Scientific Research Publishing

Journal of Intelligent Learning Systems and Applications, 2018, 10, 1-20
http://www.scirp.org/journal/jilsa
ISSN Online: 2150-8410
ISSN Print: 2150-8402

# BioBroker: Knowledge Discovery Framework for Heterogeneous Biomedical Ontologies and Data

**Feichen Shen, Yugyung Lee**

University of Missouri, Kansas City, MO, USA
Email: fsm89@mail.umkc.edu

**How to cite this paper:** Shen, F.C. and Lee, Y. (2018) BioBroker: Knowledge Discovery Framework for Heterogeneous Biomedical Ontologies and Data. *Journal of Intelligent Learning Systems and Applications*, **10**, 1-20.
https://doi.org/10.4236/jilsa.2018.101001

**Received:** February 6, 2018
**Accepted:** February 25, 2018
**Published:** February 28, 2018

## Abstract

A large number of ontologies have been introduced by the biomedical community in recent years. Knowledge discovery for entity identification from ontology has become an important research area, and it is always interesting to discovery how associations are established to connect concepts in a single ontology or across multiple ontologies. However, due to the exponential growth of biomedical big data and their complicated associations, it becomes very challenging to detect key associations among entities in an inefficient dynamic manner. Therefore, there exists a gap between the increasing needs for association detection and large volume of biomedical ontologies. In this paper, to bridge this gap, we presented a knowledge discovery framework, the BioBroker, for grouping entities to facilitate the process of biomedical knowledge discovery in an intelligent way. Specifically, we developed an innovative knowledge discovery algorithm that combines a graph clustering method and an indexing technique to discovery knowledge patterns over a set of interlinked data sources in an efficient way. We have demonstrated capabilities of the BioBroker for query execution with a use case study on a subset of the Bio2RDF life science linked data.

## Keywords

Knowledge Discovery, Ontology, Linked Data

## 1. Introduction

With a large number of ontologies have been introduced by the biomedical community in recent years, one of the issues researchers are facing in healthcare and biomedical research is the challenging in analytics associated with large, complex, and dynamic healthcare data (e.g., electronic health records (EHRs), biomedical ontologies). Since there lacks appropriate tools and computational infrastructure that can be fully understood and utilized by involved personnel, very few capacities can be found to carry out analyses of these datasets [1]. As the demand for the integration and analysis of data has been growing steadily, the first effort toward connecting scattered biomedical data materialized as a data movement by the biomedical community (*i.e.*, the Linked Data) [2].

Increasingly, we are also seeing the emergence of biomedical and scientific collaboration. The Semantic Web Health Care and Life Sciences Interest Group (HCLSIG) [3] was formed to "improve collaboration, research and development, and innovation in the information ecosystem of the health care and life science domains using Semantic Web technologies". In this drive, the large amounts of biomedical data have been specified and shared via machine-readable formats, such as the Resource Description Framework (RDF) [4] and the Ontology Web Language (OWL) [5]. Ontologies are developed to easily extend the work of others and share across different domains. These semantic web technologies make it easier and more practical to integrate, query, and analyze the full scale of relevant biomedical and healthcare data for constructing cost effective health care systems [6]. From then on, knowledge discovery for entity identification from ontologies and various datasets [7] [8] [9] has become an important research area.

Although semantic web provides a solution for biomedical information exchange, there still exist significant difficulties on semantic seamless interoperability and interchange [10] [11] [12]. What is more, existing semantic approaches for linking are promising, but due to the exponential growth of biomedical big data and their complicated associations, it needs expensive computational capabilities to find key associations among entities in an inefficient dynamic manner [13] [14] [15]. The investigation on detecting associations among entities in a single ontology or across multiple ontologies is always an interesting topic [16] [17] [18] and there exists a gap between the increasing needs for association detection and large volume of biomedical ontologies.

Many efforts have been made to perform knowledge discovery with semantic web techniques. For example, in general settings, vSparQL was introduced to enable application ontologies to be derived from these large, fragmented sources such as the FMA [19]. The SMARTSPACE proposed a distributed platform for semantic knowledge discovery from services using multi-agent approach [20]. As a knowledge discovery task combined knowledge and clinical data, clinical ontology has been incorporated into collaborative filtering algorithm in our previous work to predict rare disease diagnosis [21] [22]. The PEMAR introduced a smart phone middleware for activity recognition discovery based on semantic models [23]. The GLEEN project aims to develop a service to simplify views for complex ontologies [24]. A mobile-cloud computing framework was established to discover infrastructure condition based on a back-end semantic knowledge discovery engine [25]. In our previous work, we have built a situation aware mobile applications framework [26] [27] to discovery users' activities in a dynamic way based on the semantic web rule language (SWRL) [28]. In biomedical domains, Tao *et al.*, have investigated the usage of semantic web technologies to discovery patient group based on advanced phenotyping algorithms [29] [30]. Based on the pharmacogenomics knowledge base (pharmgkb) [31], Zhu *et al.*, have leveraged web ontology language (OWL) and cheminformatics approaches to assist drug repositioning in breast cancer [32]. However, these studies didn't investigate the knowledge discovery on heterogeneous ontologies.

In this study, we presented a knowledge discovery framework BioBroker, which equipped with innovative algorithms that combine graph clustering method and an indexing technique. The aim of this framework is to generate cohesive query statements out of heterogeneous ontologies and execute these queries for the purpose of knowledge acquisition and discovery.

In the following, we first introduce materials used in this study. Next, we describe the methods and evaluation approaches used to build and test the framework. We then present the results followed by discussion. Lastly, we conclude and discuss potential future directions.

## 2. Materials

### *The Resource Description Framework* (*RDF*)

The RDF is a standard model for data interchange and information exchange on the web. It extends the linking structure of the web to use URIs to name the relationship between things as well as the two ends of the link, which are usually referred to as a triplet <subject, predicate, object> [33]. Ontologies are built upon the RDF with restrictions and axioms.

### *Bio2RDF*

Bio2RDF is a collection of biological knowledge bases which leverages semantic web technologies to provide interlinked life science data [34]. In this study, we used Bio2RDF release 2 and picked three widely used biomedical ontologies as a group of heterogeneous datasets for evaluation. They are the DrugBank [35] ontology, the HUGO Gene Nomenclature Committee (HGNC) [36], and the Mouse Genome Informatics (MGI) database [37].

### *Cytoscape*

The Cytoscape is an open source software used to visualize bioinformatics information and network [38]. In this study, we used the Cytoscape version 3.0.2 to develop the BioBroker knowledge discovery plugin.

### *OpenLink Virtuoso*

The OpenLink Virtuoso is a triple store database for managing linked data from existing data silos [39]. In this study, we installed the Virtuoso version 6.1 to store the heterogeneous ontologies.

## 3. Methods

The objective of this research is to find predicate patterns with a high degree of connectivity and identify a relatively small number of hops via highly connected nodes to traverse the RDF graphs. We are presenting how to define and discover such patterns of those significant nodes and use them for scalable query processing. We present our predicate-centric model in terms of definition of predicate patterns, discovery of patterns, and usage of patterns during query processing. Figure 1 summarizes the proposed framework and the following paragraphs illustrate each process and methodology respectively.

***Predicate Patterns***

A predicate P is representing a binary relation between two concepts (c1 and c2) in ontology. In RDF/OWL, P is represented as a property to express any kind of relationship (e.g., SubClassOf, Type) between domain (subject) and range (object) [5]. The domain and range may be either from the same ontology or from different ontologies. In our study, relationships are defined by the empirical analysis of ontology data. We are particularly interested in predicates (relationships) that are different from existing approaches like PSPARQL [40] and SPARQLer [41]. Apart from being similar, predicates may share other aspects, e.g., sharing the same subjects or the same objects as well as the connectivity between predicates. This focuses on not only concepts among graphs but links and other structural aspects of the concepts. In this study, the two types of predicate patterns are defined as follows.

*Share Patterns*: As shown in Table 1, this type of pattern describes the comprehension of the relationships between interacting nodes such as shared subjects and shared objects through the given predicate. Assume that two predicates are given as follows: $P_1$ <*Si, Oi*> and $P_2$ <*Sj, Oj*> where *Si, Sj* are a set of subjects

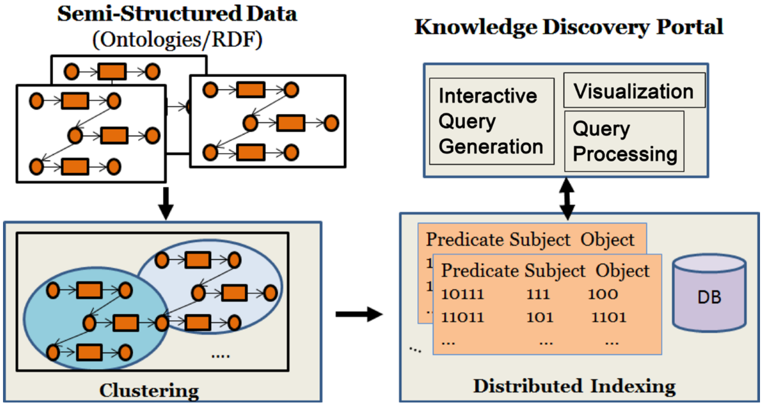


Figure 1. The BioBroker Framework.

