Semantic Web 1 (2016) 1–5
IOS Press

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

Editor(s): Name Surname, University, Country
Solicited review(s): Name Surname, University, Country
Open review(s): Name Surname, University, Country

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

[a] *University of Kragujevac, Faculty of Science, Radoja Domanovića 12, 34000 Kragujevac, Serbia*
E-mail: brankoarsic@kg.ac.rs, m.djokic@kg.ac.rs, zivanovicm@kg.ac.rs
[b] *University of Priština, Faculty of Technical Sciences, Knjaza Miloša 7, 38220 Kosovska Mitrovica, Serbia*
E-mail: petar.spalevic@pr.ac.rs
[c] *University of Niš, Faculty of Electronic Engineering, Aleksandra Medvedeva 14, 18000 Niš, Serbia*
E-mail: ivan.milentijevic@elfak.ni.ac.rs, dejan.rancic@elfak.ni.ac.rs

Abstract. Many research centers and medical institutions have been accumulating the huge amount of various biological and chemical data over the past decade and this trend continues. Their associated information models, notions, areas of interest, units of measurement, parameters and conditions for experiments are different. Based on Linked Data vision, many semantic applications for distributed access to these heterogeneous RDF (Resource Description Framework) data sources were developed. Their improvements brought about a decrease of intermediate results and an optimizing query execution plans. But still many requests are unsuccessful and they time out without producing any answer. Also, the queries over different repositories with many data sources 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. Innovativeness of the approach lays in the fact that the coordinates of graph eigenvectors are used for query join-ordering and translation of directed graph into federated SPARQL queries, instead of data statistics and classical algorithms which are applicable to the weighted graphs. Chemists and biologists could gain large benefit with the SpecINT by creating virtually distributed database as a resource for gaining new knowledge about chemical substances and compounds and their natural influence on environment. In experiments, we demonstrate the potential of our framework on a set of heterogeneous and distributed cheminformatics and bioinformatics data sources.

Keywords: federated query, data integration, matrix eigenvectors

1. Introduction

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

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

tories. For the successful performance of a biomedical research, data integration grows into an important precondition for overcoming the existing gaps and resource and time saver. In [1] the authors indicated the importance of data integration in cheminformatics and bioinformatics. Today, there are several semantic based repositories (initiatives) for biological and chemical data sources integration: Bio2RDF [2], LODD [3], Chem2Bio2RDF [4], EMBL-EBI [5] and ChemSpider.

If the researchers are not familiar with large sets of data published, they need to discover all possible interlinks for SPARQL queries construction. In many repositories, no simple rules and conventions can be followed. Making an effort and investing a lot of time to discover each connection, their orientation between data sources and specific properties over different data sources are required. The real-time solutions, which don't burden the scientists, hide its complexity from the user and offer tangible results, are desirable.

For efficient query processing in semantic-oriented environments, sophisticated query generators and benchmarking systems for their performance evaluation are developed. Drawbacks of benchmarking systems arise from the fact that they rely on a set of predefined static queries over particular data sources [6][7]. To address this problem, SPLODGE [8] and LidaQ [9] generated random query sets based on three main shapes for federated queries benchmark. DAW [10] statically generated simple queries from four public datasets, while QFed [11] produced federated SPARQL queries taking into account the characteristics of both datasets and queries. However, there are still many problems following their approaches. Firstly, a process of setting parameters for an algorithm and thresholds can be difficult without prior knowledge about the data. This lead to collecting different statistics which are changeable over time. Secondly, in such piles of generated queries, many of them are without answer, many of them return unnecessary data. At the same time, the processes of seeking the most promised 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. However, this is necessary because no repository can gather all relevant data from the web. Very often repositories integration is not possible because there are no predicate ties between them. Additional aggravating circumstances are completely different queries for a data source within different repositories.

