Conceptual Knowledge Management: the Case of Biomedical Information

Roberto Enea[1], Maria Teresa Pazienza[2], Andrea Turbati[2], Davide Vitiello[2]

[1] Guglielmo Marconi University
r.enea@unimarconi.it
[2] DII, ART Group, University of Rome, Tor Vergata
{pazienza, turbati}@info.uniroma2.it
DavideVitiello@mail.com

Abstract.

Ontologies are commonly used resources: we are witnessing to the constant grow, in number and heterogeneity, of communities working with large volumes of data. Researchers, practitioners, developers and end users deal with a huge amount of data from different perspective, topics, cultures, languages. For people involved in governing both data and processes this remains a difficult task. End users and practitioners are usually interested in merging, data generated from connected objects in customizable ways. Even if a lot of algorithms and tools have been created to achieve such goal in a full automatic manner, human contribution is nonetheless still important. On the way to reach the aforementioned results, researchers concentrated their effort in making easier both representation and visualization of data, thus simplifying user interaction with the system. The main aspect to deal with is how to represent an ontology alignment providing a good overview of the alignment and whatever is recognized to be an important detail for the user, both at the same time. Indeed, the overall view makes easier for the users to be aware of the level of interconnection between ontologies, while details allow them to evaluate the alignments and, possibly, give feedback on each of them. These two levels of information's combined requirements suggests that a static visualization of the alignment is not effective. To enhance human involvement and judgment, interactivity becomes a key issue. These tasks become much more complex when considering very large ontologies like biomedical ones. We have developed a framework for supporting easy ontology matching processes (**GENOMA**) that provides a user friendly graphical interface (***OntoChord***) to visualize structured information and reason over them. Framework's technical details will be outlined.

Furthermore, examples of user sessions and testing evaluations over biomedical ontologies will be considered.

Keywords. ontology, knowledge management, data visualization

1. Introduction

Life science involves the scientific study of living organisms: from plants, to animals and human beings. Overall, the study of organisms' behaviors (that includes not only biology, but also all related technological advances in biotechnology) led to a flourishing of specializations and new, interdisciplinary, research fields. As a consequence of all those activities, the amount of data produced has become impressive and hardly manageable in a unique systematic assessment. Several ontologies were defined as an attempt to make knowledge underlying biological and medical experiments suitable for sharing. Most of them are freely accessible.

This attempt led to several outcomes; one important research activity focuses on the possibility to structure all those heterogeneous contributions in a unique conceptual framework. Thus the interest in merging different ontologies to approach a unique knowledge representation.

Data dimension represents a further problem for both structuring and relating data in a unique context meant for end-user visualization.

Starting from such premises we developed a large framework in which the two actions (ontology matching and visualization) were feasible for any kind of user.

In order to do that, data representation and visualization was certainly one key issue. Indeed, we wanted to simplify user interaction with the automated system to achieve a better understanding of phenomena. Although this may seem a secondary aspect of ontology matching, it is nevertheless a difficult one to deal with, especially when involved ontologies reach considerable size (thousands of concepts) and consequently does their alignment. The main obstacle to overcome is to provide a good overview of the alignment without sacrificing important details. While the overall view shows the user the level of interconnection between compared ontologies, details allow to evaluate alignments and provide feedback on each of them. Having said that, a static visualization of the alignment is not effective. To enhance human involvement and judgment, interactivity becomes a key issue.

Our approach has been based on an interactive visualization method of ontology matching results (***OntoChord***). The functionalities of the tool are not limited to the visualization of the alignment but also let the user return feedbacks on the alignment process. In the first part of the paper we introduce the main problems regarding ontology alignments' visualization (while presenting a brief state of the art).

The second part is dedicated to the presentation of the tool ***OntoChord*** and the framework upon which it runs (**GENOMA**). Then we present some Chord Graphs generated by alignments via different matching architectures.

Finally, in the last section we will discuss about the system's future developments.

2. Visualization of Ontology Alignments in literature

When presenting an alignment between ontologies we need to find the right balance between overview and detail. Both aspects of the alignment cannot be omitted. If on the one hand the overall view gives the user the level of interconnection between the compared ontologies, the detail allows him to evaluate and then provide feedback on each alignment. The combined needs of these two levels of detail causes a static visualization of the alignment is not such a good method. Interactivity becomes a key issue.

