Taking a view on bio-ontologies

Simon Jupp [1]*, Andrew Gibson [2,3], James Malone [1], Helen Parkinson [1] and Robert Stevens [4]

[1]Functional Genomics Group, European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SD

[2]Biosystems Data Analysis, Swammerdam Institute for Life Sciences (SILS), University of Amsterdam, Science Park 904, 1098 XH, Amsterdam, The Netherlands

[3]Netherlands Consortium for Systems Biology, University of Amsterdam, PO Box 94215, 1090 GE, Amsterdam, The Netherlands

[4]School of Computer Science, The University of Manchester, Oxford Road, Manchester, UK M13 9PL

ABSTRACT

We present a technique for separating knowledge representation from application specific views that are currently often conflated within bio-ontologies. Many ontologies contain information for two tasks; one to represent the knowledge of some field of interest and another to support an application through providing *views* over ontologies that present the terms in a useful way for an application. We analyse this phenomenon in some bio-ontologies and suggest this separation of layers as a solution. We leave dedicated ontology languages like OWL and OBO to represent the knowledge of a field of interest, and use a more lightweight vocabulary, namely SKOS, to capture application specific views. We use this technique to encode a number of views inside the Experimental Factor Ontology. Each of these views serves a special purpose to different user communities; however, it does ensure the underlying ontology can remain for the annotation and integration of biological data. OWL and SKOS together provide a powerful, standards based, mechanism to reconstitute annotated biological data for many different application domains.

1 INTRODUCTION

Bio-ontologies have become important in the life sciences through their provision of identifiers for biomedical concepts that are defined and managed by community processes (Smith *et al.*, 2007). More effective integration, analysis and mining of information from life science datasets is being routinely and experimentally achieved as a result of their annotation with these concepts (The Gene Ontology Consortium, 2010; Camon *et al.*, 2004; Noy *et al.*, 2009; Kapushesky *et al.*, 2011). Authoritative collections of concepts have been produced such as those describing, to name a few, the characteristics of gene products in GO (Consortium, 2000), chemicals in ChEBI (Degtyarenko *et al.*, 2008) or species in the NCBI taxonomy (Federhen, 2011). As the value of annotation with ontology identifiers has been recognised, the number of bio-ontologies with different scopes has increased, as well as the number of concepts described by existing vocabularies (Castro *et al.*, 2010). Many bio-ontologies are being used to annotate biomedical data for use in different query and browsing tools. The extent of the annotation varies; some gene product data are just annotated with GO concepts (Camon *et al.*, 2004) whereas other datasets, such as those submitted to ArrayExpress (Parkinson *et al.*, 2011), require annotations that span many fields of interest, from descriptions of the experiment to the attributes of the differentially expressed genes.

A significant use of bio-ontologies and annotations is to support the users of applications that wish to view, browse and search what can be complex and high-dimensional data. The key parts of bio-ontologies that support this form of use are the hierarchical structures formed by 'is a' and often 'part of' relationships between the concepts, as well as their labels and definitions. Applications use these components to drive a presentation that allows users to inspect annotated data using criteria (concepts) with which they are most familiar. The labels within the ontology support the interaction between the user and the interface; a simple ontology-driven auto-complete function in a search box can transform an opaque dataset into a useable life science application. The hierarchical structure of bio-ontologies enables query expansion—when querying with X we also retrieve X and all the children of X as described in the ontology (e.g. medline / mesh). Bio-ontologies often include other non-hierarchical relationships between concepts that are specific to their scope that can also help applications to guide users to data or content that is semantically related to their query.

Applications that use bio-ontologies and data annotations to drive their presentation are faced with knowledge representation and presentation issues. First, when faced with a large ontology, the proportion of concepts that are relevant to (the users of) an application will be low enough that it would detract from the usability of the application if all of the concepts in the vocabulary were exposed. Thus applications need a mechanism to indicate whether or not a particular concept should be available. Second, applications will often use concepts that cover multiple biological aspects of a dataset, intersecting parts of several existing vocabularies. In order to collect, manage and use the concepts relevant to an application, developers would often like to have an ontology that only contains those concepts in which they are interested, or else have a mechanism by which they can annotate the original ontology so that their application only processes the relevant content. In this paper we refer to these kinds of collections of concepts as a 'view'.

