BIOINFORMATICS APPLICATIONS NOTE
Vol. 27 no. 17 2011, pages 2471–2472
doi:10.1093/bioinformatics/btr395

Databases and ontologies

Advance Access publication July 8, 2011

CSO validator: improving manual curation workflow for biological pathways

Euna Jeong[†], Masao Nagasaki[†,*], Emi Ikeda, Yayoi Sekiya, Ayumu Saito and Satoru Miyano
Human Genome Center, Institute of Medical Science, University of Tokyo, Tokyo 108-8639, Japan

Associate Editor: Jonathan Wren

ABSTRACT

Summary: Manual curation and validation of large-scale biological pathways are required to obtain high-quality pathway databases. In a typical curation process, model validation and model update based on appropriate feedback are repeated and requires considerable cooperation of scientists. We have developed a CSO (Cell System Ontology) validator to reduce the repetition and time during the curation process. This tool assists in quickly obtaining agreement among curators and domain experts and in providing a consistent and accurate pathway database.

Availability: The tool is available on http://csovalidator.csml.org.

Contact: masao@hgc.jp

Received on March 7, 2011; revised on June 23, 2011; accepted on June 24, 2011

1 INTRODUCTION

Modeling in systems biology is increasingly important for the system-level understanding of biological processes and predicting the behavior of the system. To obtain high-quality pathway databases, many important databases are built by manual curation. The creation of the pathway models is followed by validation of the created pathways by domain experts and update of the pathways based on appropriate feedback by curators. These procedures are iterative to record the desired specific annotated pathway. Improving the efficiency of model validation and model update is essential for reducing the time and effort required to construct high-quality biological pathways.

We had suggested a new method for validation of ontology, particularly Cell System Ontology (CSO; Jeong *et al.*, 2007). CSO is a generic framework to represent dynamic biological pathways with visualization in OWL (Web Ontology Language). Based on the proposed method (Jeong *et al.*, 2011), we have developed an efficient, user-friendly tool that considers biological meaning beyond checking for correct XML syntax.

Although the curation criteria and rules are given to curators, it is sometimes difficult correctly to assign a more specific subclass in the hierarchical structure of the ontology or a suitable term from controlled vocabularies. The correct annotation of the entity type, cellular location, the name of biological event and the number of molecules is important to represent biological meaning. It is helpful to visualize the pathway via visualization tools which can recognize and differently display the entity type and cellular location in cell system. Furthermore, it is also useful for simulation tools that can consider the concentration of the molecules and adjust parameters based on the type of biological event.

As a related work, Racunas *et al.*, 2006 carried out the verification of a pathway knowledge base in terms of event relationships. It is done on the level of the logical combinations of events, such as the order of events. However, it does not check the biological meaning of individual events. Another work is BioPaxRules which contains rules that cannot be formally defined in the BioPAX format, implemented via API (Paxtools) (http://www.biopax.org). As a complement to such efforts, we introduce a semi-automatic validation tool of ontology data in CSO which is generated via Cell Illustrator Online (CIO) (http://cionline.hgc.jp/) or via data conversion from other formats and resources.

2 CSO VALIDATOR

CSO validator itself is a stand-alone application with GUI. The tool is written in Java and needs Java Web Start. We used AllegroGraph (version 3) for the CSO data storage and query engine (http://www.franz.com/). AllegroGraph is an RDF graph database with support for SPARQL (SPARQL Protocol and RDF Query Language) as a query language (http://www.w3.org/TR/rdf-sparql-query/). The free version of AllegroGraph is enough to run CSO validator. The query manipulation and CSO data manipulation stored in AllegroGraph are carried out using Protégé OWL API (http://protege.stanford.edu/plugins/owl/api/) and Jena (http://jena.sourceforge.net/).

