Ecotoxicology Data Federation with SADI Semantic Web Services

Alexandre Riazanov[1,6] and Matthew M. Hindle[4,5] and E. Scott Goudreau[2] and Christopher J. Martyniuk[2,3] and Christopher J. O. Baker[1,6]

[1]Department of Computer Science & Applied Statistics, [2]Canadian Rivers Institute and Department of Biology, University of New Brunswick, Saint John, NB, Canada,
[3]Canada Research Chair in Molecular Ecology,
[4]SynthSys, [5]School of Informatics, Edinburgh University, UK,
[6]IPSNP Computing Inc, Canada,
[1,2]{t969c,cmartyn,bakerc}@unb.ca, [4]matthew.hindle@ed.ac.uk,
[6]alexandre.riazanov@ipsnp.com

Abstract. Biologists and biotechnologists need to draw information from numerous distributed and heterogeneous resources, such as online biomedical databases, nomenclatures and specialised bioinformatics tools. These tasks can benefit significantly from *semantic data federation* with *SADI Semantic Web services* where multiple resources exposed through SADI services are accessed as a single virtual SPARQL-queriable database. We provide evidence in support of this premise by creating and testing a kit of public SADI services for a number of bioinformatics databases and programs, and by demonstrating how it can be used to serve real information needs of ecotoxicology researchers, by using the services to answer some model queries.

1 Introduction.

The future of semantic technologies depends on how quickly and broadly they are adopted, which in turn depends on what value these technologies deliver to end users. For semantic technology researchers and engineers, this necessitates checking their ideas and prototypes in real or realistic use scenarios driven by potential end users. This paper presents the results of such "fieldwork" testing the utility of a *data federation* approach based on Semantic Web services for systems biology: we explore the use of SADI [27] Web services for *semantic querying of heterogeneous and distributed biomedical data* for the needs of *ecotoxicology* research.
The use cases we adopt to challenge the technology are provided by a working biologist (the fourth author) and correspond to actual research questions being investigated in the field of *fish toxicology*. The goals of our study are twofold: first, we would like to demonstrate that SADI can be practically useful for real-life biological research, and secondly, our aim is that this paper will provide an

exemplar for SADI deployment in biological research settings that can be followed by other SADI adopters, in bioinformatics and other application areas where semantic data federation may be useful. We also explore what can be done to further improve the utility of SADI-based semantic querying.
We would like to emphasise that this paper is not an introduction to SADI. In particular, we neither discuss the technical details of how SADI services can be discovered and invoked, although a brief overview will be given in Section 1.2, nor compare SADI to other Semantic Web services and data federation frameworks. For this, we refer the readers to [27, 25, 24].

1.1 Self-service Semantic Data Federation Vision

Many activities in biomedical research and biotech industry require finding and *combining* information from multiple heterogeneous and distributed resources. The scope of data requirements for a biologist often includes many autonomous resources, such as online biomedical databases, nomenclatures, ontologies, literature and patent repositories, clinical databases and specialised analytical Web services, such as biomolecular sequence search and alignment. The state-of-the-art approaches to *data integration* – datawarehousing and workflow scripting (see, e. g., [15] and [16])– are both limited in scope and often unaffordable for academic research groups and small biotech companies.
We are advocating the emerging *data federation* paradigm where querying multiple heterogeneous distributed resources is as easy as querying a single database. The dynamic nature of research and biotech R&D activities implies that in many cases pre-programmed, e. g., form-based, querying is not enough and *ad hoc* querying is necessary. For *ad hoc* querying to be affordable in terms of labour it has to be *self-service*, so that non-programmer users, such as biologists or clinical research professionals, can formulate and execute queries without help from programmers. This *may be* possible if the querying is *semantic*, i. e., based on the use of shared formalised vocabularies and automatic application of knowledge written in the form of ontological axioms or logical rules. If querying is semantic, end users can formulate their queries *in the terminology of their domain*, without knowing how the underlying data is structured or specific mechanisms of access to the data. Our study contributes to the research dedicated to the realisation of this vision.

1.2 SADI Semantic Web services

The SADI (*Semantic Automatic Discovery and Integration*) framework [27] is a *set of conventions*. Simple HTTP-based Web services that follow these conventions can be *fully automatically discovered,*

