SpecINT: A framework for data integration over cheminformatics and bioinformatics RDF repositories

Branko Arsić [a,*], Marija Ðokić-Petrović [a,b], Petar Spalević [c], Ivan Milentijević [d], Dejan Rančić [d], Marko Živanović [a]

[a] *Faculty of Science, University of Kragujevac, Serbia*
E-mails: brankoarsic@kg.ac.rs, marija.djokic@virtualworldservices.at, zivanovicm@kg.ac.rs
[b] *Virtual World Services GmbH, Austria*
E-mail: marija.djokic@virtualworldservices.at
[c] *Faculty of Technical Sciences, University of Priština, Serbia*
E-mail: petar.spalevic@pr.ac.rs
[d] *Faculty of Electronic Engineering, University of Niš, Serbia*
E-mails: ivan.milentijevic@elfak.ni.ac.rs, dejan.rancic@elfak.ni.ac.rs

Editor: Michel Dumontier, Maastricht University, The Netherlands
Solicited reviews: Alasdair J G Gray, Heriot-Watt University, UK; Two anonymous reviewers

Abstract. Many research centers and medical institutions have been accumulating a vast amount of various biological and chemical data over the past decade and this trend continues. Based on Linked Data vision, many semantic applications for distributed access to these heterogeneous RDF (Resource Description Framework) data sources have been developed. Their improvements have brought about a decrease of intermediate results and optimizing query execution plans. But still many requests are unsuccessful and they time out without producing any answer. Also, the applications which operate over repositories taking into consideration their specificities and inter-connections are not available. In this paper, the SpecINT is proposed as a comprehensive hybrid framework for data integration and federation in semantic data query processing over repositories. The SpecINT framework represents a trade-off solution between automatic and user-guided approaches, since it can create queries which return relevant results, while not being dependent on human work. The innovativeness of the approach lays in the fact that the coordinates of graph eigenvectors are used for the automatic sub-queries joining over the most relevant data sources within repositories. In this way searching can be effected without a common ontology between resources. In experiments, we demonstrate the potential of our framework on a set of heterogeneous and distributed cheminformatics and bioinformatics data sources.

Keywords: Federated SPARQL query, Data Integration, Matrix Eigenvectors

1. Introduction

New data about chemical compounds, the influence they have on cancer cell-lines, genes and proteins, genetic variations and cell pathways have been emerging at a staggeringly rapid pace in recent chemical and biological experiments. Research centers and laboratories work independently storing data in different data formats with different vocabularies. The very abundance of heterogenic data sources prevents the life science community reaching its maximum. In this information vortex scientists need to put effort into finding and pairing relevant information over heterogeneous data within different data sources and consoli-

[*] Corresponding author. E-mail: brankoarsic@kg.ac.rs.

dating repositories. For the successful performance of biomedical research, data integration grows into an important precondition for overcoming the existing gaps in resources and for introducing time savings. In [1] the authors indicated the importance of data integration in cheminformatics and bioinformatics.

For efficient query processing in semantic-oriented environments, sophisticated query generators and benchmarking systems for their performance evaluation have been developed. Drawbacks of benchmarking systems arise from the fact that they rely on a set of predefined static queries over particular data sources [2][3][4]. However, the automatic query generators are still faced with many problems. Firstly, the process of setting parameters for an algorithm and thresholds can be difficult without prior knowledge of the data. This leads to the collecting of different statistics which are changeable over time. Secondly, in such piles of generated queries, many are without answer, and many of them return unnecessary data. At the same time, the processes of seeking the most promising queries, their execution and evaluation are time-consuming. Even then, in most cases the results are not satisfactory for the research community which expects correct results in real-time. Thirdly, these approaches cannot explore more repositories with many data sources following their specific integration and connections. Very often repositories integration is not possible because there are no mapping schemes between them. Additional aggravating circumstances are the completely different structure and connections between data sources within different repositories.

Automatic query generation is less tedious and can produce many queries which are used only for query execution evaluation, not for end-users and their demands. Meaningful and real queries can only be generated manually (user-guided) or semi-automatically requiring a lot of effort since the content of the data sources needs to be analyzed in advance. However, the problem of integrating data from multiple data sources and repositories is still a challenge. Hand-crafted queries require a lot of effort and knowledge about data sources, whilst automatic query generation can produce many queries which should be manually tested and chosen for further distribution.

Our solution is based on a hybrid technique involving a human role in creating hand-crafted sub-queries (patterns, templates) as a very important guide for satisfactory results. The paticular query patterns are connected into queries automatically, seeking the most relevant data sources (from different repositories) which belong to and which potentially consist of the triples of interest. For the solution to this task we used the eigenvectors of a graph which enable us to follow the paths (edges) between these data sources. These edges suggest an aggregation of the most relevant data and that is why only the connected data sources are considered. All this is performed over different repositories and on-the-fly. The project contributors found that these paths lead to the best decision-making, rather than exploring every single triple in the repositories. In contrast to the state-of-the-art Federated SPARQL query engines which are dependent on the common ontology and triple statistics, our solution connects data sources within repositories by using the graph eigenvectors [5] and vertices ranking [6][7], without a common ontology between resources.

The SpecINT[1] is a support framework developed as an idea to potentially reinforce research activities in the Centre for Preclinical Testing of Active Substances (CPCTAS)[2] meeting their need to monitor results on a global scale. The contributions of the paper are:

- *Advancement:* A SPARQL query framework based on the concept of a mathematical graph is developed - the graph eigenvectors are used for the relevant data sources selection and their patterns joining.
- *Scalability:* A straightforward model for linking data from repositories on-the-fly is proposed.
- *Federation:* Generated Federated SPARQL queries gather novel and complementary data about substances in real time. Constant statistical calculations and update monitoring are avoided.
- *Availability:* Our data are made available to the entire research community. All the code to reproduce this study have been published online[3].

The lack of information about the endpoints availability and limits, makes any query not completely applicable in the context of federations of endpoints. Because of this the results could sometimes be incomplete. The current version of the framework is specialized for the life sciences, but under certain conditions it is extendable to other areas. Also, this approach is semantic-based and we are not able to collect data from other non-RDF data sources.

The rest of the paper is organized as follows: The second section gives an overview of the existing liter-

[1] http://147.91.203.161/specint

[2] CPCTAS-LCMB, Serbia, http://cpctas-lcmb.pmf.kg.ac.rs

[3] https://github.com/marijadjokic/SpecINT

ature of significance for the study area. The third section is devoted to novel data source integration reflecting the framework's scalability. We give, as a motivation example, two use cases which can be performed by using the framework in the fourth section. The fifth section describes the architecture and functioning principles of the proposed system for integration and query federation. The sixth and seventh sections discuss the results, benefits and limitations of the framework. The paper concludes with a summary of key points and directions for further work.

2. Related work

In this section, we provide an overview of both types of existing query generators, automatic and user-guided and highlight the main differences of the SpecINT framework in respect to existing generators. Basically, the SpecINT framework can be treated as a trade-off solution between these two approaches, since it can create queries which return relevant results, while not being dependent on human work and personalized experience. More precisely, it is not a completely automatic query generator able to create queries from scratch, but it picks up the existing pattern queries automatically and fits them into the final SPARQL query. Also, the framework requires less human interventions, since the simple mathematical apparatus provides satisfactory accuracy of the queries.

First, we provide a brief overview of the existing query generators developed for grained evaluation of Federated SPARQL query engines. These federation systems are basically developed for optimizing the query runtime thus their generators cannot be used for satisfactory user experience. Although some query generators can operate over distributed data sources, they cannot select data sources on-the-fly, which have the largest probability to consist of the relevant triples, neither can they connect repositories without global mapping. Some generators of this type are mentioned below. FedX [2] has been developed for comparing the general purpose of SPARQL query federation systems. It focuses on strategies which can decrease the number of query transmissions and reduce the size of intermediate results, but their drawbacks arise from the fact that they rely on a set of predefined static queries over particular data sources. The FedBench [8] is the only benchmark proposed for Federated query which evaluates the Federated query infrastructure performance including loading time and querying time. However, the FedBench has a static data source and query set, too. DAW [9] provides a set of static queries based on the characteristics of BSBM (Berlin SPARQL Benchmark) queries [10] from four public data sources. However, all the queries are statically generated thus cannot be used for specialized federation systems. Furthermore, these queries are simple in complexity (maximum of 4 triple patterns per query). To address this problem, some federation systems generate a random query set for a specified data source. A study by Umbrich et al. [11] extended query semantics for conjunctive Linked Data queries (LidaQ). LidaQ produces queries based on three main shapes (entity, star and path shapes) for Federated queries benchmark. This query generator produces sets of similar queries by doing random walks of certain breadth or depth. The query set generation of SPLODGE [12] is based on the data source characteristic that is obtained from its predicate statistic. Due to the random query generation process in SPLODGE using cardinality estimates, it is not uncommon that different queries with the same characteristics basically yield different result sizes. DARQ [13] and SPLENDID [3] make use of statistical information (using hand-crafted data source descriptions or VOID) rather than the content itself. Some data sources are continually expanding, so an application has to frequently update from RDF repositories. However, maintaining comprehensive and up-to-date cached data is an impossible task. New improvement came with ANAPSID [14] reflected in updating the data catalogue and execution plan at runtime. For a more comprehensive survey of the listed federation systems see [15]. FEASIBLE [16] is an automatic approach for the generation of benchmarks out of the query history of applications, i.e., query logs. The generation is achieved by selecting prototypical queries of a user-defined size from the input set of queries. In the paper [17] SQCFramework is proposed, a SPARQL query containment benchmark generation framework which is able to generate customized SPARQL queries from real SPARQL query logs. By using different clustering algorithms, the framework can generate benchmarks of varying sizes, with different significant (important) SPARQL features.

Beside the earlier listed shortcomings of the automatic query generators, these generators operate over the data sources given in advance, and have no ability to include other data sources without statistical calculations or global mapping. Also, they cannot handle the same data source over repositories simultaneously, where it has different predicates and connections. The only solution which explicitly deals with the integrated querying of distributed RDF repositories is described in [18]. Stuckenschmidt et. al theoretically described how to extend the Sesame RDF [19] repository to support distributed SeRQL queries over multiple Sesame RDF repositories. They use a special index structure to determine the relevant sources for a query. However, this approach is of a purely theoretical nature.