Following previous concept is useful to present the state of the art of the visualization tools highlighting the different approaches used.

Tree view is one of the most used methods. COMA++ (Aumueller, Do, Massmann, & Rahm, 2005), an ontology matching tool, represents the alignments as chords connecting entities between two ontologies. Tools following the same approach are AgreementMaker (Li, Stroe, & Cruz, 2015) and PROMPT (Noy & Musen, 2003). AgreementMaker combines the tree view visualization with the list of matching results and PROMPT, a Protégé plugin, visualizes the alignment in a tree view fashion, showing alignments as an interactive list such that the selection of a list item highlights the entities aligned in the tree view.

The advantage of the approach used by the tools above is showing each entity in its context inside the belonging ontology. The disadvantage is not being able to scale when we are dealing with very dense alignments or with large ontologies: the interpretation of the graph could thus become difficult for the user.

Other tools adopt simple graphs instead of trees, to address the issues connected to the tree view (or they add a graph to the classical tree representation). AlViz (Lanzenberger & Sampson, 2006) uses both a tree and "small world graphs" (Ham & Wijk, 2004) to help the user to examine the structure of the ontology intuitively. This method uses clusters to group the nodes of a graph according to the selected level of detail. The nodes represent the entities (concepts or instances) connected to each other, according to the selected relations.

3. *OntoChord*, a visualization tool for ontology alignments

OntoChord has been developed as part of a broader framework dedicated to the Ontology Matching, called **GENOMA** (Enea, Pazienza, & Turbati, 2015) (**GEN**eric **O**ntology **M**atching **A**rchitecture, which can be downloaded from https://bitbucket.org/aturbati/ontologymatching-architecture). It supplies a user friendly graphical tool to build different architectures using existing matchers. The system is already provided with several matchers (string-based, language-based, graph-based) and can be extended and customized by end users. **GENOMA** generates the final alignment report in RDF format according to the OAEI standard (Ontology Alignment Evaluation Initiative, http://oaei.ontologymatching.org/), which is a filtered similarity matrix.

Even if ***OntoChord*** has been designed as part of the framework **GENOMA**, it can be used also as a stand-alone tool.

As previously mentioned, the main problem to be addressed in the visualization of ontologies alignments relates to the possibility to find the right balance between overview and details. In a process of ontology matching each entity of the first ontology is compared with each entity of the other, thus generating a two-dimensional matrix. Alignment between two ontologies may be considered as the filtering of more complex data: the similarity matrix resulting from the matching operation. We adopted as a model for the representation of alignments between ontologies one of the most used diagrams in complex adjacency matrix and networks visualization (Komarek, Pavlik, & Sobeslav, 2015), the chord graph, better known as chord diagram.

The chord diagrams (Krzywinski, et al., 2009) became popular in 2007 after their use in representing similarities between the human genome and the one of other spe-

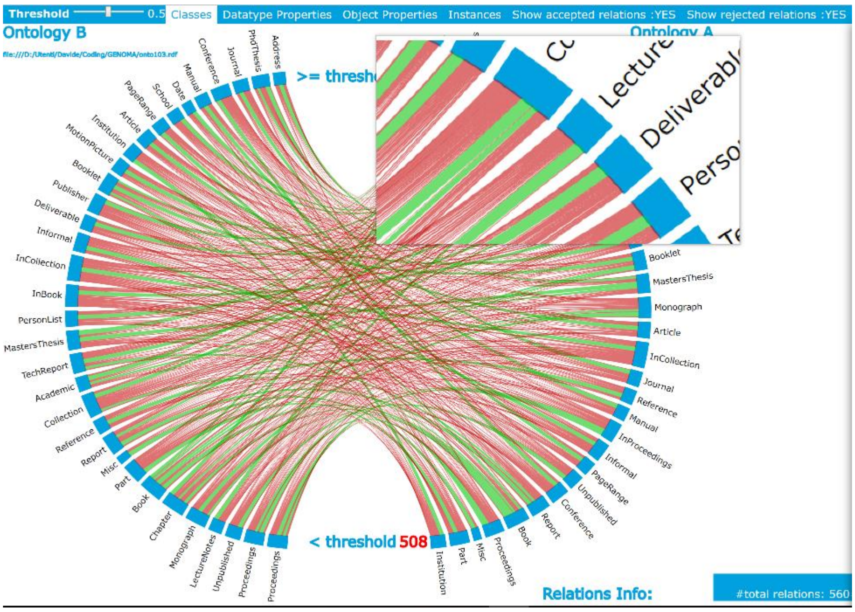


Figure 1 Chord Graph shown inside the ***OntoChord*** GUI (with a focus on a part of right arch)

cies. The diagram consists of a circumference around which related entities are arranged. The relationships between entities are represented by chords.