Problem of integrating the data from multiple data sources and repositories is still a challenge. Handcrafted 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 human role in creating hand-crafted subqueries as very important guide for satisfactory results, whilst their connection is performed automatically, seeking the path over data sources in different repositories. For path finding the eigenvectors are preferred when the weights of the vertices are known, instead of classical algorithms which are applicable to the weighted graphs. The project contributors found that these paths lead to the best decision-making, rather than exploring every single triple in repositories. These edges suggest an aggregation of the most relevant data that is why only the connected data sources are considered. In contrast to the state-of-art federated query engines, our solution changes data statistics with graph based clustering algorithm [12] and vertices ranking [13][14] for source selection and join ordering process in queries. The SpecINT[1] is a support framework developed to 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 SpecINT has the following contributions:

- *Advancement:* A SPARQL query framework based on the concept of mathematical graph is developed - the graph eigenvectors are used for query source selection, ranking and path joining.
- *Scalability:* A straightforward model for linking data from repositories is proposed.
- *Federation:* Federated SPARQL queries gather novel and complementary data about substances in real time. The constant statistical calculations and update monitoring are avoided.
- *Availability:* Our data are made available to the entire research community.

The lack of information about the endpoints availability and limits, makes that any query is not completely applicable in the context of federations of endpoints. Because of that the results could sometimes be

[1] http://147.91.203.161/specint

[2] CPCTAS-LCMB, Faculty of Science, University of Kragujevac, Serbia, http://cpctas-lcmb.pmf.kg.ac.rs

incomplete. Also, among the results duplicates may be present, because one data source could be stored in many repositories with different prefixes and that is beyond the scope of this paper. The current version of the framework is specialized for life sciences, but 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 literature of significance for the study area. The third section is devoted to a novel data source integration reflecting a framework scalability. The fourth section describes architecture and functioning principles of the proposed system for integration and query federation. The fifth and sixth sections discuss the results, benefits and limitations of the framework. The paper concludes with a wrap up of key points and directions for further work.

2. Related work

In this section, we provide an overview of existing query generators developed for grained evaluation of federated SPARQL query engines. A review of their strategies is made and some drawbacks, in the cases where real time and validity of results are crucial, are identified. However, automatic query generation is less tedious and can produce many queries with specific characteristics, even for varying data sources as in the Linked Data cloud. In contrast, meaningful and real queries can only be generated manually or semi-automatically putting a lot of effort into it because the content of the data sources needs to be analyzed in advance. The need for the real-time queries is completely understandable because automatically generated queries are used only for query execution evaluation, not for end-users and their demands. The following federation systems are developed for optimizing the query runtime and as such can not be used for satisfactory user experience with relevant final results.

FedX [6] has been developed for comparing general purpose of SPARQL query federation systems. It focuses on strategies which can decrease the number of query transmission 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 [15] is the only benchmark proposed for federated query which evaluated the federated query infrastructure performance including loading time and querying time. However, the FedBench has static dataset and query set, too. DAW [10] provides a set of static queries based on characteristics of BSBM (Berlin SPARQL Benchmark) queries [16] from four public datasets. 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 random query set for specified dataset. Umbrich et al. [9] study 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. The query complexity, using either star-shaped or path-shaped join patterns, is limited to a maximum of three joins. This query generator produces sets of similar queries by doing random walks of certain breadth or depth. The query set generation of SPLODGE [8] is based on dataset 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 [17] and SPLENDID [7] 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 [18] reflected in updating data catalogue and execution plan at runtime. These approaches need constant predicates refill and statistical calculation restart which cannot be overcome. Some algorithms rely on a very small amount of statistical information and some of them use different thresholds for the source or predicate selection. All this makes them not appropriate for immediate use. The most comprehensive survey and evaluation of some of the listed federation systems was given in [19].

However, federated queries construction over repositories is not a trivial task, and it often ends with a deadlock. Different vocabularies and nonexistent patterns bring headache on researchers. These query generators produce different queries. Hence, in our evaluation, we will not compare the performance of those query generators in queries production. To our best knowledge there is no system that supports SPARQL query federation for multiple regular SPARQL endpoints over different data sources belonging to different repositories with the same topic.

3. New data integration

To complete the background for SpecINT we devote this section to new data publishing. A new data source causes no changes in the system, justifying the system's scalability. In order to make data widely available, data should be linked to the other data sources by entity matching. According to LOD cloud statistics[3] almost all data sources have more than a thousand links to other data sources. Mapping process is time consuming, and the same data source has different predicates within different repositories. For example, the predicates for the substance targets are different (http://bio2rdf.org/drugbank_vocabulary:target; and http://chem2bio2rdf.org/drugbank/resource/CID-_GENE), which automatically means that the queries are different. The process of discovering new substances and compounds, and also the influence they have on cancer cell-lines, genes and proteins, creating a coherent unity of the results and their filtering for complementary data are impeccable reasons for data integration over data repositories of interest. The process itself is imminently problematic.

