Overview Extraction and Visualization of Linked Data Knowledge Graphs for the Cytoscape Platform (Postprint)

Authors: Jiang Ying, Zhang Jing, Zhu Lingxuan

Date: 2017-11-08T00:00:00+00:00

Abstract

[Objective] To facilitate more convenient querying and utilization of massive linked data across various domains, this paper proposes a method for generating knowledge graph overviews of linked data, enabling users to understand the internal data structure of linked data access points before querying. [Method] The method employs SPARQL to query domain knowledge relationships contained in linked data, constructs knowledge graph overview triples for each knowledge relationship to form a preliminary overview, and then extracts knowledge graph overview triples for each knowledge category, merging them with the former to create a complete knowledge graph overview. [Results] A Cytoscape plugin was developed to implement this method, providing visualization functionality for knowledge graph overviews. [Limitations] It cannot handle complex knowledge category extraction such as anonymous nodes. Conclusion In the biomedical domain, three tests were conducted: single-point extraction, linked "bridge", and linked "inclusion". The results demonstrate that the method offers fast and stable extraction, high recall, and does not require web crawlers or additional indexing work.

Full Text

Extracting and Visualizing Knowledge Graph Schema from Linked Data on the Cytoscape Platform

Jiang Ying, Zhang Jing, Zhu Lingxuan
(School of Management, Beijing Normal University, Zhuhai, Zhuhai 519087, China)

Abstract

[**Objective**] To facilitate querying and utilization of massive linked data across various domains, this paper proposes a method for generating knowledge graph schema overviews, enabling users to understand the internal data structure of linked data access points before submitting queries. [**Methods**] Through SPARQL queries, we retrieve domain knowledge relations contained in linked data, construct knowledge graph schema triples for each knowledge relation to form an initial overview, then extract knowledge graph schema triples from each knowledge class and merge them to create a complete knowledge graph schema. [**Results**] We developed a Cytoscape plugin implementing this method and providing visualization functionality for knowledge graph schema overviews. [**Limitations**] The method cannot handle complex knowledge classification extraction such as anonymous nodes. [**Conclusions**] We conducted three tests in the biomedical domain: single-point extraction, association "bridge," and association "inclusion." Results demonstrate that the method offers fast and stable extraction with high recall rates, without requiring web crawlers or additional indexing work.

Keywords: Linked Data, Knowledge Graph Schema, SPARQL, Cytoscape

Classification Number: TP393

Linked Data [1] aims to build a semantic data network that computers can understand, rather than merely a document network readable by humans, to enable more intelligent applications. Linked data is primarily represented using the Resource Description Framework (RDF) [2] format, which describes a resource as a set of triples (subject, predicate, object). SPARQL [3] is a W3C recommended standard that provides a language and protocol for querying and processing content on the Web or in RDF stores. Each linked dataset provides a SPARQL endpoint [4], an HTTP-bound protocol for executing SPARQL queries via HTTP and returning corresponding data. For example, the Linked Open Data [1] project calls for publishing existing data as linked data and interconnecting different data sources. By 2014, 1,014 datasets containing billions of RDF triples had been linked through the Linked Open Data project [5], covering domains such as healthcare, e-commerce, biochemistry, defense, military, and humanities.

Although the formalization advantages of linked data and RDF ontologies enable seamless integration of heterogeneous distributed big data, this formalization requires a format that liberates data from predefined schema constraints. Data defined using this format poses challenges for querying because users cannot rely on pre-existing data mappings for SPARQL queries. While linked data can be queried through SPARQL endpoints, users must know the internal data structure of the linked data—i.e., the knowledge graph schema [6]—before submitting queries for domain-specific data. The knowledge graph schema describes which knowledge classes and object properties the linked data contains, and how these properties connect knowledge instances to form knowledge classes [6]. Only with this understanding can users write SPARQL queries; otherwise, querying massive data becomes impossible. This is analogous to needing to understand a database' s schema before querying it. Therefore, this paper proposes a fast and effective method for extracting and visualizing linked data knowledge graph schema, implemented as a plugin on the Cytoscape [7] platform to lower the barrier for users querying and utilizing massive linked data.

2. Research Status

The extraction and visualization of knowledge graph schema are significant for knowledge engineering, primarily used to assist in perceiving, constructing, and debugging knowledge graph structure and content [8]. Since the introduction of linked data in 2007, research on related algorithms and tools has undergone two stages.