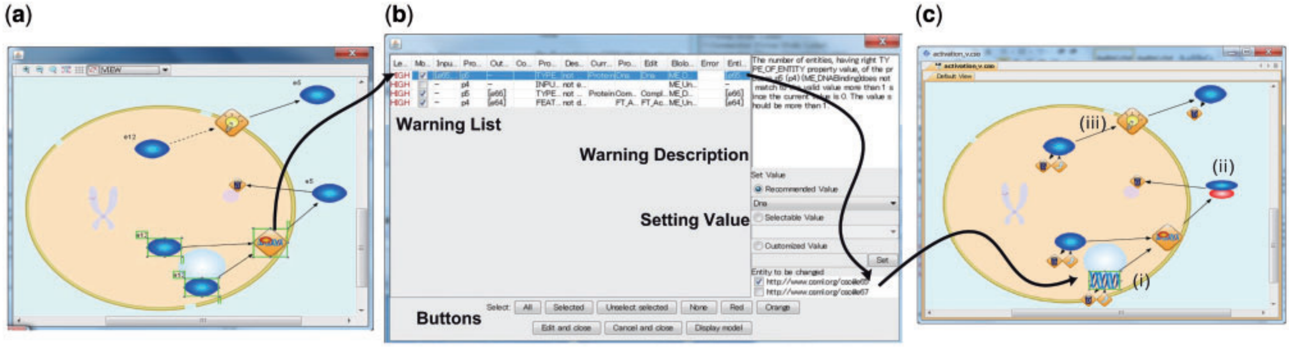
CSO validator uses the Systems Biology application XiP (eXtensible integrative Pipeline) that is a flexible, editable and modular environment with a user-friendly interface that does not require any programming skills to run, construct and edit workflows (http://xip.hgc.jp/). The pipeline used in CSO validator focuses on loading the model in CSO, storing it into the AllegroGraph database, i.e. to convert RDF to AllegroGraph format, validating the stored model, and exporting the validated model into CSO. As a companion tool, we have also developed CSML (Cell System Markup Language; an XML version of CSO) validator that can load the model in CSML and convert it to the CSO model for consuming of CSO validator.

In the main window of CSO validator, a user must specify input and output CSO file names, and database settings including a host name, a database directory, and a database name for the input file. It assumes that AllegroGraph is installed on a local PC. Basically, CSO validator does two things: validation and complementation of the given model.

- Validation: to check each biological process as to whether it is correctly embedding biological semantics by annotating event-specific

[*] To whom correspondence should be addressed.

[†] The authors wish it to be known that, in their opinion, the first two authors should be regarded as joint First Authors.



Le..	Mo..	Inpu..	Pro..	Out..	Co..	Pro..	Des..	Curr..	Pro..	Edit	Biolo..	Error	Enti..
HIGH	☑	[e65...	p5	–		TYPE..	not	Protein	Dna	Dna	ME_D..		[e65..
HIGH	☐	–	p4	–		INPU..	not e..				ME_Un..		–
HIGH	☑	–	p5	[e66]		TYPE..	not ...	ProteinCom..		Compl..	ME_D..		[e66]
HIGH	☑	–	p4	[e64]		FEAT..	not d..		FT_A..	FT_Ac..	ME_Un..		[e64]

The number of entities, having right TYPE_OF_ENTITY property value, of the pr[illegible] p5 (p4) (ME_DNABinding)does not match to [illegible] valid value more than 1 since the current [illegible] is 0. The value should be more than 1.

Set Value
● Recommended Value
Dna
○ Selectable Value
○ Customized Value

Entity to be changed
☑ http://www.csml.org/cso/e65
☐ http://www.csml.org/cso/e67

Fig. 1. (**a**) an example model to validate; (**b**) the validation window to guide correction; (**c**) the validated model, visualized on CIO. The table in (b) shows four warnings in the model. The highlighted part in (a) is a DNA binding process which is listed because there is no DNA as its input entity. Therefore, the validation window suggests that one entity from two should be changed into DNA. In the setting value pane, the recommended value is DNA and two candidate entities are listed below because the corresponding process has two input entities. In (c), the validated model shows the main changes: (i) one entity type is changed into DNA, (ii) the product of DNA binding is a complex, not a protein, and (iii) the connector is changed to represent the input entity, not an enzyme.

participants with cardinality, participant types, cellular location and others properties.

