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

Jeanet Mante,[†] Julian Abam,[†] Sai P. Samineni,[†] Isabel M. Pötzsch,[‡] Jacob Beal,[¶] and Chris J. Myers[*,†]

†*University of Colorado Boulder, Boulder, Colorado, 80309*

‡*University of Cambridge, Cambridge, UK*

¶*Raytheon BBN Technologies, Cambridge, Massachusetts 02138, United States*

E-mail: chris.myers@colorado.edu

**Abstract**

Standards support synthetic biology research by enabling the exchange of component information. However, using formal representations, such as the *Synthetic Biology Open Language* (SBOL), typically requires either a thorough understanding of these standards or a suite of tools developed in concurrence with the ontologies. Since these tools may be a barrier for use by many practitioners, the Excel-SBOL Converter was developed to facilitate the use of SBOL and integration into existing workflows. The converter consists of two Python libraries: one that converts Excel templates to SBOL, and another that converts SBOL to an Excel workbook. Both libraries can be used either directly or via a SynBioHub plugin.

## Keywords

Excel, SBOL, conversion, Excel-to-SBOL, SBOL-to-Excel, ontologies

# Introduction

Synthetic biology is bringing together engineers and biologists to design biological circuits for a variety of applications in energy, medicine, and bio-manufacturing.[1] Associated with this interdisciplinary movement is the need for tools that support reusability and 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.[2–15] Shared representations for data and metadata, grounded in well-defined ontology terms, can help reduce confusion when sharing materials between researchers.[16] The *Synthetic Biology Open Language* (SBOL)[6] has been developed to address this challenge. SBOL provides a standardized format for the electronic exchange of information on the structural and functional aspects of biological designs, supporting the use of engineering principles such as abstraction, modularity, and standardization for synthetic biology. Many tools have been created that work with SBOL,[17–26] including the SynBioHub repository for storing and sharing designs.[27] The original SBOL has been developed further to allow other data types to be represented and more information to be captured. This has led to SBOL2[28] and SBOL3.[29]

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.[30] Unfortunately, this poses a significant barrier for scientists not trained to work with such abstractions. Not using standards makes it difficult to share and reuse parts.[31–36] Therefore, the time and effort required to find parts is much greater, and the ability to use the tools to automate design is limited.

One approach to lowering the barrier for use of ontologies was demonstrated by the *Systems Biology for Micro-Organisms* (SysMO) consortium.[37] In SysMO, the *MicroArray Gene Expression Markup Language* (Mage-ML) was set up as an XML schema,[38] 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.[39] This tool is an ontology annotation and information management application that can add constrained ontology term selection to Excel spreadsheets. It 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. Spreadsheets are a popular interface as many biological workflows already use spreadsheets and *comma separated values* (CSV) files. Furthermore, several popular tools use spreadsheets and CSV files as inputs or outputs, including Addgene (`https://www.addgene.org/`), and Opentrons (`https://opentrons.com/`).

Users of SBOL and SynBioHub have also faced a steep learning curve for understanding the underlying ontology: as assessed in,[40] "For successful use and interpretation of metadata presented in SynBioHub, the semantic annotation process should be biologist-friendly and hide the underlying RDF predicates." Recently, SynBio2Easy was published as a command line tool to convert Excel spreadsheets of plasmids to SBOL.[41] The tool was designed to enable several steps of a specific workflow for designing and depositing *Synechocystis* plasmids into a public SynBioHub repository.

Embracing the same spreadsheet-based interface as these prior works, this paper presents the Excel-SBOL Converter, a tool designed to provide a simple way for users to generate and visualize SBOL data without needing a detailed understanding of the underlying ontology and associated technologies. Unlike SynBio2Easy, our converter generalizes beyond plasmids to multiple kinds of SBOL data, as well as allowing customization of the templates. This converter thus provides a simple way for users to manage data by allowing users to both download SBOL into Excel templates and to submit Excel templates for conversion into SBOL. This paper presents the architecture and key engineering decisions for the Excel-SBOL Converter, as well as it's integration with SynBioHub.

# Results

The Excel-SBOL Converter enables researchers that are more comfortable with Excel spreadsheets to make use of SBOL repositories and tools without having to manipulate or understand the SBOL data standard. The Excel-SBOL Converter is currently implemented as two separate libraries: the Excel-to-SBOL library is used to convert Excel spreadsheets formatted using pre-designed templates into SBOL data, while the SBOL-to-Excel library is used to convert SBOL data into Excel spreadsheets. These two libraries taken together enable data to be converted between Excel spreadsheets and SBOL data.

## Excel-to-SBOL

The Excel-to-SBOL library converts Excel spreadsheet templates into SBOL data. While initially designed based on fixed spreadsheet templates developed in the Defense Advanced Research Projects Agency (DARPA) Synergistic Discovery and Design (SD2) program (`https://sd2e.org/`)), the library has since been generalized to allow more flexibility. With this approach, a user needs to have only minimal knowledge of SBOL, while a template designer needs knowledge of both the SBOL data representation and the Excel-to-SBOL template structure. SBOL experts design the templates that are used repeatedly by biologists with no SBOL knowledge (Figure 1).

