BIOINFORMATICS ORIGINAL PAPER

Vol. 26 no. 8 2010, pages 1112–1118
doi:10.1093/bioinformatics/btq099

Databases and ontologies

Advance Access publication March 3, 2010

Modeling sample variables with an Experimental Factor Ontology

James Malone[1,*], Ele Holloway[1], Tomasz Adamusiak[1], Misha Kapushesky[1], Jie Zheng[2], Nikolay Kolesnikov[1], Anna Zhukova[1], Alvis Brazma[1] and Helen Parkinson[1]

[1]Microarray Informatics Team, EMBL-EBI, Wellcome Trust Genome Campus, Hinxton, Cambridgeshire CB10 1SD, UK and [2]Center for Bioinformatics, University of Pennsylvania School of Medicine, Philadelphia, PA 19104, USA

Associate Editor: Alfonso Valencia

ABSTRACT

Motivation: Describing biological sample variables with ontologies is complex due to the cross-domain nature of experiments. Ontologies provide annotation solutions; however, for cross-domain investigations, multiple ontologies are needed to represent the data. These are subject to rapid change, are often not interoperable and present complexities that are a barrier to biological resource users.

Results: We present the Experimental Factor Ontology, designed to meet cross-domain, application focused use cases for gene expression data. We describe our methodology and open source tools used to create the ontology. These include tools for creating ontology mappings, ontology views, detecting ontology changes and using ontologies in interfaces to enhance querying. The application of reference ontologies to data is a key problem, and this work presents guidelines on how community ontologies can be presented in an application ontology in a data-driven way.

Availability: http://www.ebi.ac.uk/efo

Contact: malone@ebi.ac.uk

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on November 24, 2009; revised on February 4, 2010; accepted on March 1, 2010

1 INTRODUCTION

The description of experimental variables, even within a single discipline, involves the use of many cross-domain concepts. For example, describing the characteristics of a single sample in an experiment can use terminology from cell biology, proteomics, transcriptomics, disease, anatomy and environmental science. This is not a new problem and it is not restricted to bioinformatics. However, it is pressing within this domain due to the quantity of heterogeneous data available in different formats across multiple resources (Schofield *et al*., 2009). The desire to integrate data generated with different experimental technologies and in different biological domains motivates our work.

Experimental descriptions are captured and made available as text within database records, published papers and web site content. These descriptions contain latent semantic information that is hard to extract and reflects the natural language of the domain. One solution to this problem is the use of a controlled vocabulary to describe the data. With this approach, the terminology used in a particular context is restricted to a set of terms that define important aspects of a domain or application. Ontology adds an extra layer of expressivity by structuring this vocabulary into ontological classes and by specifying the sorts of operations that can be performed on them. Importantly, the ontological models produced from this process are expressed in a language that enables human understanding and computational reasoning over the representation. Languages such as the W3C recommendation Web Ontology Language (OWL) (Horrocks *et al*., 2003) aid interoperability by standardizing the syntax across all domains. Advantageously, validation of this OWL representation can also be performed through the use of description logic reasoners (Sirin *et al*., 2007).

In bioinformatics, the interest in ontologies to model domain knowledge is apparent from the steadily increasing number of groups developing them. In an attempt to align these efforts, the OBO Foundry (Smith *et al*., 2007) provides useful guidance on 'best practice' for developing ontologies in the biomedical domain. This includes the creation of orthogonal *reference ontologies*, from which classes are considered defining units of the area they describe. Although this is a worthwhile longer term aim, the state of the art is that existing ontologies are not orthogonal or interoperable, and many present a focus that is unsuitable for gene expression data. They can, however, be used to construct application ontologies that focus on describing and structuring a data space for a particular application.

While a vision of full interoperability between ontologies overcomes some of the barriers to integration, there still remain unresolved issues for data-driven applications. Cross products, i.e. classes composed of two or more existing classes (formally in OWL, the intersection of two or more classes), are required between existing ontologies to more accurately describe 'omics' data. For example, a cell type in a given tissue or the transcription factors within a pathway activated in a disease state. Few cross products are available to date partly because many ontologies do not use a common upper level ontology. Where there are non-orthogonal ontologies, those that best describe a dataset of interest typically do not have the necessary cross products. Furthermore, combining even ontologies that are interoperable can present problems. Ontologies such as FMA (Rosse and Mejino, 2003) contain tens of thousands of classes, combined with other ontologies such as Gene Ontology (GO) and Disease Ontology (Osborne *et al*., 2009), and this presents a large model to consider; this is a particular problem if description logic reasoners are used for consistency checking and inference.

The use of multiple ontologies to annotate experimental data brings with it a considerable overhead. Consider an annotation example, where a biological user submitting data needs the term lymphoma. BioPortal (Noy *et al*., 2009) returns 629 matches from 24 ontologies. The casual user is not equipped to select from these

*To whom correspondence should be addressed.

non-orthogonal ontologies and selecting a more specific child term is more problematic; the Disease Ontology alone has 16 subclasses. In other cases, such as genetic disease, many diseases are not present in the Disease Ontology or SNOMED despite their large size. Inconsistent use of synonyms also presents a problem, as synonyms are often for a more or less granular term in another ontology. Another consideration is that ontologies change over time and managing this in the context of annotations is problematic. Mechanisms are therefore required to help manage this.