- Complementation: to add any missing processes, which allows a model to capture generic behaviors that govern system dynamics, such as protein-turnover.
  - To add a binding process for a starting complex which is not an output of any process.
  - To add a unknown production process for a starting entity except for complex, which is not an output of any process.
  - To add a degradation process for protein, complex, mRNA if they have no degradation process.

For this, a user can select which job will be done in the main window. If the validation option is checked and the given model needs to be modified, a validation window will be popped up for guiding correction. On the other hand, the complementation will be done with no prompt if selected. As an advanced option, our tool also supports the validation of multiple files stored in the same directory with JavaScript.

Figure 1 illustrates the validation procedure by using CSO validator. For the given model, CSO validator checks whether each process satisfies the given conditions (for details, refer to Jeong *et al.*, 2011). If there is any process not to satisfy conditions for validation, warning list is generated as shown in Figure 1b. The validation window consists of four panes: a table to list warnings, a warning description, setting the correct value and buttons on the bottom. Each row in the table lists the problematic process and involved entities with information such as any property against the conditions, any current value and recommended values. The detailed explanation for the warning is shown in the warning description. The setting value pane provides an interface for easy correction of the incorrect value. Combo boxes are used to display recommended values and selectable values. If there is no appropriate value in the recommended list, all possible values for the current property are listed in the selectable value combo box or a user can specify a customized value. For convenience, it is possible to browse the given model by clicking the display model button (Figure 1a) and the selected row is shown in the display model window by highlighting related elements. It is useful because during validation, a user may decide a wrong part without launching a visualization tool such as CIO. Figure 1c shows the validated result visualized via CIO. By using CIO, the validated model can be easily edited, visualized and simulated for further investigation.

3 SUMMARY

CSO validator is designed for validation of CSO models after curation and after every modification of a model by curators. It has been tested from last November by on average five curators and being applied to construct macrophage pathways (MACPAC, http://macpak.csml.org/) and osteoblast differentiation pathways. Curators take on average 3 or 4 h to learn and use CSO validator.

CSO validator reduces time spent on checking annotation mistakes and correcting problematic parts. To use our tool, knowledge of the CSO format is not necessary and most errors can be modified via an interactive GUI. How our tool finds obvious modeling errors by checking annotation mistakes was described in the paper (Jeong *et al.*, 2011). In this case, the modeling tool, CIO is needed for model modification. During construction of large pathway databases, it has an advantage to maintain consistency in terms of interpretation of experimental evidence and modeling style by suggesting minimum conditions based on 40 rules. Furthermore, when the curation criteria and rules are changed, the changes will be reflected to CSO validator and any modification for already constructed models will be done with no much burden.

Although CSO validator is for the models in the CSO format, the usage of CSO validator can be extended to other formats such as CSML, SBML (http://sbml.org/), CellML(http://www.cellml.org/), and BioPAX because CIO can read those formats and export them to CSO.

We believe that our approach can serve as a preprocessing step for model integration and the rule-based validation methodology can be applied to other ontology formats.

Conflict of Interest: none declared.

REFERENCES

Jeong,E. *et al.* (2007) Cell system ontology: representation for modeling, visualizing, and simulating biological pathways. *In Silico Biol.*, **7**, 623–638.

Jeong,E. *et al.* (2011) Ontology-based instance data validation for high-quality curated biological pathways. *BMC Bioinformatics*, **12** (Suppl. 1), S8.

Racunas,S.A. *et al.* (2006) A case study in pathway knowledgebase verification. *BMC Bioinformatics*, **7**, 196 .