Towards annotating potential incoherences in BioPortal mappings

Daniel Faria[1], Ernesto Jiménez-Ruiz[2], Catia Pesquita[1,3],
Emanuel Santos[3], and Francisco M. Couto[1,3]

[1] LASIGE, Faculdade de Ciências, Universidade de Lisboa, Portugal
[2] Department of Computer Science, University of Oxford, UK
[3] Departamento de Informática, Faculdade de Ciências, Universidade de Lisboa, Portugal

Abstract. BioPortal is a repository for biomedical ontologies that also includes mappings between them from various sources. Considered as a whole, these mappings may cause logical errors, due to incompatibilities between the ontologies or even erroneous mappings.
We have performed an automatic evaluation of BioPortal mappings between 19 ontology pairs using the mapping repair systems of LogMap and AgreementMakerLight. We found logical errors in 11 of these pairs, which on average involved 22% of the mappings between each pair. Furthermore, we conducted a manual evaluation of the repair results to identify the actual sources of error, verifying that erroneous mappings were behind over 60% of the repairs.
Given the results of our analysis, we believe that annotating BioPortal mappings with information about their logical conflicts with other mappings would improve their usability for semantic web applications and facilitate the identification of erroneous mappings. In future work, we aim to collaborate with BioPortal developers in extending BioPortal with these annotations.

1 Motivation

OWL ontologies are extensively used in biomedical information systems. Prominent examples of biomedical ontologies are the Gene Ontology [1], the National Cancer Institute Thesaurus (NCIT) [14] and the Foundational Model of Anatomy (FMA) [32].

Despite some community efforts to ensure a coordinated development of biomedical ontologies [38], many ontologies are being developed independently by different groups of experts and, as a result, they often cover the same or related subjects, but follow different modeling principles and use different entity naming schemes. Thus, to integrate data among applications, it is crucial to establish correspondences (called mappings) between the entities of the ontologies they use.

In the last ten years, the semantic web and bioinformatics research communities have extensively investigated the problem of (semi-)automatically computing correspondences between independently developed ontologies, which is usually referred to as the *ontology matching problem*. Resulting from this effort are the growing number of ontology matching systems in development [8,7,37] and the large mapping repositories that have been created (e.g., [2,10]).

One such repository, BioPortal [10,33], is a coordinated community effort which currently provides access to more than 370 biomedical ontologies and over 12 million mappings between them.[4] While not all BioPortal ontologies were originally OWL ontologies (e.g., some were developed in OBO format[5]), many have been (or can be) converted to OWL [15]. Mappings in BioPortal are either generated automatically by a sophisticated lexical matcher [13] or added manually by domain experts through the Web interface or the REST APIs [29].

OWL ontologies have well-defined semantics [4] and thus the integration of independently developed ontologies via a set of mappings (i.e., an alignment) may lead to logical errors such as unsatisfiablities [25]. BioPortal, however, explicitly supports the idea that alternative (i.e., created for different purposes) mapping sets may co-exist and that they could potentially contradict each other [29].

While it is true that many logical errors in alignments are caused by incompatibilities between the ontologies they map [19,31], some may be caused by erroneous mappings. Furthermore, logical soundness may be critical to some semantic web applications that integrate two or more ontologies [31]. For these reasons, we consider that it would be advantageous to enrich BioPortal mappings with *annotations* about potential logical conflicts with other mappings. This would improve the usability of BioPortal mappings for semantic web applications and domain users, and facilitate the identification of erroneous mappings and potential errors in the ontologies themselves.

In this paper we quantify the logical errors in the BioPortal mappings among several ontologies by applying mapping repair algorithms. Furthermore, we manually analyze a subset of the identified conflicting mappings in order to qualify the causes of the errors. Our goal is to show the importance of identifying (and annotating) logical conflicts in BioPortal and the role ontology mapping repair algorithms may play in that task.

The rest of the paper is organized as follows: Section 2 describes how mappings are represented in BioPortal; Section 3 introduces the concept of mapping repair and presents the repair algorithms used in this study; Section 4 details the automatic and manual evaluations we conducted and presents and discusses their results; and finally, Section 5 presents some conclusions and future work lines.

2 Mappings in BioPortal

Mappings are treated as first-class citizens in BioPortal [29,12], as it enables the querying, upload, and retrieval of all mappings between all of its ontologies. A survey conducted in 2009 revealed that 33% of BioPortal ontologies had 50% of their entities mapped to other ontologies [12], which indicates that BioPortal ontologies are highly interconnected.

The number of mappings in BioPortal has grown quickly in recent years, from 30,000 mappings between 20 ontologies in 2008 [29] to 9 million mappings between 302 ontologies in 2012 [33]. At the time of writing this paper, there were approximately 13 million mappings between 373 ontologies.

[4] BioPortal: `https://bioportal.bioontology.org/`

[5] `http://www.geneontology.org/GO.format.obo-1_4.shtml`

Mappings in BioPortal are represented as a 4-tuple of the form $\langle e_1, e_2, Rel, Ann\rangle$, where e_1, e_2 are the URIs of two entities from the vocabulary of two BioPortal ontologies, Rel is the semantic relationship between them, and Ann is a set of annotations or metadata associated to the mapping. The relation Rel can be of one of the following types:[6] *skos:exactMatch*, *skos:relatedMatch*, *skos:closeMatch*, *skos:narrowMatch*, *skos:broadMatch*. Ann includes, among other details, important provenance information about the mapping such as: origin (e.g., user-defined or alignment system employed), application context, creator, and creation date.

According to BioPortal authors [33], the statistics about mapping origin are the following: *(i)* 64.96% of the mappings were created by the lexical matcher LOOM [13]; *(ii)* 32.48% of the mappings had UMLS [2] as origin; *(iii)* 2.41% represented mappings between entities with the same URI; *(iv)* 0.02% came from *Xref* OBO Mappings; *(v)* finally 0.13% of the mappings were submitted by users.

Mappings between entities with the same URI are labeled *skos:exactMatch* by BioPortal, LOOM and UMLS mappings are labeled *skos:closeMatch*, and *Xref* OBO Mappings are labeled *skos:relatedMatch*. User submitted mappings can be labeled with any of the relation types listed above.