Representation of biological ontologies is necessarily complex as they have multiple purposes; explicitly modeling biological relationships, aiding interoperability with other ontologies and facilitating data annotation, to name a few. This complexity is a barrier to the consumer of ontology annotated data as they may be unfamiliar with the language, constructs and labels used. Consider the class 'information content entity' from Information Artifact Ontology (IAO) (http://purl.obolibrary.org/obo/iao) defined as 'an entity that is generically dependent on some artifact and stands in relation of aboutness to some entity'. Such a definition may be incomprehensible to a biologist, yet is an important class in Experimental Factor Ontology (EFO). A user-friendly view on upper level ontology is thus required.

In this article, we describe our data annotation and query use cases. We present an application ontology, the EFO, which fulfills the use cases in the context of gene expression data; the methodology and tools that we have developed to produce the ontology are also described and are freely available. We also illustrate the novel cross-product classes that we create using reference ontologies. Our application ontology provides a solution for integrating reference ontologies, extracting information from text, applying annotation and visualization of biological data.

1.1 Motivation: the Gene Expression Atlas

The Gene Expression Atlas (Kapushesky *et al*., 2010) provides summaries of gene expression across multiple experimental conditions, called 'experimental factors'. It also provides a gene level view of experimental data acquired from ArrayExpress (Parkinson *et al*., 2009). This data is manually curated to provide an explicit, consistent and homogenous description across a wide range of sample attributes, such as species, developmental stage, disease and tissue type. Protocol parameters related to the processing of samples, such as application of chemical compounds and sampling times, are also needed. As of November 2009, there are ∼40 000 unique annotations of sample or assay properties covering 330 species in datasets suitable for the Gene Expression Atlas.

Given the diverse nature of the annotations, there is a need to support complex queries that contain semantic information. For example, the query, 'which genes are under-expressed in brain cancer samples in human or mouse', requires the querying mechanism to understand the term 'cancer'. Annotations made at the experimental level are necessarily granular in nature; an experiment where the sample is of adenocarcinoma will be annotated with 'adenocarcinoma' rather than more generally 'cancer'. A database query requiring 'cancer' would therefore not return annotations to adenocarcinoma since this requires additional knowledge. An alternative solution would be to annotate this sample with adenocarcinoma and cancer and any other intermediate classifications such as 'carcinoma'; however, this has a number of disadvantages. First, this requires curation, a labor-intensive process. Second, it embeds the semantics within the database, tightly coupling the data with the domain knowledge. This makes the approach fragile, since a change or extension to domain knowledge may require a large database update. It also limits reuse of the knowledge within other resources.

A better solution to this problem is to annotate data using ontologies. This enables the separation of the formal description of domain knowledge, allowing reuse of these resources and improving interoperability with other data with similar semantic representations. To annotate the diverse data in the Gene Expression Atlas, classes are required from multiple existing ontologies to capture the cross-domain nature of the data.

Initially, we limited scope to data generated to 12 species including: human, mouse, rat, *Arabidopsis*, budding yeast, fission yeast, *Drosophila melanogaster*, *Caenorhabditis elegans* and zebra fish. These species have ontologies that describe anatomy and developmental stages, though the limitations of gene expression technology mean that only a subset of tissues or other variables are typically analyzed. An important use case is the comparability between experiments, for example, where the same tissue, cell type, disease and developmental stage was studied across experiments and species and the data can be potentially combined. Finally, name value pairs that could be mapped to existing domain ontologies were prioritized as these also cover the most common queries e.g. disease state, cell line, cell type developmental stage, etc. Data in the gene expression domain are typically not mapped to an ontology at the point of submission, and neither Gene Expression Omnibus nor ArrayExpress use species-specific ontologies in their submission tools. Requiring use of ontologies at this point is a barrier to data deposition, therefore, the majority of ontology mapping occurs after submission and is based on user-supplied name value pairs e.g. '*DiseaseState* = *breast cancer*'. An important use case is text mining of data prior to its inclusion in ArrayExpress.

Exploratory analyses of the data prior to the construction of EFO revealed that many terms appear at high frequencies and there is a 'long tail' on the data distribution (Malone *et al*., 2009). For example, in the ArrayExpress archive ∼1350 samples have the annotation '*heart*', 65 '*ventricle*', 14 '*myocardium*' and a single annotation for '*pericardium*'. Compare this with the representation of the human heart from the Foundational Model of Anatomy (FMA) (Rosse and Mejino, 2003) where there are >20 terms describing the various parts of the heart. It is clear that comparatively few terms are needed to describe the data in the Gene Expression domain and that the complexity in FMA is not needed. For both text mining and query purposes across free text in the data, there is a requirement for synonyms. This includes 'local synonyms', e.g. 'whole brain', to detect user-defined annotation or to deal with alternate spellings. Our approach for the gene expression domain therefore is analogous to that of the GO (Blake and Harris, 2008), which was initially developed to describe gene products for model organism databases; it has a data-driven motivation, with ontological principles such as use of an upper level ontology applied to provide robustness and to allow interoperability with other ontologies.

