Excel-SBOL Converter: Creating SBOL from Excel Templates and Vice Versa

Jeanet Mante
jet@mante.net
University of Colorado Boulder

Isabel Pötzsch
imp31@cam.ac.uk
University of Cambridge

Julian Abam
Julian.Abam@colorado.edu
University of Colorado Boulder

Jacob Beal
jakebeal@ieee.org
Raytheon BBN Technologies

Chris J. Myers
chris.myers@colorado.edu
University of Colorado Boulder

1 INTRODUCTION

Synthetic biology is bringing together engineers and biologists [10]. Associated with this interdisciplinary movement is the need for reusable tools that supplement the current understanding of genetic sequences. To satisfy this need, Synthetic Biology communities across the world have developed tools and ontologies to help describe their unique semantic annotations [1, 3–9, 13, 14, 17–19, 22]. Shared representations for data and metadata, grounded in well-defined ontology terms, can help reduce confusion when sharing materials between practitioners.[20]. The Synthetic Biology Open Language (SBOL) [5] is one of the approaches that has been developed to address this challenge. SBOL provides a standardized format for the electronic exchange of information on the structural and functional aspect of biological designs, supporting use of engineering principles of abstraction, modularity, and standardization in synthetic biology. Many tools have been created that work with SBOL, including the SynBioHub repository software for storing and sharing designs [12].

Using formal representations such as SBOL, however, typically requires either a thorough understanding of these standards or a suite of tools developed in concurrence with the ontologies [11]. Unfortunately, this poses a significant barrier to use for scientists not trained to work with such abstractions. One approach to lowering this barrier was demonstrated in the Systems Biology for Micro-Organisms (SysMO) consortium [2]. In SysMO, the MicroArray Gene Expression Markup Language (Mage-ML) was set up as an XML schema [15], and users were expected to submit data to the SysMO Assets Catalogue (called SEEK) in XML format in order to publish work. To allow the use of the Mage-ML language without having to understand XML, the RightField tool was created [21], an ontology annotation and information management application that can add constrained ontology term selection to Excel spreadsheets. This tool enables administrators to create templates with controlled vocabularies, such that the scientists utilizing the tool would never actually see the raw RightField, only the more familiar Excel spreadsheet interface.

Similarly, users of SBOL and SynBioHub have faced a steep learning curve for understanding the underlying ontology: as assessed in [16], “For successful use and interpretation of metadata presented in SynBioHub, the semantic annotation process should be biologist-friendly and hide the underlying RDF predicates.” Accordingly, the Excel-SBOL Converter presented here has been designed to provide a simple way for users to generate SBOL data without needing a detailed understanding of the underlying ontology and associated technologies. The converter provides a simple way for users to manage data by allowing users to download SBOL into Excel templates and submit Excel templates for conversion into SBOL.

2 RESULTS

The Excel-SBOL Converter enables the round trip conversion of SBOL2 into Excel Templates and Excel Templates to SBOL2, as shown in Figure 1. This metadata conversion allows researchers the flexibility of working in Excel without losing the benefits of SBOL2 and maintains inter-operability with the SBOL2 tool suite. The converter also facilitates understanding of the metadata relationships, management, and structure within the SBOL2 XML file. The converter has been implemented and published as two Python packages, Excel2SBOL and SBOL2Excel, under a free and open license and published to pypi for easy pip installation and integration with other projects. Specific benefits of each of the packages are described below.

SBOL2Excel

The SBOL2Excel package is designed in a modular way to allow further expansion. The package’s design includes an initial step from SBOL2 to a dictionary structure. This function could be swapped out with other functions to enable the conversion of other formats such as SBOL3 and GenBank to the Excel Templates.
Another key feature is that any arbitrary Component-Definition properties can be converted into Excel columns. However, some common types of properties are given additional post processing to make them more user friendly. SBOL2Excel operates in five general steps:

1. An SBOL2 document is read (with property-specific post-processing) and converted to a dictionary.
2. The dictionary is converted to a pandas dataframe.
3. The dataframe’s columns are reordered based on the default Excel Template.
4. Extraneous columns are dropped from the dataframe.
5. The dataframe is output into a formatted Excel Spreadsheet.

Figure 2 shows a high level representation of SBOL2Excel’s modular architecture.

Excel2SBOL

This package is able to parse and add information from arbitrary columns into component definitions as annotations. Specific columns are added as a particular type of annotation based on a column conversion table that is accessible in the Excel Template. The package is also able to deal with different templates as long as the Template parameters are added to the modules config file. SBOL2Excel operates in three general steps:

1. Parse the different sections of the template (Overview Information, Design Description, and Part Table).
2. Parse the table indicating how columns are converted.
3. For every row, create a component definition and add attributes based on each column in the parts table.

Figure 3 is a high level representation of Excel2SBOL’s modular architecture.

3 DISCUSSION

The next steps in the development of the Excel-SBOL Converter are:

- Making the incorporation of ontologies with Excel Templates easier and less hard-coded, perhaps by relying on RightField Templates. Integrating RightField may be useful in preparing ontologies for templatization.
- Making the benefits of adherence to community standards clearer to users to create greater uptake.
- Establishing a core set of part metadata which can then be enforced via the Excel Templates.
- Making both packages SBOL3 Compliant. Enabling this feature would allow users to perform conversions from SBOL documents (perhaps generated by SynBioHub) that contain SBOL2, SBOL3, or both.
- Creating SynBioHub plugins for both packages to integrate their functionality with SynBioHub. This would enable users to perform conversions without directly interacting with Python code.
- Expanding the range of top-level SBOL types that can be converted (e.g. Activities, Composite Components, and Modules).

ACKNOWLEDGEMENTS

JA, JM, and CM are supported by the National Science Foundation under Grant No. 1939892. JM is additionally supported by a Dean’s Graduate Assistantship at the University of Colorado Boulder. IP was supported by the Google Summer of Code. JB is partially supported by AFRL and DARPA contract FA8750-17-C-0184. This document does not contain technology or technical data controlled under either U.S. International Traffic in Arms Regulation or U.S. Export Administration Regulations. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agencies.
Figure 2: The module architecture of the SBOL2Excel Python Package. Note that the colors of the functions indicate which Python module they can be found in, as indicated in the key on the bottom right.

Figure 3: The module architecture of the Excel2SBOL Python Package. Note that the colors of the functions indicate which Python module they can be found in, as indicated in the key on the bottom right.

REFERENCES


to design formal protocols for biological laboratories. *Automated Experimentation* 2, 1 (Apr 2010), 3.