Table 1. Predicate sharing patterns.

| Patterns | Semantic and Pragmatic Knowledge | |
|---|---|---|
| | Exact | Partial |
| Subject-Object Share | Si == Sj && Oi == Oj | Si >= Sj or Si <= Sj & Oi >= Oj or Oi <= Oj |
| Subject Share | Si == Sj | Si >= Sj or Si <= Sj |
| Object Share | Oi == Oj | Oi >= Oj or Oi <= Oj |

Table 2. Predicate connectivity patterns.

| Patterns | Semantic and Pragmatic Knowledge | |
|---|---|---|
| | Symbol | Condition |
| Path Connectivity | Si → P1→ Oi → P2 → Oj | P1 ≠ P2 && Oi = Sj |
| Cycle Connectivity | Si → P1 → Oi → P2 → Oj | P1 ≠ P2 && Oi = Sj && Si = Oj |

and *Oi, Oj* are a set of objects in given ontologies.

*Connection Patterns*: According to Table 2, the connection pattern is a frequently recurring pattern with predicates observed during ontology analysis and query processing as the basis for joining one query pattern to another. This pattern is based mainly on the connectivity of concept(s) through the respective predicates. This type of pattern describes the comprehension of the connectivity relationships between interacting predicates. Assume that two predicates are given as follows: P1 <Si, Oi> and P2 <Sj, Oj> where P1 is directly connected to P2 through Oi in the given ontologies.

### *Ontology Clustering with Predicates*

Based on the defined two predicate patterns, we found out that predicates play an important role as hubs to share information and connect entities among heterogeneous data. Therefore, we gave a hypothesis that graphs can be fuzzy clustered based on predicate sharing and distance measurement, and data in the same clustered group have a closer relationship than when in different ones.

*Predicate Neighboring Level Determination*: First, we need to define the boundary of domains in terms of sets of concepts and relations over the datasets. For this purpose, we proposed a predicate neighboring algorithm to determine the closeness of each of the two different predicates. Different shapes of edges denote different relationships between predicates $p_i$ and $p_j$ through concepts *C*. Level 1 has four different combinations that are based on a predicate sharing pattern as well as a connection pattern. Levels 2 and 3 have two various paths, respectively, that are based only on a predicate connection pattern. The formal definition is shown in Definition 1. It is obvious to find that the closeness of the relationship decreases as the level increases. Here we set the upper limit to three because we assume any relationship between predicates and beyond three levels is sparse.

**Definition 1:** Given a directed graph $G(V,E)$, Vertices $V_s,V_p,V_{so}$ denote subject, predicate, and object nodes in the RDF graph, respectively. Let $d(p_i,p_j)$ represent the shortest distance between $p_i$ and $p_j$, $r(p_i,p_j)$ determine the reachability between $p_i$ and $p_j$ $n(p_i,p_j)$ indicates the neighbors' closest level between $p_i$ and $p_j$:

$$n(p_i,p_j)=\begin{cases}1, & \text{if } d(p_i,p_j)=1\\ 2, & \text{if } d(p_i,p_j)=2 \text{ and } r(p_i,p_j)=\text{true}\\ 3, & \text{if } d(p_i,p_j)=3 \text{ and } r(p_i,p_j)=\text{true}\end{cases}$$

### *Predicate Similarity Measurement Calculation*

We utilized clustering approach to discover predicate association patterns from ontologies. The similarity based confusion measurement for the clustering algorithm varies based on different neighboring levels for each pair of predicates. Basically, we give higher weightage to closer predicates and lower weightage to further predicates. We give Definitions 2, 3, and 4 based on three levels respectively. The formula to generate a confusion matrix for a clustering algorithm is given by Definition 5.

**Definition 2:** Denote $p_i$ and $p_j$ as predicates in a RDF graph. A set of sets $\left\{\{S_{i1}\},\{S_{j1}\}\right\}$ contain all the predicates such that $\forall m \in \{S_{i1}\} \rightarrow n(m, p_i) = 1$ and $\forall n \in \{S_{j1}\} \rightarrow n(n, p_j) = 1$. Let $P(x)$ represent the number of entities that directly connect to predicate set *x* and $E(e)$ represents the number of entities for a given entity set *e*. Given entity set $\{C_1\}$ so that $\forall e_1 \in \{C_1\} \rightarrow e_1 \in \{S_{i1}\}$ and $e_1 \in \{S_{j1}\}$. The probability-based similarity $PS_{ij} = \frac{E(C_1)}{P(S_{i1})} * \frac{E(C_1)}{P(S_{j1})}$.

**Definition 3:** Denote $p_i$ and $p_j$ as predicates in an RDF graph. A set of sets $\left\{\{S_{i1}\},\{S_{j1}\},\{S_{i2}\},\{S_{j2}\}\right\}$ contain all the predicates such that $\forall m \in \{S_{i1}\} \rightarrow$ $n(m, p_i) = 1$, $\forall n \in \{S_{j1}\} \rightarrow n(n, p_j) = 1$, $\forall x \in \{S_{i2}\} \rightarrow n(x, p_i) = 2$ and $\forall y \in \{S_{j2}\} \rightarrow n(y, p_j) = 2$. Let $P(x)$ represent the number of entities directly connect to predicate set *x* and $E(e)$ represent the number of entities for a given entity set *e*.

Given entity set $\{C_1\}$ such that $\forall e_1 \in \{C_1\} \rightarrow e_1 \in \{S_{i1}\}$ and $e_1 \in \{S_{i2}\}$ or $\forall e_1 \in \{C_1\} \rightarrow e_1 \in \{S_{j1}\}$ and $e_1 \in \{S_{j2}\}$. The probability-based similarity $PS_{ij}^1 = \frac{E(C_1)}{P(S_{i1})}\frac{E(C_1)}{P(S_{i1})} * \frac{E(C_1)}{P(S_{i2})}$ and $PS_{ij}^2 = \frac{E(C_1)}{P(S_{j1})} * \frac{E(C_1)}{P(S_{j2})}$.

Given entity set $\{C_2\}$ such that $\forall e_2 \in \{C_2\} \rightarrow e_2 \in \{S_{i2}\}$ and $e_2 \in \{S_{j2}\}$. The probability-based similarity $PS_{ij}^3 = \frac{E(C_2)}{P(S_{i2})} * \frac{E(C_2)}{P(S_{j2})}$. Thus,

$$PS_{ij} = Max\left(PS_{ij}^1, PS_{ij}^2\right) * PS_{ij}^3$$

**Definition 4:** Denote $p_i$ and $p_j$ as predicates in an RDF graph. A set of sets $\left\{\{S_{i1}\},\{S_{j1}\},\{S_{i2}\},\{S_{j2}\},\{S_{i3}\},\{S_{j3}\}\right\}$ contain all the predicates such that $\forall m \in \{S_{i1}\} \rightarrow n(m, p_i) = 1$, $\forall n \in \{S_{j1}\} \rightarrow n(n, p_j) = 1$, $\forall x \in \{S_{i2}\} \rightarrow n(x, p_i) = 2$ and $\forall y \in \{S_{j2}\} \rightarrow n(y, p_j) = 2$, $\forall t \in \{S_{i3}\} \rightarrow n(t, p_i) = 3$ and $\forall k \in \{S_{j3}\} \rightarrow n(k, p_j) = 3$. Let $P(x)$ represent the number of entities directly connected to predicate set *x* and $E(e)$ represents the number of entities for a given entity set *e*.

Given set $\{C_1\}$ such that $\forall e_1 \in \{C_1\} \rightarrow e_1 \in \{S_{i1}\}$ and $e_1 \in \{S_{i2}\}$ or $\forall e_1 \in \{C_1\} \rightarrow e_1 \in \{S_{j1}\}$ and $e_1 \in \{S_{j2}\}$. The probability-based similarity $PS_{ij}^1 = \frac{E(C_1)}{P(S_{i1})} * \frac{E(C_1)}{P(S_{i2})}$ and $PS_{ij}^2 = \frac{E(C_1)}{P(S_{j1})} * \frac{E(C_1)}{P(S_{j2})}$.