2 METHODS

The EFO is an application ontology—an ontology engineered for domain-specific use or application focus and whose scope is specified through testable

use cases and which maps to reference or canonical ontologies. EFO was developed following the 'middle-out' methodology first described in Uschold and Grüninger (1996) and later by (Gómez-Pérez *et al*., 2004). Ontologies, like software, should conform to a set of specifications and use cases, and can be tested using competency questions. Use cases are used to determine the classes we include, and the relations, restrictions and axioms used in our ontology:

(1) *Data annotation*—goal: the primary use case for this application is the annotation of transcriptomics data in the Gene Expression Atlas. Task: this is a *coverage* use case, i.e. can we annotate all of the data we wish to associate ontology classes with?

(2) *Query support*—goal: to enable querying across hierarchies for which data exists (and is annotated). Task: enabling queries such as 'retrieve all cell line data that is derived from epithelial tissue and are associated with cancer'.

(3) *Data visualization and exploration*—goal: to present a tree structure of annotated data within Atlas. Task: presenting an ontology tree to the user to show which classes have associated data.

(4) *Data integration*—goal: to allow integration of data both across experiments in Gene Expression Atlas and externally. Task: integrating with external resources that use or map to the same ontology class and compare data from these independent sources.

(5) *Data summarization and mining*—goal: to obtain an analysis of samples, given common conditions of interest. Task: provide a summary for gene expression data levels for samples treated across same condition, e.g. treated with bacterial toxins.

In addition to use cases, a list of competency questions allows us to evaluate at which point the ontology is able to satisfy the scope of the application (Stevens *et al*., 2000). Examples include 'Which cell lines are derived from epithelial cells?' and 'which organism parts are parts of the forebrain?' As the ontology will be applied in the context of gene expression data, e.g. 'which genes in cancerous vs. normal kidney samples in humans show differential expression?', both an ontological query and a data-driven query in the context of an application are needed. The ontology therefore should represent 'cancer', 'kidney' and 'human' to resolve this query while the differential expression is determined by the application of the ontology in the context of the data, and this competency question therefore demonstrates the application domain.

One approach to ontology development is the use of a modular methodology using a mixture of generic domain, generic task and application ontologies whose parts are clearly defined so that they can be reused (Stevens *et al*., 2000). Our methodology reuses reference ontologies (full list available at http://www.ebi.ac.uk/efo/metadata), where they exist and where they describe classes that are in scope for EFO. We also enrich these classes with additional axioms e.g. making associations between cell lines and their cell types of origin. To promote interoperability with the OBO Foundry ontologies, we have selected BFO as an upper ontology; however, we use only a subset of its classes necessary to fulfill our use cases and we provide user-friendly class labels. An outline of the high-level classes that structure EFO is illustrated in Figure 1. The five primary axes used are as follows: information, site, process, material and material property.

Our ontology development methodology is as follows (complete process documents can be found at www.ebi.ac.uk/efo):

(1) Extract data annotations from the Atlas. Determine the depth and breadth of these annotations and target the most frequently occurring annotations.

(2) Identify OBO Foundry reference ontologies relevant to an EFO category based on annotation use cases.

(3) Use the query use cases obtained from analysis of query logs to build an appropriate hierarchy.

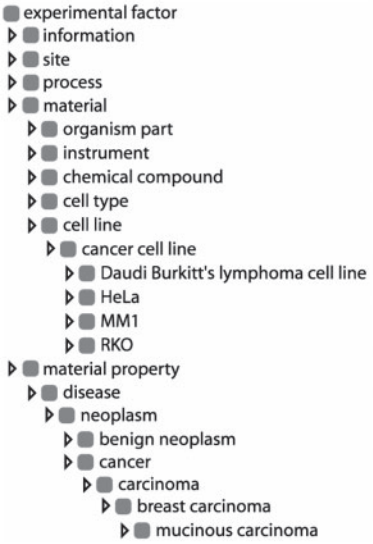

Fig. 1. EFO upper level structure used to organize the ontology with intermediate node examples.

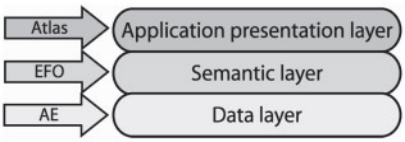

Fig. 2. Separating the ontology layer (EFO) from the data (ArrayExpress) and the presentation layers (Atlas).

(4) Perform mapping between existing annotations and reference ontologies using the Double Metaphone phonetic matching algorithm. This produces a list of candidate ontology class matches.

(5) Expert validation of candidate matches, curate and include matched classes into the EFO hierarchy with appropriate intermediate nodes. Adding classes takes two forms: Where there is no overlap between reference ontologies, import the class directly into EFO [maintaining the original Uniform Resource Identifier (URI)]. Where overlap exists, create a new EFO class (with EFO URI) as a 'mapping class' and add annotation properties with URIs of all mapped classes.

