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 Association Discovery and Integration
- It is an open source RDF platform and service registry
- SADI provides a way for GMOD services to share their data in RDF

<table>
<thead>
<tr>
<th>Source</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMOD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SADI</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

What is the purpose of SADI for GMOD?
- Provide an easy way for GMOD services to share their data in RDF using a standardized protocol (SADI)
- Provide infrastructure for data integration across GMODs and other bioinformatics software
- Support development of Semantic web service workflows
- Assembly of datasets from multiple sources

RDF primer

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

How do I set up the services?
(for GMOD providers)
1. Set up a GMOD service (as above)
2. Install the SADI server on your local machine
3. Enrich the SADI for GMOD standard and framework
4. Configure SADI services to use GMOD schema
5. Register the service in the SADI registry

Future plans
- Code support
- GMOD service (adding the view provider)
- Add new views for GMOD
- Database with Triples in OWL

Acknowledgements

Ben Vandervalk¹,³, Luke McCarthy¹, Edward Kawas, Michel Dumontier², Mark Wilkinson¹
James Hogg Research Institute, St. Paul’s Hospital, University of British Columbia¹
Carleton University²
*ben.valk@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 Access, Discovery and Integration;
- A web orientated RDF publish and share interface;
- SADI utilizes a set of OWL-based services for accessing, querying, publishing data

<table>
<thead>
<tr>
<th>Type</th>
<th>RDF/OWL Service</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDF/OWL</td>
<td>SADI</td>
<td>Semantic Access, Discovery and Integration</td>
</tr>
</tbody>
</table>

What is the purpose of SADI for GMOD?
- Provide an easy way for GMOD to share their data in RDF using a standardized protocol (SADI);
- Provide infrastructure for data integration across GMOD and other biology resources;
- Support development of semantic bio-informatics 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?

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

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

Future plans
- Candidate for future GMOD release
- SADI/OWL service enabling easier access
- GMOD/OWL service enabling easier access to GMOD data
- Integration with Tripal (Drupal)

Acknowledgements

Ben Vandervalk, Luke McCarthy, Edward Kawas, Michel Dumontier, Mark Wilkinson
James Hogg Research Institute, St. Paul's Hospital, University of British Columbia
Carleton University
*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"

<table>
<thead>
<tr>
<th>Service Name</th>
<th>Input</th>
<th>Relationship</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>get_feature_info</td>
<td>database identifier</td>
<td>is about</td>
<td>feature description</td>
</tr>
<tr>
<td>get_features_overlapping_region</td>
<td>genomic coordinates</td>
<td>overlaps</td>
<td>collection of feature descriptions</td>
</tr>
<tr>
<td>get_sequence_for_region</td>
<td>genomic coordinates</td>
<td>is represented by</td>
<td>DNA, RNA, or amino acid sequence</td>
</tr>
<tr>
<td>get_child_features</td>
<td>feature description</td>
<td>has part / derives into</td>
<td>collection of feature descriptions</td>
</tr>
<tr>
<td>get_parent_features</td>
<td>feature description</td>
<td>is part of / derives from</td>
<td>collection of feature descriptions</td>
</tr>
</tbody>
</table>
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."

<table>
<thead>
<tr>
<th>Subject</th>
<th>Predicate</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://purl.org/ko/Proteins/HEX1">http://purl.org/ko/Proteins/HEX1</a></td>
<td>has_participated_in</td>
<td><a href="http://purl.org/ko/Pathways/Glycolysis">http://purl.org/ko/Pathways/Glycolysis</a></td>
</tr>
</tbody>
</table>

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

**RDF Dataset 1**
- Protein, participates_in, Pathway
- Protein, participates_in, Pathway
- Protein, participates_in, Pathway

**RDF Dataset 2**
- Protein, has_participated_in, Process
- Protein, has_participated_in, Process

**Merged RDF Dataset**
- Protein, participates_in, Pathway
- Protein, has_participated_in, Process
- Protein, has_participated_in, Process

An RDF data set is a list of triples.

**N3 Format**

An RDF data set can be visualized as a graph.

**RDF/XML Format**

Merging can also be visualized as the overlay of two graphs.
In RDF, each statement is encoded as a triple.

Statement: “Hexokinase 1 participates in the human glycolysis pathway.”

Subject   Predicate   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_000062 = 'is participant in'
UniProt:P35557 sio:SIO_000062 KEGG:hsa00010 .  # SIO_000062 = 'is participant in'
UniProt:P35557 sio:SIO_010302 UniProt:P52792 .  # SIO_010302 = '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">
</rdf:Description>

<rdf:Description rdf:about="http://lsrn.org/UniProt:P19367">
</rdf:Description>

</rdf:RDF>
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
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/~ber/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.

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

SADI Client Software

SHARE Query Engine

SPARQL Query => SADI Workflow
http://bowl2.net/corida/STH/httf/query

SADI Taverna Plugin

Design SADI Workflows
http://sadi/taverna.org/content/2.9.0/051223/sadi-to-taverna-plugin-manual/
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)

@prefix lرن: <http://purl.oclc.org/SADI/LSRN/> .
@prefix GeneID: <http://lsr.org/GeneID/> .

