

Toward Interoperable Use of RDF and Property Graphs

Hirokazu Chiba¹, Shota Matsumoto², Ryota Yamanaka³

¹ Database Center for Life Science, Chiba 277-0871, Japan

² Lifematics Inc., Tokyo 101-0041, Japan

³ Oracle Corporation, Bangkok 10500, Thailand

Abstract Increasing amounts of scientific and social data are described and analyzed as graphs. As a format of graph data to be published, RDF is widely used. Although the RDF data can be queried using the SPARQL language, even the SPARQL-based operation has a limitation in implementing traversal or analytical algorithms. Recently, a variety of graph database implementations dedicated to analyses on the property graph model have emerged. However, the RDF model and the property graph model are not interoperable. Here, we have developed a framework based on the **Graph to Graph Mapping Language (G2GML)** for mapping RDF graphs to property graphs to make the most of accumulated RDF data. We have also designed a **Property Graph Exchange Format** which can be converted to several formats used in some graph database implementations. Using this framework, graph data described in the RDF model can be converted to the property graph model and can be loaded to several graph database engines for further analysis.

Mapping RDF Graphs to Property Graphs

We have developed a framework based on the **Graph to Graph Mapping Language (G2GML)** for mapping RDF graphs to property graphs to make the most of accumulated RDF data. **Figure 1** shows the overview of the mapping framework. In this framework, users write mappings from RDF graphs to property graphs in G2GML. This mapping can be processed by an implementation called G2G Mapper, which is implemented by authors (available on <https://github.com/g2gml>). This tool retrieves RDF data from SPARQL endpoints and converts them to property graph data into several formats used in some graph database implementations.

