

Community-based mappings in BioPortal

Natasha Noy

1 Ontologies and mappings between them

BioPortal is an open library of biomedical ontologies. Users contribute ontologies to BioPortal and they come to BioPortal when they need to find a biomedical ontology to use in their application. In BioPortal, users can search and browse the ontologies, find resources annotated with concepts from these ontologies, and download ontologies for their use.

The content in BioPortal ontologies overlaps. For instance, there are several ontologies that deal with some aspects of human anatomy, such as the Foundational Model of Anatomy (FMA) [5], the NCI Thesaurus [6], and Galen [4]. Understanding how the concepts in different ontologies relate to one another is one of the key requirements of BioPortal users. We refer to the relations between concepts in different ontologies as **concept mappings**, or simply **mappings**. For instance, we can create a mapping between the class `Body_Tissue` in the NCI Thesaurus and the class `Body tissue` in the FMA. A collection of all mappings from one ontology O_1 to another ontology O_2 is a **mapping between O_1 and O_2** .

1.1 Mappings in BioPortal

We plan to use BioPortal not only as a repository of ontologies and their respective metadata, but also as a repository of mappings between ontologies. We expect that users will both upload bulk mappings that they create using some dedicated ontology-mapping tool like PROMPT, and create single one-to-one mappings as a by-product of exploring ontologies in BioPortal. Furthermore, we expect that some mappings will contradict one another; for any source concept in one ontology, users may suggest different concepts in another ontology as the target concept. One of our goals is to provide visualization and analytical tools to help users understand the different mappings

and to resolve contradictions and inconsistencies in mappings in BioPortal.

Community members can not only contribute new mappings to BioPortal, but also discuss mappings that already exist in the BioPortal, just as they can discuss ontologies as a whole or ontology components. In many cases, reaching consensus on mappings between ontologies can be as difficult, or sometimes, as nearly impossible as reaching consensus on ontology content itself.

1.2 What relationship does a mapping represent

It is customary to think about mappings as *equivalence* mapping, and many researchers suggested using equivalence in the logical sense, essentially `owl:equivalentClass`. In most cases of inter-ontology mapping, however, the mapping is not a true logical equivalence. The latter would have implied that the two concepts share their instances, for example. More often, when we create a mapping between concepts in two different ontologies, the relationship that we are representing is that of similarity, rather than strict equivalence.

For many reasoning and querying tasks, we can treat similarity in the same way as equivalence. For instance, when we look for data annotated with a concept C_s , we may also bring in the data annotated with a concept that C_s is mapped to, C_t . However, it is important to note that the relationship here is not true logical equivalence.

1.3 Mapping As A Bridge vs Mapping As A Glue

There are two—not necessarily conflicting, but not identical—views on what a mapping between two ontologies is. In one case, we can think about a mapping between two ontologies as a *bridge*: each ontology stands on its own, and will continue to do so, but the mapping indicates the point of overlap. In this case, each ontology is an independent unit, intended to be used without the ontology it is mapped to. In another setting, the mapping serves as a *glue* that brings the two ontologies together to create a single whole, with clearly identifiable components. In this case, the ontologies that are mapped are intended to be used together, as a single unit. For example, when we create a mapping between the anatomy part of the NCI Thesaurus and the FMA, our goal is not to merge the two ontologies, but rather to help applications integrate the data that is annotated with terms from either ontology. We expect, however, that many applications will use only one or the other ontology. For the example of the second case, consider,

for instance, the following mapping (from C. Mungall [2]):

```
ZFA:heart is_a CARO:cavitated_compound_organ
```

There is no intention in the zebrafish anatomy ontology (ZFA) to define organs at the general level, as CARO does. Thus, we use the mapping to make the definition of ZFA:heart to be more precise, in essence, joining the ZFA and the CARO ontologies.

The line between the two settings can be fuzzy, and sometimes it is discernible inly through the intention of those who created a mapping.