2.1 Knowledge Graph Schema Visualization

Early knowledge graph schema visualization tools, primarily serving ontology designers and builders, were mostly provided as plugins for ontology editors. For example, visualization plugins based on the Protégé [9] ontology editor include TGVizTab [10] developed in 2003, Protégé' s built-in plugin OntoGraf [11], and the Protégé-OWL [12] plugin NavigOWL [13]. Additionally, there are plugins like KC-Viz [15] for the Neon-Toolkit [14] ontology editor. A few tools, such as GrOWL [16], were provided as standalone Java desktop applications (non-plugins). These visualization tools share several characteristics:

(1) Directed Graph Visualization Approach
Most tools map knowledge classes and relations in the schema to nodes and edges to form directed graphs, then display static or dynamic schema diagrams in specific display areas using layout algorithms (e.g., Spring Layout [10] and Power-Law Graph [13]). While alternative approaches exist (e.g., OWLGrEd [17] and OntoViz [18] convert schemas to UML diagrams), directed graphs represent the mainstream approach for knowledge graph schema visualization.

(2) Local Data Visualization Mechanism
To visualize knowledge graph schema, users must import the data into these tools, requiring data to be stored locally and users to have complete datasets. Unless users are the authors of the knowledge graph or have full read permissions, general users cannot perform schema visualization. Although some ontology editors like WebVOWL [19] provide web-based visualization interfaces, complete data must still be uploaded to the server beforehand. In summary, existing tools can only handle complete local data and cannot visualize linked data schema based on SPARQL endpoint queries.

(3) Limitations in Massive Data Visualization
Existing ontology editors typically limit the number of nodes displayed in graph panels to fewer than 10,000 to avoid massive data visualization issues [20]. For instance, analysis reports on the OntSphere [21] tool indicate that slightly over

1,000 nodes cause overlap problems when displaying nodes or labels. TGVizTab and OntoViz cannot visualize data with more than 300 nodes. NavigOWL improved the Power-Law Graph layout algorithm and conducted comparative tests on datasets with 10,000 to 100,000 nodes, showing better performance than traditional Spring Layout algorithms. Nevertheless, these tools, including NavigOWL, struggle to visualize linked data knowledge graphs with millions or tens of millions of nodes.

2.2 Knowledge Graph Schema Extraction

Since the introduction of linked data knowledge graphs in 2007, their distributed and massive data characteristics have rendered early visualization tools inadequate. As increasing domain-specific linked datasets emerged, providing users with methods and tools for visualizing linked data schema became urgent. This shift redirected research focus from visualization to extraction, as while linked data is massive and difficult to visualize directly, most data consists of specific factual information, with relatively little schema-related data. Just as a database may have millions of records but not a massive number of tables and fields, extracting the schema before visualization can significantly reduce the scale of visual content. Based on this idea, researchers have developed methods and tools for linked data schema extraction, primarily in three categories:

(1) Method A: Web Crawler-Based Linked Data Indexing
In 2011, Konrath et al., winners of the Semantic Web Challenge (Billion Triples Track), developed the SchemEX tool [6], which uses crawlers to build real-time indexes for massive linked data and provides schema extraction with linear time complexity. SchemEX features a dynamic indexing mechanism for RDF data streams, loosely coupling extraction from specific crawling methods. However, this tool requires crawling and scanning all linked data and performing additional data indexing; it cannot process data that does not permit crawling.

(2) Method B: SPARQL Queries Focusing on Knowledge Classes
This includes two main approaches: B1 uses rdf:type [22] and rdfs:subclassOf [22] to obtain RDFS-level knowledge class "hierarchy" schema overviews; B2 uses owl:domain [23], owl:range [23], and owl:Class to obtain OWL-level knowledge class "relation" schema overviews. These methods only consider explicit schema information—cases where knowledge class relationships are explicitly defined. However, according to Gottron et al.'s empirical studies on numerous linked data schemas, this approach only captures 63.5% to 88.1% of schema information [25], because linked data often lacks explicit definitions of rdf:type, rdfs:Class, and owl:Class, while implicit schema information appears in almost every linked dataset.