GeneID:49962
a lرن:GeneID_Record;
sio:SI0_000300  [ # p = 'has attribute'
a lرن:GeneID_Identifier;
sio:SI0_000300 "49962" # p = 'has value'
] .

Output RDF (N3)

@prefix region:
<http://sadiframework.org/ontologies/GMOD/BiopolymerRegion.owl#> .
@prefix range:
@prefix strand:
<http://sadiframework.org/ontologies/GMOD/Strand.owl#> .
@prefix lرن: <http://purl.oclc.org/SADI/LSRN/> .
@prefix GeneID: <http://lsr.org/GeneID/> .
@prefix FlyBase: <http://flybase.org/cgi-bin/sadi.gmod/feature?rid=0> .
@prefix sio: <http://semanticweb.org/resource/> .
@prefix SO: <http://purl.obo/owl/0000400> .

# p = 'is about'
GeneID:49962 sio:SI0_000332 FlyBase:FBgn0040037 .

# feature
FlyBase:FBgn0040037
a SO:00004004 . # o = 'gene'
range:position [ a range:RangeSequencePosition;
range:coordinate
 [ a range:StartPosition; sio:SI0_000300 26994];
range:coordinate
 [ a range:EndPosition; sio:SI0_000300 32391];
range:in relation to _minus_strand_seq
].

:_minus_strand_seq
sio:SI0_000011 [ # p = 'represents'
a strand:MinusStrand;
sio:SI0_000093 FlyBase:4 # p = 'is proper part of'
] .

# reference feature (chromosome)
FlyBase:4  # chromosome 4
a SO:0000105 . # o = 'chromosome arm'
@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'

# 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

A. Users submits query

B. For each clause in query, gather relevant triples.

B1. Find services that produce relevant data for the current query clause

B2. Invoke SADI services, add output RDF to temporary triple store

Temporary Triple Store

B3. Query triple store to solve for unknown variables.

"glycolysis pathway has participant some gene"
kegg:dme00010 slo:SIO_000132 ?gene

C. Run user’s original query against populated triple store.

Populated Triple Store
For each clause in query, gather relevant triples.

"glycolysis pathway has participant some gene"

kegg:dme00010 sio:SIO_000132 ?gene
B1 Find services that produce relevant data for the current query clause

SADI Registry

registry query
"glycolysis pathway has participant some gene"
kegg:dme00010 sio:SIO_000132 ?gene

service URLs

Temporary database

"glycolysis pathway has participant some gene"
kegg:dme00010

B For each clause, retrieve relevant triplets
Invoke SADI services, add output RDF to temporary triple store
For each clause in query, gather relevant triples.

Temporary Triple Store

query

"glycolysis pathway has participant some gene" kegg:dm00010 slo:SIO_000132 ?gene solutions for ?gene

"glycolysis pathway has participant some gene" kegg:dm00010 slo:SIO_000132 ?gene

Query triple store to solve for unknown variables.

Run user's origin query against n
SHARE Query Resolution

A. Users submits query

B. For each clause in query, gather relevant triples.

B1. Find services that produce relevant data for the current query clause

B2. Invoke SADI services, add output RDF to temporary triple store

B3. Query triple store to solve for unknown variables.

"glycolysis pathway has participant some gene" kegg:dme00010 sio:SIO_000132 ?gene

Temporary Triple Store

C. Run user’s original query against populated triple store.

Populated Triple Store

QUERY RESULTS
"Find FlyBase genes that participate in glycolysis and overlap genomic region 5,919,623..6,344,662 on chromosome 3L\"
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX sio: <http://semanticscience.org/resource/>
PREFIX pathway: <http://lsrn.org/KEGG_PATHWAY:>

WHERE {
  pathway:dme00010 sio:SIO_000132 ?kegg_gene_record.  # SIO_000132 = 'has part'
  ?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 [
      rdf:type ?strand_type;
      sio:SIO_000093 feature:3L  # SIO_000093 = 'is proper'
    ]
    ]
  ];
  FILTER ((?endpos >= 5919623) && (?startpos <= 6344662))
  FILTER ((?strand_type = strand:MinusStrand) || (?strand_type = strand:PlusStrand))
}
Submit the query to SHARE.
Wait a while...
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:>

WHERE {
  # SIO_000132 = 'has participant'
  sio:SIO_000132 .
}
```

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

**Query results**

<table>
<thead>
<tr>
<th>flybase_gene_record</th>
<th>startpos</th>
<th>endpos</th>
<th>strand_type</th>
</tr>
</thead>
</table>
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](http://sadiframework.org/registry)
3. Download the SADI for GMOD tarball and unpack into \texttt{cgi-bin}.

4. Set DB connection parameters in \texttt{cgi-bin/sadi.gmod/sadi.gmod.conf}:

   \begin{verbatim}
   [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/
   \end{verbatim}

5. Configure Dbxref mappings in \texttt{cgi-bin/sadi.gmod/dbxref.conf}:

   \begin{verbatim}
   [DBXREF_TO_LSRN]
   SwissProt = UniProt
   UniProtKB = UniProt
   SwissProt/TrEMBL = UniProt
   ...
   \end{verbatim}

6. Register the services in public SADI registry: \texttt{http://sadiframework.org/registry}.
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/register](http://sadiframework.org/register)
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

http://sadiframework.org/