We consider a view to be analogous to an ontology module, in that the goal is to reuse a subset of concepts from an existing ontology in a particular setting (Pathak *et al.*, 2009). Formal ontology modularisation is a process whereby the logical entailments of a set of axiomatically defined classes remains the same in both the original ontology and in the module. In this paper a view is a more

*To whom correspondence should be addressed: jupp@ebi.ac.uk lightweight collection of concepts from one or more ontologies where the identifiers and annotation components are useful to the application for navigation and query expansion, but where the logical entailments of the original ontology, whilst being preserved in the original ontology, are not required by the application.

The challenge for the developer is how to represent this information in line with existing standards and methods. Ideally, such annotations and mechanisms of annotation would also be available outside of a particular application.

In this paper we present an example of the above challenges as faced by the iKUP Browser, a web application for querying multi-omics data held in the Kidney and Urinary Pathway Knowledgebase (Jupp *et al.*, 2011). We then present a technique that separates the requirements of a bio-ontology as a representation of knowledge in a domain from the requirements of presentation of that data in an application setting, that is inline with current standards for bio-ontology representation. We employ features recently introduced into the OWL2 specification, in combination with a W3C specification for representing thesauri, controlled vocabularies and subject heading systems, the Simple Knowledge Organization System (SKOS). This method is applied to extract SKOS based views from the Experimental Factor Ontology (EFO) (Malone *et al.*, 2010), which is used to curate transcriptomics data and supports several applications.

2 USE CASE

The development of a kidney and urinary pathway knowledge base clearly demonstrates the need for a separation of the applications view of the ontology from the underlying knowledge representation. The Kidney and Urinary Pathway Knowledge Base (KUPKB) is a Semantic Web application being used by biologists in the study of kidney disease (Klein *et al.*, 2012). A Kidney and Urinary pathway ontology (KUPO) provides the underlying schema and model for the data held in the KUPKB. KUPO is an application ontology that brings together subsets of other ontologies to annotate multi-omic high-throughput data from experiments on kidney disease. KUPO re-uses concepts from existing ontologies wherever possible, importing concepts from various ontologies, including the cell (CTO), gene (GO), mouse anatomy (MAO), human disease (HDO), phenotype (PATO) and experimental factor (EFO). In bringing concepts together, the KUPO adds new OWL axioms that further strengthen the descriptions of various classes. For example, cell types are described in terms of their parts using part_of relationships to the MAO, and cellular functions are related to concepts from the GO using the capable_of relationship. An OWL reasoner is used to classify the KUPO class hierarchy, that is then used to drive queries in the KUPKB.

The KUPO and the ontologies it imports provides a rich underlying model for the KUPKB, where OWL semantics can be exploited for powerful query answering. In order to ask complex queries of the KUPKB, technical knowledge of the underlying ontologies, and Semantic Web query languages like SPARQL are needed. This kind of interaction is reserved for experts in the field, thus excluding a wide range of potential end users. To address this the iKUP browser (http://www.kupkb.org) was built to provide a user friendly interface to the KUPKB data. The iKUP browser uses the underlying ontologies to integrate and query the data. The ontological axioms allow for query expansion and the class hierarchies are exploited to provide faceted browsing of search results. However, the ontologies and their imports are not suitably organised for presentation in a user friendly interface such as the iKUP for several reasons:

- Classes towards the top of the hierarchy are often sufficiently abstract that they have little meaning to the user. For example, upper level classes in the CTO such as 'cell in vivo' and 'experimentally modified cell' were not useful to iKUP users.
- Some ontologies provide too many levels of granularity. For example, the NCI taxonomy contains twelve intermediate categories between humans (taxied:9606) and mammals (*taxid:40674*), which are unnecessary to view in iKUP where users only require a simple classification of species.
- Multiple concepts in the ontology may be suitable root concepts for navigational purposes. For example, cell type, *disease*, *species*, *function* are useful root concepts for navigation, but may not necessarily be default roots within an ontological context.
- Concepts that a user might expect to be organised hierarchically do not have true parent/child relationships in the ontology. For example, displaying partonomy and developmental relationship as part of the class hierarchy.