Aiming to meet the principles of Linked Data[4] and make data available to wide research community, the necessary precondition is data transformation into Semantic Web context. In order to support CPCTAS laboratory staff to quickly reference and use complex experiment structure, PIBAS (Preclinical Investigation of Bioactive Substances) ontology for modeling complex experimental structure was developed and presented in [20,21]. In order to avoid the problem of predicates resolving, there should be no dependency on a single data source, because a substance can be in one repository and not in the other. Our substances are mapped to entities related to the compounds and substances from other data sources by using its identification number (cid). This approach provides flexibility for other similar laboratories. For simplicity's sake, in performed experiments target data sources are limited to four most prevalent ones over repositories: PubChem, DrugBank, ChEBI and KEGG. This list could be extended, if necessary. Listing 1 represents 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 and CHEMBL maps demonstrate an easy usage.

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

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

```
<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. SpecINT architecture

Having in mind previously set background, let us explain the functioning of the system and what happens in the background, hidden from the researchers. An end-user sends InChIKey to the application and starts complex process with tendency to obtain relevant final results through a SPARQL query. The architecture of the framework is shown in Figure 1 and is explained in the following subsections. The constant expansion of new data sources brings about problems in analyses of the disconnected and heterogeneous data, which are crucial for future successful and purposeful surveys. Thanks to Semantic Web standards and online data exploring through open endpoints, it is possible to search these data sources in a single SPARQL query. In the following subsections, complete federated query processing model is described in more details going through a complete example. An example describes a situation when the novel data sources are integrated. If a data source is disintegrated, the framework could continue to operate without changes, thus reflecting scalability.

4.1. Source selection

Query generator should carefully determine the relevant source of subquery, because wrong choice either leads to expensive communication with many intermediate results being memorized or the system fails to contribute any results. A complete list of different source selection techniques could be seen in the technical report of Rakhmawati et al. [22]. These techniques cannot guarantee the inclusion of all subjects of interest.

In order to integrate two data sources, the substance found in one data source is required to be identi-

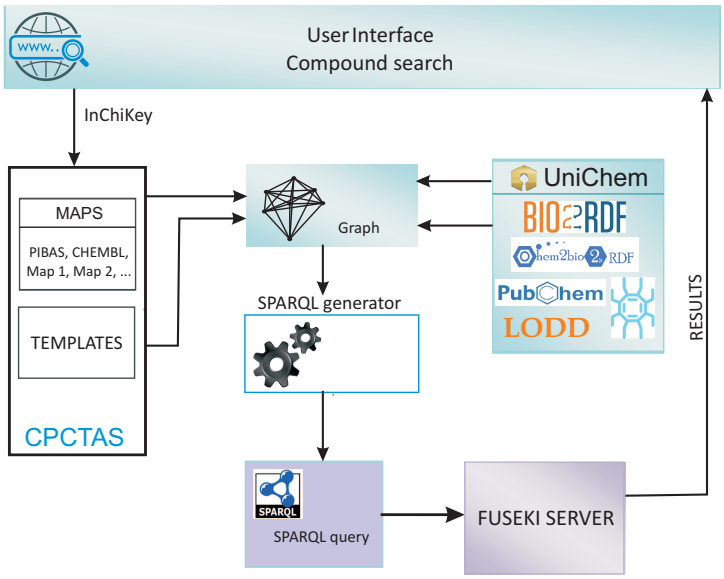

Fig. 1. SpecINT architecture

fied in other data source. In most cases, the corresponding predicate does not exist, and new alternative paths made of predicates should be found. Process of creating mapping ontology for every substance noted in data sources is a Sisyphean task. To achieve this goal, the observed chemical substance is transformed into the corresponding InChIKey identifier. Then, we use the UniChem 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 molecule based on adopted and stable standards, InChIs and InChIKeys. To be more precise, the synonyms represent the labels of substance from different data sources. For InChIKey = GUGOEEXESWIERI-UHFFFAOYSA-N, some of the returned synonyms from API are: CHEMBL17157, kegg_ligand C07463, drugbank DB00342, chebi9453, SCHEMBL5152 etc.