On the other hand, many existing applications provide a user-friendly interface for exploring bioinformatics data sources and allow users to intuitively create and perform Federated SPARQL queries, since SPARQL has a complex syntax. These applications can create useful queries which follows from the fact that the user follows the imposed steps through the interface, selects the relevant data sources (endpoints), predicates and subjects/objects, thus making room for decisions on how to connect these single pieces into a query by using the expert knowledge. Examples of such applications are: GoWeb [20], SPARQLGraph [21], Smart [22], BioQueries [23], BioSearch [24] etc. These applications were designed for the visual creation, editing and execution of biological SPARQL queries. PIBAS FedSPARQL [25] is an application that also runs Federated SPARQL queries for several bioinformatics topics. In this application the user has to navigate through the system and select query parts. As an advanced feature, PIBAS FedSPARQL provides the possibility of detecting similar data using results of predefined queries as an input.

However, all these applications are based on personal experience and affinities, while the drawbacks of some applications are also reflected in the impossibility of adding new datasets and in the supporting of a small number of specific endpoints. The SpecINT framework requires less human interventions, since the relevant data sources are selected by using the graph eigenvectors which show envious accurate results. Our approach gives more general answers to researchers who are not familiar with the SPARQL syntax and repositories organization.

Automatic query generators suffer from many disadvantages described in the previous section. The SpecINT framework represents a trade-off solution between automatic and user-guided query generators which is created in order to extract knowledge from the life science repositories. Today, there are several semantic based repositories (initiatives) for biological and chemical data sources integration: Bio2RDF [26], LODD [27], Chem2Bio2RDF [28], EMBL-EBI [29], Open PHACTS [30], ChemSpider [31] etc. Most current RDF infrastructures store information locally as a single knowledge repository according to certain design decisions. It means that the RDF models are replicated locally from remote sources and are merged into a single model regardless of the distributed nature of the Semantic Web. In many cases, we are forced to access external data sources from an RDF infrastructure without being able to create a local storage of the information we want to query. For example, we do not have permission to copy the data, data sources are too large to create a single model containing all the information, a data source is not available in RDF, but can be wrapped to produce query results in RDF format and so on [18]. On the other side, the Open PHACTS Discovery Platform [32] takes a local copy for performance reasons, but the data remain in their original form. It provides integrated access to 11 Linked Datasets covering information about chemistry, pathways, and proteins. Queries then extract relevant parts of each dataset based on contextualized instance equivalences retrieved from the Identity Mapping Service. However, no repository can cover all datasets, which only confirms the need to deal with repositories that are distributed across different locations enabling data freshness and scalability (an easy integration of novel data).

3. New data integration

This section is devoted to the publishing of new data sources. In order to make data widely available, data should be linked to other data sources by entity matching. According to LOD cloud statistics[4] almost all data sources have more than a thousand links to other data sources. But the mapping process is time consuming, and each data source has different predicates within different repositories. For example, DrugBank predicates for the drug targets in Bio2RDF and Chem2Bio2RDF are different (http://bio2rdf.org/-drugbank_vocabulary:target; http://chem2bio2rdf.org/-drugbank/resource/CID_GENE), which automatically means that the queries are different too. Following the unique identifier principle from database relation modeling, we propose a simple mapping between the data sources which can be performed very quickly. The mapping process is performed in such a way that the new data source causes no changes in the system, jus-

[4] http://lod-cloud.net/state/

tifying the system's scalability. In the following paragraph we have described in a few details how our data source is integrated with related repositories very easily.

Aiming to meet the principles of Linked Data[5] and make data available to a wide research community, the necessary precondition is data transformation into the Semantic Web context. In order to support CPCTAS laboratory staff to quickly reference and use a complex experiment structure, PIBAS (Preclinical Investigation of Bioactive Substances) ontology for modeling complex experimental structure was developed and presented in [33, 34]. Also, there should be no dependency on a single data source, because a substance can be present in one repository and not in another. Our substances are mapped to entities related to the identifier of the compounds and substances from other data sources (identification number - cid) regardless of the different URIs assigned to them. This approach provides flexibility for other similar laboratories. For simplicity's sake, in performed experiments target data sources are limited to the four most prevalent ones over repositories: PubChem [35], DrugBank [36], ChEBI [37] and KEGG [38]. This list could be extended, if necessary. Listing 1 represents the ontology map for the CPCTAS lab with some mapped substances. Similarly, following the same procedure a map for any novel data source could be created. In the experiments, PIBAS [33] and CHEMBL [39] maps demonstrate an easy usage.

```
<owl:NamedIndividual rdf:about="&PIBAS;102">
  <PIBAS:sameAs>pubchem:1235</PIBAS:sameAs>
  <PIBAS:sourceNumber>22</PIBAS:sourceNumber>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:about="&PIBAS;103">
  <PIBAS:sameAs>drugbank:DB00093</PIBAS:sameAs>
  <PIBAS:sourceNumber>2</PIBAS:sourceNumber>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:about="&PIBAS;104">
  <PIBAS:sameAs>kegg_ligand:C10107</PIBAS:sameAs>
  <PIBAS:sourceNumber>6</PIBAS:sourceNumber>
</owl:NamedIndividual>
```

Listing 1: Part of PIBAS map

4. Motivation: the SpecINT use cases

Our framework enables a variety of use cases, of which two are explained below. Note that this is a proof-of-concept project and the data is not updated.

4.1. New candidates for anti-cancer drugs

Data from the SpecINT could be of high value for chemists and biologists since these scientists have insights into the antitumor properties of complexes, which could reveal a possible strategy in the designing of new metal-based drugs. They could, for example, use our framework to link both, biological data (e.g., proteins' structure and their pathway) and chemicals (particularly drugs, interacting with proteins) together. Also, they can find out the influence the substances have on cancer cell-lines (e.g., IC_{50} values for estimation and quantification of cytotoxicity), and information about genes and proteins thus creating a coherent unity of results and complementary data. We could never be sure that all information is discovered by the framework it depends on whether data sources are updated frequently and their number, but we could always have an insight into research trends in recent years and get ideas for future research. For example, one of the major goals of modern bioinorganic and medicinal chemistry research is the development of novel metal-based drugs with pharmaceutical activity different from that of platinum-based therapeutics [40]. Among the non-platinum metal complexes studied for cancer treatment, palladium(II) derivatives were readily chosen due to their structural analogy with those containing Pt(II) complexes, good antitumor activity and lesser side-effect reactions. Recently Petrovic et al. [41] showed that the choosing of appropriate ligands could provide palladium(II) complexes, extremely cytotoxic to cancer cells.

It was shown, in the CPCTAS laboratory, that Pt(IV), Pd(II), and Rh(III) complexes induced oxidative stress and cytotoxicity in the HCT-116 colon cancer cell line [42]. Also, Živanović et al. [43] investigated the biological effects of bicyclic selenohydantoin $(Hid - Se)$ and its palladium(II) complex $((Hid-Se)_2Pd)$ on human colon HCT-116 and breast MDA-MB-231 cancer cell lines. They discovered that $Hid-Se$ and $(Hid-Se)_2Pd$ showed prooxidative and cytotoxic character, and strong antimigratory potential on metastatic MDA-MB-231 cells.

4.2. New integrated data

The SpecINT framework is not only a query generator over existing repositories. In Section 3 we described the procedure for new data source integration. This step makes all our data available to the research community and also demonstrates how oth-

[5] http://www.w3.org/standards/semanticweb/data

ers can publish their data and be connected to large initiatives such as KEGG, DrugBank, PubChem etc. These data arise from the CPCTAS investigation of the influence of bioactive substances on human cancer cell lines. Standardized tests cover monitoring of cytotoxicity, the type of cell death, the mechanisms of apoptosis, migration and angiogenesis and prooxidant-antioxidant mechanisms which are important for regulation of these processes. Experiments are based on protocols such as the MTT cytotoxicity test, AO/EtBr staining of cells for examination of the type of cell death, the Western blot technique for examining proteins, Multiplex and qRT-PCR, Transwell migration assays, Real Time Cell Analyses, and others.

It is well-known that cancer is the second leading cause of death after cardiovascular diseases, and finding the appropriate therapy is of key medical and scientific interest, with a potentially substantial economic impact. All types of cancer display a characteristic uncontrolled cell division followed by the ability of these cells to invade healthy tissues. This clearly shows the need for virtual integration of RDF data sources, since conducting all experiments which include a large number of complexes and all known cancers is a very expensive process. Beside the financial aspect, the framework enables the evidence that the researchers find for some substances to be compared with evidence included in other initiatives.

5. SpecINT architecture

The constant expansion of new data sources brings about problems in analysis of the disconnected and heterogeneous data which are crucial for future successful and purposeful surveys. Thanks to Semantic Web standards and online data exploration through open endpoints, it is possible to search these data sources in a single SPARQL query. The integration and extraction process of novel knowledge from these data is imminently problematic.

Retrieval of information about molecular structures from databases and RDF data sources is best done with unique identifiers. The IUPAC International Chemical Identifier (InChI) has recently acquired a prominent role as a unique identifier, and is increasingly used to make resources and literature machine readable [44]. Compared to the InChI, the Simplified Molecular Input Line Entry System (SMILES) is often not unique, causing relevant data to be lost in the search. In this paper, we use the InChIKey as the framework input - a hashed version of the full standard InChI, designed to allow for easy web searches of chemical compounds. Bearing in mind how the new and existing data sources are connected within repositories, let us explain the functioning of the framework and what happens in the background, from the forwarded input to the obtained SPARQL query as a result.

The architecture of the framework is shown in Figure 1 and is explained in the following subsections. The whole procedure of constructing the Federated SPARQL queries is presented part by part through the example. In Subsection 5.4 all these pieces are put together in a logical way, in order to arrive at the correct queries which encompass as many relevant results as is possible.

5.1. Sub-query patterns

The resulting large volume of data makes manual exploration very tedious and complicated. Moreover, the velocity at which these data change and the variety of formats in which bio-medical data are published makes it difficult to access them in an integrated form. In the case of semantically based data sources, the researchers have to explore each data source separately, its triples and mappings. Very often a data source consists of hundreds of thousands, even millions, of RDF triples. Further, the SPARQL queries have to be written and executed, the obtained data should be arranged in meaningful and useful knowledge, thus it can be used to support bio-medical experts during their work. In real-life applications the results should be filtered and well organized in the short term, which is almost impossible in these circumstances.