(6) Perform mappings to other reference or application ontologies where these are not provided by the source ontology.

(7) Add structure to EFO to provide an intuitive hierarchy with user-friendly labels and add restrictions to add value e.g. associate cell lines with cell types and tissues of origin.

The strategy of decoupling the data, the presentation layer and the semantic layer is illustrated in Figure 2. Using EFO as a separate layer in our application means we are able to effect changes to the ontology, such as adding new classes or new class relations, without modifying the underlying data or the presentation layer and manage changes in reference ontologies cleanly. A further advantage of this approach is that the ontology can be reused without imposing any special requirements on the implementation or on the application presentation layer, thereby enabling EFO to be used in other applications and expanded accordingly.

Our methodology also aims to observe OBO Foundry best practice guidelines. A set of OWL annotation properties are used to capture metadata

about the classes; we add human readable labels and use univocal and consistent syntax for class names. Metadata details for EFO can be found at http://www.ebi.ac.uk/efo/metadata. We also use the Relation Ontology (RO) (Smith *et al*., 2005). There are relations that are not captured by RO (such as those used in OBI), and therefore, we extend RO where necessary. Our intention is to integrate with future, richer versions of RO when available.

2.1 Detecting external ontology changes

Ontologies that are used within biology evolve rapidly due to scientific advances and because the associated computational technologies are themselves rapidly evolving (Smith *et al*., 2007). Because we consume from multiple ontologies, class information must be maintained and updated. This problem is no more severe than if we mapped data annotations to each ontology separately, rather than to EFO that maps to external reference ontologies.

Here we list the changes in external ontologies which affect EFO:

(1) An axiom is added to an existing named class.

(2) An axiom is removed from an existing named class.

(3) A new named class is added to the ontology.

(4) A named class is made obsolete.

(5) An annotation property is edited on a named class.

The OWL-API (Horridge *et al*., 2009) provides a Java-based interface which allows manipulation of OWL ontologies at the axiom level. Therefore, comparing two different versions of OWL ontologies in an axiom-based approach, as seen in the OWL-API, can be achieved using a set difference operation. In set theory this is given by a relative complement. Formally, for sets A and B the relative complement of A in B, that is, the set of elements in B, but not in A, is given as:

$$B \backslash A = \{x \in B | x \notin A\} \tag{1}$$

Given two sets of axioms, A and B, and axiom a_n:

$$A = \{a1, a2, a3, a4\}, B = \{a1, a2, a3, a5\}$$
$$B \backslash A = \{a1, a2, a3, a5\} \backslash \{a1, a2, a3, a4\}$$
$$= \{a5\}$$

For a set of axioms which are equal:

$$A = \{a1, a2, a3, a4\}, B = \{a1, a2, a3, a4\}$$
$$B \backslash A = \{a1, a2, a3, a4\} \backslash \{a1, a2, a3, a4\} = \emptyset \tag{2}$$
$$\therefore A = B \tag{3}$$

We can use this information to deduce that no changes have occurred between ontologies and moreover to infer that the classes A and B are logically equivalent. We have designed a freely available tool, Bubastis, to analyze and report on the five major types of ontology changes we enumerate. Specifically, we extract the classes mapped to EFO and check for changes. A log of any changes is created along with relevant time and date stamps and a report generated. Usefully, if there are no changes the tool will automatically report this too.

This approach allows us to computationally manage the imports and mappings we create within EFO, ensuring they are valid and reducing the overhead on ontology curation. It also allows us to manage remapping data annotations to EFO which makes the curation process easier. Importantly, this allows us to maintain a consistent use of external resources ensuring that we do not map to obsolete classes, and erroneous mappings caused by external changes are flagged. There is still an outstanding issue of how 'correct' the external resources are. For example, reference ontologies EFO has consumed contain their own mappings which we have further imported to expand interoperability. On scrutiny, some of these were found to be incorrect. For example, mappings to EFO class brain structure derived from synonyms in an external ontology Minimal Anatomical Terminology (MAT) included abnormal brain. Errors of this type are communicated back to the authors of the source ontology. This represents a useful feature of this methodology; we review how reference bio-ontologies map to one another and how correct these mappings are. It is clear that synonyms are used in different ways in different contexts and care must be exercised when using these; we now validate synonyms prior to including these and provide feedback both requesting terms and flagging errors when performing mapping.

2.2 Creating an ontology view

While an upper level framework can provide structure to the ontology, such high-level classes (cf. Fig. 1) can often appear as abstract and confusing for biological users. For example, the Basic Formal Ontology (BFO) (Grenon and Smith, 2004) contains the classes *continuant* and *occurent*. Such classes are useful to organize the ontology and to aid interoperability between ontologies, but are less helpful for a biological user. With this in mind, we use only some of BFO within EFO, and those parts are hidden from users. First, we create an annotation property, *ArrayExpress_label*, which we use to indicate a preferential label that is displayed in the Atlas browser which replaces any other label on the class, though such labels may also be synonyms and are supported for queries. For example, the BFO class *processual entity* is displayed as *process* in the Atlas user interface for readability. Second, we use a further annotation property *organizational_class* which is given a value of 'true' in any classes we wish to hide from the user (e.g. *disposition*) which are identified as structural and which are not desired to be visualized in queries. This allows us to show parts of the ontology relevant to the users, while still using an accepted upper level ontology.

