SADI for GMOD: An RDF/OWL Interface for GMOD

http://code.google.com/p/sadi/wiki/SADIforGMOD

What is SADI for GMOD?

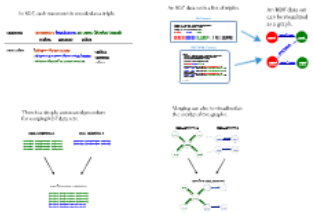
- SADI stands for Semantic Automated Discovery and Integration; it's our standard for RDF web services (more info later)
- SADI for GMOD is a set of CGI scripts for accessing sequence feature data as RDF
- "DAS for RDF"

Service Name	Input	Relationship	Output
get_feature_info	database identifier	is about	feature descriptor
get_features_overlapping_region	genomic coordinates	overlaps	collection of feature descriptors
get_sequence_for_region	genomic coordinates	is represented by	DNA, RNA, or amino acid sequence
get_child_features	feature descriptor	has part / derives into	collection of feature descriptors
get_parent_features	feature descriptor	is part of / derives from	collection of feature descriptors

What is the purpose of SADI for GMOD?

- Provide an easy way for GMODs to share their data as RDF, using a standardized protocol (SADI)
- Provide infrastructure for data integration across GMODs and other biology resources
- Support development of smarter bioinformatics software:
  - => Distributed queries
  - => Automated construction of web service workflows
  - => Assembly of datasets from multiple sources

RDF primer



What SADI for GMOD services are currently available?

To bootstrap participation, we are mirroring several GMODs (e.g. so far):

- AmoebaDB – E. histolytica
- CGD – C. albicans
- CryptoDB – C. hominis
- Flybase – D. melanogaster
- Genome – A. thaliana
- PlasmoDB – P. falciparum
- SGD – S. cerevisiae
- ToxoDB – T. gondii (501 strain)
- TriTrypDB – L. major

SADI for GMOD services are provided for each, e.g. http://rdf.wormbaseconnect.org/cgi-bin/flybase/get_feature_info

The SADI for GMOD services are in the public SADI registry.

http://sadi.mygrid.org.uk/registry/search

By the way, you can register your own SADI services here too.

http://sadiframework.org/registry

How do I use the services? (for GMOD users)

Demo (distributed query)

SHARE Query Resolution

Demo Query

How do I set up the services? (for GMOD providers)

1. Load your GFF files into a `Bio::DB::SeqFeature::Store` (will soon support Chado, also)
2. Install SADI for GMOD dependencies with CPAN
3. Download the SADI for GMOD tarball and unpack into `cgi-bin`
4. Set DB connection parameters in `cgi-bin/sadi.gmod/sadi.gmod.conf`

```
[GLOBAL]
db_adaptor  = Bio::DB::SeqFeature::Store
db_args     = -adaptor DBI::mysql
              -dsn     dbi:mysql:database=flybase
base_url    = http://flybase.org/cgi-bin/sadi.gmod/
```

5. Configure Obsref mappings in `cgi-bin/sadi.gmod/obsref.conf`

```
[OBSREF_TO_LSRN]
owlsProc = CalProc
OWLPeaCHA = CalProc
owlsProc/TrIESL = CalProc
...
```

6. Register the services in public SADI registry: `http://sadiframework.org/registry`

Future plans

- Chado support (soon!)
- add BLAST service (anything else that you want?)
- use cases and demos
- more GMOD mirrors
- page on GMOD wiki
- distribute with Tripol or GBrowse?

Acknowledgements

Team

Mark Wilkinson: Principal Investigator
Michel Dumontier: Principal Investigator (ontologies and data modelling)
Luke McCarthy: Lead Programmer, SADI & SHARE
Edward Kawas: Pod Programmer, SADI

Funding

canarie
HEARTS & STROKE FOUNDATION
Genome Canada
Genome BC
Microsoft Research

http://sadiframework.org/

Ben Vandervalk*[1], Luke McCarthy[1], Edward Kawas, Michel Dumontier[2], Mark Wilkinson[1]
James Hogg Research Institute, St. Paul's Hospital, University of British Columbia[1]
Carleton University[2]
*ben.vvalk@gmail.com

http://sadiframework.org/

SADI for GMOD: An RDF/OWL Interface for GMOD

http://code.google.com/p/sadi/wiki/SADIforGMOD

What is SADI for GMOD?

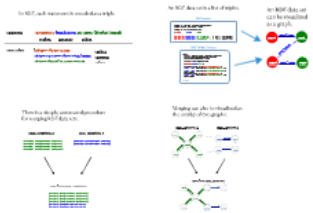
- SADI stands for Semantic Automated Discovery and Integration; it's our standard for RDF web services (more info later)
- SADI for GMOD is a set of CGI scripts for accessing sequence feature data as RDF
- "DAS for RDF"

Service Name	Input	Relationship	Output
get_feature_info	database identifier	is about	feature descriptor
get_features_overlapping_region	genomic coordinates	overlaps	collection of feature descriptors
get_sequence_for_region	genomic coordinates	is represented by	DNA, RNA, or amino acid sequence
get_child_features	feature descriptor	has part / derives into	collection of feature descriptors
get_parent_features	feature descriptor	is part of / derives from	collection of feature descriptors

What is the purpose of SADI for GMOD?

- Provide an easy way for GMODs to share their data as RDF, using a standardized protocol (SADI)
- Provide infrastructure for data integration across GMODs and other biology resources
- Support development of smarter bioinformatics software:
  - => Distributed queries
  - => Automated construction of web service workflows
  - => Assembly of datasets from multiple sources

RDF primer



What SADI for GMOD services are currently available?

To bootstrap participation, we are mirroring several GMODs (e.g. so far):

- AmoebaDB – E. histolytica
- CGD – C. albicans
- CryptoDB – C. hominis
- Flybase – D. melanogaster
- Genome – A. thaliana
- PlasmoDB – P. falciparum
- SGD – S. cerevisiae
- ToxoDB – T. gondii (501 strain)
- TriTrypDB – L. major

SADI for GMOD services are provided for each, e.g. http://rdf.wormbaseconnect.org/cgi-bin/flybase/get_feature_info

The SADI for GMOD services are in the public SADI registry.

http://sadi.mygrid.org.uk/registry/search

By the way, you can register your own SADI services here too.

http://sadiframework.org/registry

How do I use the services? (for GMOD users)

Demo (distributed query)

SHARE Query Resolution

Demo Query

How do I set up the services? (for GMOD providers)

1. Load your GFF files into a `Bio::DB::SeqFeature::Store` (will soon support Chado, also)
2. Install SADI for GMOD dependencies with CPAN
3. Download the SADI for GMOD tarball and unpack into `cgi-bin`
4. Set DB connection parameters in `cgi-bin/sadi.gmod/sadi.gmod.conf`

```
[GLOBAL]
db_adaptor  = Bio::DB::SeqFeature::Store
db_args     = -adaptor DBI::mysql
              -dsn     dbi:mysql:database=flybase
base_url    = http://flybase.org/cgi-bin/sadi.gmod/
```

5. Configure Obsref mappings in `cgi-bin/sadi.gmod/obsref.conf`

```
[OBSREF_TO_LSRN]
owlsProc = CalProc
OWLPeaCHA = CalProc
owlsProc/TrIESL = CalProc
...
```

6. Register the services in public SADI registry: `http://sadiframework.org/registry`

Future plans

- Chado support (soon!)
- add BLAST service (anything else that you want?)
- use cases and demos
- more GMOD mirrors
- page on GMOD wiki
- distribute with Tripol or GBrowse?

