RIMBO - An Ontology for Model Revision Databases

Filip Kronström[1](✉), Alexander H. Gower[1], Ievgeniia A. Tiukova[1,2], and Ross D. King[1,3,4]

[1] Chalmers University of Technology, Gothenburg, Sweden
filipkro@chalmers.se
[2] KTH Royal Institute of Technology, Stockholm, Sweden
[3] University of Cambridge, Cambridge, UK
[4] Alan Turing Institute, London, UK

Abstract. The use of computational models is growing throughout most scientific domains. The increased complexity of such models, as well as the increased automation of scientific research, imply that model revisions need to be systematically recorded. We present RIMBO (Revisions for Improvements of Models in Biology Ontology), which describes the changes made to computational biology models.

The ontology is intended as the foundation of a database containing and describing iterative improvements to models. By recording high level information, such as modelled phenomena, and model type, using controlled vocabularies from widely used ontologies, the same database can be used for different model types. The database aims to describe the evolution of models by recording chains of changes to them. To make this evolution transparent, emphasise has been put on recording the reasons, and descriptions, of the changes.

We demonstrate the usefulness of a database based on this ontology by modelling the update from version 8.4.1 to 8.4.2 of the genome-scale metabolic model Yeast8, a modification proposed by an abduction algorithm, as well as thousands of simulated revisions. This results in a database demonstrating that revisions can successfully be modelled in a semantically meaningful and storage efficient way. We believe such a database is necessary for performing automated model improvement at scale in systems biology, as well as being a useful tool to increase the openness and traceability for model development. With minor modifications the ontology can also be used in other scientific domains.

The ontology is made available at https://github.com/filipkro/rimbo and will be continually updated.

Keywords: Ontology · Knowledge representation · Database · Computational biology · Semantic web

1 Introduction

Computational models play a crucial part in our understanding of complex biological systems [1], and the further improvements of such models have been described as a grand challenge for the 21st century [2].

A. Bifet et al. (Eds.): DS 2023, LNAI 14276, pp. 523–534, 2023.
https://doi.org/10.1007/978-3-031-45275-8_35

A promising way forward, enabled by advances in AI and automation, is to develop autonomous laboratories performing experiments and discovering knowledge. This has been demonstrated by robot scientists performing cycles of experiments to determine gene functions in yeast [3], discover drugs [4], and optimise cell culturing conditions [5]. Largely automated pipelines has been used to optimise strain engineering in both *Saccharomyces cerevisiae* and *Escherichia coli* [6], and a mobile robotic chemist has searched for photocatalysts for hydrogen production [7].

Using computational models to guide experiment design and the experimental results to improve the models in a closed loop manner has proved to be a successful and scaleable way of developing systems biology models [8].

For an AI agent to be able to autonomously reason about improvements to a model, a structured and semantically unambiguous way of storing models is required. Critically, such a store should also handle large numbers of revisions to the models. A semantically meaningful representation of these revisions will enable human researchers to gain insights from the model improvement cycles, access and use the models, as well as facilitating for computer systems to reason about previous changes to models.

Across many domains the importance and use of computational models, as well as the number of models available, is increasing [9]. No matter if the model is used in science or any other field, it needs the trust of a wider community. One way this can be achieved is by making the steps taken during development more open and transparent, regardless if it was done by humans or machines.

In this work we propose an ontology, capturing and explaining changes to different types of computational biology models, which can also be used for a model revision database. Such a database is important in an automated scientific discovery setting, where we seek improvements to computational models. We also demonstrate how this ontology can be used to model community consensus updates, as well as machine generated hypotheses about improvements for yeast metabolic models.

2 Background and Related Work

There are several repositories or databases where the computational biology community share models today. Most notably BioModels [10] with over 2 000 submitted models of different types, but also BiGG Models [11] with genome-scale metabolic models (GEMs), and the CellML Model Repository [12]. Although some repositories support version control (e.g. BioModels), they are not designed to deal with the large numbers of small revisions generated when developing and refining models.

