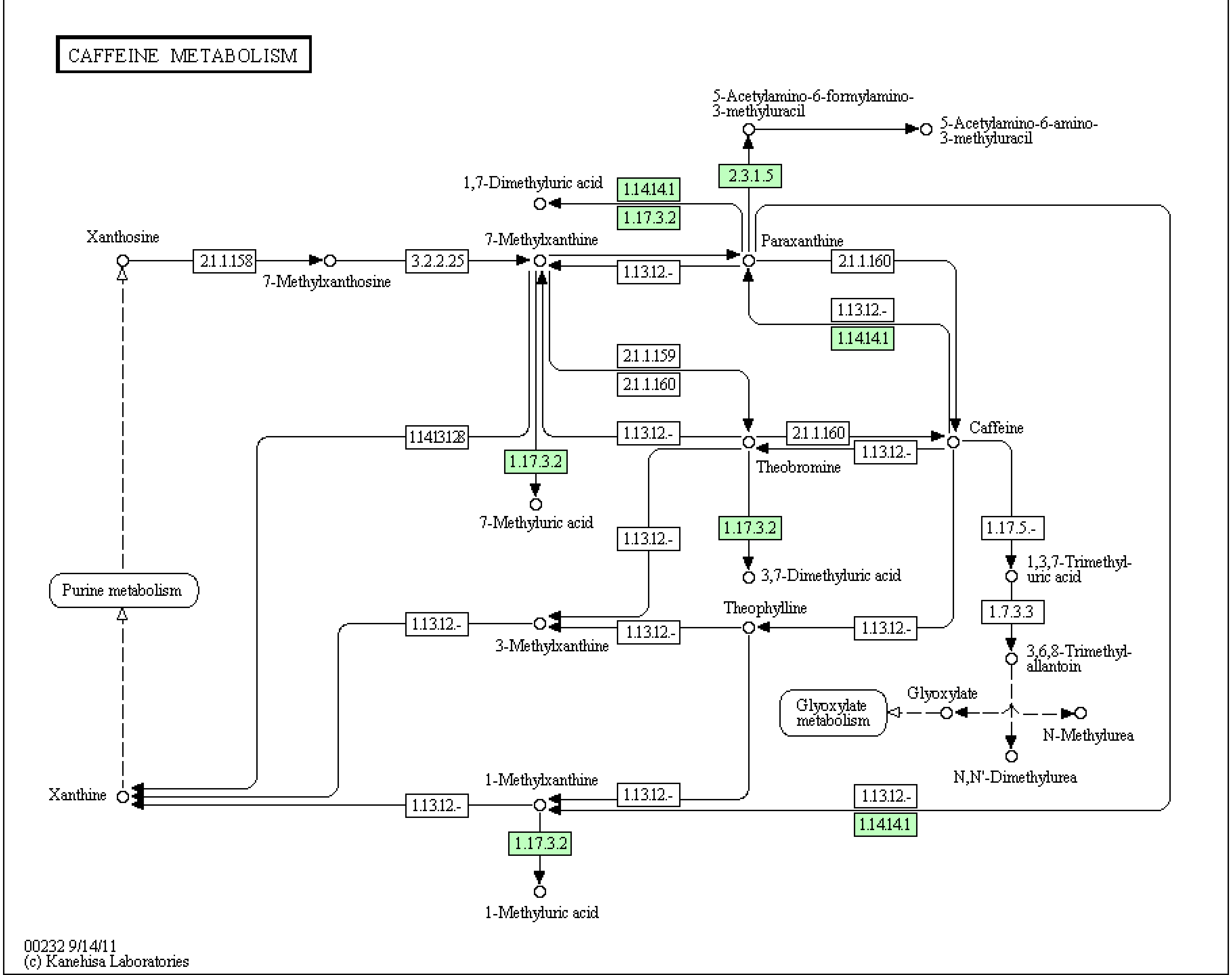
Therefore SADI Web Services behave like the Semantic Web

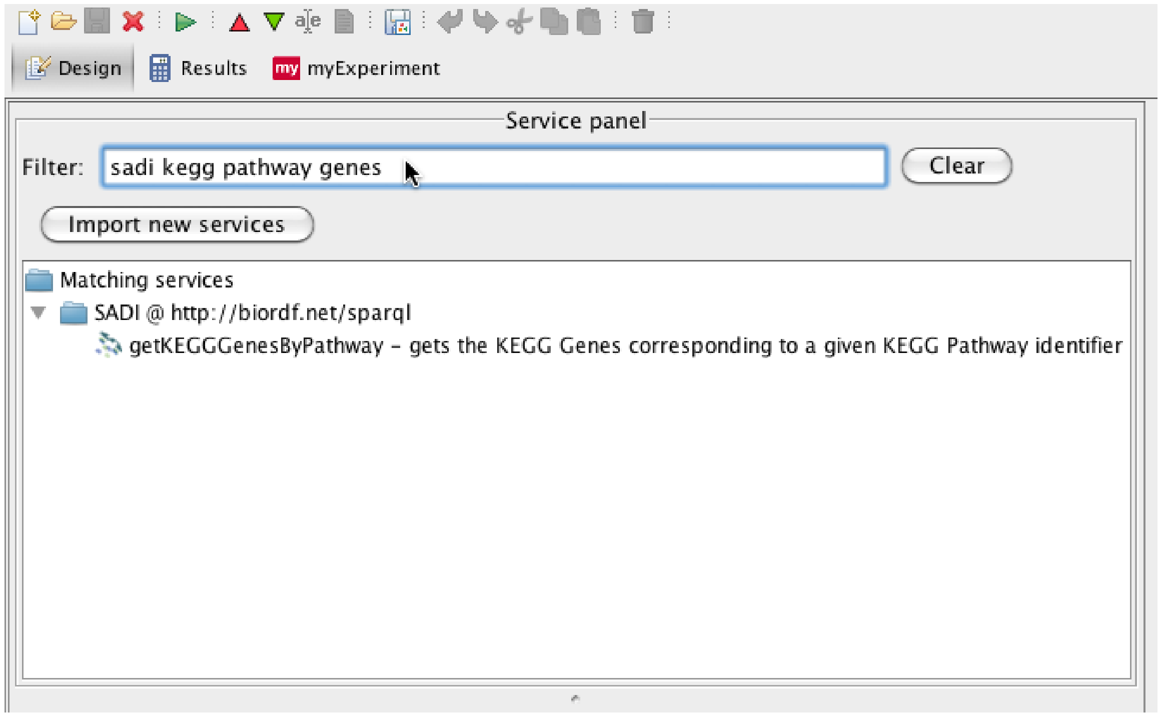
SADI Service plug-in and client

1. SADI plug-in to Taverna
   – A general-purpose workflow design tool designed to manage most Web Service, and handle data flow related to any domain of investigation.
2. Semantic Health And Research Environment (SHARE) query client

SADI in Taverna

- Example:
  - What genes are involved in KEGG pathway "hsa00232"? What proteins do those genes code for? What are the sequences of those proteins?


CAFFEINE METABOLISM
5-Acetylamino-6-formylamino-3-methyluracil
5-Acetylamino-6-amino-3-methyluracil
2.3.1.5
1,7-Dimethyluric acid
1.14.14.1
1.17.3.2
Xanthosine
2.1.1.158
7-Methylxanthosine
3.2.2.25
7-Methylxanthine
1.13.12.-
Paraxanthine
2.1.1.160
1.13.12.-
1.14.14.1
2.1.1.159
2.1.1.160
1.14.13.128
1.13.12.-
2.1.1.160
Caffeine
1.13.12.-
Theobromine
1.17.3.2
7-Methyluric acid
1.13.12.-
1.17.3.2
3,7-Dimethyluric acid
1.17.5.-
1,3,7-Trimethyluric acid
Purine metabolism
Theophylline
1.13.12.-
3-Methylxanthine
1.13.12.-
1.13.12.-
1.7.3.3
3,6,8-Trimethylallantoin
Glyoxylate
Glyoxylate metabolism
N-Methylurea
N,N'-Dimethylurea
Xanthine
1-Methylxanthine
1.13.12.-
1.13.12.-
1.13.12.-
1.14.14.1
1.17.3.2
1-Methyluric acid
00232 9/14/11
(c) Kanehisa Laboratories

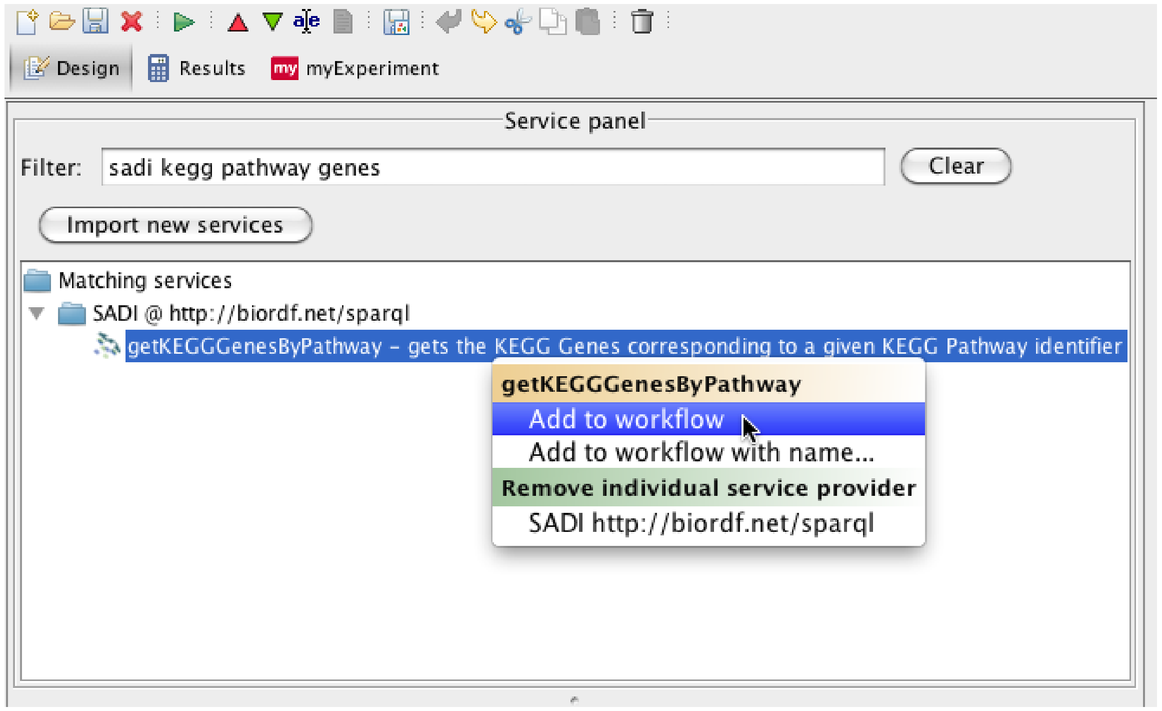


Filter: sadi kegg pathway genes

Matching services
SADI @ http://biordf.net/sparql
getKEGGGenesByPathway – gets the KEGG Genes corresponding to a given KEGG Pathway identifier

Using SADI services – building a workflow

Type ***sadi kegg pathway genes*** into the Service panel **Filter**.



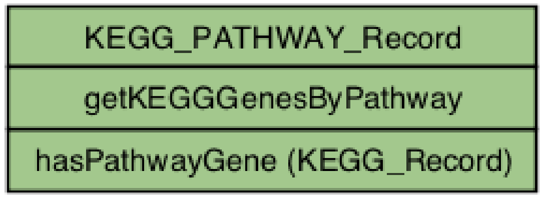
sadi kegg pathway genes

Matching services
SADI @ http://biordf.net/sparql
getKEGGGenesByPathway - gets the KEGG Genes corresponding to a given KEGG Pathway identifier

getKEGGGenesByPathway
SADI http://biordf.net/sparql

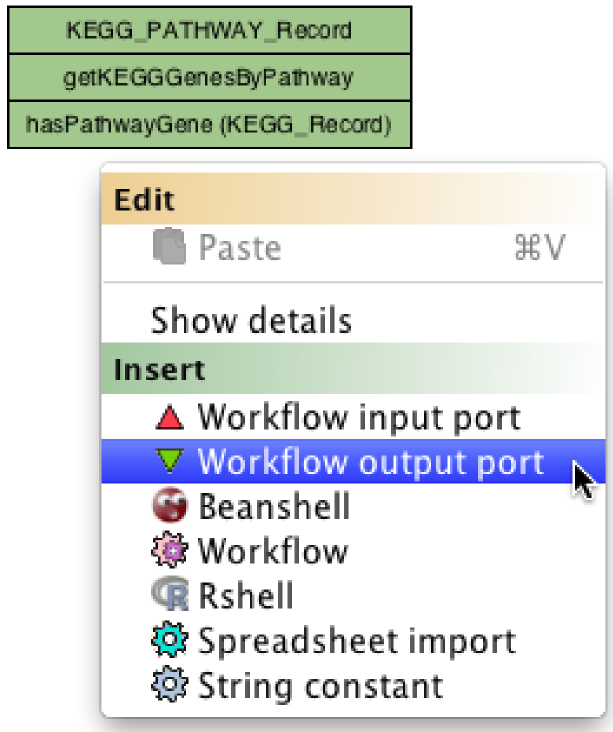
Using SADI services – building a workflow

Right click on the **getKEGGGenesByPathway** service and click **Add to workflow**.



