Publishing Diachronic Life Science Linked Data

Katerina Gkirtzou[1], Thanasis Vergoulis[1], Artemis Hatzigeorgiou[3], Timos Sellis[2], and Theodore Dalamagas[1]

[1] IMIS, Research Center "Athena", Athens, Greece,
[2] RMIT University, Melbourne, Australia
[3] University of Thessaly, Volos, Greece

Abstract. The Linked Data paradigm involves practices to publish, share, and connect data on the Web. Thus, it is a compelling approach for the dissemination and re-use of scientific data, realizing the vision of the so-called Linked Science. However, by just converting legacy scientific data as Linked Data, we do not fully meet the requirements of data re-use. Scientific data is evolving data. To ensure re-use and allow exploitation and validation of scientific results, several challenges related to scientific data dynamics should be tackled. In this paper, we deal with the publication of diachronic life science linked data. We propose a change model based on RDF to capture versioned entities. Based on this model we convert legacy data from biological databases as diachronic linked data. Our linked data server can assist biologists to explore biological entities and their evolution by either using SPARQL queries or navigating among entity versions. All services are publicly available at http://diana.imis.athena-innovation.gr/lod.

Keywords: microRNA, RDF, LD

1 Introduction

As technology advances in scientific hardware (e.g., sequencers), easing the creation and consumption of large volumes of scientific data, and as more scientific datasets are published and become available for potential data users, the current trend moves towards open science data. Linked Data is a powerful and compelling approach for spreading and consuming scientific data, involves practices how to publish, share, and connect data on the Web, and offers a new way of data integration and interoperability. The driving force to implement Linked Data spaces is the RDF technology[1]. The basic principles of the Linked Data paradigm is (a) use the RDF data model to publish structured data on the Web, and (b) use RDF links to interlink data from different data sources. Linked Data technologies have given rise to the Web of Data: "a Web of things in the world, described by data on the Web" [2].

However, by just converting legacy scientific data to Linked Data (LD), we do not fully meet the requirements of data re-use. To ensure the efficiency in data

[1] http://www.w3.org/RDF/

re-use and allow both the exploitation and validation of scientific results, several challenges related to the dynamics of scientific data must be tackled. Scientific data is diachronic data. Thus, users (*a*) should have access not only to up-to-date scientific LD, but to any of the previous versions and (*b*) should also be able to track the changes among versions, as well as their causes and effects. In this paper, we describe our work on publishing diachronic life science data, and more specifically, genomic, experimental and bibliographic data related to miRNA molecules. We have integrated data from well-known databases, tracked their evolution, and published them as diachronic linked data. We propose a change model based on RDF to capture versioned entities. Our linked data server can assist biologists to explore biological entities and their evolution.

2 Background

MicroRNAs The discovery of microRNAs (miRNAs) in early 2000s has dramatically changed the view of biologists about the role of the so-called junk DNA. miRNAs bind themselves to message RNA (mRNA) transcripts, called targets, and regulate their expression. Knowledge of *miRNA targets* is very important for therapeutic uses. For example, such knowledge could be used to down-regulate genes by introducing artificial miRNAs into the cells. Since the terms "miRNA" is nowadays used in a wide scope, it is common to distinguish between hairpin miRNAs and mature miRNAs, or just hairpins and matures from now on. The former signifies a precursor of the latter. A hairpin can actually be processed into several matures. Matures bind themselves to transcripts and prevent the creation of functional ribosomes.

Diachronic miRNA data During the past years, we have developed *DIANA tools*[2], a set of advanced Web applications supporting a series of sophisticated work-flows enabling users with no strong background in informatics to perform advanced multi-step functional miRNA analysis. Under the hood of DIANA tools lies a relational database storing information for key entities of the miRNA domain, such as hairpins, matures, transcripts, genes, KEGG pathways, and publications. Information has been aggregated from a variety of sources, i.e information concerning the miRNAs comes from the miRBase database (up to v.18)[3], gene related information from the Ensembl database (v.69)[4] and molecular interaction and reaction networks from KEGG pathways database[5].