Given set $\{C_2\}$ such that $\forall e_2 \in \{C_2\} \rightarrow e_2 \in \{S_{i2}\}$ and $e_2 \in \{S_{i3}\}$ or $\forall e_2 \in \{C_2\} \rightarrow e_2 \in \{S_{j2}\}$ and $e_2 \in \{S_{j3}\}$.

The probability-based similarity $PS_{ij}^3 = \frac{E(C_2)}{P(S_{i2})} * \frac{E(C_2)}{P(S_{i3})}$ and $PS_{ij}^4 = \frac{E(C_2)}{P(S_{j2})} * \frac{E(C_2)}{P(S_{j3})}$

Given set $\{C_3\}$ such that $\forall e_3 \in \{C_3\} \rightarrow e_3 \in \{S_{i3}\}$ and $e_3 \in \{S_{j3}\}$. The probability-based similarity $PS_{ij}^5 = \frac{E(C_3)}{P(S_{i3})} * \frac{E(C_3)}{P(S_{j3})}$ thus.

$$PS_{ij} = Max\left(PS_{ij}^1 * PS_{ij}^3, PS_{ij}^2 * PS_{ij}^4\right) * PS_{ij}^5$$

**Definition 5:** Given confusion matrix CM and total number of predicate *n*. Denote $PS_{ij}$ as the probability-based similarity score between predicates $p_i$ and $p_j$ based on different levels, so that:

$$\text{CM}\left[p_i, p_j\right] = \begin{cases} PS_{ij}, & \text{if } p_i \neq p_j, 0 \le i \le n, 0 \le j \le n \\ 1, & \text{if } p_i = p_j, 0 \le i \le n, 0 \le j \le n \end{cases}$$

We posit that predicate clustering is a required step for efficient query processing involving the alignment and integration of ontologies. Here we clarify our approach to efficient query processing and query generation within the above theoretical framework. A query processing consists of a collection of several relationships between multiple properties. Given that properties are more closely related to some properties more than others, property clustering and partition can be utilized for efficient query processing—the task of classifying a collection of properties into clusters. The guiding principle is to minimize inter-cluster similarity and maximize intra-cluster similarity, based on the notion of semantic distance.

### *Hierarchical Fuzzy C-Means Clustering Algorithm*

To discover the correlation between predicates, we used an innovative Hierarchical Fuzzy C-Means (HFCM) clustering algorithm. We created the HFCM algorithm and made a functional extension based on a Fuzzy C-Means clustering algorithm [42] [43]. In general, we set a machine capacity threshold to denote a certain number of triplets that each machine can hold. In addition, we kept applying the HFCM algorithm on each cluster until the number of triplets for each cluster was less than or equal to the threshold or no further change of numbers of elements for each cluster could be made. When compared to traditional Fuzzy C-Means algorithm, the HFCM is able to provide clustering topics in a hierarchical manner and provide flexibility to select clusters by levels. The algorithm of the HFCM is given in **Algorithm 1**.

### *Indexing for Ontology and Data*

Based on the variety of large biomedical data spreading in different clusters, a new indexing technique was developed for representing predicate patterns of ontologies from the clusters. Specifically, a two-level encoding approach has

**Algorithm 1.** Hierarchical Fuzzy C-Means Clustering

```
Input: Initialize list of data $X = \{X_1, \cdots, X_n\}$ with size n, threshold t, number of cluster c
Output: List of c cluster centers $C = \{C_1, \cdots, C_c\}$
1. for $c$ = 1 to $n$
2. List candidateList = $\arg\min_c \sum_{i=1}^{n} \sum_{j=1}^{c} w_{ij}^{m} \|x_i - c_j\|^2$
3. Apply Silhouette Width on candidateList, give value $q$
4. end for
5. Choose optimal $q$, List finallist = candidateList
6. for each cluster set s in finallist
7. int clusterSize = s.size()
8.            if(clusterSize > t)
9.                $n = c$
10.               do 1-18
11.           end if
12.           else if All clusterSize <= $t$
13.               return $C$
14.           end else if
15.           else if All clusterSize doesn't change
16.               return $C$
17.           end else if
18. end for
```

been developed to index the RDF schema, instance, and triple. For the cluster spaces, the two-level hierarchical indexing technique provides efficient representation of complex relations between nodes and predicate association patterns. We used binary encoding to index OWL/RDF schema and make binary with bitmap encoding together to index the OWL/RDF instance. For schema level, our assumption is that the size of schema for each medical and healthcare knowledge base should be a constant. The total size of schema encoding can be controlled even if binary encoding increases drastically. We used the binary index from binary 10 and started encoding with predicate to make sure all the predicate encoding was less than the entities encoding. For instance level, we assigned a unique bitmap index to each instance under its schema encoding. Our design philosophy is that instances with different schemas can share the same encoding but instances under the same schema must be assigned a unique indexing. Therefore, with the huge amount of instances, bitmap indexing colud be used in a scalable way and the combination of both binary and bitmap indexing uniquely determined an instance. For triple level, we applied logic *or* operation on schema encoding of the RDF subject, predicate, and object to generate the result. If a triple did not have a cycle, then we set the object schema encoding to be larger than the subject encoding. If a triple had a cycle, we used the right most bit as the indication of cycle bit and set the subject encoding as larger than the object encoding. In such a design, we can easily differentiate a cycle triple with a non-cycle one. Definition 6 illustrates this encoding approach in specific.

**Definition 6:** Given the i<sup>th</sup> node ($i \geq 0$) of Schema set {*S*}, j<sup>th</sup> node ($j \geq 0$) of Instance set {*I*}, predecessor set {*m*} and {*n*} contain all the father nodes of *i* and *j*, respectively. Denote each RDF triplet *t* as {*s*, *p*, *o*}. Let *S*(*i*) represent schema encoding set, *I*(*j*) represent instance encoding set, *TS*(*t*) represent triple schema encoding set, *TI*(*t*) represent triple instance encoding set and integer number *R* represent the magnitude of the data:

$$S(i)=\begin{cases}\left\{S(i-1)\vee 2^{i}\right\}, & \text{if } \{m\}\neq\varnothing \text{ and } \forall i\in\{m\}\\ \left\{2^{i}\right\}, & \text{if } \{m\}=\varnothing\end{cases}$$

$$I(j)=\begin{cases}\left\{S(i)+\dfrac{I(j-1)+1}{R}\right\}, & \text{if } \{n\}\neq\varnothing \text{ and } \forall j\in\{n\}\\ \left\{S(i)+\dfrac{1}{R}\right\}, & \text{if } \{n\}=\varnothing\end{cases}$$

$$TS(t)=\begin{cases}\left\{S(s)\vee S(p)\vee S(o)\mid S(o)>S(s)>S(p)>1\right\},\\ \quad \text{if } \{s,p,o\} \text{ does not form a cycle}\\ \left\{S(s)\vee S(p)\vee S(o)\vee 1\mid S(s)>S(o)>S(p)>1\right\},\\ \quad \text{if } \{s,p,o\} \text{ forms a cycle}\end{cases}$$

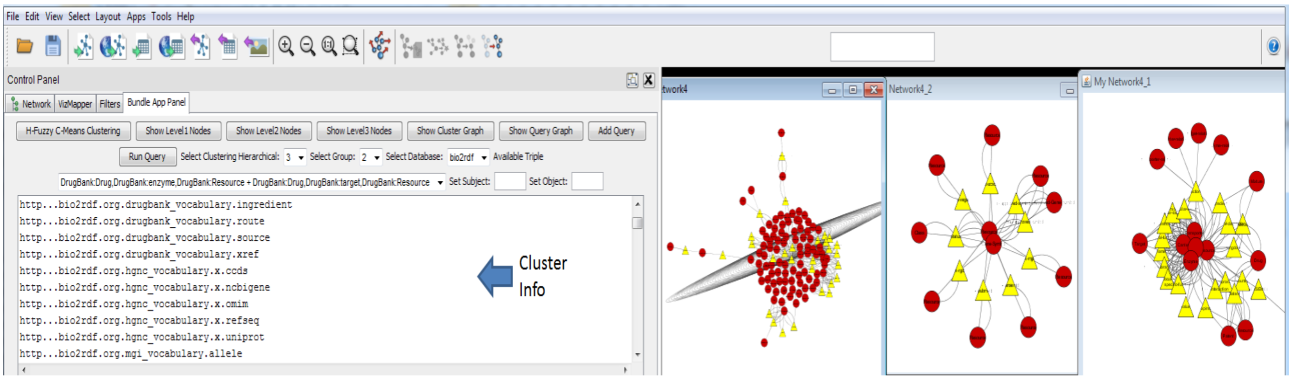
$$TI(t)=\left\{TS(t)+\frac{I(s)}{R}+\frac{I(o)}{R}\right\}$$

### *Query Processing using Predicate Patterns based Clustering and Indexing*

An intuitive query system was implemented based on clustering and indexing based on predicate patterns for imported medical data. Due to this innovated approach, the users' query could be answered with high accuracy and performance. A structured representation of semantic relations between concepts can be intuitively extended to query systems. Some features of our prototype BioBroker framework are listed below.