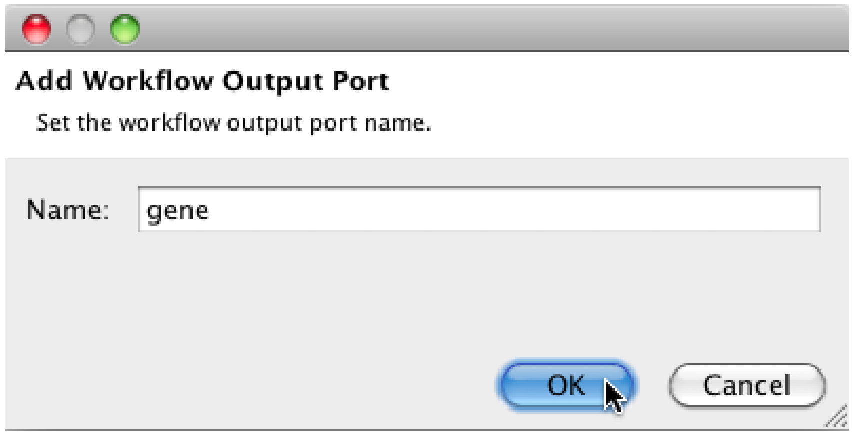
Using SADI services – building a workflow

The service input and output ports are now shown in the diagram.



Using SADI services – building a workflow

To add an output to the workflow right-click on the workflow diagram and click **Workflow output port**.

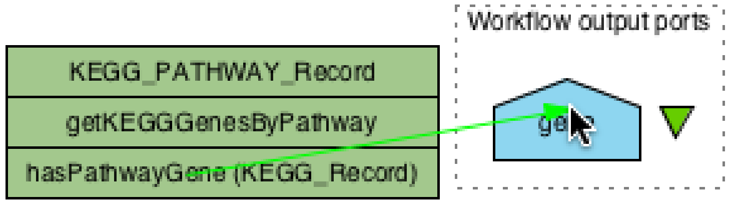

Add Workflow Output Port

Set the workflow output port name.

Name: gene

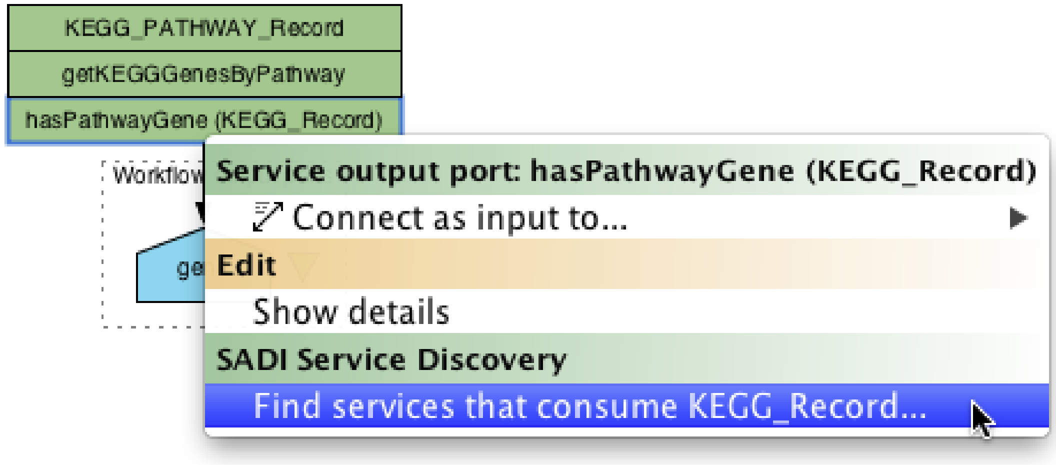
Using SADI services – building a workflow

Name the output port **gene** and click **OK**.



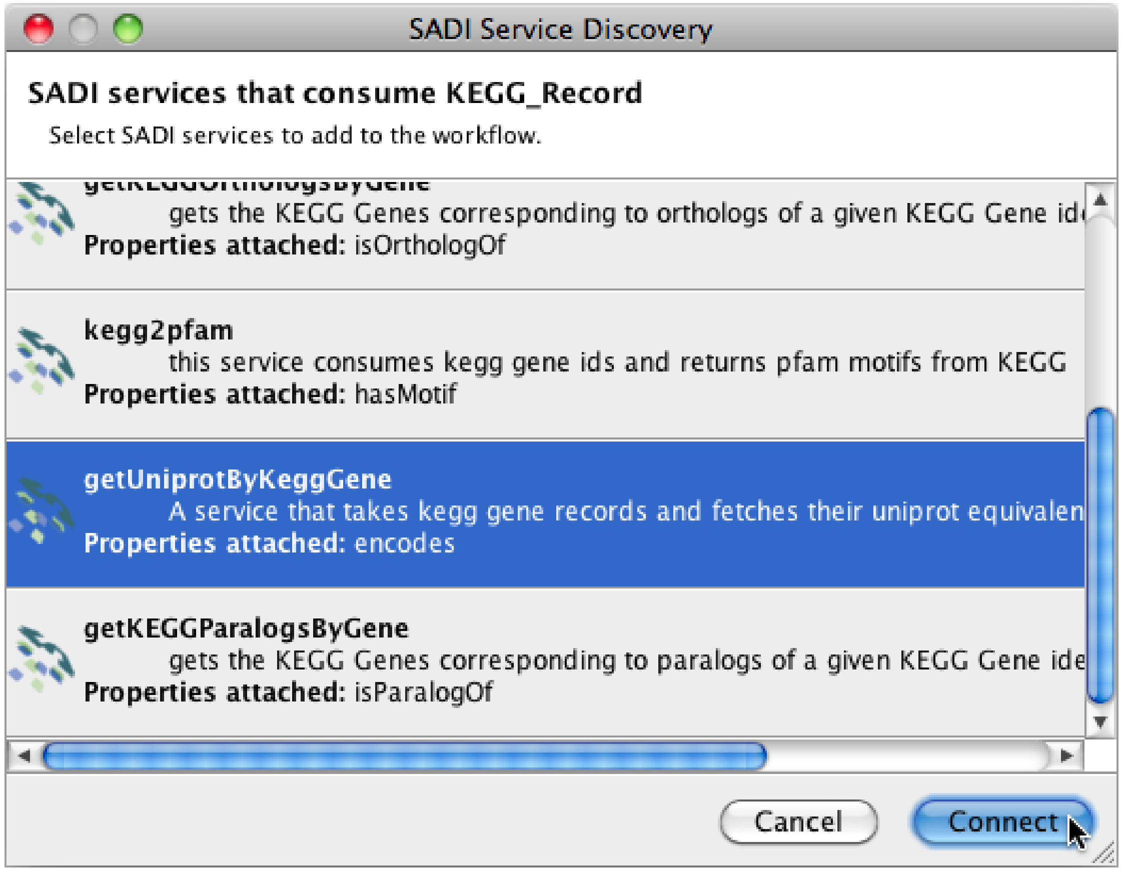
Using SADI services – building a workflow

Drag a link from the service output port to workflow output **gene**.


KEGG_PATHWAY_Record
getKEGGGenesByPathway
hasPathwayGene (KEGG_Record)
Workflow
Service output port: hasPathwayGene (KEGG_Record)
Connect as input to...
Edit
Show details
SADI Service Discovery
Find services that consume KEGG_Record...

Using SADI services – building a workflow

Right-click on the service output port and click **Find services that consume KEGG_Record…**


SADI Service Discovery

SADI services that consume KEGG_Record

Select SADI services to add to the workflow.

getKEGGOrthologsByGene
gets the KEGG Genes corresponding to orthologs of a given KEGG Gene id
Properties attached: isOrthologOf

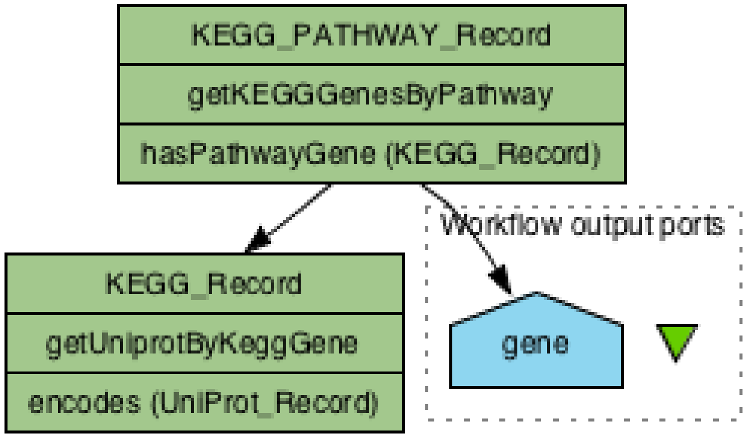
kegg2pfam
this service consumes kegg gene ids and returns pfam motifs from KEGG
Properties attached: hasMotif

getUniprotByKeggGene
A service that takes kegg gene records and fetches their uniprot equivalen
Properties attached: encodes

getKEGGParalogsByGene
gets the KEGG Genes corresponding to paralogs of a given KEGG Gene ide
Properties attached: isParalogOf

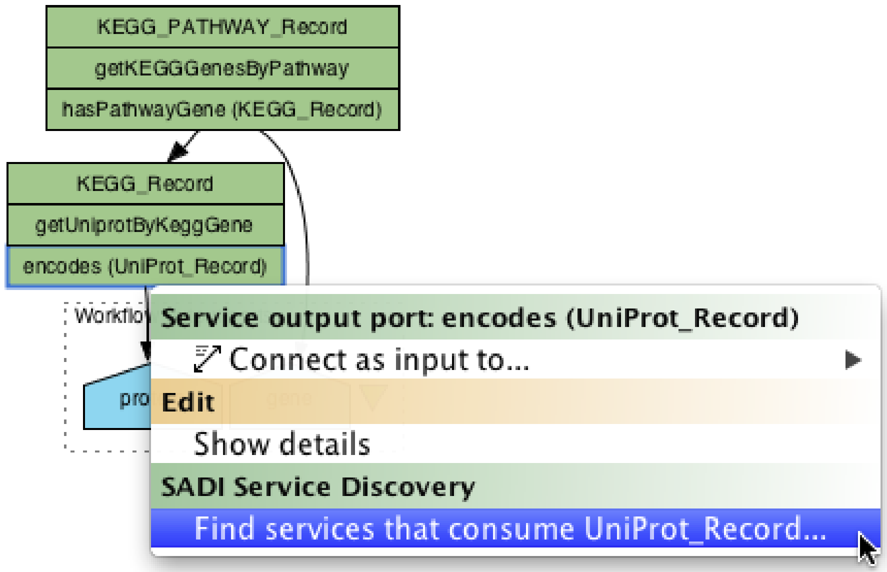
Using SADI services – building a workflow

Select **getUniprotByKeggGene** from the list of SADI services and click **Connect**.



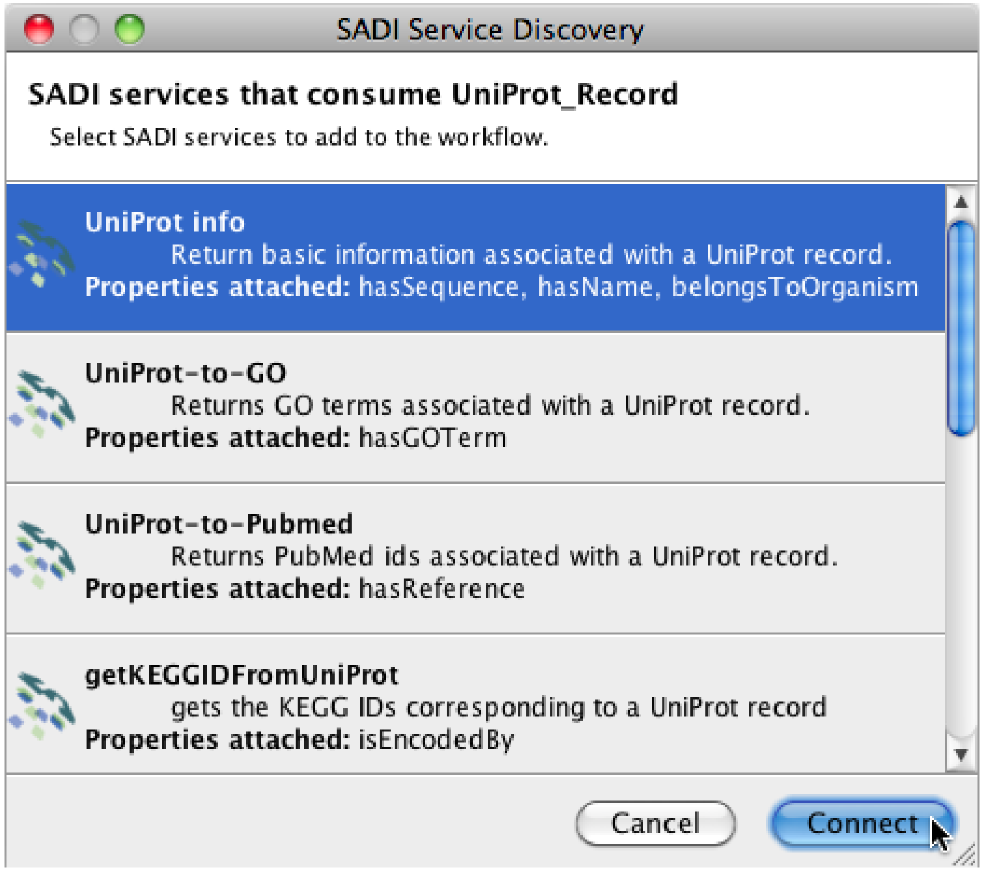
Using SADI services – building a workflow

The **getUniprotByKeggGene** service is added to the workflow and automatically connected to the output from **getKEGGGenesByPathway**.