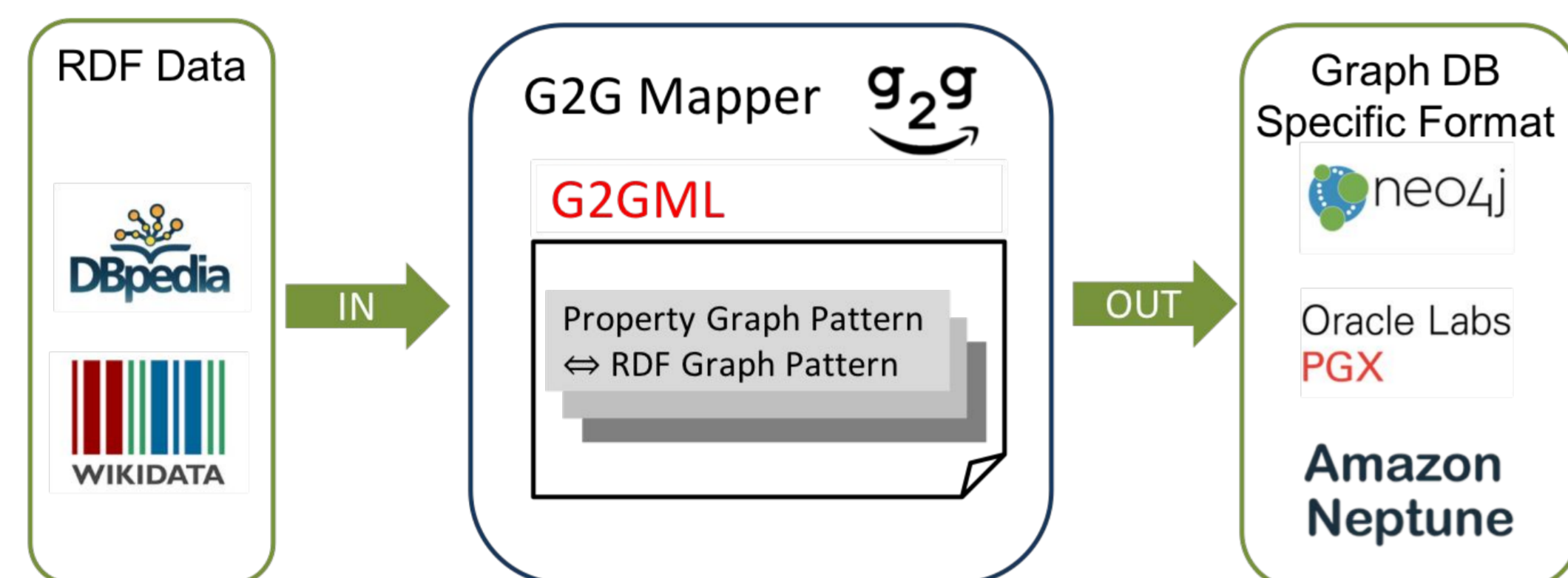


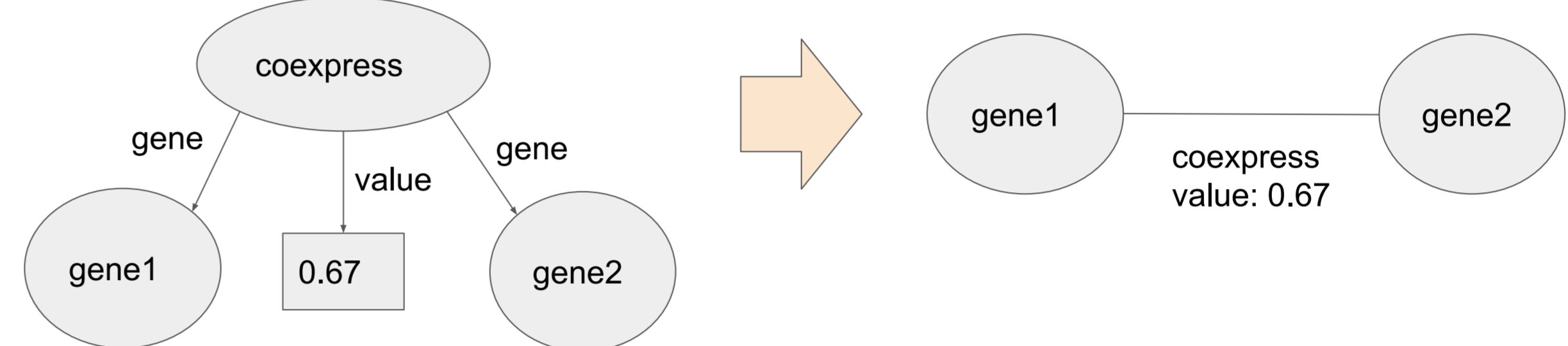
Figure 1. Overview of the mapping framework

```
$ g2g [options] <g2gml_file> <data_source>
$ g2g -f pgx examples/coexpressdb/coexpress.g2g http://coexpressdb.jp/spargl
```

Figure 2. Command-line usage of the mapping framework

Examples

Here we describe an example of human gene coexpression network, and present how to convert RDF graph into property graphs.



Mapping definition in G2GML

```
PREFIX coexpressdb: <http://coexpressdb.jp/rdf/def/0.1/>

# Node mappings
(g:Gene { geneid: id })
<http://coexpressdb.jp/rdf/db/sce.v12> coexpressdb:gene_id ?g .
BIND(strafter(str(?g), "http://bio2rdf.org/geneid:") AS ?id)

# Edge mappings
(g1:Gene)-[:coexpress {mutual_rank: mr}]-(g2:Gene) Create Property Graph
?co coexpressdb:gene_id_1 ?g1 ;
coexpressdb:gene_id_2 ?g2 ; Matching RDF
coexpressdb:mutual_rank ?mr .
```

G2GML is a declarative language which consists of pairs of RDF graph patterns and property graph patterns. An intuitive meaning of a G2GML is a mapping between RDF subgraphs that matches the described patterns and described components of the property graph.

Property Graph Exchange Format

We have designed a **Property Graph Exchange Format** which can be converted to several formats used in some graph database implementations. A simple exemplification of the proposed format is as follows.

```
# NODES
101 :person name:Ryota
102 :person name:Shota

# EDGES
101 -> 102 :knows since:2017
```

To be compliant with various database implementations, we generalize the property graph model as follows:

- A property graph contains nodes and edges.
- Each of nodes and edges can contain properties (key-value pairs).
- Each node or edge can be labeled with **zero or more labels**.
- Edges can be **directed or undirected**.