## Excel Spreadsheets for Lab Users

As an example, let us consider an Excel spreadsheet template that can organize genetic parts into several part collection libraries (Figure **??**). This example is composed of one Excel File containing 2 data sheets:

1. A sheet describing the genetic parts (Figure 3a).
2. A sheet describing composite parts (Figure 3b).

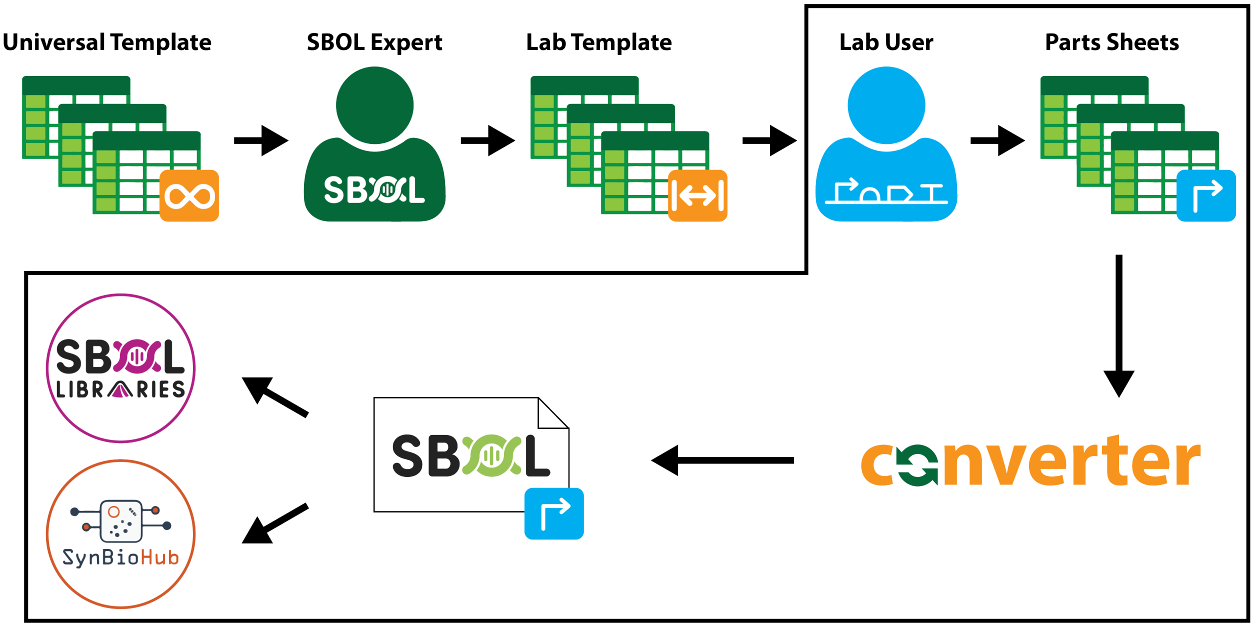


Figure 1: Excel2SBOL converter in a synbio workflow. An SBOL Expert makes a specific template for a lab based on the universal template. This template can then be used and **reused** by lab scientists to store part information. This information may be converted to SBOL using the converter presented in this paper. The parts may then be uploaded to design repositories or used in design tools.

In order to use this template, a user would enter each genetic part name, part design choices, part description, data source prefix (PubMed, GenBank etc), Data Source (e.g. the PubMed Id), source organism, and the sequence. The user can then specify composite parts in the composites sheet. The composites have a name, design notes, description, and a list of parts used (names being either other composites or from the basic parts sheet). Note that any columns may be left blank, for example if the sequence file is local then no data source is required. When all this information has been collected the user can convert the sheet to SBOL using the Excel-to-SBOL library (or a SynBioHub plugin, which is described below). The template programming described in the next session should have already been done during the set up of the template by an experienced SBOL user.