Views generated from EFO are used in both the Atlas and ArrayExpress Archive. EFO is used to improve searching across textual experimental descriptions and key value pairs used to annotate samples. When a user enters a keyword that matches an EFO class, synonyms found in *alternative_term* annotation properties in EFO classes are also used in the search, thereby returning extra matches. We also provide an option to extend searches with classes related to their query via *is_a* or *part_of* ontological relations.

This functionality as deployed in the ArrayExpress Archive is powered by the Apache Lucene as a search engine, and we have packaged the EFO-powered search extension as a separate Java library. The algorithm consists of two parts. First, EFO in OWL format is parsed; the ontology tree is traversed and synonyms, all *part_of* or *is_a* children for all classes in EFO are extracted and a map structure is built for fast lookup. Second, the map structure is used with a rewritten input Lucene Query with additional synonyms and children (if the option is selected and if they exist). For our previous example, query 'breast carcinoma' is transformed to (breast carcinoma OR 'breast cancer' OR 'ductal carcinoma in situ', etc.). This library is available as stand-alone JAR, Maven artifacts and source code from http://github.com/arrayexpress/ae-interface/tree/master/components/efo-query-expand/. The Gene Expression Atlas code base is currently under revision to create a stand-alone install anywhere utility, which will also become publicly available and open source in the near future.

2.3 Supporting the linked data vision

In addition to using EFO within the Gene Expression Atlas and ArrayExpress Archive, we also embrace the ideas of linked data and integration with external resources. In the context of the semantic web, linked data describes a method of creating typed links between data (Bizer *et al*., 2009). In EFO, we use dereferenceable URIs for all of the classes in the ontology which are assigned EFO URIs. Such classes are assigned a unique identifier, e.g. http://www.ebi.ac.uk/efo/EFO_0000001, with the number fragment incremented for each new class. Since each of these identifiers is dereferenceable via the http protocol, they can be requested

from a web server and information about the class returned as user-friendly content. These pages contain information such as the Resource Description Framework Schema (RDFS) class label, parent classes, child classes and annotation properties e.g. text definition. These pages are also machine readable: the source code for each page is actually an EFO Resource Description Framework (RDF) fragment describing a specific class. Computer agents can therefore interact with the EFO class web pages in a way that is analogous to human user interaction. Note, this does not apply to those classes in which URIs are imported from external ontologies; for such ontology URIs, the owner of these ontologies would be responsible for creating dereferenceable URIs.

There are two key elements to linking gene expression data. The first is that parts of the Atlas sample and assay data are annotated with EFO class identifiers. We associate data elements to our explicit definition of what the data represent, by annotating each experiment with EFO classes. The second element is the set of cross-ontology mappings that are maintained within EFO. As EFO is an application ontology, there is an advantage in reusing and importing classes from existing ontologies where possible. Not only does this reduce the effort in adding new classes to EFO, but it also provides interoperability (via cross references) with other resources that use these existing ontologies.

There are a number of challenges associated with this approach. One of the most challenging is deciding upon the appropriate ontology to select when attempting to reuse classes. In the simplest case, where overlap does not exist and there is a clear single authoritative reference ontology, we simply import that class with the original URI maintained, e.g. BFO. However, there are a limited number of examples for where this is the case; for many terms, there may be multiple classes that can fulfill the required definition. For example, consider the term *hypertension*: as of January 2010, there are 12 exact matches for this class label when querying the NCBO BioPortal and most of these ontologies provide definitions consistent with our data annotation use cases. For this reason, we performed some preliminary data-ontology mapping which allowed us to both assess the matching algorithm and the available reference ontologies for coverage on gene expression data (Malone *et al*., 2009). Recently, a tool designed to assist with selecting the most suitable ontologies for a given task has become available, and essentially replicate our early work in an extensible framework (Jonquet *et al*., 2009). As we require EFO to be cross-referenced to as many external functional genomics datasets as possible, we maximize interoperability and therefore add as many mappings to different ontologies as are valid for our given class and curate these. The decision to create multiple external mappings to EFO classes clearly presents additional overhead to both the initial set of mappings and the subsequent maintenance of these mappings, as both are labor intensive if performed manually. We have therefore developed semi-automatic mapping tools.

