

# OntoPlot: A Novel Visualisation for Non-hierarchical Associations in Large Ontologies

Ying Yang, Michael Wybrow, Yuan-Fang Li, Tobias Czauderna, Yongqun He

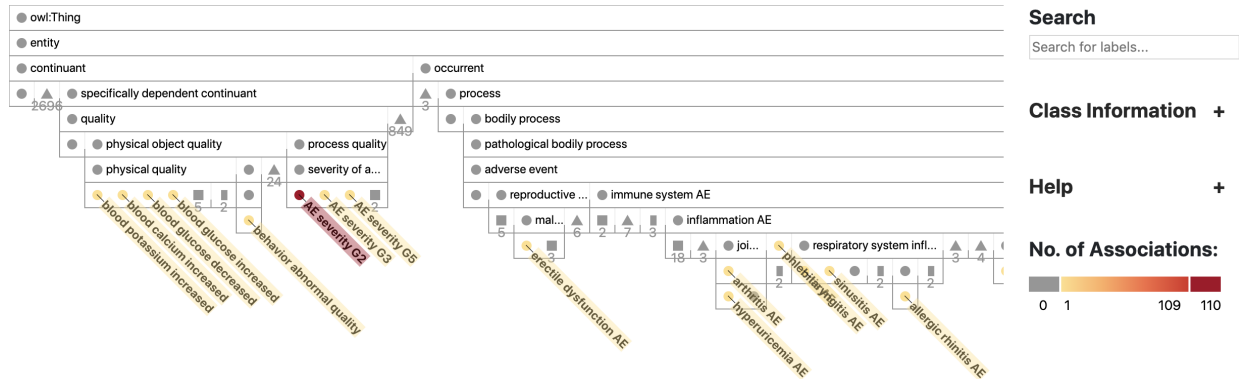


Fig. 1: OntoPlot interface. Classes with associations are highlighted in colour based on the key on the right-hand side. A panel on the left (not shown) selects different types of non-hierarchy associations found within the ontology (denoted by the OWL object properties involved). The panel on the right provides search and displays additional information for selected classes (not shown).

**Abstract**—Ontologies are formal representations of concepts and complex relationships among them. They have been widely used to capture comprehensive domain knowledge in areas such as biology and medicine, where large and complex ontologies can contain hundreds of thousands of concepts. Especially due to the large size of ontologies, visualisation is useful for authoring, exploring and understanding their underlying data. Existing ontology visualisation tools generally focus on the hierarchical structure, giving much less emphasis to non-hierarchical associations. In this paper we present OntoPlot, a novel visualisation specifically designed to facilitate the exploration of all concept associations whilst still showing an ontology's large hierarchical structure. This hybrid visualisation combines icicle plots, visual compression techniques and interactivity, improving space-efficiency and reducing visual structural complexity. We conducted a user study with domain experts to evaluate the usability of OntoPlot, comparing it with the de facto ontology editor Protégé. The results confirm that OntoPlot attains our design goals for association-related tasks and is strongly favoured by domain experts.

**Index Terms**—Ontology visualisation, visual compression, interactive exploration, ontology associations

## 1 INTRODUCTION

Semantic Web ontologies are widely used in many domains to annotate, retrieve, analyse, and integrate data and knowledge [8]. Especially in the biomedical research community, many large and complex biomedical ontologies have been developed to provide a set of formal, controlled and shared vocabularies for describing classes and relationships between them [23, 59]. They standardise biomedical concepts and structure their relations with the main aim to support data integration and information exchange [2, 45].

Many large ontologies have been developed over the years. BioPortal [53, 56], a comprehensive biomedical ontology repository, currently contains 763 ontologies with a total of almost 10 million classes.<sup>1</sup> These include the influential Gene Ontology [23], which has close to 50,000 classes, the SNOMED CT ontology [63], which currently has more than 340,000 classes, among others. Visual tool support for the

effective interrogation of these ontologies is essential to both ontology users and ontology developers. As a response to this important problem, many visualisation systems have been developed in recent decades [17, 38, 55].

Visualisation maps information to a graphical representation to convey knowledge. Effective visualisation makes it possible to obtain insights into data, support user tasks, and perform exploration on large and complex data structures. It typically reduces cognitive effort by limiting the amount of information presented to users.

Ontologies, especially those expressed in the OWL [32] and OWL 2 languages [13], usually have a hierarchical, tree-like structure defined by subsumption relationships between concepts (or classes<sup>2</sup>). Given the central importance of subsumption relationships in defining an ontology, many of existing visualisation systems rightly treat hierarchies as first-class citizens. For example, Figure 9 shows the interface of Protégé [52], the de facto ontology editor. The left-hand side of the interface shows the subsumption hierarchy of the ontology in an indented tree layout.

Besides the subsumption relationship between *named* classes, important *associations* between classes are also expressed through subsumption and the use of properties and anonymous class expressions (cf Axiom 2). These associations between concepts in biomedical ontologies define the relational expressions between the concepts in the biomedical domain [61] and they capture a variety of rich information in addition to the subsumption hierarchy.

For example, anatomy ontologies [28, 47, 62] model the parts of organisms and the structural and developmental relationships between

- Ying Yang, Michael Wybrow, Yuan-Fang Li, and Tobias Czauderna are with Monash University, Australia. Email: [ying.yang@monash.edu](mailto:ying.yang@monash.edu), [michael.wybrow@monash.edu](mailto:michael.wybrow@monash.edu), [yuanfang.li@monash.edu](mailto:yuanfang.li@monash.edu), [tobias.czauderna@monash.edu](mailto:tobias.czauderna@monash.edu).
- Yongqun He is with University of Michigan Medical School, USA. E-mail: [yongqunh@med.umich.edu](mailto:yongqunh@med.umich.edu).

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: [reprints@ieee.org](mailto:reprints@ieee.org). Digital Object Identifier: xx.xxx/TVCG.201x.xxxxxx

<sup>1</sup><https://bioportal.bioontology.org/>

<sup>2</sup>In the rest of this paper we use the terms concept and class interchangeably.

these parts. The rich set of spatial associations in anatomy ontologies can be used to define overlapping, continuous, or adjacent regions. Thus, it is important to see which parts of an organism are spatially associated with another part of that organism. Also, the regulatory associations in the Gene Ontology indicate where one process or function affects the manifestation of another process, function, or quality [22]. Multiple molecular functions regulate one target, reflecting cooperative translational control, while one molecular function may have multiple targets, indicating target multiplicity [19, 43, 49, 71].

Adverse drug events (ADEs) are undesired medical consequences of drug interventions [29, 72]. According to [69, 72], ADEs result in more than 770,000 injuries and deaths each year and cost up to \$5.6 million per hospital, and can lead to withdrawal of marketed drugs or failure of drug development. Therefore, identifying and predicting ADEs are major focuses in pharmacovigilance. Many ontologies have been developed to capture and analyse ADEs [25, 29, 33, 34, 41, 42, 57, 67]. In the ontological context, ADEs are often investigated on the class level, where a given ADE may be common to all drugs in the corresponding class, or conversely, an ADE may be associated with some class members but not with all of them. The discovery of such associations facilitates the learning and prediction of ADEs. It is also important to see which drug associates the most adverse events and vice versa.

Despite the importance of associations, most of the existing ontology visualisation tools focus on class hierarchy [17, 38, 55]. Although some tools, such as [14, 37, 40, 44, 65], do visualise ontology associations, they do not support the complex association tasks mentioned above, and users sometimes need to count the number of associations between multiple classes to perform such tasks. Additionally, existing ontology visualisation tools are often vague about the use cases and tasks they support [12, 17].

In this paper, we present OntoPlot (available at <https://ialab.it.monash.edu/ontoplot/>), a novel ontology visualisation system specifically designed to support association-related user tasks. It includes a number of novel features:

- It uses a hybrid visualisation combining icicle plots [39] with visual compression techniques to show an ontology’s inheritance backbone.
- It automatically compresses irrelevant subtrees to effectively emphasise non-hierarchy associations and uses distinct glyphs to help distinguish different structures of compressed subtrees.
- It allows users to interactively expand and collapse subtrees of the hierarchy, with additional functions like filtering by a particular property or class, search, and highlighting by colours and labels.

To evaluate OntoPlot’s effectiveness in supporting user tasks, we conducted a prototype evaluation and then a subsequent user study with domain experts. Results show that for association-oriented tasks, OntoPlot outperforms Protégé in terms of accuracy and task completion time.

While ontologies are most widely used in the biomedical domain [15], similar tasks are performed with ontologies in a number of application areas. For example, in the agronomy domain, many ontologies have been produced to represent and analyse agronomic data [16, 36]. They are used to answer questions like “what are the appropriate rice varieties for a given soil or region?” and “how many rice varieties are bred from a particular breeding station?”. In the e-government domain, they are used to analyse the information about citizens, authorities, or investment [21, 66]. For bibliometrics, they are used to investigate the research areas, scientific collaborations, publication impact factor, and granted funding of researchers, their affiliations and regions, leading to potential opportunities [1, 46, 54]. OntoPlot is domain agnostic and can facilitate equivalent tasks for any domain.