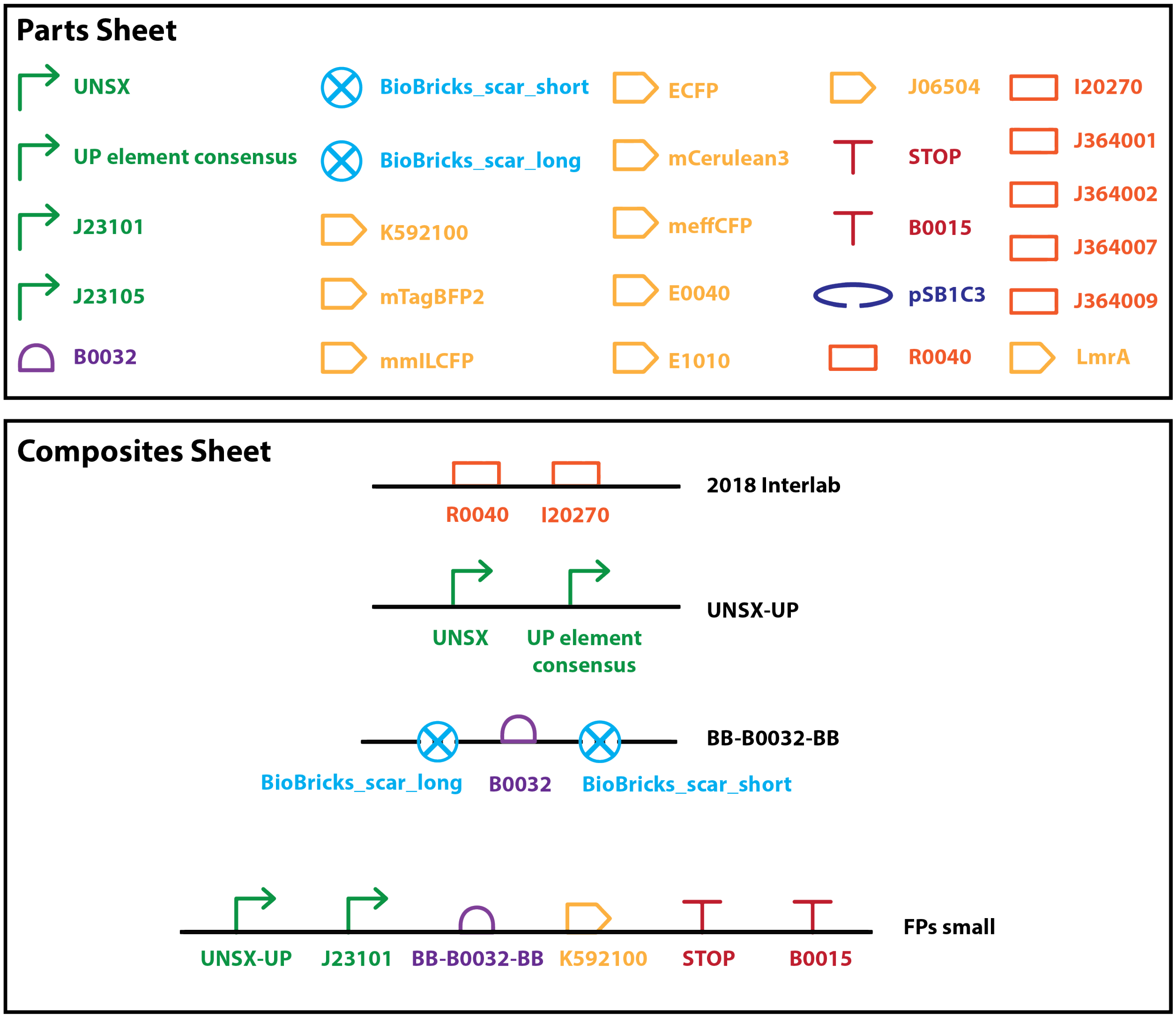
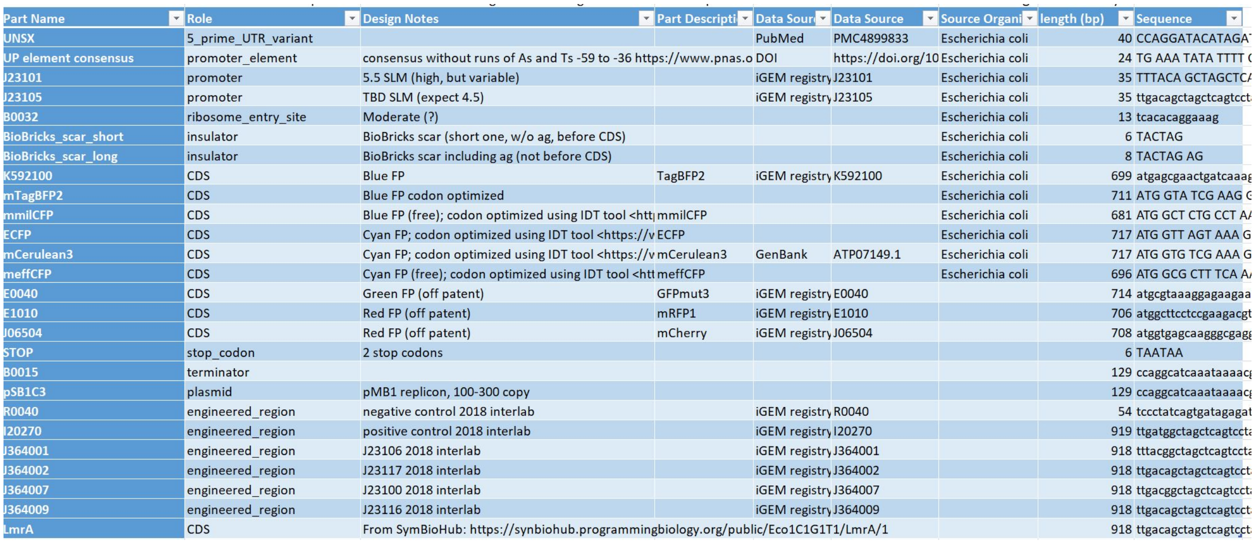
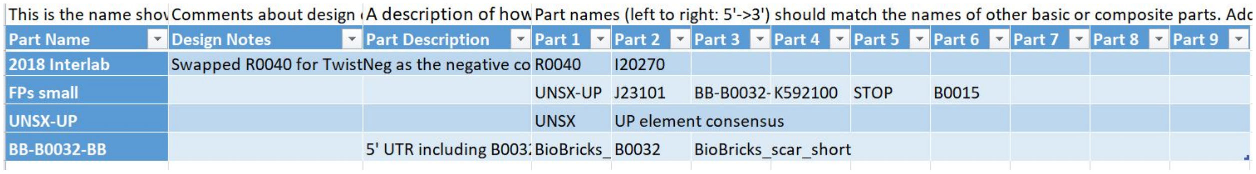


Figure 2: Visual representation of an example spreadsheet set (example sheets shown in Figure 3). This displays the information contained by two spreadsheets sheets: a parts library and a composites sheet.