Central to the increased sharing and reuse of computational models, and other biological information, are common and unambiguous model descriptions. Biological Pathway Exchange (BioPax) [13] is a language for exchange and integration of biological pathways. For computational models, CellML [14], and the Systems Biology Markup Language (SBML) [15] are widely used. They both enable the databases mentioned above, with BioModels and BiGG containing large amounts of SBML models and the CellML Model Repository naturally being for CellML models. Although slightly different, the three model formats are all XML-based.

These three formats share a heavy reliance on ontologies. Ontologies and controlled vocabularies provide semantic meaning to data, both to humans and machines. The Gene Ontology (GO) [16], provides structure and semantics to genes and gene products across species. The Systems Biology Ontology (SBO) [17] is closely tied to SBML, and contains vocabularies useful for computational modelling and systems biology, and the Kinetic Simulation Algorithm Ontology (KiSAO) [18] complements it with additional terms describing simulation and algorithms. Cell types and processes in cells can be found in the Cell Ontology (CL) [19] and the Ascomycete Phenotype Ontology (APO) [20] contains phenotypes for Ascomycete fungi. The EDAM ontologies [21,22] have vocabularies for data management and analysis. Provenance models, describing the provenance of both scientific experiments and general processes, has been encoded in ontologies like PROV-O [23] and REPRODUCE-ME [24].

The COMODI (COmputational MOdels DIffer) ontology [25] attempted to characterise changes to computational models in XML format. Changes to ”XmlEntities” were identified along with ”Reasons”, ”Intentions”, and ”Targets” for them. Such annotations provide very detailed descriptions of each change to the XML tree, which is helpful when studying single updates or for understanding the mechanics of the format the model is encoded in. However, this verbosity is not helpful when chains of revisions are studied. Instead of providing a detailed description of all changes to the encoding of the model, we argue that overarching intentions are important. There are also differences between describing a change, and providing an unambiguous and storage efficient patch that can be used to recreate the actual file, without storing a copy of it.

Apart from just offering semantic meaning, ontologies can also effectively be used as the schema for databases. By modelling data as Resource Description Framework (RDF) triples (*subject*, *predicate*, *object*) using terms from ontologies, knowledge graphs can be created. Such graphs can be queried or reasoned over, and have previously acted as the knowledge base for closed loop model improvements [8].

3 Results

We propose that model revisions are represented using the Revisions for Improvements of Models in Biology Ontology (RIMBO). It is designed to be the schema for a graph database containing iteratively improved computational biology models. In theory it can be used with any type of model, but we have focused on models that are improved by making small changes which can be described in a semantically meaningful way. Below, a non exhaustive list can be seen, illustrating the type of competency questions we want such a database to answer.

- Which model was introduced in publication p?
- Which models are derived from model M_1?
- What was the reason for the revision R?
- Which revision tried to correct the predicted essentiality of gene g in model M_2?
- Which model is a revision of model M_3, where the change affects reaction r?

The ontology is expressed in OWL2 and developed in Protégé (v. 5.5.0, https://protege.stanford.edu/). We will first, in Sect. 3.1, describe the ontology and then, in Sect. 3.2, show examples of model revisions in this format and demonstrate it can be used as a database for large numbers of revisions.