composed and called by client programs. The two main principles of SADI are as follows.
First, SADI services can only consume *input in the RDF format* and can only produce *output in the RDF format*. This completely removes the problem of syntactic interoperability – any SADI service can directly consume data produced by any other SADI service.
Second, every SADI service provides a special *semantic description* that unambiguously defines what the service does, thus facilitating the findability of the service by client programs when they need the corresponding functionality. The description specifies what kind of RDF graphs the service expects in the input and can process, in terms of the concepts (properties and classes) that can be used in the input RDF and, more importantly, specifies the concepts the service can use to form the RDF graphs in the output.
The concepts used in descriptions of a set of SADI services, together with related concepts from the underlying ontologies, constitute a *federated schema* and can be navigated by users, including *non-technical* users that understand the corresponding domain terminology, to form meaningful queries over the network of available services. Such a query can be expressed, for example, in SPARQL and executed by a special query engine that will find SADI services providing relations that may be useful to satisfy the query, identify the data that can be sent as input to these services, and invoke the services to retrieve more data, and so on, until it has enough data to answer the query. Such engines can also *apply ontological axioms or rules* to facilitate simpler and more intuitive queries. Currently, there are two prototypes that implement this functionality: open-source SHARE [25, 24] and commercial Hydra [2].
A typical scenario for publishing a resource as a set of SADI services is as follows: suppose, a user would like to access the Web site `ClinicalTrials.gov` – a registry of clinical trials – via SADI services, and wants to retrieve trials by disease names and extract information about trials, such as the names of the drugs studied. The publisher can do this with two services: `getTrialsByDiseaseName` and `getDrugNameByClinicalTrial`.
The first step of the resource publishing process is data modelling: it is necessary to decide how the data will be represented in RDF and, in particular, what relations and entities can be used. For illustration purposes, we assume that we use classes `Name`, `Disease`, `ClinicalTrial` and `Drug`, and properties `diseaseIsTargetedByTrial`, `trialStudiesDrug`, `hasName`. In our data model, a disease name can be linked to a trial, and a trial can be linked to the corresponding drug name as in the following RDF (in Notation3 syntax):

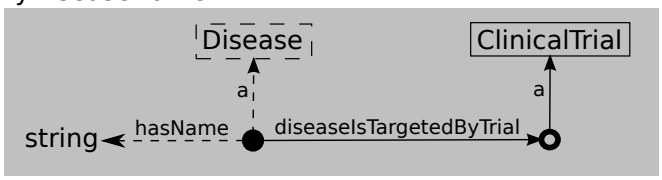
```
:disease_name a :Name;  :hasValue "Typhoid fever" .
:disease a :Disease;
     :hasName :disease_name;
```

```
      :diseaseIsTargetedByTrial :trial .
:trial a :Trial;  :trialStudiesDrug :drug .
:drug a :Drug;  :hasName :drug_name .
:drug_name a :Name; :hasValue "Ceftriaxone" .
```

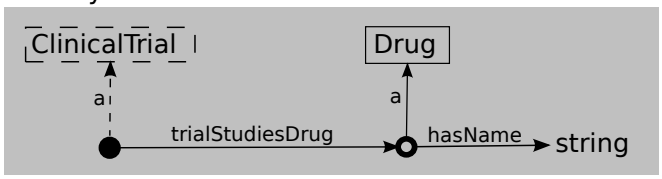
It is generally preferable to reuse some existing ontologies defining the required concepts, but when no such ontology is available, one can introduce the concepts in a small service-specific ontology and ensure that they have mnemonic names and/or descriptive textual labels, so that their meaning is apparent to users.

The next step is to define the input and output of the services in terms of the chosen entities and relations. The service `getTrialsByDiseaseName` will accept `Disease` objects with `hasName` attached to them, and in the output it will link the diseases to `ClinicalTrial` objects identified with their URLs in the `ClinicalTrials.gov` database via the property `diseaseIsTargetedByTrial`. The service `getDrugNameByClinicalTrial` will accept these `ClinicalTrial` URLs as input and link them via `trialStudiesDrug` to `Drug` objects linked to the corresponding names via `hasName`. In the schematic representation of the service descriptions below, the dashed and solid lines correspond to input and output specifications respectively.

getTrialsByDiseaseName



getDrugNameByClinicalTrial



The SADI framework uses OWL syntax to express the semantic descriptions. For example, the description of `getTrialsByDiseaseName` consists of the following two class definitions (in Protégé syntax): `(Disease` **and** `(hasName` **some** `string))` as the service input class and `(diseaseIsTargetedByTrial` **some** `ClinicalTrial)` as the output class.

1.3 Data integration requirements in Ecotoxicology

The discipline of ecotoxicology increasingly focuses on complex interactions in biological systems which, on the technical side, often requires an integrated analysis – *a systems view* – of Omics data of different types, such as proteomics and genomics data (see, e. g., [18]). This requires multiple software tools and databases supporting tasks such as *microarray analysis* (results of experimentally measuring certain values, such as gene expression intensity) or *gene annotation* (finding information about genes and corresponding proteins). As a result, biologists are faced with a bewildering array of disconnected bioinformatics resources. Drawing together and mastering these tools and resources is frequently a frustrating technical exercise in identifying common links across database records and connecting input and output formats of bioinformatics tools. Interpreting experimental Omics data in the context of the current available knowledge and methodologies from a *single query platform with explicit semantics* would be a valuable asset to ecotoxicologists. Emergence of such a framework would free toxicologists from needlessly spending time on technical and semantic idiosyncrasies, and enable the researchers to synthesize Omics information to better predict risks associated with chemical exposures.

1.4 Study outline

There are several approaches to integrating biological data sets, and some of them are based on Semantic Web. For example, projects like Bio2RDF [8] and Linked Life Data [3] have used semantic technologies to build mash-ups of current biological information. However, many biological application cases also require the integration of bioinformatics *algorithms*. Architectures where Semantic Web services are used as components in complex bioinformatics analysis pipelines, are an elegant solution to exposing knowledge, data and algorithms in a semantically explicit framework.