*Integrated OWL/RDF Schema Clustering*: Different OWL/RDF medical sets can be imported to the BioBroker. Our system is able to parse the schema based on data and apply the HFCM algorithm on schema based on predicate similarity. Clustering graphs are also generated accordingly and triple with the same predicate among different schema sets can be linked. Figure 2 shows predicate-based clustering graphs with 3 Bio2RDF data schema after suppying the BioBroker a predicate similarity feature vector by clicking the *H-Fuzzy C-means Clustering* button. Detailed predicate clustering information was also listed in the clustering panel on the left. Because we used hierarchical approach in addition to the Fuzzy C-means clustering, our system provided options to display different levels of data, Figure 2 shows an example with level 3.

*Query Boundary*: Query processing can be optimized based on the proposed concept of query boundary. The boundary can be determined by predicate association and clustering sets. A query boundary characterizes a particular dynamic reasoning and query capability of the proposed model that is specifically tailored

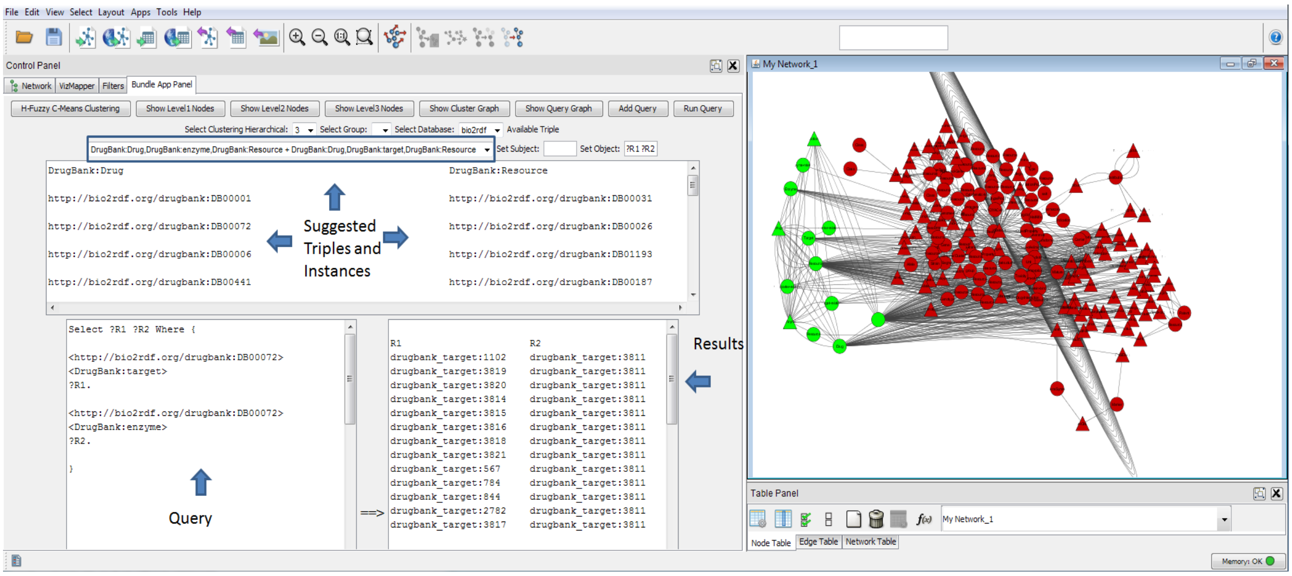


Select Clustering Hierarchical: 3 Select Group: 2 Select Database: bio2rdf

DrugBank:Drug,DrugBank:enzyme,DrugBank:Resource + DrugBank:Drug,DrugBank:target,DrugBank:Resource Set Subject: Set Object:

```
http...bio2rdf.org.drugbank_vocabulary.ingredient
http...bio2rdf.org.drugbank_vocabulary.route
http...bio2rdf.org.drugbank_vocabulary.source
http...bio2rdf.org.drugbank_vocabulary.xref
http...bio2rdf.org.hgnc_vocabulary.x.ccds
http...bio2rdf.org.hgnc_vocabulary.x.ncbigene
http...bio2rdf.org.hgnc_vocabulary.x.omim
http...bio2rdf.org.hgnc_vocabulary.x.refseq
http...bio2rdf.org.hgnc_vocabulary.x.uniprot
http...bio2rdf.org.mgi_vocabulary.allele
```

**Figure 2.** Clustering graph visualization.



Select Clustering Hierarchical: 3 Select Group: Select Database: bio2rdf

DrugBank:Drug,DrugBank:enzyme,DrugBank:Resource + DrugBank:Drug,DrugBank:target,DrugBank:Resource Set Subject: Set Object: ?R1 ?R2

```
DrugBank:Drug                                  DrugBank:Resource

http://bio2rdf.org/drugbank:DB00001            http://bio2rdf.org/drugbank:DB00031

http://bio2rdf.org/drugbank:DB00072            http://bio2rdf.org/drugbank:DB00026

http://bio2rdf.org/drugbank:DB00006            http://bio2rdf.org/drugbank:DB01193

http://bio2rdf.org/drugbank:DB00441            http://bio2rdf.org/drugbank:DB00187
```

```
Select ?R1 ?R2 Where {

<http://bio2rdf.org/drugbank:DB00072>
<DrugBank:target>
?R1.

<http://bio2rdf.org/drugbank:DB00072>
<DrugBank:enzyme>
?R2.

}
```

```
==>
    R1                      R2
    drugbank_target:1102    drugbank_target:3811
    drugbank_target:3819    drugbank_target:3811
    drugbank_target:3820    drugbank_target:3811
    drugbank_target:3814    drugbank_target:3811
    drugbank_target:3815    drugbank_target:3811
    drugbank_target:3816    drugbank_target:3811
    drugbank_target:3818    drugbank_target:3811
    drugbank_target:3821    drugbank_target:3811
    drugbank_target:567     drugbank_target:3811
    drugbank_target:784     drugbank_target:3811
    drugbank_target:844     drugbank_target:3811
    drugbank_target:2782    drugbank_target:3811
    drugbank_target:3817    drugbank_target:3811
```

**Figure 3.** Customized query design and suggestion with query boundary in integrated graph.

for query semantics. More specifically, it can be proved that for a specific kind of user's query, there exist a fixed set of abstract patterns that are involved in the query processing process. This fixed set is called the query boundary for the specific type of users' query. A query was described into query boundary within clusters and the BioBroker used a different green color to indicate such boundary specifically. As an example shown in Figure 3, we included three non-built-in predicates which are *drugbank:enzyme*, *drugbank:action* and *drugbank:drug*.

*Interactive Query Design with Suggestion*: The predicates extracted from a user's SPARQL query [44] are mapped to the predicate neighboring and clustering results. If matched, then a set of relevant queries for different medical data sets can be composed with the significant properties (relationships) between concepts that are identified by the proposed formula in this paper. The BioBroker provides a customized query design and suggestion feature to make it convenient for users to design benchmark queries. An extended function is developed to add query based on predicate association. Users can choose available triple from the drop down list and they can also assign variables for subjects and objects. As shown in **Figure 3**, a query schema suggestion (Drug -> enzyme -> Resource and Drug -> target -> Resource) was given to user. Meanwhile, all the related instances were read from database and listed for users to choose. Here we gave *DB*00072 as subject drug name and set ?R1 and ?R2 as objects. In this example, The BioBroker was able to find target names for *drugbank*:*enzyme* and *drugbank*:*target* based on given drug instance. A query boundary with integrated graph is also showed in this example.

*Query Indexing to Optimize Benchmark Query Performance*: The BioBroker translates each SPARQL query to query indexing format based on medical Ontology and data indexing. Therefore, executing the SPARQL query is actually performing logical operations on schema binary indexing and mathematical operations on instance bitmap indexing. In **Figure 3**, a SPARQL query was given and its corresponding query graph was shown. The BioBoker translated the SPARQL query into binary format and generated results for user.

**Evaluation**

The BioBroker prototype system was implemented using Java on Eclipse Juno Integrated Development Environment [45]. Apache Jena API was used to parse OWL/RDF datasets and retrieve triple information. We used R computing environment [46] to implement algorithms and generate predicate clusters. We designed a plugin to generate query and schema graphs by programming with CytoScape 3.0.238. We embedded an encoding query engine in the plugin and provided suggested query option based on the clustering results. To report the similarity measurements of the predicates in these datasets on to excel files, we used Java Excel API [47].