Many data sources included within different repositories are not involved in UniChem. With desire to encompass as more data sources as possible, for every single UniChem substance obtained, related substance synonyms from one repository are found. This repository could have been Chem2Bio2RDF, Bio2RDF, LODD, etc. Than, all the substance synonyms from UniChem and selected repository are obtained.

4.2. *Graph construction*

Taking into account that SPARQL query originates from directed graph, it is necessary to construct a graph from selected sources which should be used as a path between data sources for new queries beforehand. This procedure involves two basic steps.

A. *Undirected graph construction*. This step reveals our hidden intention to save the information about vertices affiliation and connecting vertices between two systems, because the next graph perturbations and edge removal would mix up known affiliation. The repositories affiliations are very important, because the same data source could have different predicates and interlinks orientation.

All labels, found in previous step, form a complete graph K_n, because every label presents the same substance in different data sources. In this way substance can be connected to all its representations in data sources within selected repository. The previous steps for graph construction are repeat for the another repository. This brings us to two complete graphs, K_n and K_m. Following the background idea of saving information for one common vertex which is a nexus between two repositories, coalescence $V_{n,m}$ between two obtained graphs can be formed with any label from repositories intersection (see Figure 2). According to the results of Theorem 1 (see Appendix A) we can calculate Fiedler eigenvector s (sign) for graph $V_{n,m}$ and

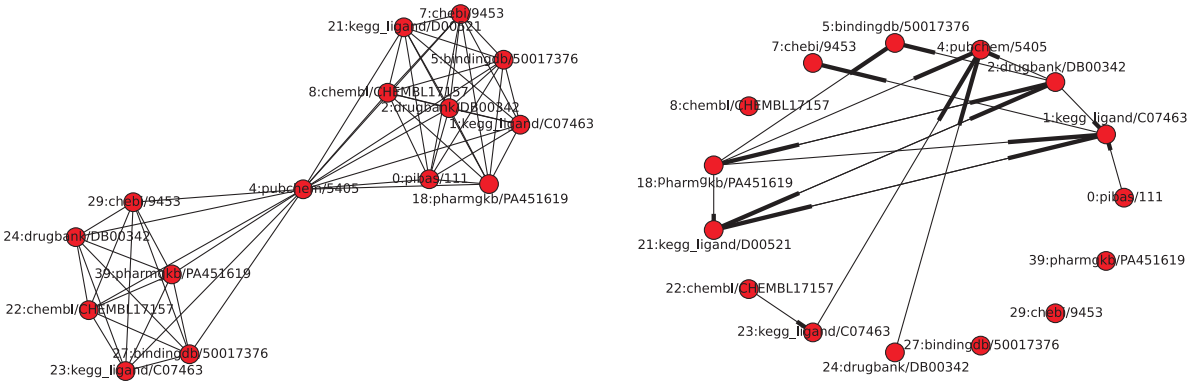

Fig. 2. Graph coalescence between Chem2Bio2RDF and Bio2RDF repositories.

divide vertices set into two disjoint sets, with positive and negative vertices, and one *null* vertex.

B. *Directed graph construction*. The next step is creating the best sequence of the labels for valid results retrieval. Prior to query generation, a processor has to check the existence and orientation of the edges. Then it should transform undirected edges into directed ones according to the SPARQL query nature. 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 triples (?source ?property ?target) we can convert graph $V_{n,m}$ into digraph $D_{n,m}$. This step includes removing all the nonexistent edges, but not the isolated vertices. Previously, with Fiedler eigenvector we paved the way for converting process. In case of disconnected graph, we repeat the whole procedure for unused label. Also, the self-loops and fictitious vertex for source favor are used. This is especially important in case when a novel source is integrated or the specific answers are preferred.

It was known that the importance of each vertex is proportional to the sum of the importances of all of the vertices that link to it. Simple calculation says that this is an eigenvalue and eigenvector problem (more details can be found in [13]). 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 one with the largest entry in that eigenvector, the next most important has the second largest entry, and so forth. Now, we follow the most probable path over repositories as search engines do.