Pragmatically, with mappings of the first kind (more of a bridge), similarity, or generalization/specialization are the more common mapping relationships. In the second case, any mappings are possible: for instance, a class in one ontology could be a range for a property for another. This last type of mapping is hardly present in the bridge setting (e.g. CL:nucleate_erythrocyte has_part GO:nucleus [2])

2 Sources Of Mappings In BioPortal

The mappings that we store and visualize in BioPortal can come originally from several different types of sources or can be generated by several methods. We distinguish the sources based on the human involvement in creating the mappings and, as a consequences, whether the mapping involves some element of human judgement or whether it is purely “mechanical” and, therefore, can be easily re-generated again for the same ontologies. With an eye towards storing the mappings in BioPortal, we highlight which meta-information about the mapping-generation method we must store along with the mapping itself. Note that when we talk about mappings between two ontologies (rather than two concepts) here, we propagate the same meta-informaion about the mapping-generation method to each pair of correspondences produced by the method.

1. **Automatic mapping algorithm, with no tunable parameters:**

These algorithms take as input two ontologies and produce a mapping between them (pairs of correspondences). For these algorithms, users cannot tune specific parameters. Therefore, given the same two ontologies, the algorithm produces the same results on repeated executions (e.g., simple ontology mapping as done by QOM [1]). The date on which the algorithm was run may still be important because the results may differ even for these types of algorithms. First, the algorithm, may evolve over time, thus the same set of inputs could result

in a different result. Second, the external sources that the algorithm uses can change. For instance, an algorithm relying on Swoogle, or Wikipedia, or UMLS can produce different results over time.

2. **Automatic algorithms with tunable parameters:** These algorithms take two ontologies and a set of parameters as input and produce a set of mappings. These parameters can include specific configurations of the algorithm, such as edit-distance metric used, ways subtasks are composed, threshold values, weights for different components, and so on. Some algorithms rely on a set of initial mappings; a different set of initial mappings will lead to a different result. The results of these algorithms can change over time for the same reasons that the results for non-tunable algorithms. But the results can also change when the values for the tunable parameters change.
3. **Interactive algorithms and tools** These algorithms combine some elements of automatic matching with interactive mapping by the user (e.g., PROMPT [3]). Therefore, the result depends on specific steps taken by the user during the interactive process. Thus, we cannot reproduce the mapping result if we need to re-create the mapping.
4. **Manual mapping** These mappings are created when a user looks at a pair of concepts individually, considering each one of them separately. This process could happen in a specialized interface or as a side-effect of browsing or editing an ontology. For instance, mappings to standard terminologies that are common as values for annotation properties in biomedical ontologies (e.g., the corresponding UMLS identifier) are manual mappings that ontology authors create when they define new ontology concepts.

Where do the mappings come from: Open issues

- How do we bootstrap the mappings? Should we run some well-known algorithms to populate the mappings? Perhaps even string comparison or something simple like that?

3 Bootstrapping Mappings for BioPortal

In the alpha version of the BioPortal (as of March, 2008) we have the following sources for mappings:

- For ontologies represented in UMLS (GO, ICD9, FMA, NCIT), we create correspondences for classes corresponding to the same CUI.
- For NCI and Galen, we used the part of PROMPT that performs simple string matching on class names.
- For ontologies containing representation of anatomy (FMA, adult mouse anatomy, fly anatomy, zebrafish anatomy), we used simple string matching of class names or synonyms to UMLS terms (from Nigam Shah).
- Simple stemming+synonym string comparison algorithm by Chris Mungall (zebrafish anatomy, FMA, fly anatomy)

4 Representing mappings and mapping metadata in BioPortal

For the moment, we consider only **one-to-one mappings** in BioPortal: One-to-one mapping is a mapping between two concepts from different ontologies, a source and a target. We plan to extend this representation to more complex mappings (such as one-to-n mappings) in the future.

We represent mappings in BioPortal as instances in the **mapping ontology** (Figure 1).¹ Each instance corresponds to a single mapping between concepts (not ontologies). Each mapping instance points to the two concepts being mapped (the *source* concept and the *target* concept), and to the metadata about this mapping, such as how the mapping was created and when, the type of the mapping, additional comments, and so on.²

Thus, BioPortal has a single knowledge base that contains all mappings between all ontologies in BioPortal.