In Section 2 we provided an overview of the existing query generators, but this is not what we need for real-life tasks. Also, we listed several reasons why we cannot use these queries as the patterns (sub-queries) for our SPARQL queries. The lack of an integrated vocabulary makes querying this data more difficult, especially in situations when the URIs over repositories are not the same. Even when all generated sub-queries are valid, it is almost impossible to fit them all into one complex query which operates over repositories. Here, we use the pattern queries that were partially handpicked from initiative examples and partially handcrafted, since the correct results are important for our framework. Some examples of the used patterns are shown in Table 1. The bolded terms are unknown subjects and objects which are determined on-the-fly and changed with corresponding instances

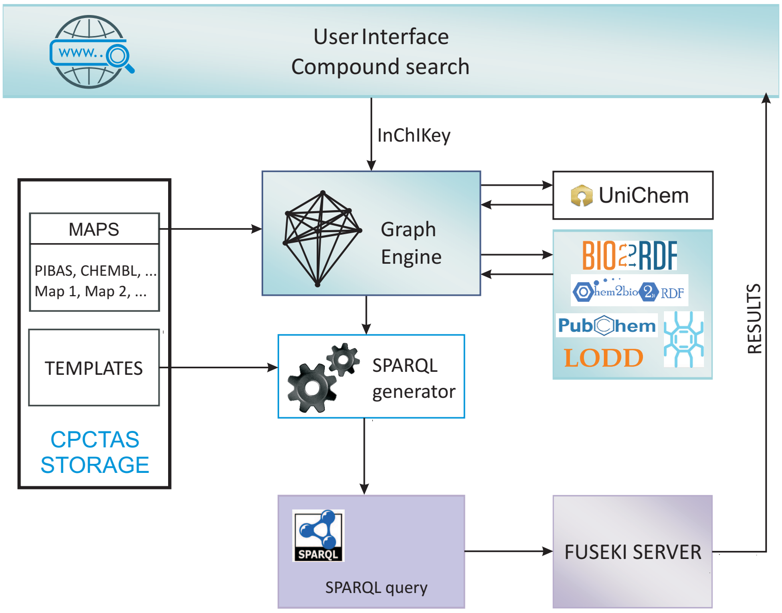

Fig. 1. SpecINT architecture

Table 1

Data source patterns within repositories.

Data source	Pattern for drug targets
DrugBank/Bio2RDF	**?drugbank_id** <http://bio2rdf.org/drugbank_vocabulary:target> **?target**
DrugBank/Chem2Bio2RDF	**?isValueOf** <http://chem2bio2rdf.org/drugbank/resource/DBID> **?drugbank_id** . **?isValueOf** <http://chem2bio2rdf.org/drugbank/resource/CID_GENE> **?target:**
Chembl/EMBL-EBI	**?activity** <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://rdf.ebi.ac.uk/terms/chembl#Activity> . **?activity** <http://rdf.ebi.ac.uk/terms/chembl#hasMolecule> **?chembl_id** . **?activity** <http://rdf.ebi.ac.uk/terms/chembl#hasAssay> **?assay** . **?assay** <http://rdf.ebi.ac.uk/terms/chembl#hasTarget> **?target** .

(URIs), while the predicates are bounded. Later, we will explore the "same as" kind of relationships within repositories which are used for the connection of these query patterns, without a common ontology between repositories.

5.2. *Data sources pre-selection*

In this subsection we describe the process of selecting data sources which looks at the most prominent resources for data of interest. In later steps, these results could be additionally filtered. The query generator should carefully determine the data sources for the query, since a wrong choice either leads to expensive communication with many intermediate results being memorized or the system failing to contribute any results.

The most practical way to connect two data sources is to use the values of the main notions which the data source is created around. Consider, for example, all drug information in KEGG can be connected with drugs in DrugBank by the owl:sameAs relation; which is an identity link that joins two entities having the same identity. To gather all information about a specific substance, the chemical structure of the substance is transformed into the corresponding InChIKey identifier. Then, we use the UniChem [45] search API from the European Bioinformatics Institute (EBI) to obtain a list of substance synonyms, but without their corresponding URIs. UniChem as a free available service allows mappings of small molecules based on adopted and stable standards, InChIs and InChIKeys. To be more precise, the synonyms represent the labels of a substance belonging to different data sources. For

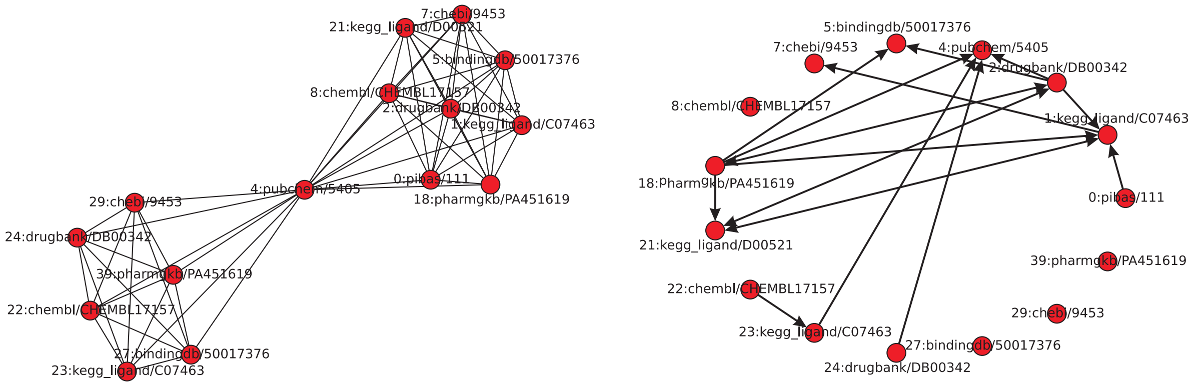

Fig. 2. Graph coalescence between Chem2Bio2RDF and Bio2RDF repositories.

InChIKey = GUGOEEXESWIERI-UHFFFAOYSA-N, some of the returned synonyms from the API are: CHEMBL17157, kegg_ligand C07463, drugbank DB00342, chebi9453, SCHEMBL5152 etc. Many data sources included within different repositories are not involved in UniChem. With the desire to encompass as many data sources as possible, related substance synonyms from one repository are added to this set of synonyms. This repository could have been Chem2Bio2RDF, Bio2RDF, LODD, etc. Then, the union of the obtained substance synonyms are used as the vertices in the graph (see Figure 2).

5.3. *Graph construction*

Taking into account that SPARQL query originates from the directed graph, we construct a graph from the obtained synonyms as the vertices labels following the relationships within each repository. This step will determine how the repositories can be connected, and additional filtering performed. Following the certain paths in the graph, the order of patterns is determined thus searching can be effected without a common ontology between resources. If these paths are wrong, the query will not be able to connect successive patterns and automatically the query will not be valid. This procedure involves two basic steps.

A. *Undirected graph construction*. This step reveals our hidden intention to save the information about vertices affiliation, connecting vertex between repositories, since the following graph perturbations and edge removal would mix up known affiliations. The repositories affiliations (URIs) are very important, because the same data source could have different predicates and interlink orientation. These URIs can be obtained easily from the URI pattern belonging to the relevant repository, but they are omitted for figure clarity.

All labels, found in the previous step, form the complete graphs K_n and K_m (for each repository), since every label represents the same substance from a different data source. In this way a substance is connected to all its representations within the selected repository. Following the background idea of saving vertices affiliation, a coalescence $V_{n,m}$ between two obtained graphs can be performed with any vertex whose label belongs to the repositories intersection (see left side of Figure 2). The selected vertex is used as a bridge for the crossing from one repository to another. According to the results of Theorem 1 (see Appendix A) we can calculate Fiedler eigenvector s (sign) for the graph $V_{n,m}$ and divide the vertices set into two disjoint sets, with positive and negative vertices (the sign of vector coordinates), and one *null* vertex (bridge).

For the better explanation of our example, besides the dataset label, all vertices also have number label, starting from 0 to $|V|$, where V is the number of selected datasets. After the isolation of non-relevant or non-connected vertices, some of the numbers are lost. Deleted nodes do not influence the algorithm since the graph spectrum and the corresponding eigenvectors are graph invariants (a property of a graph that is preserved by isomorphism). These numbers represent the coordinates' numbers in the vectors s and r. In our example, vertex with label 0 takes value from the first vector coordinate, label 1 from the second coordinate and so on.

B. *Directed graph construction*. For the valid results' retrieval, the process of creating the most suitable sequence of the data source labels is performed. Prior to query generation, the framework has to check the existence and orientation of the edges. It is possible to find interlinks between data sources by searching the specific keywords being a substring of a property string in a tie between two compounds. With edges (source, target) obtained from the triples (?source ?property ?target) we can convert the graph $V_{n,m}$ into the digraph $D_{n,m}$ according to the nature of the SPARQL query. For each substance, different $V_{n,m}$ and $D_{n,m}$ are obtained. This step includes removing all the nonexistent edges, but not the isolated vertices, since the Fiedler eigenvector is previously determined for the graph with all vertices. With Fiedler eigenvector we paved the way for the conversion process. When the digraph $D_{n,m}$, without isolated vertices, is disconnected, the whole procedure for the unused label is repeated.

It is known that the importance of each vertex is proportional to the sum of the importance of all the vertices that link to it. Simple calculation says that this is an eigenvalue and eigenvector problem (more details can be found in [6]). Now, for oriented graph $D_{n,m}$ we can determine nonnegative eigenvector r (rank), coordinates of which measure the relative importance of the vertices. Once we have the eigenvector, the most important vertex is the one with the largest entry in that eigenvector, the next most important has the second largest entry, and so forth. Now, we can follow the most probable path over repositories as search engines do, taking care of vertices affiliation. All steps are presented in Procedure 1.

However, this procedure does not guarantee that all data sources we want are covered. In the case when a novel source is integrated (low rank value) or the specific answers are preferred (located in specific data sources) it is necessary to provide a way to force the selection of these data sources. For example, if drug targets are in focus, specific vertices will be favored for better results. For this purpose we developed a simple ontology which consists of information about data sources. Also, we developed several heuristics which are capable of influencing the vector r and covering such vertices according to the ontology content. The tested heuristics and all results will be presented in the evaluation section.