Our matching approach uses the Metaphone (Phillips, 1990) and Double Metaphone algorithms (Phillips, 2000), which were selected following an empirical study of commonly used matching algorithms and their utility in the biomedical domain (Malone *et al*., 2008).[1] We were particularly interested in algorithms yielding low false positive rates, as we wished to use the same algorithm for semiautomatic annotation of incoming data to the ArrayExpress Archive as a curator aid. Following the evaluation of several algorithms, a combined strategy was implemented using Metaphone for a first pass and then falling back to Double Metaphone for those terms not matched by Metaphone. This strategy yields the highest overall number of matches with minimal human intervention (required only for multiple matches). Verified matched terms identified by this strategy were included as valid mappings in EFO and added to a *definition_citation* annotation property. We have developed a species-specific ranked list of preferred ontologies with known good coverage when mapping to new terms. We prefer to use OBO Foundry candidate ontologies when these provide good matches and use general uncurated resources like Unified Medical Language System (UMLS) only when necessary.

[1]Tools available at http://www.ebi.ac.uk/efo/tools

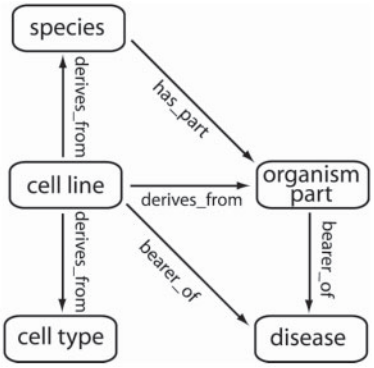

Fig. 3. Added value relations between classes in EFO. The figure illustrates the existential restrictions (i.e. one or more relationship) placed on some of the subclasses of the classes shown (classes shown in boxes).

3 RESULTS

The diversity of experiments captured in the Gene Expression Atlas and ArrayExpress provides a wide range of experimental variables. A typical experiment includes factors such as disease, anatomical parts, developmental stage, species and chemical compounds. Within these experimental factors, there is additional knowledge that we capture to support our use cases. Consider the query, 'retrieve all data for cancer cell line samples'. This query requires more than just samples in the database which have been annotated with cancer and with a cell line. The query is more accurately expressed as cell lines that are derived from some diseased sample. We therefore add logical relations between classes in the context of EFO; these can serve as OBO Foundry integration use cases. An example of some of the existential restrictions between classes is shown in Figure 3.

The ability to explicitly express richer statements of knowledge (such as the example above) is one of the major advantages of using ontologies; however, with increased complexity comes increased possibility of contradiction and inconsistent expression. To help manage this issue, we chose to use the Web Ontology Language (Horrocks *et al*., 2003), the recommendation for representing knowledge with formally defined meaning. Using the OWL-DL flavor of the language, we are able to create axiomatic statements about classes, and use the Pellet 1.5.2 description logic reasoner (Sirin *et al*., 2007) to ensure the ontology is consistent, i.e. class membership is axiomatically correct and there are no contradictions in the model. OWL also offers the ability to create 'equivalent classes' (often called defined classes) which are useful for inferring hierarchies and managing multiple inheritance. Considering the previous example of the EFO class 'cancer cell line', this is defined in OWL as any class which has the 'bearer_of some cancer' axiom. In other words, any cell line which bears the disease cancer will be inferred to be a subclass of this type. Here we illustrate some of these examples from our application.

3.1 Querying gene expression data

In order to demonstrate that EFO is fit for purpose, we evaluate it against the competency questions and use cases. One of the primary

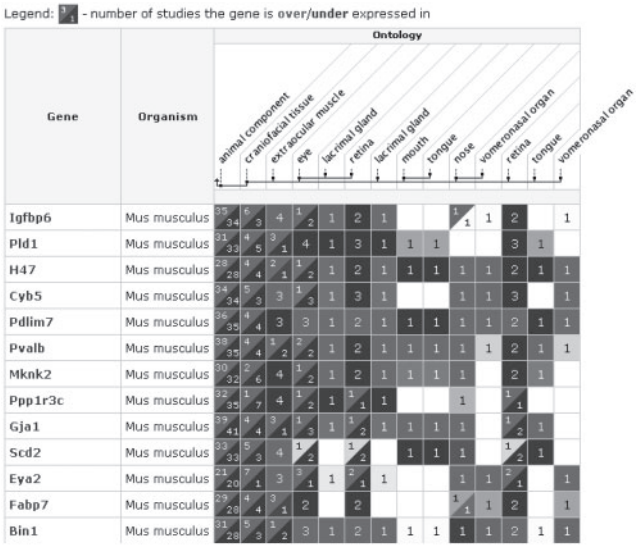

Fig. 4. Gene Expression Atlas Query for genes under- or overexpressed in mammalian 'craniofacial tissues'.

use cases for EFO was to annotate ArrayExpress data (i.e. providing ontological coverage) and to ask meaningful questions of gene expression data.

In a recent review conducted by Jonquet *et al*. (2009), EFO was assessed for coverage in annotating biological datasets using an NCBO tool, the Open Biomedical Annotator. Alongside 98 English ontologies in UMLS 2008AA and 92 of the BioPortal ontologies, in total, these resources offer a dictionary of 3 582 434 classes and 7 024 618 textual terms. Following experimentation with three biological datasets, EFO is reported as fourth best in all three tests. EFO performs well in these tests due to the data-driven development, cross-domain method we use. It is also noticeable that the ontologies that finished in the top three were significantly larger than EFO, for example, NCI Thesaurus has ∼35 000 classes compared with ∼2600 classes in EFO.