In the case of iKUP, each of these issues were handled in the user interface code. User feedback was used to determine which concepts from the ontology should be presented. This approach proved difficult to maintain and also means the view logic is hidden in the iKUP code and cannot be exposed in other tools, such as Protégé, outside of the iKUP applications.

To address some of these issues, the Open Biomedical Ontology (OBO) community have adopted an annotation mechanism which they use to generate subsets or 'slims' to mark up concepts in the ontology that belong to a particular view. The Gene Ontology provides slims of the ontology that are designed for specific communities or applications. The GO slims give a broad overview of the ontology content without the detail of the fine grained concepts. They are useful in applications such as over expression analysis (Yi *et al.*, 2007). When converting OBO into OWL the annotation property (oboInOwl:subset) is used to indicate that a concept belongs to a particular slim. The OBO slim pattern benefits from being simple and has been adopted in some notable bio-ontologies, however, as annotations, they lack any kind of real semantics so ontology development tools and applications need specialist knowledge before they can be exploited. OWL annotations provide an obvious mechanism for encoding view information and tools like OntoDog [1] have been developed to assist users in extracting views from an existing ontology based on it's annotations. However, there remains no common design patterns and a real lack of generic tooling to support the creation and maintenance of these views.

2.1 View requirements

We have in the KUKPB a scenario where two tasks can be easily conflated: representing entities ontologicallly and also adding information that aids presentation, navigation and searching within the application setting. This suggests separating out these two needs into separate *ontology layers* and *user layers*. We have languages

[1] http://ontodog.hegroup.org

for the ontology layer, such as OWL and OBO format, that are widely used to author ontologies. However, a standard method for capturing this *user layer* is needed if we want to share these views between applications. This standard must meet the following basic requirements:

1. To identify one or more views within an ontology;
2. Assign concepts in the ontology to one or more of the internal views;
3. Assert semantic relationships between concepts in the ontology that provide alternative navigational paths around the ontology;
4. Assert anchors in the ontology that indicate root concepts for presentation.

We propose that the W3C Simple Knowledge Organisation System (SKOS) provides a minimal model for capturing these views within an OWL ontology. Unlike OWL, the emphasis of SKOS is not so much on the formal (logical or ontological) representation of the information, but instead provides a schema in which concepts can be organized in a lightweight fashion for concept schemes, cataloguing, indexing and information retrieval tasks. SKOS models concepts as instances of one OWL class - skos:Concept. By modelling concepts as instances rather than classes, SKOS shifts the knowledge representation strategy to a different meta-level. SKOS provides hierarchical properties in broader and narrower, as well as the non-hierarchical related property. These semantic relationship lack any formal definition, however, their semantics are sufficiently defined by the W3C in order for applications to make assumptions on how they should be interpreted computationally. One of the major advantages of SKOS is that it provides a significant amount of support for describing the annotation components of an ontology, including labels, definitions and multiple languages, major desirable components of most biomedical ontologies. Although SKOS concepts cannot be (logically) defined as extensively as OWL classes, they can be (usefully) described just as well for most user-facing applications.