## 2 RELATED WORK

Here we give an overview of biomedical ontologies and discuss biomedical ontology associations as examples of use cases to which OntoPlot

can be applied, describe visualisation methods for ontologies, and then explore approaches for visual compression.

### 2.1 Ontology

The term *ontology* comes from philosophy. It refers to the study of things that exist in nature and how to describe and group them. It has been adopted by computer science to represent a formal specification of conceptualisation [3].

An ontology describes knowledge in a domain. It consists of concepts and relationships between these concepts. Typically, a concept is a set of objects, and their relationships are defined as binary relations. The most prevalent relationship is that of *inheritance*, where one class  $C$  is stated as a *subclass* of another class  $C'$ , if every object in  $C$  is included in  $C'$ . The inheritance relationships in an ontology typically form a tree-like hierarchy.

For example, the Cardiovascular Disease Ontology (CVDO) [6] is a medium-sized ontology with approx. 500 concepts. The concept ‘familial atrial fibrillation’ (ID `DOID_0050650`) is a subclass of concept ‘atrial fibrillation (disease)’ (ID `CVDO_0000092`).

Besides the inheritance hierarchy, ontologies also define other types of relationships between classes. These relationships capture richer associations between classes within the same ontology or even across different ontologies. These associations are typically defined over *predicates* (binary relations) and other concepts.

In CVDO, the concept ‘familial atrial fibrillation’ is further constrained such that it must be mapped through the predicate ‘has material basis at all times’ (ID `BFO_0000113`) to another concept, ‘genetic disorder’ (ID `OGMS_0000047`).

Expressed in the OWL DL syntax [32], the above two definitions can be formally expressed as follows. Axiom 1 states the subclass relationship (denoted ‘ $\sqsubseteq$ ’) between ‘familial atrial fibrillation’ and ‘atrial fibrillation (disease)’. Axiom 2 states the association on ‘familial atrial fibrillation’, asserting it as a subclass of a *someValuesFrom* value restriction.

$$\text{DOID\_0050650} \sqsubseteq \text{CVDO\_0000092} \quad (1)$$

$$\text{DOID\_0050650} \sqsubseteq \exists \text{BFO\_0000113.OGMS\_0000047} \quad (2)$$

Ontologies have been widely adopted for the purpose of knowledge representation in a number of areas, especially in biological and medical research. A key motivation of their adoption is that an ontology can provide a basis for integrating and understanding knowledge from multiple sources. In this research, we will only discuss biomedical ontologies and their use. However, we note that OntoPlot is not restricted to only this domain, as described in Section 1.

Biomedical research is one of the popular applications of ontologies [59, 64], where they drive the computational use of biological data. In biomedical research, there is an abundance of heterogeneous data, including genes, proteins, clinical observations, and laboratory data, that need to be integrated to facilitate the formulation, evaluation, and refinement of hypotheses. Biomedical ontologies achieve this by organising and classifying knowledge in a formalised and structured manner, providing unambiguous and shareable descriptions.

The shared understanding of collected data is essential for biologists to describe the same entities in the same way. One typical example is the widely-used Gene Ontology [4], which defines a large number of concepts (or classes or terms) to annotate biological entities (i. e., genes and gene products) that result from high-throughput experiments.

One example ontology with rich non-hierarchical associations is the Ontology of Drug Neuropathy Adverse Events (ODNAE) [25]. ODNAE is developed to support the study of drug-associated neuropathy adverse events. It extracts classes from different ontologies and integrates them to generate an ontology-based semantic framework that brings all related knowledge together in a logical and structured format for interdisciplinary representation and analysis. Extending the Ontology of Adverse Events (OAE) [29], ODNAE imports related drugs from the Drug Ontology (DrON) [26] with their chemical components defined in the Chemical Entities of Biological Interest (ChEBI) ontology [27], drug mechanisms of action from NDF-RT [10], and biological

process in the Gene Ontology (GO) [4], and links these classes with semantic relations. Totally, ODNAE contains 1,579 classes.

While performing analysis of drug-associated neuropathy adverse events, queries can be used on ODNAE to answer specific questions, such as: “how many associations between drugs and their corresponding neuropathy adverse events are at different levels in the hierarchy?”, “how many neuropathy-inducing drug chemicals are classified at different levels of ChEBI?”, and “how many adverse events are related to different groups of drug molecular entities?”. One significant question is about the *class effect*. Given an adverse event and a drug class, a class effect exists for the drug class when all its subclass drugs (drug chemical ingredients or drug products) are associated with the adverse event. In other words, if there is a class effect [69], it means that the effect is exhibited by every subclass of the class. Non-hierarchical associations are essential in answering these queries, and we believe an intuitive, task-supportive visualisation can assist people to better understand such complex ontologies.

## 2.2 Ontology Visualisation

With the increased adoption of ontologies in diverse fields, there is a growing need for effective ontology visualisations to support development, management, and utilisation of ontologies.

Compared to visualising strict hierarchies, ontologies are more challenging. Firstly, ontologies often contain multiple inheritance. This is typically solved by duplicating a concept under each of its parents or by using multiple edges to link a concept to all of its parents, either of which has its own drawbacks. With duplication there is the problem of redundancy, whereas with multiple edges there is the problem of visual occlusion.

Secondly, an ontology can contain a rich set of non-hierarchical relationships, which are typically defined using object properties, datatype properties, or annotation properties. Hierarchies are also used to represent other types of information, including concept equivalence and disjointness [5]. However, most ontology visualisations still target the hierarchical structure of ontologies. Some of them visualise all the relations, while some visualise exclusively the hierarchy. Moreover, each concept may have instances, ranging from one or two to thousands [38]. Depending on the task, sometimes instances are required to be visualised.

The most widely used ontology visualisation method is that of indented trees, which is employed by the de facto ontology editor Protégé [52]. It primarily visualises the inheritance hierarchy of an ontology and duplicates concepts for multiple inheritance. Non-hierarchical associations are listed textually in a separate pane.

Network diagrams are another popular method to visualise ontologies. OWLViz [31], a plugin for Protégé, uses a layered node-link (network) diagram to visualise the inheritance relationships, providing an alternative view for the hierarchy, but does not display any non-hierarchical associations. WebVOWL [44] was developed as a visual notation for OWL. It models concept interrelations in ontologies but does not visually differentiate hierarchical relationships and non-hierarchical associations.

Several well-known tools such as Jambalaya [65], Knoocks [37], OntoViewer [14], and a multiple view visualisation tool developed by Kuhar and Podgorelec [40] use node-link or space-filling strategies to represent the ontology inheritance hierarchy structure, and visualise the non-hierarchical associations as links between the classes in the hierarchy. None of these tools appear to be actively maintained.

A comprehensive survey on ontology visualisation [38] categorises systems for visualising ontologies based on their visualisation types: indented list, node-link and tree, zoomable, space-filling, focus + context or distortion, and 3D information landscapes. A recent survey [55] proposes two categories: graph-based methods and multi-method visualisation techniques. The latest survey [17] provides a useful classification and comprehensive evaluation of available ontology visualisation tools. The results show that most visualisation systems focus on class hierarchies, and that their maturity, usability, and scalability are still limited. Interested readers are referred to these surveys for a comprehensive overview of ontology visualisation methods.

## 2.3 Visual Compression of Large Ontologies

The visualisation of large ontologies, or large hierarchies in general, is challenging. The difficulty for ontology developers during the creation of an ontology is how to use available screen space for presentation most effectively. For ontology users the challenge is how to explore and interact with a large ontology most efficiently.

The visualisation of large ontologies, or large hierarchies, can be done using explicit methods (explicit representation of parent-child relations, e.g., by edges) or implicit methods (implicit representation of parent-child relations, e.g., by positional encoding). The latter of the two approaches allows for four axes in the design space: 1) dimensionality (2D or 3D), 2) node representation (graphics, primitives, glyphs), 3) edge representation (inclusion, overlap, adjacency), and 4) layout (subdivision, packing) [58].

In particular, the utilisation of glyphs for the representation of groups of nodes for the visualisation of graphs and hierarchies has drawn some attention in recent years since it allows for a more compact representation by compressing or simplifying the topology. For graphs, Dunne and Shneiderman introduced motif simplification for node-link diagrams which replaces common patterns of nodes and links with compact and meaningful glyphs [18]. A similar approach to motif simplification has been proposed by Shi et al. [60]. Their structural equivalence grouping considers nodes with similar connectivity behaviour and patterns (but not necessarily close proximity) as a group. However, this approach seems to be limited to a particular topology. Yoghoudjian et al. proposed graph thumbnails for identification and comparison of large graphs [70], but these visual summaries hide a lot of connectivity information necessary to understand associations. More comprehensive discussions of glyph-based visualisation strategies, guidelines and techniques in general can be found in [68] and [9].