Acknowledgements

Team

Mark Wilkinson: Principal Investigator
Michel Dumontier: Principal Investigator (ontologies and data modelling)
Luke McCarthy: Lead Programmer, SADI & SHARE
Edward Kawas: Pod Programmer, SADI

Funding

canarie
HEARTS & STROKE FOUNDATION
Genome Canada
Genome BC
Microsoft Research

http://sadiframework.org/

Ben Vandervalk*[1], Luke McCarthy[1], Edward Kawas, Michel Dumontier[2], Mark Wilkinson[1]
James Hogg Research Institute, St. Paul's Hospital, University of British Columbia[1]
Carleton University[2]
*ben.vvalk@gmail.com

http://sadiframework.org/

What is SADI for GMOD?

- SADI stands for Semantic Automated Discovery and Integration; it's our standard for RDF web services (more info later)
- SADI for GMOD is a set of CGI scripts for accessing sequence feature data as RDF
- "DAS for RDF"

Service Name	Input	Relationship	Output
get_feature_info	database identifier	is about	feature description
get_features_ overlapping_region	genomic coordinates	overlaps	collection of feature descriptions
get_sequence_ for_region	genomic coordinates	is represented by	DNA, RNA, or amino acid sequence
get_child_features	feature description	has part / derives into	collection of feature descriptions
get_parent_features	feature description	is part of / derives from	collection of feature descriptions

What is the purpose of SADI for GMOD?

- Provide an easy way for GMODs to share their data as RDF, using a standardized protocol (SADI)
- Provide infrastructure for data integration across GMODs and other biology resources
- Support development of smarter bioinformatics software:

  => Distributed queries

  => Automated construction of web service workflows

  => Assembly of datasets from multiple sources

RDF primer

In RDF, each statement is encoded as a triple.

Statement: "Hexokinase 1 participates in the human glycolysis pathway."

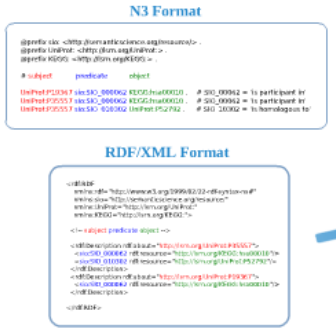
Subject *Predicate* *Object*

RDF Triple: (http://lsrn.org/UniProt:P19367, *Subject*

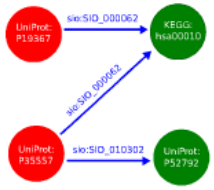
http://semanticscience.org/resource/SIO_000062, *Predicate*

http://lsrn.org/KEGG_PATHWAY:hsa00010) *Object*

An RDF data set is a list of triples.



An RDF data set can be visualized as a graph.



There is a simple, automated procedure for merging RDF data sets.

RDF Dataset 1

(ProteinA, ParticipatesIn, PathwayA)
(ProteinB, ParticipatesIn, PathwayA)
(ProteinC, ParticipatesIn, PathwayA)
(ProteinD, ParticipatesIn, PathwayA)

RDF Dataset 2

(ProteinA, HasHomolog, ProteinE)
(ProteinB, HasHomolog, ProteinF)

Merged RDF Dataset

(ProteinA, ParticipatesIn, PathwayA)
(ProteinB, ParticipatesIn, PathwayA)
(ProteinC, ParticipatesIn, PathwayA)
(ProteinD, ParticipatesIn, PathwayA)
(ProteinA, HasHomolog, ProteinE)
(ProteinB, HasHomolog, ProteinF)

Merging can also be visualized as the overlay of two graphs.

RDF Dataset 1

RDF Dataset 2

Merged RDF Dataset

In RDF, each statement is encoded as a triple.

Statement: "Hexokinase 1 participates in the human glycolysis pathway."

Subject *Predicate* *Object*

RDF Triple: (http://lsrn.org/UniProt:P19367, *Subject*

http://semanticscience.org/resource/SIO_000062, *Predicate*

http://lsrn.org/KEGG_PATHWAY:hsa00010) *Object*

An RDF data set is a list of triples.

N3 Format

```
@prefix sio: <http://semanticscience.org/resource/> .
@prefix UniProt: <http://lsrn.org/UniProt:> .
@prefix KEGG: <http://lsrn.org/KEGG:> .

# subject        predicate        object

UniProt:P19367 sio:SIO_000062 KEGG:hsa00010 .    # SIO_00062 = 'is participant in'
UniProt:P35557 sio:SIO_000062 KEGG:hsa00010 .    # SIO_00062 = 'is participant in'
UniProt:P35557 sio:SIO_010302 UniProt:P52792 .    # SIO_10302 = 'is homologous to'
```

RDF/XML Format

```
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:sio="http://semanticscience.org/resource/"
   xmlns:UniProt="http://lsrn.org/UniProt:"
   xmlns:KEGG="http://lsrn.org/KEGG:">

  <!-- subject predicate object -->

  <rdf:Description rdf:about="http://lsrn.org/UniProt:P35557">
    <sio:SIO_000062 rdf:resource="http://lsrn.org/KEGG:hsa00010"/>
    <sio:SIO_010302 rdf:resource="http://lsrn.org/UniProt:P52792"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://lsrn.org/UniProt:P19367">
    <sio:SIO_000062 rdf:resource="http://lsrn.org/KEGG:hsa00010"/>
  </rdf:Description>

</rdf:RDF>
```

set is a list of triples.

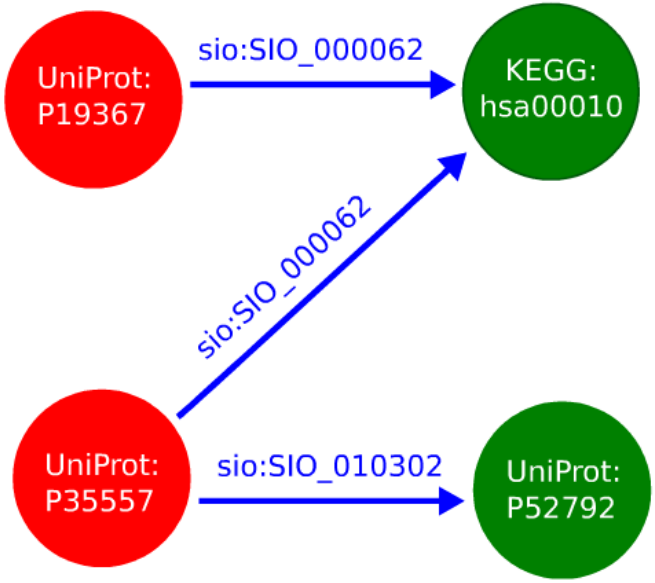
N3 Format

```
nce.org/resource/> .
UniProt:> .
EGG:> .

bject

EGG:hsa00010 .    # SIO_00062 = 'is participant in'
EGG:hsa00010 .    # SIO_00062 = 'is participant in'
niProt:P52792 .   # SIO_10302 = 'is homologous to'
```

F/XML Format

```
www.w3.org/1999/02/22-rdf-syntax-ns#"
semanticscience.org/resource/"
tp://lsrn.org/UniProt:"
://lsrn.org/KEGG:">

te object -->

 about="http://lsrn.org/UniProt:P35557">
df:resource="http://lsrn.org/KEGG:hsa00010"/>
df:resource="http://lsrn.org/UniProt:P52792"/>

 about="http://lsrn.org/UniProt:P19367">
df:resource="http://lsrn.org/KEGG:hsa00010"/>
```

An RDF data set can be visualized as a graph.



There is a simple, automated procedure for merging RDF data sets.