One of the major improvements of OWL2 was the removal of the constraint that a named OWL entity must be assigned one ’role’ in an ontology. This ’punning’ strategy means that it is now permitted to specify, for example, that an entity is both an OWL class and also an individual - thereby introducing a basic meta-modelling capability into the Semantic Web suite of specifications. Figure 1 illustrates how we can use meta-modelling to make assertions between classes in our ontology. In the way that we described OWL and SKOS above, it initially seems that they are mutually exclusive, different interpretations of the same thing. Infact we can exploit OWL2 punning to integrate both types of representation into one. We can keep the formal axiomatic view as an OWL ontology and by punning specify a set of SKOS concepts that happen to have the same name. In doing this, we can now start to refer to the SKOS individual representations of concepts in a vocabulary as if they were items in a dataset rather than defined entities as part of a formal theory.

SKOS also provides one other key modelling component when we are considering views: the concept scheme. A concept scheme is an entity to which SKOS concepts can be mapped with the skos:inScheme property. It can be used as a way of grouping together a set of concepts, and the annotation of the concept scheme can be used to add a description of why those concepts are part of that concept scheme. We can define anchors within our views using the skos:topConceptOf property. skos:topConceptOf can be used to assert that a particular concept should be viewed at the root of a particular concept scheme.

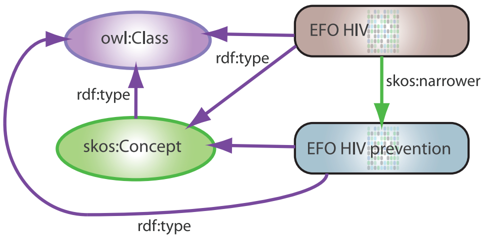

Fig. 1. OWL 2 meta-modelling allows us to assert that both the HIV concept and HIV Prevention concepts are both OWL classes and instances of the SKOS Concept class. At the instance level we can make object property assertions between HIV and HIV Prevention using the skos:narrower property.

This ability to do some meta-modelling in OWL allows us to use the SKOS vocabulary to make additional assertions on our ontology. SKOS can be used to index concepts in our ontology, and to define alternate navigational hierarchies around the ontology. These SKOS ”views” are not intended as a replacement for the OWL, but rather an extension to the underlying knowledge representation that supports the application setting. Capturing this information using a standard vocabulary like SKOS means we can begin to exploit generic SKOS tooling to support how the views are created, maintained and used. SKOS can satisfy our previous requirements as follows:

1. For requirement one, we use the skos:ConceptScheme to represent a particular view within an ontology
2. For requirement two, for each class that is to be included in the view from the ontology, we add them as an instance of skos:Concept. These concepts can then be associated to a particular view, i.e. the skos:ConceptScheme via the skos:inScheme property.
3. For requirement three we use a combination of SKOS broader, narrow and related properties to provide the appropriate structure to the view.
4. For requirement four we use the skos:hasTopConcept property to assert that particular concepts are anchors, or root concepts in a particular view.

3 THE EXPERIMENTAL FACTOR ONTOLOGY

To demonstrate the technique we attempt to extract a subset of views that are encoded with the Experimental Factor Ontology (EFO). The EFO is an application ontology developed to describe experimental variables used in transcriptomics data (Malone *et al.*, 2010). EFO brings parts of disparate bio-ontologies together to provide an application ontology for both annotating the data, and data exploration through tools such as the Gene Expression Atlas (Kapushesky *et al.*, 2011) and the ArrayExpress

Archive (Parkinson *et al.*, 2011). Data are initially annotated with ontology classes which enable more powerful searching, such as synonym expansion and traversing hierarchies based on an ontology view. The latter however, has presented challenges in the way EFO has been developed.

In constructing the EFO application ontology several expedient representational compromises were made. EFO includes two annotation properties that are used as ‘flags’ to indicate that a class should be either hidden from view in the application or that it should be used as an anchor, i.e. a starting point at which to begin browsing the hierarchy. Hidden flags are often used on classes such as upper level ontology classes such as those from the basic formal ontology (BFO), which are alien to a biologist user. Anchor flags are used on classes such as cell line and disease, these indicate common starting points of interest to users navigating the ontology in an application scenario.