Procedure 1: Graph construction procedure.

```
Data: InChIKey, data repositories $R_1$ and $R_2$
Result: Directed graph $D_{n,m}$, eigenvectors $s$ and $r$
1  Intersection:={common data sources};
2  UniChem:= {UniChem synonyms for InChIKey};
3  firstGraph:= {synonyms from $R_1$ for every UniChem label)};
4  secondGraph:= {synonyms from $R_2$ for every UniChem label};
5  Construct complete graphs $K_n$ from firstGraph and $K_m$ from secondGraph labels;
6  Construct coalescence $V_{n,m}$ with any label from Intersection and calculate eigenvector $s$;
7  Convert $V_{n,m}$ to digraph $D_{n,m}$;
8  Remove nonexistent arcs and favor vertices;
9  if $D_{n,m}$ is disconnected then
10     goto step 6 and try unused Intersection label;
11 end
12 Calculate eigenvector $r$ of $D_{n,m}$;
```

5.4. Join ordering and building queries

This subsection is dedicated to the algorithm for the SPARQL queries construction from two vectors s and r and the hand-crafted patterns. Here, we will explain how to follow vertex affiliations and align the most relevant data sources in a query in order to ensure results over repositories. Also, in this phase prefixes related to the data sources of vertices and their corresponding query patterns are determined. This vertex serves as a neutral source (bridge) of the RDF triple, either the subject or object in the patterns. Finally, when the idea is exposed, the highest ranked vertices are used in order to find the best path to the central vertex from both sides, positive and negative.

The input for this phase are two eigenvectors, s and r. The first eigenvector s splits the graph $D_{n,m}$ into two connected components with different signs of coordinates, and one connecting *null* vertex. This eigenvector carries vertices affiliation, and after graph transformations, these signs carry the vertices origin. The *null*-vertex presents an articulation point (bridge). The second eigenvector r represents the most important vertices in both connected components. Coordinate values in r actually suggest the most probable paths to the bridge within sign zones providing repositories link-up (see the right side of Figure 2).

For simplicity's sake, let us suppose that the first connected component contains vertices with positions $0, 1, \ldots, n-2$, for the cut-vertex it is $n-1$, and the second connected component is with $n, n+1, ..., n+m-2$ positions. In general, the query path consists of two simple paths: one from any positive vertex to the null-vertex and the second one from any negative vertex to the null-vertex. The best ranked vertex is selected for the initial vertex. In the path, the subsequent object represents the best ranked vertex from the subject's neighborhood. If a choice of multiple vertices with the same rank is present, path construction will diverge simultaneously for each vertex. Once created, path over repository means different information availability for a substance. For every vertex in the path we use specific patterns for sub-query completeness (see examples in Table 1). Edges between the vertices of the digraph $D_{n,m}$ are used for the patterns chaining, in such a way that an object from a pattern becomes a subject in the following pattern. For example, the subject *drugbank:DB00342* is obtained as an object from the triple (kegg_ligand:D00521, http://bio2rdf.org/kegg_vocabulary:x-drugbank, ?drugbank), whose predicate represents an edge in digraph $D_{n,m}$. In this way we can connect our substance with the same substances over different repositories.

Let us see the algorithm in action. For the two graphs in Figure 2 two eigenvectors are calculated: Fiedler eigenvector s and rank eigenvector r. Their coordinates are s = [0.231, 0.231, 0.231, 0, 0.231,-0.231, 0.231, -0.309, 0.231, 0.231, -0.309, -0.309, -0.309, -0.309, -0.309] and r = {24:0.0094, 39:0.0094, 27:0.0094, 21:0.0393, 22:0.1477, 23:0.0722, 18:0.0125,-29:0.0094, 1:0.0882, 0:0.1477, 2:0.0223, 5:0.0143,-4:0.0490, 7:0.0344, 8:0.0094}. Following the steps of Algorithm 1 we obtain two paths. The path over positive vertices belonging to Bio2RDF initiative is: $0 \rightarrow 1 \rightarrow 21 \rightarrow 2 \rightarrow 4$ ($pibas/111 \rightarrow kegg\_ligand/C07463 \rightarrow drugbank/DB00342 \rightarrow pubchem/5405$). The second path over negative vertices belonging to Chem2Bio2RDF initiative is: $22 \rightarrow 23 \rightarrow 4$ ($chembl/CHEMBL17157 \rightarrow kegg\_ligand/C07463 \rightarrow pubchem/5405$). We allocated Bio2RDF to the positive side, and Chem2Bio2RDF to the negative side, but the same process can be revolved to obtain a slightly different query. Finally, by following these paths and vertex patterns we construct SPARQL query (see Listing 2) which retrieves *targets* for the initial substance. For more examples visit the website[6].

Algorithm 1 Federated SPARQL queries generator.

```
Data: Fiedler eigenvector $s = \{s_0, s_1, ..., s_{n+m-2}\}$,
      rank eigenvector $r = \{r_0, r_1, ..., r_{n+m-2}\}$,
      repositories $R_1$ and $R_2$
Result: Federated SPARQL query
$query = \varnothing$
$null\_vertex \leftarrow n - 1$
$subject \leftarrow label(i)$, $i$ - the best ranked positive vertex
repeat
    $neighbors \leftarrow$ positive neighbors for $subject$
      $object \leftarrow$ label(the best ranked neighbor)
      $add\_subquery(subject, object, R_1, pattern)$
      $subject \leftarrow object$
until $object = null\_vertex$;
$add\_subquery(subject, null\_vertex, R_1\ or\ R_2, pattern)$
  $subject \leftarrow label(i)$, $i$ - the best ranked negative vertex
repeat
    $neighbors \leftarrow$ negative neighbors for $subject$
      $object \leftarrow$ the best ranked neighbor
      $add\_subquery(subject, object, R_2, pattern)$
      $subject \leftarrow object$
until $object = null\_vertex$;
return $query$
```

6. Evaluation

In this section we give an evaluation of the framework's ability to select the most relevant data sources over repositories, taking into account their specificities. We checked the correctness of the generated queries too. Our methodology is not based on the common ontology which connects repositories, but on the detection of the "same as" relationships between data sources. The framework builds the graphs of these relationships within each repository, then uses them to build appropriate SPARQL queries which operate over repositories. The goals of the evaluation are (1) to measure the performance of the SpecINT engine in terms of relevant data sources selection, and (2) to check the validity of the created SPARQL queries. Evaluation in this context basically means checking if the generated queries meet our primary goals, i. e. whether they can actually retrieve relevant results from different data sources (and repositories) and whether the framework responses could show the actual trends in research communities related to the anti-cancer drugs. In the following, we explain our experimental setup and the evaluation results.

[6] http://147.91.203.161/specint/example.html

```
PREFIX  drugbank: <http://bio2rdf.org/drugbank:>
PREFIX  pibas: <http://cpctas-lcmb.pmf.kg.ac.rs/2012/3/PIBAS#>
PREFIX  drugbank1: <http://chem2bio2rdf.org/drugbank/resource/drugbank_drug/>
PREFIX  kegg_ligand: <http://bio2rdf.org/kegg:>
PREFIX  chembl_molecule: <http://rdf.ebi.ac.uk/resource/chembl/molecule/>
PREFIX  cco:  <http://rdf.ebi.ac.uk/terms/chembl#>
PREFIX  chembl_mapp: <http://cpctas-lcmb.pmf.kg.ac.rs/2012/3/chembl#>

SELECT DISTINCT  ?target
FROM <http://cpctas-lcmb.pmf.kg.ac.rs/2012/3/PIBAS/pibasmapping.owl>
FROM <http://cpctas-lcmb.pmf.kg.ac.rs/2012/3/PIBAS/chemblmapping.owl>
WHERE
  { {   { pibas:111 pibas:sameAs kegg_ligand:C07463 .
          pibas:111 pibas:hasTarget ?target.
        }
      UNION
        { SERVICE SILENT <http://kegg.bio2rdf.org/sparql>
            { kegg_ligand:C07463 <http://bio2rdf.org/kegg_vocabulary:gene> ?target;
                                      <http://bio2rdf.org/kegg_vocabulary:same-as> ?kegg_ligand.
            }
        }
      UNION
        { SERVICE SILENT <http://kegg.bio2rdf.org/sparql>
            { kegg_ligand:D00521 <http://bio2rdf.org/kegg_vocabulary:gene> ?target;
                                       <http://bio2rdf.org/kegg_vocabulary:x-drugbank> ?drugbank.
            }
        }
      UNION
        { SERVICE SILENT <http://drugbank.bio2rdf.org/sparql>
            { drugbank:DB00342 <http://bio2rdf.org/drugbank_vocabulary:target> ?target;
                                   <http://bio2rdf.org/drugbank_vocabulary:x-pubchemcompound> ?pubchem.
            }
        }
      UNION
        { SERVICE SILENT <http://147.91.203.161:8890/sparql>
            { ?value <http://chem2bio2rdf.org/pubchem/resource/CID> pubchem:5405.
              ?value <http://chem2bio2rdf.org/pubchem/resource/CID_GENE> ?target.
            }
        }
      UNION
        { SERVICE SILENT <http://147.91.203.161:8890/sparql>
            { ?isValueOf <http://chem2bio2rdf.org/drugbank/resource/DBID> drugbank1:DB00342.
              drugbank1:DB00342 <http://chem2bio2rdf.org/drugbank/resource/CID> ?pubchem.
              ?isValueOf <http://chem2bio2rdf.org/drugbank/resource/CID_GENE> ?target.
            }
        }
      UNION
        { SERVICE SILENT <https://www.ebi.ac.uk/rdf/services/chembl/sparql/>
            { ?activity <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> cco:Activity.
              ?activity cco:hasMolecule chembl_molecule:CHEMBL17157.
              chembl_molecule:CHEMBL17157 cco:moleculeXref ?drugbank1.
              ?activity cco:hasAssay ?assay.
              ?assay cco:hasTarget ?target.
            }
        }
    }
}
```

Listing 2: Final SPARQL query

6.1. Experimental Setup

The SpecINT framework provides information about physical and chemical properties of a substance, substance interaction with various protein targets, substance cytotoxicity on various cell-lines and so on. In order to evaluate the framework's ability to collect specific data, the researchers started the framework for 50 substances/compounds, randomly selected from the data sources used in the experiments. For the methodology testing, only substances which belong to both repositories are selected. Table 2 lists one part of the used InChIKeys with their molecular formulas and short names.