More formally,

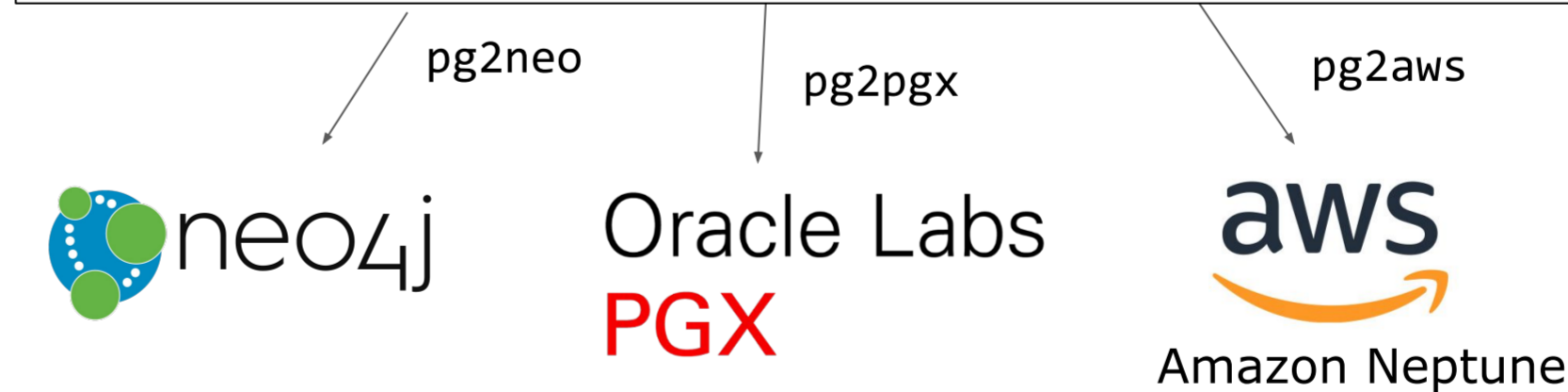
Definition 1 (Property Graph). A Property Graph is a tuple $PG = (N, E_u, E_d, S, V, P, e, l_e, l_e, p_e)$, where:

1. N is a set of nodes,
2. E_u is a set of undirected edges,
3. E_d is a set of directed edges,
4. E is a set of edges where $E = E_d \cup E_u$,
5. S is a set of strings,
6. V is a set of values of arbitrary data types,
7. P contains each properties that has a form $p = (k, v)$, where $k \in S$ and $v \in V$,
8. $e: E \rightarrow (N \times N)$ is a function which yields the endpoints of each directed or undirected edge (if the edge is directed, the first node is a source and the second node is a destination),
9. $l_n: N \rightarrow 2^S$ is a function mapping each nodes to their multiple labels,
10. $l_e: E \rightarrow 2^S$ is a function mapping each edge to their multiple labels,
11. $p_n: N \rightarrow 2^P$ is a function used to assign nodes to their multiple properties.
12. $p_e: E \rightarrow 2^P$ is a function used to assign edges to their multiple properties.