Mappings are stored independently of the ontologies themselves. All mappings are directional: they connect source to target. Thus, for symmetric mappings (such as similarity), there are two instances, each corresponding to a different direction of the mapping. We can decide to add the symmetric mapping by default: always create the arrow in the opposite direction.

In addition to the source and target concepts of the mappings, each mapping can contain a set of additional metadata that describe the additional

¹This ontology is an extension of the simple mapping ontology used for the Ontology Alignment Evaluation Initiative (OAEI): <http://oaei.ontologymatching.org/2007/align.html>.

²We address the issue of ontology versioning in Section 5.

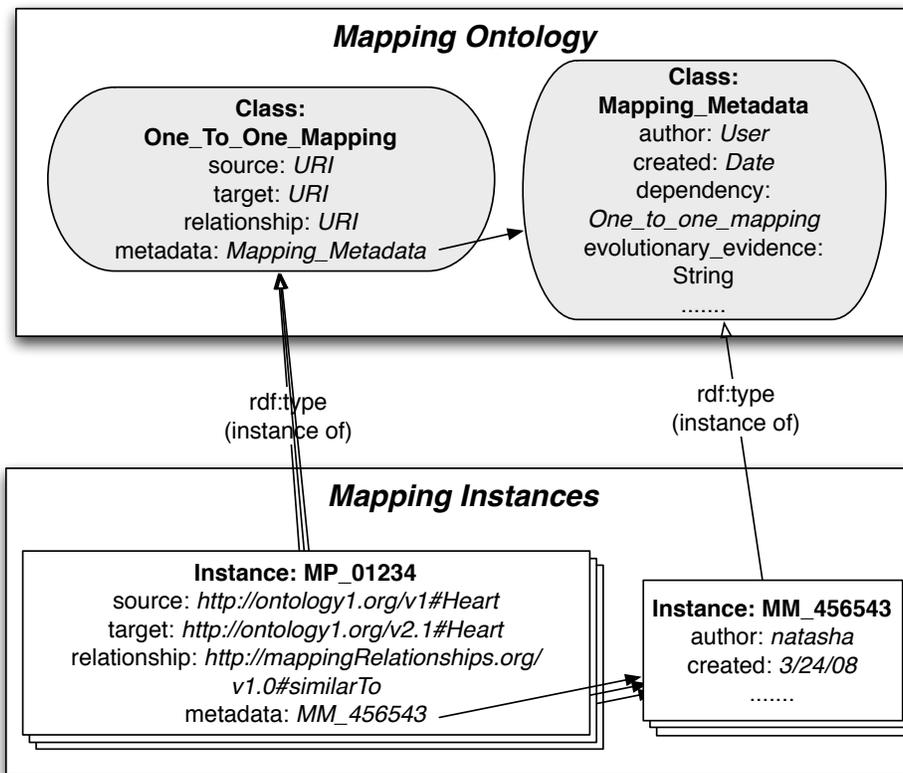


Figure 1: Mapping ontology and its instances. Each mapping is an instance of the class `One_to_One_Mapping`, which refers to the source and target concepts of the mapping, and to the metadata associated with the mapping.

properties of the mapping and the information about how the mapping was created (cf. Section 2). First, we list the information about the mapping itself, regardless of how and when it was created:

- **Mapping relationship:** Not all mappings are equivalence or similarity mappings. There can be other relationships, such as specialization/generalization, or, in fact, any other relation that an ontology language, such as OWL, supports. For the moment, this property is just a string, indicating the mapping relationship; it can also be a URI of a relationship from another ontology.
- **General comment:** General comment about the mapping, usually

added by the person who created the mapping.

- **Discussion thread:** For community-based mappings—there can be a discussion thread associated with a mapping; mappings are first-class objects that others can comment on and discuss.
- **Application context:** Some mappings may apply in certain situations and not others. That is, in some cases, the context of an application for which the mapping was performed could be an important piece of meta-information.
- **External references:** If the mapping is based on some references to external sources (e.g., publications), this information can be part of the meta-data.
- **Mapping dependency:** One mapping can depend on another: “*If X is Y, then A is B*”

We envision including domain-specific information as a separate meta-property on the mapping, such as homology mapping:

- **Evolutionary evidence for homology mappings:** From C. Mungall [2]:

The focus will be on homology mappings rather than mappings based on structural or functional analogy. Thus all mappings will **require** evolutionary evidence.