Moreover, for the experiments we use substances from the CPCTAS laboratory, originally synthesized by chemists for new experiments. CPCTAS possesses a certain number of various healthy and cancer cell-lines, and at the beginning of every investigation it is of crucial importance to know whether the substance of interest has already been analyzed. The researchers could get information about the synthesis of similar substances, and substance properties, getting evidence and comparing their findings with the findings for similar substances.

Table 2

One part of the tested InChIKeys with primary information.

Id	InChIKey	Name	Formula
1.	WNMJYKCGWZFFKR-UHFFFAOYSA-N	ALFUZOSIN	C19H27N5O4
2.	IRYJRGCIQBGHIV-UHFFFAOYSA-N	TRIMETHADIONE	C6H9NO3
3.	MHWLWQUZZRMNGJ-UHFFFAOYSA-N	NALIDIXIC ACID	C12H12N2O3
4.	CXOXHMZGEKVPMT-UHFFFAOYSA-N	CLOBAZAM	C16H13ClN2O2
5.	MJFJKKXQDNNUJF-UHFFFAOYSA-N	METHIXENE	C20H23NS
6.	GUGOEEXESWIERI-UHFFFAOYSA-N	TERFENADINE	C32H41NO2
7.	GSDSWSVVBLHKDQ-UHFFFAOYSA-N	OFLOXACIN	C18H20FN3O4
8.	PTOAARAWEBMLNO-KVQBGUIXSA-N	CLADRIBINE	C10H12ClN5O3

6.2. Repositories

In the last decade, several large cheminformatics and bioinformatics repositories were founded. Every initiative is special in some manner and all of them have made a comprehensive shift towards presenting data to a wide research community. Some of the most popular solutions based on Semantic Web technologies are: Bio2RDF [26], LODD [27], Chem2Bio2RDF [28] and Open PHACTS [30].

In our experiments we focus on two repositories Chem2Bio2RDF and Bio2RDF. Chem2Bio2RDF is one of the most popular repositories based on Semantic Web technologies which store more than 80 million triples. It covers around 25 different data sources relating to chemical/biological needs which aggregate genes, compounds, drugs, pathways, side effects, diseases, and MEDLINE/PubMed documents (last update in 2009). Bio2RDF manages to integrate public bioinformatics databases and convert them into 11 billion triples across 35 datasets. Its last release (third) dates from July 2014. Also, in our experiments we have worked with the underlying ChEMBL database from the EBI[7] and PIBAS database form CPCTAS laboratory to demonstrate an easy data integration.

6.3. Ground-truth

The task for the framework evaluation was assigned to the chemists and biologists employed at the Centre for Preclinical Testing of Active Substances (CPCTAS), Faculty of Science, University of Kragujevac. First, the members of CPCTAS, with our help, explored Chem2Bio2RDF and Bio2RDF repositories taking into account that these repositories have different predicates. This is an important step for creating a general picture of all data sources, their content and how data are connected. For the evaluation, 3 biologists and 3 chemists reviewed each recognized link between data sources manually. They checked whether the edges (predicates) between substances are real, i.e. whether there exists a triple which contains the entities of the substances. The entities position in a triple (subject or object) is an important part for the final results, since the edge orientation determines the path thus influencing the patterns order in a final query. For the specific task (targets, IC_{50} and cell lines) they counted the number of relevant data sources which are included in the final query. They also checked whether the queries are valid, i.e. whether the obtained results correspond to the asked question and substance. With appropriate use of the application PIBAS FedSPARQL [25] a double check of these results is performed.

6.4. Heuristics for the path navigation

Although we provide some theoretical evidence for the eigenvector coordinates signs (see Appendix A), we could not be sure that the vertex ranking will lead us to the best possible results. More precisely, the main task is to find a way of connecting two paths in the cut-vertex, positive and negative, but this selection should include the most relevant data sources from both sides. The cut-vertex serves as a mediator (bridge) for the crossing from repository to repository. One thing that is immediately apparent is that it would be impractical to explore all data sources and all their paths. The state-of-the-art algorithms for path finding such as Prim's and Kruskal's algorithms are not applicable in the case when it is necessary to favor particular vertices. One of the ideas could be increasing the edge weight, but it cannot be performed when the existing edges differ from substance to substance. Now, the question is what is the best approach for path selection which can include as many relevant data sources as is pos-

[7] https://www.ebi.ac.uk/chembl/

sible thus the loss in information is minor. We therefore tested the following three easily implementable heuristic methods that use only the vertices degrees of $D_{n,m}$. In the step when the new data source should be selected among the neighbors, it is selected in one of the following ways:

- **Degree:** It selects the vertices with the largest degree.
- **PageRank:** It selects the vertices with the largest rank.
- **Favored PageRank:** It selects the vertices with the largest rank which are user-guided.

The concepts of the first two heuristics are clear. The third heuristic, *Favored PageRank*, is introduced since the forced vertex selection is a necessary precondition if we want to favor specific data sources depending on the question. For this purpose a new fictitious vertex with a large rank is added to the graph. The edges from this high-rank vertex to the low-rank vertices can increase their rank without violating the existing graph structure. If we want to include high-rank vertex in the path (e.g. PIBAS vertex), new edges which point to it are created. For clarity's sake the fictitious vertex is removed from the figure. Similarly, for the vertices of interest that might be encountered on the path, the vertex self-loops can be used in accordance with the present knowledge about data sources. This additional knowledge is stored in a simple ontology which includes data sources categorization and affiliation.

6.5. Results

In this subsection we present the obtained results for 50 substances and for each heuristic. Here, only the results related to the drug targets are shown, since the results for the cancer cell-lines and IC_{50} value are similar. For each substance the framework was started three times, for each heuristic separately, and its positive and negative paths over repositories were found. Notice that not all selected data sources carry the information related to the question, in this case for the drug targets. For the evaluation task only the data sources which potentially consist of relevant triples are taken into consideration, although all vertices are used for the query. In Table 3 the numbers of relevant sources which are covered by the algorithm (query), for each heuristic versus the ground-truth, are shown.

The most effective selecting method turned out to be *Favored PageRank*. Involving domain knowledge in the algorithm improves the final results. This means that one could upgrade the algorithm by using novel expert knowledge which is especially important in the case when the unknown data sources are integrated. In very rare cases a *Degree* approach can join the paths over repositories. Even when it succeeds to construct a full-path, this path contains a small number of relevant sources. Also, the *Degree* approach has not proved as a good solution in practise because of the query branching. It considers execution and evaluation of several queries resulting in the framework slowing down. The *PageRank* approach gives much better results in most cases including a higher percentage of success in paths joining than the *Degree* does. These good results could be explained by the fact that the best ranked vertices are connected with many data sources and it is easer to perform merging. The main drawback of this approach is that low-rank vertices (new data sources) could not be covered in the paths. For example, a substance from CPCTAS is connected with only one substance and as such is worthless compared to the "strong" data sources. Also, this approach has not convinced us that the paths include all data sources of interest.

After the manual inspection, the results of the evaluators showed that by using SpecINT (*Favour PageRank*), we achieve a precision of 86% in covering relevant data sources for the drug targets. The precision of 71% and 75% is achieved for the cancer cell-lines and IC_{50} value, respectively. The results could vary from time to time depending on the availability of the endpoints. This means that some edges will not exist in the graph and that the paths are changed.

Beside the relevant data sources selection, the query validity is also tested. The large number of covered data sources does not guarantee that the paths are connected in the connection vertex. Although the used graph construction enables repositories visiting, sometimes there is no edge which connects the vertex before the last in the path with the cut-vertex. Even when the necessary edges exist, it does not mean that the edge orientations are appropriate. There are about 13% of such cases, which automatically means that the queries cannot be created. For example, the substance GSDSWSVVBLHKDQ-UHFFFAOYSA-N cannot return the results.

As an addition to the use case from Subsection 4.1, *Favour PageRank* heuristic for the tested substances is applied. For every substance *Id*, Table 4 shows the number of extracted targets, cell-lines and IC_{50} values for every data source, as well as repository affiliation. These data give a short overview where the substance

Table 3

Number of relevant sources on the path per heuristic.

InChIKey	Degree	PageRank	Favour PageRank	Ground-truth
WNMJYKCGWZFFKR-UHFFFAOYSA-N	4	4	6	6
IRYJRGCIQBGHIV-UHFFFAOYSA-N	0	4	6	7
MHWLWQUZZRMNGJ-UHFFFAOYSA-N	0	0	4	7
CXOXHMZGEKVPMT-UHFFFAOYSA-N	0	4	6	7
MJFJKKXQDNNUJF-UHFFFAOYSA-N	4	4	6	6
GUGOEEXESWIERI-UHFFFAOYSA-N	0	0	5	6
GSDSWSVVBLHKDQ-UHFFFAOYSA-N	0	0	0	8
PTOAARAWEBMLNO-KVQBGUIXSA-N	4	4	5	6

Table 4

Obtained results for previously listed substances. For every substance are presented the numbers of items for targets, cell-lines and IC_{50} values, found in data sources within different repositories

	Bio2RDF			**Chem2Bio2RDF**			**CHEMBL**			**CPCTAS**		
Id	*Target*	*CL*	IC_{50}	*Target*	*CL*	IC_{50}	*Target*	*CL*	IC_{50}	*Target*	*CL*	IC_{50}
1.	4	0	0	0	0	0	32	2	6	1	1	1
2.	1	0	0	0	0	0	129	0	0	1	1	1
3.	1	0	0	0	0	0	161	0	0	1	1	1
4.	1	0	0	0	0	0	7	0	0	1	1	1
5.	5	0	0	0	0	0	2	0	0	1	2	4
6.	7	0	0	2	0	0	210	10	29	1	1	1
7.	3	0	0	0	0	0	260	2	2	1	1	1
8.	10	0	0	2	0	0	170	24	27	1	1	1

was tested and where additional information could be found.