4. The *OntoChord* tool functionalities

OntoChord receives a set of filtering settings input parameters. For example, it is possible to filter the entities that we want to be included in the diagram, like classes, datatype properties, object properties or instances. The filter flags are defined during the composition of the architecture in the Engine Configurator tool. **GENOMA** generates an XML output file including the similarity matrix and the names of the entities, together with the URIs of the matched ontologies.

OntoChord graphical interface (Figure 1) makes the alignment readable also to non-experts through the chord graph. We structured the GUI trying to focus on the chord graph functionality for viewing relations and saving the user feedback. We also made it possible to change the cut-off threshold: this functionality is of vital importance to the phase of fine tuning. ***OntoChord*** was developed using web technologies (HTML, javascript[1], JSP, RDF4J) and can be accessed using a common browser. It gets as input the full URL of the XML file generated by a dedicated module in **GENOMA.**

This feature makes the graphical interface of the ***OntoChord*** completely independent from the framework **GENOMA**: it can be used by any other tool that can populate the XML file with the required information.

The chords in the graph are colored in green or in red depending on the similarity value they represent; the green chords are the ones that overcome the threshold while the red ones stay below the threshold. In such a way, the user has a glance at existing similarity relations between the two ontologies and how many of them may be valid.

[1] We used D3.JS (d3js.org) for the creation of the chord graph and hierarchy graphs. The overall functions are built on native JavaScript and D3.JS.

The threshold is dynamic and can be modified by the user acting on a slider bar (see Figure 1) The presence of a dynamic threshold is a very important aspect of the analysis phase because it helps the user to assess the best cut-off threshold to be applied. The importance lies in the fact that usually the cut-off threshold must be adapted to the architecture used for matching and therefore is part of the fine-tuning phase of the system. There are no effective thresholds in advance.

To attain interactivity, mouse events are used to highlight the details of the graph in a very comfortable way. For example, it is possible to select a single chord or, to select only the similarity relations existing between that entity and the others entities of the opposite ontology, respectively by moving the mouse pointer over the graph until reaching the chord, or by moving the mouse pointer over one entity in the semi-arc. Once a relation is selected, the user can accept or reject the alignment suggested by the system for that relation.

OntoChord can provide three different diagrams depending on how the filtering flag is set:

- class comparison,
- property (datatype and object) comparison and
- instance comparison.

In the graph interface menu, it is possible to select which diagram is to be shown. The similarity matrix has been generated through **GENOMA**. By combining **GENOMA** and ***OntoChord*** changing matchers to get new results visualized becomes very easy and fast for the end user.

OntoChord is an innovative tool which visually exposes ontologies matchings while providing further support to user for a better understanding of interconnections.

The philosophy behind ***OntoChord*** is that once an ontology matching has been computed the human agent should be able to go through the entities deemed similar by the matching algorithm. Then, he should be allowed to accept or reject the similarity relations encompassing the entities. When aligned ontologies are big and when the alignment is very dense, going through this process can be difficult and slow.