3.1 Description of RIMBO

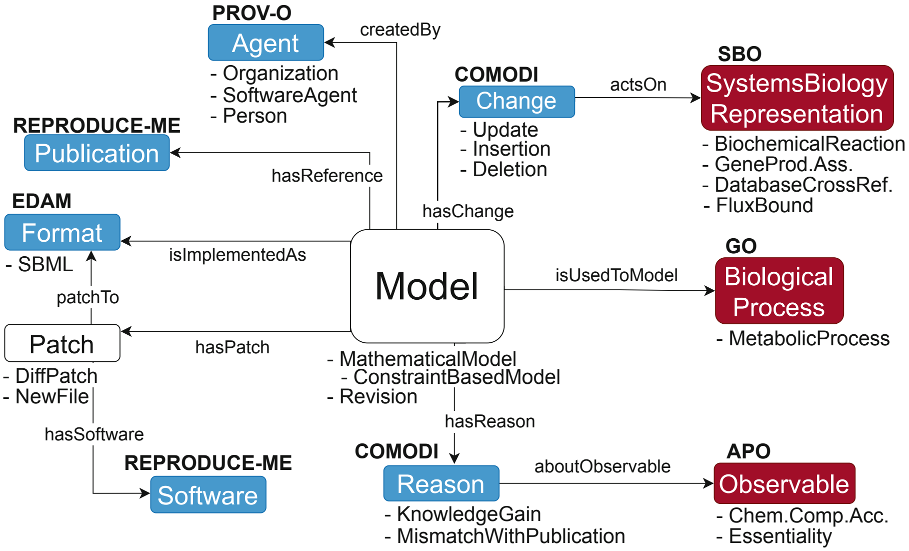


Fig. 1. Overview of RIMBO showing classes, how they are connected, and which ontologies they are from. The text under the boxes specifies subclasses used for the demonstration in Sect. 3.2. Red boxes denotes domain specific classes that would need replacing if the ontology is applied to another domain. Blue denotes classes from other foundational scientific ontologies and the white boxes are classes introduced in RIMBO. (Color figure online)

RIMBO combines classes from different ontologies and an overview illustrating how this is done can be seen in Fig. 1. To connect these classes the relations described in Table 1, along with their domains and ranges, are introduced.

The central class in RIMBO is `Model`, being a superclass to different modelling types, imported from ontologies such as the Mathematical Modelling

Ontology (MAMO) and EDAM. Information about this model is provided through links to other concepts. For example, BiologicalProcess classes from GO or CL can describe which phenomenon is being modelled, and terms from REPRODUCE-ME and PROV-O specify important metainformation, such as when and by whom it was created, as well as links to relevant publications. The model is also linked to the model file, represented by an instance of its corresponding Format class from EDAM. This connects either to an external reference to a filestore or an online resource, or a representation of the file in the graph. There are advantages and disadvantages to each option. External references require maintenance to ensure they point to correct locations, but are more storage-efficient. Having large files in the graph may affect query performance.

Table 1. The relations used to model revisions with RIMBO, along with domains and ranges when applicable. The namespaces specify which ontology the classes are from, when no namespace is specified the term is introduced in RIMBO. rep-me is short for REPRODUCE-ME.

Relation	Domain	Range	Description
aboutObservable	comodi: Reason	apo: Observable	Describes the Reason by linking it to Observable terms from APO.
actsOn	comodi: Change	sbo: Sys.Bio.Repr.	Describes what part of the model is affected by the Change.
createdBy	Model	prov: Agent	Specifies who created a Model (Organization, Person ⊑ Agent).
hasChange	Revision	comodi: Change	Connects the Revision to a Change.
hasPatch	Revision	Patch	Connects the Revision to a Patch.
hasReason	Revision	comodi: Reason	Connects the Revision to a Reason.
hasReference		rep-me: Publication	Links, e.g., a Model or Mism.W.Pub. to its Publication.
hasSoftware		rep-me: Software	Specifies software, e.g., used to find the diff-patch.
isImplementedAs	Model	edam: Format	Links the Model to Format with information about the model file.
isUsedToModel	Model		Describes which phenomenon is modelled.
ofMaterialEntity	apo: Observable	Mat.Entity	Links Observable to Mat.Entity (Gene, Chem.Entity ⊑ Mat.Entity).
patchTo	Patch	edam: Format	Links a Patch to the model file it applies to.
revisionTo	Revision	Model	Links the Revision to the Model it was revised from.