RDF Dataset 1

(ProteinA, ParticipatesIn, PathwayA)
(ProteinB, ParticipatesIn, PathwayA)
(ProteinC, ParticipatesIn, PathwayA)
(ProteinD, ParticipatesIn, PathwayA)

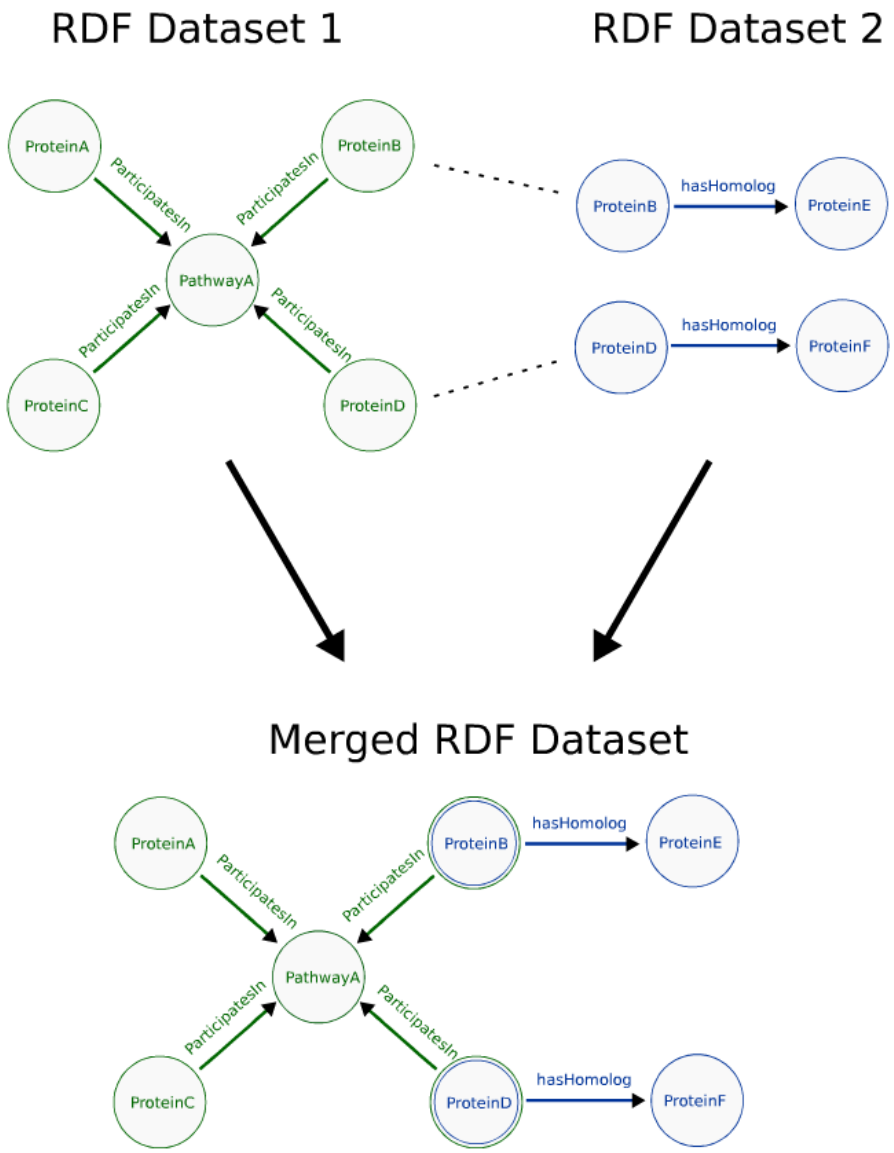
RDF Dataset 2

(ProteinA, HasHomolog, ProteinE)
(ProteinB, HasHomolog, ProteinF)

Merged RDF Dataset

(ProteinA, ParticipatesIn, PathwayA)
(ProteinB, ParticipatesIn, PathwayA)
(ProteinC, ParticipatesIn, PathwayA)
(ProteinD, ParticipatesIn, PathwayA)
(ProteinA, HasHomolog, ProteinE)
(ProteinB, HasHomolog, ProteinF)

Merging can also be visualized as the overlay of two graphs.



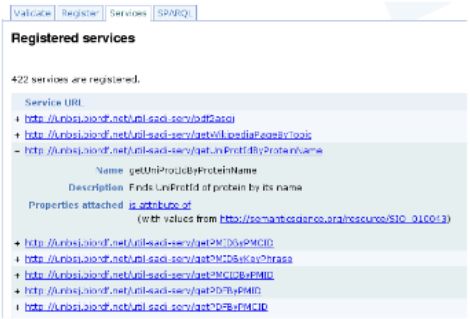
What SADI for GMOD services are currently available?

To bootstrap participation, we are mirroring several GMODs (9 so far):

- AmoebaDB -- E. histolytica
- CGD -- C. albicans
- CryptoDB -- C. hominis
- FlyBase -- D. melanogaster
- Gramene -- A. thaliana
- PlasmoDB -- P. falciparum
- SGD -- S. cerevisiae
- ToxoDB -- T. gondii (RH strain)
- TriTrypDB -- L. major

SADI for GMOD services are provided for each, e.g.
http://s7.semanticscience.org/~ben/cgi-bin/FlyBase/get_feature_info

The SADI for GMOD services are in the public SADI registry.



http://sadiframework.org/registry/services/

By the way, you can register your own SADI services here too.

http://sadiframework.org/registry/register/

To bootstrap participation, we are mirroring several GMODs (9 so far):

- AmoebaDB -- E. histolytica
- CGD -- C. albicans
- CryptoDB -- C. hominis
- FlyBase -- D. melanogaster
- Gramene -- A. thaliana
- PlasmoDB -- P. falciparum
- SGD -- S. cerevisiae
- ToxoDB -- T. gondii (RH strain)
- TriTrypDB -- L. major

SADI for GMOD services are provided for each, e.g. http://s7.semanticscience.org/~ben/cgi-bin/FlyBase/get_feature_info

The SADI for GMOD services are in the public SADI registry.

Validate | Register | Services | SPARQL

Registered services

422 services are registered.

Service URL
+ http://unbsj.biordf.net/util-sadi-serv/pdf2ascii
+ http://unbsj.biordf.net/util-sadi-serv/getWikipediaPageByTopic
– http://unbsj.biordf.net/util-sadi-serv/getUniProtIdByProteinName
Name getUniProtIdByProteinName
Description Finds UniProtId of protein by its name
Properties attached is attribute of (with values from http://semanticscience.org/resource/SIO_010043)
+ http://unbsj.biordf.net/util-sadi-serv/getPMIDByPMCID
+ http://unbsj.biordf.net/util-sadi-serv/getPMIDByKeyPhrase
+ http://unbsj.biordf.net/util-sadi-serv/getPMCIDByPMID
+ http://unbsj.biordf.net/util-sadi-serv/getPDFByPMID
+ http://unbsj.biordf.net/util-sadi-serv/getPDFByPMCID

http://sadiframework.org/registry/services/

By the way, you can register your own SADI services here too.

http://sadiframework.org/registry/register/

How do I use the services? (for GMOD users)

SADI in a Nutshell

- **to invoke a SADI service:**
  - HTTP POST an RDF document to the service URL
- **to get service metadata**:
  - HTTP GET on service URL
  - returns an RDF document with service name, description, etc.
- **structure of input/output data is described in OWL**
  - service provider specifies one *input OWL class* and one *output OWL class*
- **strengths of SADI**
  - no framework-specific messaging formats or ontologies
  - supports batch processing of inputs
  - supports long-running services (asynchronous services)