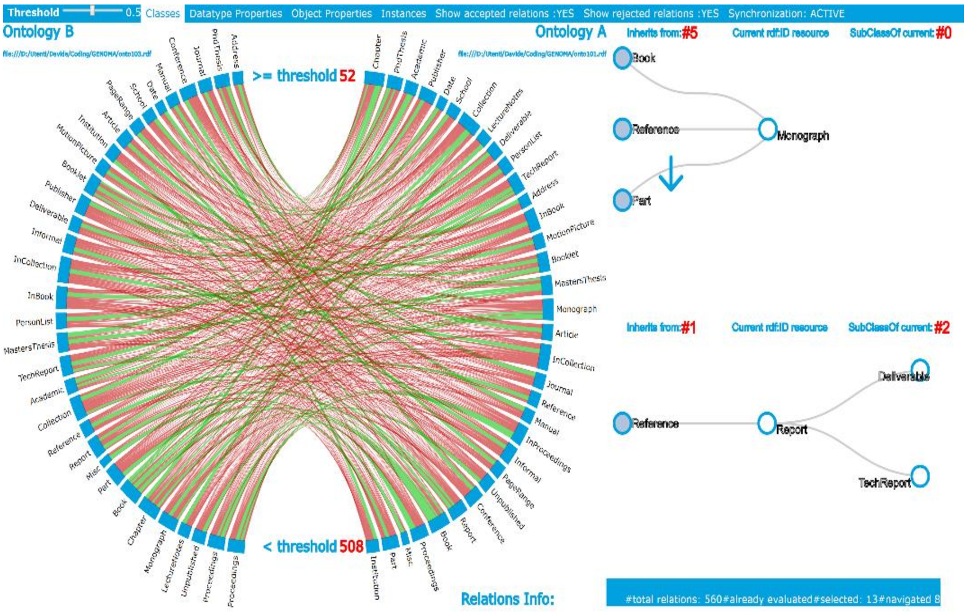
It should also address density issues, in the sense of presenting all the similarities relations in the clearer fashion possible, without, however, hiding details.

The capability of ***OntoChord*** tool can be assessed from different perspectives.

- Raw visualization of alignment relations
- Interaction with relations
- Visualization of hierarchy
- Interaction with hierarchy graph

So far ***OntoChord*** has evolved from being a convenient alignment tool to a detail-oriented inspection and modification tool for ontologies and matching results.

Indeeed, while a simple syntactic matching task, whereby two classes must be considered similar if they share the first n letters of their labels does not need any human intervention, more complex semantic matchings probably needs corrections.



Threshold 0.5

Show accepted relations :YES Show rejected relations :YES Synchronization: ACTIVE

Ontology B file:///D:/Utenti/Davide/Coding/GENOMA/onto103.rdf

Ontology A file:///D:/Utenti/Davide/Coding/GENOMA/onto101.rdf

>= threshold 52

< threshold 508

Inherits from:#5 Current rdf:ID resource SubClassOf current:#0

Inherits from:#1 Current rdf:ID resource SubClassOf current:#2

Relations Info: #total relations: 560#already evaluated#selected: 13#navigated 8

Figure 2 Chord graph (on the left) and hierarchical view (on the right). While the Chord Diagram is immediately available as OntoChord runs, hierarchy graphs ‘spawn’ after the user selects and clicks on one of the relations, which are graphically chords.

The functions immediately available as the tool opens acts upon the relations shown as Chord Graph’s chords. A secondary set of functions, which acts upon single ontologies hierarchies, becomes available through click events on chords.

4.1. Class Hierarchy in ***OntoChord***

The Chord representation is only one way to see and understand the results of the matching process. The limit of such depiction is that it is not able to show the context of the classes/concepts that were matched.

An ontology (and a thesaurus as well) places each resource in a hierarchy through the relations between them.

In the absence of this kind of representation, the user might have a hard time validating the alignment since he may not be able to fully understand the meaning of a class/concept.

Such a limitation can be overcome through:

- The hierarchical view of both ontologies (one centered in an ‘A’ ontology’s resource and the other in a ‘B’ ontology’s resource)

- The value associated to both resources in respect to both the original ontologies (such as labels, descriptions, notes) and aligment (such as resources’ properties similarities, which will be subsequently shown).

An example of the hierarchical view can be seen in Figure 2.

We now describe more in details the features of ***OntoChord***, providing some explanatory examples.

4.1.1. Static features

CutChordGraph makes use of a simple diagram to show all the entities deemed similar by the matching algorithm in use.

The problem underlying such a visualization is the cost-benefit ratio of the overall view on the detailed one.

4.1.2. Dynamic features

The dynamic features of CutChordGraph are:

- Zooming (see Figure 1)
- Adjustable similarity acceptance threshold (see Sect. 4.1.2.1)
- Chords Acceptance/Rejection (see Sect. 4.1.2.2)
- Accepted / Rejected Chords filter (see Sect. 4.1.2.3)

As the alignments become numerous CutChordGraph can be zoomed into internal arches representing the entities.