The other central class in this ontology is Revision, which is also a subclass of Model, and describes a modified version of a Model. An important thing to note is that the Revision class is not disjoint with classes describing the model type, for example MathematicalModel classes from MAMO. Hence, a revised model is described as the intersection of its model type and a Revision. Recording the reason along with descriptions of the changes made to models is important, both when improvements are generated by humans and machines. For a human generated revision, it can, for example, be used as a way of documenting the research. For a machine, it enables the system to reason about the effect of previous changes, as well as providing a way of communicating and motivating its findings with human researchers. The `Reason` class is from COMODI and has subclasses such as `MismatchWithPublication` and `KnowledgeGain`. Linking this to terms from ontologies like APO and relevant genes or chemicals gives a description of the cause of a change. As one revision might be made up of several changes, such as the addition of multiple new reactions, it is described by a `Change` collecting, possibly several, instances of `Deletions`, `Insertions`, or `Updates`, all from the COMODI ontology. The change can be described by linking these classes to subclasses of `SystemsBiologyRepresentation` from SBO and specific reactions or genes.

The actual change to the file is saved using the `Patch` class, with subclasses `DiffPatch` and `NewFile`. As iterative changes often are small, in terms of the actual changes to the files, it makes sense to just store the differences between the two files to the database. This is done with the `DiffPatch` class along with information on what software was used to find it. In some cases it might be desirable to just store a new version of the model file, for instance for binary model representations, for larger changes, or to avoid lengthy chains of patches. This is done using the `NewFile` class.

3.2 Demonstration

To demonstrate the usefulness of this ontology and a resulting database, we have generated a demonstration knowledge graph with model revisions. This example is based on revisions to the genome-scale metabolic model (GEM) Yeast8 [26] for the yeast species *Saccharomyces cerevisiae*. A GEM is a network collecting information about, for example, genes and reactions in a biological system. First, we model a part of a community update of Yeast8, from v8.4.1 to v8.4.2. Then, by expressing the model in first-order logic, an algorithm using abductive reasoning, LGEM$^+$ [27], was used to suggest modifications to the theory. Finally, we perform 31 400 random revisions.

The first update, from v8.4.1 to v8.4.2, was about improving the simulation of alcoholic fermentation conditions by adding several fatty acid ester producing reactions. The modification suggested by LGEM$^+$ was to remove the gene YJL130C as a requirement for an enzyme catalysing the reaction carbamoyl-phosphate synthase (glutamine-hydrolysing). This was suggested as a remedy to YJL130C being predicted as essential for growth, when empirical evidence showed it was not [28]. Finally, starting with this version, random revisions were generated by iteratively either removing a reaction, modifying the gene requirements for a reaction, or modifying the flux bounds for a reaction.