6.6. Comparison with other frameworks

In this section we provide a comparison of our framework with large Open PHACTS project, which is now used in pharmaceutical companies. Two systems can only be compared if they produce results for a given substance/compound. The sources which contribute to evaluation process are shown in Section 6.2 (including their versions). Although the SpecINT framework is proof-of-concept project, designed to operate over repositories without a common ontology between resources, we are able to return fresh data and integrate novel datasets. The goal of this comparison is to show that the Federated SPARQL queries, generated on the SpecINT, are able to achieve some complementary result compared to the results obtained by using Open PHACTS Discovery Platform [32]. For the evaluation task, the query results designed for three different tasks are tested. These tasks include discovering of 1) targets, 2) cell-lines, and 3) the corresponding IC_{50} values. The evaluation comprises a total of 24 queries: 8 queries for targets, 8 queries for cell-lines and 8 queries for IC_{50} value. All queries on SpecINT framework are generated as described in previous sections, while API version v2.2[8] for the Open PHACTS platform is used. In the following text, we provide results obtained from these two frameworks.

For each InChIKey used as an input in the SpecINT, corresponding compound URI (seed URI) was used for the Open PHACTS platform. The experts from CPCTAS checked the results and made comparison between outputs from both platforms. One part of the tested substances (InChIKeys) and returned results for the targets are presented in Table 5. As far as the drug target task is concerned, we can notice that overlapping results come from ChEMBL dataset. In this case, the SpecINT produced some complementary results obtained from DrugBank dataset (from Bio2RDF and Chem2Bio2RDF repositories) and PIBAS ontology. Open PHACTS API does not return these additional results, although it contains DrugBank dataset. An overlapping between outputs of two platforms is also

[8] https://dev.openphacts.org/admin/access_details accessed 15 May 2018

Table 5

Number of returned results. One part of the tested InChIKeys for drug targets.

Id	InChIKey	OpenPhact	SpecINT
1.	WNMJYKCGWZFFKR-UHFFFAOYSA-N	32	40
2.	IRYJRGCIQBGHIV-UHFFFAOYSA-N	129	132
3.	MHWLWQUZZRMNGJ-UHFFFAOYSA-N	161	163
4.	CXOXHMZGEKVPMT-UHFFFAOYSA-N	7	10
5.	MJFJKKXQDNNUJF-UHFFFAOYSA-N	2	13
6.	GUGOEEXESWIERI-UHFFFAOYSA-N	210	220
7.	GSDSWSVVBLHKDQ-UHFFFAOYSA-N	260	267
8.	PTOAARAWEBMLNO-KVQBGUIXSA-N	170	183

large in the case of cell-lines and IC_{50} values, since both platforms include EBI-RDF ChEMBL dataset. All these results are publicly available in more details on figshare repository[9].

From obtained results, we conclude that both approaches offer a great starting point for discovering data. Although a difference between compared results is small, both frameworks can offer complementary data to the research community. New experimental results from CPCTAS laboratory are publicly available through the PIBAS ontology (see Subsection 4.1). On the other side, Open PHACTS, as well as the SpecINT, offers a possibility for novel data integration thus providing a possibility for additional data. For example, for the substance with InChIKey=GSDSWSVVBLHKDQ-UHFFFAOYSA-N, Open PHACTS discovery platform returns more information about tested cell-lines than the SpecINT. Also, Open PHACTS API offers more options that the SpecINT improved version will implement.

6.7. Usability and Usefulness

The user interface on the top of the framework is developed too. It presents an easily understandable view of the information obtained in the back-end. After the methodology was developed, we had to ensure that the user interface is useful enough to be potentially used for real life cases. To assess the potential usability of our system, we used the seven-item Likert scale-based System Usability (SUS) questionnaire [46]. The survey was completed by CPCTAS staff. In order to numerically analyze the survey results, we translated the Likert scale responses to numbers using the following five point scale: 1 = strongly disagree; 2 = disagree, 3 = neutral; 4 = agree; 5 = strongly agree. The results of the survey are shown in Figure 3.

The responses to question 1 (I felt very confident using the system) suggest that our system is very well adopted by users (average score to question $1 = 4.1 \pm 0.91$). The responses to question 2 (I think the system was easy to use) implies that our system is comfortable and simple to use (average score to question $2 = 4.7 \pm 0.66$). The users positively rated (average score to question $3 = 4.5 \pm 0.76$) question 3 (I found the various features in this system were well implemented). Implementation of graphics and possibility to see a real, live feedback from online endpoints have a much better effect on users. This additionally motivated us for further development. The comebacks to question 4 (I will recommend the system to other users) suggest that our system has positive feedback from users (average score to question $4 = 4.3 \pm 0.73$). The responses to question 5 (I think that the system gives me complementary data) indicates that our system was supportive in searching for complementary data that would be used for future QSAR analysis (average score to question $5 = 3.9 \pm 0.91$). The responses to question 6 (I would like that the system supports more than two repositories) suggests that users find our system positive for their needs and that the adding of new initiatives would only be a plus (average score to question $6 = 4.3 \pm 0.73$). The responses to question 7 (I think that the system does not always work) indicate that our users found the system readily available (average score to question $7 = 1.2 \pm 0.18$). The score from this question could be justified by the fact that endpoints are sometimes not reachable. Generally, we achieved an average score of 4.23. The data indicate that the overall impression was positive and encouraging, and that we found the SpecINT to be very useful.

[9] https://figshare.com/articles/Evaluation/6352496

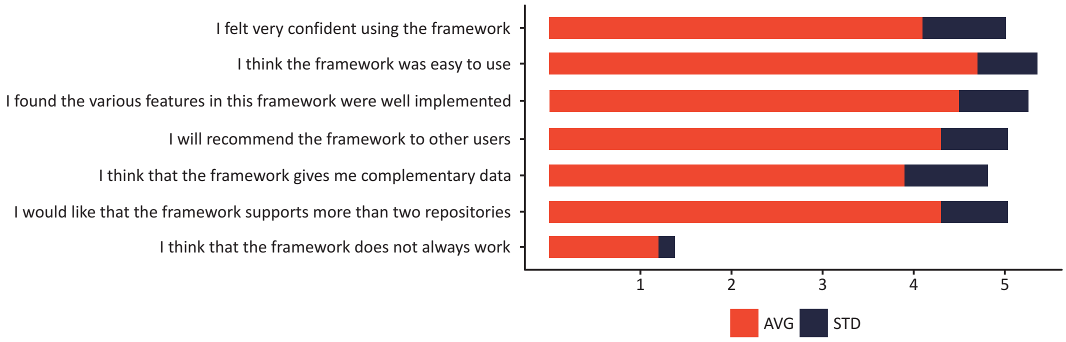

Fig. 3. Result of usefulness evaluation by using our custom questionnaire.

7. Limitations of the framework

In this section we cover some of the known limitations of our framework.

Endpoint is down: the SpecINT depends on the availability of the used SPARQL endpoints. A local copy of the endpoints cannot be retrieved due to the large size of the data source measured in terabytes. Generated queries skip an endpoint which is down, but the constantly present fact is that some of the edges are not found.

Cut-vertex choice: choosing a different cut-vertex can decrease the number of selected vertices in the entire path over repositories, thus excluding some vertices of interest. Experiments show that managing a balance between paths and cut-vertex selection is not an easy task. Although the differences in results are very small, future work should be focused on obtaining maximal performances. However, this implementation demands novel preprocessing steps and longer execution time.

Two repositories: the main drawback of the current version of the framework is the operation with two repositories. A similar theorem should be proved for the coalescence of the chain of complete graphs. However, this brings greater software complexity and new problems related to the previous cut-vertex problem. Also, these repositories should deal with similar topics and have at least one vertex in common thus they can be connected following the idea.

Up-to-date data: this is a proof-of-concept project and the data is not updated. This severely limited the data available in the SpecINT framework. For example, it uses Chem2Bio2RDF and Bio2RDF conversion of PubChem which have not been kept up-to-date with the underlying PubChem RDF database [47].

8. Conclusion and future work

In this paper, we have presented a new approach for information retrieval set in the background of the SpecINT framework. The most important contribution of this work is Federated SPARQL queries construction in a scalable manner according to the existing paths in the graph. The "same as" kind of relationships within repositories are detected for the graph construction thus the searching process can be effected without a common ontology between resources. The framework appears as a trade-off solution between automatic and user-guided generators for the Federated SPARQL queries. This framework requires less human interventions avoiding personal experience and affinities, since it uses the coordinates of the graph eigenvectors for the most relevant data sources selection and automatic joining of their sub-queries. Moreover, the achieved improvement is also reflected in the fact that the framework is able to operate over repositories taking into consideration their specific data representation and inter-connections. This methodology is not dependent on constant update monitoring, but everything is done on-the-fly, therefore expensive statistical calculations are avoided.

The SpecINT enables scientists to find information of interest on the web (under certain circumstances, e.g. up-to-date data), and it also encourages other laboratories to publish data thus extending the general idea. New members can publish their experimental results easily and become an integral part of the new virtual space dedicated to chemistry and biology. The framework arises from an optimistic idea to potentially save time and resources needed for chemical and biological investigations.

For future work, it would be interesting to study the weighted graphs obtained from RDF data, the effects of changing weighted functions of the edges and vertices for path generation and eigenvectors coordinate changes. The eigenvectors represent an excellent mathematical apparatus for future framework improvements when more than two repositories are considered.

Acknowledgement

The authors are grateful to the Ministry of Education, Science and Technological Development of the Republic of Serbia for financial support (Grant Nos. III41010 and 174033).

Appendix A. Coalescence with complete graphs

Fielder's papers [48, 49] initiated a new era in which we can use the sign of the eigenvectors' coordinates for cut finding. In [49] it was proved that the second smallest eigenvector of the Laplacian matrix can be used for determining positive and negative vertices in a graph thus providing room for distinguishing the connected components of a graph after vertex removal. Let $G_1 = K_n$ and $G_2 = K_m$ be the complete graphs with n and m vertices respectively, and let $V_{n,m} = G_1 \cdot G_2$ be its coalescence with vertex v_n. By removing a cut-vertex of the graph $G = G_1 \cdot G_2$, we get a disconnected graph with two components. In the following, as a sequel of Theorem 3.12 in [49], we proved that for $V_{n,m}$ always holds case B, $\forall\, n, m \in N$. In this way we concluded that no component of $V_{n,m}/\{v_n\}$ contains both positively and negatively valuated vertices.