4.1.2.1. Use of the acceptance threshold

The threshold adjustment (Figure 3) is meant to help the final user in speeding up the validation process: by changing the acceptance threshold the distribution of green and red chords changes accordingly. This is meant as a starting point, in that the next steps are manual, user-driven acceptances or rejections of single chords (relations). The feature is especially useful when considering the big size of some ontology matchings. In a sense, setting the threshold is a first pass, which can automatically set a wide group of relations as accepted or rejected.

From another point of view, the threshold helps the user to visually spot relations with marginal values, evidencing most probably similar and dissimilar relations.

After accepting or rejecting a chord the related relation will be green or red due to its similarity value been set to 1 or 0 respectively. This prevents accepted and rejected chord color changes as the threshold is being readjusted.

The value can then be reset to its original value at will and CutChordGraph will keep track of all accepted, rejected and reset relations during the whole session. This can be especially useful when relations are many and the updating task spans over several sessions.

4.1.2.2. Acceptation of chords

Recording feedback from the user is another important feature of the tool. It can be done by clicking on a single relation. That switches the current acceptance value of the relation assigning to it the minimum or maximum value coloring the relative chord accordingly. Despite the change, the original value is memorized by OntoChord.

Acceptation, rejection and reset of chords is done by hovering them and clicking **a, r** and **s .**

4.1.2.3. Filter on chords

To better work on big alignments, CutChordGraph implements a filter for accepted and rejected chords, in order to free the view from unnecessary relations.

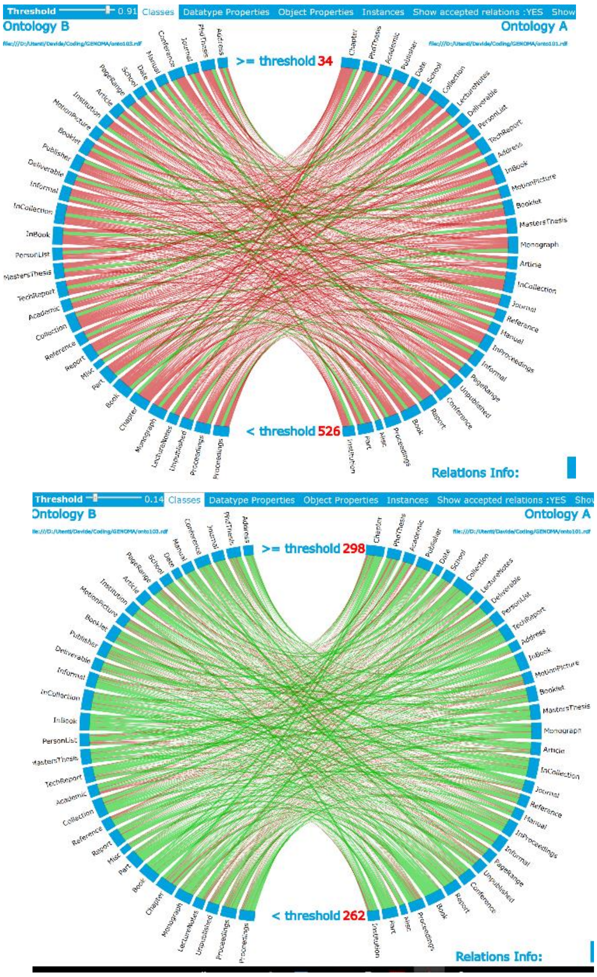


Figure 3 Threshold can be adjusted to spot most probably similar relations

4.1.3. Hierarchy Graphs

The features comprising all the interactions with the single ontologies are not immediately available and can be accessed by hovering and pressing **g** over a relation.

That will spawn the hierarchical graphs of all the entities of ontologies A and B. Due to the graph dimensions and the fact that end users can be interested in deeply understand interrelations among a few specific concepts, only a subgraph of the hierarchies is initially shown: the entity nodes that make up the selected relations, its parent entities and its child entities.