SADI for GMOD: Structure of Service Input/Output RDF

Input RDF (N3)

HTTP POST

get_feature_info

Output RDF (N3)

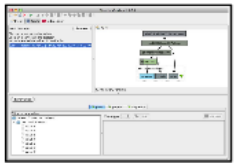
SADI Client Software

SHARE Query Engine

SPARQL Query => SADI Workflow

http://biordf.net/cardioSHARE/query

SADI Taverna Plugin



Design SADI Workflows

http://sadiframework.org/content/2010/05/03/sadi-taverna-plugin-tutorial/

SADI in a Nutshell

- **to invoke a SADI service:**
  - HTTP POST an RDF document to the service URL
- **to get service metadata**:
  - HTTP GET on service URL
  - returns an RDF document with service name, description, etc.
- **structure of input/output data is described in OWL**
  - service provider specifies one *input OWL class* and one *output OWL class*
- **strengths of SADI**
  - no framework-specific messaging formats or ontologies
  - supports batch processing of inputs
  - supports long-running services (asynchronous services)

SADI for GMOD: Structure of Service Input/Output RDF

ass

Input RDF (N3)

```
@prefix lsrn: <http://purl.oclc.org/SADI/LSRN/> .
@prefix GeneID: <http://lsrn.org/GeneID:> .

GeneID:49962
    a lsrn:GeneID_Record;
    sio:SIO_000008 [ # p = 'has attribute'
        a lsrn:GeneID_Identifier;
        sio:SIO_000300 "49962" # p = 'has value'
    ] .
```

HTTP
POST

get_feature_info

Output RDF (N3)

```
@prefix region:
<http://sadiframework.org/ontologies/GMOD/BiopolymerRegion.owl#> .
@prefix range:
<http://sadiframework.org/ontologies/GMOD/RangedSequencePosition.owl#> .
@prefix strand: <http://sadiframework.org/ontologies/GMOD/Strand.owl#> .
@prefix lsrn: <http://purl.oclc.org/SADI/LSRN/> .
@prefix GeneID: <http://lsrn.org/GeneID:> .
@prefix FlyBase: <http://flybase.org/cgi-bin/sadi.gmod/feature?id=> .
@prefix sio: <http://semanticscience.org/resource/> .
@prefix SO: <http://purl.org/obo/owl/SO#> .

# p = 'is about'
GeneID:49962 sio:SIO_000332 FlyBase:FBgn0040037 .

# feature

FlyBase:FBgn0040037
    a SO:SO_0000704 . # o = 'gene'
    range:position  [
        a range:RangedSequencePosition;
        range:coordinate
             [ a range:StartPosition; sio:SIO_000300 26994];
        range:coordinate
            [ a range:EndPosition; sio:SIO_000300 32391];
        range:in_relation_to _:minus_strand_seq
    ] .

_:minus_strand_seq
    sio:SIO_000011 [ # p = 'represents'
        a strand:MinusStrand;
        sio:SIO_000093 FlyBase:4 # p = 'is proper part of'
    ] .

# reference feature (chromosome)

FlyBase:4  # chromosome 4
    a SO:SO_0000105 . # o = 'chromosome arm'
```

Input RDF (N3)

```
@prefix lsrn: <http://purl.oclc.org/SADI/LSRN/> .
@prefix GeneID: <http://lsrn.org/GeneID:> .

GeneID:49962
    a lsrn:GeneID_Record;
    sio:SIO_000008 [ # p = 'has attribute'
        a lsrn:GeneID_Identifier;
        sio:SIO_000300 "49962" # p = 'has value'
    ] .
```

```
@prefix sio: <http://semanticscience.org/resource/> .
@prefix SO: <http://purl.org/obo/owl/SO#> .

# p = 'is about'
GeneID:49962 sio:SIO_000332 FlyBase:FBgn0040037 .

# feature

FlyBase:FBgn0040037
    a SO:SO_0000704 . # o = 'gene'
    range:position  [
        a range:RangedSequencePosition;
        range:coordinate
             [ a range:StartPosition; sio:SIO_000300 26994];
        range:coordinate
            [ a range:EndPosition; sio:SIO_000300 32391];
        range:in_relation_to _:minus_strand_seq
    ] .

_:minus_strand_seq
    sio:SIO_000011 [ # p = 'represents'
        a strand:MinusStrand;
        sio:SIO_000093 FlyBase:4 # p = 'is proper part of'
    ] .

# reference feature (chromosome)

FlyBase:4  # chromosome 4
    a SO:SO_0000105 . # o = 'chromosome arm'
```