For hierarchies, the Cheops method uses triangles to visually compress hierarchical datasets based on context and user interaction but is limited to horizontal compression [7]. Jiao et al. [35] apply the compression technique to leaf nodes if the number of leaves for a parent node is above a certain threshold, and use a single large node to represent those leaves to save space. Heer and Card [30] have explored Degree-of-Interest (DOI) trees where some uninteresting branches are collapsed so the tree can be arranged within a constrained area. They allow users to interactively explore by collapsing and expanding, and they progressively recompute DOI values. More recently, Nobre et al. [51] used similar hierarchical DOI compression for family trees. Their visualisation displays a hierarchy arranged horizontally with a separate row dedicated to each person of interest, each displaying multiple attributes that can be easily compared between nodes. They summarise uninteresting subtrees and siblings by showing them as small icons on the rows of interest.

Approaches for the visual compression of large hierarchies have, to the best of our knowledge, not been applied to the visualisation of large ontologies and their non-hierarchical associations. However, it seems that these approaches are very suitable to create more compact visualisations and to make it easier to interactively explore such visualisations. While glyph encoding techniques can convey abstract structural information and hide complexity [48], they could be improved to give a visual summary of hidden structure.

## 3 ONTOPLOT DESIGN

In this section we list the use cases for biomedical ontologies that were our original motivation, identify design requirements arising from these, and then describe the OntoPlot visualisation in detail.

### 3.1 Motivation

In Section 2.1 we outlined the process of using biomedical ontologies for exploring and cataloguing the adverse effects from drug use. This type of work involves understanding not only the underlying hierarchy, but also types and strengths of associations between classes in different parts of the ontology. During early interviews with our co-author, Yongqun He, an expert in bioinformatics, we identified a number of common use cases that are important for such work (see also Table 1):

Table 1: Common use cases when working with association data in biomedical ontologies.

Label	Description	Need
U1	Discover new knowledge	Access the entire ontology and its contained information.
U2	Generalise concepts	See the path from a class to the root.
U3	Discover common knowledge	Find the lowest level of common ancestors for associations.
U4	Explore a class' associations	See the distribution as well as details of associations.
U5	Detect significant associations	Compare relative association strength of classes.
U6	Identify class effect	See when associations apply to a number of child classes.
U7	Predict possible associations	Show the sibling of classes with associations.

- U1: In order to discover new knowledge, users need to be able to access the entire ontology and its contained information.
- U2: In order to generalise concepts, users must be able to clearly trace the path from a given concept to the root concept of the hierarchy.
- U3: In order to discover common knowledge, given two classes of interest the user must be easily able to trace their paths towards the root and determine the concept that is the lowest common ancestor.
- U4: For a class of interest, users must be able to see the distribution of associations across the ontology hierarchy. They must also be able to see the number of associations and association details.
- U5: In order to detect significant associations, users must be able to easily identify the associations with the greatest strength in the ontology, and the classes to which they apply.
- U6: In order to identify class effects (as described in Section 2.1), users need to be able to clearly identify when particular types of associations apply to most or all child classes of a given class, i.e., the association effects on a class of things.
- U7: In order to predict possible associations, users need to be able to explore the siblings of classes with a given association.

## 3.2 Design Requirements

From the use cases described above, along with the typical nature of ontological data, we can identify a number of design requirements for an interactive system to explore associations.

As described in U1, the entire ontology should be embodied in the visualisation. Since ontologies can be very large, the visualisation is required to maximise the use of available space (R1).

Also, ontologies are generally broad (i.e., much wider than they are deep), with traditional tree visualisations, the branches with large numbers of leaf nodes will take up significant amounts of horizontal space. There is a need to give less prominence to branches with large numbers of leaf nodes. Moreover, to support the access of information contained in large ontologies, users should be able to explore the ontology and easily find desired information (R2).

Ontologies encode a clear hierarchical structure through sub-class-of relationships. Even though we are primarily interested in non-hierarchical associations, the hierarchy is still the most useful way to arrange ontologies, so this must be prominently represented (R3). In addition, the hierarchical structure is essential for use cases U2 and U3.

When considering associations in large ontologies, those associations might only apply to a small subset of the ontology. An effective

visualisation needs to clearly highlight the parts of the ontology with relevant associations (R4) and emphasise those with the greatest strength (U5).

Furthermore, where there are large parts of the ontology without relevant associations, the visualisation should be able to hide or show these (R5) so that the user can consider just the relevant parts of the ontology (U4, U6, U7), or optionally view additional information (R6) as desired (U4).

## 3.3 Visual Design

OntoPlot is similar in style to an icicle plot [39]. The basic visual style of OntoPlot can be seen in Figure 1. It primarily emphasises the tree structure of the hierarchy using boxes (R3), where the children of a given item are displayed directly below the parent item, and the parent box's width is the total width of all of its children. However, the use of a standard icicle plot to visualise an ontology hierarchy with many leaf nodes would not be ideal since the overall width of the visualisation would be proportional to the number of leaf nodes. For this reason, we take the basic layout of an icicle plot but represent nodes in the hierarchy as circle glyphs within the boxes traditionally used in icicle plots. Where a number of a given node's children are leaf nodes, we consolidate (wrap) these together in a single box that is taller and wider in order to accommodate multiple circle glyphs (see Figure 3a and 3e). Thus, we reduce the overall width of the visualisation at the cost of a moderate increase in height (R1). A similar approach of wrapping leaves was identified in [24] which arranges leaf nodes in grids under the enclosure of their parent node. Even with this wrapping of leaf nodes, a large ontology may still be very wide. For this reason, OntoPlot lets the user easily scroll the visualisation horizontally.

OntoPlot does not display labels for all classes by default since these take up a large amount of space and can cause problems with occlusion when densely packed. Boxes for displayed subtrees are often quite wide so labels for these nodes are shown greedily where space exists (see Figure 2a). To differentiate between neighbouring boxes that contain siblings of the same parent class versus neighbouring boxes from different subtrees, OntoPlot uses either a partial and faint line in the first case and a solid line in the second case (see Figure 2b).

When OntoPlot is loaded, the user is presented a list of non-hierarchy association types (predicates as described in Section 2.1) found within the ontology. They select the association type they are interested in (with the option to change this at any time). OntoPlot visualises these associations by labelling and colouring the circle glyphs of classes to which that type of associations apply. These classes are frequently, but not always, leaf nodes in the hierarchy. A range of colours are used, where intensity is used to indicate the number of associations applying to that class. The colour key is dynamic depending on the maximum number of those associations applying to any one class. Nodes with the minimum and maximum number of associations are clearly coloured (R4) and further colours are used to categorise values interpolated between these (e.g., see bottom-right in Figure 1). This colouring is also applied to the labels for these classes to make the associations stand out. Labels for classes with associations are positioned diagonally to allow labelling of neighbouring classes without occlusion.

When the user selects an association type, much of the hierarchy will be *uninteresting* in the sense it doesn't contain any classes with these associations. OntoPlot detects these and uses a form of visual compression to collapse uninteresting subtrees within the ontology (see Figure 2c). This is described in detail in the following section.

## 3.4 Visual Compression

While considering a particular type of associations and the classes to which they apply, there will often be a large subset of the ontology which is *uninteresting* in terms of those associations. For this reason, we designed a form of visual compression that allows us to compress the uninteresting subtrees in order to give more prominence to the *interesting* parts of the hierarchy (R5).

We identified three cases worthy of compression and use distinct glyphs to represent their different structure (see Figure 3):

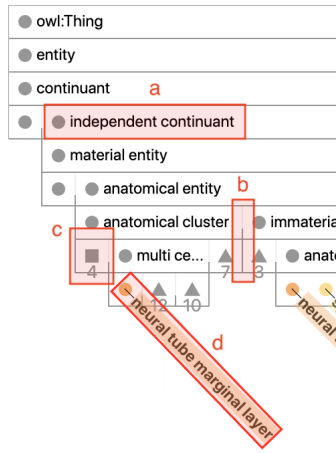


Fig. 2: OntoPlot visual design. (a) Parent classes are labelled where possible. (b) Solid lines are shown between different subtree boxes (bottom) and a partial, faint line is shown between sibling boxes (top). (c) Uninteresting subtrees are compressed. (d) Classes with associations are labelled and coloured based on the number of associations.

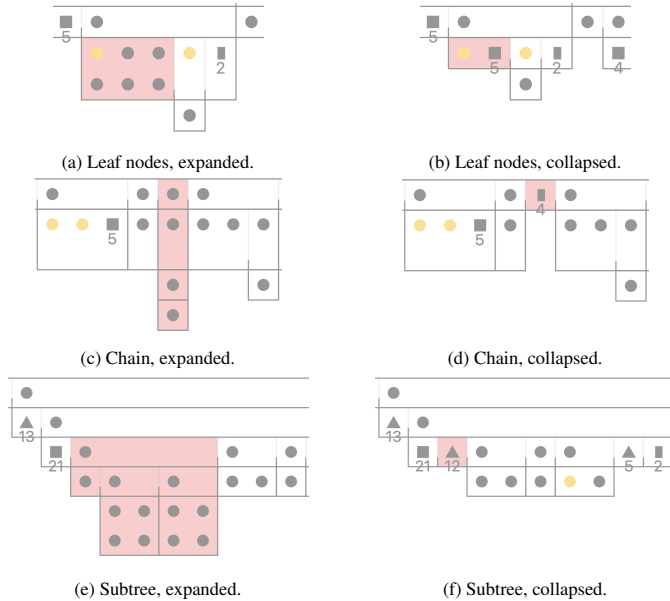


Fig. 3: Examples of OntoPlot visual compression.

