Virtual Fly Brain - Using OWL to support the mapping and genetic dissection of the *Drosophila* brain.

David Osumi-Sutherland[1], Marta Costa[2], Gregory S.X.E. Jefferis[3]

European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cams, UK

Abstract. A massive effort is underway to map the struture of the *Drosophila* nervous system and to genetically dissect its function. Virtual Fly Brain (VFB; http://www.virtualflybrain.org) is a popular, OWL-based resource providing neuroinformatics support for this work. Here we present details of the current use of OWL on VFB - underlying tools for searching and querying across curated information from the literature in combination with information mined from bulk data sets.
In order to keep reasoning fast and scaleable, we have, up to now, restricted expressiveness to the EL profile of OWL and used the ELK reasoner. As a result, we have been unable to provide queries involving negation, despite there being cases where there is clear user demand and sufficient information to support these queries. Recent developments in reasoning technology may make these queries practical. We present ontology design patterns to support the queries with negation that our users want.

Keywords: OWL, neurobiology, neuron, DL reasoning, negation, closure axioms, ontology design pattern

1 Introduction

1.1 Mapping and genetically dissecting the *Drosophila* nervous system

A massive effort is underway to map the neural circuitry of the *Drosophila* nervous system and to genetically dissect its function. New microscopy and image analysis techniques are facilitating the collection and integration of the massive 3D image data sets required to map the structure and connectivity of the nervous system down to the single neuron level [REFS]. New genetic techniques allow researchers to precisely target elements of the neural circuitry to inhibit or activate it in order assess the effects of nervous system function and behavior [REFS]. The scale of this effort, and the huge volumes of data involved, mean that its success depends on suitable informatics support. Virtual Fly Brain (VFB) [3, 4] is an OWL-based, open source resource dedicated to this role. Usage is growing rapidly among the community it serves. The site currently gets 15-20,000 page views per month.

The adult *Drosophila* nervous system contains an estimated 200,000 neurons [REF?]. These can be grouped into classes that share similar location, morphology and lineage. The number of such classes is likely to be at least an order of magnitude smaller than the number of neurons [REF - PC Rubin?]. Mapping the neural circuitry of *Drosophila* requires ways to track the classification of these neurons and their properties, including their relationships to each other and to the gross anatomy of the nervous system, musculature, sense organs and neuro-endocrine system. This work requires synthesis of many of qualitative assertions from the literature and its integration with information from bulk data sources, much of it quantitative. OWL is an ideal technology for building and maintaining these queryable classifications. There will always be a need for direct mathematical access to quantitative data. But if suitable cutoffs can be chosen to make qualitative assertions from quantitative data, OWL provides a means to integrate qualitative and quantitative data into a queryable whole.

Modulating the activity of specific neuron classes requires finding reagents whose expression sufficiently specific. Finding such reagents frequently requires mining 3D image data of expression patterns. Integrating the phenotypic results of modulating neuronal activity into the bigger picture of nervous system function requires ways to keep track of the phenotypes associated with modulating the neuronal activity of connected neurons. Annotation with OWL ontology terms - either semi-formalised in a database or fully formalised in an OWL knowledgeBase provides a means of storing this information in queryable form.

1.2 The Drosophila anatomy ontology

Virtual Fly Brain is built around the *Drosophila* anatomy ontology (DAO, Costa et al., 2013), an OWL ontology of *Drosophila* anatomy, over 45% of which (3875/8576 classes) is devoted to representing neuroanatomy. The DAO is largely manually curated from the literature and includes a large textual component in the form of referenced synonym lists and definitions/descriptions - making it searchable by and accessible to biologists. These are used to drive auto-suggestion based searching on VFB and to populate term information pages for specific neuron classes and nervous system regions. The DAO is also richly formalised, using 44 object properties in >17000 Subclassing axioms and >2000 Equivalent Class axioms. This axiomatisation infers almost 50% of >10,000 classifications and allows a rich variety of biologically interesting queries.

1.3 Annotation queries

One major usage of VFB is as a means to query for expression of genes, transgenes and phenotypes in specified anatomical classes. These queries use information curated from the literature and bulk data sets by VFB and FlyBase curators using an semi-formalised tagging system. All queries of these annotations start with a query for subclasses, parts and overlapping cells. The resulting list is then used to query the FlyBase SQL database of annotations. 10's of thousands of annotations are available from these queries.