BioPortal mappings can be retrieved via its REST API, being straightforward to identify the entities involved in the mapping, its origin, and the source ontologies.[7]

In this paper, we have focused only on *skos:closeMatch* mappings, which account for the large majority of BioPortal mappings. We represented these mappings as OWL 2 equivalence axioms since that is typically the semantic relation they convey (the tag *skos:closeMatch* is used to link concepts that can be used interchangeably, at least in a given context). Mapping annotations Ann have (optionally) been represented as OWL 2 axiom annotations. This representation of mappings enables the reuse of the extensive range of OWL 2 reasoning infrastructure that is currently available. Note that alternative formal semantics for ontology mappings have been proposed in the literature (e.g., [3,6,28]).

3 Mapping repair

The ontology resulting from the integration of $\mathcal{O}_1$ and $\mathcal{O}_2$ via a set of mappings $\mathcal{M}$, may entail axioms that do not follow from $\mathcal{O}_1$, $\mathcal{O}_2$, or $\mathcal{M}$ alone. These new semantic consequences can be captured using the notion of *deductive difference* [24,18], and can be divided into desired and undesired entailments. Undesired entailments are typically introduced due to erroneous mappings in $\mathcal{M}$. However, even if all mappings in $\mathcal{M}$ are correct, undesired entailments may occur due to conflicting descriptions between the overlapping entities in $\mathcal{O}_1$ and $\mathcal{O}_2$. Undesired entailment can be divided into two groups: entailments causing unsatisfiable classes, which can be easily detected using (automatic) logical reasoning; and entailments not causing unsatisfiable classes, which require domain knowledge to decide whether they are indeed undesired. In this paper we only focus on the first group of undesired entailments.

[6] http://www.bioontology.org/wiki/index.php/BioPortal_Mappings

[7] http://data.bioontology.org/documentation#Mapping

A set of mappings that leads to unsatisfiable classes in $\mathcal{O}_1 \cup \mathcal{O}_2 \cup \mathcal{M}$ is referred to as *incoherent* w.r.t. $\mathcal{O}_1$ and $\mathcal{O}_2$ [26].

Definition 1 (Mapping Incoherence). *A set of mappings $\mathcal{M}$ is incoherent with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$, if there exists a class A in the signature of $\mathcal{O}_1 \cup \mathcal{O}_2$ such that $\mathcal{O}_1 \cup \mathcal{O}_2 \not\models A \sqsubseteq \bot$ and $\mathcal{O}_1 \cup \mathcal{O}_2 \cup \mathcal{M} \models A \sqsubseteq \bot$.*

An incoherent set of mappings $\mathcal{M}$ can be fixed by removing mappings from $\mathcal{M}$. This process is referred to as *mapping repair* (or repair for short).

Definition 2 (Mapping Repair). *Let $\mathcal{M}$ be an incoherent set of mappings w.r.t. $\mathcal{O}_1$ and $\mathcal{O}_2$. A set of mappings $\mathcal{R} \subseteq \mathcal{M}$ is a mapping repair for $\mathcal{M}$ w.r.t. $\mathcal{O}_1$ and $\mathcal{O}_2$ if $\mathcal{M} \setminus \mathcal{R}$ is coherent w.r.t. $\mathcal{O}_1$ and $\mathcal{O}_2$.*

A trivial repair is $\mathcal{R} = \mathcal{M}$, since an empty set of mappings is obviously coherent. Nevertheless, the objective is to remove as few mappings as possible. Minimal (mapping) repairs are typically referred to in the literature as *mapping diagnosis* [25].

In the literature there are different approaches to compute a repair or diagnosis for an incoherent set of mappings. Early approaches were based on Distributed Description Logics (DDL) (e.g. [27,28,30]). The work presented in [30] deserves special mention, as it reports on a preliminary coherence evaluation of BioPortal mappings using DDL.[8] The authors, however, emphasized the problems of efficiency of the coherence checking task due to the reasoning complexity of DDL and suggest the use of approximate techniques in the future.

Alternatively, if mappings are represented as OWL 2 axioms, mapping repairs can also be computed using the state-of-the-art approaches for debugging and repairing inconsistencies in OWL 2 ontologies, which rely on the extraction of justifications for the unsatisfiable classes (e.g. [36,22,39,18]). However, justification-based technologies do not scale when the number of unsatisfiabilities is large (a typical scenario in mapping repair problems [16]).

To address this scalability issue, mapping repair systems usually compute an *approximate repair* using incomplete reasoning techniques (e.g. [17,25,9]). An approximate repair $\mathcal{R}^{\approx}$ does not guarantee that $\mathcal{M} \setminus \mathcal{R}^{\approx}$ is coherent, but it will (in general) reduce significantly the number of unsatisfiabilities caused by the original mappings $\mathcal{M}$. Indeed, approximate repair techniques have been successfully applied to audit the UMLS metathesaurus [19,17].

In this paper, we have applied the *approximate* mapping repair techniques implemented in LogMap [17,20,21] and AgreementMakerLight (AML) [9,35] to the BioPortal mappings. As described in Section 2, we have represented the BioPortal mappings as OWL 2 equivalence axioms. Note that, although both LogMap and AML were originally implemented as ontology matching systems, they can also operate as a standalone mapping repair systems. From this point onwards, we will refer to LogMap's and AML's repair modules as LogMap-Repair and AML-Repair respectively.

Algorithm 1 AML-Repair algorithm

Input: $\mathcal{O}_1$, $\mathcal{O}_2$: input ontologies; $\mathcal{M}$: input mappings
Output: $\mathcal{M}'$: output mappings; $\mathcal{R}^{\approx}$: approximate mapping repair; $\mathcal{CS}$: identified conflicting sets; $\mathcal{M}_{\mathcal{CS}}$: mappings involved in conflicting sets

```
1: $\mathcal{M}' := \mathcal{M}$
2: $\mathcal{R}^{\approx} := \emptyset$
3: $\langle \mathcal{O}'_1, \mathcal{O}'_2, \text{Checkset} \rangle := \mathsf{BuiltCoreFragments}(\mathcal{O}_1, \mathcal{O}_2, \mathcal{M}')$
4: $\mathcal{CS} := \mathsf{ConflictSets}(\mathcal{O}'_1, \mathcal{O}'_2, M', \text{Checkset})$
5: $\mathcal{M}_{\mathcal{CS}} := \mathsf{MappingsInConflictSets}(\mathcal{CS})$
6: $\mathcal{CS}' := \mathcal{CS}$
7: while $|\mathcal{CS}'| > 0$ do
8:     $w := \mathsf{SelectMappingToRemove}(\mathcal{CS}')$
9:     $\mathcal{CS}' := \mathsf{RemoveMapping}(\mathcal{CS}', w)$
10:    $\mathcal{M}' : = \mathcal{M}' \setminus \{w\}$;
11:    $\mathcal{R}^{\approx} := \mathcal{R}^{\approx} \cup \{w\}$
12: end while
13: return $\langle \mathcal{M}', \mathcal{R}^{\approx}, \mathcal{CS}, \mathcal{M}_{\mathcal{CS}} \rangle$
```