(3) Method C: RDF Graph Summarization-Based Alternative
Zneika et al. [26] proposed a top-k approximation-based RDF graph summarization method that converts massive linked data into knowledge graph schema describing repository content. The drawback is that the resulting schema is

only an “approximation” of the content and structure, with certain errors.

Unlike these three methods, this paper proposes a schema extraction approach that uses knowledge relations as the entry point and relies entirely on SPARQL queries. This method incorporates some ideas from Method B during extraction and adopts the directed graph approach for visualization. Our method offers fast and stable extraction with high recall, requires no web crawling or additional indexing, and avoids missing cases where rdf:type, rdfs:Class, and owl:Class [23] are not explicitly defined.

3. Research Approach and Framework

The knowledge graph schema extraction approach is illustrated in Figure 1 [Figure 1: see original paper]. Our method consists of five main steps:

(1) Query the set of knowledge relations P contained in the linked data. The SPARQL query is:

```
SELECT distinct ?p WHERE { ?s ?p ?o .}
```

(2) Filter out knowledge relations in set P that are in the rdf (http://www.w3.org/1999/02/22-rdf-syntax-ns), rdfs (http://www.w3.org/2000/01/rdf-schema#), and owl (http://www.w3.org/2002/07/owl#) namespaces and are unrelated to domain knowledge, obtaining set P’ . Specifically, for each knowledge relation p in set P, extract its namespace n; if n is rdf, rdfs, or owl, add p to the filtering set Q; finally, remove all elements in Q from P to obtain set P’ .

(3) Construct knowledge graph schema G1 for each knowledge relation p in set P’ .

From the subject and object perspective: Query subjects s and objects o of triples with predicate p, then query their respective knowledge class sets C1 and C2; if both C1 and C2 are non-empty, for each (c1, c2) combination (where c1 C1, c2 C2), construct a schema triple with c1 as subject, p as predicate, and c2 as object, and add it to schema G1. The SPARQL query is:

```
CONSTRUCT { ?class1 <" + p + "> ?class2 } WHERE { ?s <" + p + "> ?o . ?s a ?class1 . ?o a ?c
```

From the predicate perspective: Query object c1 of triples with subject p and predicate rdfs:domain [25]; query object c2 of triples with subject p and predicate rdfs:range [25]; if both c1 and c2 can be retrieved, construct a schema triple with c1 as subject, p as predicate, and c2 as object, and add it to schema G1.

(4) Query knowledge graph triples describing direct relationships between knowledge classes and add them to schema G2. Specifically, query triples with predicates rdfs:subClassOf, owl:equivalentClass [23], owl:complementOf [23], or owl:disjointWith [23]; if neither subject nor object of these triples are anonymous nodes, add them to schema G2.

(5) Merge G1 and G2 into G, which is the final knowledge graph

schema. Specifically, remove duplicate schema triples from G1 to form G1’ ; remove duplicates from G2 to form G2’ ; merge G1’ and G2’ into G’ ; finally, remove duplicates from G’ to obtain the final schema G.

4. Experiments

4.1 Experimental Objects

Given the emergence of numerous queryable structured biomedical linked datasets in recent years, we selected biomedical linked data as our experimental domain. For biomedical researchers, utilizing these massive datasets presents high barriers, particularly regarding user interaction interfaces (SPARQL endpoints) and SPARQL query integration issues, preventing full exploitation of the knowledge classes, instances, and relations contained in the data. Meanwhile, SPARQL syntax complexity and linked data intricacies further increase the difficulty for biomedical researchers. Therefore, we selected several commonly used, freely available biological pathway ontology datasets to assist researchers in querying, retrieving, parsing, and reorganizing biomedical linked data. Specifically, the experimental objects include: Pathway Commons (PC) [27] from the Bader Lab at the University of Toronto, BioCyc [28] with an open academic research license from SRI International, the open-source Reactome [29], HGNC (Human Gene Nomenclature Database) [30] from University College London, and BioModel [31] from the European Bioinformatics Institute.

4.2 Experimental Tools

Cytoscape is an open-source network visualization and analysis platform developed by the NBRB open-source organization. Its core functionality provides basic network layout and querying capabilities, generating visual networks from underlying data. Initially applied in biology, it is commonly used to integrate complex molecular interaction networks with related molecular state information and widely employed for visualizing protein, DNA, and other databases.