The evaluation of the BioBroker system is conducted in terms of the valid of clustering result and justified query benchmark generation. We used three ontologies from Bio2RDF release 2 to evaluate our system. Detailed information for each ontology is given in **Table 3**. In addition to that, we eliminated some RDF built-in predicates and types for getting the best clustering result.

To select the optimal clustering algorithm for knowledge discovery, we first compared performances yielded by the Hierarchical Fuzzy C-Means (HFCM), the Partition Around Medoids (PAM) algorithm [48], the Clustering Large Application (CLARA) algorithm [49], the K-Means clustering algorithm [50] and the Hierarchical Clustering (HC) algorithm [51]. To get the optimal number of clusters, we used Silhouette Width (SW) [52] to evaluate different results and chose the one with the biggest score. In addition, we used the Sum of Squares for Error (SSE) metric [53] to double check the optimal number of clusters for the

**Table 3.** Bio2RDF ontology information.

| Ontology | Triple Types | Entities | Properties | Triple Instances |
|---|---|---|---|---|
| Drugbank | 306 | 91 | 56 | 3,649,750 |
| HUGO Gene Nomenclature Committee (HGNC) | 84 | 19 | 18 | 3,628,205 |
| Mouse Genome Informatics (MGI) | 140 | 17 | 19 | 8,206,813 |

selected optimal clustering algorithm.

For query evaluation, we selected eight query benchmarks [54] [55] and used BioBroker and Virtuoso to execute each of them for query outputs validation and query execution performance test. The machine we used to execute queries has an Intel Pentium G3220 3.00 GHz CPU. The memory size is 12 GB and the storage size is 1 TB.

## 4. Results

*Evaluation for HFCM Algorithm*

As shown in Figure 4, according to SW score, all algorithms produced the optimal performances at the point when number of clusters became 2. We found that K-Means yielded the highest SW score as 0.9, HFCM produced the suboptimal performance as 0.88, and the other three algorithms contributed to a same SW score as 0.76. Although SW for K-Means was higher than the one for HFCM, there is no statistical significant difference between them. Therefore, we selected HFCM as the optimal algorithm since it is able to provide additional soft partition capabilities, which was useful for distributed query processing. As a result, the HFCM produced 7 clusters in total as final outputs based purely on non-built-in RDF predicates. We then used the SSE metric to confirm the optimal number of clusters for the HFCM. As shown in Figure 5, the first concavity point for the SSE plot proved that the optimal number of clusters is 2.

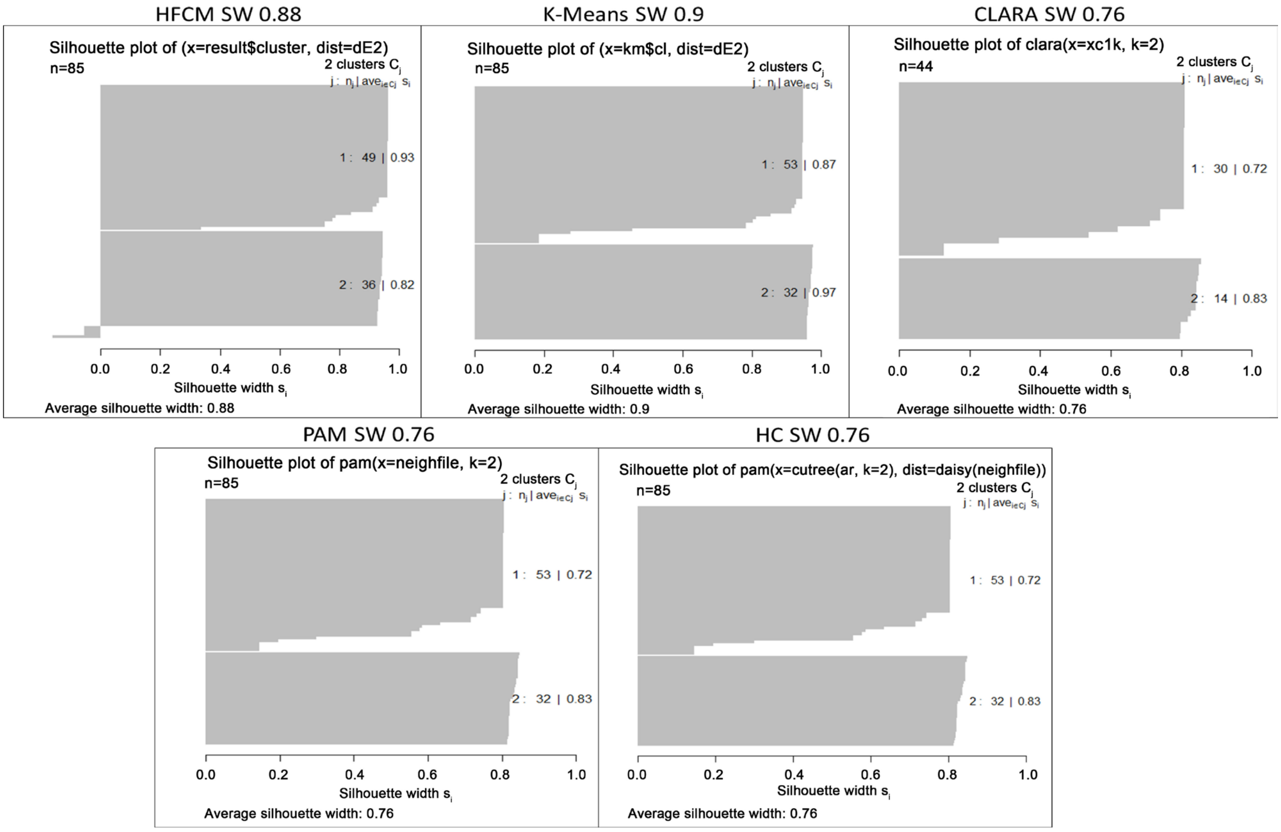


Figure 4. Evaluations for clustering algorithms.

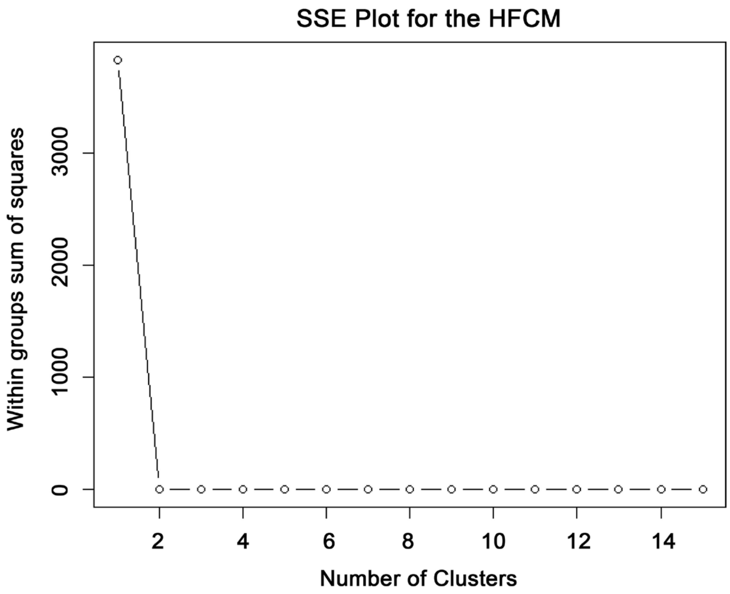


Figure 5. The sum of squares for error (SSE) plot for the HFCM.

Table 4. Hierarchical fuzzy c-means clustering result.

