ICBO: International Conference on Biomedical Ontology
July 24-26, 2009 ▪ Buffalo, New York, USA

Generating Homology Relationships by Alignment of Anatomical Ontologies

Frederic B. Bastian[1,2], Gilles Parmentier[1,2], Marc Robinson-Rechavi[1,2]
[1]Department of Ecology and Evolution, University of Lausanne, Switzerland;
[2]Swiss Institute of Bioinformatics, Lausanne, Switzerland

Abstract

The anatomy of model species is described in ontologies, which are used to standardize the annotations of experimental data, such as gene expression patterns. To compare such data between species, we aim to establish homology relations between ontologies describing different species. We present a new algorithm, and its implementation in the software Homolonto, to create new relationships between anatomical ontologies, based on the homology concept. These relationships and the Homolonto software are available at http://bgee.unil.ch/

Introduction

To be able to compare biological data, we need to use ontologies, to ensure that a biological concept is unambiguously associated to a unique identifier. To achieve this, ontologies such as the Gene Ontology[1] are increasingly used. Websites dedicated to model species also rely on the use of ontologies, for example the zebrafish anatomy for ZFIN[2], or the Mouse gross anatomy and development[3].

We are interested in integrating and comparing gene expression patterns between several species[4]. This raises the question of encoding corresponding information between ontologies which describe different anatomies (e.g. zebrafish and human). The most widely accepted criterion to make such comparisons in biology is homology[5]. Homology is classically defined as the relation between structures which derive from a same ancestral structure, although other definitions are discussed. It should be noted that the exact definition is up to the user, whose input will define which pairs of terms are defined as homologous.

To apply this concept in practice, hundreds of terms must be compared between ontologies. Although a purely manual annotation of homologies is possible, it would be too time consuming to be done for all terms between several divergent species. Kruger et al.[6] used a manual approach to find similarities between simplified anatomy ontologies for human and mouse. There is also an on-going effort to integrate anatomical ontologies, the Common Anatomy Reference Ontology project CARO[7].

Since the problem of aligning anatomical ontologies is to find correspondences between the concepts of two ontologies, we draw on methods from "schema matching", or "ontology alignment"[8,9]. Ontology alignment is the process of determining correspondences between ontology concepts. Usually, this technique is used to find the common concepts present in two ontologies. In the case of anatomical ontologies, the concepts to align are not strictly common, but rather, related: a homology relationship is not an equivalence relationship. For this reason, ontology alignment approaches developed for other applications (e.g. medicine oriented descriptions of human[9, 10]) cannot be applied as such: these methods would be misled by the existence of elements of same names and related to the same concept, but not homologous (e.g. eye of insects and of vertebrates), or reciprocally, homologous elements with different names (e.g. pectoral fin and upper limb).

We present here a new algorithm, and its implementation in the Java software Homolonto, to create new relationships between anatomical ontologies, based on the homology concept. Thus the basic aim of Homolonto is to propose in priority to the user the best candidate pairs of homologs, and avoid the need to consider many irrelevant pairs.

Homolonto Algorithm

1) *Computing word specific scores:* Score modifiers are computed for all words of the ontologies being aligned. Each word present at least once in both ontologies being aligned (O1 and O2) is given a score modifier based on its number of occurrences f(word, O):

$Mod(word, Oi) = 1/(1+\log10(f(word, Oi)))$ *eq. 1*

$Mod(word) = Mod(word, O1) * Mod(word, O2)$ *eq. 2*

2) *Starting list of propositions:* To initialize the algorithm we define first obvious similarities between the terms of the ontologies to align. Based on the assumption that two structures that have the same name are likely homologous, the initial propositions are formed of terms with identical names. In this process, we also consider the synonym field of the terms. Each pair of names n1, n2, is given a base score, dependent on the words shared:

$Base\_score(n1, n2) = base\_homonymy\_score * \max(Mod(word)) * |n1 \cap n2| / \max(|n1|, |n2|)$ *eq. 3*

Where |n| is the number of words in n, |n1∩n2| is the number of words shared by n1 and n2, and max(Mod(word)) is computed over all shared words. In the starting list, |n1∩n2| = |n1| = |n2| by definition, but this is not the case at further iterations of the algorithm.