2 OWL queries and design patterns for neuroanatomy

The DAO uses an integrated set of relations and design patterns to classify neurons according to their location, connectivity, lineage and function [3, 4]. The neuronal connectivity relations, along with some basic mereological reasoning, drive the query system on VFB (see figure 1). VFB takes advantage of term classification in the DAO to serve only queries that are appropriate to the term displayed. So, for example, the queries available for neurons are different to those available for brain regions.

The typical mereological relationship between a neuron and gross neuroanatomy is overlap: most neurons have parts in many parts of the brain. In an insect brain, each neuron has a cell body (soma) in the cortex and has long, branching projections that extend to multiple brain regions. Many projections bundle (fasciculate) together to form tracts. On exiting a tract, the projection enters a region called neuropil where it typically branches extensively and connects to other neuron projections via synapses.

The *Drosophila* brain contains many neuron classes that can be defined via some combination of: soma location, tracts fasciculated with; neuropils in which they form input or output synaptic connections with other neurons; neuron classes synapsed with; the developmental origin of the neuron. The DAO takes advantage of this to automate classification of neurons based on these properties via EquivalentClass expressions.

Central to the basic mereological reasoning on VFB is an **overlaps** relation defined using **part_of** and its inverse **has_part**

X **overlaps** Y iff: *exists some* Z *and* X **part_of** Z *and* Y **has_part** Z[1]
part_of *subPropertyOf* **overlaps**
has_part *subPropertyOf* **overlaps**
has_part *o* **part_of** *subPropertyOf* **overlaps**
overlaps *o* **part_of** *subPropertyOf* **overlaps**
has_part *o* **overlaps** *subPropertyOf* **overlaps**

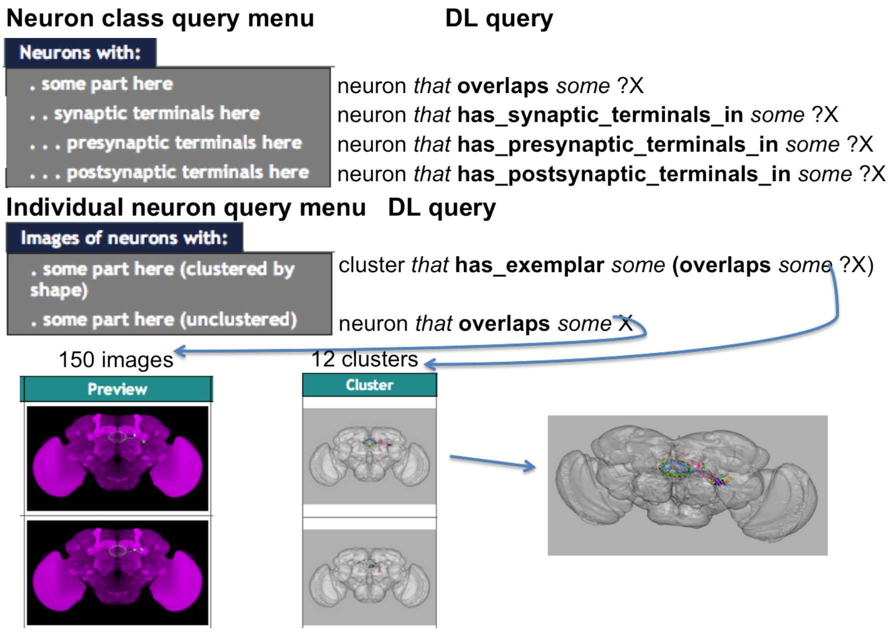
The property chains allow inference over partonomy. This is central to the function of the query system on VFB - allowing queries for overlap from any level of granularity in the partonomy.

Typically, **overlaps** is too abstract to be directly useful in class restrictions. Instead we use a range of subproperties of **overlaps** that record something useful about the nature of the overlap, such as which tract(s) a neuron fasciculates with and which neuropils it form synapses in. Like **overlaps**, relations recording synaptic terminal location also propagate over partonomy via property chains, allowing queries from any level of the partonomy. For example:

[1] **part_of** and **has_part** are both transitive and reflexive; **part_of** *inverseOf* **has_part**

has_synaptic_terminal_in *o* **part_of** *subPropertyOf* **has_synaptic_terminal_in**
has_part *o* **has_synaptic_terminal_in** *subPropertyOf* **has_synaptic_terminal_in**

VFB also provides combinatorial query functionality, via its query builder tool[2] allowing users to query for neurons based on their pattern of synapsing. This functionality is currently limited to query legs combined with‘*and*’, and does not support negation.