The real return for the user when using an ontology is in the additional relations used for improving queries. Due to the relations used in EFO, we are able to ask general questions without requiring that every subtype is enumerated in the query. For example, for experiments about cancer, we want all subtypes of cancer, for example prostate carcinoma, without requiring the user to specifically enumerate these subtypes and we want to return only subtypes for which we have data. Similarly, we want a user to be able to ask for 'forebrain' and the query to return data that is annotated with forebrain substructures such as hypothalamus. Finally, for mouse we want to return data annotated to *Mus musculus* and substrains thereof.

Figure 4 shows the results of a query for 'gene that is expressed in craniofacial tissues or sub structures'. Within the ontology, relations are made between classes such as those seen in Figure 4. Specifically here, the query is asking for *genes which are over- or underexpressed in assays that are annotated with an organism part that is craniofacial tissue or a sub-structure*. Figure 4 presents the parts of the ontology that satisfy this query at the top of the image. The tree includes classes such as *eye* (synonym *eye structure*), which expands to include its substructures such as *retina*.

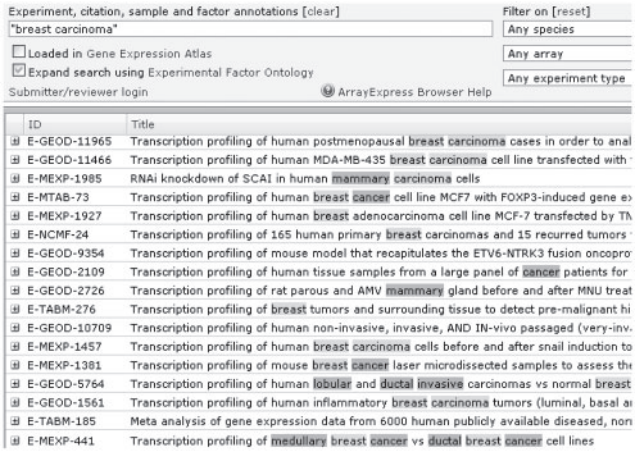

Fig. 5. Ontology-enabled search using EFO, showing query expansion for keyword 'cancer' with breast carcinoma selected. Subtypes (red), synonyms (green) and matches to the search term (yellow) shown in the ArrayExpress Archive.

As described earlier, EFO has also been used in the ArrayExpress Archive (http://www.ebi.ac.uk/arrayexpress) to enrich querying. Figure 5 illustrates that in addition to keyword 'breast carcinoma' (yellow), EFO-enabled search returns experiments matching is-a children, e.g. 'medullary breast cancer'.

3.2 Linking data through BioPortal

An additional advantage to using an ontology to annotate data in the Atlas is in the use of external ontology tools. The BioPortal resource at NCBO is an open repository of biomedical ontologies that provides access via web services and web browsers to ontologies developed in OWL, RDF and OBO format (Noy *et al*., 2009). It allows the searching of biomedical data resources such as ArrayExpress, through the annotation and indexing of these resources with ontologies that can be accessed through BioPortal.

Since EFO is used to annotate data in ArrayExpress and also provides multiple mappings to other ontologies, it is possible to query data through BioPortal using ontology class names and return annotated data from multiple resources via the BioPortal's Resources facility, for example, pathway data from Reactome.

4 DISCUSSION

In this article, we present EFO, an application ontology driven by the annotation and query needs of samples in 'omics' datasets. Our approach to ontology engineering uses the many existing reference bio-ontologies while allowing us to develop a hierarchy that supports our use cases. EFO enables queries of the data that were not previously possible because we add value to existing ontologies by adding explicit relations and because we have adopted a data-driven methodology. Furthermore, EFO separates knowledge from the experimental data, is reusable and easy to maintain; when modification to the knowledge is required, modification to the data is not. We believe the methodology and tools present a reproducible and maintainable strategy to create ontological solutions for a particular application focus. EFO has also proven to be useful for text mining annotation of gene expression datasets and has been used

in data mining. An EFO-R package that facilitates such analysis is currently under development.

Essentially EFO represents a custom view of several domain-specific ontologies. We believe that use of 'ontology views' will help end users to understand and use ontologies. An advantage of EFO for ArrayExpress staff is that they do not need specialist domain knowledge of multiple ontologies and are able to apply EFO consistently to data, while users typically do not perform well as annotators. Ideally, views should contain a subset of the ontology that is still logically consistent containing only classes, instances and properties that are desirable. The requirements of a view are likely to be driven by particular applications and user communities as described here. Improved tools that support the creation and use of views will help the users of bioinformatic resources overcome one of the largest obstacles of using ontologies: that the learning curve is extremely steep and the climb is a disincentive to users.