In the study presented here, we consider a number of research questions from the field of *toxicology* that require data and algorithm integration. To facilitate the integration of multiple resources which are required for obtaining insights on these research questions, we have created a kit of SADI services exposing these resources. In this paper we describe how the relevant data is modelled, how the corresponding SADI services work and how they can be used via SADI query engines to implement *semantic federated querying* of the resources. We provide three example SPARQL queries that demonstrate the potential utility of SADI-based semantic data federation for biological research.

2 Target information needs in fish toxicology and corresponding resources.

Fish toxicology is a sub-field of ecotoxicology that studies the responses of fish to exposure to various pollutants, such as fertilisers and pesticides. Ultimately, such research is meant to facilitate the preservation of fish populations, develop more efficient aquaculture methods and also help to discover general biological mechanisms that may transfer to other species, especially humans.
Typically, fish toxicologists want to know what sequences of molecular events are triggered in fish after they are exposed to chemicals of a certain type, and what organism functions and biological processes are affected. More specific examples of research questions might include *"For all fish microarray studies, what complexes of biochemical reactions are commonly affected?"*, *"Does the exposure to chemical X affect the production of proteins with common structural properties?"* or *"What types of human diseases are known to be related to the genes affected by pesticide Y in fish brain?"*. Answering questions of this kind often depends on the ability of researchers to analyse experiment results in the context of available biomedical knowledge. Here we describe the main types of databases and algorithmic resources that can be used in combination to get insights on fish toxicology-related questions.
Microarray experiment repositories. The toxicity of many chemicals is manifested by affecting the *expression* of genes: some genes are *up-regulated* (produce more RNA than usual), others are *down-regulated* (produce less RNA). Consequently, an important technique for learning about the effects of a chemical on an organism is to measure the expression intensity for various genes by conducting a *DNA microarray experiment*. Experiments of this kind are also used for other tasks, such as measuring protein quantities in biological samples. Large numbers of results of microarray experiments are deposited in public online databases, such as ArrayExpress [12], in standartised flat-file or XML-based formats.
Sequence processing tools. Omics experiments often deal with molecular sequences of different types, such as DNA and proteins, so analysis of experimental data requires computation on sequences. Two heavily used types of algorithms – BLAST [7] and HMMER3 [11] – allow searching for *similar* sequences in large sequence databases, such as NCBI RefSeq [20] and Pfam [14], and aligning sequences. DNA sequences in microarrays, especially in fish microarrays, are often incomplete, contain missing gene fragments and start at nucleotides that do not correspond to fragments that translate into amino acid sequences. The OrfPredictor tool [19] helps to cope with this problem by identifying Open Reading Frames (ORF) – parts of DNA sequences that actually encode genes.

Gene Ontology (GO) and model organism databases. Meaningful interpretation of gene expression experiments often requires mapping genes to GO annotations specifying molecular functions of the corresponding proteins or biological processes they are involved in. Experimentally derived annotations of this kind are available from a number of genomic databases for well-studied (*"model"*) organisms, such as human protein annotations at EBI [1].

Miscellaneous resources. Various other biomedical resources may be necessary for fish toxicology data analyses. For example, if a biologist needs to analyse data for a genus or a whole class, rather than a species, he will have to use a taxonomy, such as the NCBI Taxonomy [4], to enumerate all species names. Another example of a popular resource is the UniProt database [6] containing information about proteins, such as references to biochemical reactions they participate in.

3 Data modelling.

This section describes how we model the data discussed in the previous section. Choosing appropriate ontological primitives and RDF modelling patterns is crucial for *convenient and flexible querying* as well as for the semantic *interoperability* of services.

Ontologies. In order to improve the *re-usability* of our SADI services, wherever possible we reference existing upper and domain ontologies. Table 1 lists the ontologies used by the SADI Web services written for our experiment: The Semanticscience Integrated

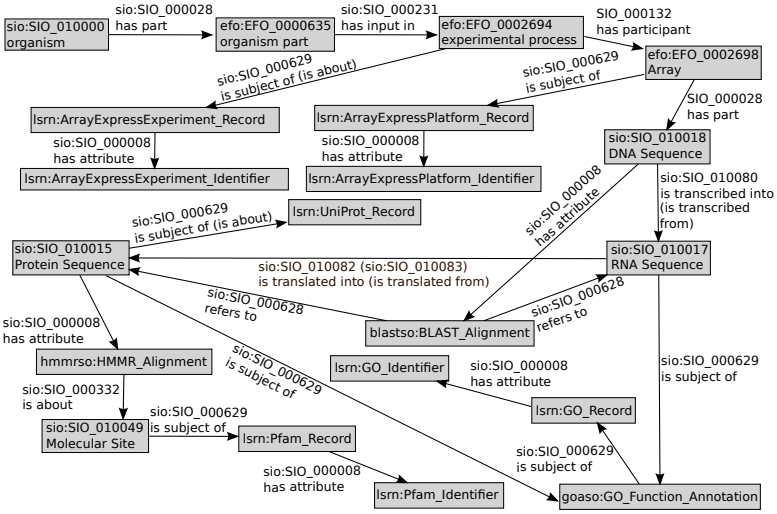
Prefix	URL
sio	http://semanticscience.org/resource/
lsrn	http://purl.ovclc.org/SADI/LSRN/
efo	http://www.ebi.ac.uk/efo/
ncbi	http://purl.org/obo/owl/NCBITaxon#
blastso	http://cbakerlab.unbsj.ca:8080/BLAST-sadi-service-ontology.owl#
hmmrso	http://cbakerlab.unbsj.ca:8080/fishtox/HMMR-sadi-service-ontology.owl#
goaso	http://unbsj.biordf.net/fishtox/GOA-sadi-service-ontology.owl#
maso	http://cbakerlab.unbsj.ca:8080/fishtox/arrayexpress-sadi-service-ontology.owl#
tsso	http://cbakerlab.unbsj.ca:8080/fishtox/record-translation-sadi-service-ontology.owl#
stso	http://cbakerlab.unbsj.ca:8080/fishtox/seq-tools-sadi-service-ontology.owl#