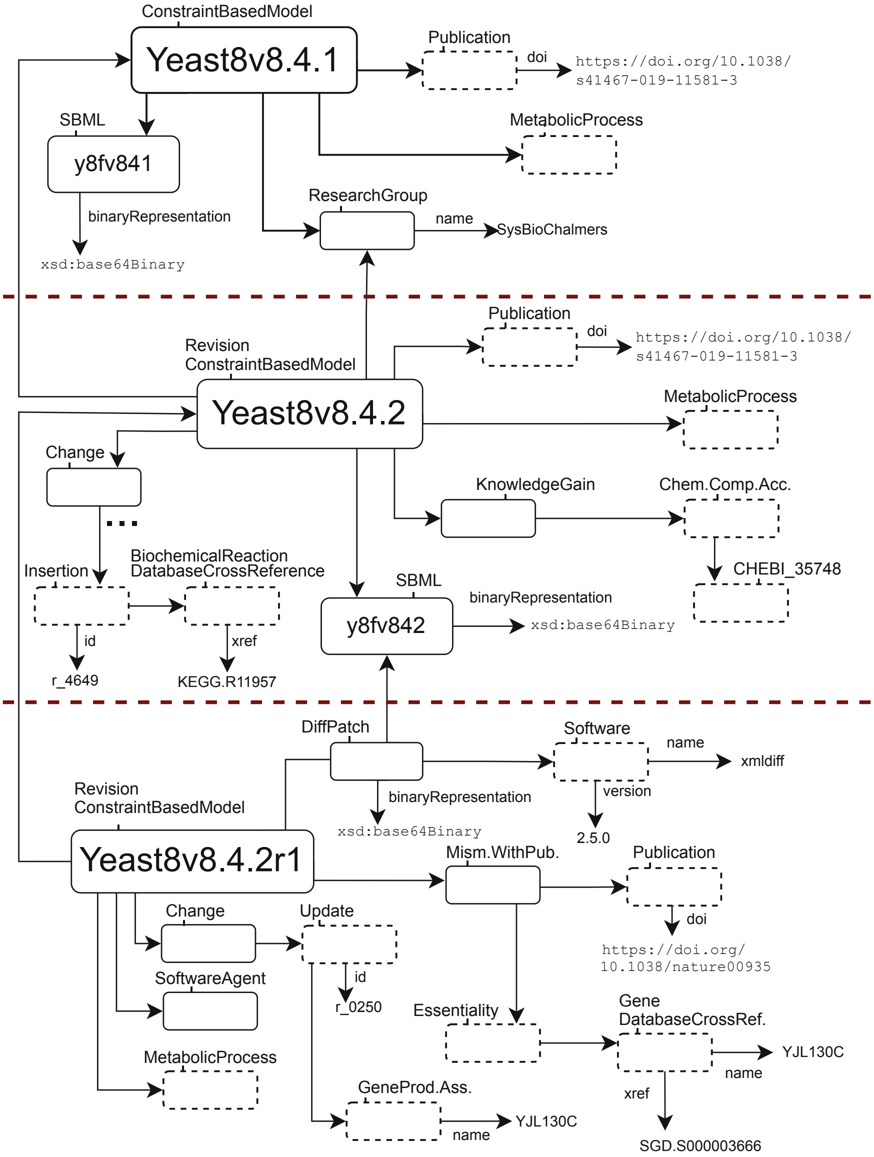


Fig. 2. The knowledge graph containing the base model, Yeast8v.8.4.1, the update to v8.4.2, and the revision changing the gene reaction rule for reaction r_0250, described in Sect. 3.2. The boxes are instances of the classes named above them, solid boxes represent named nodes and dashed correspond to blank nodes. The dashed red lines separates entries belonging to the different models/revisions. (Color figure online)

The knowledge graph with the first two revisions can be seen in Fig. 2, where the base model, Yeast8 v8.4.1, is added as an instance of a `ConstraintBasedModel` modelling a `MetabolicProcess`. It is linked to the

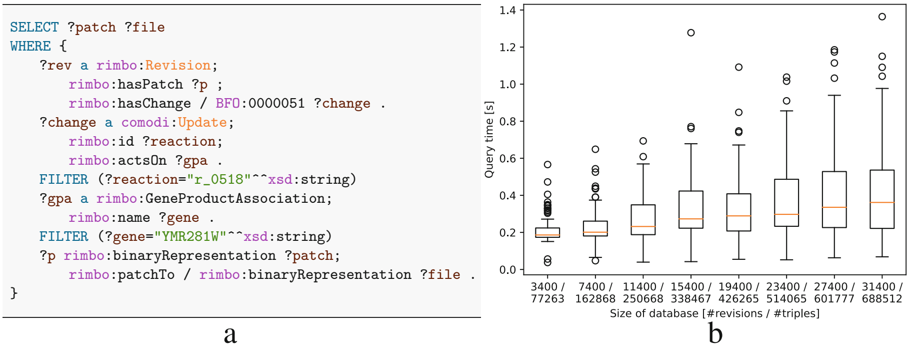
ResearchGroup "SysBioChalmers", who are maintaining the model on Github, as well as the corresponding Publication by Lu et al. [26]. The model file itself is represented as an instance of the SBML format which links to a compressed copy of the original model file as a literal of type xsd:byte64Binary.