The miRBase database is a searchable database of published miRNAs and maintains information for 18,589 hairpins and 21,881 matures. It also maintains a list of files that record successive versions along with the changes between them. By examining all these files, a number of changes are observed. Analytically, we observe the following changes: (*a*) NAME/SEQUENCE : change

[2] http://diana.imis.athena-innovation.gr/DianaTools/index.php
[3] http://www.mirbase.org/
[4] http://www.ensembl.org/index.html
[5] http://www.genome.jp/kegg/pathway.html

over the values of name and sequence respectively for both hairpins and matures, (*b*) NEW/DELETE : creation and deletion of a hairpin/mature entry, (*c*) FORWARD : merging of multiple hairpin entries into a single one, (*d*) ADD HAIRPIN PARENT/REMOVE HAIRPIN PARENT : creation/deletion of the relationship among a pair of hairpin and mature. Being able to track the changes of the miRNA and having access to older versions it is a crucial information when working with miRNAs. For example, the hairpin MI0001364 was introduced into miRBase at v5.0 with name dre-mir-10b and at v7.0 its name changed to dre-mir-10b-1. Having access to all possible names, allows us to better track bibliographic references independent of its latest name.

3 Change model based on RDF

A challenging problem in LD publishing is how to deal with evolving data. In the general case, changes are complex objects that interact with data objects, they can have structural, temporal and probabilistic characteristics, thus they cannot be handled as plain differences neither transformation operations between datasets. This is evident especially in life science LD, since by just converting legacy scientific data as LD, we do not fully meet the requirements of data reuse. To support the above requirements, we adopt versioned RDF entities and version properties, and we also model changes as first class citizens, i.e. RDF resources, in the LD itself. Based on our approach, both query and navigation services can be developed on top of diachronic LD sets.

3.1 Versioned entities and properties

To represent up-to-date RDF descriptions, we use a general URI that is based on the class of the resource and an id that uniquely identifies the resource: **http://.../resource/class-name/resource-id**. By using such URIs, one can retrieve the RDF description for the current version of a requested RDF resource. To retrieve the RDF description for an RDF resource in a previous version of the LD set, we use version timestamps to append the general URIs: **http://.../resource/class-name/resource-id/timestamp**. For example, the URI `http://.../resource/matures/MIMAT0009477` retrieves the current version of the mature with id MIMAT0009477, while `http://.../resource/matures/MIMAT0009477/17` the description of the same mature for v17.

While versioned URIs support the navigation among versioned entities by following those URIs, we use four version properties to facilitate the querying of diachronic LD set: (*a*) `:label`, with the constant value `"now"`, (*b*) `:version`, with version timestamp values, (*c*) `:previousVersion`, and (*d*) `:nextVersion`. The first property can be used in SPARQL to provide access only to the current version of the RDF resources, while the second can be used to form queries that span several versions of the LD set. The other two properties are used to ease the navigation among different resource's versions back and forward, respectively.

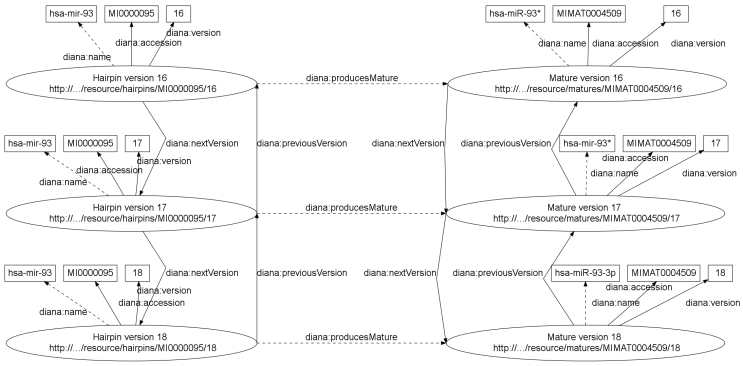

Fig. 1. An RDF subgraph of the miRNA LD showing versioned RDF resources of the hairpin miRNA MI0000095 and its produced mature miRNA MIMAT0004509 for the miRBase v16, v17 and v18.