Table 1. Main ontologies used in our study

Ontology (SIO) is an *upper ontology* providing a broad set of general classes and properties, and is used extensively by many SADI

services. The Life Science Resource Name (LSRN) ontology provides classes for records and identifiers from standard databases and nomenclatures, such as `lsrn:UniProt_Record` and `lsrn:GO_Identifier`. The Experimental Factor Ontology (EFO) [13] provides classes and properties for describing gene expression experiments and is intended to support querying over experimental data and data integration. We use EFO, in particular, to leverage interoperability with the Gene Expression Atlas [12] where it is used extensively. We also use an OWL version of the NCBI taxonomy to identify species, genera, etc. Our service-specific ontologies (with the prefixes ending with "so") mainly contain input and output class definitions for our SADI services.

Modelling patterns. The following figure shows a fragment of the federated schema for our SADI services, which can be used to design SPARQL queries. The predicates in brackets are the inverse of the properties represented by the connecting arrows.



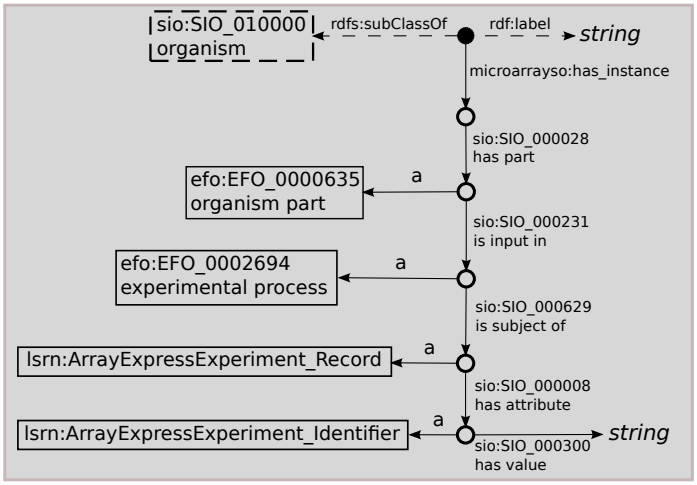
Our modelling facilitates many different Omics-related queries and is also future-proof because many SADI services, based on different but related data sources, can use the same or compatible modelling.

4 SADI services.

We have created almost 60 SADI services specifically for our fish toxicology use cases, that expose resources mentioned in Section 2. Where possible, we wrapped existing Web services implemented by database and tool providers, as SADI services to access live data. This ensures results are current and helps to avoid the maintenance

cost associated with data mirrors. Descriptions of the services are provided at `http://cbakerlab.unbsj.ca:8080/FISHTOX-SADIServices`. We only briefly describe some of the services here.

ArrayExpress-based services. Several services which find experiments deposited in ArrayExpress and retrieve information about them, were created by wrapping RESTful services provided by ArrayExpress. The data was modelled using a combination of EFO (natively supported by the database) and SIO properties. The following figure shows the IO modelling for the service `SpeciesName2AEExperimentalRecord` that searches for ArrayExpress records by a species name:



The service implementation simply extracts the species name string from the input RDF, makes a request to the ArrayExpress search service and converts the resultant ArrayExpress record IDs to RDF.

Sequence search and alignment. The HMMER Web site provides a RESTful service for sequence search and alignment, so we created a SADI service that, given an amino acid sequence, finds Pfam records for *protein domains* similar to parts of the input sequence. The input class of the service is a `'protein sequence'` and the output class is defined in the `hmmrso` ontology as a class that `'has attribute' some (lsrn:HMMR_Alignment that ('is about' some ('molecular site' that ('is subject of' some lsrn:Pfam_Record))))` (here and throughout the rest of the paper single-quoted labels represent the corresponding identifiers from SIO). Similarly, SADI services for BLAST were created by wrapping NCBI Web services. Their inputs can be protein, DNA or RNA sequence strings, depending on the variant of BLAST. The output is defined using a special alignment class, in an approach similar to

HMMER3 services. For example, the output for the BLAST on proteins is defined in the `blastso` ontology as `'has attribute' some (blastso:BLAST_Alignment that ('refers to' some ('protein sequence' that ('is subject of' some lsrn:NCBI_NP_Record))))`.