3) *Initial propagation step:* The score of these propositions is propagated between neighbors. This initial propagation is bidirectional, and limited to already defined propositions. For example, the score of the "optic cup" pair is added to the score of the "eye" pair, as "optic cup" is part of "eye", and both pairs are initial propositions. Symmetrically the score of the "eye" pair is added to the "optic cup" pair. But the score of "eye" is not propagated to e.g. the pairing of "visual system" (ZFA[2] parent of "eye") with "sensory organ" (EHDAA[11, 12] parent of "eye"), because this pair is not an initial proposition. The aim of this step is to increase the score of the most likely homologs.

4) *Cleaning the initial proposition list:* The design of some ontologies may generate many false positives, typically through repetition of the same name as a child of diverse structures (e.g. 76 occurrences of "mesenchyme" in EHDAA). To avoid this, if a term is a member of several propositions with different scores, we initially keep only the best scoring proposition. If there are more than 5 highest scoring propositions for a given term, we remove all propositions for this term.

5) *Evaluation step:* Each proposition is presented to the user, in descending order of scores. The user has to validate, invalidate, or delay decision regarding the proposed homology.

6) *Computation step:* If one of the terms of a validated pair is already a member of an homology group, then the other term is added to the homology group. Otherwise, a new homology group is created, containing both terms of the validated pair. The information of homology is propagated through the hierarchy by the use of a validated homology score (eq. 4). The underlying idea is that if two terms A and B are homologous, then one of the children of A is probably homologous to one of the children of B. During the propagation the validated homology score is added to the base score (eq. 3) of pairs of terms:

$$\text{Propagated\_score}(a, b) = \text{validated\_homology\_score} * (\text{max\_depth} + 1 - \text{present\_depth}) / (\text{max\_depth} + 1) \quad \textit{eq. 4}$$

$$\text{Total\_score}(a, b) = \text{Propagated\_score}(a, b) + \text{Base\_score}(n_a, n_b) \quad \textit{eq. 5a}$$

Where n_a is the name of term a. In the present implementation, the max_depth is 1, and the validated homology score is 1.5 times the base homonymy score. For pairs of terms which are not yet a proposition, a new proposition is created, and the base score is computed. This will include cases of partial homonymy, for which eq. 3 down weights names which share a lower proportion of words. Pairs which have been previously invalidated by the user will not receive a propagated score, and will remain invalidated.

To down weight potential false positives due to validation of terms with many children, the propagated score is reduced proportionally to the number of new propositions for each term of the ontology to align (eq. 5b).

$$\text{Total\_score}(a, b_i) = \text{Propagated\_score}(a, b_i) / ((|b| + 1) * 2) + \text{Base\_score}(n_a, n_{bi}) \quad \textit{eq. 5b}$$

Where a is a term of the ontology to align, b_i is a term of the reference ontology, and |b| is the number of new propositions for term a. When a proposition (a, b_i) is invalidated, |b| is updated, and the Total score(a, b_i) increases for the remaining propositions.

When the terms of an invalidated proposition share common words, then the score modifiers of all shared words is diminished (eq. 6). As this is repeated, words which tend to generate false positives will be increasingly down weighted.

$$\text{Mod'(word)} = \text{Mod(word)} * 0.9 \quad \textit{eq. 6}$$

7) *Iteration:* Evaluation of propositions (step 5), ordered by total score (base score + propagated score), and computation (step 6), is repeated until the user decides to terminate, or no more propositions are generated.

Homolonto Results

Homolonto has been used to align six anatomical ontologies to date, representing four vertebrate species (human and mouse have different ontologies for adult and embryonic stages). We will present more in detail two alignments: zebrafish (ZFA ontology[2]) / Xenopus (XAO ontology[13]), which illustrates a best case scenario of two recently updated ontologies, conforming to the CARO standards[7], with annotations of synonyms and definitions, and low redundancy. And human (EHDAA ontology[11, 12]) / mouse (EMAPA ontology[11, 3]) which, despite the similarity in anatomy, illustrates a more difficult scenario of large ontologies, with issues such as repetition of names (76 occurrences of "mesenchyme" in human, 93 in mouse), due to splitting of concepts among morphological structures or among developmental stages.