Figure 1 presents an RDF subgraph from the miRNA LOD published using our proposed method. At the left column from top to bottom, it shows three RDF resources of the hairpin MI0000095 for miRBase versions 16, 17 and 18. Similarly at the right column, it shows the RDF resources of the mature MIMAT0004509 for the same miRBase versions. The mature MIMAT000450 is produced by hairpin MI0000095, a relationship that is reflected in the RDF graph via the property `diana:producesMature`. Figure 1 also shows some of the properties related with the Hairpin and Mature class, such as `diana:name` and `diana:accession`. The property `diana:name` represents the name of each miRNA, a property that evolves over time, while `diana:accession` representing the miRBase id of each miRNA, which remains unchanged over their lifespan.

3.2 Change Management

A key requirement for diachronic LD is the need to trace the origins of data and transformations that occurred. Capturing the changes themselves and having querying capabilities on them, actually give us the possibility to step back and examine how and when the changes took place. In other words, it opens the key challenging problem of preservation and provenance support. Our approach uses the RDF schema itself to represent changes. We propose two different strategies depending whether the change occurs in an RDF triple that involves a property with a literal object or a property connecting two resources.

Property to Literal In this case, the literal value is substituted by a new value. We model this change by adding to the RDF schema an extra property `:change`, which is a property to literal and describes the type of change. Figure 2 shows an RDF subgraph of the miRNA LOD, where a change occurs to the property `diana:name` of the hairpin MI0000069 at v2.0. On the left side, we see its RDF resource at the previous miRBase v1.5, which also contains the old name value

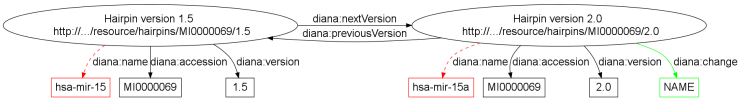

Fig. 2. An RDF subgraph of the miRNA LD showing how a change in property `diana:name` (property to Literal) is modeled.

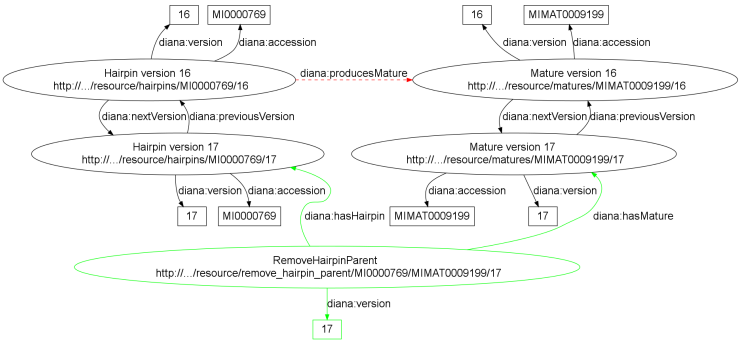

Fig. 3. An RDF subgraph of the miRNA LD showing how a change in a property connecting two resources, such as `diana:producesMature`, is modeled.

`hsa-mir-15a` (depicted in red), while on the right side we see the RDF resource of the hairpin at miRBase version 2.0 with the new name value `hsa-mir-15a` and the extra property `diana:change` with literal value the type of change –in this case `"NAME"`– that models the change itself (depicted in green).

Property to Resource In this case, the change is an RDF resource itself that links to the two resources of the property (the subject and the object) that changed. This "change" resource apart from links to the corresponding resources, it contains also a time property, showing at which time point the change occurred. Other characteristics, like probabilistic ones, can easily be modeled by adding extra properties to the "change" resource. Figure 3 shows an RDF subgraph of the miRNA LOD when a change occurs at property `diana:producesMature` between the hairpin MI0000769 and the mature MIMAT0009199 at v17. At the top of the Figure 3 we see the two connected RDF resources at the previous v16 (depicted in red). At the bottom, we see the two resources at v17 where the property `diana:producesMature` has been dropped. To proper model proper this change, we extend the RDF graph with an instance of the class `RemoveHairpinParent` (depicted in green) that contains links to the respective resources at v17.