All steps are presented in Procedure 1.

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 Calculate eigenvector r;
10 **if** $D_{n,m}$ *is disconnected* **then**
11 goto step 6 and try unused *Intersection* label;
12 **end**

4.3. Join ordering and building queries

This subsection is dedicated to an algorithm for creating SPARQL queries from two obtained vectors and templates. After a digraph $D_{n,m}$ with all existing edges is obtained, we have to decide how to align selected sources in a query in order to ensure results over repositories. This phase takes a selecting of prefixes related to the data sources of vertices and corresponding templates into account. A background idea is to use the highest ranked vertices for finding the best path to the central vertex, from both sides, positive and negative. The entrances 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 co-

Table 1
Data source patterns within repositories.

Data source	Pattern for 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.

ordinates, and one connecting, *null* vertex. This eigenvector carries vertices affiliation and after graph transformations these signs carry vertices origin. The *null*-vertex presents an articulation point. The second vector r represents the most important vertices in both connected components. Coordinate values in the second eigenvector actually suggests most probable paths to the *null*-vertex within sign zones providing repositories link-up (see Figure 2).

For simplicity's sake, let suppose that the first connected component contains vertices with positions $0, 1, \ldots, n-2$, for cut-vertex is $n-1$, and the second connected component with $n, n+1, ..., n+m-2$ positions. 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 nodes with the same rank is present, path construction will diverge simultaneously for each vertex. Once created path over repositories means different information availability for a substance. The isolated vertices are skipped deliberately because the no edges existence indicates different purpose of vertex or substance non-existence. For every vertex in the path we use specific templates for sub-query completeness (see examples in Table 1). Edges between these vertices (from digraph $D_{n,m}$) are used for the templates chaining, in a such way that an object from a template becomes a subject in the following template. 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 $D_{n,m}$. Templates are collected from initiative examples and parts of them are hand-crafted. In this way we can connect our substance with the same substances over different repositories.

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)
  $add\_subquery(subject, object, R_1, template)$
  $subject \leftarrow object$
until $object = null\_vertex$;
$add\_subquery(subject, null\_vertex, R_1 \; or \; R_2, template)$
$subject \leftarrow label(i)$, i - the best ranked negative vertex
repeat
  $neighbors \leftarrow$ negative neighbors for $subject$
  $object \leftarrow$ the best ranked neighbor
  $add\_subquery(subject, object, R_2, template)$
  $subject \leftarrow object$
until $object = null\_vertex$;
return $query$

For the graphs from Figure 2 two eigenvectors are calculated: Fiedler eigenvector s and rank eigenvector r. 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: $0 \rightarrow 1 \rightarrow 21 \rightarrow 2 \rightarrow 4$. The second path over negative vertices belonging to Chem2Bio2RDF initiative is: $22 \rightarrow 23 \rightarrow 4$. We allocated Bio2RDF to positive side, and Chem2Bio2RDF to negative side, but the