| Part Name | Role | Design Notes | Part Descripti | Data Sour | Data Source | Source Organi | length (bp) | Sequence |
|---|---|---|---|---|---|---|---|---|
| UNSX | 5_prime_UTR_variant | | | PubMed | PMC4899833 | Escherichia coli | 40 | CCAGGATACATAGA |
| UP element consensus | promoter_element | consensus without runs of As and Ts -59 to -36 https://www.pnas.o | | DOI | https://doi.org/10 | Escherichia coli | 24 | TG AAA TATA TTTT |
| J23101 | promoter | 5.5 SLM (high, but variable) | | iGEM registry | J23101 | Escherichia coli | 35 | TTTACA GCTAGCTC |
| J23105 | promoter | TBD SLM (expect 4.5) | | iGEM registry | J23105 | Escherichia coli | 35 | ttgacagctagctcagtcct |
| B0032 | ribosome_entry_site | Moderate (?) | | | | Escherichia coli | 13 | tcacacaggaaag |
| BioBricks_scar_short | insulator | BioBricks scar (short one, w/o ag, before CDS) | | | | Escherichia coli | 6 | TACTAG |
| BioBricks_scar_long | insulator | BioBricks scar including ag (not before CDS) | | | | Escherichia coli | 8 | TACTAG AG |
| K592100 | CDS | Blue FP | TagBFP2 | iGEM registry | K592100 | Escherichia coli | 699 | atgagcgaactgatcaaa |
| mTagBFP2 | CDS | Blue FP codon optimized | | | | Escherichia coli | 711 | ATG GTA TCG AAG |
| mmilCFP | CDS | Blue FP (free); codon optimized using IDT tool <htt | mmilCFP | | | Escherichia coli | 681 | ATG GCT CTG CCT A |
| ECFP | CDS | Cyan FP; codon optimized using IDT tool <https://v | ECFP | | | Escherichia coli | 717 | ATG GTT AGT AAA G |
| mCerulean3 | CDS | Cyan FP; codon optimized using IDT tool <https://v | mCerulean3 | GenBank | ATP07149.1 | Escherichia coli | 717 | ATG GTG TCG AAA G |
| meffCFP | CDS | Cyan FP (free); codon optimized using IDT tool <htt | meffCFP | | | Escherichia coli | 696 | ATG GCG CTT TCA A |
| E0040 | CDS | Green FP (off patent) | GFPmut3 | iGEM registry | E0040 | | 714 | atgcgtaaaggagaagaa |
| E1010 | CDS | Red FP (off patent) | mRFP1 | iGEM registry | E1010 | | 706 | atggcttcctccgaagacgt |
| J06504 | CDS | Red FP (off patent) | mCherry | iGEM registry | J06504 | | 708 | atggtgagcaagggcgag |
| STOP | stop_codon | 2 stop codons | | | | | 6 | TAATAA |
| B0015 | terminator | | | | | | 129 | ccaggcatcaaataaaac |
| pSB1C3 | plasmid | pMB1 replicon, 100-300 copy | | | | | 129 | ccaggcatcaaataaaac |
| R0040 | engineered_region | negative control 2018 interlab | | iGEM registry | R0040 | | 54 | tccctatcagtgatagagat |
| I20270 | engineered_region | positive control 2018 interlab | | iGEM registry | I20270 | | 919 | ttgatggctagctcagtcct |
| J364001 | engineered_region | J23106 2018 interlab | | iGEM registry | J364001 | | 918 | tttacggctagctcagtcct |
| J364002 | engineered_region | J23117 2018 interlab | | iGEM registry | J364002 | | 918 | ttgacagctagctcagtcct |
| J364007 | engineered_region | J23100 2018 interlab | | iGEM registry | J364007 | | 918 | ttgacggctagctcagtcct |
| J364009 | engineered_region | J23116 2018 interlab | | iGEM registry | J364009 | | 918 | ttgacagctagctcagtcct |
| LmrA | CDS | From SymBioHub: https://synbiohub.programmingbiology.org/public/Eco1C1G1T1/LmrA/1 | | | | | 918 | ttgacagctagctcagtcct |

(a)



| This is the name shov | Comments about design ( | A description of how | Part names (left to right: 5'->3') should match the names of other basic or composite parts. Adc | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Part Name** | **Design Notes** | **Part Description** | **Part 1** | **Part 2** | **Part 3** | **Part 4** | **Part 5** | **Part 6** | **Part 7** | **Part 8** | **Part 9** |
| 2018 Interlab | Swapped R0040 for TwistNeg as the negative co | | R0040 | I20270 | | | | | | | |
| FPs small | | | UNSX-UP | J23101 | BB-B0032- | K592100 | STOP | B0015 | | | |
| UNSX-UP | | | UNSX | UP element consensus | | | | | | | |
| BB-B0032-BB | | 5' UTR including B003: | BioBricks_ | B0032 | BioBricks_scar_short | | | | | | |

(b)

Figure 3: The Lab User Sheets. **(a) Parts sheet.** This is where the users add parts. Each part has a role (e.g. promoter), design notes (comments about design decisions), a description (what is it meant to do), a data source (where is it coming from and what is the ID e.g. PubMed and the accession number), the source organism (the organism from which the DNA came), length (automatically generated using a formula), and the sequence (a set of ATGCs). **(b) Composites sheet.** This is a sheet where users can specify composite parts. The names in the part columns can refer to other composites or basic parts from the parts sheet (Figure 3a). The user can specify design notes and a description and then as many parts as needed.