Using SADI services – building a workflow

The next step in the workflow is to find a SADI service that takes the proteins and returns sequences of those proteins. Right-click on the **encodes** output port and click **Find services that consume UniProt_Record…**



SADI services that consume UniProt_Record
Select SADI services to add to the workflow.

UniProt info
Return basic information associated with a UniProt record.
Properties attached: hasSequence, hasName, belongsToOrganism

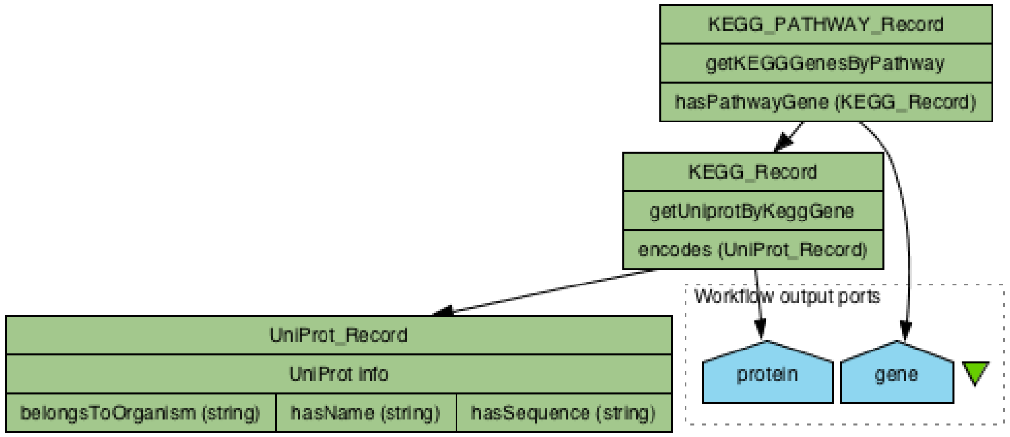
UniProt-to-GO
Returns GO terms associated with a UniProt record.
Properties attached: hasGOTerm

UniProt-to-Pubmed
Returns PubMed ids associated with a UniProt record.
Properties attached: hasReference

getKEGGIDFromUniProt
gets the KEGG IDs corresponding to a UniProt record
Properties attached: isEncodedBy

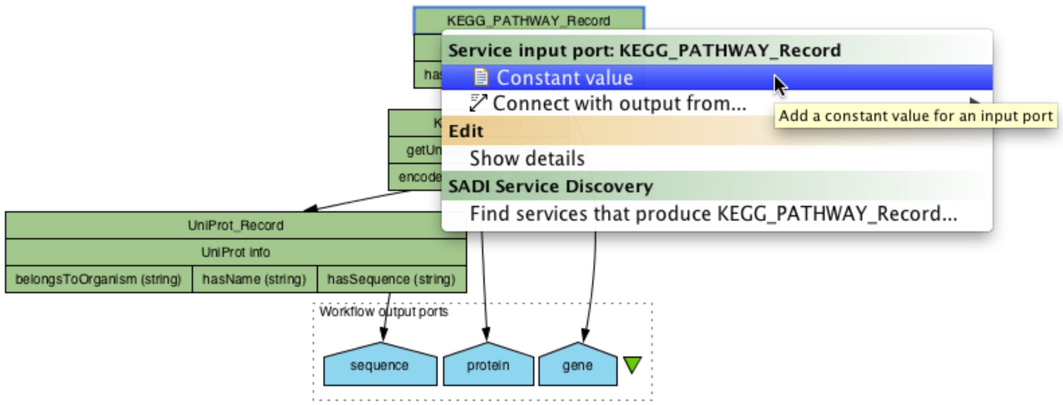
Using SADI services – building a workflow

The **UniProt info** service attaches the property **hasSequence** so select this service and click **Connect**.



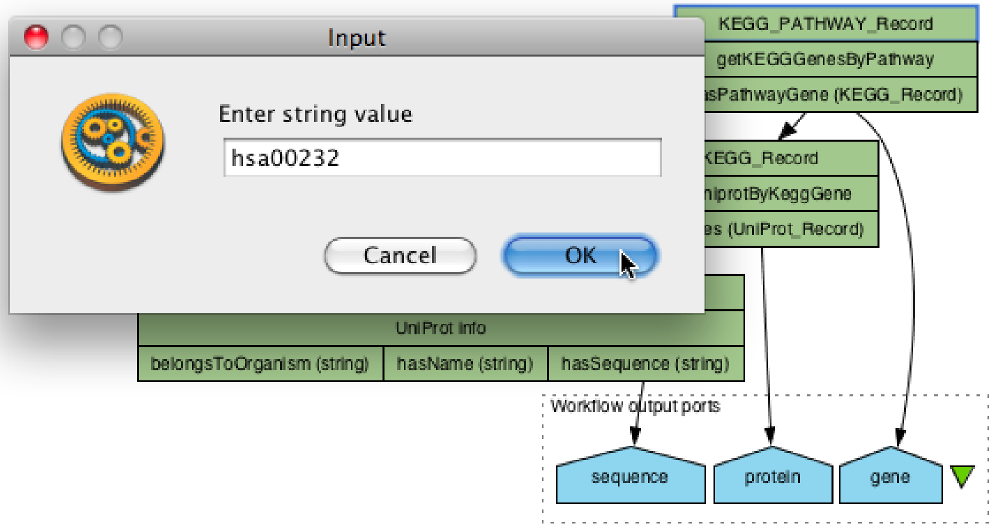
Using SADI services – building a workflow

The **UniProt info** service is added to the workflow and automatically connected to the output from **getUniprotByKeggGene** .


KEGG_PATHWAY_Record
Service input port: KEGG_PATHWAY_Record
Constant value
Connect with output from...
Add a constant value for an input port
Edit
Show details
SADI Service Discovery
Find services that produce KEGG_PATHWAY_Record...
UniProt_Record
UniProt info
belongsToOrganism (string)
hasName (string)
hasSequence (string)
Workflow output ports
sequence
protein
gene

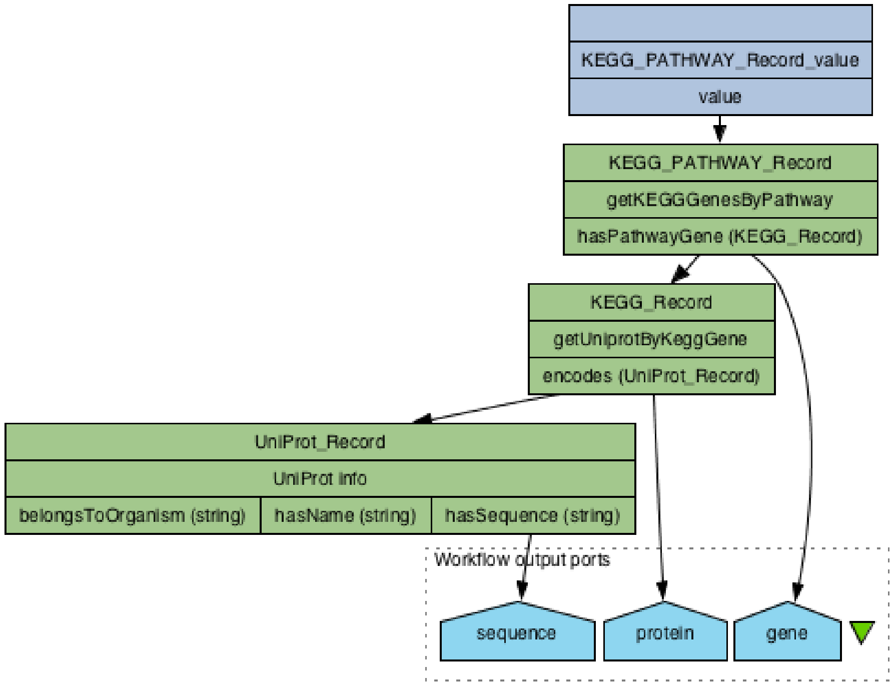
Using SADI services – building a workflow

The KEGG pathway were interested in is "hsa00232", so we'll add it as a constant value. Right-click on the **KEGG_PATHWAY_Record** input port and click **Constant value**.



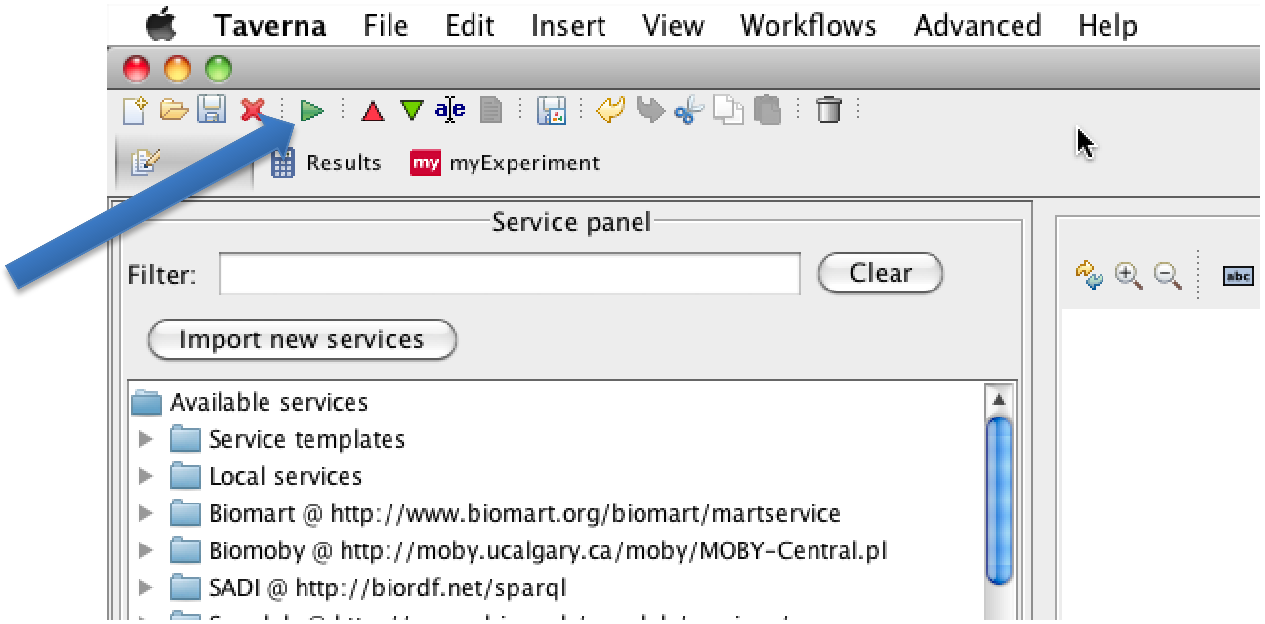
Using SADI services – building a workflow

Enter the value ***hsa00232*** and click **OK**.



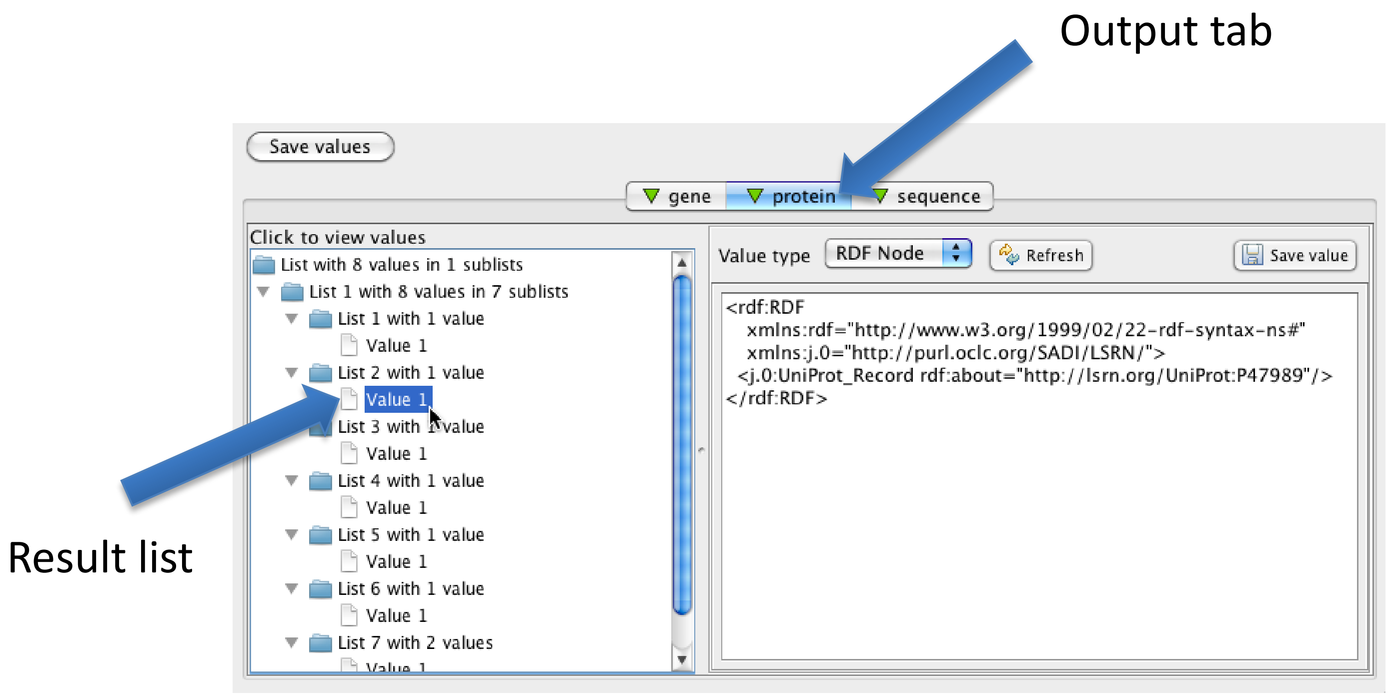
Using SADI services – building a workflow

The workflow is now complete and ready to run.