The overall hierarchy is totally available and different nodes can be inspected both through the graphs itself when an adjacent node is clicked or by pressing **g** over a different relation of the CutChordGraph. While the hierarchy is meant to aid the user in gathering information about the entities deemed similar, the graphs are an overall hierarchy view per se. That means the user can also navigate a single ontology graph (even

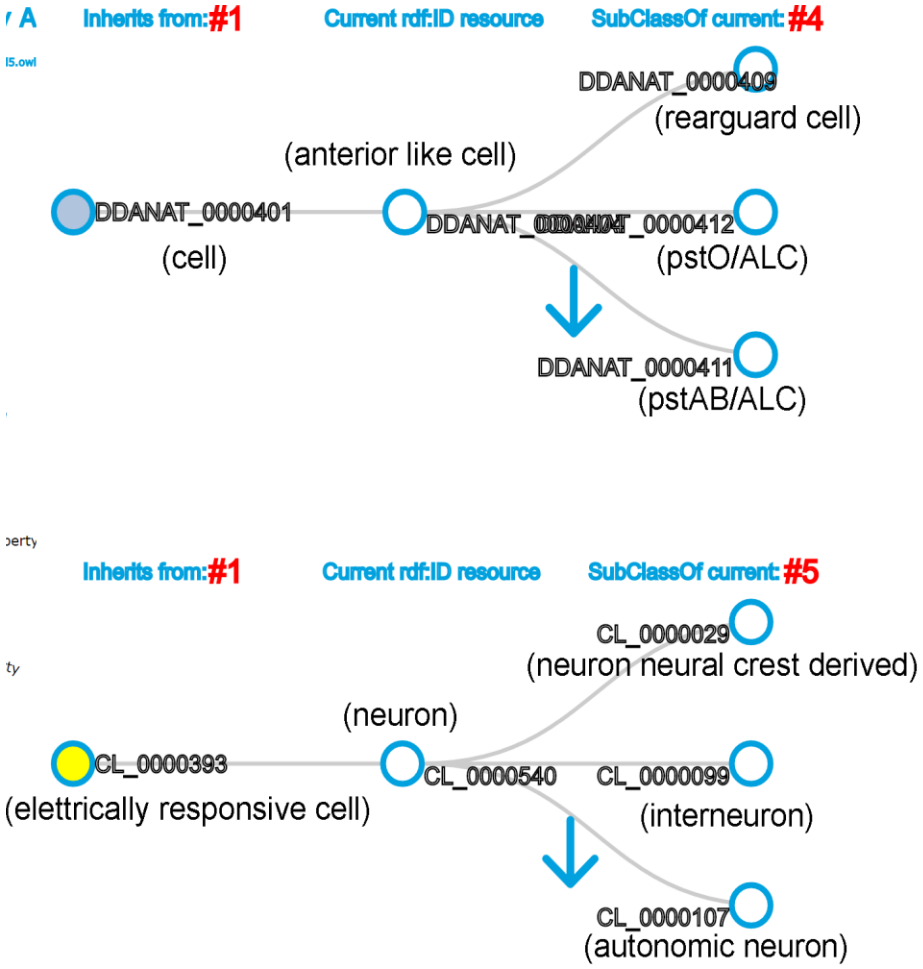


Figure 4 An example of very interconnected hierarchies manageable by the Ontochord GUI

in those nodes that are not present in the alignment), thus evaluating the alignment in light of a wider context.

The hierarchy tool is very flexible, allowing a dynamic popup view of entities with many child or parent nodes (Figure 4). The aim of the hierarchy tool is to help users in deciding whether they should consider entities as similar basing the decision on a different kind of information.

Very interconnected hierarchies are manageable by the Ontochord GUI thanks to the subgraph-like fashion in which it displays the entity nodes. In this example of the figure the number of subclasses exceeds the default limit of 3, but they can be still inspected using navigation arrows.

To assess the functionality of the tool in the field of biomedicine, we hereby show in Figure 5 the graphical interface of ***OntoChord*** with the alignment between the AEO (Anatomical Entity Ontology: http://obofoundry.org/ontology/aeo.html) and the DDANAT (Dictyostelium Discoideum Anatomy: http://obofoundry.org/ontology/ddanat.html).