4 Publishing and exploring LD miRNA data

All the curated information related to miRNAs for DIANA tools is stored in a relational database, but a Linked Data representation is necessary and useful for biologists. Such a LD representation will be open and available to everyone for re-use, exploration and validation. Moreover, it can be used for inference and can be potentially be part of an automated interpretation tool of the role of non-coding RNAs. Due to the internal relational storage of the data, we adopt the "virtual RDF" approach: accessing a non-RDF database using an RDF view. Such an approach enables the access of non-RDF, legacy databases without having to replicate the whole database into RDF. We use the D2R server [3], a popular tool that follows this approach.

In order to show the abilities of our proposed model and its flexibility, we present a number of SPARQL queries. The queries are categorized into the following types: (*a*) retrieval of up-to-date entities, (*b*) retrieval of diachronic entities and (*c*) change exploration. The retrieval of up-to-date entities requires the triple pattern `?e diana:label "now"` as mentioned at Section 3.1. The following SPARQL query is an example of this type, retrieving the targets of miRNAs predicted by microT-ANN and showing the whole biogenesis path. Thus, we get the hairpin's URI, the URI of its produced mature and the transcript's URI that is targeted by the mature.

```
SELECT ?hairpin ?mature ?target WHERE
{
   ?hairpin rdf:type diana:Hairpin.
   ?hairpin diana:label "now".
   ?hairpin diana:producesMature ?mature.
   ?interaction rdf:type diana:Interaction.
   ?interaction diana:hasMature ?mature.
   ?interaction diana:hasTarget ?target.
   ?interaction diana:application "microT-ANN (v4.0)".
}
```

To retrieve diachronic data we need just to specify the period of interest. For example, if we want to reproduce the previous query, but for a specified previous miRBase version, e.g version 13.0, we have the following SPARQL query :

```
SELECT ?hairpin ?mature ?target WHERE
{
   ?hairpin rdf:type diana:Hairpin.
   ?hairpin diana:version "13.0".
   ?hairpin diana:producesMature ?mature.
   ?interaction rdf:type diana:Interaction.
   ?interaction diana:hasMature ?mature.
   ?interaction diana:hasTarget ?target.
   ?interaction diana:application "microT-ANN (v4.0)".
}
```

Finally we present two examples of SPARQL queries that show the evolution of data. The first one retrieves the hairpins that changed their name throughout their lifespan and returns the hairpin's URI, the miRBase version the name changed, the old name value and the new name value:

```
SELECT ?hairpin ?version ?old_name ?new_name WHERE
{
    ?hairpin rdf:type diana:Hairpin.
    ?hairpin diana:change "NAME".
    ?hairpin diana:version ?version.
    ?hairpin diana:name ?new_name.
    ?hairpin diana:previousVersion ?hairpin_prev.
    ?hairpin_prev diana:name ?old_name.
}
```

In order to retrieve the old name values of the Hairpins, we can access the previous version of the hairpin via the property `diana:previousVersion`. The second one retrieves all currently not-related hairpin and mature pairs that used to be related in the past and returns from left to right the hairpin's URI, the miRBase version the property `diana:producesMature` dropped and the mature's URI :

```
SELECT DISTINCT ?hairpin ?version ?mature WHERE
{
   ?connection rdf:type diana:removeHairpinParent.
   ?connection diana:version ?version.
   FILTER (?version <= "18").
   ?connection diana:hasHairpin ?hairpin.
   ?connection diana:hasMature ?mature.
}
```

Using the property `diana:version` in `FILTER` we can retrieve all changes in the relationship between hairpin and mature miRNAs before that version. Note also that since the property `diana:producesMature` connects two resources, a change in that relationship-type property we model it as a new RDF class.