3.1 Mapping repair using AML-Repair

The pseudocode of the algorithm implemented by AML-Repair is described in Algorithm 1. The algorithm is divided in three main tasks:

1. The computation of the core fragments (see [34]) (step 3);
2. The search for all (minimal) conflicting sets of mappings $\mathcal{CS}$, i.e. mappings that lead to an incoherence (step 4);
3. The resolution of incoherences using a heuristic to minimize the set of mappings removed from every conflicting set (step 8 to 11);
4. The algorithm outputs a set of repaired mappings $\mathcal{M}'$, an approximate mapping repair $\mathcal{R}^{\approx}$, conflicting sets of mappings $\mathcal{CS}$, and the set of all mappings involved in at least one conflicting set $\mathcal{M}_{\mathcal{CS}}$.

AML-Repair implementation is based on a modularization of the input ontologies, called core fragments, that only contains the necessary classes and relations to detect all existing incoherences [34]. This modularization is computed by the BuildCoreFragments method (Step 3 of Algorithm 1), which also computes a minimal set of classes (the Checkset) that need to be checked for incoherences.

AML-Repair determines subsumption relations between atomic classes syntactically (i.e., without using an OWL 2 Reasoner) and it also considers disjointness axioms between atomic classes. Unlike LogMap-Repair, equivalence mappings are considered indivisible units and are never split into two subsumption mappings. Thus, an input mapping is either removed or kept in the alignment during the repair procedure.

The ConflictSets method (step 4) returns all mapping sets that will lead to an incoherence by doing a full depth-first search in the core fragments structure for each class in the Checkset. This way, AML-Repair determines all minimal sets of mappings, called

[8] To the best of our knowledge, no automatic repair was conducted.

Algorithm 2 LogMap-Repair algorithm based on Horn propositional reasoning

```
Input: $\mathcal{O}_1$, $\mathcal{O}_2$: input ontologies; $\mathcal{M}$: input mappings
Output: $\mathcal{M}'$: output mappings; $\mathcal{R}^{\approx}$: approximate mapping repair; $\mathcal{CG}$: conflicting groups;
$\overline{\mathcal{M}_{\mathcal{CG}}}$: mapping average in conflicting groups
 1: $\mathcal{M}' := \mathcal{M}$
 2: $\mathcal{R}^{\approx} := \emptyset$
 3: $\mathcal{CG} := \emptyset$
 4: $\langle \mathcal{P}_1, \mathcal{P}_2 \rangle := \mathsf{PropEncoding}(\mathcal{O}_1, \mathcal{O}_2)$
 5: for each $C \in \mathsf{OrderedVariables}(\mathcal{P}_1 \cup \mathcal{P}_2)$ do
 6:     $\mathcal{P}_C := \mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{M}' \cup \{\mathsf{true} \rightarrow C\}$
 7:     $\langle sat, \mathcal{M}_\perp \rangle := \mathsf{DowlingGallier}(\mathcal{P}_C)$
 8:     if $sat = \mathsf{false}$ then
 9:         $\mathcal{CG} := \mathcal{CG} \cup \{\mathcal{M}_\perp\}$
10:         $Rep := \emptyset$
11:         $rep\_size := 1$
12:         repeat
13:             for each subset $\mathcal{R}_C$ of $\mathcal{M}_\perp$ of size $rep\_size$ do
14:                 $sat := \mathsf{DowlingGallier}(\mathcal{P}_C \setminus \mathcal{R}_C)$
15:                 if $sat = \mathsf{true}$ then $Rep := Rep \cup \{\mathcal{R}_C\}$
16:             end for
17:             $rep\_size := rep\_size + 1$
18:         until $Rep \neq \emptyset$
19:         $\mathcal{R}_C :=$ element of $Rep$ with minimum aggregated confidence.
20:         $\mathcal{M}' := \mathcal{M}' \setminus \mathcal{R}_C$
21:         $\mathcal{R}^{\approx} := \mathcal{R}^{\approx} \cup \mathcal{R}_C$
22:     end if
23: end for
24: $\overline{\mathcal{M}_{\mathcal{CG}}} := \mathsf{AverageMappingsInConflictGroups}(\mathcal{CG})$
25: return $\langle \mathcal{M}', \mathcal{R}^{\approx}, \mathcal{CG}, \overline{\mathcal{M}_{\mathcal{CG}}} \rangle$
```

conflicting sets $\mathcal{CS}$, which cause the incoherences. Since conflicting sets are minimal, a conflicting set is resolved if at least one of its mappings is removed. The algorithm also keeps the set $\mathcal{M}_{\mathcal{CS}}$ containing all mappings involved in a conflicting set (Step 5).

AML-Repair aims to minimize the number of removed mappings by determining a minimal set of mappings that intersect all conflict sets. Given that computing this set is NP-Complete, AML-Repair uses an efficient heuristic procedure that consists of iteratively removing the mappings that belong to the highest number of conflicting sets (as identified in Step 8 of Algorithm 1), and in case of tie, those that have the lowest confidence values. This strategy typically produces near-optimal results.

3.2 Mapping repair using LogMap-Repair

Algorithm 2 shows the pseudocode of the algorithm implemented by LogMap-Repair. Steps 1-3 initialise the output variables. LogMap-Repair encodes the input ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ as Horn propositional theories $\mathcal{P}_1$ and $\mathcal{P}_2$ (Step 4) and exploits this encoding to subsequently detect unsatisfiable classes in an efficient and sound way during the repair process. The theory $\mathcal{P}_1$ (resp. $\mathcal{P}_2$) consists of the following Horn rules:

- A rule $A \rightarrow B$ for all distinct classes A, B such that A is subsumed by B in $\mathcal{O}_1$ (resp. in O_2); subsumption relations can be determined using either an OWL 2 reasoner, or syntactically (in an incomplete way).
- Rules $A_i \wedge A_j \rightarrow \mathsf{false}$ $(1 \leq i < j \leq n)$ for each disjointness axiom of the form $DisjointClasses(A_1, \ldots, A_n)$.
- A rule $A_1 \wedge \ldots \wedge A_n \rightarrow B$ for each subclass or equivalence axiom having the intersection of $A_1, \ldots A_n$ as subclass expression and B as superclass.