Database ID mapping. Answering many queries requires cross-mapping of information from different databases, in one form or another. For example, we may know a RefSeq ID of a protein sequence, but retrieving Gene Ontology annotation requires UniProt IDs of the protein. The RefSeq-to-UniProt mapping is already available from RefSeq and, in general, biomedical databases often reference other databases, so we only need an appropriate modelling in order to implement the DB record mapping as a SADI service. Our `RefSeqNP2UniProt` service takes an instance of `(lsrn:NCBI_NP_Record and ('has attribute' some (lsrn:NCBI_NP_Identifier and ('has value' some string))))` as input and annotates it as an instance of the following output class: `('is about' some ('protein sequence' and ('is subject of' some lsrn:UniProt_Record)))`.

Gene Ontology annotation. We created several SADI services annotating proteins with GO identifiers of known molecular functions they have and biological processes they participate in. The data for several model organisms, such as *humans*, *zebrafish* and *house mouse*, originating from the corresponding model organism databases, are taken from the GO website [5] that contains up-to-date annotations. The services expect URIs of model organism database records as input and return outputs conforming to the following class: `'is about' some ('deoxyribonucleic acid sequence' and ('is transcribed into' some ('ribonucleic acid sequence' and ('is translated into' some ('protein sequence' and ('is subject of' some (goaso:GO_Function_Annotation or goaso:GO_Process_Annotation)))))))`, where the annotation classes represent resources that are directly linked to some GO records.

5 SPARQL queries.

In this section we present three example queries which address the types of questions pertinent to the analysis of fish toxicology experiments with DNA microarrays. However, the SADI services we have built, and the modelling we use, are not limited to these examples. Different combinations of our SADI services, together with multiple public SADI services developed for other purposes, can be used to answer other queries in contexts other than fish toxicology – the methodology we are using here is widely applicable to gene expression analysis in general.

We ran the queries with two SADI query engines – SHARE [25] and Hydra [2], which compute SPARQL queries by picking and calling

suitable SADI services from a dedicated registry of fish toxicology-related services. The engines are currently only proof-of-concept prototypes, so we did not pay attention to the performance. At this stage, we are primarily concerned with demonstrating the *principle possibility of using such tools for data federation*, which will justify further efforts on improving the tool performance and the framework itself – such work requires highly specialised skills and is very costly, so extensive evidence of the utility of SADI is necessary to obtain adequate government sponsorship or private investment. We use *two* query engine prototypes to be able to execute as many queries as possible in practically acceptable time.

5.1 Query I: Finding relevant microarray experiments.

Experimentalists in fish toxicology need to compare their work with previous published experiments, which requires locating microarray experiments with related parameters. For example, a biologist may be interested in identifying genes whose expression has been measured in the tissue *hypothalamus* of the species *largemouth bass* in existing experiments. Currently, the main option is to use Web tools provided by experiment databases like ArrayExpress, which requires multiple searches and manual inspections of many experiments each of which may use different microarray platforms. SADI allows it to be done in a much easier way. A declarative SPARQL query whose simplified pseudo-SPARQL version is given below, expresses the question formally:

```
SELECT ?experiment_id ?tissue_name ?platform_id ?gene_id
FROM <http://cbakerlab.unbsj.ca:8080/fishtox/large-mouth-bass.owl>
WHERE {
 ?org_class aeso:has_instance ?org_instance .
 ?org_instance a ncbi:NCBITaxon_27706 . # largemouth bass .
 ?org_instance 'has part' ?org_part .
 ?org_part 'is input in' ?exp_process .
 ?exp_process 'is subject of' ?exp_record .
 ?exp_record 'has attribute' ['has value' ?experiment_id] .
 ?exp_record 'is about' ?exp_process .
 ?exp_process 'has participant' ?array .
 ?array 'is subject of' ?array_platform_record .
 ?array_platform_record a lsrn:ArrayExpressPlatform_Record .
 ?array_platform_record 'has attribute'
                              ['has value' ?platform_id] .
 ?array_platform_record 'is about' ?array .
 ?array 'has part' [rdfs:label ?gene_id] .
 ?org_part rdfs:label ?tissue_name .
 FILTER regex(?tissue_name,"hypothalamus","i") .
}
```

Note that the RDF file `large-mouth-bass.owl` specified in the FROM clause contains the URI of the species to instantiate `?org_class`. The query was submitted to SHARE which resolved it by finding and calling the SADI service `SpeciesName2AEExperimentalRecord` to identify relevant ArrayExpress experiment records, the service `AEExperiment2Platform` to retrieve the corresponding microarray type (*"platform"*) records and the service `AERecord2Microarray` to extract microarray details, such as the DNA sequences. No understanding of the ArrayExpress semantic idiosyncrasies or data syntax was required to formulate the query. SHARE identified two microarray experiments satisfying the query.
Despite the fact that the SADI registry used in our experiments only contained services necessary for our experiments presented here, the query was quite hard for SHARE: it required an overnight run on a commodity quad-core server running both SHARE and the services, and practically all the time was spent inside the query engine itself. Hydra produced first answers in less than 4 minutes. When it was terminated after 1 hour, it had produced over 12,000 answers and was generating more answers. Less than 1/3 of the time was spent on behalf of Hydra itself, which is a good progress relative to SHARE. Moreover, a majority of the executed service calls were redundant, due to the current lack of corresponding optimisations in Hydra, which allows to estimate that Hydra's performance will be several times better when such optimisations are introduced.