Using SADI services – running the workflow

To run the workflow click on the green arrow in the tool bar. Taverna will switch to the results view and start running the workflow.

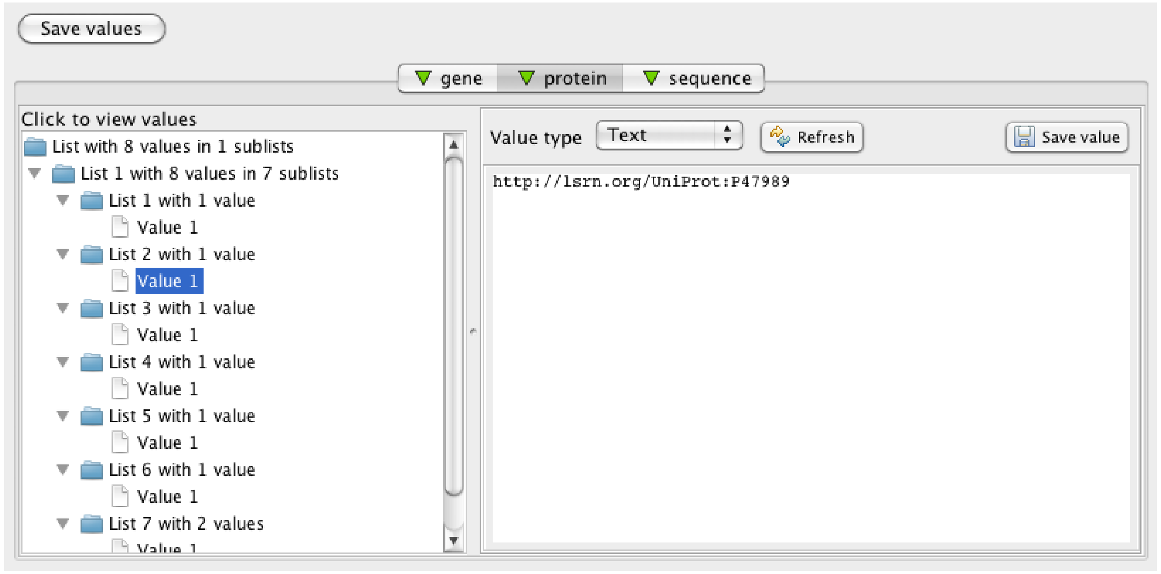


List with 8 values in 1 sublists
List 1 with 8 values in 7 sublists
List 1 with 1 value
Value 1
List 2 with 1 value
Value 1
List 3 with 1 value
Value 1
List 4 with 1 value
Value 1
List 5 with 1 value
Value 1
List 6 with 1 value
Value 1
List 7 with 2 values
Value 1

```
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:j.0="http://purl.oclc.org/SADI/LSRN/">
  <j.0:UniProt_Record rdf:about="http://lsrn.org/UniProt:P47989"/>
</rdf:RDF>
```

Using SADI services – viewing the results

To see the all the results for an output click on the **output tab** for that output. To see an individual result click on the value in the **result list**.



```
http://lsrn.org/UniProt:P47989
```

Using SADI services – viewing the results

When the value type is set to **Text** just the URL for the protein is displayed.

Names and origin

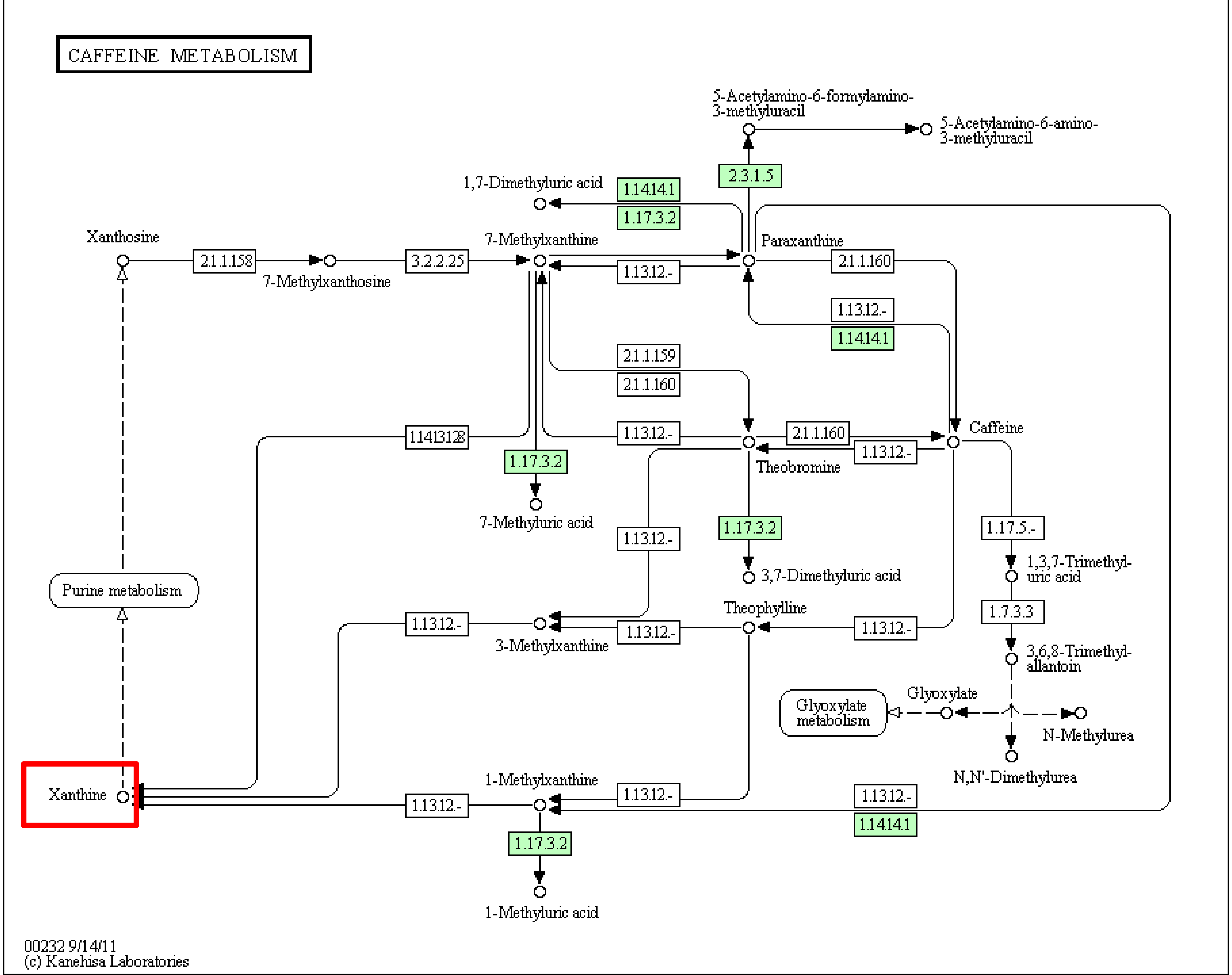
Protein names	*Recommended name:* **Xanthine dehydrogenase/oxidase** Including the following 2 domains: 1. ***Xanthine dehydrogenase*** Short name=XD EC=1.17.1.4 2. ***Xanthine oxidase*** Short name=XO EC=1.17.3.2 *Alternative name(s):* Xanthine oxidoreductase
Gene names	Name: **XDH** Synonyms:XDHA
Organism	**Homo sapiens (Human)**
Taxonomic identifier	9606 [NCBI]
Taxonomic lineage	Eukaryota › Metazoa › Chordata › Craniata › Vertebrata › Euteleostomi › Mammalia › Eutheria › Euarchontoglires › Primates › Haplorrhini › Catarrhini › Hominidae › Homo

Protein attributes

Sequence length	1333 AA.
Sequence status	Complete.
Sequence processing	The displayed sequence is further processed into a mature form.
Protein existence	Evidence at protein level

General annotation (Comments)

Function	Key enzyme in purine degradation. Catalyzes the oxidation of hypoxanthine to xanthine. Catalyzes the oxidation of xanthine to uric acid. Contributes to the generation of reactive oxygen species. Has also low oxidase activity towards aldehydes (in vitro). Ref.11
Catalytic activity	Hypoxanthine + NAD^+ + H_2O = xanthine + NADH. Ref.5 Ref.11 Xanthine + H_2O + O_2 = urate + H_2O_2. Ref.5 Ref.11
Cofactor	Binds 2 2Fe-2S clusters. Ref.11


CAFFEINE METABOLISM
5-Acetylamino-6-formylamino-3-methyluracil
5-Acetylamino-6-amino-3-methyluracil
2.3.1.5
1,7-Dimethyluric acid
1.14.14.1
1.17.3.2
Xanthosine
2.1.1.158
7-Methylxanthosine
3.2.2.25
7-Methylxanthine
1.13.12.-
Paraxanthine
2.1.1.160
1.13.12.-
1.14.14.1
2.1.1.159
2.1.1.160
1.14.13.128
1.13.12.-
2.1.1.160
Caffeine
1.17.3.2
7-Methyluric acid
1.13.12.-
Theobromine
1.13.12.-
1.17.3.2
3,7-Dimethyluric acid
1.17.5.-
1,3,7-Trimethyluric acid
Purine metabolism
1.13.12.-
3-Methylxanthine
1.13.12.-
Theophylline
1.13.12.-
1.7.3.3
3,6,8-Trimethylallantoin
Glyoxylate metabolism
Glyoxylate
N-Methylurea
N,N'-Dimethylurea
Xanthine
1-Methylxanthine
1.13.12.-
1.13.12.-
1.13.12.-
1.14.14.1
1.17.3.2
1-Methyluric acid
00232 9/14/11
(c) Kanehisa Laboratories

SADI-Taverna Summary

- Search for the property of the data you desire
- Automatically adds the service
  - Correctly connected automatically
- The SADI plugin handles parsing into and out of RDF format automatically and transparently
  - Easy to connect SADI with non-SADI services

CARDIO
SHARE

Data + Knowledge
for Cardiac Researchers

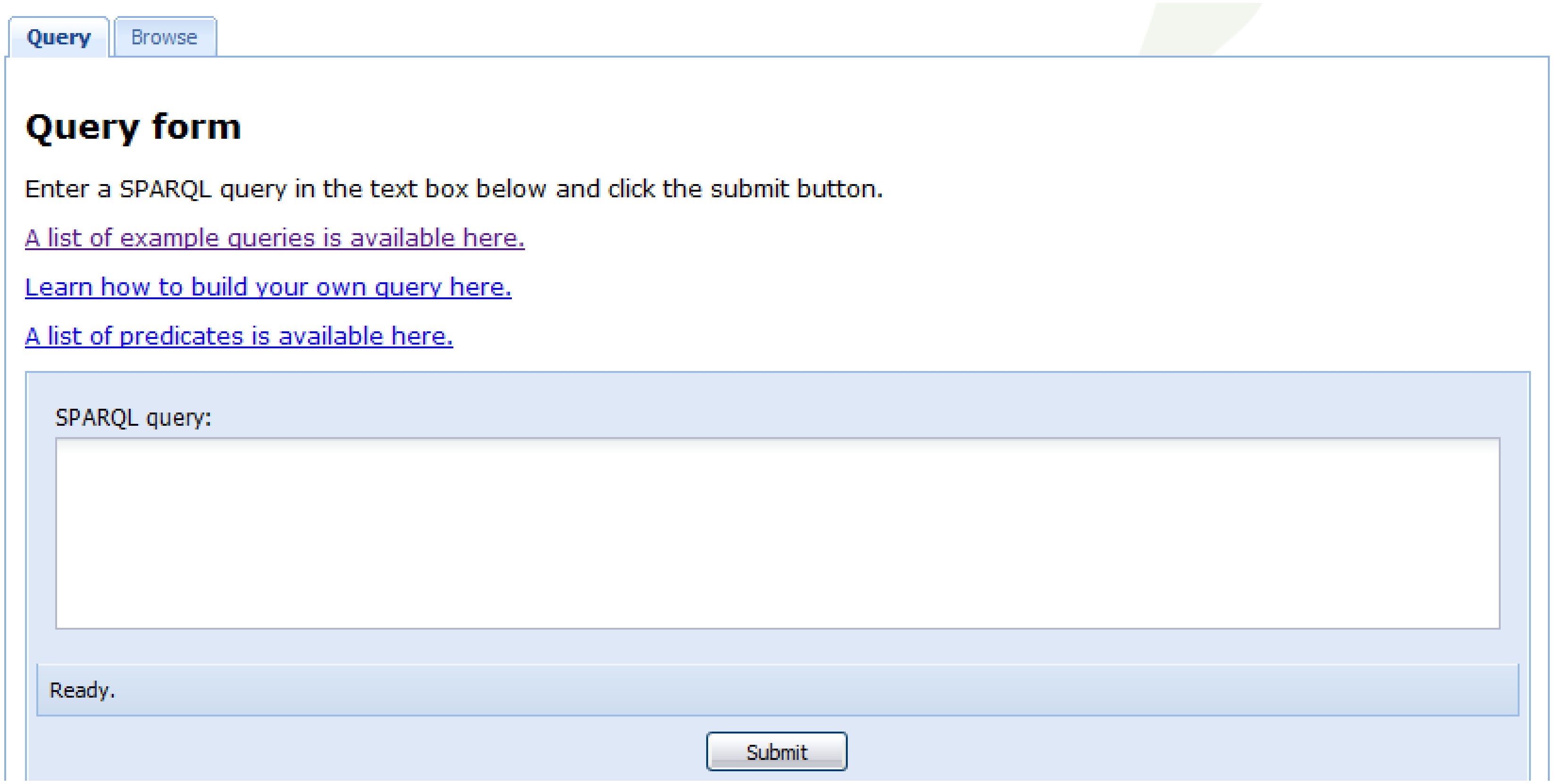
Powered by SADI

Semantic **H**ealth **A**nd **R**esearch **E**nvironment

SPARQL enhanced by SADI

http://biordf.net/cardioSHARE/

HEART & STROKE FOUNDATION OF BC & YUKON
Finding answers. For life.