## Excel Spreadsheet Template Programming

Excel-to-SBOL is designed to parse a variety of interlinking SBOL object types in a flexible manner, whilst maintaining code simplicity. This makes it possible to design specific excel templates for different users and process them all using a single code library. The details of how the parsing work, and how to build a specific template are outlined in the supplemental information. The main template features are:

- Sheets can refer to items on other sheets
- Several columns may be used together to generate an SBOL property
- A single column can be converted into several properties
- Human readable terms can be converted to machine readable terms using a combination of software tools and substitution of terms from pre-coded look-up sheets
- Cell inputs (or the machine readable version) can be checked against a regular expression

## SBOL-to-Excel

The SBOL-to-Excel converter performs the opposite task of taking SBOL data and converting it into an Excel spreadsheet. This may be useful if data needs to be copied out, or editted and re-uploaded (this functionality is still in progress). This converter reads an SBOL file in any standard RDF format (e.g., XML, JSON-LD, Turdle, N-triples), and for every SBOL object found, it creates a row in a spreadsheet that has a column for each property of the object. In this case, the an RDF (Resource Description Framework) python library is used rather than an SBOL library in order to allow the conversion of different SBOL object types in any version of SBOL whilst maintaining code simplicity. This is possible as each SBOL version is a specific kind of RDF.

An example Excel spreadsheet generated by the SBOL-to-Excel converter from the SBOL generated for the library example presented earlier is shown in Supplemental Figure 5. This spreadsheet looks similar to the templates described earlier, but has a few key differences. First, all column names are based on the property name rather than human readable column names. Second, no Init or Column Definitions sheet is present in the output (these are explained in the supplemental material). Finally, many of the properties are full URLs rather than human readable names. For example, http://examples.org/ECFP instead of ECFP. The addition of the examples.org is what allows linking to parts in a database. If the data was uploaded to SynBioHub and downloaded into a spreadsheet it would be http://synbiohub.org/ECFP which is where information about the part could be accessed. Future work will address these differences to allow full round-tripping of data from Excel to SBOL and back again.

## SynBioHub Plugins

SynBioHub plugins for the Excel-SBOL Converter were developed to enable the converter to be integrated into a data processing workflow using the SynBioHub repository (see Figure 4). SynBioHub plugins are a way to create modular extensions to the capabilities of the SynBioHub repository.[42] The plugins developed as part of this project allow the integration of SynBioHub into synthetic biology workflows with spreadsheets. A submit plugin was developed for the Excel-to-SBOL converter. This plugin takes in Excel spreadsheet templates, converts them to SBOL, and uploads the converted SBOL to SynBioHub. Similarly, a download plugin was developed using the SBOL-to-Excel library that allows a user to download SBOL stored in SynBioHub in the form of an Excel spreadsheet.

Each of these four samples then has two signals (i.e., *optical density* (OD), *green fluorescent protein* (GFP)) measured for 100 time points, resulting in a total of 800 measurements.

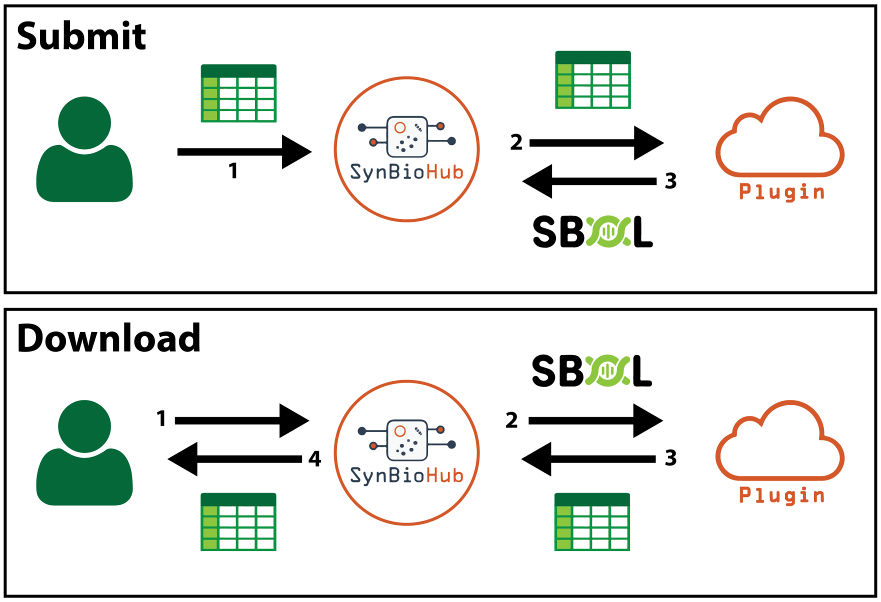


Figure 4: Integration of the Excel-SBOL Converter with SynBioHub via plugins. **Submit**: When a user uploads an Excel spreadsheet template to the submit endpoint, SynBioHub sends it to the Excel-to-SBOL plugin, which returns SBOL to be deposited in SynBioHub. **Download**: If a user requests an Excel spreadsheet template to be downloaded from SynBioHub, SynBioHub sends the appropriate SBOL to the SBOL-to-Excel plugin. The plugin returns an Excel spreadsheet to SynBioHub, which is then returned to the user.

# Discussion

We have presented two Python converter libraries: Excel-to-SBOL and SBOL-to-Excel. The development of these libraries simplifies the incorporation of SBOL into existing synthetic biology workflows that make use of spreadsheets for data storage and exchange. The Excel-to-SBOL converter allows multiple sheets with different SBOL object types to be processed. The converter can be used to convert spreadsheet columns into any RDF properties, and can output either SBOL Version 2 or 3 documents, as well as validating information using user supplied regular expressions and converting user-friendly names into ontology terms. Its complement, the SBOL-to-Excel Converter, can process any RDF file and turn it into a spreadsheet. This allows both SBOL2 and SBOL3 documents to be converted to spreadsheets. The conversion makes the values more human readable, and splits different SBOL object types across sheets.