Functionally, Cytoscape can display biomolecular interaction networks using different visualization styles, layout networks in two dimensions with multiple algorithm options (including circular and spring-embedded layouts), and enable free browsing, zooming, and panning. It provides bird’ s-eye navigation for large networks (100,000 nodes and edges) with an efficient rendering engine, allows annotation of nodes, edges, and borders with customizable labels and colors, and easily organizes and manages multiple networks, supporting session file saving. However, Cytoscape lacks native linked data visualization and schema extraction capabilities. We developed a linked knowledge graph schema extraction plugin for Cytoscape as our experimental tool for single-point extraction, association “bridge,” and association “inclusion” experiments.

4.3 Single Endpoint Extraction Experiment

(1) Comparative Experiment Across Different Datasets
Figure 2 [Figure 2: see original paper] visualizes the knowledge graph schema for the Pathway Commons [27] linked data endpoint. Circular nodes represent knowledge classes, and arrows between nodes represent knowledge relations, pointing from subject class to object class. Both classes and relations display URI tooltips.

As shown in Table 1 , schema extraction for most SPARQL endpoints completes within 10 minutes, achieving practical utility. After extraction, users can save visualization results for future use without re-extraction.

(2) Comparative Experiment of Different Extraction Methods
Using MeSH (Medical Subject Headings) [28] linked data as the experimental object, we compared our method with Method B (including B1 and B2). Since authors of Methods A and C did not provide downloadable tools, only Method B was analyzed.

Preliminary analysis shows that although MeSH contains over 600,000 triples, it only includes 28 knowledge classes and 152 knowledge relations. It primarily uses its own meshv (http://id.nlm.nih.gov/mesh/vocab#) namespace vocabulary, with partial imports from VirtRDF [29], DC [30], and FOAF [31]. As shown in Table 2 , only 6 and 10 knowledge relations are marked with rdfs:domain and rdfs:range, respectively; only 16 knowledge classes are marked as owl:Class (all in the meshv namespace), with 0 explicitly marked as rdfs:Class. Table 3 shows only 8 knowledge classes marked with rdfs:subClassOf; excluding owl:Thing [23], only meshv:broaderQualifier [28] and meshv:Qualifier [28] explicitly mark subclass-superclass relationships.

As shown in Table 4 , among the 35 schema triples that should be extracted from MeSH, Method B only extracts 14, achieving a recall rate of 40%; our method reaches 94.28% recall. Figure 3 [Figure 3: see original paper] shows that Method B fails to extract schema triples primarily from VirtRDF and unmarked classes in MeSH. Our method successfully extracts them because it follows a bottom-up approach: inferring upper-level class relationships from knowledge relations' associations with lower-level instances, thus avoiding issues from missing rdfs:domain and rdfs:range markings. Our method still missed two schema triples because MeSH used two complex class definitions based on anonymous nodes when defining owl:Thing, preventing our recall from reaching 100%.

4.4 Multiple Endpoints Extraction Experiment

Linked data sources are not isolated. Different data sources may originate from different systems within or across organizations, with varying content, storage locations, and methods, yet still maintain associations. Our Cytoscape plugin can extract schema from multiple linked data endpoints and display their interconnections, including two main scenarios: association "inclusion" and association "bridge."

(1) Association "Inclusion"
As shown in Table 5 , we extracted schemas from two biological pathway datasets (HGNC [32] and MeSH) in approximately 3 minutes. In Figure 4 [Figure 4: see original paper], different shapes represent knowledge class nodes from different SPARQL endpoints. However, the visualization does not show triangular nodes representing MeSH classes because HGNC' s schema includes all MeSH content, with the included portion also displayed as circles.

(2) Association "Bridge"
In Figure 5 [Figure 5: see original paper], the association "bridge" between Pathway Commons, Linkedspl, and BioModel is represented by circular nodes between triangles and squares—these are knowledge classes contained in all three datasets. The enlarged subgraph in Figure 5' s upper right shows the "bridge" is the class "http://www.biopax.org/release/biopax-level3.owl#Pathway" . Users can view each circular node' s "endpoint" attribute to identify its SPARQL endpoints. With this schema visualization, users can perform cross-dataset queries through this association "bridge." Dataset information for these three sources appears in Table 6 .