New applications adopting the approach of EFO, such as the Genome Wide Association Study (GWAS) browser [2] and the European Nucleotide Archive (ENA) [3] each require subsets of classes within EFO, but viewed in bespoke ways to suit their application needs. We have, therefore, a situation where two tasks have been conflated in the EFO: representing entities in transcriptomics experiments and also adding information that aids presentation, navigation and searching within the application setting. In order to avoid duplicating and minting of new terms to serve specific application requirements, these additional views are embedded within the ontology using both logical OWL axioms and specialised annotation properties. To demonstrate the applicability of SKOS we extracted three separate views from the EFO OWL file and represented them in SKOS. We then show how these views can be visualised alongside the original EFO ontology using a generic SKOS tool called SKOSEd.

3.1 Generating EFO SKOS

We converted the various views in EFO to SKOS using bespoke scripts that extract the views based on existing annotations and convert them into SKOS concept schemes. These scripts are written with the Java OWL API (version 3) and SKOS API (version 3). These views are available to download from BioPortal under the EFO ”views” section. These views can be views in any valid SKOS aware application. The Protégé 4.1 [4] SKOSEd plugin [5] was used to view and evaluate the SKOS conversions.

For each view we follow the basic pattern:

- Classify EFO with the HermiT [6] reasoner
- Create a SKOS concept scheme for the view
- All classes that are flagged as organisational classes are discarded
- All classes flagged as part of a the current view are converted to SKOS Concepts and added to the Concept Scheme. As the same URI is used for Class and Concept, all annotations such as labels are preserved.
- All superclasses of the flagged classes up to hidden organisational classes are added as skos:broader assertion (corresponding skos:narrower assertions are also added
- Any classes flagged as branch or anchor classes are asserted as top concepts using skos:hasTopConcept in the current concept scheme
- Any flagged properties are mapped to SKOS semantic relationship. If a property is mapped to a skos relationship then class restrictions along the mapped property get translated to the appropriate skos relationship e.g. If part_of is mapped to skos:broader then (subClassOf (X, part_of some Y) becomes (X skos:broader Y).

The first view extracted is the *EFO basic view*. This view is currently used to serve both the ArrayExpress and gene expression atlas query expansion and results summary view applications. In this conversion all classes apart from organisation classes were converted for the view. Additionally, the part_of relationship was mapped to skos:broader in order to incorporate the partonomy views into the concept hierarchy.

The second view represents a view generated for the GWAS catalogue terms. By capturing our view in a standard language like SKOS we can begin to exploit existing SKOS aware tools to visualise the GWAS view in EFO for the first time. Figure 2 shows a portion of the GWAS viewed in the generic SKOSEd extension for Protégé, no special configuration was required for this view to be exposed. The terms present in this view are selected by GWAS annotators for the annotation of studies submitted to the GWAS catalogue. These terms are currently used to populate a drop down list, but will soon form part of a more sophisticated information retrieval system. Only classes flagged as ”gwas” and their parent classes were converted to SKOS concepts; all organisation classes were ignored.

The third view represents a subset of terms from the European Nucleotide Archive (ENA) [7]. These terms are used to annotate submission to the ENA, these annotations are used to describe submitted datasets which are used by other databases, such as ArrayExpress. The ENA currently only requires few terms and have little in the way of hierarchy. They have their own categories for terms, these categories have no natural place within EFO. In the case of ENA we define some additional concepts within our view in order to categorise some of the terms from EFO for the ENA application.