5.2 Query II: Finding functional information about genes.

Gene Ontology annotation is very valuable for understanding toxicity of chemicals as it tells the toxicologist what (parts of) biochemical reactions are disrupted when a chemical affects the expression of a particular gene. However, such functional annotation is not directly available for many non-model species, such as *largemouth bass*, so biologists have to infer GO annotations based on sequence similarity with known genes in model organisms for which experimental evidence is recorded in public repositories.
The following query annotates ten most significantly affected genes in *largemouth bass* treated with the pesticide *dieldrin* [17]:

```
SELECT ?GO_record
FROM <http://cbakerlab.unbsj.ca:8080/fishtox/10genes.rdf>
WHERE {
 ?DNA_chip_sequence a 'deoxyribonucleic acid sequence'. # DNA
 ?DNA_chip_sequence 'has attribute' ?alignment .
 ?alignment 'refers to' ?sequence_hit .
 ?sequence_hit 'is subject of' ?refseq_record .
 ?refseq_record 'is about' ?protein_sequence .
```

```
 ?protein_sequence 'is subject of' ?UniProt_record .
 ?UniProt_record a lsrn:UniProt_Record .
 ?UniProt_record 'is about' ?UniProt_protein_sequence .
 ?UniProt_protein_sequence 'is subject of' ?GO_annotation .
 ?GO_annotation a goa:GO_Function_Annotation .
 ?GO_annotation 'is subject of' ?GO_record .
 ?GO_record a lsrn:GO_Record .
}
```

The gene sequences are provided in the file `10genes.rdf`. We ran this query with Hydra and its earlier version with SHARE. To execute the query, the Hydra engine calls the BLAST service `BLASTx2RefSeqProtein` to find proteins similar to the ones corresponding to the input DNA in the NCBI RefSeq database, retrieves the protein IDs for them with `RefSeqNP2UniProt`, and then retrieves GO annotations for these proteins from a number of GO datasets for different model organisms with services like `HumanEBIUniProtRecord2GO`. The results obtained by Hydra indicate that the exposure to dieldrin might affect ribosomal processes. Retrospectively this is not surprising, but it shows that our methodology may be useful in future experiment analyses.
This query provides an interesting observation. The line `?refseq_record 'is about' ?protein_sequence` seems to duplicate `?sequence_hit 'is subject of' ?refseq_record` because `'is about'` is inverse to `'is subject of'` and the user could also postulate that the property `'is about'` is *functional* for RefSeq records. Unfortunately, we cannot replace the query with a more natural version by merging these two lines because neither SHARE nor Hydra currently support this level of expressivity.
In addition to tabular results, the RDF returned from the query captures detailed information regarding how candidate functions relate to the query inputs. For example: the quality scores for the alignments of similar proteins and the experimental evidence for the assigned functions of candidate orthologs.
This query also required an overnight run with SHARE. Hydra was able to execute it in less than 1.5 hour on the same hardware.

5.3 Query III: Identifying protein domains.

Another way to informatively characterise an affected gene – in addition to GO annotations – is to identify *domains*, i. e., subsequences with known biological functions, the corresponding protein contains. Often, this cannot be done for microarray sequences directly, because they are incomplete, and the use of ORF prediction tools is necessary to identify plausible protein sequences. Then, a sequence search and alignment procedure, such as HMMER, can be used to retrieve the domains. The following query implements

this combination of tools and retrieves Pfam names for domains in proteins produced by the same ten *largemouth bass* genes as in the previous query:

```
SELECT ?protein_domain_name
FROM <http://cbakerlab.unbsj.ca:8080/fishtox/10genes.rdf>
WHERE {
 ?DNA_chip_sequence 'is transcribed into' ?RNA_sequence .
 ?RNA_sequence 'is translated into' ?protein_sequence .
 ?protein_sequence 'has attribute' ?alignment .
 ?alignment 'is about' ?molecular_site .
 ?molecular_site 'is subject of' ?pfam_record .
 ?pfam_record rdfs:label ?protein_domain_name
}
```

Three services are called by SHARE: the service `DNA2RNATranscriber` transcribing a DNA into an RNA, the service `ORFPredictor` predicting the ORF and translating the RNA to a protein sequence, and the service `HMMR3PFamA` retrieving Pfam domain IDs for subsequences of the protein. In several minutes the query engine returned the answer that a domain related to ribosomal activities was found on the gene `UF_Msa_AF_100231`, which accords with the results for Query II. The low coverage on genes (1/10) is not surprising given that our HMMER3 SADI service was configured to retrieve only strongly similar sequences. The service could be given more liberal parameters, but the service configuration functionality is not yet implemented in SHARE or Hydra.