Examples

The Property Graph Exchange Format can be converted to several formats used in some graph database implementations.

```
"http://bio2rdf.org/geneid:850303" :Gene geneid:850303
"http://bio2rdf.org/geneid:850300" :Gene geneid:850300
"http://bio2rdf.org/geneid:851758" :Gene geneid:851758
"http://bio2rdf.org/geneid:851759" :Gene geneid:851759
"http://bio2rdf.org/geneid:851756" :Gene geneid:851756
"http://bio2rdf.org/geneid:851772" :Gene geneid:851772
"http://bio2rdf.org/geneid:851762" :Gene geneid:851762
"http://bio2rdf.org/geneid:850297" :Gene geneid:850297
"http://bio2rdf.org/geneid:1466445" :Gene geneid:1466445
"http://bio2rdf.org/geneid:851762" -- "http://bio2rdf.org/geneid:850297" :coexpress mutual_rank:77.4
"http://bio2rdf.org/geneid:851758" -- "http://bio2rdf.org/geneid:850300" :coexpress mutual_rank:5.8
"http://bio2rdf.org/geneid:851759" -- "http://bio2rdf.org/geneid:1466445" :coexpress mutual_rank:42.7
"http://bio2rdf.org/geneid:851756" -- "http://bio2rdf.org/geneid:850297" :coexpress mutual_rank:66.2
"http://bio2rdf.org/geneid:851772" -- "http://bio2rdf.org/geneid:850303" :coexpress mutual_rank:77.3
```



We have also defined a JSON version of the format, which will be useful for exchanging property graph data through Web APIs.

```
{
  "nodes": [
    {
      "id": "http://bio2rdf.org/geneid:850303",
      "labels": ["Gene"],
      "properties": {
        "geneid": "850303"
      }
    },
    {
      "id": "http://bio2rdf.org/geneid:850300",
      "labels": ["Gene"],
      "properties": {
        "geneid": "850300"
      }
    },
    {
      "id": "http://bio2rdf.org/geneid:851758",
      "labels": ["Gene"],
      "properties": {
        "geneid": "851758"
      }
    },
    {
      "id": "http://bio2rdf.org/geneid:851759",
      "labels": ["Gene"],
      "properties": {
        "geneid": "851759"
      }
    },
    {
      "id": "http://bio2rdf.org/geneid:851756",
      "labels": ["Gene"],
      "properties": {
        "geneid": "851756"
      }
    },
    {
      "id": "http://bio2rdf.org/geneid:851772",
      "labels": ["Gene"],
      "properties": {
        "geneid": "851772"
      }
    },
    {
      "id": "http://bio2rdf.org/geneid:851762",
      "labels": ["Gene"],
      "properties": {
        "geneid": "851762"
      }
    },
    {
      "id": "http://bio2rdf.org/geneid:850297",
      "labels": ["Gene"],
      "properties": {
        "geneid": "850297"
      }
    },
    {
      "id": "http://bio2rdf.org/geneid:1466445",
      "labels": ["Gene"],
      "properties": {
        "geneid": "1466445"
      }
    }
  ],
  "edges": [
    {
      "from": "http://bio2rdf.org/geneid:851762",
      "to": "http://bio2rdf.org/geneid:850297",
      "undirected": true,
      "labels": ["coexpress"],
      "properties": {
        "mutual_rank": "77.4"
      }
    },
    {
      "from": "http://bio2rdf.org/geneid:851758",
      "to": "http://bio2rdf.org/geneid:850300",
      "undirected": true,
      "labels": ["coexpress"],
      "properties": {
        "mutual_rank": "5.8"
      }
    },
    {
      "from": "http://bio2rdf.org/geneid:851759",
      "to": "http://bio2rdf.org/geneid:1466445",
      "undirected": true,
      "labels": ["coexpress"],
      "properties": {
        "mutual_rank": "42.7"
      }
    },
    {
      "from": "http://bio2rdf.org/geneid:851756",
      "to": "http://bio2rdf.org/geneid:850297",
      "undirected": true,
      "labels": ["coexpress"],
      "properties": {
        "mutual_rank": "66.2"
      }
    },
    {
      "from": "http://bio2rdf.org/geneid:851772",
      "to": "http://bio2rdf.org/geneid:850303",
      "undirected": true,
      "labels": ["coexpress"],
      "properties": {
        "mutual_rank": "77.3"
      }
    }
  ]
}
```

Rereferences

Shota Matsumoto, Ryota Yamanaka, Hirokazu Chiba: Mapping RDF Graphs to Property Graphs. arXiv preprint, <https://arxiv.org/abs/1812.01801> (2018)

- <https://github.com/g2gml> G2G project home
- <http://g2g.bio> G2G Sandbox