In Step 5, propositional variables in $\mathcal{P}_1$ (resp. in $\mathcal{P}_2$) are ordered such that a variable C in $\mathcal{P}_1$ (resp. in $\mathcal{P}_2$) comes before D whenever D is subsumed by C in $\mathcal{O}_1$ (resp. in $\mathcal{O}_2$). This is a well-known repair strategy: subclasses of an unsatisfiable class are also unsatisfiable and hence before repairing an unsatisfiable class one first needs to repair its superclasses. Satisfiability of a propositional variable C is determined by checking satisfiability of the propositional theory $\mathcal{P}_C$ (Step 6) consisting of *(i)* the rule $(\mathsf{true} \rightarrow C)$; *(ii)* the propositional representations $\mathcal{P}_1$ and $\mathcal{P}_2$; and *(iii)* the current set of output mappings $\mathcal{M}'$ (seen as propositional implications). Note that LogMap-Repair splits equivalence mappings into two equivalent subsumption mappings.

LogMap-Repair implements the classical Dowling-Gallier algorithm for propositional Horn satisfiability [5,11]. LogMap-Repair's implementation of Dowling-Gallier's algorithm also records all mappings potentially involved in an unsatisfiability. Thus, a call to Dowling-Gallier returns a satisfiability value sat and, optionally, the (overestimated) group of conflicting mappings $\mathcal{M}_{\perp}$ (see Steps 7 and 14). For statistical purposes, the set $\mathcal{CG}$ keeps all conflicting groups for the identified unsatisfiable classes (Step 9). An unsatisfiable class C is repaired by discarding conflicting mappings for C (Steps 10 to 21). Thus, subsets $\mathcal{R}_C$ of $\mathcal{M}_{\perp}$ of increasing size are then identified until a repair is found (Steps 12-18). Note that, LogMap-Repair does not compute a diagnosis for the unsatisfiable class C but rather the repairs of smallest size. If several repairs of a given size exist, the one with the lowest aggregated confidence is selected according to the confidence values assigned to mappings (Step 19). Steps 20 and 21 update the output mappings $\mathcal{M}'$ and the approximate mapping repair $\mathcal{R}^{\approx}$ by extracting and adding $\mathcal{R}_C$, respectively. Finally, Step 24 calculates the average number of mappings in each identified conflicting group $\mathcal{CG}$.

Algorithm 2 ensures that $\mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{M}' \cup \{\mathsf{true} \rightarrow C\}$ is satisfiable for each C occurring in $\mathcal{P}_1 \cup \mathcal{P}_2$. The propositional encoding of $\mathcal{O}_1$ and $\mathcal{O}_2$ is, however, incomplete and hence the algorithm does not ensure satisfiability of each class in $\mathcal{O}_1 \cup \mathcal{O}_2 \cup \mathcal{M}'$. Nevertheless, the number of unsatisfiable classes remaining after computing an approximate repair $\mathcal{R}^{\approx}$ is typically small.

4 Evaluation

In order to evaluate the coherence of BioPortal mappings, we manually selected 19 ontology pairs from BioPortal such that *(i)* each pair had at least 500 mappings listed in BioPortal, *(ii)* at least one of the ontologies in the pair contained disjointness clauses between their classes, and *(iii)* the domain of both ontologies was biomedical. The purpose of the first two criteria is to exclude ontology pairs that are uninteresting from an (automatic) mapping repair perspective, whereas the third criterion ensures that we are

Table 1: Ontologies comprising the 19 ontology pairs selected.

Ontology	Acronym	# Classes	Source
Bone Dysplasia Ontology	BDO	13,817	BioPortal
Cell Culture Ontology	CCONT	14,663	BioPortal
Experimental Factor Ontology	EFO	14,499	BioPortal
Human Developmental Anatomy Ontology, timed ver.	EHDA	8,340	OBO Foundry
Cardiac Electrophysiology Ontology	EP	81,957	BioPortal
Foundational Model of Anatomy	FMA	83,280	BioPortal
Mouse Adult Gross Anatomy Ontology	MA	3,205	OBO Foundry
NCI Thesaurus	NCIT	105,347	BioPortal
Online Mendelian Inheritance in Man	OMIM	76,721	BioPortal
Sleep Domain Ontology	SDO	1,382	BioPortal
SNP ontology	SNP	2,206	BioPortal
Sequence Types and Features Ontology	SO	2,021	BioPortal
Teleost Anatomy Ontology	TAO	3,372	OBO Foundry
Uber Anatomy Ontology	UBERON	15,773	OBO Foundry
Zebrafish Anatomy and Development Ontology	ZFA	2,955	OBO Foundry

Table 2: BioPortal mappings for the selected ontology pairs

Ontology Pair	Listed Mappings	Retrieved Mappings	Actual Mappings	Unsat. Classes
BDO-NCIT	1,637	1,636	1,636	34,341
CCONT-NCIT	2,815	2,813	2,097 (-19)	50,304
EFO-NCIT	3,289	3,287	2,507	60,347
EHDA-FMA	3,731	2,496	2,496	0
EP-FMA	79,497	78,489	78,489	210
EP-NCIT	2,468	2,465	2,465 (-1)	14,687 (-1)
MA-FMA	5,491	961	961	850
OMIM-NCIT	5,198	5,198	5,178	70,172
SDO-EP	662	135	135	44
SDO-FMA	593	529	529 (-1)	0
SNPO-SO	2,168	2,150	2,028 (-1)	0
UBERON-FMA	2,233	1,932	1,932	4,753
ZFA-CCONT	532	437	333	0
ZFA-EFO	773	538	427	913
ZFA-EHDA	2,595	1,809	1,809	0
ZFA-FMA	1,240	265	265	0
ZFA-MA	1,639	129	129	0
ZFA-TAO	1,737	1,524	1,521	0
ZFA-UBERON	817	724	724	104

able to manually evaluate the repair results as they lie within our domain of expertise. This selection was not exhaustive, as our goal was merely to select a substantial and representative set of ontology pairs.