Yeast8 v8.4.2 is still a ConstraintBasedModel, but also a Revision, meaning this entry is the intersection of the two classes. The reason for the this revision is modelled as a KnowledgeGain about ChemicalCompoundAccumulation of CHEBI_35748 (fatty acid ester) and it is described by Insertions of BiochemicalReactions and TransportReactions with references to the KEGG Reaction database. As the number of changes to the model file, going from v8.4.1 to v8.4.2, is rather large, we save a compressed copy of the entire file, represented as a NewFile, linked with a new instance of SBML.

The reason for the change from the abduction algorithm is contradicting results in a publication. Hence, it is modelled as a MismatchWithPublication referring to a Publication representing the work by Giaever et al. [28], as well as the predicted Essentiality of the gene, "YJL130C". The revision is described as an Update of the reaction r_0250's GeneProductAssociation associated to the aforementioned gene. Unlike the previous models, this iteration was not generated by "SysBioChalmers", instead it is linked to a SoftwareAgent referring to LGEM$^+$. This time the model file is represented by the difference to Yeast8v8.4.2. An instance of DiffPatch links a literal of the type xsd:base64Binary, containing the patch recreating the updated model, to the revision and the previous model file. The software and version, xmldiff, v2.5.0 (https://xmldiff.readthedocs.io/), used to find the patch is specified using the Software class.

To demonstrate that a database using this ontology can handle large numbers of revisions, chains of thousands of modified models were added, along with metainformation describing the change and who made it. The modifications of the models were performed using COBRApy (v.0.26.3, https://cobrapy.readthedocs.io/). When altering the gene-reaction rule a randomly picked gene was either removed or added to the rule1of a random reaction. For the flux-bound modifications either the upper or the lower bound for some reaction was updated randomly such that it still is valid. Removing a reaction was done by chosing a random reaction to delete from the model. The different actions were not picked uniformly to better reflect real revisions, resulting in 25 793 modified gene-reaction rules, 3 857 altered flux bounds, and 1 750 removed reactions. In our implementation of the database a copy of every 100th model file was saved to reduce the sizes of the patches stored for every revision. The knowledge graph with 31 400 revisions contains 688 512 triples and the size of it, serialised as a .ttl-file, is 1.17 GB (as a reference, one uncompressed Yeast8 file in SBML format has a file size of ~10 MB).

To validate the database, iterations containing more and more data were deployed on an Apache Jena Fuseki server running on a 2021 MacBook Pro M1. The growing database was queried for the binary patch, along with the file it should be applied to, belonging to revisions updating the gene-reaction rules of specific reactions. Figure 3a shows an example of the queries executed in the experiment where the gene-reaction pairs were varied. In Fig. 3b a box-plot of the query times is shown, based on 100 queries for each database. All pairs were present in every iteration of the databases and the same series of queries were executed between the iterations. The query times increase with growing database size, but the majority of the queries show a rather small increase. The major difference between the different database iterations is the worst case queries, which is primarily explained by the number of results retrieved. The gene-reaction pairs are not necessarily unique and with a bigger database we can expect more duplicates.



```
SELECT ?patch ?file
WHERE {
    ?rev a rimbo:Revision;
        rimbo:hasPatch ?p ;
        rimbo:hasChange / BFO:0000051 ?change .
    ?change a comodi:Update;
        rimbo:id ?reaction;
        rimbo:actsOn ?gpa .
    FILTER (?reaction="r_0518"^^xsd:string)
    ?gpa a rimbo:GeneProductAssociation;
        rimbo:name ?gene .
    FILTER (?gene="YMR281W"^^xsd:string)
    ?p rimbo:binaryRepresentation ?patch;
        rimbo:patchTo / rimbo:binaryRepresentation ?file .
}
```

Fig. 3. (a) shows a query retrieving the patch and the file it applies to for a revision where a modification, involving the gene YPL280W, of the gene-reaction rule for reaction r_4133 is performed. This type of queries, but with varying gene-reaction pairs were used to generate (b), showing a box-plot of query times from 100 queries for databases of different size, deployed on an Apache Jena Fuseki server.