Query form

Enter a SPARQL query in the text box below and click the submit button.

A list of example queries is available here.

Learn how to build your own query here.

A list of predicates is available here.

SPARQL query:

Ready.

http://biordf.net/cardioSHARE/

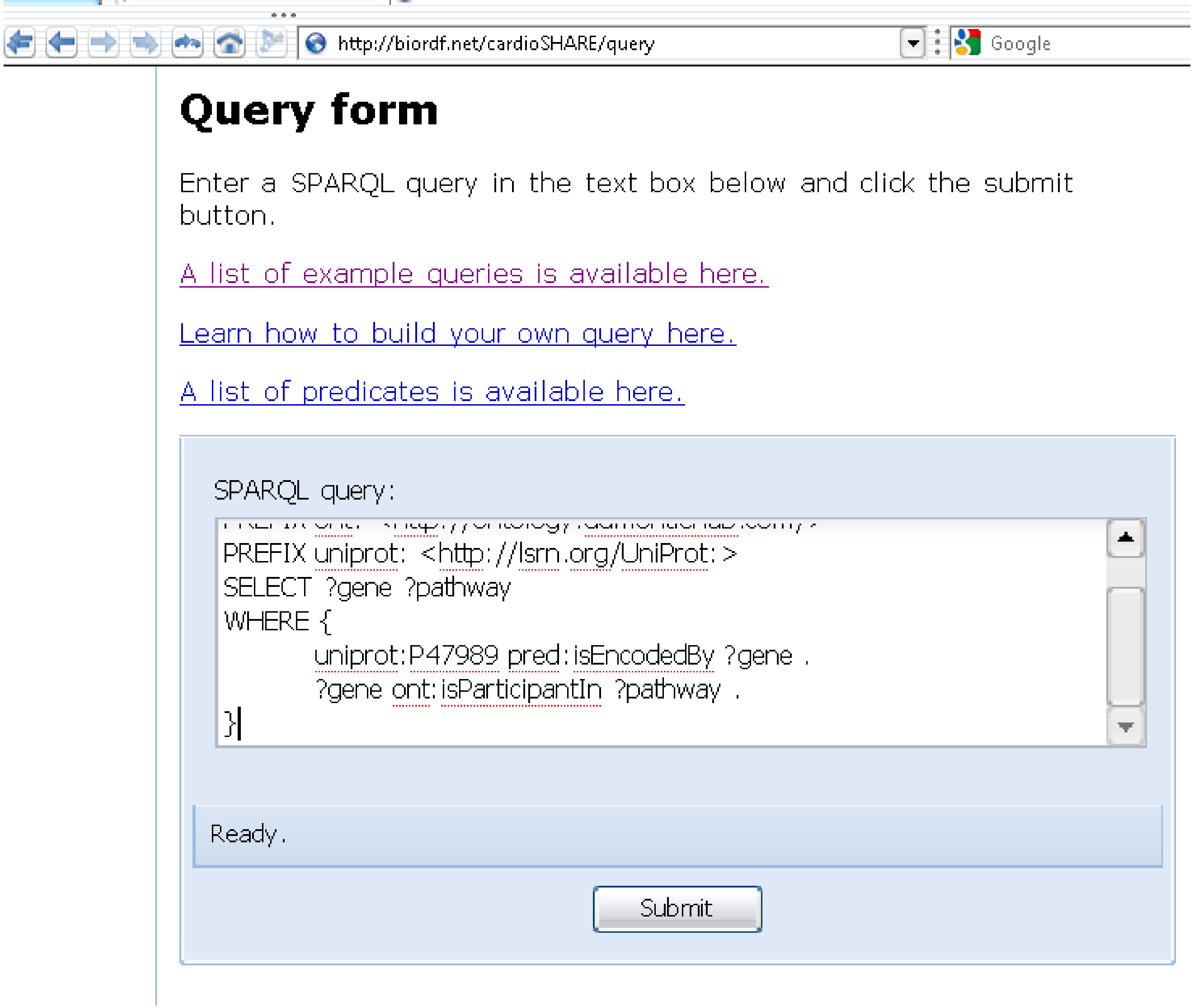
SHARE

- Use SADI to automatically construct a workflow that creates a query-specific database.
- Generates an RDF triple output containing the

  <**subject**(input), **object**(output), **predicate**(relationship determined by service)>.
- A SHARE query is resolved according to below:
  1. Each predicate in query is examined and any matching services are retrieved from the registry.
  2. The services are called upon, results converted to RDF, data is stored in local triple.
  3. The query engine is executed as normal against the local triple.

What pathways does UniProt protein P47989 belong to?

```
PREFIX pred: <http://sadiframework.org/ontologies/predicates.owl#>
PREFIX ont: <http://ontology.dumontierlab.com/>
PREFIX uniprot: <http://lsrn.org/UniProt:>
SELECT ?gene ?pathway
WHERE {
       uniprot:P47989 pred:isEncodedBy ?gene .
       ?gene ont:isParticipantIn ?pathway .
}
```



Query form

Enter a SPARQL query in the text box below and click the submit button.

A list of example queries is available here.

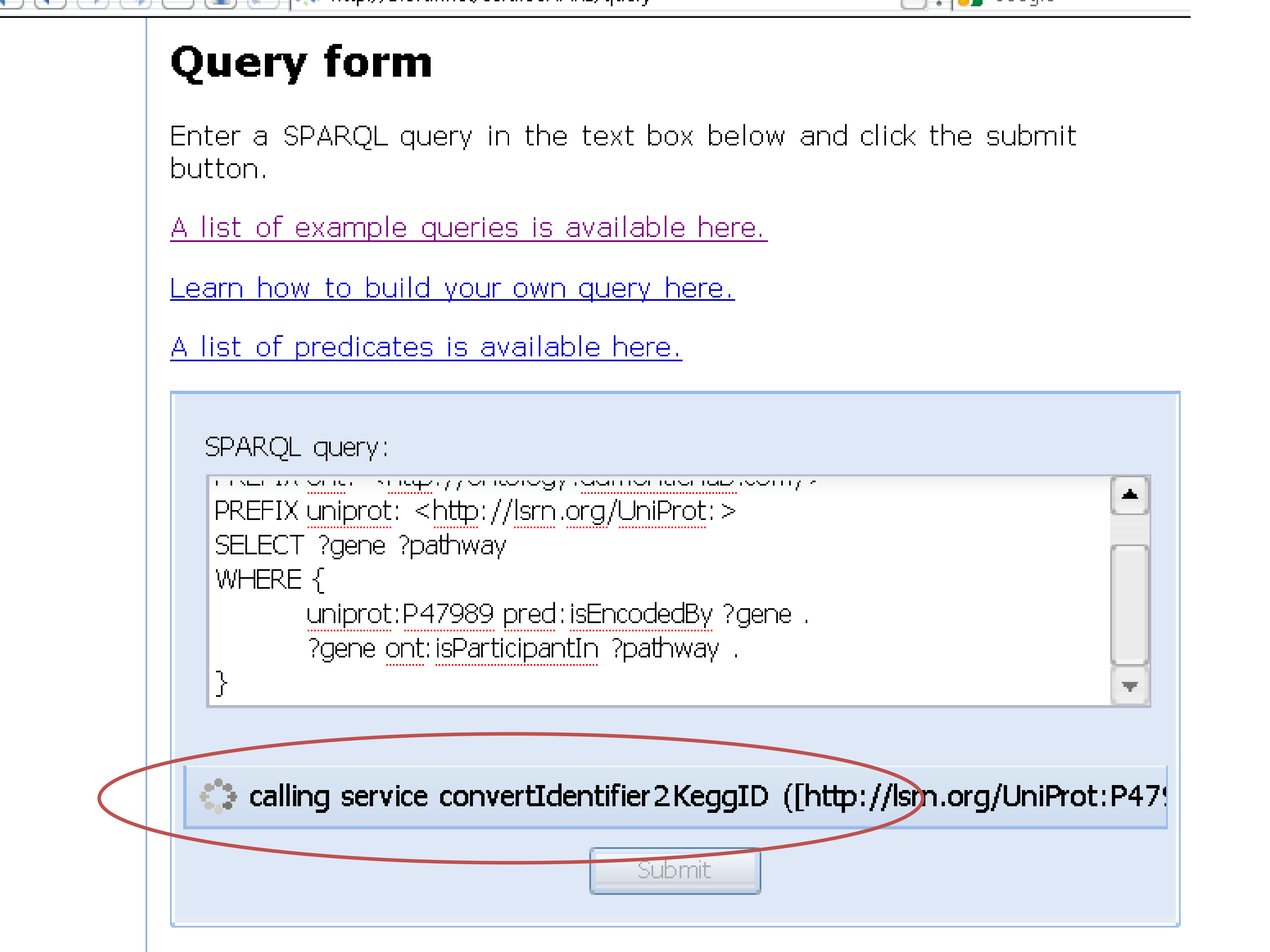
Learn how to build your own query here.

A list of predicates is available here.

SPARQL query:

```
PREFIX uniprot: <http://lsrn.org/UniProt:>
SELECT ?gene ?pathway
WHERE {
        uniprot:P47989 pred:isEncodedBy ?gene .
        ?gene ont:isParticipantIn ?pathway .
}
```

Ready.



Query form

Enter a SPARQL query in the text box below and click the submit button.

A list of example queries is available here.

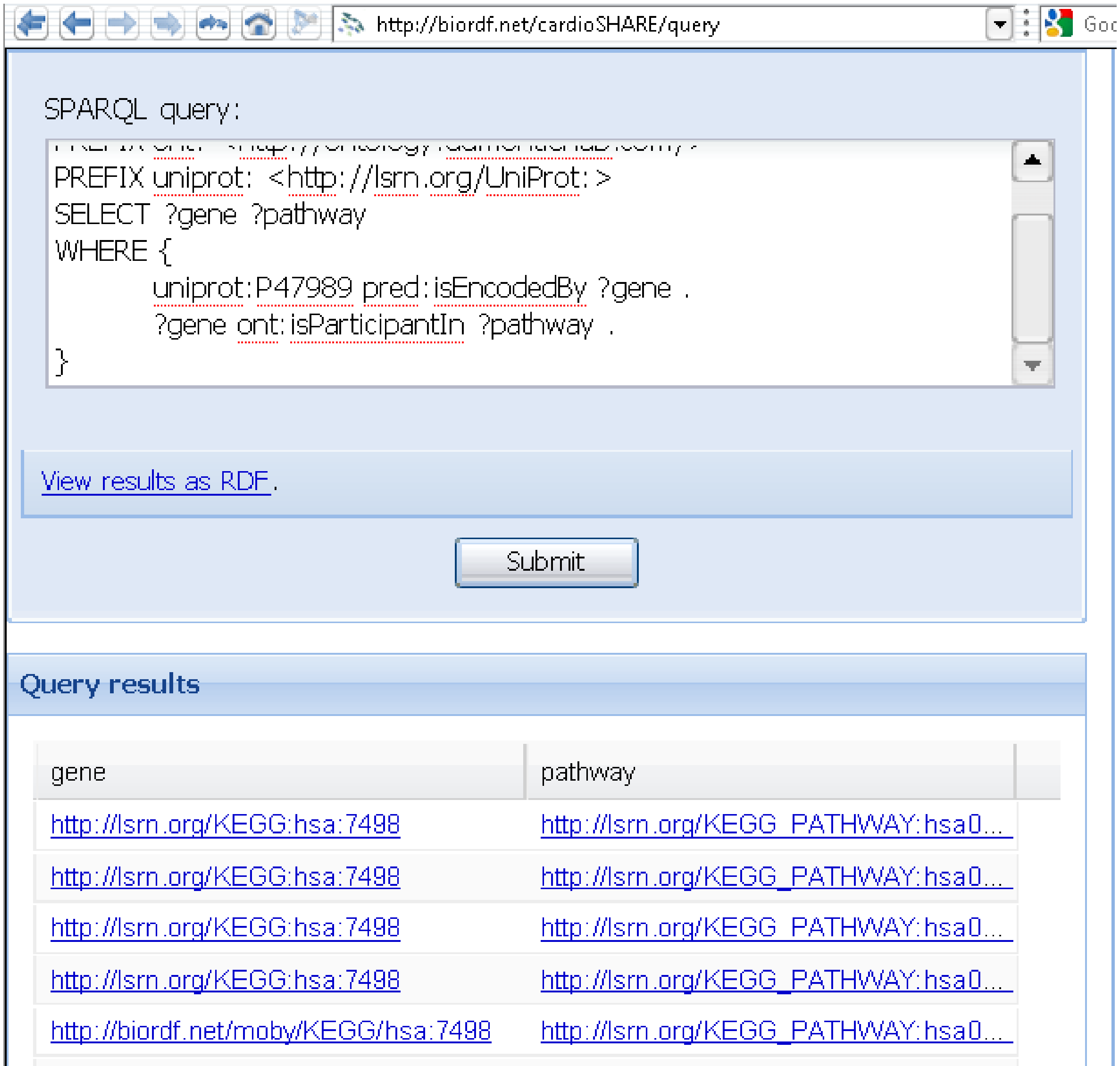
Learn how to build your own query here.