The 15 ontologies comprising these 19 pairs are listed in Table 1. We retrieved the latest OWL version of each ontology from BioPortal, except for the ontologies that

Algorithm 3 Automatic repair evaluation of BioPortal mappings

```
Input: $\mathcal{O}_1$, $\mathcal{O}_2$: two BioPortal ontologies; $\mathcal{M}$: the set of BioPortal mappings between them
Compute all conflict sets of mappings $\mathcal{CS}$, the total number of mappings involved in conflicts $\mathcal{M}_{\mathcal{CS}}$, and the approximate repair $\mathcal{R}^{\approx}$ using AML-Repair system    ▷ See Algorithm 1
Get unsatisfiable classes of $\mathcal{O}_1 \cup \mathcal{O}_2 \cup \mathcal{M} \setminus \mathcal{R}^{\approx}$ using ELK reasoner
Compute the conflicting mapping groups $\mathcal{CG}$ per unsatisfiability, the average number of mappings per conflict group $\overline{\mathcal{M}_{\mathcal{CG}}}$, and the approximate repair $\mathcal{R}^{\approx}$ using LogMap-Repair system    ▷ See Algorithm 2
Get unsatisfiable classes of $\mathcal{O}_1 \cup \mathcal{O}_2 \cup \mathcal{M} \setminus \mathcal{R}^{\approx}$ using ELK reasoner
```

were only available in OBO format. Because AML is currently not set-up to handle ontologies in the OBO format, we retrieved the latter from the OBO Foundry[9] [38] in OWL format (making sure the versions matched those in BioPortal).

We implemented a script that, given a pair of ontologies, uses BioPortal's REST API to retrieve all mappings between those ontologies. We focused only on *skos:closeMatch* mappings and we represented them as OWL 2 equivalence axioms. We did not consider *skos:exactMatch* mappings since they represent correspondences between entities with the same URI, which in OWL ontologies are considered equivalent (even though the equivalence between them is not explicitly defined). We also excluded a few mappings that had only a *null* source or involved only one entity.

The mappings between the 19 selected ontology pairs are listed in Table 2. We verified that the number of retrieved mappings did not match the number of mappings listed in the BioPortal website, and that sometimes the discrepancy was large (i.e., not accounted for by the small fraction of mappings we excluded). BioPortal developers confirmed that there is indeed an inconsistency between the metrics and the available mappings. Furthermore, in several cases, some of the mappings retrieved pointed to classes that were not found in the ontologies (possibly obsolete classes), so the actual mappings between the ontologies were less than those retrieved. Additionally, in some cases less mappings were found when using the Jena API to read the ontologies (used by AML) than when using the OWL API (used by LogMap). The difference between the two is shown in parenthesis in Table 2.

Finally, we computed the satisfiability of each alignment with the OWL 2 EL reasoner ELK [23], finding several unsatisfiable classes in 11 of the alignments. We opted for ELK for the sake of efficiency, given the size of some of the ontologies. ELK is incomplete and thus the identified unsatisfiabilities represent a lower bound of the actual number of such logical errors.

4.1 Automatic Repair Evaluation

For each of the 11 ontology pairs that had incoherent mapping sets (as detected by ELK and listed in Table 2), we conducted the evaluation detailed in Algorithm 3. The results we obtained are shown in Table 3. Note that LogMap-Repair splits equivalence mappings into two subsumption mappings, so the value of $\mathcal{R}^{\approx}$ is not directly comparable with AML-Repair (the latter should be doubled to compare it with the former).

[9] `http://www.obofoundry.org/`

Table 3: Automatic mapping repair using AML-Repair and LogMap-Repair

Ontology Pairs	$\mathcal{M}$	**AML-Repair**				**LogMap-Repair**			
		$\lvert\mathcal{CS}\rvert$	$\lvert\mathcal{M}_{\mathcal{CS}}\rvert$	$\lvert\mathcal{R}^{\approx}\rvert$	**Unsat.**	$\lvert\mathcal{CG}\rvert$	$\overline{\mathcal{M}_{\mathcal{CG}}}$	$\lvert\mathcal{R}^{\approx}\rvert$	**Unsat.**
BDO-NCIT	1,636	1,649	1,374 (84%)	53	0	125	3.2	154	0
CCONT-NCIT	2,097	1,197	1,136 (55%)	55	3,630	125	2.7	119	75
EFO-NCIT	2,507	1,731	1,541 (61%)	143	3,687	311	4.3	353	73
EP-FMA	78,489	348	109 (0.1%)	16	0	168	11.0	168	0
EP-NCIT	2,465	363	307 (12%)	69	253	136	3.8	180	0
MA-FMA	961	21	22 (2%)	1	0	1	2.0	2	0
OMIM-NCIT	5,178	1,800	1,078 (21%)	154	0	396	10.2	536	0
SDO-EP	135	3	3 (2%)	1	0	1	3.0	3	0
UBERON-FMA	1,932	486	121 (6%)	19	25	70	6.3	85	25
ZFA-EFO	427	7	11 (3%)	5	0	10	2.6	10	0
ZFA-UBERON	724	0	0 (0%)	0	104	0	0	0	104
Average	8,777	691	518 (22%)	94	700	122	5	146	25

$\mathcal{M}$: number of BioPortal mappings; $\lvert\mathcal{CS}\rvert$: number of conflict sets; $\lvert\mathcal{M}_{\mathcal{CS}}\rvert$: number of distinct mappings in conflict sets; $\lvert\mathcal{CG}\rvert$: number of conflict groups; $\overline{\mathcal{M}_{\mathcal{CG}}}$: average number of mappings per conflict group; $\lvert\mathcal{R}^{\approx}\rvert$: repair size in number of mappings (AML) or number of half-mappings (LogMap); **Unsat.**: unsatisfiable classes after repair.

The incoherence of the repaired mapping sets has been significantly reduced, and in many cases completely removed. The one exception was the ZFA-UBERON case, as neither AML-Repair nor LogMap-Repair could detect and repair any of the unsatisfiabilities in this alignment. Furthermore, the computed (approximate) repairs were not aggressive, as they removed at most 5.7% (AML-Repair) and 7% (LogMap-Repair) of the mappings (in the EFO-NCIT case).

In addition to producing a repair, AML-Repair also identifies the number of conflicting mapping sets $\mathcal{CS}$ and the total number of mappings that are involved in at least one conflict $\mathcal{M}_{\mathcal{CS}}$. For example, in the BDO-NCIT case, AML-Repair identifies 1,649 conflicting sets which involve 84% of the mappings $\mathcal{M}$ in this alignment. Given that these mappings were leading to 34,341 unsatisfiable classes (see Table 2), the fact that only 53 equivalence mappings were removed indicates that (at least) some of these were causing several unsatisfiabilities, likely because they were in conflict with multiple other mappings.