While the Excel-SBOL Converter is already useful for many applications, there are several improvements planned for the future. For the Excel-to-SBOL Converter, real-time sheet checking using Excel plugins would enable errors to be detected and fixed more efficiently. Second, more standard Excel spreadsheet templates with example data should be created to provide a starting point for both users and template creators. For the SBOL-to-Excel Converter, the next step is to record the conversion process into a column definition sheet in order to support round-trip conversion back to SBOL.

# Methods

Below is a more in depth description of the methods used by the Excel-to-SBOL and SBOL-to-Excel Converters, as well as the methods used in the case studies. Note that all code was written in Python.

## Excel-to-SBOL

This converter relies on a set of Python modules to function. The modules used and their functionality in the converter is explained below.

- **Openpyxl** is a Python library to read and write Excel files. This library is used together with Pandas to read in the data from the Excel workbook and store it as a data frame.
- **pandas** is a data analysis library. It is used for the reading in of the Excel workbook and the further storage and manipulation of the data.
- **pySBOL2 and pySBOL3** are libraries for the reading, writing, and manipulation of SBOL data. These libraries are used to create the SBOL objects required and write them out to an SBOL document.

- **Tyto** is a tool to make the semantic web more accessible. It is used to convert cell values to ontology terms.

## SBOL-to-Excel

SBOL-to-Excel relies on a set of libraries to ensure the smooth processing of a given SBOL document. These libraries facilitate advanced processing procedures, and ensure that the data is properly output to Excel for the user to analyze.

- **RDFLib** is a python package that enables the user to work with the Resource Description Framework (RDF). RDF serves as an important proponent to this project. Its functionality includes the ability to parse through RDF triples within an Excel file. With access to the subject, object, and predicate triples, the library made it possible to extract these values, and convert them into a form that can be easily interpreted in order to eventually be output to Excel.

- **Openpyxl** is a Python library to read and write Excel files. The Openpyxl library is crucial to the Converter, as its functionality allows the writing of SBOL data into the Excel spreadsheet. It is then also used to format the spreadsheet (including creating the hyperlinks).

- **Pandas** is a python library that provides data structures with capabilities for the accessing and manipulating of data that it holds, and does so with the information processed from the SBOL collection. This occurs specifically when the data is being read in. The subjects, objects, and predicates are set into the pandas data frame in order to be processed by various modules, before being output into Excel.

- **Validators** is used to check items expected to be URLs are valid URLS.

- **Pytest** makes it easy to write small tests, as well as, scales to support testing for large applications. This library enabled modular testing, ensuring that the correct forms of data were being passed through the converter at the appropriate points.

## Acknowledgement

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. JA is additionally supported by a CU Boulder Discovery Learning Apprenticeship. IP was supported by the Google Summer of Code and the Bischöfliche Studienförderung Cusanuswerk. PS was supported by the SBOL Industrial Consortium. JB and CM are partially supported by the Air Force Research Laboratory (AFRL) contracts FA8750-17-C-0184 and FA8750-17-C-0229. SynBioHub and the Excel-SBOL Converter plugins are run on a Microsoft Azure Server provided by Microsoft Research. Microsoft Excel is trademark of the Microsoft group of companies. 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. All authors contributed to the writing of this manuscript.

### Author Contributions

All authors contributed to the writing of this manuscript. JM worked on Excel-to-SBOL, supervising IP in the initial design and then taking over. JA worked on SBOL-to-Excel with the help of JM. SS worked on Excel-to-SBOL template testing with the help of JA and is working on a case study. PS is working on Tyto integration into Excel. CM supervised the project. JB contributed guidance and the initial DARPA spreadsheets.

### Conflicts of Interest

The authors declare no conflicts of interest.

# Supporting Information Available

### Excel2SBOL

- GitHub: `https://github.com/SynBioDex/Excel-to-SBOL`
- Documentation: `https://github.com/SynBioDex/Excel-to-SBOL/wiki`
- Plugin: `https://github.com/SynBioHub/Plugin-Submit-Excel2SBOL`
- PyPI: `https://pypi.org/project/excel2sbol/`

### SBOL2Excel

- GitHub: `https://github.com/SynBioDex/SBOL-to-Excel`
- Documentation: `https://github.com/SynBioDex/SBOL-to-Excel/wiki`
- Plugin: `https://github.com/SynBioHub/Plugin-Download-SBOL2Excel`
- PyPI: `https://pypi.org/project/sbol2excel/`

### Supplemental Files

- Additional technical information about the conversion Excel-SBOL Converter Supplementary Information.pdf
- Template for Excel to SBOL Conversion SBOL2_simple_parts_template.xlsx
- Output of SBOL to Excel sbol2excel_output.xlsx

# References

(1) Khalil, A. S.; Collins, J. J. Synthetic biology: applications come of age. *Nature Reviews Genetics* **2010**, *11*, 367–379.

(2) Hucka, M.; Finney, A.; Sauro, H. M.; Bolouri, H.; Doyle, J. C.; Kitano, H.; Arkin, A. P.; Bornstein, B. J.; Bray, D.; Cornish-Bowden, A. et al. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics (Oxford, England)* **2003**, *19*, 524–531.

(3) Hucka, M.; Nickerson, D. P.; Bader, G. D.; Bergmann, F. T.; Cooper, J.; Demir, E.; Garny, A.; Golebiewski, M.; Myers, C. J.; Schreiber, F. et al. Promoting Coordinated Development of Community-Based Information Standards for Modeling in Biology: The COMBINE Initiative. *Frontiers in Bioengineering and Biotechnology* **2015**, *3*, 19.