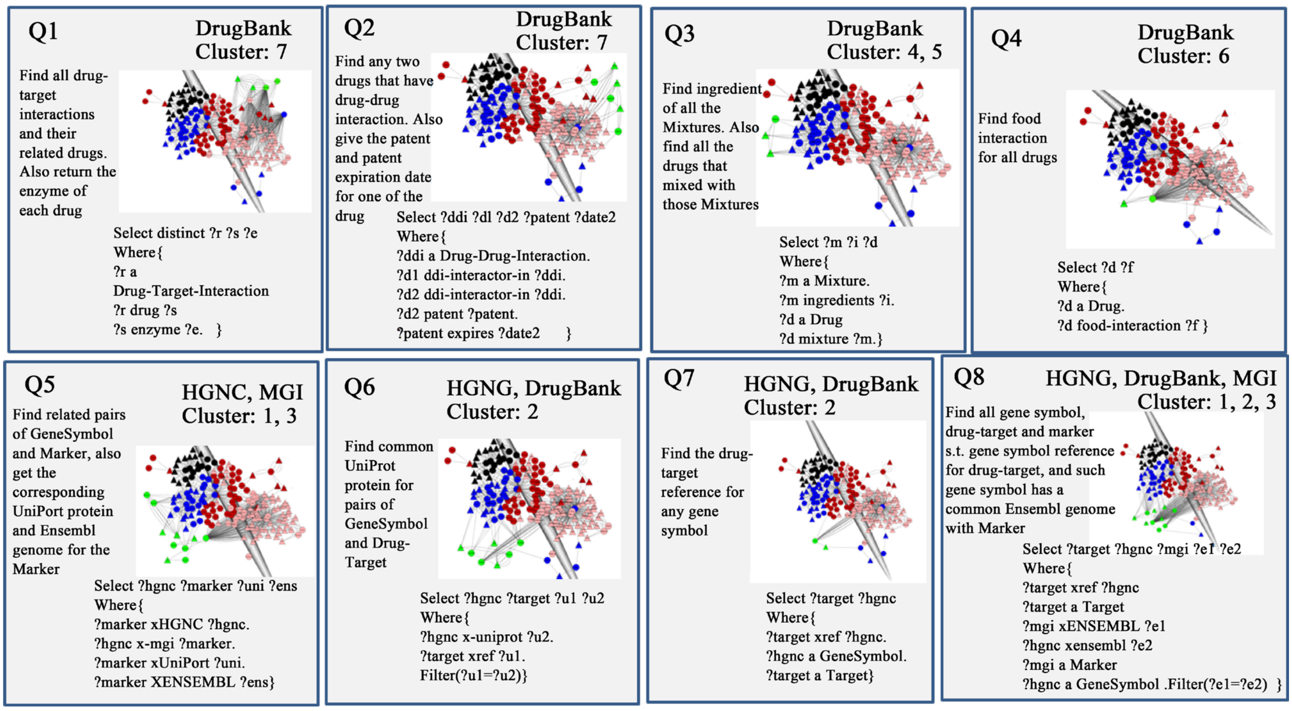
| ID | Size | Predicate Name |
|---|---|---|
| Cluster 1 | 17 | MGI:allele, MGI:marker, MGI:phenotype, MGI:strain.type, MGI:x.ensembl.protein, MGI:x.ensembl.transcript, MGI:x.genbank, MGI:x.pubmed, MGI:x.refseq.protein, MGI:x.refseq.transcript, MGI:x.trembl, MGI:x.uniprot, MGI:x.vega.protein, MGI:x.vega.transcript, MGI:xHGNC, MGI: theoretical.pi, MGI:xENSEMBL |
| Cluster 2 | 6 | HGNC:x.ccds, HGNC:x.ncbigene, HGNC:x.omim, HGNC:x.refseq, HGNC:x.uniprot, DrugBank:xref |
| Cluster 3 | 9 | HGNC:has.approved.symbol, HGNC:is.approved.symbol.of, HGNC:status, HGNC:x.ensembl, HGNC:x.mgi, HGNC:x.pubmed, HGNC:x.rgd, HGNC:x.ucsc, HGNC:x.vega |
| Cluster 4 | 6 | DrugBank:form, DrugBank:ingredient, DrugBank:ingredients, DrugBank:route,DrugBank:source,DrugBank:molecular.weight |
| Cluster 5 | 18 | DrugBank:manufacturer, DrugBank:mechanism.of.action, DrugBank:molecular.weight, DrugBank:name, DrugBank:packager, DrugBank:pharmacology, DrugBank:protein.binding, DrugBank:route.of.elimination, DrugBank:specific.function, DrugBank:substructure, DrugBank:synonym, DrugBank:target, DrugBank:mixture, DrugBank:theoretical.pi, DrugBank:toxicity, DrugBank:transmembrane.regions, DrugBank:transporter, DrugBank:value, DrugBank:volume.of.distribution |
| Cluster 6 | 19 | DrugBank:absorption, DrugBank:action, DrugBank:affected.organism, DrugBank:biotransformation, DrugBank:brand, DrugBank:calculated.property, DrugBank:category, DrugBank:cellular.location, DrugBank:dosage, DrugBank:drug, DrugBank:experimental.property, DrugBank:food.interaction, DrugBank:gene.name, DrugBank:general.function, DrugBank:half.life, DrugBank:indication, DrugBank:kingdom, DrugBank:locus, |
| Cluster 7 | 10 | DrugBank:approved, DrugBank:country, DrugBank:ddi.interactor.in, DrugBank:enzyme, DrugBank:expires, DrugBank:mixture, DrugBank:patent, DrugBank:price, DrugBank:product, DrugBank:drug |

Detailed clustering information by the HFCM is given in Table 4. We found that cluster 1 mainly focused on the homogeneous ontology MGI and provided knowledge about associations for phenotype, gene marker, and protein and so on. Cluster 2 discovered cross-domain knowledge between the HGNC and the DrugBank, indicating their associations with other ontologies, such as the OMIM [56] and the UniProt [57]. Cluster 3 depicted biomedical information in HGNC, including gene symbol, ensemble, and outgoing linkage to other knowledge bases. Cluster 4, 5, 6, and 7 are all about the DrugBank with different focuses on ingredient, target, interaction, and enzyme respectively.

*Evaluation for Query Performance*

Query benchmark was established and detailed information of queries can be found in Figure 6. Specifically, Query 2 and 5 were designed based on the online benchmarks with some modifications due to the data version compatible issue, and the rest were designed from the BioBroker suggestions by choosing predicates from single cluster or multiple clusters. In these query graphs, we used color black, blue, pink, red and green color to present entities from the HGNC, the MGI, the DrugBank, built-in predicates/entities, and query boundary respectively. Queries 1 to 4 were designed mainly based on homogeneous DrugBank and the rest queries were designed based on heterogeneous ontologies. Query 1 was about finding interactions between drug and enzyme. Query 2 aimed to detect interactions among drugs. The objective of query 3 is to find ingredient of all mixtures. Query 4 targeted on mining food interactions with drugs. Query 5 was composed of knowledge from the HGNC and the MGI, describing associations among gene symbols, markers, and proteins. Query 6 was composed of information extracted from the HGNC and the DrugBank, illustrating the common protein for pairs of gene symbols and drug target. Query 7 was also made up of information from the HGNC and the DrugBank, introducing the relationship between drug targets and gene symbols. Query 8 is a mixed query with all three ontologies, which aimed to find all gene symbols, drug-targets and gene markers with a common ensemble genome.

We executed all queries on the Virtuotoso Database and retrieved relevant



Find all drug-target interactions and their related drugs. Also return the enzyme of each drug

```
Select distinct ?r ?s ?e
Where{
?r a
Drug-Target-Interaction
?r drug ?s
?s enzyme ?e.   }
```

Find any two drugs that have drug-drug interaction. Also give the patent and patent expiration date for one of the drug

```
Select ?ddi ?dl ?d2 ?patent ?date2
Where{
?ddi a Drug-Drug-Interaction.
?d1 ddi-interactor-in ?ddi.
?d2 ddi-interactor-in ?ddi.
?d2 patent ?patent.
?patent expires ?date2      }
```

Find ingredient of all the Mixtures. Also find all the drugs that mixed with those Mixtures

```
Select ?m ?i ?d
Where{
?m a Mixture.
?m ingredients ?i.
?d a Drug
?d mixture ?m.}
```

Find food interaction for all drugs

```
Select ?d ?f
Where{
?d a Drug.
?d food-interaction ?f }
```

Find related pairs of GeneSymbol and Marker, also get the corresponding UniPort protein and Ensembl genome for the Marker

```
Select ?hgnc ?marker ?uni ?ens
Where{
?marker xHGNC ?hgnc.
?hgnc x-mgi ?marker.
?marker xUniPort ?uni.
?marker XENSEMBL ?ens}
```

Find common UniProt protein for pairs of GeneSymbol and Drug-Target

```
Select ?hgnc ?target ?u1 ?u2
Where{
?hgnc x-uniprot ?u2.
?target xref ?u1.
Filter(?u1=?u2)}
```

Find the drug-target reference for any gene symbol

```
Select ?target ?hgnc
Where{
?target xref ?hgnc.
?hgnc a GeneSymbol.
?target a Target}
```

Find all gene symbol, drug-target and marker s.t. gene symbol reference for drug-target, and such gene symbol has a common Ensembl genome with Marker

```
Select ?target ?hgnc ?mgi ?e1 ?e2
Where{
?target xref ?hgnc
?target a Target
?mgi xENSEMBL ?e1
?hgnc xensembl ?e2
?mgi a Marker
?hgnc a GeneSymbol .Filter(?e1=?e2)  }
```

Figure 6. Homogeneous and heterogeneous query graphs.