5 Related Work

This work is an extension of [5], which was our first attempt to publish legacy miRNA data as LD. Compared to that work, (*a*) we use a different database schema in order to support a wider range of diachronic queries, and (*b*) we have integrated new miRNA-related datasets. Various approaches have been proposed in the bibliography studying the problems of evolution, versioning and change detection in the context of LD. The Memento framework [8] handles different versions of LD by attaching time-specific attributes as global version indicator. [9] focuses on monitoring and examining the dynamics of LD, a knowledge that can be leveraged to optimize existing systems and algorithms, especially indexing techniques. Other publications, such as [7] and [11], deal with changes in the linkages between datasets and the problem of broken links. A comparative study of tools and approaches that deal with the problems of LD dynamics has also been presented in [10]. The survey studies the aspects of discovery, change detection at several granularity level (i.e at the dataset level, at triple level, etc), and description of the changes, as well as the detection algorithms and notification mechanisms. Evolutionary Terminology Auditing (ETA) examines how terminology changes reflect evolutions in the underlying domain. An example

of this work can be seen in the context of SNOMED CT in [1]. Finally, named graphs [4], RDF graphs which are assigned a name in the form of a URIref, allow easier provenance tracking and trustness warranties.

6 Conclusion - Future Work

In this paper we presented our work on the publication of diachronic life science LD. More specifically, we aggregated data from well known databases related to miRNA molecules and exported as LD. The curated data contains information from experimental observations, as well as change and version information about the miRNA entities. The miRNA LD server can assist biologists to explore biological entities, navigate between versions of the resources and via the SPARQL endpoint it provides access to applications for querying historical miRNA data and tracking their changes. As future work, we plan to interlink our linked data set with the linked data provided by the EBI RDF Platform [6]. Futhermore, we will focus on providing keyword search querying capabilities on LD.

Acknowledgments This study has been supported by LODGOV project, Research Programme ARISTEIA (EXCELLENCE), General Secretariat for Research and Technology, Ministry of Education, Greece and the European Regional Development Fund.

References

1. Applying Evolutionary Terminology Auditing to SNOMED CT. In: AMIA 2010 Proceedings. pp. 96–100 (2010)
2. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. International Journal on Semantic Web and Information Systems 5(3), 1–22 (2009)
3. Bizer, C., Cyganiak, R.: D2r server publishing relational databases on the semantic web (2006)
4. Carroll, J.J., Bizer, C., Hayes, P., Stickler, P.: Named graphs. Web Semantics: Science, Services and Agents on the World Wide Web 3(4), 247 – 267 (2005)
5. Dalamagas, T., Bikakis, N., Papastefanatos, G., Stavrakas, Y., Hatzigeorgiou, A.G.: Publishing life science data as linked open data: the case study of mirbase. In: WOD. pp. 70–77 (2012)
6. Jupp, S., Malone, J., Bolleman, J., Brandizi, M., Davies, M., Garcia, L., Gaulton, A., Gehant, S., Laibe, C., Redaschi, N., Wimalaratne, S.M., Martin, M., Le Novre, N., Parkinson, H., Birney, E., Jenkinson, A.M.: The EBI RDF Platform: Linked Open Data for the Life Sciences. Bioinformatics (2014)
7. Popitsch, N., Haslhofer, B.: Dsnotify: handling broken links in the web of data. In: WWW. pp. 761–770 (2010)
8. de Sompel, H.V., Sanderson, R., Nelson, M.L., Balakireva, L., Shankar, H., Ainsworth, S.: An http-based versioning mechanism for linked data. CoRR (2010)
9. Umbrich, J., Hausenblas, M., Hogan, A., Polleres, A., Decker, S.: Towards Dataset Dynamics: Change Frequency of Linked Open Data Sources. In: LDOW (2010)
10. Umbrich, J., Villazn-Terrazas, B., Hausenblas, M.: Dataset dynamics compendium: A comparative study. In: COLD. vol. 665 (2010)
11. Volz, J., Bizer, C., Gaedke, M., Kobilarov, G.: Discovering and maintaining links on the web of data. In: ISWC 2009. pp. 650–665 (2009)