```
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 mapping_node .
          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

same process can be revolved to obtain a little bit different query. Finally, by following these paths and vertex templates we construct SPARQL query (see Listing 2) which retrieves *targets* for initial substance. For more examples visit website[5].

5. Evaluation

In this section we give an evaluation of our generated queries and current issues. Evaluation in this context basically means checking if the generated queries meet our primary goals, i. e. if they can actually retrieve relevant results from different data sources (and repositories) and whether these results mean an improvement and support for laboratory work. Task for evaluation was assigned to the chemists and biologists employed at Centre for Preclinical Testing of Active Substances (CPCTAS), Faculty of Science, University of Kragujevac. In order to meet the CPCTAS needs and the expectations of framework, the first task is to find a way to join two paths in a cut-vertex, from both sides, positive and negative. The cut-vertex serves as a mediator from repository to repository crossing. The second

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

task is to select the most relevant data sources from both repositories. The state-of-art algorithms for path finding are not applicable in a case when the vertices favor are necessary. One of the idea should be increasing the weight of edges between the data sources, but it cannot be done when the edges differ from substance to substance.

For collection of all relevant data from repositories three approaches for path selection are tested. The same principle for initial vertex and the following vertices setting is used. The following vertices are selected between neighbors according to:

- **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.

Tests were originally performed on 50 substances/-compounds. The number of relevant data sources included for every approach could be seen in Figure 3. The substances are sorted in ascending order by the average results for every approach. In very rare cases *Degree* approach can join the paths over repositories. Even when it succeeds in a full-path construction, this path contains a small number of relevant sources. Degree approach has not proved as good in practise because of branching. It considers execution and evaluation of several queries resulting in slowing down experimental work. The *PageRank* approach gives much better results in most cases including a higher percentage of success in paths joining. In exceptional cases, when the paths joining is impossible, the change of cut-vertex is necessary. These good results are expected because the best ranked vertices are connected with many data sources and it is very easy to perform merging. The main drawback of this approach comes from the fact that novel data sources could not be involved in the path because of the small rank. For example, a substance from CPCTAS is connected with only one substance and as such is worthless in comparison with the ”strong” data sources. Also, this approach has not convinced us that the paths include all data sources of interest. With *Favored PageRank* the forced vertex selection is possible. For this purpose a new fictitious vertex with large rank is created. The edges from this vertex to the target vertices increase their rank without violating the existing graph structure. Such a case is presented through the example where the PIBAS and CHEMBL vertices were favored. For clarity's sake the

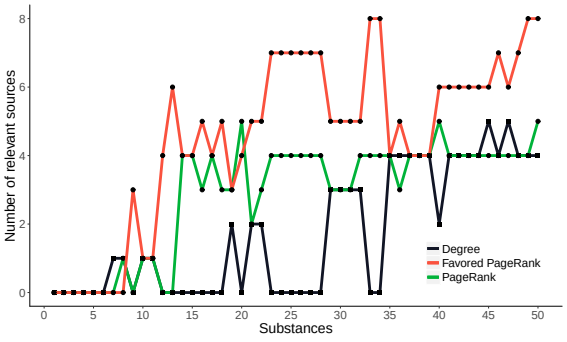

Fig. 3. The number of relevant data sources for every substance.

fictitious vertex is removed from the figure. Similarly, for the vertices of interest that might be encountered on the path, self-loops are used in accordance with previous knowledge about data sources. In this way the framework dependance on the substances intermapping is avoided. The following subsection shows the framework at work.

5.1. The SpecINT in use

CPCTAS possess 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 SpecINT framework provides information about physical and chemical properties of the substance, substance interaction with various protein targets, substance cytotoxicity on various cell-lines and so on. In order to evaluate the benefits of new tool, the researchers started the framework for different substances from CPCTAS laboratory, originally synthesized for new experiments. Therefore, we generated query sets with different purpose and, finally, queries were executed to obtain complementary results.

Suppose that we want to check all tested targets for compound ”Sertindole” with InChIKey=GZKLJWGUPQBVJQ-UHFFFAOYSA-N. For this substance Bio2RDF/DrugBank returns 8, Chem2Bio2RDF/DrugBank endpoint returns 6 and CHEMBL endpoint returns 38 targets, simultaneously. The SpecINT returns the union of these targets including the targets from integrated data source. Table 2 lists one part of InChIKeys with their molecular formulas and short names. For every substance *Id*, Table 3 presents 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 was tested and where additional information could be

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

Table 3

Obtained results for previously listed substances. For every substance are presented the numbers of items for targets, cell-lines and IC50 values, found in data sources within different repositories

	Bio2RDF			**Chem2Bio2RDF**			**CHEMBL**			**CPCTAS**		
Id	*Target*	*CL*	*IC50*	*Target*	*CL*	*IC50*	*Target*	*CL*	*IC50*	*Target*	*CL*	*IC50*
1.	4	0	0	0	0	0	27	0	0	1	1	1
2.	1	0	0	0	0	0	128	0	0	1	1	1
3.	1	0	0	0	0	0	123	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	189	9	23	1	1	1
7.	3	0	0	0	0	0	152	2	2	1	1	1
8.	10	0	0	2	0	0	162	24	27	1	1	1

found. This principle of exploring data is used in the Center, and some new experiments were performed. Novel results were published in [26][27].