- **Leaf nodes:** Where an interesting node has multiple uninteresting leaf nodes as children, these nodes will already be shown as a number of circle glyphs in a single box. We replace these multiple dots with a single square glyph, labelled with the number of hidden nodes.
- **Chain:** Where an interesting node has a descendent subtree that is a chain of only uninteresting nodes, we replace this chain with a single box containing a thin block glyph, labelled with the number of nodes in the chain.
- **Subtree:** Where an interesting node has a descendent subtree that contains only uninteresting nodes and doesn't fall into the previous two categories, we replace the subtree with a single box containing a triangle glyph, labelled with the number of nodes in the subtree.

Whenever we show a particular set of associations, each class in the hierarchy can be considered *interesting* or *uninteresting* depending if

it has any associations. Using this, we can walk the hierarchy to get the set of boxes that can be compressed as described above. We do this by performing a recursive depth-first traversal of the tree from the root box, where for each box the recursive call returns the number of interesting nodes in the subtree and an array of any collapsible boxes. If the active box is *interesting*, it adds to the collapsible box array any of the children that contain no interesting nodes. Since we can discover these collapsible nodes in a single depth-first traversal, this process is linear in the number of classes.

### 3.5 Interaction

OntoPlot displays labels for parent nodes greedily where possible, and labels for classes with associations as described earlier. The user can click and drag these associated class labels if they happen to be obscured (shown in Figure 4a). For all other classes, OntoPlot displays class labels and other information in a pop-up window while the user hovers over the glyph corresponding to the class (R2), as shown in Figure 5.

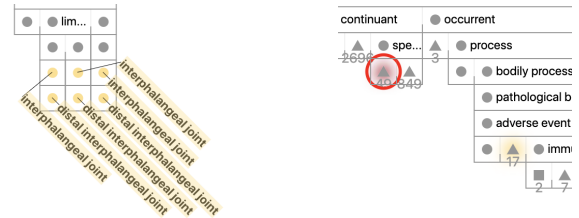


Fig. 4: Interactive features of OntoPlot.

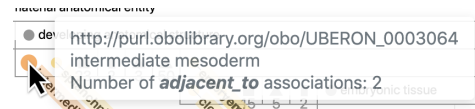


Fig. 5: OntoPlot shows labels while hovering over classes.

While OntoPlot performs automatic visual compression to hide the uninteresting parts of the ontology hierarchy, the user can always interactively expand and collapse subtrees in order to show or hide sections of the ontology. To expand a subtree, the user can double-click on the glyph of a compressed section of the tree (square, thin block or triangle). The user can also compress a particular subtree by double-clicking on the glyph corresponding to the root of that subtree (R2). Subtrees can be collapsed regardless of whether the classes they contain are interesting or uninteresting. If a subtree contains interesting classes, the glyph is displayed with a coloured shadow to which indicates the maximum number of associations in the collapsed subtree (shown in Figure 4b).

Interactive collapsing or expanding operations are performed efficiently without recomputing and redrawing the entire visualisation. Instead we compute just the changes (box size and box position translations) that need to be performed to sections of the visualisation as the result of any interaction. To help preserve the user's mental map, OntoPlot highlights the portion of the tree being collapsed or expanded prior to the operation and then highlights the same portion for a moment after the operation has completed. This highlighting is shown in Figure 3.

The user can select or deselect a class by clicking on it. When they do so the visualisation updates to show label and highlight with colour only the classes that the selected class has associations with.

The selected class is given a black outline and subtle arrows denoting the direction of the associations (one pointing in at the top-left if it is the target of the selected association type and another pointing out at the top-right if it is the source of the selected association type). While a class is selected, the right-hand panel of the interface displays additional information on that class, including a textual list of all its associations (R6). A popover window is also displayed below the selected node (see Figure 6). This indicates the number of associations and gives the user a “Pin Label” button to mark the class with its label and a “Focus Mode” button (R5) to compress the hierarchy to show only associations the selected class is involved in (described below). If the user collapses a subtree containing a selected class, a pulsing red circle will be shown around the glyph for the collapsed subtree (shown in Figure 4b).



Fig. 6: When a class is selected in OntoPlot highlights it and shows additional information and controls.

The right-hand panel also offers a search field. When the user enters a search term, that panel displays a scrollable list of matching classes in the ontology. Selecting a class from this list selects the class in the visualisation and scrolls the main OntoPlot view to make that class visible (R2). If the user ever scrolls the visualisation away from the selected node, OntoPlot shows a pulsing arrow at the edge of the view that points to the glyph for the selected class.

### 3.6 Focus Mode

When the user wants to focus on the associations for a particular class. They can click the button shown in the popover below the selected node. This causes OntoPlot to recompute the interesting and uninteresting parts of the ontology and visually compress the uninteresting subtrees to show a view that emphasises just the selected class and other classes directly associated with that class (as shown in Figure 7). The user can interactively explore this view including expanding and collapsing subtrees. While in this mode, a dark bar is shown at the top of the interface as a reminder. A “Reset View” button is available that returns to a view of the hierarchy which shows all classes with the given association type.

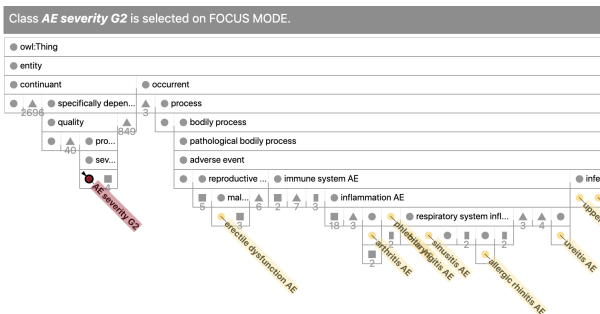


Fig. 7: OntoPlot showing the same ontology as in Figure 1 but after the user has selected *focus mode* for class ‘AE severity G2’ to concentrate on the associations only for this class.

## 4 PROTOTYPE EVALUATION

We conducted a user-based evaluation of an earlier prototype version of OntoPlot. We recruited 20 participants, including 2 domain experts and 18 general users for this controlled experiment. The study design and procedure were similar to the expert user study described below. Participants performed tasks with two different sized ontologies

to gauge performance at two distinct levels of difficulty; CVDO [6] (536 classes) and OCVDAE [67] (4,589 classes). Overall, the results showed that Protégé [52] slightly outperformed OntoPlot for most of the Hierarchy-related tasks on both accuracy and completion time. For Association-related tasks, OntoPlot significantly outperformed Protégé on accuracy, but the completion times using both tools were similar. The detailed results for accuracy, completion time and subjective rating of this study can be found in the supplementary materials. We don’t present details of the prototype study in this paper for several reasons; firstly, the study identified a number of issues with the OntoPlot interface which we have since addressed (again, see details in the supplementary materials), secondly, the study design had some issues (the training was inadequate and not every participant completed each task for all ontologies), and thirdly, most participants were non-expert users (which makes the results less appropriate for evaluating our original aims).

## 5 EXPERT USER EVALUATION

In order to determine if the design of the OntoPlot system meets our original design requirements, we conducted an expert user study with 12 new participants, all domain experts or experienced ontology users.

### 5.1 Study Design

In the user study, we compare OntoPlot with Protégé [52].

As mentioned in Section 2.2, several tools support the display of non-hierarchical associations alongside an ontology’s inheritance hierarchy. When considering these for our preliminary study, we encountered scalability issues with WebVOWL [44] when visualising medium to large ontologies (hundreds or thousands of classes and their associations). Jambalaya [65] and Knoocks [37] are no longer maintained and do not run. Neither OntoViewer [14] nor the multiple view tool described in [40] are publicly available.

We choose Protégé because we want to compare OntoPlot to a robust tool. Protégé is the most widely used and actively maintained tool for ontology creation and editing in the ontology engineering community (based on citations). It provides a baseline representation—an indented list—for ontology hierarchy browsing and visualises non-hierarchical associations as text lists in separate views (see Figure 9).

Also, as mentioned in Section 2.1, the domain experts we consulted (prior to the design of OntoPlot) frequently use Protégé to perform their ontology-based analysis, and present their work using screenshots of the Protégé indented list view with manually added annotations to indicate the association strength in hierarchies [25]. (see Figure 8).