Examples

1. human heart and mouse heart
2. human arm and bat wing *as forelimb*

Counter-examples

1. bat wing **not** *homologous_to* chicken wing *as wing*

In addition, there is information about the source of the mapping, parameters of the algorithm the user who created it, and so on:

- **Mapping algorithm:** the name of the algorithm that was used to create the mappings.
- **Version or date for the mapping algorithm:** as algorithms evolve, they can produce different results on the same input as new features are added to the algorithm.

- **The date the mapping was created:** the date the mappings was created.
- **The parameters for the algorithm:** the set of parameters and/or inputs that was used in tuning the algorithm
- **The user who performed the mapping:** the name of the user whose input affected the mapping outcome is important for the interactive and manual algorithms

5 Mappings and ontology versioning

Ontologies change overtime and users submit new versions of their ontologies to BioPortal. Thus, we must address the issue of maintaining ontology mappings and the ontologies themselves evolve.

We can envision two “extreme” approaches to this maintenance problems. On the one extreme, any time an ontology author submits a new version to the BioPortal, we discard all the mappings that were associated with the old version. This approach is clearly not practical as most of the mappings are still valid for the concepts in the new version. At the other extreme, we can associate mappings with a name of a concept, rather than with a concept in a specific version. This solution will also create problems, because occasionally some of mappings will no longer be valid in a new version and a “wholesale” migration of mappings is not necessarily a practical approach either.

We propose a middle-ground approach: each mapping is associated with a concept in the specific ontology version that was considered when the mapping was created. However, when we access the mappings for a concept in the latest version, we retrieve the mappings for that concept for all the previous version as well. The user gets the context for the mappings and knows whether the mapping was created for the current version of the ontology or for some earlier one (and if it is the latter, which earlier version). While in the current prototype users cannot invalidate mappings, we envision allowing users to delete mappings that are no longer valid.

Figure 2 provides a sketch of maintaining mappings through different ontology versions. Each ontology has a virtual URI in BioPortal that always resolves to the latest version of that ontology. Each version also has its own, version-specific URI. Thus, the current version in BioPortal can be addressed by any of the two URIs: the permanent version-specific URI and the virtual

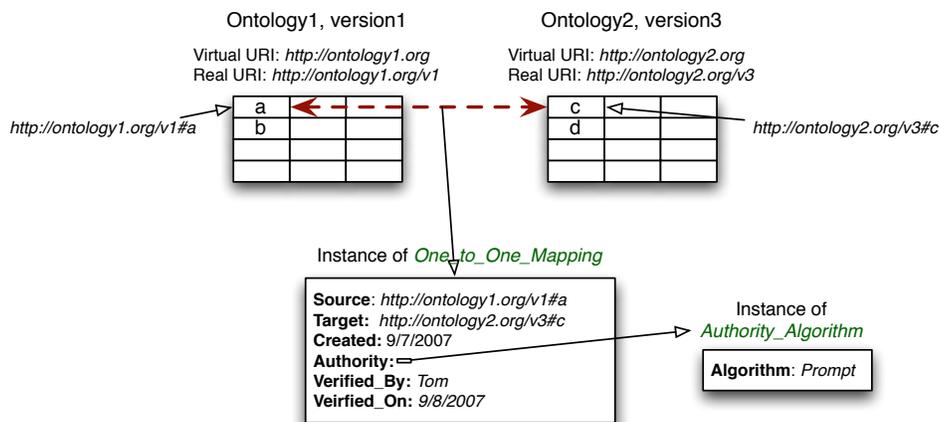


Figure 2: Maintaining mappings through ontology versioning. Each mapping refers to version-specific URI of a concept as the source and the target of mapping. When we need to find all mappings for a concept in the current version of an ontology, we create version-specific URIs for this concept in for all the previous versions, and retrieve the mappings with these URIs as the source.