5.2. Usability and Usefulness

To assess the usability of our system, we used the nine-item Likert scale-based System Usability (SUS) questionnaire. 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 survey are shown in Figure 4. 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 $1 = 4.1 \pm 0.91$). This information is supported by the fact, that our system represents a great starting point for finding novel substances used for experiments, which are published in []. The responses to question 2 (I think the system was easy to use) implie 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 $3 = 4.5 \pm 0.76$) the 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) suggests that our system has positive feedback on users (average score to question $4 = 4,3 \pm 0,73$). The responses to question 5 (I think that I would like to use this system frequently) suggests that our system has positive effects on the users (average score to question $5 = 4.0 \pm 0.73$). This has encouraged us and we will continue to listen to the demands of users and try to tailor the system to their needs. The responses to question 6 (I think that system gives me a complementary data) indicates that our system was supportive in searching for complementary data, that would be used for feature QSAR analysis (average score to question $6 = 3.9 \pm 0.91$). The responses to question 7 (I would like that system supports more than two initiatives) suggests that users discover our system as positive to their needs and that adding of new initiatives would be only a plus (average score to question

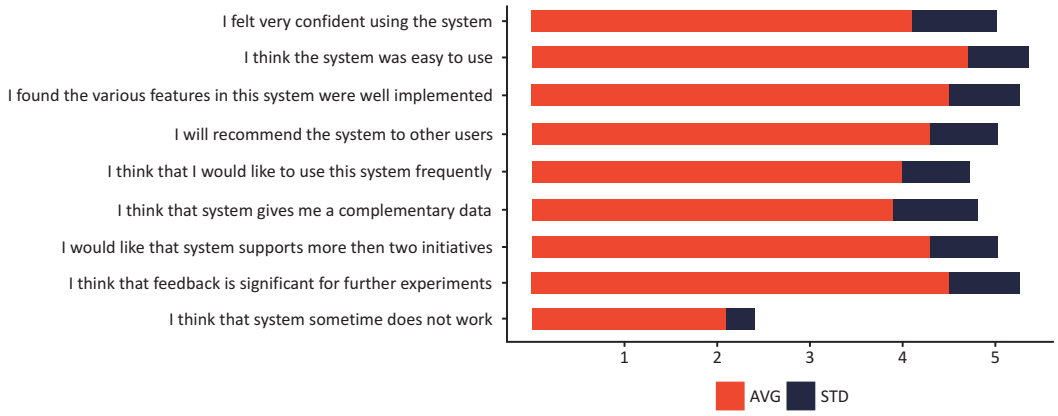

Fig. 4. Result of usefulness evaluation by using our custom questionnaire.

$7 = 4,3 \pm 0,73$). The responses to question 8 (I think that feedback is significant for further experiments) suggests that our system can return essential and important information for further experimental analysis (average score to question $8 = 4.5 \pm 0.76$). The responses to question 9 (I think that system sometimes does not work) indicate that our system can sometimes be stuck (average score to question $9 = 2,1 \pm 0,3$). The score of this question could be justified by the fact that endpoints are sometimes not reachable. Generally, we achieved an average score of 4.04. The data indicate that the overall impression was positive and encouraging, and we found the SpecINT to be very useful.

6. Limitations and possible solutions

In this section we cover a number of known limitations of our framework, and provide possible solutions for future work.

Endpoint is down: the SpecINT depends on the availability of the used SPARQL endpoints. A local copy of the endpoints can not 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.

A cut-vertex choice: choosing 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 two tasks is not easy. Although the differences in results are very small, future work should be focused on obtaining maximum performance. However, this implementation demands novel preprocessing steps and longer execution time.

Two repositories: the main drawback of the current version of 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.

7. Conclusion and future work

Relevant pre-selection information could transform experimental research into a more efficient data processing for the objects of interest. In our paper, we presented a new approach for information retrieval laying behind the SpecINT framework. The solution itself was foreseen to become a milestone in CPCTAS laboratory work. Additionally, it enables scientists to find information of interest on the web, and it also encourages other laboratories to publish data thus extending an 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 most important contribution is SPARQL queries construction in scalable manner according to existing paths in a generated graph. Simultaneously, a simple vertex ranking method based on eigenvectors of generated graph and digraph is introduced. Our approach is not dependent on constant update monitoring, therefore the expensive statistical calculations are avoided. Time spent in finding relevant data is reduced with matching done in real time. Framework is flexible and

adjustable for new area adaptation. For future work, it would be interesting to study the weighted graphs obtained from RDF data, the effect of changing edges and vertices weighted functions for path generation and eigenvectors coordinate changes. The eigenvectors present the excellent mathematical apparatus for additional speed improvements in the SPARQL query. New progress could also be achieved by extending the framework with API-based repositories such as ChemSpider.