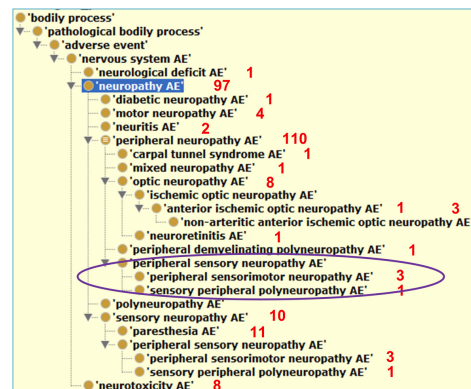


Fig. 8: The manual approach used by domain experts to show associations within an ontology: association numbers written next to branches of the hierarchy on a screenshot of Protégé indented list view (from [25], used with permission).

Protégé is a fully-featured ontology engineering environment, and there are many panes, views and functionality not necessary for our experiment. To avoid confusing our participants with a complex interface, we simplified Protégé by removing the unnecessary items from

the interface, such as “Data properties” and “Individuals” panes, and the “Class Annotations” views. We also deselected some check boxes in the views and search window to avoid irrelevant information being shown to participants. To better support the tasks, we modified the interface layout of Protégé to avoid view switching. We positioned the “Object properties” pane and the “Classes” pane side-by-side, placed the “Class Description” view and the “Class Usage” view next to each other on top of the “Property Usage” view. Figure 9 shows the interface layout configured to clearly show all the views and functions needed in the experiment.

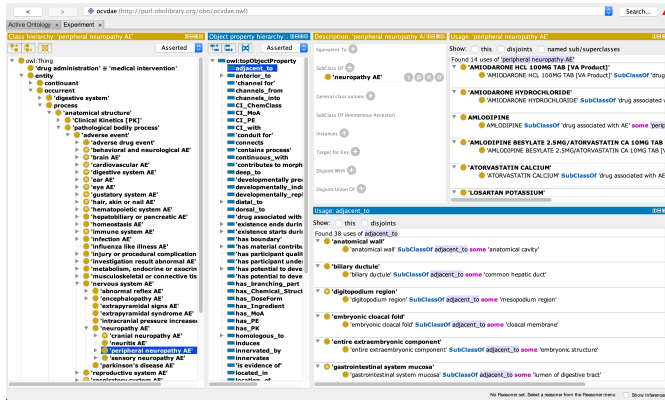


Fig. 9: Protégé interface as configured for use in the study. Left: Class pane, centre: Object property pane, right-half-top-left: Class Description view, right-half-top-right: Class Usage view, right-half-bottom: Property Usage view.

## 5.2 Tasks

As mentioned in Section 3.1, we identified a range of important use cases and user needs for biomedical ontologies from the literature as well as from discussions with domain experts. To test the usability of OntoPlot with respect to the identified user needs, we designed ten tasks from the use cases and organised them into three groups, shown in Table 2.

For the first group of tasks (G1), we focus on the hierarchical structure of ontologies. While these are basic hierarchy comprehension tasks, they are essential to almost all analysis of ontologies. For example, T1, T2, and T3 ask about parent-child relationships, requiring exploration of the ontology hierarchical structure (U1), and investigate whether the visual compression and glyphs in OntoPlot impact the cognition of the ontology hierarchy. Similarly, T4 asks a user to trace the hierarchical path from a class to the root which is related to generalising concepts (U2). T5 examines the intersection of two subtrees supporting common knowledge discovery (U3).

The second group of tasks (G2) focus on non-hierarchical associations. Both T6 and T7 require an exploration of the associations for a class (U4). While T6 asks for all classes associated with a class, T7 asks for the total number of them. T8 requires users to find the class with the highest number of associations in the ontology, which identifies significant classes (U5).

The third group of tasks (G3) further examines the associations together with the hierarchical structure. These tasks are the most complex ones but essential for analysing associations on the class level. T9 asks for the parent having the most children with associations, which helps determining the *class effect* (U6). T10 finds the outlier (class without associations) among a group of sibling classes with associations, providing evidence for predicting undiscovered associations (U7).

### 5.3 Hypotheses

We hypothesised that OntoPlot would perform similarly to Protégé for G1 hierarchy tasks (H1), since both tools clearly emphasise the hierarchical structure of ontologies. We believed that OntoPlot would outperform Protégé on G2 association tasks (H2) and G3 hierarchy

and association combined tasks (H3), since OntoPlot was designed to support ontology association analysis.

## 5.4 Datasets

We use two biomedical ontologies: CVDO [6] and OCVDAE [67]. CVDO (536 classes) was used for the training tasks. OCVDAE (4,589 classes) was used for the study tasks. In total, there are 8 object properties and 551 non-hierarchical associations in CVDO. In OCVDAE, there are 118 object properties and 20,269 non-hierarchical associations. In order to keep the experiment to a reasonable time, we selected classes with less than 25 associations to ask questions about.

The preliminary user study used a small manually constructed (and hence unrealistic) ontology for the training, and participants did the tasks with both CVDO and OCVDAE. That study found little difference in the results between the two ontology sizes, hence the decision to evaluate only the larger ontology in this expert study and use the smaller ontology for the training tasks.

## 5.5 Procedure

We used a within-subjects design for the experiment: 2 tools  $\times$  1 ontology size  $\times$  10 tasks (+ training).

To ensure consistent difficulty of tasks, the same ontology was used for the tasks performed using each tool. To avoid issues of memorisation, class and object property labels were consistently renamed to be different for each tool.

We fixed the order of tasks for each tool but counterbalanced the order of tools shown to different participants.

Participants were required to complete training before performing the experimental tasks. They were firstly shown an introductory document to explain the terminology used in the experiment. Participants also finished a training for each tool before using them. They were shown an introductory document to demonstrate the interface and functions of the tool and were then required to use the tool to answer 10 sample questions with the training ontology. The sample questions covered all experiment tasks in order to allow participants to be familiarised with the tools and the tasks. While answering the sample questions, participants were guided to practise the functions that were needed in the actual tasks for each tool, such as searching, clicking classes or object properties, double-clicking to expand or collapse sub-trees, hovering the mouse cursor over classes to read class labels and association information, and marking classes by pinning labels on them in OntoPlot, and going back or forward in Protégé. After each question, participants were shown the correct answer, and an explanation was given if they didn't answer correctly.

The participants were given access to a study website that guided them through the study, gave them access to training instructions, tasks, and survey questions. When the participants started a task, this was recorded by the investigator. When they completed a task, the participant would signal this to the investigator who would record their answer and completion time. For any task, if participants found it too difficult to complete, they could choose to skip that task.

After completing the tasks for each tool, participants were asked to complete a survey, rating the difficulty level and the confidence level of their answers for each group of tasks. We also collected participants' preferences and comments at the end of the experiment. Answers to survey questions were entered by participants into a Google Form.

After the experiment, participants were asked to answer some questions regarding their background knowledge and experience with ontologies, Protégé, and ontology visualisation tools.

Each experiment session lasted approximately one and a half hours, including training and surveys.

## 5.6 Participants and Apparatus

All 12 participants had experience in the field of ontologies or knowledge graphs. Eleven of them identified as having experience using ontologies, including three with more than three years experience. Ten participants had experiences using Protégé, one of whom had more than three years of experience. Another two participants had used other ontology tools, including the tools developed by the Gene Ontology

Table 2: Tasks in the experiment.

Group	Task	Use case	Description	Example instruction
G1. Hierarchy	T1	U1	Identify the parent of a class.	Please tell me the parent of “skin of body”.
	T2	U1	Identify the child(ren) of a class.	Please tell me the children of “limb segment”.
	T3	U1	Identify the sibling(s) of a class.	Please tell me the siblings of “anatomical space”.
	T4	U2	Identify the path from a class to the root.	Please tell me the path from “process” to the root.
	T5	U3	Identify the closest common ancestor of two classes.	Please tell me the closest common ancestor of “anatomical collection” and “anatomical surface”.
G2. Association	T6	U4	Identify the classes associated with a class.	Please tell me the classes which have the “may_prevent” association with the “Pain” class.
	T7	U4	Identify the number of associations of a class.	Please tell me the number of “may_prevent” associations of the “Hypertrophy” class.
	T8	U5	Identify the class having the highest number of associations.	Please tell me the class which has the most “may_treat” associations.
G3. Hierarchy + Association	T9	U6	Identify the parent class with the most children who are associated with a class.	Please tell me the class which has the most children that have the “adjacent.to” association with the “full formed stage” class.
	T10	U7	Identify a class that is not associated with a specified class, but all of its sibling(s) are associated with that class.	Please tell me the class whose siblings all have the “site_of.metabolism” association with the “Channelopathy” class, but that class itself does not have such an association.

Consortium and a proprietary tool used for a knowledge graph construction engine. Of the 12 participants, three were female and nine were male. Their age ranged from 18 to 41. All participants had normal or corrected-to-normal vision, and none suffered colour blindness.

The six participants recruited from the authors’ university used a 2.3 GHz Intel Core i5 laptop with 8GB of RAM, using a 24-inch monitor with a resolution of 3840x2160 pixels. The six participants recruited from other institutions did the experiment remotely, using their own computers at a resolution of 1600x900 pixels. For the remote participants the experiments were observed via video call.