Table 5. Query execution results.

| Query Number | Answers |
|---|---|
| Q1 | r = DrugBank:DB00157_711, s = DrugBank:DB00157, e = DrugBank:12 |
| Q2 | ddi = DrugBank:DB00001_DB01381, d1 = DrugBank;DB00001, d2 = DrugBank:DB00001, patent = uspatent:5180668, data2 = 2010-01-19 |
| Q3 | m = DrugBank:Cauterex, i = domase alfa + fibrinolysin + gentamicin sulfate, d = DrugBank:DB00003 |
| Q4 | d = DrugBank:DB00006, f = Dan Shen, dong quai, evening primrose oil, gingko, policosanol, willow bark |
| Q5 | hgnc = HGNC:26946, marker = MGI:1913367, mgi = MGI:1913367, uni = Uniprot:Q9CR13, ens = Ensembl: ENSMUSG00000019689 |
| Q6 | Hgnc = HGNC:7863, target = DrugBank:11, u1 = UNIPROT:Q13423, u2 = Uniprot:Q13423 |
| Q7 | Target = DrugBank:9, hgnc = HGNC:5211 |
| Q8 | Target = DrugBank:6601, hgnc = HGNC:24427, mgi = MGI:88574, e1 = Ensembl: ENSMUSG00000015340, e2 = ENSMUSG000000197953 |

Table 6. Query performance comparison (in milliseconds).

| | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 |
|---|---|---|---|---|---|---|---|---|
| BioBroker | 8 | 17 | 10 | 5 | 8 | 4 | 4 | 25 |
| Virtuoso | 525 | 954 | 590 | 125 | 403 | 219 | 168 | 1037 |

results as shown in Table 5. Here we only demonstrated one output for each query.

We also tested query execution performances on Bio2RDF DrugBank, HGNC and MGI dataset with query 1 - 8. We compared our indexed query performance with Virtuoso based SPARQL query performance. The small scale data we used has 3,651,750 triples and 105 properties. The performance comparison results are showed in Table 6. We observed that the BioBroker has a significant faster execution performance than Virtuoso in millisecondes, which indicated that the use of distributed index technique is able to accelerate the query process.

## 5. Discussion

There are several studies for extension of the SPARQL query with some extended patterns such as path SPARQL [40] and semantic Association discovery [41]. Protein-protein interaction was analyzed with SPARQL based RDF decomposition [3]. However, these are all graph based pattern matching approaches that may not be appropriate for a huge volume of evolving data and subsequently, not suitable for discovering assertions from such data. Therefore, we used a pattern-based approach for analyzing ontologies whose concepts were either subjects or objects in the discovered predicate patterns and used them for query processing. The clustering enhanced the query designing and query processing by providing an ultimately better comprehension of the relationships between interacting nodes on the data. The dynamic clustering allowed us to execute highly specific queries and dynamically expand or slink knowledge and data space as well as share new data with other clouds making it possible to achieve scalable reasoning.

## 6. Conclusions and Future Work

This paper presents a predicate pattern based model equipped with index technique for query suggestion, visualization, scalable query and reasoning with large biomedical ontology schema and data. The proposed model transforms conjunctive SPARQL queries into efficient pattern based queries over a set of interlinked medical data sources. The benefits of predicate-based query processing were shown with discovery of predicate patterns. The proposed model was evaluated with the Bio2RDF datasets and the experimental results of the query designing and results showed the superiority of the proposed predicate-centric model compared to existing query models.

In the future, we will combine graph network analysis approaches [58] [59] with clustering algorithm to provide network motif [60] analysis and LDA-based topic modelling [61]. Furthermore, parallel and distributed algorithms, using the indexing technique, will be developed.

Human Phenotype Ontology (HPO) [62] has been developed as a controlled vocabulary for phenotypes by mining and integrating phenotype knowledge from medical literature and ontologies. HPO also provides associations with other biomedical resources such as the Gene Ontology [63]. We have developed an annotation pipeline leveraging HPO for phenotypic characterization on clinical data [64] [65]. In the future, we will combine knowledge-driven and data-driven approaches to investigate knowledge discovery from clinical domains to facilitate translational research.

## References

[1] Nekrutenko, A., *et al.* (2012) Next-Generation Sequencing Data Interpretation: Enhancing Reproducibility and Accessibility. *Nature Reviews Genetics*, **13**, 667. https://doi.org/10.1038/nrg3305

[2] Bizer, C., *et al.* (2009) Linked Data—The Story So Far. *International Journal on Semantic Web and Information Systems*, **5**, 1-22. https://doi.org/10.4018/jswis.2009081901

[3] Semantic Web Health Care and Life Sciences Interest Group (2018) http://www.w3.org/2001/sw/hcls/

[4] Lassila, O., *et al.* (1999) Resource Description Framework (RDF) Model and Syntax Specification. W3C (MIT, INRIA, Keio), 1-39.

[5] Bechhofer, S. (2009) OWL: Web Ontology Language. Encyclopedia of Database Systems: Springer, Berlin, 2008-2009.

[6] Luciano, J.S., *et al.* (2011) The Translational Medicine Ontology and Knowledge Base: Driving Personalized Medicine by Bridging the Gap between Bench and Bedside. *Journal of Biomedical Semantics*, **2**, S1. https://doi.org/10.1186/2041-1480-2-S2-S1

[7] Shen, F., *et al.* (2016) Predicate Oriented Pattern Analysis for Biomedical Knowledge Discovery. *Intelligent Information Management*, **8**, 66. https://doi.org/10.4236/iim.2016.83006

[8] Shen, F., *et al.* (2017) Populating Physician Biographical Pages Based on EMR Data. *AMIA Summits on Translational Science Proceedings*, **2017**, 522.

[9] Shen, F. (2015) A Pervasive Framework for Real-Time Activity Patterns of Mobile Users. *Pervasive Computing and Communication Workshops* (*PerCom Workshops*), 2015 *IEEE International Conference on*, St. Louis, 23-27 March 2015, 248-250.

[10] Sheth, A.P. (1999) Changing Focus on Interoperability in Information Systems: From System, Syntax, Structure to Semantics. Interoperating Geographic Information Systems: Springer, Berlin, 5-29. https://doi.org/10.1007/978-1-4615-5189-8_2

[11] Shvaiko, P., *et al.* (2008) Ten Challenges for Ontology Matching. *OTM Confederated International Conferences* "*On the Move to Meaningful Internet Systems*", Monterrey, 9-14 November 2008, 1164-1182. https://doi.org/10.1007/978-3-540-88873-4_18

[12] Shvaiko, P., *et al.* (2013) Ontology Matching: State of the Art and Future Challenges. *IEEE Transactions on Knowledge and Data Engineering*, **25**, 158-176. https://doi.org/10.1109/TKDE.2011.253

[13] Wu, X., *et al.* (2014) Data Mining with Big Data. *IEEE Transactions on Knowledge and Data Engineering*, **26**, 97-107. https://doi.org/10.1109/TKDE.2013.109

[14] Kambatla, K., *et al.* (2014) Trends in Big Data Analytics. *Journal of Parallel and Distributed Computing*, **74**, 2561-2573. https://doi.org/10.1016/j.jpdc.2014.01.003

[15] Reed, D.A., *et al.* (2015) Exascale Computing and Big Data. *Communications of the ACM*, **58**, 56-68. https://doi.org/10.1145/2699414

[16] Shen, F., *et al.* (2016) Knowledge Discovery from Biomedical Ontologies in Cross Domains. *PLoS ONE*, **11**, e0160005. https://doi.org/10.1371/journal.pone.0160005

[17] Shen, F. (2016) A Graph Analytics Framework For Knowledge Discovery. PhD Dissertation, University of Missouri, Kansas City. https://mospace.umsystem.edu/xmlui/handle/10355/49408

[18] Shen, F., *et al.* (2018) MedTQ: Dynamic Topic Discovery and Query Generation for Medical Ontologies. arXiv Preprint, arXiv:180203855.

[19] Shaw, M., *et al.* (2008) Generating Application Ontologies from Reference Ontologies. *AMIA Annual Symposium Proceedings*, Washington DC, 8-12 November 2008, 672-676.

[20] Dasgupta, S., *et al.* (2014) SMARTSPACE: Multiagent Based Distributed Platform for Semantic Service Discovery. *IEEE Transactions on Systems*, *Man*, *and Cybernetics*: *Systems*, **44**, 805-821. https://doi.org/10.1109/TSMC.2013.2281582

[21] Shen, F., *et al.* (2017) Leveraging Collaborative Filtering to Accelerate Rare Disease Diagnosis. American Medical Informatics Association, Washington D.C.