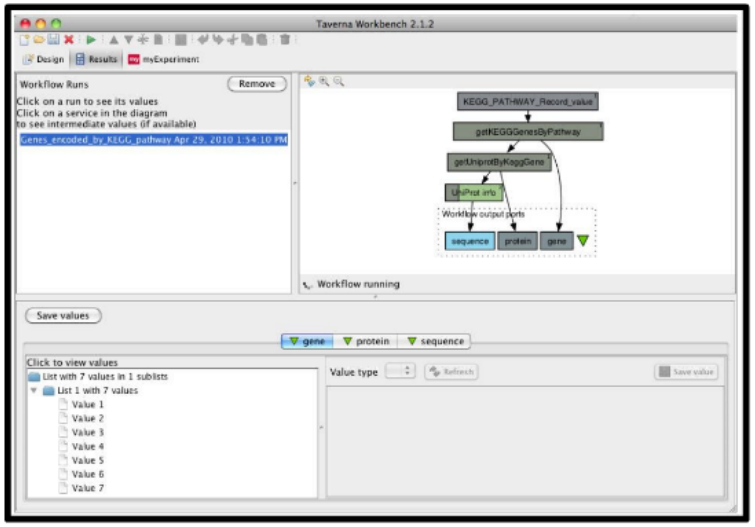
SADI Client Software

SHARE Query Engine

SPARQL Query => SADI Workflow

http://biordf.net/cardioSHARE/query

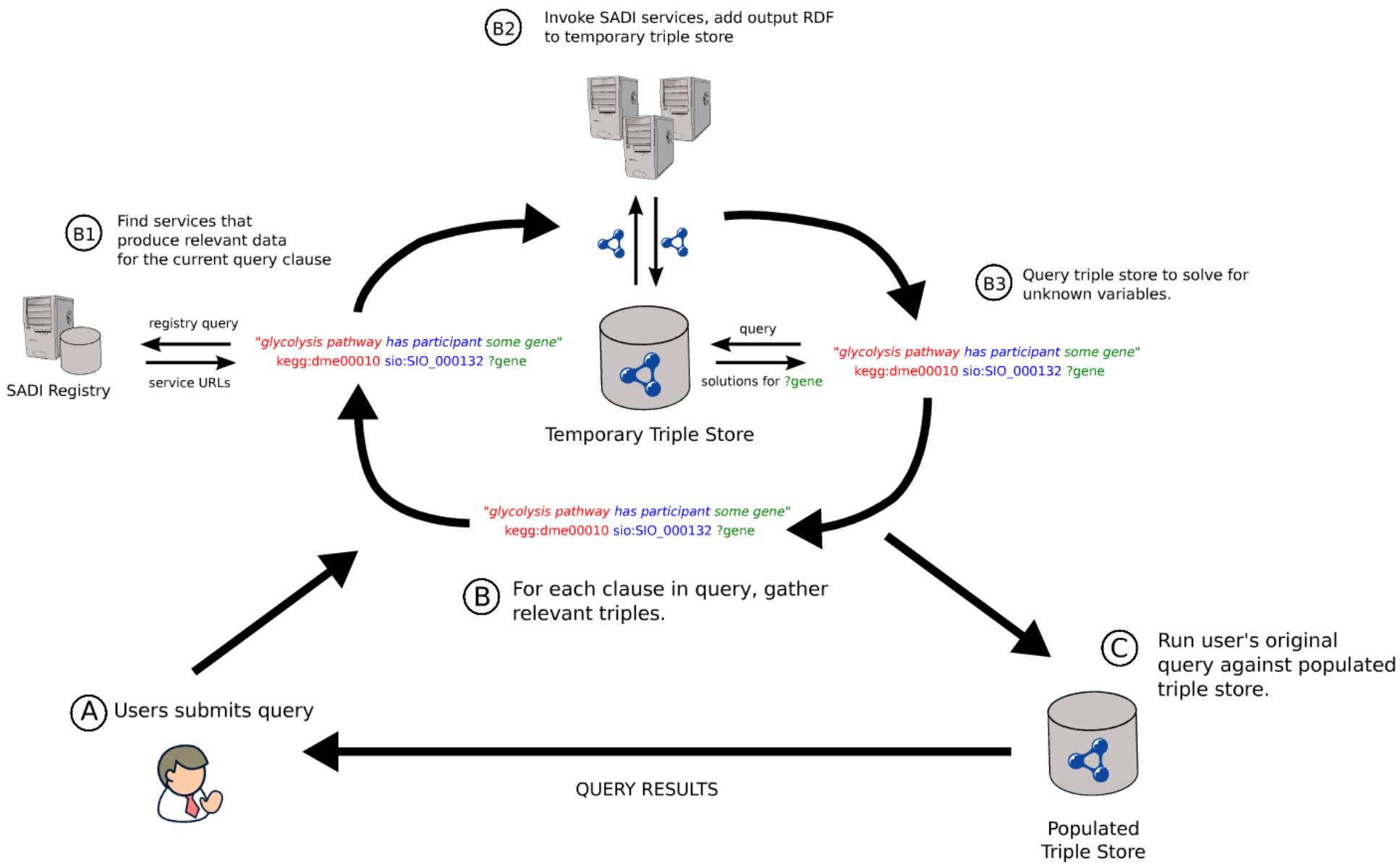
SADI Taverna Plugin



Design SADI Workflows

http://sadiframework.org/content/2010/05/03/sadi-taverna-plugin-tutorial/