Proposition 1. *(see [50], p.185) We have $\mu_1(\overline{G}) = 0$ and $\mu_i(\overline{G}) = n - \mu_{n-i+2}(G)$ for $(i = 2, 3, ..., n)$, where $\overline{G}$ denotes the complement of G.*

Theorem 1. *Let $z = (z_i)$ be the Fiedler vector of the graph $G = V_{n,m}$. Vertices belonging to $N(z)$ are in one block, while vertices belonging to $P(z)$ are in another block of the graph G. Exception is cut-vertex v_n which has 0-value coordinate in the eigenvector z.*

Proof. It was proved earlier that $z_2(G) = z_n(\overline{G})$ (see the proof of Proposition 1). Instead of finding the eigenvector corresponding to the second smallest Laplacian eigenvalue μ_2 of the graph G, we shall find the eigenvector corresponding to the μ_n eigenvalue of the graph $\overline{G} = K_{n-1,m-1} \cup \{v_n\}$. Since the graph $\overline{G}$ has one isolate vertex v_n, we can calculate an eigenvector for μ_n for the subgraph $H = K_{n-1,m-1}$, and after that we can add zero-value to the eigenvector in the n-th place.

On the other hand, instead of an eigenvector for μ_n for the subgraph $H = K_{n-1,m-1}$ we shall find an eigenvector for μ_2 for $\overline{H}$. In the Laplacian spectrum for the graph $\overline{H}$ we have two 0-valued eigenvalues, $\mu_1 = \mu_2 = 0$. The eigenvectors for the graphs K_{n-1} and K_{m-1} corresponding to the zero-valued eigenvalues are $e(K_{n-1}) = \underbrace{(1, 1, ..., 1)}_{\text{n-1}}$ and $e(K_{m-1}) = \underbrace{(1, 1, ..., 1)}_{\text{m-1}}$.

Vectors $x_2(\overline{H})$ and $e(\overline{H})$ are orthogonal which implies that $\alpha(n-1) + \beta(m-1) = 0$, wherefrom we obtain that α and β are scalars with different signs (*).

$$
\begin{aligned}
& x_2(\overline{H}) = x_n(H) \\
\Rightarrow \quad & x_n(H) = (\underbrace{\alpha, \alpha, \ldots, \alpha}_{\text{n-1}}, \underbrace{\beta, \beta, \ldots, \beta}_{\text{m-1}}) \\
\Rightarrow \quad & x_n(\overline{G}) = (\underbrace{\alpha, \alpha, \ldots, \alpha}_{n-1}, 0, \underbrace{\beta, \beta, \ldots, \beta}_{\text{m-1}}), \\
& \text{because } \overline{G} = HUK_1 \\
\Rightarrow \quad & x_2(G) = z = (\underbrace{\alpha, \alpha, \ldots, \alpha}_{\text{n-1}}, 0, \underbrace{\beta, \beta, \ldots, \beta}_{\text{m-1}})
\end{aligned}
$$

From (*) we conclude that vertices from two blocks of G without v_n, $\{v_1, v_2, ..., v_{n-1}\}$ and $\{v_{n+1}, v_{n+2}, ..., v_{n+m-1}\}$, belong to different sets $N(z)$ and $P(z)$, while v_n is the null vertex.■

References

[1] D. J. Wild, Y. Ding, A. P. Sheth, L. Harland, E. M. Gifford, and M. S. Lajiness. Systems chemical biology and the Semantic Web: what they mean for the future of drug discovery research. In *Drug discovery today*, 17(9):469–474. Elsevier, 2012. DOI: 10.1016/j.drudis.2011.12.019.

[2] A. Schwarte, P. Haase, K. Hose, R. Schenkel, and M. Schmidt. FedX: Optimization Techniques for Federated Query Processing on Linked Data. In L. Aroyo, C. Welty, H. Alani, J. Taylor, A. Bernstein, L. Kagal, N. Noy, and E. Blomqvist, editors, *The Semantic Web - ISWC 2011*, volume 7031 of *Lecture Notes in Computer Science*, pages 601–616. Springer Berlin Heidelberg, 2011. DOI: 10.1007/978-3-642-25073-6_38.

[3] O. Görlitz and S. Staab. SPLENDID: SPARQL Endpoint Federation Exploiting VoID Descriptions. In O. Hartig, A. Harth, and J. F. Sequeda, editors, *2nd International Workshop on Consuming Linked Data, COLD 2011*, in *CEUR Workshop Proceedings*, volume 782, pages 13–24, 2011. ISSN: 1613-0073.

[4] R. Angles, P. Boncz, J. Larriba-Pey, I. Fundulaki, T. Neumann, O. Erling, . . . and I. Toma. The linked data benchmark council: a graph and RDF industry benchmarking effort. *ACM SIGMOD Record*, 43(1):27–31, 2014. DOI: 10.1145/2627692.2627697.

[5] U. Von Luxburg. A tutorial on spectral clustering. In *Statistics and computing*, 17(4):395–416. Springer, 2007. DOI: 10.1007/s11222-007-9033-z.

[6] S. Brin and L. Page. Reprint of: The anatomy of a large-scale hypertextual web search engine. In *Computer networks*, 56(18):3825–3833. Elsevier, 2012. DOI: 10.1016/j.comnet.2012.10.007.

[7] A. Altman and M. Tennenholtz. Ranking systems: The pagerank axioms. In *Proceedings of the 6th ACM Conference on Electronic Commerce, EC '05*, pages 1–8, New York, NY, USA, 2005. ACM. DOI: 10.1145/1064009.1064010.

[8] M. Schmidt, O. Görlitz, P. Haase, G. Ladwig, A. Schwarte, and T. Tran. FedBench: A Benchmark Suite for Federated Semantic Data Query Processing. In L. Aroyo, C. Welty, H. Alani, J. Taylor, A. Bernstein, L. Kagal, N. Noy, and E. Blomqvist, editors, *The Semantic Web - ISWC 2011*, volume 7031 of *Lecture Notes in Computer Science*, pages 585–600. Springer Berlin Heidelberg, 2011. DOI: 10.1007/978-3-642-25073-6_37.

[9] M. Saleem, A.-C. Ngonga Ngomo, J. Xavier Parreira, H. Deus, and M. Hauswirth. DAW: Duplicate-AWare Federated Query Processing over the Web of Data. In H. Alani, L. Kagal, A. Fokoue, P. Groth, C. Biemann, J. Parreira, L. Aroyo, N. Noy, C. Welty, and K. Janowicz, editors, *The Semantic Web - ISWC 2013*, volume 8218 of *Lecture Notes in Computer Science*, pages 574–590. Springer Berlin Heidelberg, 2013. DOI: 10.1007/978-3-642-41335-3_36.

[10] C. Bizer and A. Schultz. The Berlin SPARQL Benchmark. In *International Journal on Semantic Web and Information Systems (IJSWIS)*, volume 5, pages 1–24. IGI Global, 2009. DOI: 10.4018/jswis.2009040101.

[11] J. Umbrich, A. Hogan, A. Polleres, and S. Decker. Improving the recall of live linked data querying through reasoning. In M. Krötzsch, and U. Straccia, editors, *the 6th International Conference on Web Reasoning and Rule Systems, RR*, volume 7497 of *Lecture Notes in Computer Science*, pages 188–204. Springer Berlin Heidelberg, 2012. DOI: 10.1007/978-3-642-33203-6.

[12] O. Görlitz, M. Thimm, and S. Staab. SPLODGE: Systematic Generation of SPARQL Benchmark Queries for Linked Open Data. In P. Cudré-Mauroux, J. Heflin, E. Sirin, T. Tudorache, J. Euzenat, M. Hauswirth, J. Parreira, J. Hendler, G. Schreiber, A. Bernstein, and E. Blomqvist, editors, *The Semantic Web - ISWC 2012*, volume 7649 of *Lecture Notes in Computer Science*, pages 116–132. Springer Berlin Heidelberg, 2012. DOI: 10.1007/978-3-642-35176-1_8.

[13] B. Quilitz, and U. Leser. Querying Distributed RDF Data Sources with SPARQL. In S. Bechhofer, M. Hauswirth, J. Hoffmann, and M. Koubarakis, editors, *The Semantic Web: Research and Applications*, volume 5021 of *Lecture Notes in Computer Science*, pages 524–538. Springer Berlin Heidelberg, 2008. DOI: 10.1007/978-3-540-68234-9_39.

[14] M. Acosta, M.-E. Vidal, T. Lampo, J. Castillo, and E. Ruckhaus. ANAPSID: An Adaptive Query Processing Engine for SPARQL Endpoints. In L. Aroyo, C. Welty, H. Alani, J. Taylor, A. Bernstein, L. Kagal, N. Noy, and E. Blomqvist, editors, *The Semantic Web - ISWC 2011*, volume 7031 of *Lecture Notes in Computer Science*, pages 18–34. Springer Berlin Heidelberg, 2011. DOI: 10.1007/978-3-642-25073-6_2.

[15] M. Saleem, Y. Khan, A. Hasnain, I. Ermilov, and A.-C.N. Ngomo. A fine-grained evaluation of SPARQL endpoint federation systems. *Semantic Web Journal*, 7(5):493–518, 2014. DOI: 10.3233/sw-150186.

[16] M. Saleem, Q. Mehmood, and A.-C. N. Ngomo. Feasible: A feature-based sparql benchmark generation framework. In M. Arenas et al., editors , *The Semantic Web - ISWC 2015*, volume 9366 of *Lecture Notes in Computer Science*, pages 52–69, Springer, Cham, 2015. DOI: 10.1007/978-3-319-25007-6_4.

[17] M. Saleem, C. Stadler, Q. Mehmood, J. Lehmann, and A.-C. N. Ngomo. SQCFramework: SPARQL Query Containment Benchmark Generation Framework. In *Proceedings of the Knowledge Capture Conference*, p. 28, ACM, 2017. DOI: 10.1145/3148011.3148017.