(4) Waltemath, D.; Adams, R.; Bergmann, F. T.; Hucka, M.; Kolpakov, F.; Miller, A. K.; Moraru, I. I.; Nickerson, D.; Sahle, S.; Snoep, J. L. et al. Reproducible computational biology experiments with SED-ML–the Simulation Experiment Description Markup Language. *BMC systems biology* **2011**, *5*, 198.

(5) Novère, N. L.; Hucka, M.; Mi, H.; Moodie, S.; Schreiber, F.; Sorokin, A.; Demir, E.; Wegner, K.; Aladjem, M. I.; Wimalaratne, S. M. et al. The Systems Biology Graphical Notation. *Nature Biotechnology* **2009**, *27*, 735–741.

(6) Galdzicki, M.; Clancy, K. P.; Oberortner, E.; Pocock, M.; Quinn, J. Y.; Rodriguez, C. A.; Roehner, N.; Wilson, M. L.; Adam, L.; Anderson, J. C. et al. The Synthetic Biology Open Language (SBOL) provides a community standard for communicating designs in synthetic biology. *Nature Biotechnology* **2014**, *32*, 545–550.

(7) Gleeson, P.; Crook, S.; Cannon, R. C.; Hines, M. L.; Billings, G. O.; Farinella, M.; Morse, T. M.; Davison, A. P.; Ray, S.; Bhalla, U. S. et al. NeuroML: A Language for Describing Data Driven Models of Neurons and Networks with a High Degree of Biological Detail. *PLOS Computational Biology* **2010**, *6*, e1000815.

(8) Cannon, R. C.; Gleeson, P.; Crook, S.; Ganapathy, G.; Marin, B.; Piasini, E.; Silver, R. A. LEMS: a language for expressing complex biological models in concise and hierarchical form and its use in underpinning NeuroML 2. *Frontiers in Neuroinformatics* **2014**, *8*, 79.

(9) Faeder, J. R.; Blinov, M. L.; Hlavacek, W. S. In *Systems Biology*; Maly, I. V., Ed.; Methods in Molecular Biology; Humana Press, 2009; Vol. 500; p 113–167.

(10) Bergmann, F. T.; Adams, R.; Moodie, S.; Cooper, J.; Glont, M.; Golebiewski, M.; Hucka, M.; Laibe, C.; Miller, A. K.; Nickerson, D. P. et al. COMBINE archive and OMEX format: one file to share all information to reproduce a modeling project. *BMC bioinformatics* **2014**, *15*, 369.

(11) Gennari, J. H.; Neal, M. L.; Galdzicki, M.; Cook, D. L. Multiple ontologies in action: composite annotations for biosimulation models. *Journal of Biomedical Informatics* **2011**, *44*, 146–154.

(12) Wolstencroft, K.; Krebs, O.; Snoep, J. L.; Stanford, N. J.; Bacall, F.; Golebiewski, M.; Kuzyakiv, R.; Nguyen, Q.; Owen, S.; Soiland-Reyes, S. et al. FAIRDOMHub: a repository and collaboration environment for sharing systems biology research. *Nucleic Acids Research* **2017**, *45*, D404–D407.

(13) Wittig, U.; Kania, R.; Golebiewski, M.; Rey, M.; Shi, L.; Jong, L.; Algaa, E.; Weidemann, A.; Sauer-Danzwith, H.; Mir, S. et al. SABIO-RK–database for biochemical reaction kinetics. *Nucleic Acids Research* **2012**, *40*, D790–796.

(14) Nickerson, D.; Atalag, K.; de Bono, B.; Geiger, J.; Goble, C.; Hollmann, S.; Lonien, J.; Müller, W.; Regierer, B.; Stanford, N. J. et al. The Human Physiome: how standards, software and innovative service infrastructures are providing the building blocks to make it achievable. *Interface Focus* **2016**, *6*, 20150103.

(15) Yu, T.; Lloyd, C. M.; Nickerson, D. P.; Cooling, M. T.; Miller, A. K.; Garny, A.; Terkildsen, J. R.; Lawson, J.; Britten, R. D.; Hunter, P. J. et al. The Physiome Model Repository 2. *Bioinformatics (Oxford, England)* **2011**, *27*, 743–744.

(16) Wolstencroft, K.; Owen, S.; Horridge, M.; Jupp, S.; Krebs, O.; Snoep, J.; Du Preez, F.; Mueller, W.; Stevens, R.; Goble, C. Stealthy annotation of experimental biology by spreadsheets. *Concurrency and Computation: Practice and Experience* **2013**, *25*, 467–480.

(17) Watanabe, L.; Nguyen, T.; Zhang, M.; Zundel, Z.; Zhang, Z.; Madsen, C.; Roehner, N.; Myers, C. IBIOSIM 3: A Tool for Model-Based Genetic Circuit Design. *ACS Synthetic Biology* **2019**, *8*, 1560–1563.

(18) Myers, C. J.; Barker, N.; Jones, K.; Kuwahara, H.; Madsen, C.; Nguyen, N.-P. D. iBioSim: A Tool for the Analysis and Design of Genetic Circuits. *Bioinformatics* **2009**, *25*, 2848–2849.

(19) Roehner, N.; Myers, C. J. Directed Acyclic Graph-Based Technology Mapping of Genetic Circuit Models. *ACS Synthetic Biology* **2014**, *3*, 543–555.

(20) Chen, Y.; Zhang, S.; Young, E. M.; Jones, T. S.; Densmore, D.; Voigt, C. A. Genetic circuit design automation for yeast. *5*, 1349–1360.