The main observation is that our algorithm is successful at ordering propositions. In the "easy" case of zebrafish / Xenopus, there are only seven invalidated propositions in the first 150 (95% validation). This is followed by a relatively short interval of iterations where validated and invalidated propositions are mixed: 46% of validations between iterations 151 and 200, and 20% between 201 and 250. Further iterations generate mostly invalidated propositions (3% validation from 251 to 735). Thus 93% of all validations occurred in the first 250 iterations.

The pattern is similar for the human / mouse alignment. In the first 1400 iterations, 99% of propositions are validated. In the next 600 iterations, the figure reduces to 63%, and in the last 962 iterations it falls to 21%. This slower decrease illustrates the complexity of this alignment. The validation rate of 66% shows that the propositions were mostly worth considering, and that the high number of propositions was due indeed to the size of the ontologies, not to a default in the algorithm. Results also show that manual expertise is necessary, since even in the high scoring propositions some are invalid. Overall, 27% of invalidations are pairs of terms with identical names. Interestingly, Homolonto manages to give these misleading homonyms low priority: homonyms within the first 1000 iterations have a 99% chance of being homologs, whereas homonyms within the last 1000 iterations only have a 19% chance of being homologs. Thus 93% of invalidated homonyms appear after iteration 1400.

Generating Relationships between Groups of Homologs

Homolonto is used to generate pairwise homology relationships between anatomical ontologies. As homology relationships are transitive, these pairwise alignments can be merged into homologous organs groups (HOGs). Homolonto thus generates HOGs, and mapping of species-specific anatomical structures to these HOGs. HOGs then need to be structured as an ontology to allow reasoning on them. This means that, at a minimum, relationships amongst them have to be designed. Another algorithm has thus been developed to infer relationships between HOGs.

1) *Initial Step:* all possible paths between HOGs are retrieved. For instance, if an anatomical structure "a", mapped to the HOG "A", has a *part_of* relationship to the anatomical structure "b", mapped to the HOG "B", then a putative *part_of* relationship is defined between HOGs "A" and "B".

Relationships between HOGs are often indirect (e.g. structure "a", mapped to HOG "A", *part_of* structure "c", *part_of* structure "b", mapped to HOG "B"). If the first relation (the relation "outgoing" from the child HOG, "A" in the previous example) and the last relation (the relation “incoming” to the parent HOG, "B" in the previous example) are of the same type (e.g. *part_of*, *is_a*), then the putative relationship is defined as this type. Otherwise, the relationship is defined as the SKOS[14] type *broader_than*.

2) *Skipping relations from not-trusted ontologies:* some ontologies do not follow the OBO principles, and implement for instance only one type of relation amongst all concepts (e.g. EV[15] only uses *is_a* relationships). The user may choose to not use these ontologies to define relation types. All the putative relations inferred by these ontologies at step 1 are then set as *broader_than*. But the final relation type between these HOGs can still be inferred thanks to other ontologies.

3) *Skipping relations defined by too few species:* if the proportion of species defining a relation, compared to the total number of species involved in the creation of the HOGs, is below a threshold defined by the user ("species coverage"), then the relation is defined to the type *broader_than*, and the algorithm stops examining relations between these HOGs. Indeed, in such case, inferred relation types may not be trusted.

4) *Defining within-ontology agreement*: several anatomical structures from the same ontology can belong to the same HOG. This can generate a within-ontology conflict for defining a relation type. For instance, structures "a" and "b" allow to define a putative *part_of* relationship between HOGs "A" and "B", while structures "a’" and "b’", belonging to the same ontology, define a putative *is_a* relationship between these HOGs. The algorithm then calculates, for each relation type, the proportion that the number of paths defining this relation type represents, compared to the total number of paths between these two HOGs for this ontology. If, for a type, this proportion exceeds a threshold ("within-ontology agreement"), defined by the user and at least greater than 0.5, then this relation type is attributed for this species between these HOGs. Otherwise, the relation is defined to the type *broader_than* for this ontology.