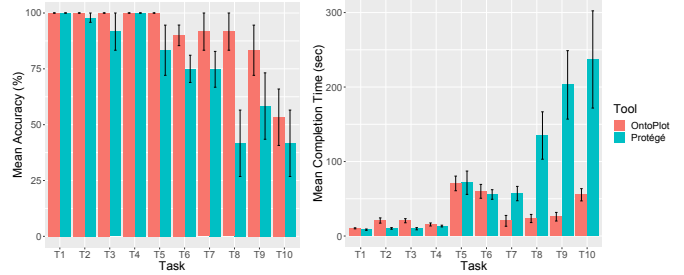
## 5.7 Results

All 12 participants completed the study. Unlike in the preliminary study (prototype evaluation described in Section 4), all 12 participants completed all tasks on the same ontology, OCVDAE. We measured accuracy and completion time for each task, and collected difficulty level, confidence level, preference ranking, and learning effort as rated by the participants. As the data is not normally distributed, we used the non-parametric *Wilcoxon test* to compare accuracy between the two tools [20]. For the completion time data, we only considered the time for answers with an accuracy greater than 0%. Therefore, we used the non-parametric *Whitney-Mann test* for unequal samples [50]. For the rated results, we also used *Wilcoxon test* to analyse significance.

The main evaluation results are summarised in Table 4, including statistical significance for each item. Below we discuss them in detail.

**Accuracy.** Figure 10a shows the details of mean accuracy for each tool per task. We found overall, participants achieved higher accuracy on most tasks with OntoPlot than with Protégé. The two exceptions are for T1 (finding parent) and T4 (finding path), which have equal accuracy (100%) for both tools. The *Wilcoxon test* revealed that for T8 (finding class with most associations), OntoPlot significantly outperformed Protégé ( $p < 0.05$ ).

**Completion Time.** Results for completion time are shown in Figure 10b. We found for most of the Hierarchy tasks (G1), participants spent less time on Protégé than on OntoPlot. Especially for T2 (finding children) and T3 (finding siblings), The *Whitney-Mann test* revealed that Protégé significantly outperformed OntoPlot ( $p < 0.01$ ). For T5 (finding common ancestor), OntoPlot and Protégé had very close completion time, with OntoPlot being slightly faster. Of the Association tasks (G2), for task T6 (finding individual associations) OntoPlot had a slightly longer completion time than Protégé. The results show that tasks for finding and counting most associations (T7, T8), OntoPlot significantly outperformed Protégé ( $p < 0.001$ ). Highly significant differences were also found for the combined Hierarchy + Association tasks (G3) (T9, T10), with OntoPlot substantially outperforming Protégé. Taking accuracy into account, these results indicate that, espe-



(a) Mean accuracy for each tool per task. (b) Mean completion time for each tool per task.

Fig. 10: Participants’ performance of the two tools in the expert user study.

cially for complex tasks (G3), OntoPlot requires substantially less time and achieves much higher accuracy than Protégé.

**Participant Rating.** We use a five-point Likert scale ranging from 1–5 to measure participants’ rating of difficulty (lower is better) and confidence (higher is better) for each group of tasks and each tool. Figure 11a and Figure 11c show the percentage of participants’ rating results, and Table 3 summarises the results.

Table 3: Summary of average of difficulty and confidence ratings as shown in Figures 11a and 11c.

Group	Difficulty		Confidence	
	OntoPlot	Protégé	OntoPlot	Protégé
G1	1.583	1.5	4	3.917
G2	2.083	3	3.75	3.167
G3	2.417	3.5	3.667	2.5

Overall, participants rated G1 tasks performed in Protégé as slightly less difficult than in OntoPlot. For G2 and G3 tasks, participants rated OntoPlot as less difficult than Protégé. Three participants rated Protégé difficulty at 5 (highest) for G3 tasks.

When asked about confidence rating, participants felt slightly more confident with OntoPlot than with Protégé for G1 tasks and gave much higher confidence rating to OntoPlot for G2 and G3 tasks.

Figure 11b shows the result of the preference rating. For G1 tasks, seven participants preferred Protégé over OntoPlot, whereas the situation is entirely reversed for G2 and G3 tasks. All the participants

preferred OntoPlot for these tasks.

The result of the learning effort rating is shown in Figure 11d, also using a five-point Likert scale ranging from 1 (easiest) to 5 (hardest). One participant rated learning effort 1 for OntoPlot, while one participant rated it 5 for Protégé. The average rating is 2.625 for OntoPlot and 3.25 for Protégé. There is no significant difference between the tools ( $p = 0.056$ ).

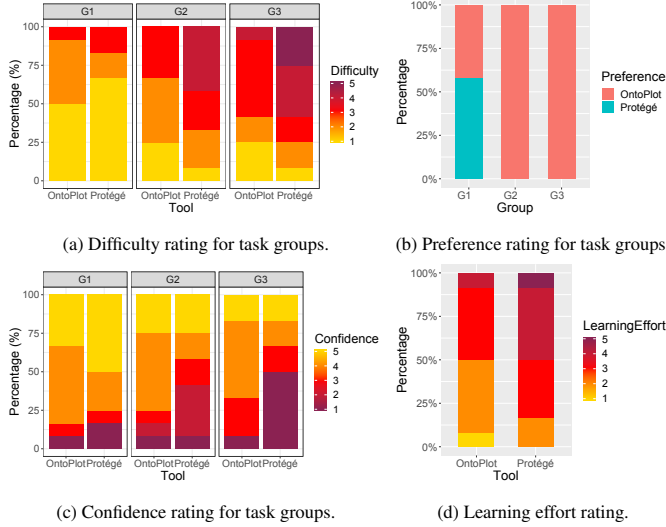


Fig. 11: Participants' rating of the two tools in the expert user study.

**Participant Feedback.** At the end of the experiment each participant was given the chance to provide feedback and give comments. Some participants felt Protégé was more familiar and acceptable, e.g., commenting “The vertical aligned indented list is easier to perceive hierarchy structures”. Most of the participants gave positive feedback for OntoPlot, e.g., commenting “OntoPlot interface is more friendly”, “OntoPlot needs effort to learn, but makes tasks easier”, or “OntoPlot has more compact view of the ontology”. Some participants also provided more specific feedback such as “Lighter lines and darker lines are helpful for distinguishing siblings and non-siblings”, “The labels make finding associations much easier”, “Association labels are easy to read”, or “Tagging feature is nice”. One participant also commented on Protégé: “That is very difficult to find common ancestors with Protégé”.

A few participants also provided helpful feedback for further improvements of OntoPlot, e.g., “Probably can use colour coding for the sibling lines to make them more obvious”, “The subtle arrows could be more effective if can indicate the number of pointing in and pointing out associations”, or “Probably can filter association classes further when there are many associations”.

**Summary.** Table 4 presents a summary of all the results. Overall, OntoPlot moderately outperformed Protégé on accuracy for most tasks, and significantly (i.e., statistically significantly) outperformed Protégé for the task T8. On completion time, OntoPlot was outperformed by Protégé for most G1 tasks (significantly for two tasks), while OntoPlot significantly outperformed Protégé for most G2 and G3 tasks. No significant difference was revealed by the statistical test for the participants' rating data. These results are consistent with those from the first user study, while in the expert study the users had noticeably better accuracy rates using both tools and they performed significantly faster using OntoPlot than Protégé on the G2 and G3 (association) tasks.

## 5.8 Discussion

The expert user study shows that OntoPlot slightly outperformed Protégé for Hierarchy tasks (G1) on accuracy, which accepted our hypothesis H1 (Section 5.3). A common error made by several participants in Protégé was to mistake the sibling shown above a class (at the same indentation level) as the parent of that class, often when there was a some distance between them in the indented list. On completion

Table 4: Summary of the expert user study results (Acc: Accuracy, Diff: Difficulty, Conf: Confidence, Pref: Preference, L.Effort: Learning Effort). The tool mentioned in the columns outperforms the other in terms of the given metric (O: OntoPlot, P: Protégé).

Group	Task	Acc	Time	Diff	Conf	Pref	L.Effort
G1	T1	-	P				
	T2	O	P**				
	T3	O	P**	P	O	P	
	T4	-	P				
	T5	O	O				O
G2	T6	O	P				
	T7	O	O***	O	O	O	
	T8	O*	O***				
G3	T9	O	O***	O	O	O	
	T10	O	O**				
Significance: *** $p < 0.001$ ** $p < 0.01$ * $p < 0.05$							

time, OntoPlot was significantly outperformed by Protégé for the tasks of finding children and siblings. This can be explained by the fact that most participants were Protégé users and were familiar with the indented list for showing hierarchy structure. Also, in order to test the participants' perception of glyph compression, this group of tasks was designed to force participants to collapse or expand the subtrees. The participants spent some time on understanding which glyph or class they should collapse or expand in OntoPlot, and double-checked their answers. In Protégé most of the participants can skillfully interact with the indented list. However, for the finding common ancestor task, the participants spent a little less time in OntoPlot than in Protégé as they can mark the classes by labels, and this made the task easier.