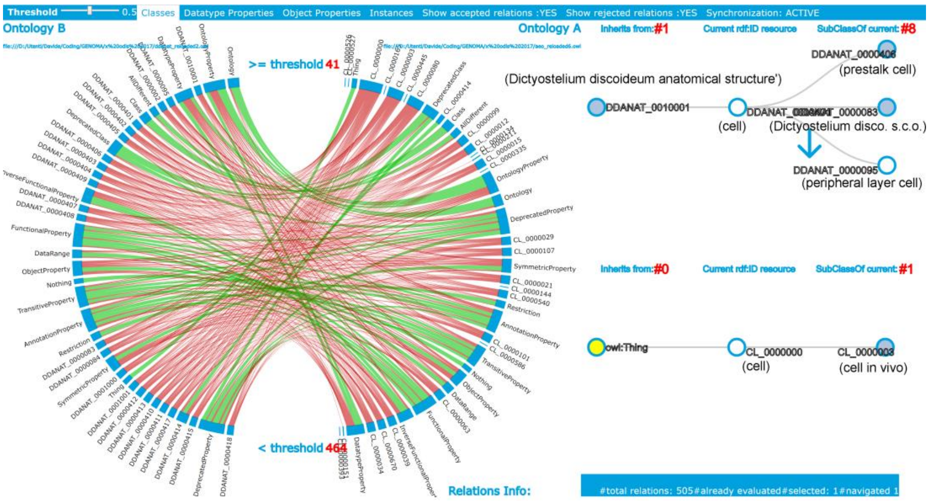


Figure 5 Hierarchies and CutChordGraph of the alignment between AEO and DDANAT with 505 similarity relations.

OntoChord offers and immediate overview on most connected entities, who are characterized by greater thickness and bigger inner arches (attachment points) on the ChordGraph sides.

Once most trivial similarities have been validated by the end-user, the available space can be freed from unnecessary chords, making it easier to manage less connected entities. Even when considering a very dense CutChordGraph, relations can be safely selected thanks to the top panel that shows which chord the cursor is hovering.

At the same time, a Bottom-left popup will show the current state of the relation (Figure 6).

4.1.3.1. Comparison of entities' properties

To add even more useful data to the context, the user can see entity nodes' properties.

Left-clicking on nodes opens a table showing all the properties of that entity in ontology A and B and highlights differences between them (Figure 7).

Indeed, the table is comprised of 3 sections:

- Properties shared by the related entities in ontology A and B whose values are identical.
- Properties shared by the related entities in ontology A and B whose values are different.
- Properties missing in ontology A but present in B, or viceversa.

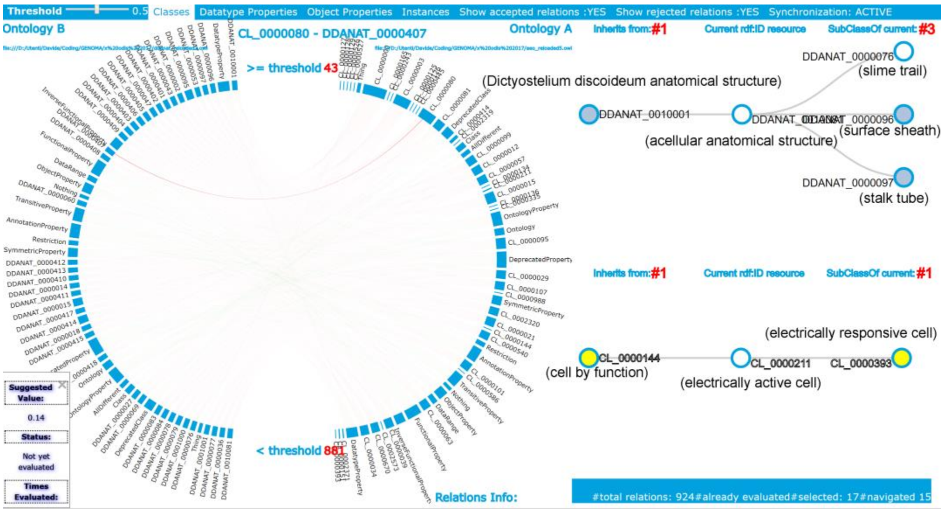
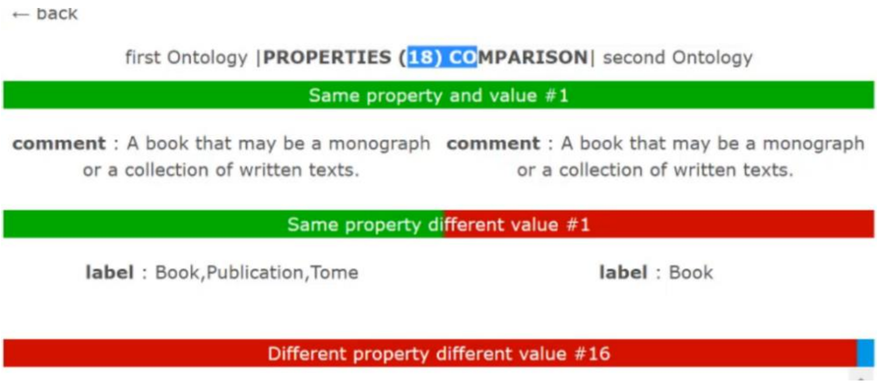


Figure 6 CL_0000144 and CL_0000393 are shown as yellow nodes because they are missing in the alignment.