LogMap-Repair, on the other hand, identifies groups of potentially conflicting mappings $\mathcal{CG}$ (which contain one or more $\mathcal{CS}$) involved in each unsatisfiability, and the average number of mappings in each conflicting group $\overline{\mathcal{M}_{\mathcal{CG}}}$. $\mathcal{CG}$ represents a lower bound of the total number of groups, since LogMap-Repair repairs on-the-fly and removing one mapping may solve multiple unsatisfiabilities. For example, in the BDO-NCIT case, LogMap-Repair only identifies 125 conflicting groups with an average of 3.2 mappings per group. Note that solving the 125 unsatisfiabilities corresponding to the conflict groups is sufficient to repair the original 34,341 unsatisfiabilities, which again suggests that a few mappings in the conflict groups were causing many of these errors.

4.2 Manual Analysis

To complement the automatic repair evaluation and investigate the causes behind the incoherences identified therein, we analyzed manually the mappings removed by AML-Repair and LogMap-Repair (up to a maximum of 100 mappings per ontology pair, and in the case of LogMap-Repair, only the cases where the subsumption mappings were removed in both directions).

For each removed mapping, we assessed whether it was correct or erroneous (within the context of the ontologies). We deemed a mapping to be erroneous if it falls into one of the following categories:

1. At least one of the entities it maps is obsolete/retired, as in the mapping: BDO#HP_0001596 (Alopecia) ⇔ NCIT#C2865 (Alopecia), where the latter class is retired in NCIT.
2. The entities it maps are not directly related, as in the mapping: BDO#PATO_0001901 (Back) ⇔ NCIT#C13062 (Back), where the former class stands for the directional qualifier and the latter stands for the body part.
3. The entities it maps are related but the relationship between them should not be modeled as *skos:closeMatch*, as in the the mapping: BDO#G0000064 (CREBP) ⇔ NCIT#C17803 (CREB-Binding Protein), which maps entities that are related (the gene and corresponding protein) but semantically distinct. Moreover, this mapping conflicts with the correct (protein-protein) mapping: BDO#P000022 (CREB-Binding Protein) ⇔ NCIT#C17803 (CREB-Binding Protein).

Additionally, when the removed mapping was deemed to be correct, we analyzed the conflict sets $\mathcal{CS}$ in which the removed mapping was present (computed by AML-Repair, see Algorithm 3) and assessed whether the mappings in conflict with it were correct or erroneous. For the purpose of our evaluation, the main issue is not whether the repair algorithms remove erroneous mappings, but rather whether any of the unsatisfiabilities in which it is involved are caused by erroneous mappings. Thus, if either the removed mapping or at least one of its conflicting mappings was erroneous, we attributed the cause of removal to a mapping error. If the mapping itself and all of its conflicting mappings were correct, we considered the cause of removal to be an incompatibility between the ontologies.

The results of our manual analysis are summarized in Table 4. In total, over 40% of the mappings removed by both repair systems were indeed erroneous. Furthermore, errors in the mappings were the cause of removal of over 60% of the mappings.

We found that category 1 errors (i.e., mappings including obsolete/retired classes) were relatively common in all alignments that included NCIT. Furthermore, there were two common category 3 error patterns in these alignments: gene-protein matches, and human-mouse matches. The former pattern consists of a mapping between a gene and its corresponding protein or vice-versa, as exemplified above. The latter pattern consists of a mapping between a (Human) Health/Anatomy classes and a corresponding NCIT Mouse class (from the Mouse Pathologic Diagnoses or Mouse Anatomy Concepts sections) which naturally conflicts with the mapping to the main (Human) NCIT sections. One example of this pattern is the mapping: BDO#HP_0010786 (Urinary Tract Neoplasm) ⇔ NCIT#C25806 (Mouse Urinary Tract Neoplasm). Another pattern of this

Table 4: Manual evaluation of the repaired BioPortal mappings

Ontology Pairs	**AML-Repair**			**LogMap-Repair**		
	Analyzed	**Erroneous**	**Err. Cause**	**Analyzed**	**Erroneous**	**Err. Cause**
BDO-NCIT	53	55%	83%	52	62%	83%
CCONT-NCIT	55	33%	62%	68	43%	59%
EFO-NCIT	100	53%	91%	100	54%	93%
EP-FMA	16	0%	0%	84	0%	0%
EP-NCIT	69	43%	71%	78	60%	73%
MA-FMA	1	100%	100%	1	100%	100%
OMIM-NCIT	100	48%	71%	100	49%	76%
SDO-EP	1	100%	100%	0	N/A	N/A
UBERON-FMA	19	0%	0%	20	0%	0%
ZFA-EFO	5	60%	100%	4	75%	100%
Total	419	44%	71%	507	42%	62%

category that occurred in the OMIM-NCIT alignment consists of a mapping between a disease/symptom and a corresponding adverse event, such as: OMIM#MTHU023845 (Neck Pain) ⇔ NCIT#C56135 (Neck Pain Adverse Event).

Regarding category 2 errors, there were few patterns other than number mismatches, such as in the mapping: BDO#G0000133 (TBX4) ⇔ NCIT#C101638 (TBX3 Gene). While the cause of these mismatches was often clear, as in the 'back' example above, some defy reason as the case of: EFO#CHEBI_15366 (Acetic Acid) ⇔ NCIT#C37392 (C58 Mouse).

As for incompatibilities between the ontologies, one of the most interesting cases is the EP-FMA alignment, which is actually an alignment between the OBO version of FMA (which is imported by EP) and the BioPortal version of FMA. Indeed, it was surprising to find that the alignment is incoherent, given that all mappings are true equivalences. It turns out that there are a few structural differences between the two versions of the ontology which cause the incoherences, as some entities are modeled as 'Material Anatomical Entity' in the OBO version and as 'Immaterial Anatomical Entity' in the BioPortal version (with the latter appearing to be more correct in most cases). The same type of structural differences is also behind the incoherences in the UBERON-FMA alignment. Also interesting is the OMIM-NCIT alignment, as OMIM models diseases as subclasses of the anatomical structures where they occur, whereas NCIT models diseases as disjoint from anatomical structures, making it impossible to obtain a coherent alignment between the ontologies where both diseases and anatomical structures are mapped.

4.3 Discussion

The results of our study reveal that many sets of BioPortal mappings lead to logical incoherences when taken as a whole, and that many of these incoherences involve erroneous mappings. Thus, adding annotations to BioPortal mappings about potential logical incompatibilities with other mappings would not only improve their usability for semantic web applications, which require logical integrity, but also contribute to identify and discard erroneous mappings.