There is a great deal of useful work presently under way within the bio-ontology community. However, it is impractical and undesirable to import, wholesale, ontologies that touch upon many domains and expect users to apply them consistently. Guidelines on development of application ontologies and appropriate reuse of existing resources would be useful. In particular, maintenance and mapping of original ontology identifiers and development of public domain tools are important. Similarly, there are several important challenges facing reference ontologies. One of the most challenging is mapping anatomy between multiple species. This is not in scope for EFO and we look forward to consuming such reference ontologies, but application data should inform some of this work.

Our work with ontologies is focused on enabling us to do novel research with the experimental data we have, such as answer more complex questions and integrate multiple data sources. In this respect, ontologies are a means to an end; our work here is based on describing experimental data, and we believe this should be the driving force behind ontology development and consumption.

Future work will develop an RDF triple store representation of Atlas and provide federated querying using SPARQL end points. An ontology-enabled annotation application for functional genomics data—Annotare (*code.google.com/p/annotare/*) is being collaboratively developed, which allows users and curators to select terms from multiple ontologies, including EFO. We hope this will expose users to ontologies in a user-friendly way and help provide better annotated datasets.

ACKNOWLEDGEMENTS

We thank Eric Neumann, Mélanie Courtot, Frank Gibson, Alan Rector, the Gen2Phen and Engage consortia and P3G colleagues for sharing use cases and data annotations, and the ArrayExpress and Gene Expression Atlas team for their implementation of EFO in the Atlas UI and the 3 anonymous reviewers for their comments.

Funding: European Commision grants FELICS (contract number 021902); EMERALD (project number LSHG-CT-2006-037686); Gen2Phen (contract number 200754); European Molecular Biology Laboratory.

Conflict of Interest: none declared.

REFERENCES

Blake,J.A. and Harris,M.A. (2008) The Gene Ontology (GO) project: structured vocabularies for molecular biology and their application to genome and expression analysis. *Curr. Protoc. Bioinformatics*, Chapter 7, Unit 7.2.

Bizer,C. *et al*. (2009) Linked data - the story so far. *Int. J. Semant. Web Inf. Syst.*, **5**, 1–22.

Grenon,P. and Smith,B. (2004) SNAP and SPAN: towards dynamic spatial ontology. *Spat. Cogn. Comput.Interdiscip. J.*, **4**, 69–104.

Gómez-Pérez,A. *et al*. (2004) Ontological engineering. Springer-Verlag, New York.

Horridge,M. (2009) The OWL API: A Java API for Working with OWL 2 Ontologies. In *Proceedings of the OWL: Experiences and Directions 2009, Chantilly, USA, CEUR Workshop Proceedings*.

Horrocks,I. *et al*. (2003) From SHIQ and RDF to OWL: the making of a web ontology language. *J. Web Sem.*, **1**, 7–26.

Jonquet,C. *et al*. (2009) Prototyping a Biomedical Ontology Recommender Service. In *Proceedings of the 12th Annual Bio-Ontologies: Knowledge in Biology, SIG, ISMBECCB 2009*, Stockholm, Sweden, pp. 65–68.

Kapushesky,M. *et al*. (2010) Gene Expression Atlas at the European Bioinformatics Institute. *Nucleic Acids Res.*, **38**, D690–D698.

Malone,J. *et al*. (2008) Developing an application focused experimental factor ontology: embracing the OBO community. In *Proceedings of ISMB 2008 SIG meeting on Bio-ontologies.* Toronto, Canada.

Noy,N.F. *et al*. (2009) BioPortal: ontologies and integrated data resources at the click of a mouse. *Nucleic Acids Res*., **37**, W170–W173.

Osborne,J.D. *et al*. (2009) Annotating the human genome with Disease Ontology. *BMC Genomics*, **10**, S1–S6.

Parkinson,H. *et al*. (2009) ArrayExpress update–from an archive of functional genomics experiments to the atlas of gene expression. *Nucleic Acids Res*., **37**, D868–D872.

Phillips,L. (1990) Hanging on the Metaphone. *Comput. Lang.*, **7**, 39–43.

Phillips,L. (2000) The Double Metaphone Search Algorithm. *C/C++ User's Journal*, **18**, 38–43.

Rosse,C. and Mejino,J.V.L. (2003) A reference ontology for biomedical informatics: the Foundational Model of Anatomy. *J. Biomed. Inform.*, **36**, 478–500.

Schofield,P.N. *et al*. (2009) Post-publication sharing of data and tools. *Nature*, **461**, 171–173.

Sirin,E. *et al*. (2007) Pellet: a practical OWL-DL reasoner. *J. Web Sem.*, **5**, 51–53.

Smith,B. *et al*. (2005) Relations in biomedical ontologies. *Genome Biol*., **6**, R46.

Smith,B. *et al*. (2007) The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nat. Biotechnol.*, **25**, 1251–1255.

Stevens,R. *et al*. (2000) Ontology-based knowledge representation for bioinformatics. *Brief. Bioinformatics*, **1**, 398–414.

Uschold,M. and Grüninger,M. (1996) Ontologies: principles, methods and applications. *Knowl. Eng. Rev.*, **11**, 93–115.