4 DISCUSSION

Bio-ontologies are primarily used to represent domain knowledge from areas of interest to the community, however, the application of such ontologies to data and data providing services is of increasing importance. In the case of applications, we need to separate the concerns of knowledge representation and user presentation—a classic software engineering approach. We leave the ontology as an ontology (in OWL or OBO format) and capture application knowledge in a SKOS representation, a simple transformation which is suited to the needs of an application or for local problem solving. Such a separation also means we can have different application specific *user layers* for the same *knowledge layer* or

[2] http://www.genome.gov/gwastudies/

[3] http://www.ebi.ac.uk/ena

[4] http://protege.stanford.edu

[5] http://code.google.com/p/skoseditor

[6] http://www.hermit-reasoner.com/

[7] http://www.ebi.ac.uk/ena

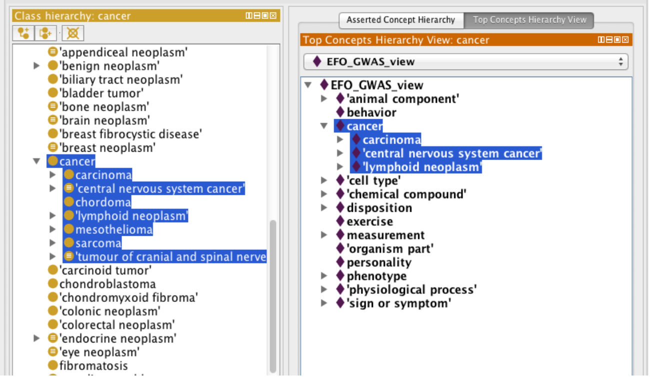

Fig. 2. The GWAS subset of EFO as views in the SKOSEd Protégé 4.1 plugin. The left hand side shows the OWL class hierarchy around Cancer, the right hand view shows the SKOS concept hierarchy representing a subset of Cancer terms relevant only to the GWAS concept scheme.

ontology, without undermining ongoing work to make domain ontologies interoperable.

The problems encountered developing applications around the KUPO and EFO highlight a scenario that will emerge many times over as more bioinformatics tools move to exploiting ontologies in user facing applications. Whilst other similar patterns may emerge, the approach outlined in the paper demonstrates how aligning to a standard vocabulary language like SKOS allows us to exploit existing infrastructure. The views extracted from EFO allow the application developers to visualise the application views in ways that were not previously possible in standard ontology editing environments. SKOS provides one means to share these views across communities and applications, and is an attractive solution for the scenarios outlined in this paper.

5 CONCLUSION

The SKOS vocabulary has some adoption in the life science ontologies—in particular the labelling and mapping properties. However, the notion of concept schemes and semantic relationships have been less well adopted, and these are components that fulfil our requirements. By taking standard approaches we allow existing tools that consume SKOS access to the terminological information of bio-ontologies. There is now a need for better tool support to enable life scientists to work with SKOS more easily. This paper demonstrates how separating the concerns of knowledge representation and user presentation into layers and adopting standards such as SKOS offers new possibilities for data sharing and re-use.

ACKNOWLEDGEMENTS

AG is supported by the BioRange programme of The Netherlands Bioinformatics Centre (NBIC; http://www.nbic.nl), supported by a BSIK grant through The Netherlands Genomics Initiative (NGI) and the research programme of the Netherlands Consortium for Systems Biology (NCSB), which is part of the Netherlands Genomics Initiative/Netherlands Organization for Scientific Research. We acknowledge funds from EMBL (JM, HP) and The National Center for Biomedical Ontology, one of the National Centers for Biomedical Computing supported by the NHGRI, the NHLBI, and the NIH Common Fund under grant U54-HG004028 (SJ).

REFERENCES

Camon, E., Magrane, M., Barrell, D., Lee, V., Dimmer, E., Maslen, J., Binns, D., Harte, N., Lopez, R., and Apweiler, R. (2004). The Gene Ontology Annotation (GOA) database: sharing knowledge in Uniprot with Gene Ontology. *Nucleic Acids Research*, **32**(suppl 1), D262–D266.

Castro, A. G., O'Neill, K., Garca-Castro, L. J., Lord, P. W., Stevens, R., Corcho, ., and Gibson, F. (2010). Developing ontologies within decentralised settings. In H. Chen, Y. Wang, and K.-H. Cheung, editors, *Semantic e-Science*, volume 11 of *Annals of Information Systems*, pages 99–139. Springer.