6 Discussion.

We have demonstrated that semantic data federation with SPARQL querying of SADI services representing multiple bioinformatics databases and analytical tools can help to answer relevant research questions in fish toxicology. The SADI service kit we have created exposes the databases and programs in a semantically explicit way and enables diverse and powerful queries using query engines like SHARE or Hydra. When using this framework, an ecotoxicologist would not have to understand the semantics and technicalities of the underlying resources in order to formulate queries across databases and tools. He would not have to program any additional scripts of workflows as the query execution happens completely automatically.

Novel and emergent tools and algorithms that can be utilized for query resolution, such as new sequence alignment methods or reading frame prediction, can be incorporated into query resolution by implementing a SADI web service wrapper that uses the relevant

service ontologies. The new services are registered to the SADI repository and these services will be incorporated without changing the query.

We conclude that altogether the approach promises to be practically relevant at least for Omics-based analyses of ecotoxicology data. We would like to emphasise that the queries we consider in this paper can be used in contexts other than ecotoxicology: e.g., the ability to search for relevant microarray experiments or Gene Ontology annotation of DNA sequences may be useful in many contexts where DNA microarray experiments are used.

Currently, query composition required a semantic-web domain expert to compose SPARQL queries in close consultation with a biologist. Query composition required careful consideration of the semantic modelling, the available pool of services, and the performance limitations of the query-resolution engine. In many instances additional services were required to resolve new queries, so the design and testing of queries was an iterative process. We expect that query composition should become a faster and less painful process as the ecosystem of SADI tools and services matures. Client performance limitations are already being addressed with the development of the Hydra engine.

Regarding what could be improved in SADI and SADI query clients, our current observations support most conclusions of [21], namely (i) better user interfaces are needed to relieve bioinformaticians from the necessity of writing SPARQL and for easier exploration of semantic data schemas, (ii) query answers should carry verifiable provenance information so that they can be used beyond the discovery phases of research, and (iii) the ontology-based data modelling part of the SADI service development process could be better supported, although this will be less of a problem when a critical mass of reusable public SADI services is created. One item we would like to add to this wish list is better logical expressivity of queries, as illustrated in Section 5.2.

We would like to discuss (i) in more detail here. In principle, non-programmer users can be trained to write SPARQL queries: for example, the third author, who is an undergraduate Physics student, actively participated in composing and debugging the queries. A query is typically composed incrementally: the user write a few lines initially, runs a query engine, assesses the results, adds a few more lines, an so on. This process will be significantly simplified when tools for extraction of semantic schemas from service descriptions and ontologies are implemented: such tools facilitate easy look-up of classes and properties that are meaningful in a particular query context.

However, even simplified by schema navigation, SPARQL querying is not sufficient to fully realise the vision of self-service querying

because it is difficult to expect that it can appeal to the mass user. More friendly graphical or keyword-based query interfaces, as in [23], have to be used on top of SPARQL. Intutitive graphical representation of queries would relieve the users from the need to learn the SPARQL notation. Facilities for mapping keywords to query fragments would help users to focus the class and predicate look-up better. Finally, the incremental query composition driven by concrete (instance-level) data rather than just a schema, can be supported by a form of *faceted browsing* (see, e. g., [26]).

7 Related and future work.

The SADI framework has been compared to related approaches in a number of publications – see, e.g., [27, 21], so we omit this discussion here and focus on the work on SADI applications.

In an early bioinformatics case study [21], SADI was used as a medium for deploying *text mining* software that extracts mentions of mutations and their impacts on protein properties from biomedical texts. Similarly to what we do in this paper, the utility of SADI, especially its integrative power, was demonstrated in a number of biologically meaningful scenarios through a SPARQL interface. The data obtained by text mining was integrated with multiple sources of data on genes, proteins, biochemical reactions and drugs, as well as some molecular structure visualisation programs.

In [10, 9] Chepelev *et al* conduct two *cheminformatics* case studies. The first paper describes a SADI-based prototype for integrating components of a lipid classification pipeline – a molecular substructure detection program and an ontology-based molecule classifier – with each other and with external biomedical data. The second paper describes a package of SADI services based on a Java library for cheminformatics and discusses, as a use case, detection of drug-like chemicals by ontology-based querying of the SADI services.

The study presented here differs from the ones mentioned above by exploring *real* use cases, albeit simplified, corresponding to real research questions, as opposed to *realistic* ones used in the earlier projects, and the involvement of a working biologist as a target end user. Another difference is the focus on the *analysis of experiment results*, as a type of research activity where semantic data federation may have a particularly strong impact.

We would also like to mention a case study [22] for SADI that explores the possibility of using it for *clinical intelligence* purposes, more specifically for surveillance of hospital-acquired infections, although it primarily focuses on using SADI as a vehicle for semantic querying of relational databases rather than for data integration.