5) *Defining inter-ontology agreement:* different ontologies can define different relation types between two related HOGs. This conflict is resolved in the same way as at step 4, by using a threshold ("inter-ontology agreement"), defined by the user and at least greater than 0.5.

6) *Removing cyclic relationships:* by inferring automatically the relationships between HOGs, cycles

may be generated (e.g. HOG "A" *part_of* HOG "B" *part_of* HOG "A"), whereas the ontology has to be acyclic. If such cycles are detected, the algorithm stops with an error message prompting the user to make a decision: the user has then to manually remove one of the involved relationships.

7) *Removing redundancies:* if several relationships are redundant, only the deepest relationship is conserved; for instance, if a HOG "A" has two substructures by a *part_of* relationship, "B" and "C", and if "C" is also a substructure of "B", then the direct relationship between the HOGs "A" and "C" is removed.

8) *Curation step:* a curator has then to manually review all the *broader_than* relations, to attribute them to a type defined by the OBO Relation Ontology[16]. Some custom relationships, not inferred by the algorithm, can also be added at this step.

Conclusion

To date, the use of Homolonto, followed by a curation process, allowed to define 1004 HOGs, involving 4088 structures from 6 anatomical ontologies (ZFA[2], EHDAA[11, 12], EV[15], EMAPA[11, 12], MA[17], and XAO[13]).

The algorithm to design relationships amongst the HOGs inferred 1188 relations. With the more stringent parameters (species coverage = 1, within-ontology agreement = 1, inter-ontology agreement = 1), 341 of them are defined as *part_of*, all the others as *broader_than*. The curation step to review these *broader_than* relations is currently under process.

The HOG ontology has been successfully implemented into Bgee[4], a database for studying gene expression evolution, and already allows to perform automated, cross-species, gene expression pattern comparisons.

The Homolonto software and source code, and the HOG ontology, are available from the download section of the Bgee website (http://bgee.unil.ch). The algorithm to generate relationships between groups of homologs will be available soon.

References

1. Gene Ontology Consortium. The Gene Ontology (GO) project in 2006. Nucleic Acids Res 2006; 34: D322–6.
2. Sprague J, *et al*. The Zebrafish Information Network: The zebrafish model organism database. Nucleic acids research 2006; 34: D581–5.
3. Baldock RA, *et al*. EMAP and EMAGE: A framework for understanding spatially organized data. Neuroinformatics 2003; 1: 309–25.
4. Bastian F, *et al*. Bgee: Integrating and Comparing Heterogeneous Transcriptome Data Among Species. LNBI Springer 2008; 124–31.
5. Hall B. Homology: The Hierarchical Basis of Comparative Biology. Academic Press 1994.
6. Kruger A, *et al*. Simplified ontologies allowing comparison of developmental mammalian gene expression. Genome Biol 2007; 8: R229.
7. Haendel MA, *et al*. CARO – The Common Anatomy Reference Ontology. Springer 2008; 327–49.
8. Euzenat J and Shvaiko P. Ontology Matching. Springer Verlag 2007.
9. Lambrix P and He T. Ontology alignment and merging. Springer 2008; 133–49.
10. Mork P and Bernstein PA. Adapting a Generic Match Algorithm to Align Ontologies of Human Anatomy. IEEE 2004.
11. Aitken S. Formalizing concepts of species, sex and developmental stage in anatomical ontologies. Bioinformatics 2005; 21: 2773–79.
12. Hunter A, *et al*. An ontology of human developmental anatomy. Journal of anatomy 2003; 203: 347–55.
13. Bowes JB, *et al*. Xenbase: A Xenopus biology and genomics resource. Nucleic Acids Res 2008; 36: D761–7.
14. Miles A and Brickley D. Simple Knowledge Organisation System (SKOS). Aug 2008; http://www.w3.org/TR/2008/WD-skos-reference-20080829/.
15. Kelso J, *et al*. eVOC: A controlled vocabulary for unifying gene expression data. Genome Res 2003; 13: 1222–30.
16. Smith B, *et al*. Relations in biomedical ontologies. Genome Biol 2005; 6: R46.
17. Smith CM, *et al*. The Mouse Gene Expression Database (GXD): 2007 update. Nucleic Acids Res 2007; 35: D618–23.