(21) Nielsen, A. A. K.; Der, B. S.; Shin, J.; Vaidyanathan, P.; Paralanov, V.; Strychalski, E. A.; Ross, D.; Densmore, D.; Voigt, C. A. Genetic circuit design automation. *352*, Publisher: American Association for the Advancement of Science _eprint: https://science.sciencemag.org/content/352/6281/aac7341.full.pdf.

(22) Baig, H.; Madsen, J. A Top-down Approach to Genetic Circuit Synthesis and Optimized Technology Mapping. 9th International Workshop on Bio-Design Automation (Pittsburgh, PA). 2017; pp 1–2.

(23) Gaspar, P.; Oliveira, J. L.; Frommlet, J.; Santos, M. A. S.; Moura, G. EuGene: Maximizing Synthetic Gene Design for Heterologous Expression. *Bioinformatics* **2012**, *28*, 2683–2684.

(24) Chen, J.; Densmore, D.; Ham, T. S.; Keasling, J. D.; Hillson, N. J. DeviceEditor Visual Biological CAD Canvas. *Journal of Biological Engineering* **2012**, *6*, 1.

(25) Terry, L.; Earl, J.; Thayer, S.; Bridge, S.; Myers, C. J. SBOLCanvas: A Visual Editor for Genetic Designs. *ACS Synthetic Biology* **2021**, *10*, 1792–1796.

(26) Czar, M. J.; Cai, Y.; Peccoud, J. Writing DNA with GenoCADTM. *Nucleic Acids Research* **2009**, *37*, W40–W47.

(27) McLaughlin, J. A.; Myers, C. J.; Zundel, Z.; Mısırlı, G.; Zhang, M.; Ofiteru, I. D.; Goñi-Moreno, A.; Wipat, A. SynBioHub: A Standards-Enabled Design Repository for Synthetic Biology. *ACS Synthetic Biology* **2018**, *7*, 682–688.

(28) Roehner, N.; Beal, J.; Clancy, K.; Bartley, B.; Misirli, G.; Grünberg, R.; Oberortner, E.; Pocock, M.; Bissell, M.; Madsen, C. et al. Sharing Structure and Function in Biological Design with SBOL 2.0. *ACS Synthetic Biology* **2016**, *5*, 498–506.

(29) McLaughlin, J. A.; Beal, J.; Mısırlı, G.; Grünberg, R.; Bartley, B. A.; Scott-Brown, J.; Vaidyanathan, P.; Fontanarrosa, P.; Oberortner, E.; Wipat, A. et al. The Synthetic Biology Open Language (SBOL) Version 3: Simplified Data Exchange for Bioengineering. *Frontiers in Bioengineering and Biotechnology* **2020**, *8*.

(30) Maccagnan, A.; Riva, M.; Feltrin, E.; Simionati, B.; Vardanega, T.; Valle, G.; Cannata, N. Combining ontologies and workflows to design formal protocols for biological laboratories. *Automated Experimentation* **2010**, *2*, 3.

(31) Curty, R. G.; Crowston, K.; Specht, A.; Grant, B. W.; Dalton, E. D. Attitudes and norms affecting scientists' data reuse. *PLOS ONE* **2017**, *12*, e0189288.

(32) Pasquetto, I. V.; Borgman, C. L.; Wofford, M. F. Uses and Reuses of Scientific Data: The Data Creators' Advantage. *Harvard Data Science Review* **2019**, *1*.

(33) Zuiderwijk, A.; Shinde, R.; Jeng, W. What drives and inhibits researchers to share and use open research data? A systematic literature review to analyze factors influencing open research data adoption. *PLOS ONE* **2020**, *15*, e0239283.

(34) Costello, M. J.; Michener, W. K.; Gahegan, M.; Zhang, Z.-Q.; Bourne, P. E. Biodiversity

data should be published, cited, and peer reviewed. *Trends in Ecology & Evolution* **2013**, *28*, 454–461.

(35) Borgman, C. L. The conundrum of sharing research data. *Journal of the American Society for Information Science and Technology* **2012**, *63*, 1059–1078.

(36) Frey, K.; Hafner, A.; Pucker, B. *The Reuse of Public Datasets in the Life Sciences: Potential Risks and Rewards*; 2020.

(37) Booth, I. R. SysMO: back to the future. *Nature Reviews Microbiology* **2007**, *5*, 566–566.

(38) Spellman, P. T.; Miller, M.; Stewart, J.; Troup, C.; Sarkans, U.; Chervitz, S.; Bernhart, D.; Sherlock, G.; Ball, C.; Lepage, M. et al. Design and implementation of microarray gene expression markup language (MAGE-ML). *Genome Biology* **2002**, *3*, research0046.1.

(39) Wolstencroft, K.; Owen, S.; Horridge, M.; Krebs, O.; Mueller, W.; Snoep, J. L.; du Preez, F.; Goble, C. RightField: embedding ontology annotation in spreadsheets. *Bioinformatics* **2011**, *27*, 2021–2022.

(40) Urquiza-García, U.; Zieliński, T.; Millar, A. J. Better research by efficient sharing: evaluation of free management platforms for synthetic biology designs. *Synthetic Biology* **2019**, *4*, ysz016.

(41) Zieliński, T.; Hay, J.; Romanowski, A.; Nenninger, A.; McCormick, A.; Millar, A. J. SynBio2Easy—a biologist-friendly tool for batch operations on SBOL designs with Excel inputs. *Synthetic Biology* **2022**, *7*, ysac002.

(42) Mante, J.; Zundel, Z.; Myers, C. Extending SynBioHub's Functionality with Plugins. *ACS Synthetic Biology* **2020**, *9*, 1216–1220.

# Graphical TOC Entry