4 Discussion and Conclusion

In this work we demonstrate the usefulness of a structured and semantically sound representation for computational models, not only for sharing with the community, but also during the development process. We view RIMBO as a complement to public model repositories, such as BioModels, BiGG Models, and CellML Model Repository, providing structure and transparency to model development. One could envision revision traces, expressed in controlled vocabularies, describing the provenance published along with new models. We argue this would be useful both for automated and traditional labs, as it could greatly increase the openness and traceability of research. Along with this, RIMBO also works as a useful tool to organise the models during development in a storage efficient manner.

The ontology based graph structure allows for flexibility in the implementation of a database. For example, what level of detail to use when explaining a change might vary depending on the needs of the specific lab, and what kind of model is revised. One might be interested in recording more fine-grained descriptions of a change for a more specific model. Sometimes it could also be useful to describe the actual changes to the XML-tree using the COMODI ontology. Depending on the domain and what kind of changes are made, there might also be a need to introduce new terms to describe the revision. APO and SBO, along with some new classes describing terms connected to the SBML Level 3 Flux Balance Constraints package covers our current needs, working with yeast systems biology, but other domains most likely need other, domain specific, classes.

A planned future extension and generalisation of this work, interesting for both traditional and autonomous labs, is to also model and record hypotheses, e.g., generating the revised model. This would build on previous work attempting to formalise scientific discovery, such as the HELO ontology [29] and be a way of connecting improvements to computational models with experimental data and back to new biological knowledge. For this, information about how to test and evaluate hypotheses should be described, such as unambiguous instructions on what simulations to run and which data to compare the results to. Currently, RIMBO is, to some extent, aligned with PROV-O. With this extension more work is needed to align it with an upper level ontology, such as the Basic Formal Ontology (BFO) [30], to easier interface with ontologies describing for example experimental data.

As with computational models, ontologies change. As we use RIMBO to represent models and revisions to models in our project, it will be continuously developed and new releases will be published here: https://github.com/filipkro/rimbo.

Although this work is focused on computational biology models, the techniques and ideas presented are not domain specific. As the iterative nature of new knowledge gain is common for most fields, we think the approach of recording smaller changes to models, no matter if the improvements have been found by humans or machines, along with reasons and intentions for changes can be useful in many scientific fields.

5 Code and Availability

The code and knowledge graph for the demonstration is available here: https://github.com/filipkro/rimbo-demo. The ontology and future updates of it, is available here: https://github.com/filipkro/rimbo.

Acknowledgements. We want to thank the rest of the Ross King Group at Chalmers University for their thoughtful insights and discussions. This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Alice Wallenberg Foundation. Funding was also provided by the Chalmers AI Research Centre, the UK Engineering and Physical Sciences Research Council (EPSRC) grant nos: EP/R022925/2 and EP/W004801/1, as well as the Swedish Research Council Formas (2020-01690).