[22] Shen, F., *et al.* (2017) Accelerating Rare Disease Diagnosis with Collaborative Filtering. American Medical Informatics Association, Washington D.C.

[23] Vaka, P., *et al.* (2015) PEMAR: A Pervasive Middleware for Activity Recognition with Smart Phones. *IEEE International Conference on Pervasive Computing and Communication Workshops*, St. Louis, 23-27 March 2015, 409-414. https://doi.org/10.1109/PERCOMW.2015.7134073

[24] Detwiler, L.T., *et al.* (2008) Regular Paths in SparQL: Querying the NCI Thesaurus. *AMIA Annual Symposium Proceedings*, Washington DC, 8-12 November 2008,

161-165.

[25] Chen, Z., *et al.* (2013) Collaborative Mobile-Cloud Computing for Civil Infrastructure Condition Inspection. *Journal of Computing in Civil Engineering*, **29**, Article ID: 04014066. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000377

[26] Shen, F., *et al.* (2015) SAMAF: Situation Aware Mobile Apps Framework. *IEEE International Conference on Pervasive Computing and Communication Workshops*, St. Louis, 23-27 March 2015, 26-31.

[27] Shen, F. (2012) Situation Aware Mobile Apps Framework. Master Thesis, University of Missouri, Kansas City. https://mospace.umsystem.edu/xmlui/handle/10355/15637

[28] Horrocks, I., *et al.* (2004) SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Member Submission, 79.

[29] Tao, C., *et al.* (2013) Phenotyping on EHR Data using OWL and Semantic Web Technologies. *International Conference on Smart Health*, Beijing, 3-4 August 2013, 31-32. https://doi.org/10.1007/978-3-642-39844-5_5

[30] Shen, F., *et al.* (2014) Using Semantic Web Technologies for Quality Measure Phenotyping Algorithm Representation and Automatic Execution on EHR Data. *IEEE-EMBS International Conference on Biomedical and Health Informatics*, Valencia, 1-4 June 2014, 531-534.

[31] Hewett, M., *et al.* (2002) PharmGKB: The Pharmacogenetics Knowledge Base. *Nucleic Acids Research*, **30**, 163-165. https://doi.org/10.1093/nar/30.1.163

[32] Zhu, Q., *et al.* (2014) Exploring the Pharmacogenomics Knowledge Base (Pharmgkb) for Repositioning Breast Cancer Drugs by Leveraging Web Ontology Language (OWL) and Cheminformatics Approaches. 19*th Pacific Symposium on Biocomputing*, Kohala Coast, 3-7 January 2014, 172-182.

[33] Resource Description Framework (RDF). https://wwww3org/RDF/

[34] Callahan, A., *et al.* (2013) Bio2RDF Release 2: Improved Coverage, Interoperability and Provenance of Life Science Linked Data. *Extended Semantic Web Conference*, Montpellier, 26-30 May 2013, 200-212. https://doi.org/10.1007/978-3-642-38288-8_14

[35] Wishart, D.S., *et al.* (2007) DrugBank: A Knowledgebase for Drugs, Drug Actions and Drug Targets. *Nucleic Acids Research*, **36**, D901-D906.

[36] Povey, S., *et al.* (2001) The HUGO Gene Nomenclature Committee (HGNC). *Human Genetics*, **109**, 678-680. https://doi.org/10.1007/s00439-001-0615-0

[37] Bult, C.J., *et al.* (2008) The Mouse Genome Database (MGD): Mouse Biology and Model Systems. *Nucleic Acids Research*, **36**, D724-D728.

[38] Shannon, P., *et al.* (2003) Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. *Genome Research*, **13**, 2498-2504. https://doi.org/10.1101/gr.1239303

[39] Erling, O., *et al.* (2009) RDF Support in the Virtuoso DBMS. In: Pellegrini, T., Auer, S., Tochtermann, K. and Schaffert, S., Eds., *Networked Knowledge-Networked Media*, Springer, Berlin, 7-24. https://doi.org/10.1007/978-3-642-02184-8_2

[40] Alkhateeb, F., *et al.* (2009) Extending SPARQL with Regular Expression Patterns (for Querying RDF). *Journal of Web Semantics*, **7**, 57-73. https://doi.org/10.1016/j.websem.2009.02.002

[41] Kochut, K.J., *et al.* (2007) SPARQLeR: Extended SPARQL for Semantic Association Discovery. *European Semantic Web Conference*, Innsbruck, 3-7 June 2007, 145-159. https://doi.org/10.1007/978-3-540-72667-8_12

[42] Bezdek, J.C., *et al.* (1984) FCM: The Fuzzy c-Means Clustering Algorithm. *Computers & Geosciences*, **10**, 191-203. https://doi.org/10.1016/0098-3004(84)90020-7

[43] Shen, F., *et al.* (2015) BmQGen: Biomedical Query Generator for Knowledge Discovery. *IEEE International Conference on Bioinformatics and Biomedicine*, Washington DC, 9-12 November 2015, 1092-1097.

[44] Prud, E., *et al.* (2006) SPARQL Query Language for RDF.

[45] Eclipse Juno Integrated Development Environment. https://wwweclipseorg/juno/

[46] The R Project for Statistic. http://wwwr-projectorg/

[47] JExcelAPI. http://jexcelapisourceforgenet/

[48] Kaufman, L., *et al.* (1990) Partitioning around Medoids (Program PAM). In: Kaufman, L. and Rousseeuw, P., Eds., *Finding Groups in Data*: *An Introduction to Cluster Analysis*, John Wiley, New York, 68-125.

[49] Ester, M., *et al.* (1996) A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.

[50] Hartigan, J.A., *et al.* (1979) Algorithm AS 136: A k-Means Clustering Algorithm. *Journal of the Royal Statistical Society Series C* (*Applied Statistics*), **28**, 100-108. https://doi.org/10.2307/2346830

[51] Johnson, S.C. (1967) Hierarchical Clustering Schemes. *Psychometrika*, **32**, 241-254. https://doi.org/10.1007/BF02289588

[52] Rousseeuw, P.J. (1987) Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. *Journal of Computational and Applied Mathematics*, **20**, 53-65. https://doi.org/10.1016/0377-0427(87)90125-7

[53] Morgan, J., *et al.* (1972) Calculation of the Residual Sum of Squares for All Possible Regressions. *Technometrics*, **14**, 317-325. https://doi.org/10.1080/00401706.1972.10488918

[54] Query Repository. https://githubcom/bio2rdf/bio2rdf-scripts/wiki/Query-repository

[55] Querying Bio2RDF Data. http://wwwslidesharenet/alisoncallahan/querying-bio2rdf-data

[56] Hamosh, A., *et al.* (2005) Online Mendelian Inheritance in Man (OMIM), a Knowledgebase of Human Genes and Genetic Disorders. *Nucleic Acids Research*, **33**, D514-D517.

[57] Consortium, U. (2014) UniProt: A Hub for Protein Information. *Nucleic Acids Research*, **43**, D204-D212. https://doi.org/10.1093/nar/gku989

[58] Zhang, Y., *et al.* (2013) An Integrative Computational Approach to Identify Disease-Specific Networks from PubMed Literature Information. *IEEE International Conference on Bioinformatics and Biomedicine*, Shanghai, 18-21 December 2013, 72-75. https://doi.org/10.1109/BIBM.2013.6732738

[59] Zhang, Y., *et al.* (2018) Systematic Identification of Latent Disease-Gene Associations from PubMed Articles. *PLoS ONE*, **13**, e0191568.

[60] Grochow, J.A., *et al.* (2007) Network Motif Discovery Using Subgraph Enumeration and Symmetry-Breaking. *Annual International Conference on Research in Computational Molecular Biology*, Oakland, 21-25 April 2007, 92-106. https://doi.org/10.1007/978-3-540-71681-5_7

[61] Blei, D.M., *et al.* (2003) Latent Dirichlet Allocation. *Journal of Machine Learning Research*, **3**, 993-1022.

[62] Robinson, P.N., *et al.* (2010) The Human Phenotype Ontology. *Clinical Genetics*, **77**, 525-534. https://doi.org/10.1111/j.1399-0004.2010.01436.x

[63] Ashburner, M., *et al*. (2000) Gene Ontology: Tool for the Unification of Biology. *Nature Genetics*, **25**, 25-29. https://doi.org/10.1038/75556

[64] Shen, F., *et al*. (2017) Phenotypic Analysis of Clinical Narratives Using Human Phenotype Ontology. *Studies in Health Technology and Informatics*, **245**, 581-585.

[65] Shen, F., *et al*. (2017) Using Human Phenotype Ontology for Phenotypic Analysis of Clinical Notes. *Studies in Health Technology and Informatics*, **245**, 1285.