For association-related tasks (G2 and G3), OntoPlot outperformed Protégé on most of the tasks as expected (accepting H2, H3). Especially, for the completion time, there are some significant differences between the tools. We observed that the main reason why participants spent more time in Protégé was because in Protégé a user cannot select both classes and associations at the same time. Thus, the participants had to distinguish different classes or associations by themselves. We also observed that the reason why OntoPlot took marginally more time for task T6 (finding individual association classes) was that some participants spent some time on scrolling the visualisation or dragging the association labels.

## 6 CONCLUSION AND FUTURE WORK

Expressive ontologies contain rich information captured by complex associations involving classes, properties and individuals. However, most existing ontology visualisation systems focus on class hierarchies, making it hard to find information about these associations. In this paper, we presented OntoPlot, a novel visualisation system specifically designed to support the interrogation of *non-hierarchical* associations while still showing the class hierarchy of an ontology. OntoPlot improves space efficiency by a combination of hybrid icicle plots, visual compression techniques and interactivity. We compared OntoPlot with Protégé, the de facto ontology editor, and found that OntoPlot significantly outperformed Protégé on efficiency for the complex association-based tasks and was strongly favoured by the domain experts. While we have evaluated OntoPlot on ontologies, it can be applied to other hierarchically structured data, e.g., research collaborations between organisations, where the number of relationships between individuals in the hierarchy shows how often the researchers worked together.

We plan to add a mini map with linking and brushing to OntoPlot, for easier navigation and giving an overview of the entire ontology. Furthermore, we plan to improve the visual glyphs by providing more information about the size of hidden subtrees. As the same association may have different connectivities in different ontologies [11], we will investigate how to represent multiple hierarchical structures with OntoPlot and show the differences between them in a clear manner.

## REFERENCES

- [1] D. Adam. The counting house. *Nature*, 415(6873):726–729, Feb. 2002. doi: 10.1038/415726a
- [2] E. Antezana, M. Kuiper, and V. Mironov. Biological knowledge management: the emerging role of the Semantic Web technologies. *Briefings in Bioinformatics*, 10(4):392–407, 2009.
- [3] G. Antoniou, P. Groth, F. v. v. Harmelen, and R. Hoekstra. *A Semantic Web Primer*. The MIT Press, 3rd ed., 2012.
- [4] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, et al. Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.
- [5] F. Baader and W. Nutt. Basic description logics. In F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, eds., *The Description Logic Handbook: Theory, Implementation, and Applications*, pp. 43–95. Cambridge University Press, 2003.
- [6] A. Barton, A. Rosier, A. Burgun, and J.-F. Ethier. The Cardiovascular Disease Ontology. In *FOIS*, pp. 409–414, 2014.
- [7] L. Beaudoin, M.-A. Parent, and L. C. Vroomen. Cheops: A Compact Explorer for Complex Hierarchies. In *Proceedings of the 7th Conference on Visualization '96, VIS '96*, pp. 87–92, 1996.
- [8] C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - the story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3):1–22, 2009.
- [9] R. Borgo, J. Kehler, D. H. S. Chung, E. Maguire, R. S. Laramée, H. Hauser, M. Ward, and M. Chen. Glyph-based Visualization: Foundations, Design Guidelines, Techniques and Applications. In *Eurographics 2013 - State of the Art Reports*, pp. 39–63. Eurographics Association, 2013.
- [10] S. H. Brown, P. L. Elkin, S. T. Rosenbloom, C. S. Husser, B. A. Bauer, M. J. Lincoln, J. S. Carter, M. Erlbaum, and M. S. Tuttle. VA National Drug File Reference Terminology: a cross-institutional content coverage study. *Medinfo*, 11(Pt 1):477–481, 2004.
- [11] A. Burger, D. Davidson, Y. Yang, and R. Baldock. Integrating partonomic hierarchies in anatomy ontologies. *BMC Bioinformatics*, 5(1):184, 2004.
- [12] S. Carpendale, M. Chen, D. Evanko, N. Gehlenborg, C. Görg, L. Hunter, F. Rowland, M.-A. Storey, and H. Strobel. Ontologies in Biological Data Visualization. *IEEE Computer Graphics and Applications*, 34(2):8–15, 2014.
- [13] B. Cuenca-Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, and U. Sattler. OWL 2: The next step for OWL. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 6(4):309–322, 2008.
- [14] I. C. S. da Silva, C. M. D. S. Freitas, and G. Santucci. An Integrated Approach for Evaluating the Visualization of Intensional and Extensional Levels of Ontologies. In *Proceedings of the 2012 BELIV Workshop: Beyond Time and Errors-Novel Evaluation Methods for Visualization*, p. 2. ACM, 2012.
- [15] M. d'Aquin and N. F. Noy. Where to publish and find ontologies? A survey of ontology libraries. *Journal of Web Semantics*, 11:96–111, 2012.
- [16] B. Drury, R. Fernandes, M.-F. Moura, et al. A survey of semantic web technology for agriculture. *Information Processing in Agriculture*, 2019.
- [17] M. Dudaš, S. Lohmann, V. Svátek, and D. Pavlov. Ontology visualization methods and tools: a survey of the state of the art. *The Knowledge Engineering Review*, 33:e10.1–39, 2018.
- [18] C. Dunne and B. Shneiderman. Motif Simplification: Improving Network Visualization Readability with Fan, Connector, and Clique Glyphs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 3247–3256. ACM, 2013.
- [19] A. J. Enright, B. John, U. Gaul, T. Tuschl, C. Sander, and D. S. Marks. MicroRNA targets in Drosophila. *Genome Biology*, 5(1):R1, 2003.
- [20] A. Field, J. Miles, and Z. Field. *Discovering Statistics Using R*. SAGE Publications, 2012.
- [21] J. Fraser, N. Adams, A. Macintosh, A. McKay-Hubbard, T. P. Lobo, P. F. Pardo, R. C. Martínez, and J. S. Vallecillo. Knowledge Management Applied to E-Government Services: The Use of an Ontology. In *IFIP International Working Conference on Knowledge Management in Electronic Government*, pp. 116–126. Springer, 2003.
- [22] Gene Ontology Consortium. The Gene Ontology in 2010: extensions and refinements. *Nucleic Acids Research*, 38(suppl 1):D331–D335, 2009.
- [23] Gene Ontology Consortium. Gene Ontology Consortium: going forward. *Nucleic Acids Research*, 43(D1):D1049–D1056, 2015.
- [24] M. Graham and J. Kennedy. Visual exploration of alternative taxonomies through concepts. *Ecological Informatics*, 2(3):248–261, 2007.
- [25] A. Guo, R. Racz, J. Hur, Y. Lin, Z. Xiang, L. Zhao, J. Rinder, G. Jiang, Q. Zhu, and Y. He. Ontology-based collection, representation and analysis of drug-associated neuropathy adverse events. *Journal of Biomedical Semantics*, 7(1):29, 2016.
- [26] J. Hanna, E. Joseph, M. Brochhausen, and W. R. Hogan. Building a drug ontology based on RxNorm and other sources. *Journal of Biomedical Semantics*, 4(1):44, 2013.
- [27] J. Hastings, P. de Matos, A. Dekker, M. Ennis, B. Harsha, N. Kale, V. Muthukrishnan, G. Owen, S. Turner, M. Williams, et al. The ChEBI reference database and ontology for biologically relevant chemistry: enhancements for 2013. *Nucleic Acids Research*, 41(D1):D456–D463, 2012.
- [28] T. F. Hayamizu, R. A. Baldock, and M. Ringwald. Mouse anatomy ontologies: enhancements and tools for exploring and integrating biomedical data. In *Mammalian Genome*, 2015.
- [29] Y. He, S. Sarntinvijai, Y. Lin, Z. Xiang, A. Guo, S. Zhang, D. Jagannathan, L. Toldo, C. Tao, and B. Smith. OAE: The Ontology of Adverse Events. *Journal of Biomedical Semantics*, 5(1):29, 2014.
- [30] J. Heer and S. K. Card. DOITrees revisited: Scalable, space-constrained visualization of hierarchical data. In *Proceedings of the Working Conference on Advanced Visual Interfaces, AVI '04*, pp. 421–424. ACM, New York, NY, USA, 2004.
- [31] M. Horridge. OWLViz. [protegewiki.stanford.edu/wiki/OWLViz](http://protegewiki.stanford.edu/wiki/OWLViz), [github.com/protegeproject/owlviz](https://github.com/protegeproject/owlviz), 2005. Accessed June 2019.
- [32] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From SHIQ and RDF to OWL: The Making of a Web Ontology Language. *Journal of Web Semantics*, 1(1):7–26, 2003.
- [33] J. Hur, A. Özgür, and Y. He. Ontology-based literature mining and class effect analysis of adverse drug reactions associated with neuropathy-inducing drugs. *Journal of Biomedical Semantics*, 9(1):17, 2018.
- [34] S. V. Iyer, R. Harpaz, P. LePend, A. Bauer-Mehren, and N. H. Shah. Mining clinical text for signals of adverse drug-drug interactions. *Journal of the American Medical Informatics Association*, 21(2):353–362, 2013.
- [35] Z. L. Jiao, Q. Liu, Y.-F. Li, K. Marriott, M. Wybrow, et al. Visualization of large ontologies with landmarks. In *GRAPP/IVAPP*, pp. 461–470, 2013.
- [36] C. Jonquet, A. Toulet, E. Arnaud, S. Aubin, E. D. Yeumo, V. Emonet, J. Graybeal, M.-A. Laporte, M. A. Musen, V. Pesce, et al. AgroPortal: A vocabulary and ontology repository for agronomy. *Computers and Electronics in Agriculture*, 144:126–143, 2018.
- [37] A. Jurcik. Knoocks - Ontology Visualization Plug-in for Protégé. In *Proceeding of CESCOG 2012: The 16th Central European Seminar on Computer Graphics*, 2012.
- [38] A. Katifori, C. Halatsis, G. Lepouras, C. Vassilakis, and E. Giannopoulou. Ontology Visualization Methods – A Survey. *ACM Computing Surveys (CSUR)*, 39(4), 2007.
- [39] J. B. Kruskal and J. M. Landwehr. Icicle Plots: Better Displays for Hierarchical Clustering. *The American Statistician*, 37(2):162–168, 1983.
- [40] S. Kuhar and V. Podgorelec. Ontology Visualization for Domain Experts: a New Solution. In *2012 16th International Conference on Information Visualisation*, pp. 363–369. IEEE, 2012.
- [41] M. Kuhn, M. Campillos, I. Letunic, L. J. Jensen, and P. Bork. A side effect resource to capture phenotypic effects of drugs. *Molecular Systems Biology*, 6(1):343, 2010.
- [42] Y. Lin and Y. He. Ontology representation and analysis of vaccine formulation and administration and their effects on vaccine immune responses. *Journal of Biomedical Semantics*, 3(1):17, 2012.
- [43] B. Liu, J. Li, and M. J. Cairns. Identifying miRNAs, targets and functions. *Briefings in Bioinformatics*, 15(1):1–19, 2012.
- [44] S. Lohmann, V. Link, E. Marbach, and S. Negru. WebVOWL: Web-based visualization of ontologies. In *International Conference on Knowledge Engineering and Knowledge Management*, pp. 154–158. Springer, 2014.
- [45] C. M. Machado, D. Rebholz-Schuhmann, A. T. Freitas, and F. M. Couto. The semantic web in translational medicine: current applications and future directions. *Briefings in Bioinformatics*, pp. 89–103, 2013.
- [46] H. F. Moed. *Citation Analysis in Research Evaluation*, vol. 9. Springer Science & Business Media, 2006.
- [47] C. J. Mungall, C. Torniai, G. V. Gkoutos, S. E. Lewis, and M. A. Haendel. Uberon, an integrative multi-species anatomy ontology. *Genome Biology*, 13(1):R5, 2012.
- [48] T. Munzner, F. Guimbretière, S. Tasiran, L. Zhang, and Y. Zhou. TreeJuxtaposer: Scalable Tree Comparison using Focus+ Context with Guaranteed Visibility. *ACM Transactions on Graphics (TOG)*, 22(3):453–462, 2003.
- [49] S. Nam, B. Kim, S. Shin, and S. Lee. miRGator: an integrated system for functional annotation of microRNAs. *Nucleic Acids Research*, 36(suppl.1):D159–D164, 2007.