References

1. Noble, D.: The rise of computational biology. Nat. Rev. Mol. Cell Biol. **3**(6), 459–463 (2002)
2. Omenn, G.S.: Grand challenges and great opportunities in science, technology, and public policy. Science **314**(5806), 1696–1704 (2006)
3. King, R.D., et al.: Functional genomic hypothesis generation and experimentation by a robot scientist. Nature **427**(6971), 247–252 (2004)
4. Williams, K., et al.: Cheaper faster drug development validated by the repositioning of drugs against neglected tropical diseases. J. Roy. Soc. Interface **12**(104), 20141289 (2015)
5. Kanda, G.N., et al.: Robotic search for optimal cell culture in regenerative medicine. eLife **11**, e77007 (2022)
6. Singh, A.H., et al.: An automated scientist to design and optimize microbial strains for the industrial production of small molecules (2023)
7. Burger, B., et al.: A mobile robotic chemist. Nature **583**(7815), 237–241 (2020)
8. Coutant, A., et al.: Closed-loop cycles of experiment design, execution, and learning accelerate systems biology model development in yeast. Proc. Natl. Acad. Sci. **116**(36), 18142–18147 (2019)
9. Barton, C.M., et al.: How to make models more useful. Proc. Natl. Acad. Sci. **119**(35), e2202112119 (2022)
10. Malik-Sheriff, R.S., et al.: BioModels-15 years of sharing computational models in life science. Nucleic Acids Res. **48**, D407–D415 (2020)
11. King, Z.A., et al.: BiGG models: a platform for integrating, standardizing and sharing genome-scale models. Nucleic Acids Res. **44**, D515–D522 (2016)
12. Lloyd, C.M., et al.: The CellML model repository. Bioinformatics **24**(18), 2122–2123 (2008)
13. Demir, E., et al.: The BioPAX community standard for pathway data sharing. Nat. Biotechnol. **28**(9), 935–942 (2010)
14. Lloyd, C.M., et al.: CellML: its future, present and past. Prog. Biophys. Mol. Biol. **85**(2), 433–450 (2004)
15. Hucka, M., et al.: The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics **19**(4), 524–531 (2003)
16. Ashburner, M., et al.: Gene ontology: tool for the unification of biology. Nature Genet. **25**(1), 25–29 (2000)
17. Juty, N., le Novère, N.: Systems biology ontology. In: Dubitzky, W., et al. (eds.) Encyclopedia of Systems Biology, pp. 2063–2063. Springer, New York (2013). https://doi.org/10.1007/978-1-4419-9863-7_1287
18. Zhukova, A., et al.: Kinetic simulation algorithm ontology. Nat. Proc. (2011)
19. Diehl, A.D., et al.: The cell ontology 2016: enhanced content, modularization, and ontology interoperability. J. Biomed. Semant. **7**(1), 44 (2016)
20. Costanzo, M.C., et al.: New mutant phenotype data curation system in the saccharomyces genome database. Database J. Biol. Databases Curation **2009**, bap001 (2009)
21. Black, M., et al.: EDAM: the bioscientific data analysis ontology (update 2021). F1000Research, vol. 11 (2022)
22. Kalaš, M., et al.: EDAM-bioimaging: the ontology of bioimage informatics operations, topics, data, and formats (2019 update) [version 1; not peer reviewed]. F1000Research, vol. 8(ELIXIR), p. 158 (2019)

23. Lebo, T., et al.: PROV-o: the PROV ontology. Technical report, World Wide Web Consortium (2013)
24. Samuel, S., König-Ries, B.: End-to-end provenance representation for the understandability and reproducibility of scientific experiments using a semantic approach. J. Biomed. Semant. **13**(1), 1 (2022)
25. Scharm, M., et al.: COMODI: an ontology to characterise differences in versions of computational models in biology. J. Biomed. Semant. **7**(1), 46 (2016)
26. Lu, H., et al.: A consensus s. cerevisiae metabolic model yeast8 and its ecosystem for comprehensively probing cellular metabolism. Nat. Commun. **10**(1), 3586 (2019)
27. Gower, A.H., et al.: LGEM$^+$: a first-order logic framework for automated improvement of metabolic network models through abduction. arXiv, arXiv:2306.06065 (2023)
28. Giaever, G., et al.: Functional profiling of the saccharomyces cerevisiae genome. Nature **418**(6896), 387–391 (2002)
29. Soldatova, L.N., et al.: Representation of probabilistic scientific knowledge. J. Biomed. Semant. **4**(1), S7 (2013)
30. Arp, R., et al.: Building Ontologies with Basic Formal Ontology. The MIT Press, Cambridge (2015)