Our study also demonstrates that approximate repair algorithms such as AML-Repair and LogMap-Repair can effectively identify most of the logical conflicts in BioPortal mappings, as well as the mappings that cause them. Furthermore, unlike complete repair algorithms such as those based on DDL [30], AML-Repair and LogMap-Repair are feasible in practice (repair times in AML-Repair ranged from 10 seconds to 10 minutes, whereas in LogMap-Repair they ranged from 3 to 92 seconds, in a quad-core computer with 8 GB allocated RAM). Thus these algorithms could play a pivotal role in the task of identifying and annotating conflicts between BioPortal mappings.

Furthermore, in addition to the annotation of existing mappings, AML-Repair and LogMap-Repair could be employed to screen newly submitted mappings, so that those leading to logical conflicts can be reviewed before being integrated into BioPortal. This could effectively preclude the addition of some erroneous mappings, and would enable the immediate annotation of the mappings accepted for integration.

Regarding the categories of erroneous mappings found, the category 1 errors (i.e, mappings that include retired/obsolete entities) are straightforward to identify and handle automatically, even without the use of repair algorithms. These mappings should not be maintained as 'active' mappings in BioPortal given that retired/obsolete entities are no longer be considered an active part of the ontologies. However, it makes sense to keep track of such mappings, so the best solution would be to annotate the mappings themselves as obsolete.

Category 2 errors (i.e., mappings between unrelated classes) should definitely be excluded from BioPortal, whereas category 3 errors (i.e., mappings between classes that are related but not closely) can be addressed either by exclusion or by changing the semantic relationship. For instance, a gene and its corresponding protein could be considered *skos:relatedMatch* rather than *skos:closeMatch*, although ideally a more descriptive mapping relation should be used. However, if both ontologies describe the gene and the protein, at least one ontology describes the relation between them, and BioPortal includes both the gene-gene and the protein-protein mappings; then maintaining a gene-protein mapping is semantically redundant.

Although finding category 2 and 3 errors typically requires human intervention, the use of mapping repair algorithms is critical to facilitate their detection. Note that, not all erroneous mappings necessarily lead to logical conflicts, particularly when the ontologies lack disjointness definitions. Nevertheless, addressing conflict-causing errors will surely be a significant improvement, and the common error patterns thus identified can be employed to search for (non-conflict causing) errors, even in ontologies that lack disjointness restrictions.

The identification of logical conflicts caused by inherent incompatibilities between the ontologies is also critical to understand the limits of interoperability. For instance, integrating OMIM and NCIT requires excluding either mappings between anatomic entities or mappings between diseases (depending on the intended application), or ultimately relaxing the disjointness restrictions in the NCIT. Additionally, such incompatibilities may point to modeling errors in the ontologies, as in the EP-FMA case, and enable their correction.

5 Conclusions

BioPortal fulfills a critical need of the biomedical community by promoting integration and interoperability between the numerous biomedical ontologies with overlapping domains. However, the different scopes of these ontologies often lead to incompatible views of a given domain, placing restrictions on interoperability. Maintaining conflicting mappings may best serve the needs of the community, as a wider mapping coverage will satisfy more users and enable more applications. Nevertheless, if BioPortal mappings are to be usable on a large scale, and particularly by automatic applications, then identifying those that lead to logical errors is paramount.

Another issue that affects BioPortal are mapping errors, which are inevitable on this scale, particularly when most mappings are produced by automated ontology matching techniques. Finding and correcting these errors is a daunting task, but one of the utmost importance, as they are likely to propagate if used to draw inferences. While not all errors cause logical conflicts, many do, and as our evaluation illustrates, identifying these enables the discovery of error patterns that can be applied to identify further errors.

Identifying logical conflicts in BioPortal mappings thus serves the dual purpose of improving usability and facilitating error detection. Given that using complete reasoners for this task is unfeasible, due to the scale of BioPortal, approximate mapping repair systems such as AML-Repair and LogMap-Repair appear to be the ideal solution. Indeed, our study has shown that these systems are both effective and efficient in tackling large sets of mappings, and will be even more efficient considering that the goal is only to identify conflicts rather than to repair them.

Our proposal is that BioPortal mappings be enriched with annotations about other mappings they conflict with, a solution which fits into BioPortal's community-driven and multiple-perspective approach. While distinguishing mappings errors from incompatibilities will require manual analysis, this is a task that could be carried out gradually by the community once mappings are annotated with logical conflicts.

There is, however, one type of error that can be addressed immediately: mappings that include obsolete/retired entities. Despite the fact that there are different representations of these entities among BioPortal ontologies, identifying them (and their mappings) should be straightforward to do automatically. We propose that such mappings be annotated as obsolete, which would enable BioPortal and its users to keep track of them while allowing their automatic exclusion by applications.

Our next step will be to contact BioPortal developers and collaborate with them in the process of finding and annotating mappings with information about logical conflicts, by applying our repair algorithms to the whole BioPortal.

Acknowledgements

This work was supported by the EU FP7 IP project Optique (no. 318338) and the EPSRC project Score!, and by the Portuguese FCT through the SOMER project (PTDC/

EIA-EIA/119119/2010) and the LASIGE Strategic Project (PEst-OE/EEI/UI0408/2014).

We would like to thank the invaluable help provided by Bernardo Cuenca Grau and Ian Horrocks in the development of LogMap-Repair, and by Isabel F. Cruz in the development of AML-Repair.