Conclusion

Our linked data knowledge graph schema extraction method, built on the Cytoscape platform, integrates various schema components through stepwise extraction using a bottom-up approach. Biomedical domain experiments demonstrate that this knowledge-relation-focused, SPARQL-only method offers fast, stable extraction with high recall, without requiring web crawlers or additional indexing. Future work includes: developing algorithms for extracting complex class schemas based on anonymous nodes (e.g., owl:unionOf [23], owl:allValuesFrom [23]) and integrating them into our method; and enhancing the plugin to provide OWL 2 [12] schema visualization functionality.

References

[1] Bizer C, Heath T, Berners-Lee T. Linked Data—The Story So Far[J]. International Journal on Semantic Web & Information Systems, 2009, 5(3): 1-22.

[2] Klyne G, Carroll J J, McBride B. RDF 1.1 Concepts and Abstract Syntax [EB/OL]. (2014-02-25). [2017-01-25]. https://www.w3.org/TR/rdf11-concepts/.

[3] Harris S, Seaborne A. SPARQL 1.1 Query Language [EB/OL]. (2013-03-21). [2017-01-25]. https://www.w3.org/TR/sparql11-query/.

[4] Feigenbaum L, Williams G T, Clark K G, et al. SPARQL 1.1 Protocol [EB/OL]. (2013-03-21). [2017-01-25]. https://www.w3.org/TR/sparql11-

protocol/.

[5] Schmachtenberg M, Bizer C, Paulheim H. Adoption of the Linked Data Best Practices in Different Topical Domains[C]//Proceedings of the 13th International Semantic Web Conference (ISWC 2014), Seattle, USA. Germany: Springer, 2014.

[6] Konrath M, Gottron T, Scherp A. Schemex-Web-Scale Indexed Schema Extraction of Linked Open Data[C]//Proceedings of the 13th International Semantic Web Conference (ISWC 2011) Submission to the Billion Triple Track, Bonn, Germany. Springer, 2011:52-58.

[7] Shannon P, Markiel A, Ozier O, et al. Cytoscape: A Software Environment Integrated Models of Biomolecular Interaction Networks [J]. Genome Research, 2003, 13(11): 2498-2504.

[8] Lanzenberger M, Sampson J, Rester M. Visualization in Ontology Tools[C]//Proceedings of the 2nd International Conference on Complex, Intelligent and Software Intensive Systems (CISIS 2009), Burgos, Spain. New York, USA: IEEE, 2009: 705-711.

[9] Gennari J H, Musen M A, Fergerson R W, et al. The Evolution of Protégé: An Environment for Knowledge-based Systems Development[J]. International Journal of Human-Computer Studies, 2003, 58(1): 89-123.

[10] Alani H. TGVizTab: An Ontology Visualisation Extension for Protégé [C]//Proceedings of the 2nd International Conference on Knowledge Capture (K-Cap' 03), Florida, USA. ACM, 2003.

[11] Falconer S. OntoGraf [EB/OL]. [2017-01-25]. http://protegewiki.stanford.edu/wiki/OntoGraf.

[12] Knublauch H, Fergerson R W, Noy N F, et al. The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications[C]//Proceedings of the 3rd International Semantic Web Conference (ISWC 2004), Karlsruhe, Germany. Springer, 2004: 229-243.

[13] Hussain A, Latif K, Rextin A T, et al. Scalable Visualization of Semantic Nets Using Power-Law Graphs [J]. Applied Mathematics & Information Sciences, 2014, 8(1): 355-367.

[14] Haase P, Lewen H, Studer R, et al. The Neon Ontology Engineering Toolkit[C]//Proccedings of the 17th International World Wide Web Conference (WWW 2008), Beijing, China. ACM, 2008.

[15] Motta E, Mulholland P, Peroni S, et al. A Novel Approach to Visualizing and Navigating Ontologies[C]//Proceedings of the 10th International Semantic Web Conference (ISWC 2011), Bonn, Germany. Springer, 2011: 470-486.

[16] Krivov S, Villa F, Williams R, et al. On Visualization of OWL Ontologies[A]//Baker C J, Cheung K H. Semantic Web[M]. Springer Berlin Heidelberg, 2007: 205-221.

[17] Bārzdiņš J, Bārzdiņš G, Čerāns K, et al. UML Style Graphical Notation and Editor for OWL 2[C]//Proceedings of the 9th International Conference on Business Informatics Research (BIR 2010), Rostock, Germany. Springer, 2010: 102-114.