[18] H. Stuckenschmidt, R. Vdovjak, G. J. Houben, and J. Broekstra. Index structures and algorithms for querying distributed RDF repositories. In *Proceedings of the 13th international conference on World Wide Web*, pages 631–639, ACM, 2004. DOI: 10.1145/988672.988758.

[19] J. Broekstra, A. Kampman, and F. van Harmelen. Sesame: A generic architecture for storing and querying RDF and RDF schema. In I. Horrocks, and J. Hendler, editors, *The Semantic Web - ISWC 2002*, volume 2342 of *Lecture Notes in Computer Science*, pages 54–68, 2002. DOI: 10.1007/3-540-48005-6_7.

[20] H. Dietze and M. Schroeder. GoWeb: a semantic search engine for the life science web. *BMC Bioinformatics*, 10(10):S7, 2009. DOI: 10.1186/1471-2105-10-s10-s7.

[21] D. Schweiger, Z. Trajanoski, and S. Pabinger. SPARQLGraph: a web-based platform for graphically querying biological semantic web databases. *BMC Bioinformatics*, 15(1):279, 2014. DOI: 10.1186/1471-2105-15-279.

[22] A. De Leon Battista , N. Villanueva-Rosales, M. Palenychka, and M. Dumontier. Smart: a web-based, ontology-driven, semantic web query answering application. *Semantic Web Challenge at the International Semantic Web Conference 2007*, volume 295, pages 129–36, 2007.

[23] M. J. García-Godoy, I. Navas-Delgado, and J. Aldana-Montes. Bioqueries: a social community sharing experiences while querying biological linked data. In *Proceedings of the 4th International Workshop on Semantic Web Applications and Tools for the Life Sciences*, pages 24–31, 2011. ACM. DOI: 10.1145/2166896.2166906.

[24] W. Hu, H. Qiu, J. Huang, and M. Dumontier. BioSearch: a semantic search engine for Bio2RDF. *Database*, 2017. DOI: 10.1093/database/bax059.

[25] M. Djokic-Petrovic, V. Cvjetkovic, J. Yang, M. Zivanovic, and D. J. Wild. PIBAS FedSPARQL: a web-based platform for integration and exploration of bioinformatics datasets. *Journal of Biomedical Semantics*, 8(1):42, 2017. DOI: 10.1186/s13326-017-0151-z.

[26] F. Belleau, M. A. Nolin, N. Tourigny, P. Rigault, and J. Morissette. Bio2RDF: Towards a mashup to build bioinformatics knowledge systems. In *Journal of Biomedical Informatics*, 41(5):706–716. Elsevier, 2008. DOI: 10.1016/j.jbi.2008.03.004.

[27] A. Jentzsch, M. Samwald and B. Andersson. Linking Open Drug Data. In A. Paschke, H. Weigand, W. Behrendt, K. Tochtermann, and T. Pellegrini, editors, *Proceedings of the International Conference on Semantic Systems*, I-SEMANTICS'09, pages 3–6, 2009.

[28] B. Chen, X. Dong, D. Jiao, H. Wang, Q. Zhu, Y. Ding, and D.J. Wild. Chem2Bio2RDF: a semantic framework for linking and data mining chemogenomic and systems chemical biology data. In *BMC bioinformatics*, 11(1):1–13. Springer, 2010. DOI: 10.1186/1471-2105-11-255.

[29] S. Jupp, J. Malone, J. Bolleman, M. Brandizi, M. Davies, L. Garcia, A. Gaulton, S. Gehant, C. Laibe, N. Redaschi, S. M. Wimalaratne, M. Martin, N. Le Novère, H. Parkinson, E. Birney, and A. M. Jenkinson. The EBI RDF platform: linked open data for the life sciences. In *Bioinformatics*, 30(9):1338–1339. Oxford University Press, 2014. DOI: 10.1093/bioinformatics/btt765.

[30] A. J. Williams, L. Harland, P. Groth, S. Pettifer, C. Chichester, E. L. Willighagen, C. T. Evelo, N. Blomberg, G. Ecker, C. Goble, and B. Mons. Open PHACTS: semantic interoperability for drug discovery. *Drug Discovovery Today*, volume 17, pages 1188–1198, 2012. DOI: 10.1016/j.drudis.2012.05.016.

[31] H. E. Pence and A. Williams. ChemSpider: an online chemical information resource. *Journal of Chemical Education*, volume 87, pages 1123–1124, 2010. DOI: 10.1021/ed100697w.

[32] A. J. G. Gray, P. Groth, A. Loizou, S. Askjaer, C. Brenninkmeijer, K. Burger, C. Chichester, C. T. Evelo, C. Goble, L. Harland, S. Pettifer, M. Thompson, A. Waagmeester, and A. J. William. Applying linked data approaches to pharmacology: Architectural decisions and implementation. *Semantic Web Journal*, 5(2):101–113, 2014. DOI: 10.3233/SW-2012-0088.

[33] V. Cvjetković, M. Đokić, B. Arsić, and M. Ćurčić. The ontology supported intelligent system for experiment search in the scientific research center. In *Kragujevac Journal of Science*, volume 36, pages 95–110, 2014. DOI: 10.5937/kgjsci1436095c.

[34] B. Arsić, M. Đokić, V. Cvjetković, P. Spalević, M. Živanović, and M. Mladenović. Integration of bioactive sub-stances data for preclinical testing with Cheminformatics and Bioinformatics resources. In *Proceedings of 23nd International Electrotechnical and Computer Science Conference, ERK 2014*, pages 146–149, Ljubljana, Slovenia, 2014. IEEE. ISSN: 1581-4572.

[35] Y. Wang, J. Xiao, T. O. Suzek, J. Zhang, J. Wang, and S. H. Bryant. PubChem: a public information system for analyzing bioactivities of small molecules. *Nucleic Acids Research*, 37(2):623–633, 2009. DOI: 10.1093/nar/gkp456.

[36] C. Knox, V. Law, T. Jewison, P. Liu, S. Ly, A. Frolkis, A. Pon, K. Banco, C. Mak, V. Neveu, Y. Djoumbou, R. Eisner, A.C. Guo, and D.S. Wishart. DrugBank 3.0: a comprehensive resource for 'omics' research on drugs. *Nucleic Acids Research*, 39(suppl_1):D1035–D1041, 2011. DOI: 10.1093/nar/gkq1126.

[37] P. De Matos, R. Alcántara, A. Dekker, M. Ennis, J. Hastings, K. Haug, I. Spiteri, S. Turner, and C. Steinbeck. Chemical entities of biological interest: an update. *Nucleic Acids Research*, volume 38, pages 49–54, 2009. DOI: 10.1093/nar/gkp886.

[38] M. Kanehisa, S. Goto, Y. Sato, M. Furumichi, and M. Tanabe. Kegg for integration and interpretation of large-scale molecular datasets. *Nucleic Acids Researc*h, volume 40, pages D109–114, 2012. DOI: 10.1093/nar/gkr988.

[39] A. P. Bento, A. Gaulton, A. Hersey, L. J. Bellis, J. Chambers, M. Davies, . . . , and M. Nowotka. The ChEMBL bioactivity database: an update. *Nucleic Acids Research*, volume 42, pages D1083–D1090, 2014. DOI: 10.1093/nar/gkt1031.

[40] A. Bergamo and G. Sava. Linking the future of anticancer metal-complexes to the therapy of tumour metastases. *Chemical Society Reviews*, 44(24):8818–8835, 2015. DOI: 10.1039/c5cs00134j.

[41] V. P. Petrović, M. N. Živanović, D. Simijonović, J. Đorović, Z. D. Petrović, and S. D. Marković. Chelate N,O-palladium(II) complexes: synthesis, characterization and biological activity. *RSC Advances*, volume 5, pages 86274–86281, 2015. DOI: 10.1039/c5ra10204a.

[42] P. Canovic, J. Bogojeski, J. Kosaric, S. Markovic, M. Zivanovic. Pt (IV), Pd (II), and Rh (III) complexes induced oxidative stress and cytotoxicity in the HCT-116 colon cancer cell line. *Turkish Journal of Biology*, 41(1):141–147, 2017. DOI: 10.3906/biy-1605-77.

[43] M. N. Živanović, J. V. Košarić, B. Šmit, D. S. Šeklic, R. Z. Pavlović, and S. D. Marković. Novel seleno-hydantoin palladium (II) complex-antimigratory, cytotoxic and prooxidative potential on human colon HCT-116 and breast MDA-MB-231 cancer cells. *General physiology and biophysics*, 36(2):187–196, 2017. DOI: 10.4149/gpb_2016036.

[44] S. J. Coles, N. E. Day, P. Murray-Rust, H. S. Rzepa, and Y. Zhang. Enhancement of the chemical semantic web through the use of InChI identifiers. *Organic & biomolecular chemistry*, 3(10):1832–1834, 2005. DOI: 10.1039/b502828k.

[45] J. Chambers, M. Davies, A. Gaulton, A. Hersey, S. Velankar, R. Petryszak, J. Hastings, L. Bellis, S. McGlinchey, and J. P. Overington. UniChem: a unified chemical structure cross-referencing and identifier tracking system. *Journal of Cheminformatics*, 5(1):3, 2013. DOI: 10.1186/1758-2946-5-3.

[46] J. R. Lewis, and J. Sauro. The factor structure of the system usability scale. In M. Kurosu, editors, *International Conference on Human Centered Design*, volume 5619 of *Lecture Notes in Computer Science*, pages 94–103. Springer, Berlin, Heidelberg, 2009. DOI: 10.1007/978-3-642-02806-9_12.

[47] G. Fu, C. Batchelor, M. Dumontier, J.Hastings, E. Willighagen, and E. Bolton. PubChemRDF: towards the semantic annotation of PubChem compound and substance databases. *Journal of Cheminformatics*, 7(1):34, 2015. DOI: 10.1186/s13321-015-0084-4.

[48] M. Fiedler. Algebraic connectivity of graphs. In *Czechoslovak mathematical journal*, 23(2):298–305, 1973.

[49] M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. In *Czechoslovak Mathematical Journal*, 25(4):619–633, 1975.

[50] D. Cvetković, P. Rowlinson, and S. Simić. An Introduction to the Theory of Graph Spectra. *London Mathematical Society Student Texts*. Cambridge University Press, 2010. ISBN: 978-0-521-11839-2. DOI: 10.1017/CBO9780511801518.