URI.³

Each mapping uses a version-specific URI for a concept. When we need to retrieve all mappings for a concept C in the current version of the ontology O , we first get all the version-specific URIs for O . We then attach $\#C$ (or whatever the URI scheme that we use for specific concepts) to each of those version-specific URIs to get a set of version-specific URIs for C . We then query for all mappings with this version-specific URIs for C as the mapping source.

When presenting mappings in the user interface or returning them as data dump to the user, we have direct access to the information on the specific version of the ontologies for which the mapping was created: this information is part of the version-specific URIs for the source and target of the mapping.

Another option—for the mappings that were produced by automatic algorithms—is to re-run the algorithms with the same parameters. This approach may not be practical in the BioPortal setting but may work in other settings.

³This scheme is used, for example, by W3C Documents.

6 Aggregating mappings

In a community-based settings, some mappings will reinforce each other and other mappings would contradict each other. Consider a concept C_s in an ontology O . We can envision several mapping with C_s as the mapping source:

- Several mappings to the same concept C_t from another ontology O' coming from different sources (e.g., several algorithms producing this mapping). In this case, we can suggest that the mappings reinforce each other.
- One mapping to a class C_t and another mapping to its subclass or superclass. The mappings are not contradictory but one of them appears to be more precise than another.
- One mappings to a class C_t and another to a class that is—implicitly or explicitly—disjoint with it. These two mappings are contradictory.
- It is possible that an algorithm—or, more realistically, a user—would produce a “not” mapping: “ C_s is definitely not similar to C_t ” (the negative mapping can be one of the relationship types for mappings). This mapping may contradict a positive mapping for the same pair of concepts.

For the moment we do not provide any analysis of aggregate mappings of this sort. We simply present multiple mappings to users for their analysis.

7 User experience with mappings in BioPortal

The users will be able to access and to use mappings in the following ways:

- when browsing a class, see all other classes this class is mapped to and the metadata on those mappings (Figure ??);
- selecting an ontology and having an overview of the mappings with concepts from that ontology as source (e.g., as a tag cloud where the size of the concept is determined by the number of mappings that term has);
- selecting two ontologies and visualizing all mappings between them, Jambalaya-style;

- on myOntology page, see the RSS feed of new mappings

Thus, BioPortal must support the following mapping-related queries:

- Given an ontology (by its virtual URI) and a concept name (in other words, a virtual URI for a concept), return all the mappings with that concept as a source
- Given an ontology and a specific version and a concept name (in other words, a real URI for a concept), return all the mappings with that concept as a source

Other BioPortal services for mappings:

- filter a set of mappings (e.g., between two specific ontologies, created in a specific range, of specific type) and download these mappings as a set of RDF instances (e.g., “give all the user-generated mappings between NCIT and FMA created in the last three months”, or “give me all UMLS mappings between GO and NCIT”).
- upload bulk mappings as a set of RDF instances (if the users use the non-versioned class names in those mappings, we attach them to the latest version of the ontology)

References

- [1] Marc Ehrig and Steffen Staab. QOM - Quick Ontology Mapping. In *3rd International Semantic Web Conference (ISWC2004)*, Hiroshima, Japan, 2004.
- [2] C. Mungall. Mappings in OBO Foundry, 2008.
- [3] N. F. Noy and M. A. Musen. The PROMPT suite: Interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, 59(6):983–1024, 2003.
- [4] A.L. Rector, J.E. Rogers, and P. Pole. The galen high level ontology. In *Fourteenth International Congress of the European Federation for Medical Informatics, MIE-96*, Copenhagen, Denmark, 1996.
- [5] C. Rosse and J. L. V. Mejino. A reference ontology for bioinformatics: The foundational model of anatomy. *Journal of Biomedical Informatics.*, 2004.

- [6] N. Sioutos, S. de Coronado, M.W. Haber, F.W. Hartel, W.L. Shaiu, and L.W. Wright. NCI Thesaurus: A semantic model integrating cancer-related clinical and molecular information. *Journal of Biomedical Informatics*, 40(1):30–43, 2007.