5. Future development

In order to manage the matching between big ontologies, ***OntoChord*** will be provided with a double threshold filter. Using a double cursor slider bar, the user will be able to set the value range of the relations he wants to show. As an example, the user will be able to choose to see just the relations that have a similarity value included in the range [0.8,1.0] to reduce the numbers of chords and entities showed.

In any case ***OntoChord*** by itself will automatically reduce the number of entities showed if it exceeds a predefined threshold to avoid browser's memory overload.


← back

first Ontology |**PROPERTIES (18) COMPARISON**| second Ontology

Same property and value #1

comment : A book that may be a monograph or a collection of written texts. **comment** : A book that may be a monograph or a collection of written texts.

Same property different value #1

label : Book,Publication,Tome **label** : Book

Different property different value #16

Figure 7 Values comparison between the two selected resources of the two ontologies

6. Conclusions

In this paper, we presented a tool for ontology alignment visualization called ***OntoChord.*** that has been implemented as part of the **GENOMA** framework but can be also used as a stand-alone service. The novelty of the tool is the use of a Cut Chord Graph to show the similarity matrix resulting from the matching operation. This kind of visualization gives to the user a good overview of the similarity relations existing between the ontologies matched and, at the same time, through the interactive features of the tool, provides the user with good details and also the opportunity to edit the graph returning a feedback on each relation.

The thickness of the chord depends on the similarity value of the link. This feature allows the user to give a quick glance at the level of importance of the relation at hand. Furthermore, the user can also set an acceptance threshold that paints the chord green if its similarity value is over the threshold and red otherwise.

The tool is also provided with some features that can return important information regarding the structure of the ontologies matched. The hierarchy graphs give an idea of the context of the entities matched and further information related to the comparison of their properties. This view is a key feature for the visualization of alignments of biomedical ontologies where knowledge is predominantly structured in taxonomies based on *subclass_of* and *part_of* relationships. ***OntoChord*** is able to show also complex hierarchies focusing the attention of the user to the two matched entities' neighbors, where it is most likely to find correspondences.

Refererences

Aumueller, D., Do, H.-H., Massmann, S., & Rahm, E. (2005). Schema and ontology matching with COMA++. *SIGMOD '05 Proceedings of the 2005 ACM SIGMOD international conference on Management of data* (pp. 906-908). Baltimore, MD, USA: ACM.

Enea, R., Pazienza, M. T., & Turbati, A. (2015). GENOMA: GENeric Ontology Matching Architecture. *AI*IA 2015 Advances in Artificial Intelligence* (pp. 303-315). Ferrara, Italy: Springer.

Ham, F. v., & Wijk, J. J. (2004). Interactive Visualization of Small World Graphs. *Proc. of IEEE Symposium on Information Visualization 2004* (pp. 199-206). CSPRess.

Komarek, A., Pavlik, J., & Sobeslav, V. (2015). Network Visualization Survey. *7th International Conference, ICCCI 2015* (pp. 275-284). Madrid: Springer.

Krzywinski, M., Schein, J., Birol, I., Connors, J., Gascoyne, R., Horsman, D., . . . Marra, M. A. (2009). Circos: an Information Aesthetic for Comparative Genomics. *Genome Research*, 1639-1645.

Lanzenberger, M., & Sampson, J. (2006). AlViz - A Tool for Visual Ontology Alignment. *Information Visualization, 2006. IV 2006. Tenth International Conference on.* London, United Kingdom.

Li, Y., Stroe, C., & Cruz, I. F. (2015). Interactive Visualization of Large Ontology Matching Results. *14th International Semantic Web Conference (ISWC).*

Noy, N. F., & Musen, M. A. (2003). The PROMPT suite: interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, 983–1024.