SHARE Query Resolution

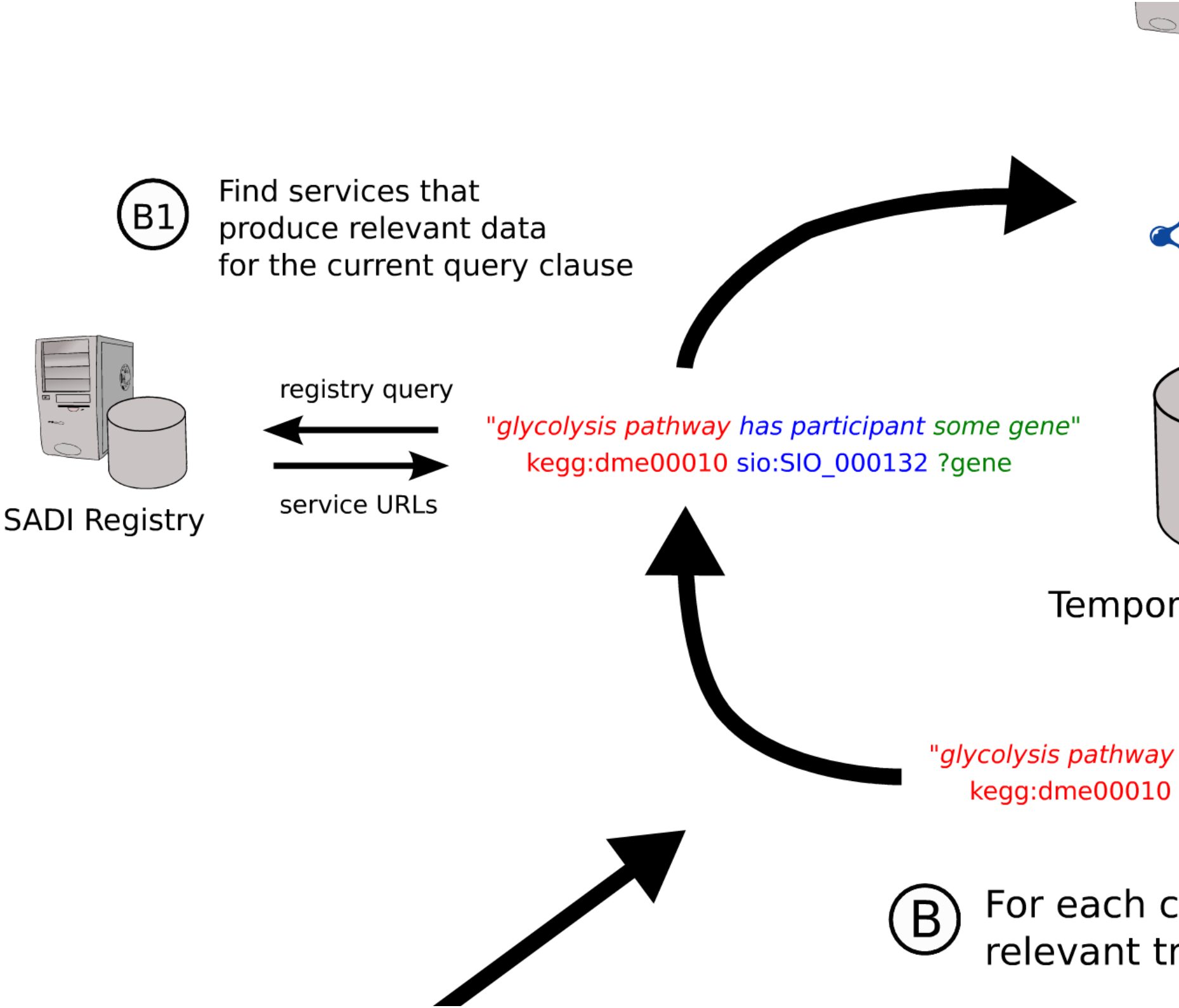


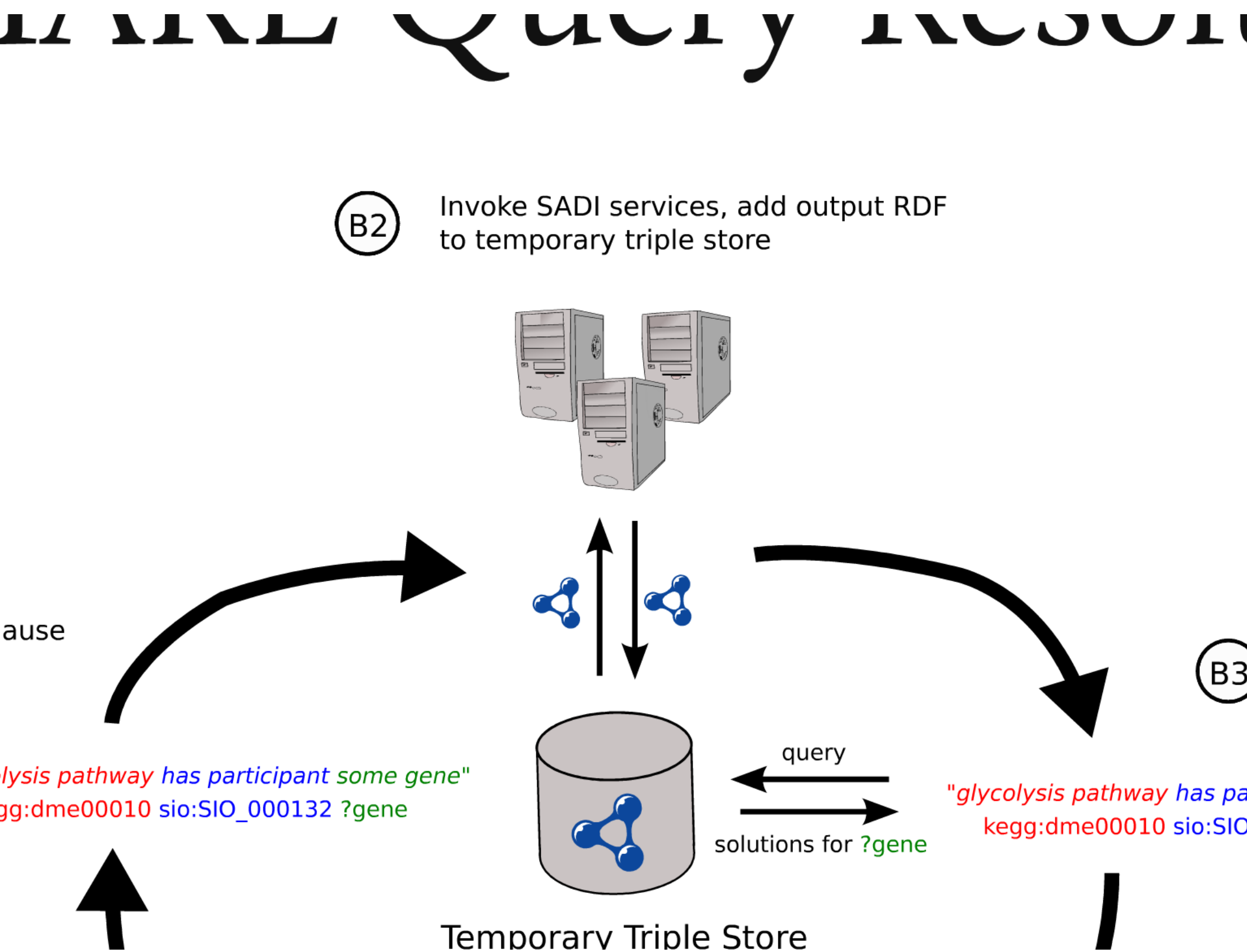
"*glycolysis pathway has participant some gene*"

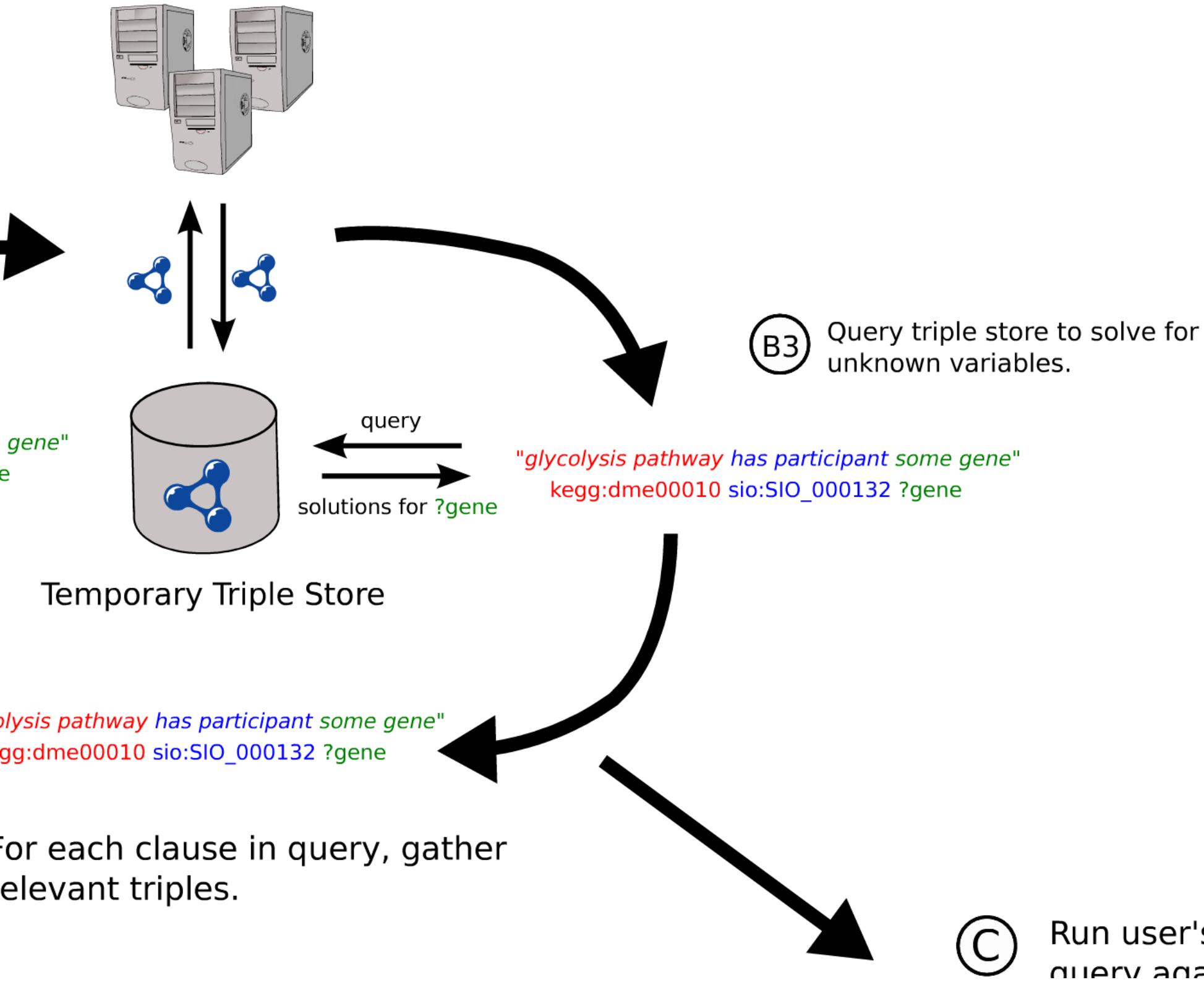
kegg:dme00010 sio:SIO_000132 ?gene

Ⓑ For each clause in query, gather relevant triples.