Neuron class query menu
DL query
Neurons with:
. some part here
. . synaptic terminals here
. . . presynaptic terminals here
. . . postsynaptic terminals here
Individual neuron query menu
DL query
Images of neurons with:
. some part here (clustered by shape)
. some part here (unclustered)
150 images
12 clusters
Preview
Cluster

neuron *that* **overlaps** *some* ?X
neuron *that* **has_synaptic_terminals_in** *some* ?X
neuron *that* **has_presynaptic_terminals_in** *some* ?X
neuron *that* **has_postsynaptic_terminals_in** *some* ?X

cluster *that* **has_exemplar** *some* **(overlaps** *some* ?X)
neuron *that* **overlaps** *some* X

Fig. 1. VFB query menus with the DL queries they run.

3 Integration of images using OWL

Neurobiology is a very visual subject. While it is useful to read both informal and formal descriptions of neuron classes and brain regions, there is no substitute for being able to see images of them. VFB is built around a standard, 3D adult brain image. Major brain regions are defined as 3D painted regions on this image according to an expert-defined standard [REF:BrainName]. These regions are modelled as individual members of the relevant ontology classes, but are also related to brain region classes via an axiom of the form:

has_exemplar *value* ‘individual region’

[2] `http://www.virtualflybrain.org/site/tools/query_builder/`

This indicates that the individual provides a standard reference for the boundaries of a brain region. A simple DL query is used to find images to illustrate term pages for these brain regions.

VFB also incorporates large datasets of 3D images of single neurons (>16,000), neuron clones (>200) and expression patterns (>3500). As for painted brain regions, structures depicted in these images are modelled as OWL individuals. Importantly, all of these images are registered (morphed) onto the standard brain. This allows direct comparison - both automated and manual - of registered images.

From image analysis, we can determine which gross brain regions a neuron, clone or expression pattern *overlaps*, recording this using a *Type* statement on the individual. These drive queries for single neuron images by location (figure 1). A more sophisticated form of image analysis, developed by G Jefferis [REF preprint], compares pairs of neurons, giving each pair a score for similarity of morphology and location. A clustering algorithm is then used to group neurons with similar morphology and location and to assign an exemplar neuron for each cluster.

We treat clusters as individuals, with single neurons standing in a **member_of** relationship to a cluster. A subproperty of **member_of**, **exemplar_of**, is used to relate exemplars to clusters. This simple formalism allows VFB to group the very large numbers of images that often result from queries of brain regions for ovelapping neurons into a much smaller number of clusters of similar neurons (see figure 1).

In many cases, the resulting clusters correspond largely or completely to well characterized neurons from the literature, for which the DAO has classes defined by lineage, tract and location of synaptic connections. Where this is the case, we add manual typing statements. In other cases, manual annotation of neurons with *Type* statements provides sufficient information for automated classification in the ontology.

3.1 An owl design pattern for modelling expression patterns

A key aim of VFB is to provide a means for biologists to find candidate transgenes suitable for use in genetic dissection of nervous system function. This is aided immensely by providing images of the expression patterns found.

We treat expression patterns as anatomical structures defined as the mereological sum of all cells that express a particular gene or transgene. We record the expressed gene or transgene shown in all images, including those of neurons and neuron clones as well as expression patterns. But individual neurons and clones are typically fragments of expression patterns. Users would find it very unintuitive if queries for expression patterns mixed in images of expression pattern fragments. But once an expression pattern has been found, it is useful to be able to get a list of component parts.

We define **expresses** using slightly more expressive logic than OWL can cope with:

> x **expresses** y iff: x **has_part** *some* cell *and* for all cell(c) and **part_of** y: ('gene expression' *and* **has_product** *some* y) **occurs_in** c

An expression pattern of a gene/transgene can then be defined and its partonomy populated using the pattern:

> 'gene B expression pattern' *EquivalentTo*: 'expression pattern' *that* **expresses** *some* 'gene B'

> *GCI*: **expresses some** 'gene B' *EquivalentTo* **part_of** *some* 'B expression pattern'

This system is currently used on VFB to provide images of transgene expression patterns found via SQL queries. The above formalisation provides an obvious way to convert the semi-formalised annotations in SQL to OWL

For brain regions:

'expression pattern of X' **overlaps** *some* 'brain region Y'

For cells, we can make a stronger assertion:

'expression pattern of X' **has_part**[3] *some* 'cell Y'

We can then find anatomical structures in which there is some expression via "**overlaps** *some* X"

As discussed in the next section, this formalisation can be used to make safe queries for expression patterns involving negation possible.