A list of predicates is available here.

SPARQL query:

```
PREFIX ont: <http://ontology.dumontierlab.com/>
PREFIX uniprot: <http://lsrn.org/UniProt:>
SELECT ?gene ?pathway
WHERE {
        uniprot:P47989 pred:isEncodedBy ?gene .
        ?gene ont:isParticipantIn ?pathway .
}
```

calling service convertIdentifier2KeggID ([http://lsrn.org/UniProt:P47



http://biordf.net/cardioSHARE/query

SPARQL query:

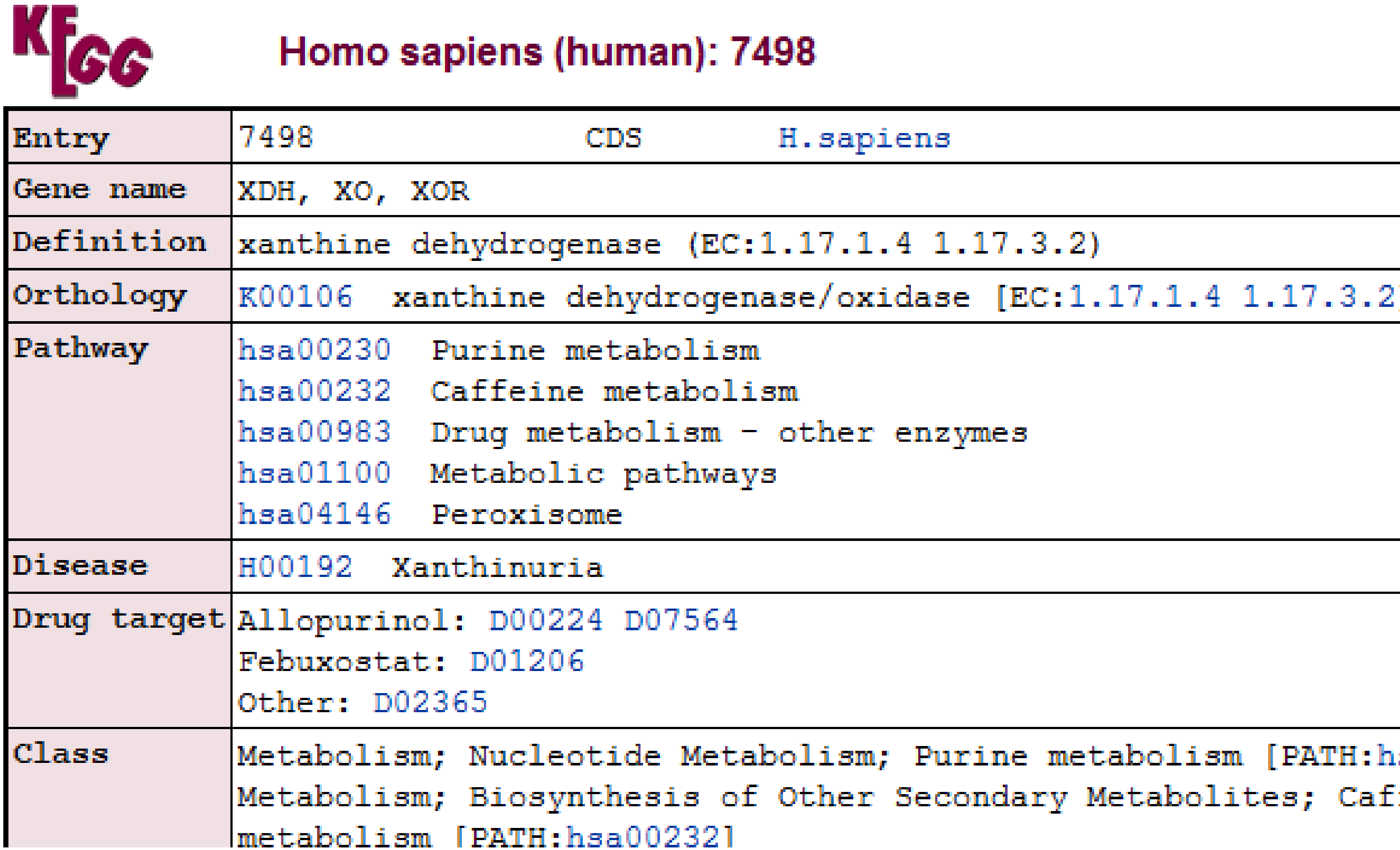
```
PREFIX uniprot: <http://lsrn.org/UniProt:>
SELECT ?gene ?pathway
WHERE {
        uniprot:P47989 pred:isEncodedBy ?gene .
        ?gene ont:isParticipantIn ?pathway .
}
```

View results as RDF.

Submit

Query results

gene	pathway
http://lsrn.org/KEGG:hsa:7498	http://lsrn.org/KEGG_PATHWAY:hsa0...
http://lsrn.org/KEGG:hsa:7498	http://lsrn.org/KEGG_PATHWAY:hsa0...
http://lsrn.org/KEGG:hsa:7498	http://lsrn.org/KEGG_PATHWAY:hsa0...
http://lsrn.org/KEGG:hsa:7498	http://lsrn.org/KEGG_PATHWAY:hsa0...
http://biordf.net/moby/KEGG/hsa:7498	http://lsrn.org/KEGG_PATHWAY:hsa0...

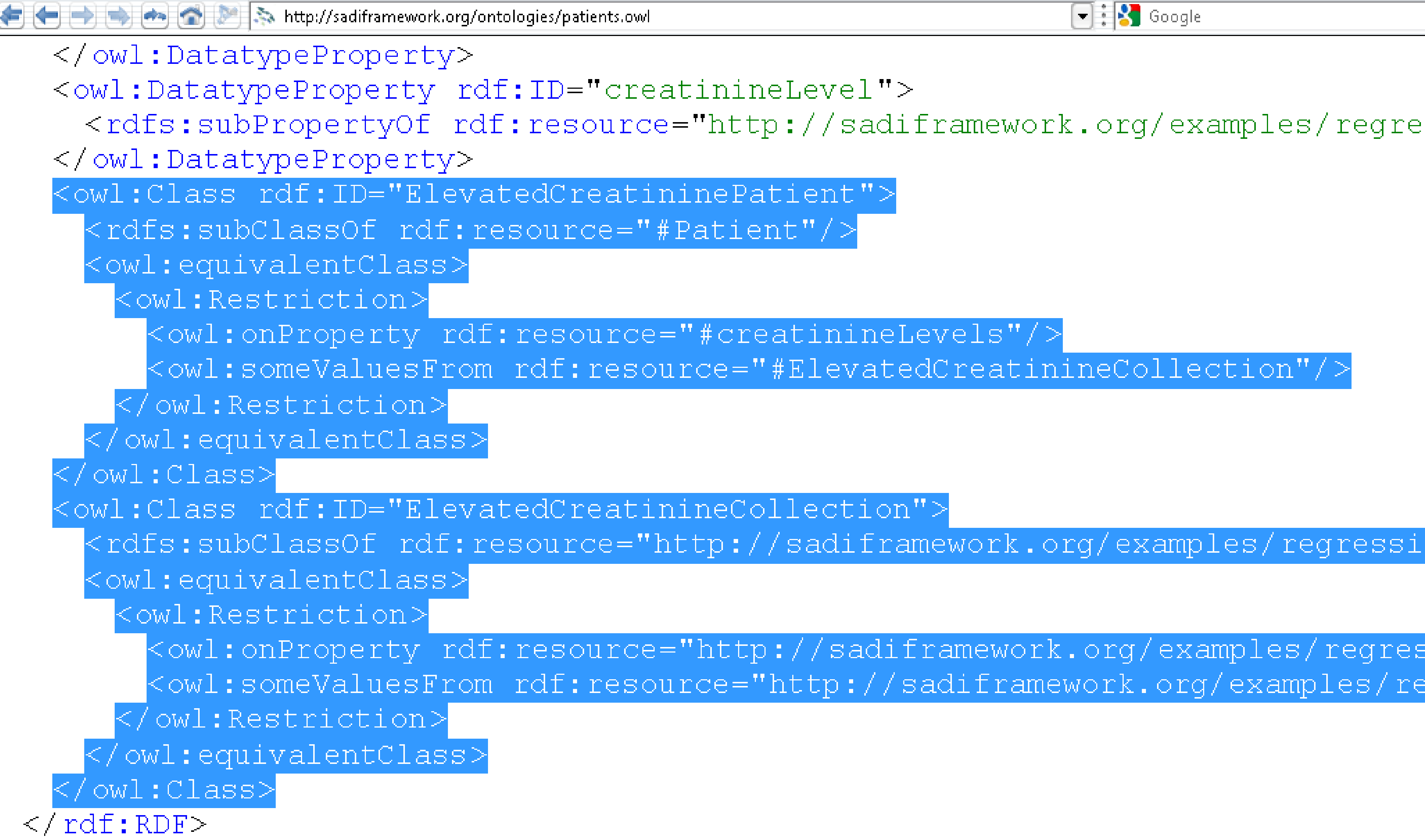


Homo sapiens (human): 7498

Entry	7498 CDS H.sapiens
Gene name	XDH, XO, XOR
Definition	xanthine dehydrogenase (EC:1.17.1.4 1.17.3.2)
Orthology	K00106 xanthine dehydrogenase/oxidase [EC:1.17.1.4 1.17.3.2
Pathway	hsa00230 Purine metabolism hsa00232 Caffeine metabolism hsa00983 Drug metabolism - other enzymes hsa01100 Metabolic pathways hsa04146 Peroxisome
Disease	H00192 Xanthinuria
Drug target	Allopurinol: D00224 D07564 Febuxostat: D01206 Other: D02365
Class	Metabolism; Nucleotide Metabolism; Purine metabolism [PATH:h Metabolism; Biosynthesis of Other Secondary Metabolites; Caf metabolism [PATH:hsa00232]

Show me the latest Blood Urea Nitrogen and Creatinine levels of patients who appear to be rejecting their transplants

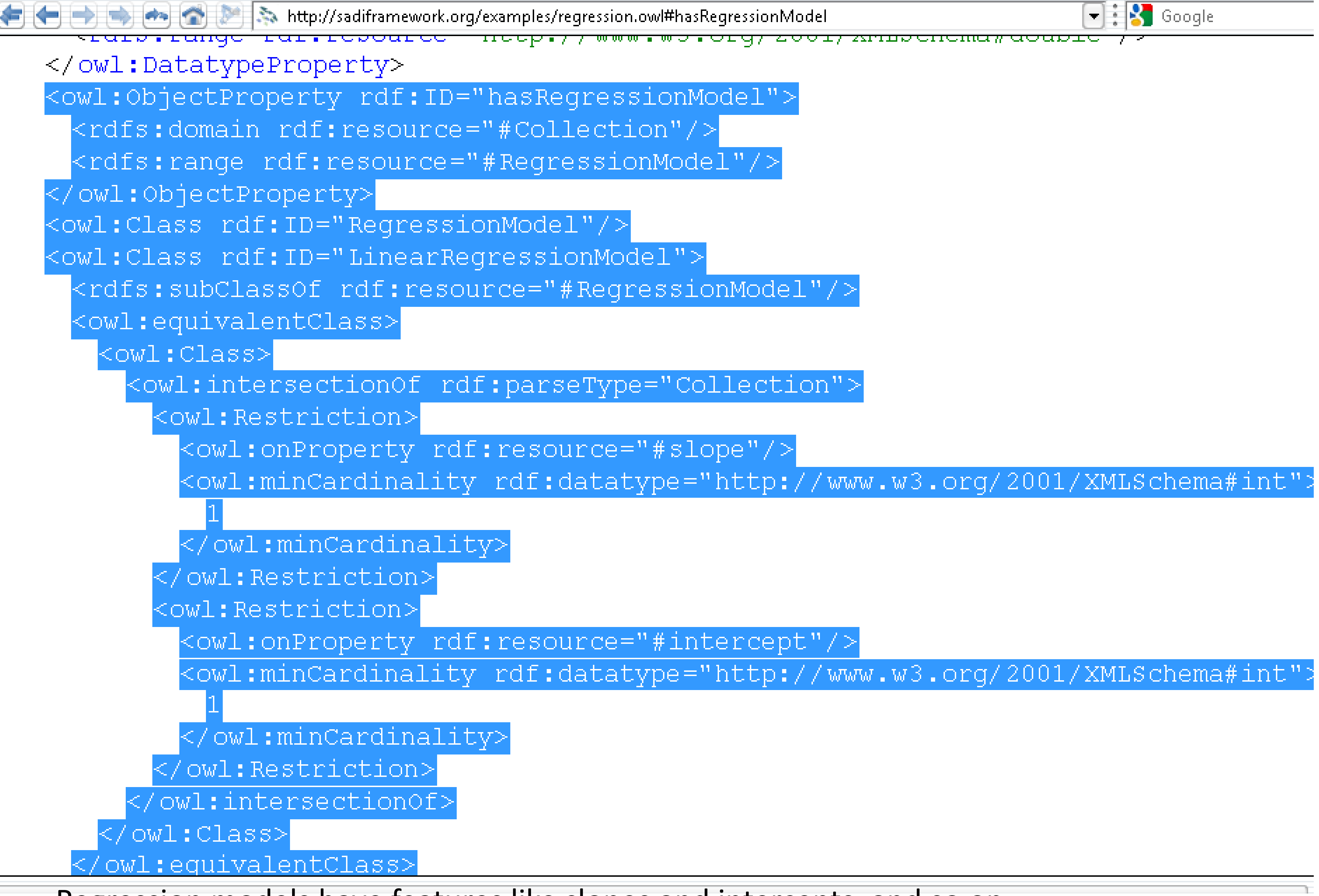
```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX patient: <http://sadiframework.org/ontologies/patients.owl#>
PREFIX l: <http://sadiframework.org/ontologies/predicates.owl#>
SELECT ?patient ?bun ?creat
FROM <http://sadiframework.org/ontologies/patients.rdf>
WHERE {
       ?patient rdf:type patient:LikelyRejecter .
       ?patient l:latestBUN ?bun .
       ?patient l:latestCreatinine ?creat .
}
```