Acknowledgments

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).

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*, volume 17(9), pages 469–474. Elsevier, 2012.

[2] F. Belleau, M. A. Nolin, N. Tourigny, P. Rigault, and J. Morissette. Bio2RDF: Towards a mashup to build bioinfor-matics knowledge systems. In *Journal of Biomedical Informatics*, volume 41(5), pages 706–716. Elsevier, 2008.

[3] A. Jentzsch, M. Samwald and B. Andersson. Linking Open Drug Data. In *Proceedings of the International Conference on Semantic Systems*, I-SEMANTICS'09, pages 3–6, 2009.

[4] 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*, volume 11(1), pages 1–13. Springer, 2010.

[5] 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, volume 30(9), pages 1338–1339. Oxford University Press, 2014.

[6] 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 601Ű616. Springer Berlin Heidelberg, 2011.

[7] 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.

[8] 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.

[9] J. Umbrich, A. Hogan, A. Polleres, and S. Decker (2012). Improving the recall of live linked data querying through reasoning. In Proc. of *the 6th International Conference on Web Reasoning and Rule Systems, RR*, pages 188–2014. Springer Berlin Heidelberg, 2012.

[10] 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.

[11] N. A. Rakhmawati, M. Saleem, S. Lalithsena, and S. Decker. QFed: Query Set For Federated SPARQL Query Benchmark. In *Proceedings of the 16th International Conference on Information Integration and Web-based Applications & Services*, iiWAS '14, pages 207–211, New York, NY, USA, 2014. ACM.

[12] U. Von Luxburg. A tutorial on spectral clustering. In *Statistics and computing*, volume 17(4), pages 395–416. Springer, 2007.

[13] S. Brin and L. Page. Reprint of: The anatomy of a large-scale hypertextual web search engine. In*Computer networks*, volume 56(18), pages 3825–3833. Elsevier, 2012.

[14] 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.

[15] 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 585Ű600. Springer Berlin Heidelberg, 2011.

[16] 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.

[17] 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.

[18] 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.

[19] M. Saleem, Y. Khan, A. Hasnain, I. Ermilov, and A.-C.N. Ngomo. A fine-grained evaluation of SPARQL endpoint federation systems. In *Semantic Web Journal*, volume 7(5), pages 493–518, 2014.

[20] 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.

[21] 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.

[22] N. A. Rakhmawati, J. Umbrich, M. Karnstedt, A. Hasnain, and M. Hausenblas. Querying over Federated SPARQL Endpoints - A State of the Art Survey. In *CoRR*, volume abs/1306.1723, 2013.

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

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

[25] D. Cvetković, P. Rowlinson, and S. Simić. An Introduction to the Theory of Graph Spectra. *London Mathematical Society Student Texts*. Cambridge University Press, 2010.

[26] V. P. Petrović, D. Simijonović, M. N. Živanović, J. V. Košarić, Z. D. Petrović, S. Marković, and S. D. Marković. Vanillic Mannich bases: synthesis and screening of biological activity. Mechanistic insight into the reaction with 4-chloroaniline. *RSC Adv*, volume 4, pages 24635–24644, 2014.

[27] 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 Adv*, volume 5, pages 86274–86281, 2015.

Appendix

A. Coalescence with complete graphs

Fielder's papers [23,24] initiated the new era in which we can use sign of the eigenvectors' coordinates for cut finding. In [24] it was proved that the second smallest eigenvector of Laplacian matrix can be used for determining positive and negative vertices in graph thus providing a 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 disconnected graph with two components. In a following, as a sequel of Theorem 3.12 in [24], 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 [25], 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 eigenvector corresponding to the second smallest Laplacian eigenvalue μ_2 of the graph G, we shall find eigenvector corresponding to the μ_n eigenvalue of the graph $\overline{G} = K_{n-1,m-1} \cup \{v_n\}$. Because 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 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 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 null vertex.■