- [50] H. Niroumand, M. F. M. Zain, and M. Jamil. Statistical Methods for Comparison of Data Sets of Construction Methods and Building Evaluation. *Procedia-Social and Behavioral Sciences*, 89:218–221, 2013.
- [51] C. Nobre, N. Gehlenborg, H. Coon, and A. Lex. Lineage: Visualizing Multivariate Clinical Data in Genealogy Graphs. *IEEE Transactions on Visualization and Computer Graphics*, 2018.
- [52] N. Noy, R. Ferguson, and M. Musen. The Knowledge Model of Protégé-2000: Combining Interoperability and Flexibility. In *Knowledge Engineering and Knowledge Management Methods, Models, and Tools*, pp. 17–32. Springer, Berlin, Heidelberg, 2000.
- [53] N. F. Noy, N. H. Shah, P. L. Whetzel, B. Dai, M. Dorf, N. Griffith, C. Jonquet, D. L. Rubin, M.-A. Storey, C. G. Chute, et al. BioPortal: ontologies and integrated data resources at the click of a mouse. *Nucleic Acids Research*, 37(suppl\_2):W170–W173, 2009.
- [54] S. Peroni and D. Shotton. The SPAR Ontologies. In *International Semantic Web Conference*, pp. 119–136. Springer, 2018.
- [55] A. Saghaei. Visualizing Ontologies - A Literature Survey. In *International Conference on Conceptual Structures*, pp. 204–221. Springer, 2016.
- [56] M. Salvadores, P. R. Alexander, M. A. Musen, and N. F. Noy. BioPortal as a Dataset of Linked Biomedical Ontologies and Terminologies in RDF. *Semantic Web*, 4(3):277–284, 2013.
- [57] S. Sarntivijai, S. Zhang, D. G. Jagannathan, S. Zaman, K. K. Burkhart, G. S. Omenn, Y. He, B. D. Athey, and D. R. Abernethy. Linking MedDRA-coded clinical phenotypes to biological mechanisms by the Ontology of adverse events: a pilot study on tyrosine kinase inhibitors. *Drug Safety*, 39(7):697–707, 2016.
- [58] H.-J. Schulz, S. Hadlak, and H. Schumann. The Design Space of Implicit Hierarchy Visualization: A Survey. *IEEE Transactions on Visualization and Computer Graphics*, 17(4):393–411, 2011.
- [59] N. Shah and M. Musen. Ontologies for Formal Representation of Biological Systems. In *Handbook on Ontologies*, pp. 445–461. Springer, 2009.
- [60] L. Shi, Q. Liac, X. Sun, Y. Chen, and C. Lin. Scalable network traffic visualization using compressed graphs. In *2013 IEEE International Conference on Big Data*, pp. 606–612, 2013.
- [61] B. Smith, W. Ceusters, B. Klagges, J. Köhler, A. Kumar, J. Lomax, C. Mungall, F. Neuhaus, A. L. Rector, and C. Rosse. Relations in biomedical ontologies. *Genome Biology*, 6(5):R46, 2005.
- [62] J. Sprague, L. Bayraktaroglu, D. Clements, T. Conlin, D. Fashena, K. Frazer, M. Haendel, D. G. Howe, P. Mani, S. Ramachandran, et al. The Zebrafish Information Network: the zebrafish model organism database. *Nucleic Acids Research*, 34(suppl 1):D581–D585, 2006.
- [63] M. Q. Stearns, C. Price, K. A. Spackman, and A. Y. Wang. SNOMED clinical terms: overview of the development process and project status. In *Proceedings of the AMIA Symposium*, p. 662. American Medical Informatics Association, 2001.
- [64] R. Stevens and P. Lord. Application of Ontologies in Bioinformatics. In *Handbook on Ontologies*, pp. 735–756. Springer, 2009.
- [65] M.-A. Storey, M. Musen, J. Silva, C. Best, N. Ernst, R. Ferguson, and N. Noy. Jambalaya: Interactive visualization to enhance ontology authoring and knowledge acquisition in Protégé. In *Workshop on Interactive Tools for Knowledge Capture*, vol. 73, 2001.
- [66] C. Wagner, K. S. Cheung, R. K. Ip, and S. Bottcher. Building Semantic Webs for e-government with Wiki technology. *Electronic Government*, 3(1):36–55, 2006.
- [67] L. Wang, M. Li, J. Xie, Y. Cao, H. Liu, and Y. He. Ontology-based systematical representation and drug class effect analysis of package insert-reported adverse events associated with cardiovascular drugs used in China. *Scientific Reports*, 7(1), 2017.
- [68] M. O. Ward. A Taxonomy of Glyph Placement Strategies for Multidimensional Data Visualization. *Information Visualization*, 1(3–4):194–210, 2002.
- [69] R. Winnenburg, A. Sorbello, and O. Bodenreider. Exploring adverse drug events at the class level. *Journal of Biomedical Semantics*, 6(1):18, 2015.
- [70] V. Yoghoudjian, T. Dwyer, K. Klein, K. Marriott, and M. Wybrow. Graph Thumbnails: Identifying and Comparing Multiple Graphs at a Glance. *IEEE Transactions on Visualization and Computer Graphics*, 24(12):3081–3095, 2018.
- [71] S. Yoon and G. De Micheli. Prediction of regulatory modules comprising microRNAs and target genes. *Bioinformatics*, 21(suppl 2):ii93–ii100, 2005.
- [72] S. Zaman, S. Sarntivijai, and D. R. Abernethy. Use of Biomedical Ontologies for Integration of Biological Knowledge for Learning and Prediction of Adverse Drug Reactions. *Gene Regulation and Systems Biology*, 11:1177625017696075, 2017.