[18] Sintek M. OntoViz [EB/OL]. [2017-01-25]. http://protegewiki.stanford.edu/wiki/OntoViz.

[19] Lohmann S, Link V, Marbach E, et al. WebVOWL: Web-based Visualization of Ontologies[C]//Proceedings of the International Conference on Knowledge Engineering and Knowledge Management (EKAW 2014), Linköping, Sweden. Springer, 2014: 154-158.

[20] Katifori A, Halatsis C, Lepouras G, et al. Ontology Visualization Methods —A Survey[J]. ACM Computing Surveys, 2007, 39(4): Article No. 10.

[21] Bosca A, Bomino D, Pellegrino P. OntoSphere: More than a 3D Ontology Visualization Tool[C]//Proceedings of the 2nd Italian Semantic Web Workshop (SWAP 2005), Trento, Italy.

[22] Brickley D, Guha R V. RDF Schema 1.1[EB/OL]. (2013-03-21). [2017-01-25]. http://www.w3.org/TR/rdf-schema/.

[23] Bechhofer S, Harmelen F V, Hendler J, et al. OWL Web Ontology Language Reference [EB/OL]. (2004-02-10). [2017-01-25]. https://www.w3.org/TR/owl-ref/.

[24] Grau B C, Horrocks I, Motik B, et al. OWL 2: The Next Step for OWL[J]. Web Semantics: Science, Services and Agents on the World Wide Web, 2008,6(4): 309-322.

[25] Gottron T, Knauf M, Scheglmann S, et al. A Systematic Investigation of Explicit and Implicit Schema Information on the Linked Open Data Cloud[C]//Proceedings of the 10th European Semantic Web Conference (ESWC 2013), Montpellier, France. Springer, 2013: 228-242.

[26] Zneika M, Lucchese C, Vodislav D, et al. RDF Graph Summarization Based on Approximate Patterns[C]//Proceedings of the International Workshop on Information Search, Integration, and Personalization (ISIP 2015), Grand Forks, USA. Springer, 2015: 69-87.

[27] Cerami E G, Gross B E, Demir E, et al. Pathway Commons, A Web Resource for Biological Pathway Data[J]. Nucleic Acids Research, 2011, 39(Database Issue): D685-D690.

[28] Caspi R, Billington R, Ferrer L, et al. The MetaCyc Database of Metabolic Pathways and Enzymes and the BioCyc Collection of Pathway/Genome Databases[J]. Nucleic Acids Research, 2014, 42(Database Issue): D459-D471.

[29] Joshi-Tope G, Gillespie M, Vastrik I, et al. Reactome: A Knowledgebase of Biological Pathways[J]. Nucleic Acids Research, 2005, 33(Database Issue): D428-D432.

[30] Bruford E A, Lush M J, Wright M W, et al. The HGNC Database in 2008: A Resource for the Human Genome[J]. Nucleic Acids Research, 2008, 36(Database Issue): D445-D448.

[31] Novère N L, Bornstein B, Broicher A, et al. BioModels Database: A Free, Centralized Database of Curated, Published, Quantitative Kinetic Models of Biochemical and Cellular Systems[J]. Nucleic Acids Research, 2006, 34(Database Issue): D689-D691.

[32] Erling O, Mikhailov I. RDF Support in the Virtuoso DBMS[A]//Networked Knowledge-Networked Media[M]. Springer Berlin Heidelberg, 2009: 7-24.

[33] Weibel S. The Dublin Core: A Simple Content Description Model for Electronic Resources[J]. Bulletin of the Association for Information Science and Technology, 1997, 24(1): 9-11.

[34] Brickley D, Miller L. FOAF Vocabulary Specification 0.91[EB/OL]. (2007-11-02). [2017-01-25]. http://xmlns.com/foaf/spec/20071002.html.

Author Contributions: Jiang Ying: Conceived research ideas, designed research plan, wrote and revised paper; Zhang Jing: Conducted experiments, collected, cleaned, and analyzed data; Zhu Lingxuan: Developed model system.

Conflict of Interest Statement: All authors declare no conflict of interest.

Supporting Data: The supporting data is self-archived by the authors, E-mail: jpz6311whu@bnuz.edu.cn. [1] Jiang Ying. src.rar. Cytoscape plugin source code.

Received Date: 2017-01-18
Revised Date: 2017-02-14

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv —Machine translation. Verify with original.