http://sadiframework.org/ontologies/patients.owl

```
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="creatinineLevel">
    <rdfs:subPropertyOf rdf:resource="http://sadiframework.org/examples/regre
  </owl:DatatypeProperty>
  <owl:Class rdf:ID="ElevatedCreatininePatient">
    <rdfs:subClassOf rdf:resource="#Patient"/>
    <owl:equivalentClass>
      <owl:Restriction>
        <owl:onProperty rdf:resource="#creatinineLevels"/>
        <owl:someValuesFrom rdf:resource="#ElevatedCreatinineCollection"/>
      </owl:Restriction>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="ElevatedCreatinineCollection">
    <rdfs:subClassOf rdf:resource="http://sadiframework.org/examples/regressi
    <owl:equivalentClass>
      <owl:Restriction>
        <owl:onProperty rdf:resource="http://sadiframework.org/examples/regres
        <owl:someValuesFrom rdf:resource="http://sadiframework.org/examples/re
      </owl:Restriction>
    </owl:equivalentClass>
  </owl:Class>
</rdf:RDF>
```

Start burrowing through the LikelyRejector OWL class
→ find that we need a regression model OWL class

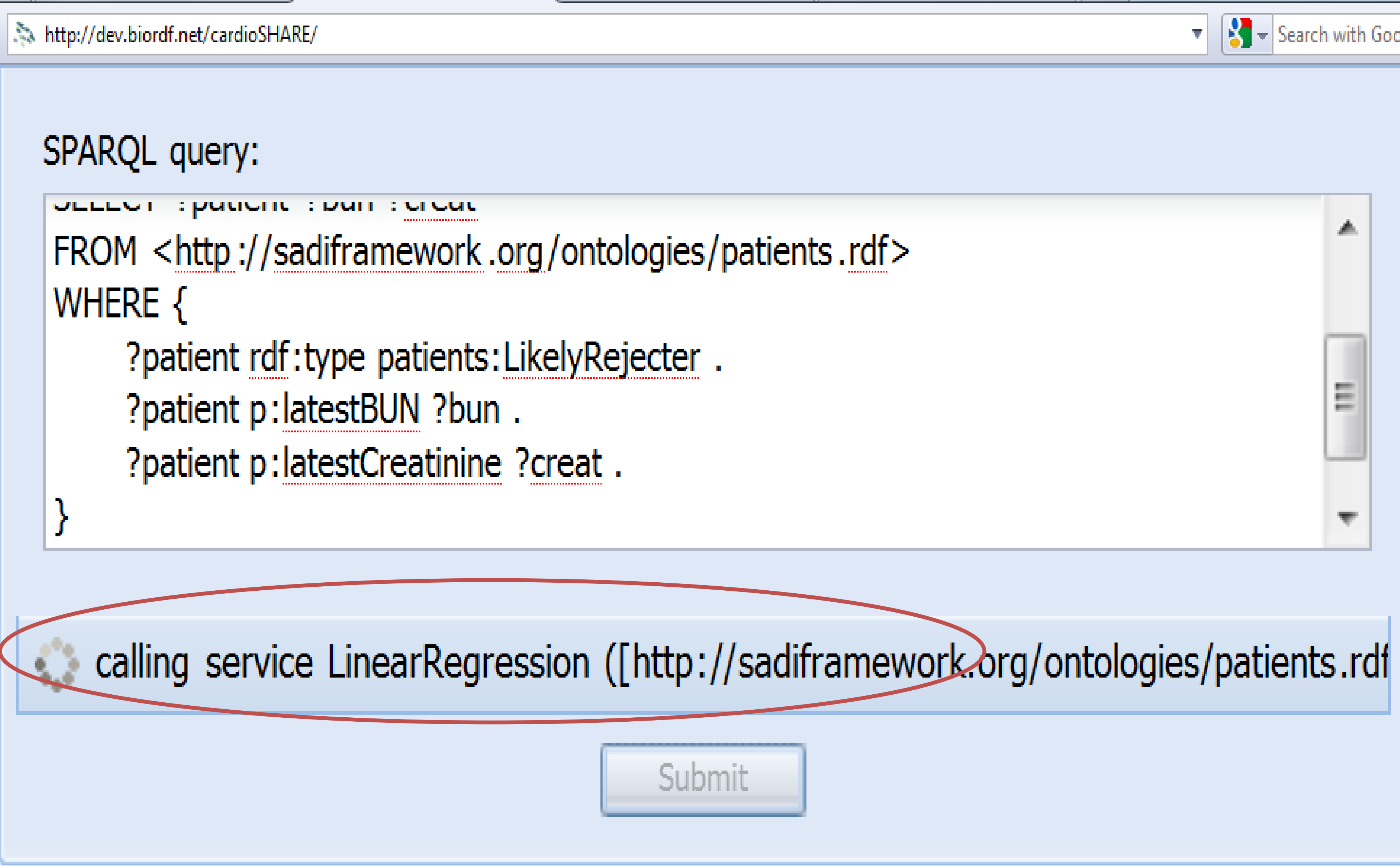
“the regression line over creatinine measurements should have an increasing slope”



http://sadiframework.org/examples/regression.owl#hasRegressionModel

```
    </owl:DatatypeProperty>
    <owl:ObjectProperty rdf:ID="hasRegressionModel">
      <rdfs:domain rdf:resource="#Collection"/>
      <rdfs:range rdf:resource="#RegressionModel"/>
    </owl:ObjectProperty>
    <owl:Class rdf:ID="RegressionModel"/>
    <owl:Class rdf:ID="LinearRegressionModel">
      <rdfs:subClassOf rdf:resource="#RegressionModel"/>
      <owl:equivalentClass>
        <owl:Class>
          <owl:intersectionOf rdf:parseType="Collection">
            <owl:Restriction>
              <owl:onProperty rdf:resource="#slope"/>
              <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
                1
              </owl:minCardinality>
            </owl:Restriction>
            <owl:Restriction>
              <owl:onProperty rdf:resource="#intercept"/>
              <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
                1
              </owl:minCardinality>
            </owl:Restriction>
          </owl:intersectionOf>
        </owl:Class>
      </owl:equivalentClass>
```

Regression models have features like slopes and intercepts, and so on.
The class is completely decomposed until a set of required Services are discovered capable of creating all these necessary properties



http://dev.biordf.net/cardioSHARE/

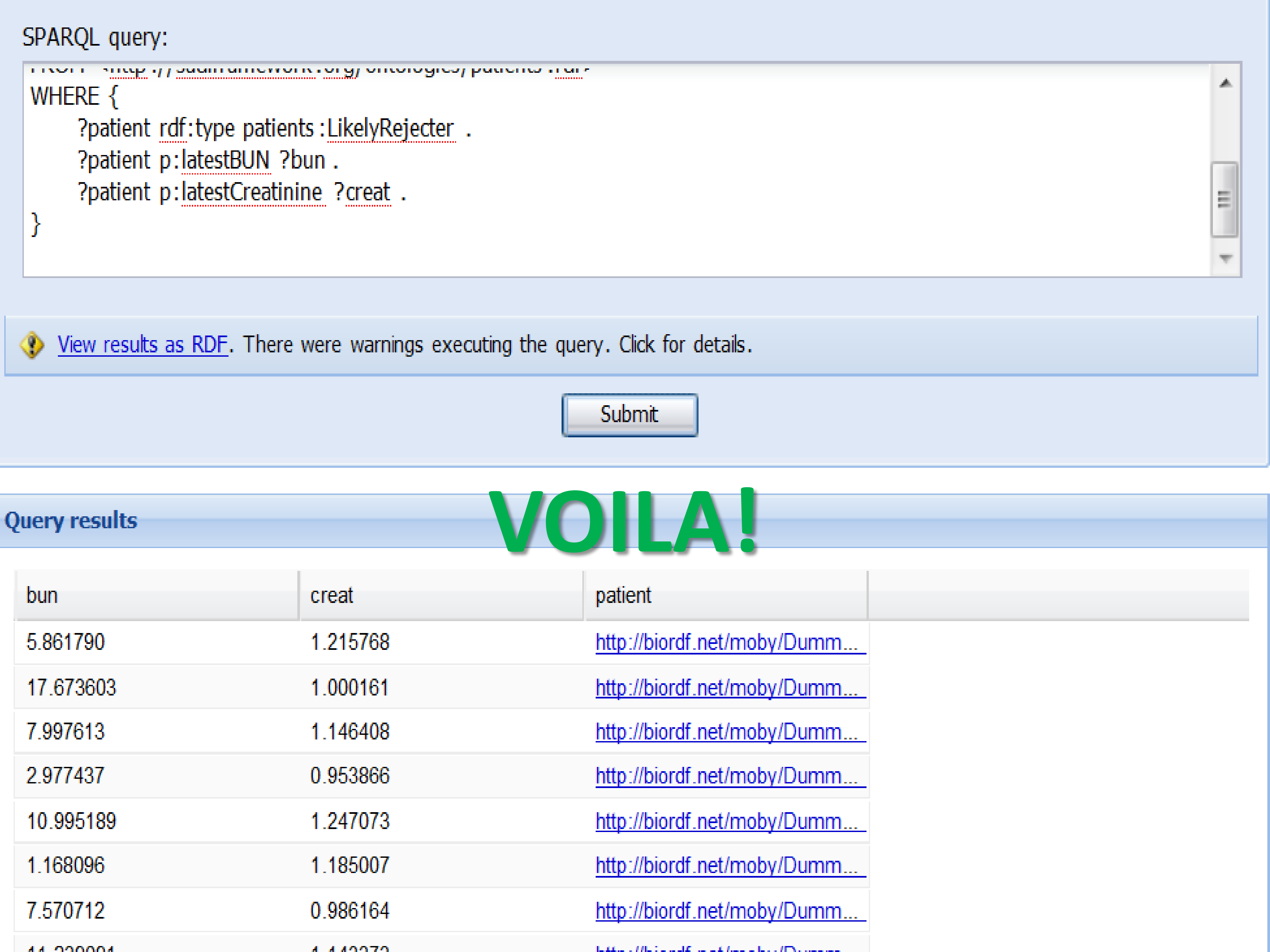
SPARQL query:

```
SELECT ?patient ?bun ?creat
FROM <http://sadiframework.org/ontologies/patients.rdf>
WHERE {
      ?patient rdf:type patients:LikelyRejecter .
      ?patient p:latestBUN ?bun .
      ?patient p:latestCreatinine ?creat .
}
```

calling service LinearRegression ([http://sadiframework.org/ontologies/patients.rdf

Decomposition of the OWL class uncovers the need for a Linear Regression analysis on the patient blood chemistry data

VOILA!



SPARQL query:

```
WHERE {
     ?patient rdf:type patients:LikelyRejecter .
     ?patient p:latestBUN ?bun .
     ?patient p:latestCreatinine ?creat .
}
```

View results as RDF. There were warnings executing the query. Click for details.