Consortium, T. G. O. (2000). Gene Ontology: Tool for the Unification of Biology. *Nature Genetics*, **25**, 25–29.

Degtyarenko, K., de Matos, P., Ennis, M., Hastings, J., Zbinden, M., McNaught, A., Alcntara, R., Darsow, M., Guedj, M., and Ashburner, M. (2008). ChEBI: a database and ontology for chemical entities of biological interest. *Nucleic Acids Research*, **36**(suppl 1), D344–D350.

Federhen, S. (2011). The NCBI taxonomy database. *Nucleic Acids Research*.

Jupp, S., Klein, J., Schanstra, J., and Stevens, R. (2011). Developing a kidney and urinary pathway knowledge base. *Journal of Biomedical Semantics*, **2**(Suppl 2), S7.

Kapushesky, M., Adamusiak, T., Burdett, T., Culhane, A., Farne, A., Filippov, A., Holloway, E., Klebanov, A., Kryvych, N., Kurbatova, N., Kurnosov, P., Malone, J., Melnichuk, O., Petryszak, R., Pultsin, N., Rustici, G., Tikhonov, A., Travillian, R. S., Williams, E., Zorin, A., Parkinson, H., and Brazma, A. (2011). Gene Expression Atlas update - a value-added database of microarray and sequencing-based functional genomics experiments. *Nucleic Acids Research*.

Klein, J., Jupp, S., Moulos, P., Fernandez, M., Buffin-Meyer, B., Casemayou, A., Chaaya, R., Charonis, A., Bascands, J., Stevens, R., and Schanstra, J. (2012). A novel web application to access multi-omics data on kidney disease. *Faseb J, In press.*

Malone, J., Holloway, E., Adamusiak, T., Kapushesky, M., Zheng, J., Kolesnikov, N., Zhukova, A., Brazma, A., and Parkinson, H. (2010). Modeling sample variables with an Experimental Factor Ontology. *Bioinformatics*, **26**(8), 1112–1118.

Noy, . . F., Shah, . . H., Whetzel, . . L., Dai, . ., Dorf, . ., Griffith, . ., Jonquet, . ., Rubin, . . L., Storey, . .-A., Chute, . . G., and Musen, . . A. (2009). BioPortal: ontologies and integrated data resources at the click of a mouse. *Nucleic Acids Research*, **37**(suppl 2), W170–W173.

Parkinson, H., Sarkans, U., Kolesnikov, N., Abeygunawardena, N., Burdett, T., Dylag, M., Emam, I., Farne, A., Hastings, E., Holloway, E., Kurbatova, N., Lukk, M., Malone, J., Mani, R., Pilicheva, E., Rustici, G., Sharma, A., Williams, E., Adamusiak, T., Brandizi, M., Sklyar, N., and Brazma, A. (2011). Arrayexpress update - an archive of microarray and high-throughput sequencing-based functional genomics experiments. *Nucleic Acids Research*, **39**(suppl 1), D1002–D1004.

Pathak, J., Johnson, T. M., and Chute, C. G. (2009). Survey of modular ontology techniques and their applications in the biomedical domain. *Integr. Comput.-Aided Eng.*, **16**, 225–242.

Smith, B., Ashburner, M., Rosse, C., Bard, J., Bug, W., Ceusters, W., Goldberg, L. J., Eilbeck, K., Ireland, A., Mungall, C. J., The OBI Consortium, Leontis, N., Rocca-Serra, P., Ruttenberg, A., Sansone, S.-A., Scheuermann, R. H., Shah, N., Whetzel, P. L., and Lewis, S. (2007). The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nat Biotechnol*, **25**(11), 1251–1255.

The Gene Ontology Consortium (2010). The Gene Ontology in 2010: extensions and refinements. *Nucleic Acids Research*, **38**, D331–D335.

Yi, G., Sze, S.-H., and Thon, M. R. (2007). Identifying clusters of functionally related genes in genomes. *Bioinformatics*, **23**(9), 1053–1060.