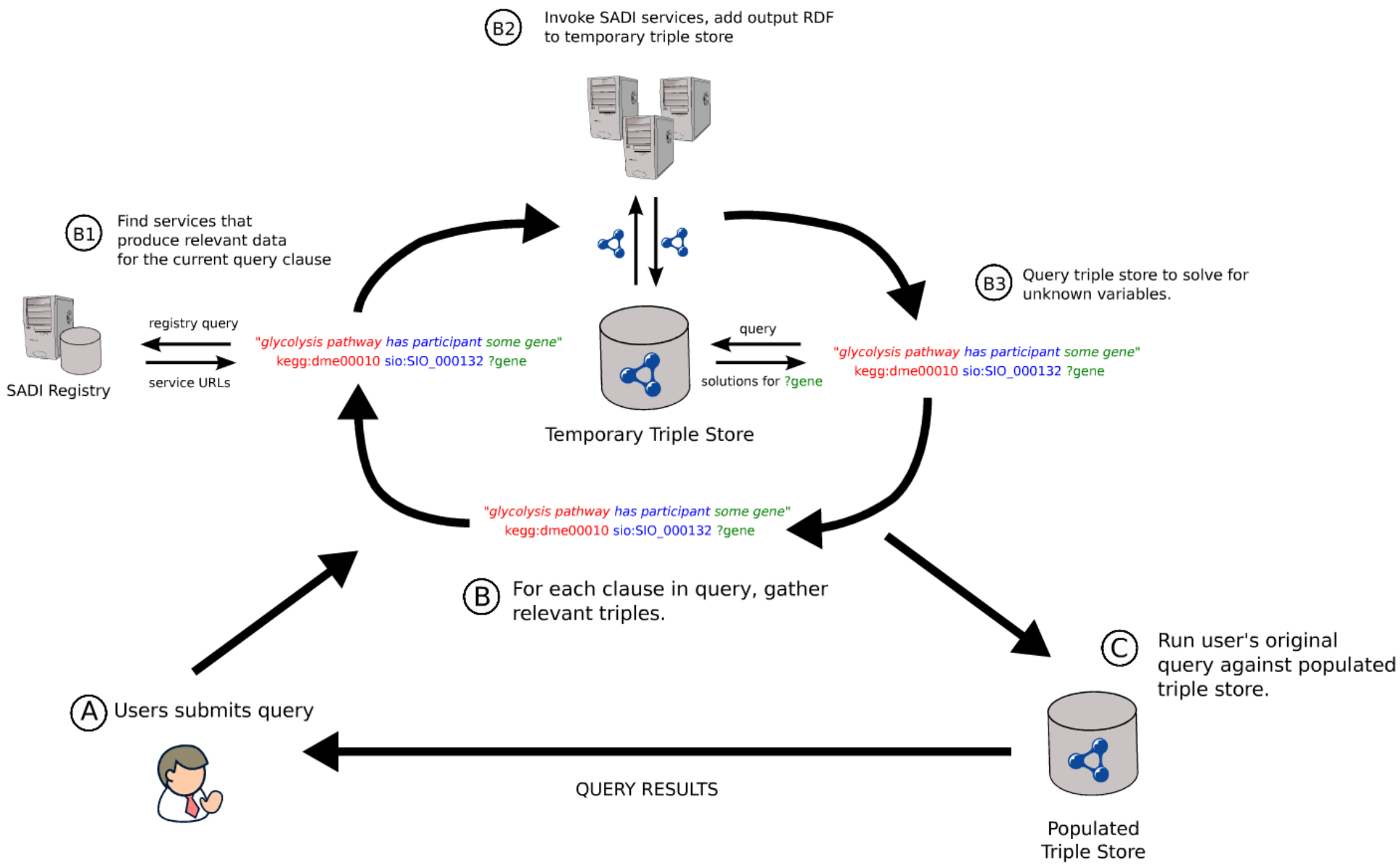
QUERY RESULTS


B1
Find services that
produce relevant data
for the current query clause
registry query
service URLs
SADI Registry
"glycolysis pathway has participant some gene"
kegg:dme00010 sio:SIO_000132 ?gene
Tempor
"glycolysis pathway
kegg:dme00010
B
For each c
relevant tr


B2
Invoke SADI services, add output RDF
to temporary triple store
ause
B3
lysis pathway has participant some gene"
gg:dme00010 sio:SIO_000132 ?gene
query
solutions for ?gene
"glycolysis pathway has pa
kegg:dme00010 sio:SIO
Temporary Triple Store


B3
Query triple store to solve for unknown variables.
query
solutions for ?gene
"glycolysis pathway has participant some gene"
kegg:dme00010 sio:SIO_000132 ?gene
Temporary Triple Store
gene"
e
olysis pathway has participant some gene"
gg:dme00010 sio:SIO_000132 ?gene
For each clause in query, gather
elevant triples.
C
Run user's origi
query against p

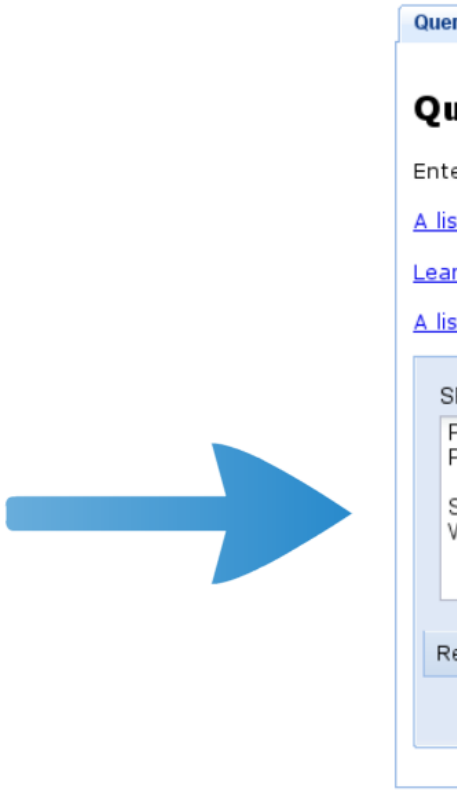
SHARE Query Resolution



Demo Query

"Find FlyBase genes that participate in glycolysis and overlap genomic region 5,919,623..6,344,662 on chromosome 3L"

```
PREFIX feature: <http://s7.semanticscience.org/~ben/cgi-bin/FlyBase/feature?id=>
PREFIX range: <http://sadiframework.org/ontologies/GMOD/RangedSequencePosition.owl#>
PREFIX region: <http://sadiframework.org/ontologies/GMOD/BiopolymerRegion.owl#>
PREFIX strand: <http://sadiframework.org/ontologies/GMOD/Strand.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX sio: <http://semanticscience.org/resource/>
PREFIX pathway: <http://lsrn.org/KEGG_PATHWAY:>

SELECT ?flybase_gene_record ?startpos ?endpos ?strand_type
WHERE {

  pathway:dme00010     sio:SIO_000132  ?kegg_gene_record .      # SIO_000132 = 'has participant'
  ?kegg_gene_record    owl:sameAs      ?flybase_gene_record .
  ?flybase_gene_record sio:SIO_000332  ?flybase_feature .       # SIO_000332 = 'is about'

  ?flybase_feature region:position  [
    range:coordinate [ rdf:type range:StartPosition; sio:SIO_000300 ?startpos ];
    range:coordinate [ rdf:type range:EndPosition; sio:SIO_000300 ?endpos ];
    range:in_relation_to [
      sio:SIO_000210 [                                          # SIO_000210 = 'represents'
        rdf:type ?strand_type;
        sio:SIO_000093 feature:3L                               # SIO_000093 = 'is proper part of'
      ]
    ]
  ];

  FILTER ((?endpos >= 5919623) && (?startpos <= 6344662))
  FILTER ((?strand_type = strand:MinusStrand) || (?strand_type = strand:PlusStrand))

}
```



```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX sio: <http://semanticscience.org/resource/>
PREFIX pathway: <http://lsrn.org/KEGG_PATHWAY:>

SELECT ?flybase_gene_record ?startpos ?endpos ?strand_type
WHERE {

   pathway:dme00010      sio:SIO_000132   ?kegg_gene_record .        # SIO_000132 = 'has parti
   ?kegg_gene_record     owl:sameAs       ?flybase_gene_record .
   ?flybase_gene_record  sio:SIO_000332   ?flybase_feature .          # SIO_000332 = 'is about'

   ?flybase_feature region:position  [
      range:coordinate [ rdf:type range:StartPosition; sio:SIO_000300 ?startpos ];
      range:coordinate [ rdf:type range:EndPosition; sio:SIO_000300 ?endpos ];
      range:in_relation_to [
         sio:SIO_000210 [                                            # SIO_000210 = 'represen
            rdf:type ?strand_type;
            sio:SIO_000093 feature:3L                                # SIO_000093 = 'is prope
         ]
      ]
   ];

   FILTER ((?endpos >= 5919623) && (?startpos <= 6344662))
   FILTER ((?strand_type = strand:MinusStrand) || (?strand_type = strand:PlusStrand))

}
```

Query | Browse

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.