A natural continuation of the work presented in this paper is to try to answer some open fish toxicology questions. To this end, we

are currently extending the repertoire of services and queries targeting a number of questions about the effects of pesticides and steroids on fish, that can be answered by federated querying of public biomedical resources. As another future work direction, we will try to reproduce a subset of microarray and toxicology workflows from the myExperiment repository [16] with SADI services and SPARQL queries.

References

1. Gene Ontology Annotation (UniProt-GOA) Database @ EBI. http://www.ebi.ac.uk/GOA/.
2. Hydra: A scalable SPARQL engine for SADI. http://ipsnp.wikidot.com/hydra.
3. Linked Life Data. http://linkedlifedata.com/.
4. NCBI Taxonomy. http://www.ncbi.nlm.nih.gov/taxonomy.
5. The Gene Ontology, Current Annotations. http://www.geneontology.org/GO.downloads.annotations.shtml.
6. UniProt database. http://www.uniprot.org/.
7. S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic Local Alignment Search Tool. *J.Mol.Biol*, 215(3), 1990.
8. F. Belleau, M.-A. Nolin, N. Tourigny, P. Rigault, and J. Morissette. Bio2RDF: Towards a mashup to build bioinformatics knowledge systems. *J. of Biomed. Informatics*, 41(5), 2008.
9. L. L. Chepelev and M. Dumontier. Semantic Web integration of Cheminformatics resources with the SADI framework. *J Cheminform. 2011*, 3, 2011.
10. L. L. Chepelev, A. Riazanov, A. Kouznetsov, H. S. Low, M. Dumontier, and C. J. O. Baker. Prototype Semantic Infrastructure for Automated Small Molecule Classification and Annotation in Lipidomics. *BMC Bioinformatics 2011*, 12(1), 2011.
11. S.R. Eddy. A new generation of homology search tools based on probabilistic inference. *Genome Inf*, 23, 2009.
12. H. Parkinson et al. ArrayExpress update – from an archive of functional genomics experiments to the atlas of gene expression. *Nucl. Acids Res.*, 37(S1), 2008.
13. J. Malone et al. Modeling sample variables with an Experimental Factor Ontology. *Bioinformatics*, 26(8), 2010.
14. R. D. Finn et al. The Pfam protein families database. *Nucl. Acids Res.*, 36(D), 2008.
15. T. J. Lee et all. BioWarehouse: a bioinformatics database warehouse toolkit. *BMC Bioinformatics*, 7, 2006.
16. C.A. Goble, J. Bhagat, S. Aleksejevs, D. Cruickshank, D. Michaelides, D. Newman, M. Borkum, S. Bechhofer, M. Roos, P. Li, and D. De Roure. myExperiment: a repository and social network for the sharing of bioinformatics workflows. *Nucl. Acids Res.*, 38(Suppl 2), 2010.

17. C. J. Martyniuk, K. J. Kroll, N. J. Doperalski, D. S. Barber, and N. D. Denslow. Genomic and Proteomic Responses to Environmentally Relevant Exposures to Dieldrin: Indicators of Neurodegeneration? *Toxicological Sciences*, 117(1), 2010.
18. C.J. Martyniuk, R.J. Griffitt, and N.D. Denslow. Omics in aquatic toxicology: not just another microarray. *Environ Toxicol Chem*, 30(2), 2011.
19. X. J. Min, G. Butler, R. Storms, and A. Tsang. OrfPredictor: predicting protein-coding regions in EST-derived sequences. *Nucl. Acids Res.*, 33(S2), 2005.
20. K. D. Pruitt, T. Tatusova, and D. R. Maglott. NCBI Reference Sequence (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucl. Acids Res.*, 33(D), 2005.
21. A. Riazanov, J.B. Laurilla, and C. J. O. Baker. Deploying Mutation Impact Text-Mining Software with the SADI Semantic Web Services Framework. *BMC Bioinformatics 2011*, 12 (Suppl 4):S6.
22. A. Riazanov, G. W. Rose, A. Klein, A. J. Forster, C. J. O. Baker, A. Shaban-Nejad, and D. L. Buckeridge. Towards Clinical Intelligence with SADI Semantic Web Services: a Case Study with Hospital-Acquired Infections Data. In *SWAT4LS '11*. ACM, 2012.
23. T. Tran, P. Cimiano, S. Rudolph, and R. Studer. Ontology-Based Interpretation of Keywords for Semantic Search. In *The Semantic Web*, volume 4825 of *LNCS*. 2007.
24. B. Vandervalk. The SHARE System: A Semantic Web Based Approach for Evaluating Queries Across Distributed Bioinformatics Databases and Software, MSc thesis, UBC, 2011.
25. B. P. Vandervalk, E. L. McCarthy, and M.D. Wilkinson. SHARE: A Semantic Web Query Engine for Bioinformatics. In *ASWC 2009*.
26. A. Wagner, G. Ladwig, and T. Tran. Browsing-oriented Semantic Faceted Search. In *Database and Expert Systems Applications*, 2011.
27. M. D. Wilkinson, B. Vandervalk, and L. McCarthy. The Semantic Automated Discovery and Integration (SADI) Web service Design-Pattern, API and Reference Implementation. *JBMS*, 2(8), 2012.