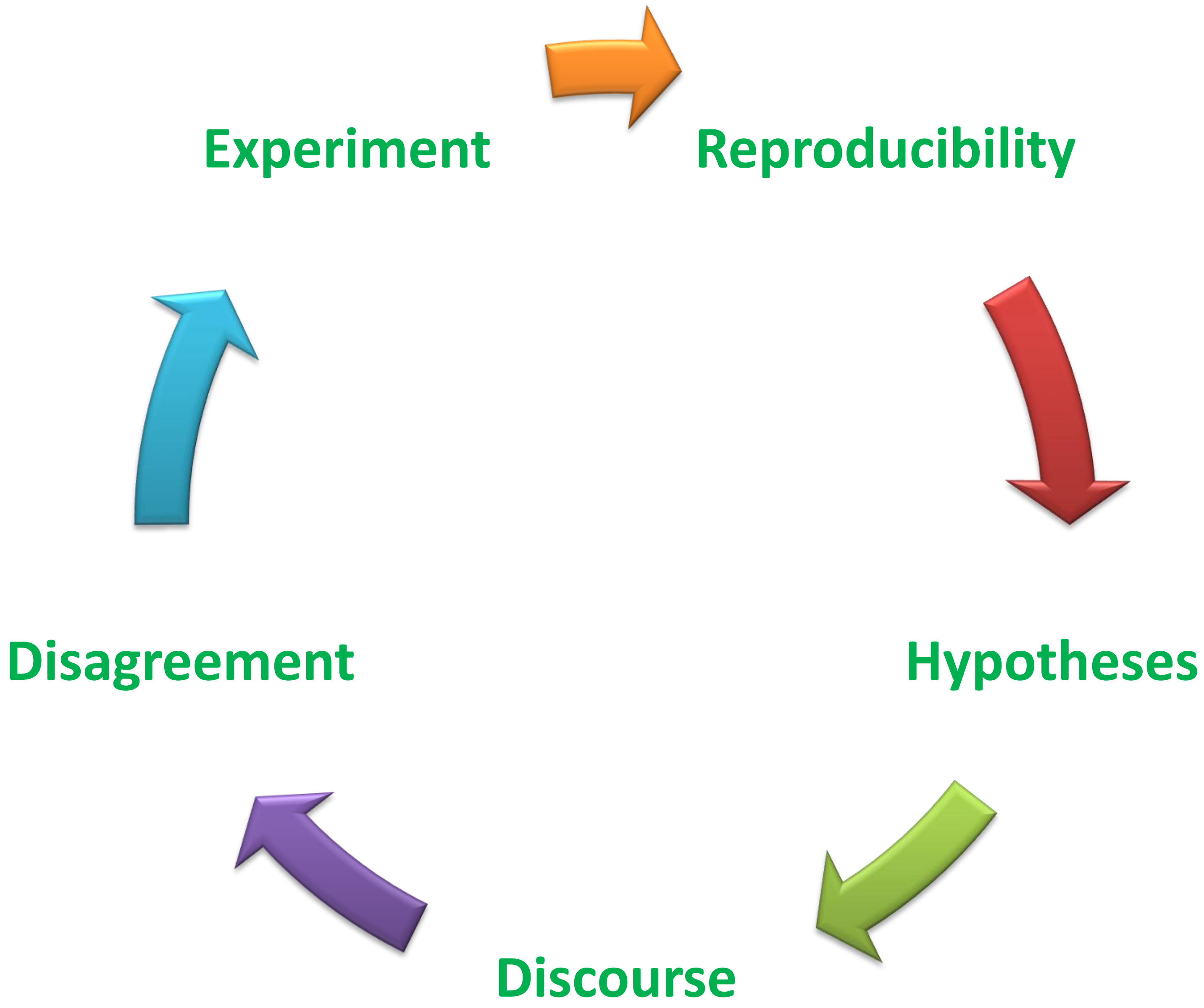
Submit

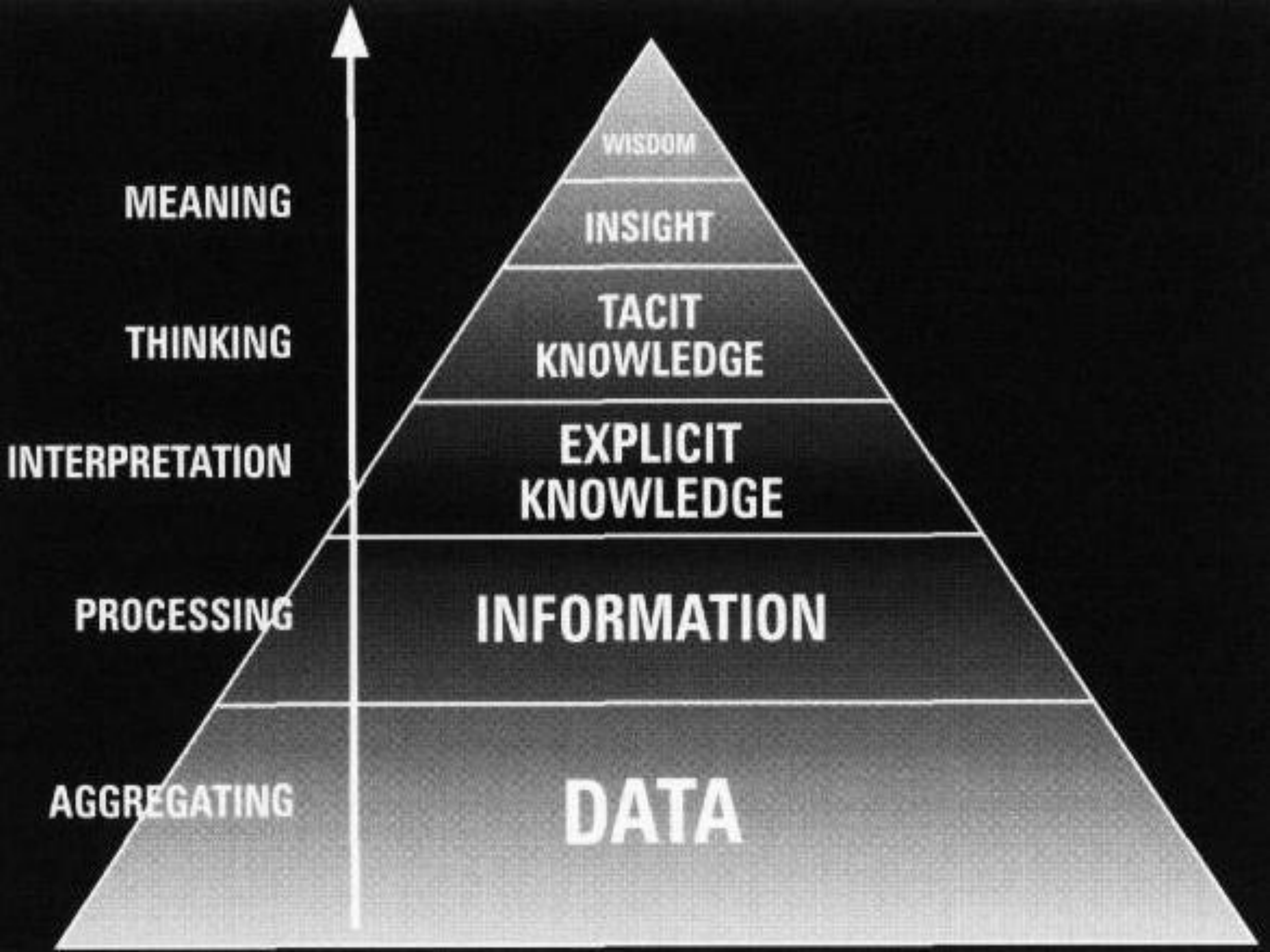
Query results

bun	creat	patient
5.861790	1.215768	http://biordf.net/moby/Dumm...
17.673603	1.000161	http://biordf.net/moby/Dumm...
7.997613	1.146408	http://biordf.net/moby/Dumm...
2.977437	0.953866	http://biordf.net/moby/Dumm...
10.995189	1.247073	http://biordf.net/moby/Dumm...
1.168096	1.185007	http://biordf.net/moby/Dumm...
7.570712	0.986164	http://biordf.net/moby/Dumm...

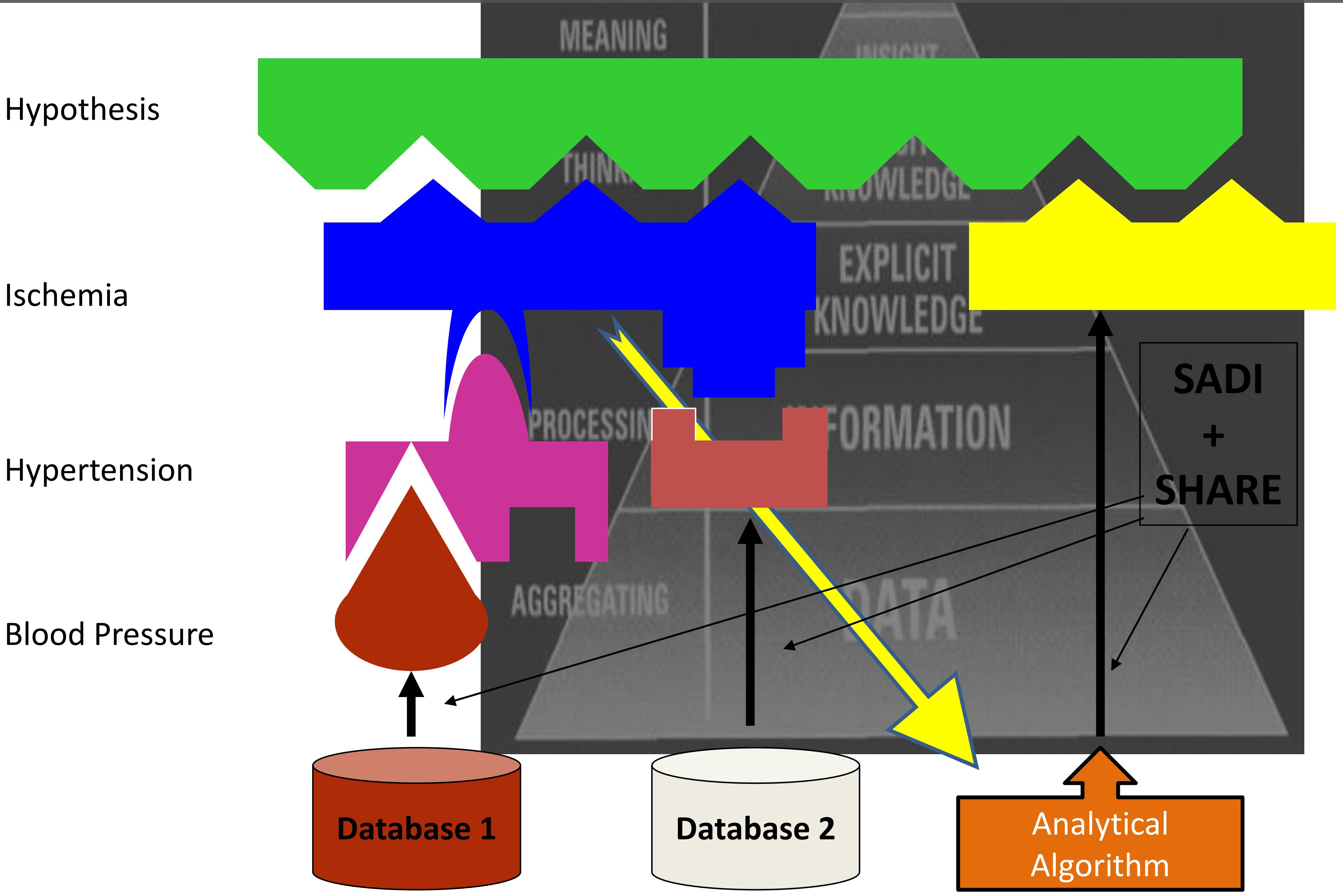
Consequences

- User gets to create their own definition and ontology
  - Ex. LikelyRejecter
- It can be modified and re-used by the user, published for other users to use, modify and compare to their own world-view
  - The user's personal world-view is explicitly expressed and can be dynamically evaluated against global data and knowledge
  - Ontology development is distributed and personal rather than centralized


Experiment
Reproducibility
Hypotheses
Discourse
Disagreement


WISDOM
INSIGHT
TACIT KNOWLEDGE
EXPLICIT KNOWLEDGE
INFORMATION
DATA
MEANING
THINKING
INTERPRETATION
PROCESSING
AGGREGATING

Ontologically-expressed Hypotheses *drive* the discovery, assembly, and analysis of data capable of evaluating their validity



Advantages

- Design patterns are supported by an accompanying codebase and plug-in tools almost completely automated.
- Simplifies the planning process for providers, by reducing the number of “arbitrary” decisions they need to make.
- The specification was specifically designed to support multiplexed messages. Responses from each processor may simply be concatenated regardless of order.
- Enforces other best-practices in Web development, thus helping providers generate robust, error-free systems, and tools are available to regularly evaluated and validated service functionality.
- Not in conflict with any existing network security software or protection model.

Limitations

- Utility of SADI is entirely dependent on the number of providers who adopt its conventions.
- There is an extensive tooling support for traditional Web services and there is a perceived simplicity of XML compared to RDF/OWL.
- Success of the SADI architecture will largely depend on widespread re-use of publicly-available and well-defined ontological predicates, and the definition of service inputs in terms of OWL restrictions on these properties.

References

- Tutorial/Demonstration slides from Prof. Mark Wilkinson of University of British Columbia at http://www.slideshare.net/markmoby.
- SADI http://sadiframework.org.
- SHARE http://biordf.net/cardioSHARE.
- Wilkinson M, Vandervalk B, McCarthy L (2011). The Semantic Automated Discovery and Integration (SADI) Web service Design-Pattern, API and Reference Implementation. Journal of Biomedical Semantics 2:8 doi:10.1186/2041-1480-2-8.
- Withers D, Kawas E, McCarthy L, Vandervalk B, Wilkinson M (2010). Semantically-Guided Workflow Construction in Taverna: The SADI and BioMoby Plug-Ins. In Texts in theoretical computer science 301-312.
- Wilkinson MD, Vandervalk B, McCarthy L (2009). SADI Semantic Web Services - 'cause you can't always GET what you want! In Proceedings of the IEEE APSCC.
- Wilkinson M, Vandervalk B, McCarthy L (2008). CardioSHARE: Web Services for the Semantic Web.