References

1. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., et al.: Gene Ontology: tool for the unification of biology. Nature genetics 25(1), 25–29 (2000)
2. Bodenreider, O.: The unified medical language system (UMLS): integrating biomedical terminology. Nucleic Acids Research 32, 267–270 (2004)
3. Borgida, A., Serafini, L.: Distributed description logics: Assimilating information from peer sources. J. Data Sem. 1, 153–184 (2003)
4. Cuenca Grau, B., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P.F., Sattler, U.: OWL 2: The next step for OWL. J. Web Sem. 6(4), 309–322 (2008)
5. Dowling, W.F., Gallier, J.H.: Linear-time algorithms for testing the satisfiability of propositional Horn formulae. J. Log. Prog. 1(3), 267–284 (1984)
6. Euzenat, J.: Semantic precision and recall for ontology alignment evaluation. In: Int'l Joint Conf. on Artif. Intell. (IJCAI). pp. 348–353 (2007)
7. Euzenat, J., Meilicke, C., Stuckenschmidt, H., Shvaiko, P., Trojahn, C.: Ontology alignment evaluation initiative: Six years of experience. J. Data Sem. 15, 158–192 (2011)
8. Euzenat, J., Shvaiko, P.: Ontology matching. Springer (2007)
9. Faria, D., Pesquita, C., Santos, E., Palmonari, M., Cruz, I.F., Couto, F.M.: The agreementmakerlight ontology matching system. In: OTM Conferences. pp. 527–541 (2013)
10. Fridman Noy, N., Shah, N.H., Whetzel, P.L., Dai, B., Dorf, M., Griffith, N., Jonquet, C., Rubin, D.L., Storey, M.A.D., Chute, C.G., Musen, M.A.: BioPortal: ontologies and integrated data resources at the click of a mouse. Nucleic Acids Research 37(Web-Server-Issue) (2009)
11. Gallo, G., Urbani, G.: Algorithms for testing the satisfiability of propositional formulae. J. Log. Prog. 7(1), 45–61 (1989)
12. Ghazvinian, A., Noy, N.F., Jonquet, C., Shah, N.H., Musen, M.A.: What four million mappings can tell you about two hundred ontologies. In: Int'l Sem. Web Conf. (ISWC) (2009)
13. Ghazvinian, A., Noy, N.F., Musen, M.A.: Creating mappings for ontologies in biomedicine: Simple methods work. In: AMIA Annual Symposium (AMIA) (2009)
14. Golbeck, J., Fragoso, G., Hartel, F.W., Hendler, J.A., Oberthaler, J., Parsia, B.: The National Cancer Institute's Thésaurus and Ontology. J. Web Sem. 1(1), 75–80 (2003)
15. Golbreich, C., Horridge, M., Horrocks, I., Motik, B., Shearer, R.: OBO and OWL: Leveraging Semantic Web Technologies for the Life Sciences. In: Int'l Sem. Web Conf. (2007)
16. Jiménez-Ruiz, E., Cuenca Grau, B., Horrocks, I.: On the feasibility of using OWL 2 DL reasoners for ontology matching problems. In: OWL Reasoner Evaluation Workshop (2012)
17. Jiménez-Ruiz, E., Cuenca Grau, B.: LogMap: Logic-based and Scalable Ontology Matching. In: Int'l Sem. Web Conf. (ISWC). pp. 273–288 (2011)
18. Jiménez-Ruiz, E., Cuenca Grau, B., Horrocks, I., Berlanga, R.: Ontology integration using mappings: Towards getting the right logical consequences. In: Eur. Sem. Web Conf. (2009)
19. Jiménez-Ruiz, E., Cuenca Grau, B., Horrocks, I., Berlanga, R.: Logic-based Assessment of the Compatibility of UMLS Ontology Sources. J. Biomed. Semant. 2(Suppl 1), S2 (2011)
20. Jiménez-Ruiz, E., Cuenca Grau, B., Zhou, Y., Horrocks, I.: Large-scale interactive ontology matching: Algorithms and implementation. In: Europ. Conf. on Artif. Intell. (ECAI) (2012)

21. Jiménez-Ruiz, E., Meilicke, C., Grau, B.C., Horrocks, I.: Evaluating mapping repair systems with large biomedical ontologies. In: Description Logics. pp. 246–257 (2013)
22. Kalyanpur, A., Parsia, B., Horridge, M., Sirin, E.: Finding all justifications of OWL DL entailments. In: Int'l Sem. Web Conf. (ISWC). pp. 267–280. Springer (2007)
23. Kazakov, Y., Krötzsch, M., Simancik, F.: Concurrent classification of EL ontologies. In: Int'l Sem. Web Conf. (ISWC). pp. 305–320 (2011)
24. Konev, B., Walther, D., Wolter, F.: The Logical Difference Problem for Description Logic Terminologies. In: Int'l Joint Conf. on Automated Reasoning (IJCAR). pp. 259–274 (2008)
25. Meilicke, C.: Alignments Incoherency in Ontology Matching. Ph.D. thesis, University of Mannheim (2011)
26. Meilicke, C., Stuckenschmidt, H.: Incoherence as a basis for measuring the quality of ontology mappings. In: Ontology Matching Workshop (2008)
27. Meilicke, C., Stuckenschmidt, H., Tamilin, A.: Repairing ontology mappings. In: Proc. of AAAI Conf. on Artif. Intell. pp. 1408–1413 (2007)
28. Meilicke, C., Stuckenschmidt, H., Tamilin, A.: Reasoning support for mapping revision. J. Log. Comput. 19(5) (2009)
29. Noy, N.F., Griffith, N., Musen, M.A.: Collecting community-based mappings in an ontology repository. In: International Semantic Web Conference (ISWC). pp. 371–386 (2008)
30. Pathak, J., Chute, C.G.: Debugging Mappings between Biomedical Ontologies: Preliminary Results from the NCBO BioPortal Mapping Repository. In: Int'l Conf. on Biomedical Ontology (ICBO) (2009)
31. Pesquita, C., Faria, D., Santos, E., Couto, F.M.: To repair or not to repair: reconciling correctness and coherence in ontology reference alignments. In: Ontology Matching (OM) (2013)
32. Rosse, C., Mejino Jr., J.: A reference ontology for biomedical informatics: the Foundational Model of Anatomy. J. Biomed. Informatics 36(6), 478–500 (2003)
33. Salvadores, M., Alexander, P.R., Musen, M.A., Noy, N.F.: BioPortal as a dataset of linked biomedical ontologies and terminologies in RDF. Semantic Web 4(3), 277–284 (2013)
34. Santos, E., Faria, D., Pesquita, C., Couto, F.: Ontology alignment repair through modularization and confidence-based heuristics. arXiv:1307.5322 preprint (2013)
35. Santos, E., Faria, D., Pesquita, C., Couto, F.M.: Ontology alignment repair through modularization and confidence-based heuristics. CoRR abs/1307.5322 (2013)
36. Schlobach, S.: Debugging and semantic clarification by pinpointing. In: The Semantic Web: Research and Applications, pp. 226–240. Springer (2005)
37. Shvaiko, P., Euzenat, J.: Ontology matching: State of the art and future challenges. IEEE Trans. Knowledge and Data Eng. (2012)
38. Smith, B., Ashburner, M., Rosse, C., Bard, J., Bug, W., Ceusters, W., Goldberg, L.J., Eilbeck, K., Ireland, A., Mungall, C.J., Leontis, N., Rocca-Serra, P., Ruttenberg, A., Sansone, S.A., Scheuermann, R.H., Shah, N., Whetzel, P.L., Lewis, S.: The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. Nat Biotech 25(11) (2007)
39. Suntisrivaraporn, B., Qi, G., Ji, Q., Haase, P.: A modularization-based approach to finding all justifications for OWL DL entailments. In: Asian Sem. Web Conf. (ASWC) (2008)