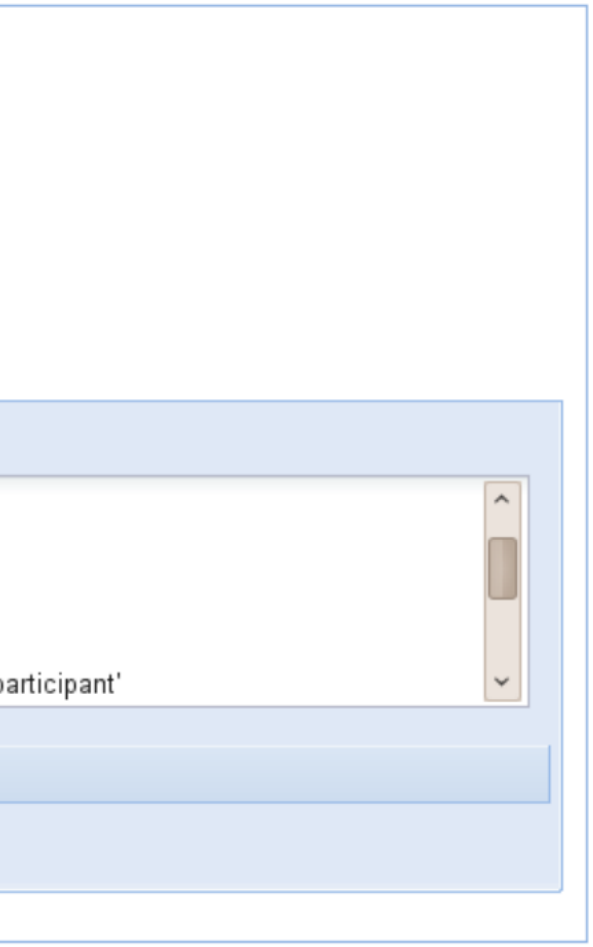
SPARQL query:

```
PREFIX sio: <http://semanticscience.org/resource/>
PREFIX pathway: <http://lsrn.org/KEGG_PATHWAY:>

SELECT ?flybase_gene_record ?startpos ?endpos ?strand_type
WHERE {

    pathway:dme00010      sio:SIO_000132   ?kegg_gene_record .          # SIO_000132 = 'has participant'
```

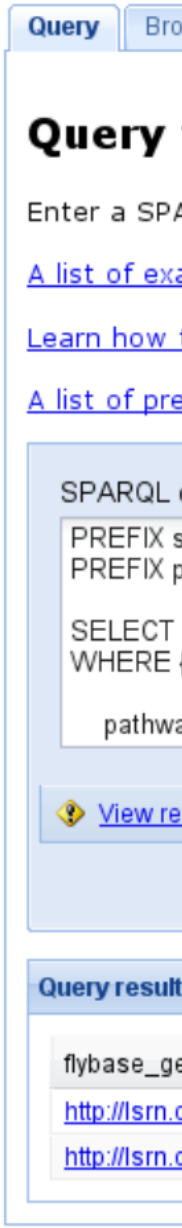
Ready.

Submit

Submit the query to SHARE.



Wait a while...



RE.

Query | Browse

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 sio: <http://semanticscience.org/resource/>
PREFIX pathway: <http://lsrn.org/KEGG_PATHWAY:>

SELECT ?flybase_gene_record ?startpos ?endpos ?strand_type
WHERE {

    pathway:dme00010      sio:SIO_000132   ?kegg_gene_record .          # SIO_000132 = 'has participant'
```

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

Submit

Query results

flybase_gene_record	startpos	endpos	strand_type
http://lsrn.org/FLYBASE:FBgn0001258	6252592	6255793	http://sadiframework.org/ontologies/GMOD/Strand.owl#MinusStrand
http://lsrn.org/FLYBASE:FBgn0035679	6086304	6087836	http://sadiframework.org/ontologies/GMOD/Strand.owl#MinusStrand

How do I set up the services? (for GMOD providers)

1. Load your GFF files into a `Bio::DB::SeqFeature::Store` (will soon support Chado, also)

2. Install SADI for GMOD dependencies with CPAN

3. Download the SADI for GMOD tarball and unpack into `cgi-bin`

4. Set DB connection parameters in `cgi-bin/sadi.gmod/sadi.gmod.conf`

```
[GENERAL]
db_adaptor = Bio::DB::SeqFeature::Store
db_args    = -adaptor DBI::mysql
             -dsn     dbi:mysql:database=flybase
base_url   = http://flybase.org/cgi-bin/sadi.gmod/
```

5. Configure Dbxref mappings in `cgi-bin/sadi.gmod/dbxref.conf`

```
[DBXREF_TO_LSRN]
SwissProt = UniProt
UniProtKB = UniProt
SwissProt/TrEMBL = UniProt
...
```

6. Register the services in public SADI registry: `http://sadiframework.org/registry`

3. Download the SADI for GMOD tarball and unpack into `cgi-bin`

4. Set DB connection parameters in `cgi-bin/sadi.gmod/sadi.gmod.conf`

```
[GENERAL]
db_adaptor = Bio::DB::SeqFeature::Store
db_args    = -adaptor DBI::mysql
             -dsn     dbi:mysql:database=flybase
base_url   = http://flybase.org/cgi-bin/sadi.gmod/
```

5. Configure Dbxref mappings in `cgi-bin/sadi.gmod/dbxref.conf`

```
[DBXREF_TO_LSRN]
SwissProt = UniProt
UniProtKB = UniProt
SwissProt/TrEMBL = UniProt
...
```

6. Register the services in public SADI registry: `http://sadiframework.org/re`

```
db_args     = -adaptor DBI::mysql
              -dsn     dbi:mysql:database=flybase
base_url    = http://flybase.org/cgi-bin/sadi.gmod/
```

5. Configure Dbxref mappings in `cgi-bin/sadi.gmod/dbxref.conf`

```
[DBXREF_TO_LSRN]
SwissProt = UniProt
UniProtKB = UniProt
SwissProt/TrEMBL = UniProt
...
```

6. Register the services in public SADI registry: `http://sadiframework.org/re`

Ben Vandervalk*[1], L
James Hogg Rese

Future plans

- Chado support (soon!)
- add BLAST service (anything else that you want?)
- use cases and demos
- more GMOD mirrors
- page on GMOD wiki
- distribute with Tripal or GBrowse?

Acknowledgements

Team

Mark Wilkinson: Principal Investigator
Michel Dumontier: Principal Investigator (ontologies and data modelling)
Luke McCarthy: Lead Programmer, SADI & SHARE
Edward Kawas: Perl Programmer, SADI

Funding

canarie
Canada's Advanced Research and Innovation Network
Le réseau évolué de recherche et d'innovation du Canada

HEART & STROKE FOUNDATION

NSERC CRSNG

CIHR IRSC
Canadian Institutes of Health Research
Instituts de recherche en santé du Canada

GenomeCanada

Microsoft Research

http://sadiframework.org/