4 Beyond EL: supporting queries with negation.

In order to remain computationally tractable and scalable, VFB restricts expressiveness to the EL profile of OWL and uses the ELK reasoner [2] during development and to drive live OWL queries on the site. Recent advances in reasoning technology may make scaling with more expressive forms of OWL practical. For example, Zhou and colleagues have recently published impressive results for fast query answering by combining triple store based RL reasoning with a HermiT DL reasoner [5].

Some types of queries that would be extremely useful to our users require more expressiveness. In particular, there are a number of cases where queries involving negation would be useful. For example, for some neurons, we know all of the brain regions overlapped, all of the tracts fasciculated with and the location of all synaptic terminals. It would be useful, in such cases, to allow users to add negative legs to the compound queries for neuron classes that VFB already supports. For some transgene expression patterns in the adult brain, we have both negative an positive assertions about where a transgene is expressed. It would be very useful for researchers to be able to add negative clauses to queries for expression as this can be critical to attempts to find sufficiently specific reagents to use in experiments to genetically dissect the function of specific neurons and brain regions.

[3] **has_part** entails **overlaps**

The most efficient way to support queries involving negation is to a combination of closure axioms and disjointness declarations. For example, the neuron DL1 adPN fasciculates with only one tract, the iACT. We currently record this as:

‘DL1 adPN’ *subClassOf* **fascicualtes_with** *some* iACT

But if we also have the axioms:

‘DL1 adPN’ *subClassOf* **fascicualtes_with** *only* iACT
'great commissure’ *disjointWith* iACT

Then we can find 'DL1 adPN’ with the query:

neuron *and not* (**fascicualtes_with** *some* ‘great commissure’)

For cases where a neuron **fasciculates_with** multiple tracts, the closure axioms can simply combine multiple classes using *or*. Unfortunately, our use of inference over partonomy rules out this pattern of closure axioms for many important relations used in querying. For example:

overlaps *o* **part_of** *subPropertyOf* **overlaps**
X **overlaps** *some* Y
X **overlaps** *only* Y
Y **part_of** *some* Z
Z *disjointWith* X
=>inconsistency: X **overlaps** *some* Z, X not (**overlaps** *some* Z)

We can get around this by using closure axioms of the form “**rel** *only* (**has_part** some X)” and declaring spatial disjointness between brain regions (which also provide a useful integrity check). Spatial disjointness can be declared using a simple GCI:

part_of *some* X *disjointWith* **part_of** *some* Y

For example, we can represent that the neuron DL1 adPN only has synaptic terminals in DL1v(part of the antennal lobe) and the lateral horn[4] with:

‘DL1 adPN’
subClassOf: **has_synaptic_terminals_in** *some* DL1
subClassOf: **has_synaptic_terminals_in** *some* ‘lateral horn’
subClassOf: **has_synaptic_terminals_in** *only* (**has_part** *only* (DL1 *or* ‘lateral horn’))
‘fan-shaped body’ *subClassOf*: **part_of** *some* ‘central complex’
DL1 *subClassOf*: **part_of** *some* ‘antennal lobe’

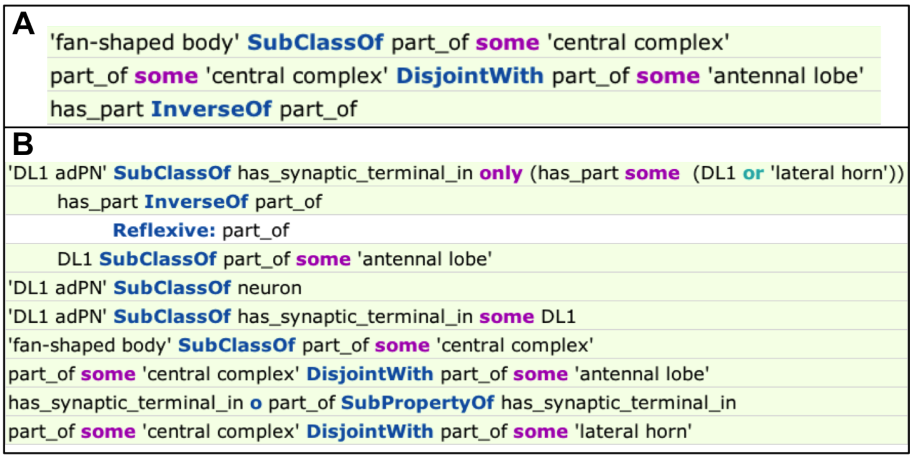
[4] There is actually one additional region, but we simplify here in order to provide a more compact example

With ‘antennal lobe’, ‘lateral horn’ and ‘central complex’ declared spatially disjoint , DL1 adPN is returned by the query:

> neuron that (**has_synaptic_terminal_in** *some* ‘antennal lobe’) and not (**has_synaptic_terminal_in** *some* ‘fan-shaped body’)

An explanation is shown in figure 2B). There is no need to assert **has_part** relationships. The *inverseOf* axiom between **has_part** and **part_of** is sufficient to infer not **has_part** from spatial disjointness axioms (figure 2A).

This pattern is also dependent on the reflexive nature of **part_of** and **has_part** (figure 2B).



'fan-shaped body' SubClassOf part_of some 'central complex'
part_of some 'central complex' DisjointWith part_of some 'antennal lobe'
has_part InverseOf part_of

'DL1 adPN' SubClassOf has_synaptic_terminal_in only (has_part some (DL1 or 'lateral horn'))
has_part InverseOf part_of
Reflexive: part_of
DL1 SubClassOf part_of some 'antennal lobe'
'DL1 adPN' SubClassOf neuron
'DL1 adPN' SubClassOf has_synaptic_terminal_in some DL1
'fan-shaped body' SubClassOf part_of some 'central complex'
part_of some 'central complex' DisjointWith part_of some 'antennal lobe'
has_synaptic_terminal_in o part_of SubPropertyOf has_synaptic_terminal_in
part_of some 'central complex' DisjointWith part_of some 'lateral horn'

Fig. 2. A. Explanation for why the query “*not* **has_part** *some* ‘fan-shaped body’ ” returns ‘antennal lobe’. Note that direct assertion of **has_part** restriction axioms are not necessary. **B.** Explanation for why the query “neuron that (**has_synaptic_terminal_in** *some* ‘antennal lobe’) and not (**has_synaptic_terminal_in** *some* ‘fan-shaped body’)” returns the neuron ‘DL1 adPN’.

Negative query legs in compound queries for expression patterns would be especially useful to our users. Our ability to provide these is limited by the extent to which it is possible to specify which regions lack expression. It is generally not possible to provide an exhaustive list of all regions that have some part with some level of expression for a closure axiom with overlaps. However some datasets come with explicit assertions about regions not overlapped. For example, the largest transgene expression dataset that VFB currently hosts [Ref Jennett] was provided with annotations recording the presence or absence of expression in every major neuropil in the adult brain. These can easily be translated programatically into restriction axioms asserting **overlaps** and *not* **overlaps** on expression pattern classes.

5 Discussion and future directions

Virtual Fly Brain uses OWL to provide a unique service to the *Drosophila* neurobiology community, integrating a wealth of information from the literature and bulk datasets into an easily queryable resource. Much of this would be difficult or impossible to provide using a conventional relational database. OWL provides a sustainable way to develop and maintain a queryable classification of anatomical structures and neurons. OWL axiomatisation allowing inference over partonomy drives queries that return complete information about neuronal overlap and synaptic terminal location from any level of the partonomy. OWL reasoning also provides a way to group annotations of expression and phenotypes based on classification, partonomy and cell overlap. This massively enriches the results of annotation queries.

VFB has so far avoided taking advantage of the full expressiveness of OWL. Reasoners such as HermiT and FaCT++ [REFS] are many orders of magnitude slower at classifying the DAO and answering queries than ELK, the EL reasoner that we rely on. They do not even complete when reasoning across the DAO combined with the VFB knowledgeBase of individuals. However, we have one use case for which DL expressiveness would be extremely useful: compound queries for neurons or expression patterns involving negation.

There are two major barriers to achieving this. The most serious barrier is the ability to query across an ontology or combined ontology and knowledgeBase with DL expressiveness. Zhou and colleagues have recently published impressive results for fast query answering by combining triple store based RL reasoning with a HermiT DL reasoner [5]. We are working with the authors to test query speed for compound queries with negation for test datasets using the design patterns outlined in this paper.

A more cleaerly sumountable barrier is the lack of tooling support for some of the axiomatisation required in the design patterns we propose. In particular, adding GCIs to record spatial disjointness is currently very tedious to do by hand in Protege 5. This may be achievable by scripting, but in order for the approach to be accessible for any ontology builder ideally this would be achievable via a plugin for a popular editor such as Protege. By analogy with support for the addition of class disjointness axioms in Protege, this work by allowing users to navigate down a partonomy tree, adding disjointness axioms to whole sets of sibling terms at once.

6 Methods

For details of construction and maintenance of the *Drosophila* anatomy ontology please see Costa et al., 2013 [1]. The ontology is available from `http://purl.obolibrary.org/obo/fbbt`

VFB is an open source project. All code is available from `https://github.com/VirtualFlyBrain` OWL individuals files used on VFB are available from `https://github.com/VirtualFlyBrain/VFB_owl/tree/master/src/owl`

A test ontology illustrating implementation of the DL patterns for negative queries can be found here: TBA!

6.1 VFB architecture

All queries for anatomical classes or individuals on VFB are live DL queries via the elk OWL reasoner. All queries of annotation begin with a DL query for subclasses, parts and overlapping cells. The resulting list is then used to query annotations store in the FlyBase Postgresql database. More details of the overall architecture of the project cen be found at `https://github.com/VirtualFlyBrain/VFB#overall-architecture-of-project`

6.2 Database representation of OWL individuals

Details of individuals are maintained in a SQL database (`https://github.com/VirtualFlyBrain/VFB_owl/wiki/Individuals-DB`) and programmatically converted to OWL via scripting over the OWL-API (`https://github.com/VirtualFlyBrain/VFB_owl/`). A standard DB representation of OWL ontologies/individuals would be preferable to our bespoke solution, which limits axiom expressiveness in order to keep the DB structure simple. We are currently unaware of any viable, non-proprietary alternatives.

Author's contributions

The OWL design patterns and queries presented in this paper were designed and tested by DOS He also designed the database representation of OWL individuals and wrote the code the translates this representation into OWL. The portion of the DAO representing neuroanatomy was built by DOS and MC. MC and GJ were responsibile for all annotation, processing and analysis of images. GJ developed the clustering algorithm and had the idea of assigning examplars to clusters.

Acknowledgments.

We thank Simon Reeve for his contributions to the representation of neuroanatomy in the DAO. We thank Nestor Milyaev for his work on the VFB website.

Funding This work was largely supported by:'Standardising the representation of Drosophila anatomy and development for databases' ; BBBSRC:BB/G02233X/1 awarded 2009 to J.Douglas Armstrong, Michael Ashburner and David Osumi-Sutherland 'Structured and graphical queries for Drosophila neuroscience data'; An Isaac Newton Trust grant to Cahir O'Kane to fund the work of Marta Costa, Awarded 2012: 'Neuroinformatic identification of new types of neuron in the Drosophila brain.' VFB is currently supported by a Wellcome Trust grant to Cahir O'Kane, J. Douglas Armstrong, Gregory S.X.E Jefferis, Helen Parkinson and David Osumi-Sutherland: 'Virtual Fly Brain: a global informatics hub for Drosophila neurobiology' WT105023MA.

References

1. Marta Costa, Simon Reeve, Gary Grumbling, and David Osumi-Sutherland. The Drosophila anatomy ontology. *Journal of biomedical semantics*, 4(1):32, January 2013.
2. Yevgeny Kazakov, Markus Krötzsch, and František Simančík. Elk reasoner: Architecture and evaluation. *CEUR Workshop Proceedings*, 858, 2012.
3. N. Milyaev, D. Osumi-Sutherland, S. Reeve, N. Burton, R. A. Baldock, and J. D. Armstrong. The Virtual Fly Brain browser and query interface. *Bioinformatics*, 28(3):411–415, Feb 2012.
4. D. Osumi-Sutherland, S. Reeve, C. J. Mungall, F. Neuhaus, A. Ruttenberg, G. S. Jefferis, and J. D. Armstrong. A strategy for building neuroanatomy ontologies. *Bioinformatics*, 28(9):1262–1269, May 2012.
5. Yujiao Zhou, Yavor Nenov, Bernardo Cuenca Grau, and Ian Horrocks. Pay-as-you-go OWL query answering using a triple store. In *Proc. of the 28th Nat. Conf. on Artificial Intelligence (AAAI 14)*, 2014.