SBOL: A community standard for communicating designs in synthetic biology

Michal Galdzicki, Mandy L. Wilson, Cesar A. Rodriguez, Ernst Oberortner, Matthew Pocock, Laura Adam, J. Christopher Anderson, Bryan A. Bartley, Jacob Beal, Deepak Chandran, Joanna Chen, Douglas Densmore, Drew Endy, Raik Grünberg, Jennifer Hallinan, Nathan J. Hillson, Haiyao Huang, Jeffrey D. Johnson, Allan Kuchinsky, Matthew Lux, Goksel Misirli, Chris J. Myers, Jean Peccoud, Hector A. Plahar, Nicholas Roehner, Evren Sirin, Guy-Bart Stan, Alan Villalobos, Anil Wipat, John H. Gennari, Herbert M. Sauro

Abstract

The Synthetic Biology Open Language (SBOL) is a proposed data standard for exchanging designs within the synthetic biology community. SBOL represents synthetic biology designs in a community-adopted, formalized format for exchange between software tools, research groups, and commercial service providers. The re-use of previously validated designs is critical to the evolution of synthetic biology from a research discipline to an engineering practice. As a community-driven standard, SBOL adapts as synthetic biology evolves, providing specific capabilities for different aspects of the synthetic biology workflow. The SBOL Developers Group has implemented SBOL 1.1 as an XML/RDF serialization and provides software libraries and specification documentation to help developers implement SBOL in their own software. This paper also reports on early successes, including a demonstration of the utility of SBOL for information exchange between three different tools from three academic sites.

Introduction

Synthetic biology treats biological systems as a new technological medium with a unique set of characteristics, such as the ability to self-repair, evolve, and replicate. These characteristics create their own engineering challenges, but offer a rich and largely untapped source of potential applications to benefit society[1,2]**.** Applications such as bio-molecular computing[3], metabolic engineering[4], or the re-construction and exploration of natural cell biology[5,6] commonly requires the design of new genetically encoded systems. As engineers, synthetic biologists most often base their design on previously described segments of DNA to meet their project's requirements.

Every engineering field relies on a set of *standards* that practitioners follow[7] (Terms in *italics* are defined in Supplementary Table 1). The representation of synthetic biology designs in a computer-readable format, and the exchange of corresponding information using various tools, facilitates forward-engineering and re-use of components in new biological applications. Such a *data standard* enables an engineer to develop portions of a design on one software tool, and then refine it in another tool, or to transmit that design electronically to a colleague or commercial fabrication company. Moreover, synthetic biology companies could offer catalogs of devices or parts via computer-readable data sheets, just as modern semiconductor companies do with electronic devices.

A *standard exchange format* for these designs would dramatically improve the ability to reproduce published results[8]. Currently, it is extremely difficult to extract workable designs from literature because designs are usually described using imprecise (and error prone) English prose. All too often, critical information is accidentally omitted or implicitly assumed, and critical data, such as the exact DNA sequence, are simply not available.

Unfortunately, although there are standards for experimentally measuring the key processes of synthetic biological parts[9,10], and standards for the construction of composite DNA[11] for synthetic biology, description of the designs themselves are not standardized. Further, standard file formats for importing and exporting DNA sequences, (such as FASTA[12], GenBank's flat file format[13] and GFF[14]**)**, cannot easily be adapted to accommodate the unique requirements of synthetic biology design. (For a specific comparison of the differences between GenBank and SBOL, see Supplementary Table 2.) Synthetic biology is about synthesis of novel DNA, rather than sequencing an extant molecule.

This paper describes our proposed standard for synthetic biology, the *Synthetic Biology Open Language* (SBOL), for the representation of synthetic biology designs. Our long-term goal is to increase productivity in the design, building, testing, and dissemination of synthetic biological systems. The SBOL Developers Group is designing this standard to meet the specific needs of synthetic biologists.

As with the design of electronic circuits, synthetic biology designs are composed hierarchically from sets of reusable DNA components. Typically these components (ie, *promoters*, *protein coding sequences* (CDS), or *transcriptional terminators)* are defined in terms of the functions they perform, in a defined context. Reusability requires that such functional definitions be unambiguous. The supplier of a DNA component library and the designer who uses parts from that library must both use the same term to describe, for example, a CDS. No ambiguity must exist as to whether the CDS includes a starting codon. The meaning must be made explicit by the definition of the term, so that it is used consistently.

Another aspect of synthetic biology design is its iterative nature. At the beginning stages of a design, a synthetic biologist may not yet have a specific DNA sequence chosen. Therefore, the specific sequence of a DNA component should be optional and specified at a later stage of the engineering process. The hierarchical composition of synthetic biology designs allows for a mix of *DNA segments* with specified and unspecified sequences, permitting the designer to assign the sequences as the design matures, and to exchange partial specifications with collaborators. Early stage design may only capture partial-order relationships among segments of DNA. If a standard requires a premature ordering of a design that is more precise than is required, it could lead to unexpected dependencies and design flaws.

Finally, in order for synthetic biology design to scale up, researchers must make use of specialized synthetic biology design tools and libraries. These tools must be able to easily incorporate a standard means for communication across tools. For example, a gene network design created by tools such as the Proto Biocompiler[15] could be exported to and then simulated with iBioSim[16]. The establishment and wide adoption of a standard would allow a growing number of software tools to more directly support the integrated design workflow[17] of the synthetic biologist in research and commercial institutions.

In addition to describing SBOL, this paper also presents preliminary work that demonstrates the broad benefit of SBOL and SBOL-compliant tools to the community. As described below, SBOL allows engineers to select components from an SBOL-compliant library, to develop portions of a design on one tool, and then to send that design to a different tool (at a different institution, or with different analysis capabilities) for further development, and then finally to send the design to other colleagues or perhaps a commercial fabrication company.

The SBOL Standard

SBOL's foundation is the *core data model*. This central core was first released in November, 2011; Version 1.1.0 was released and ratified in October 2012. SBOL Core provides the central module for the specification of DNA-level designs. SBOL Core defines DNA segments as *DNA components,* and allows for their hierarchical composition, making it possible to include the substructure and lineage of each design element. SBOL Core also includes a *collection* data structure, which allows researchers to group DNA components into meaningful libraries or catalogs of components. SBOL Core leverages prior work in the development of the *Sequence Ontology* (SO)[17,18], a controlled vocabulary with a strictly defined set of concepts and relationships for DNA sequences involved in a biological process. SBOL uses SO terms to unambiguously label components in a design. Figure 1(a) shows an example toy design, with a hierarchical arrangement of components, each of which is labeled with an SO term to indicate its function.

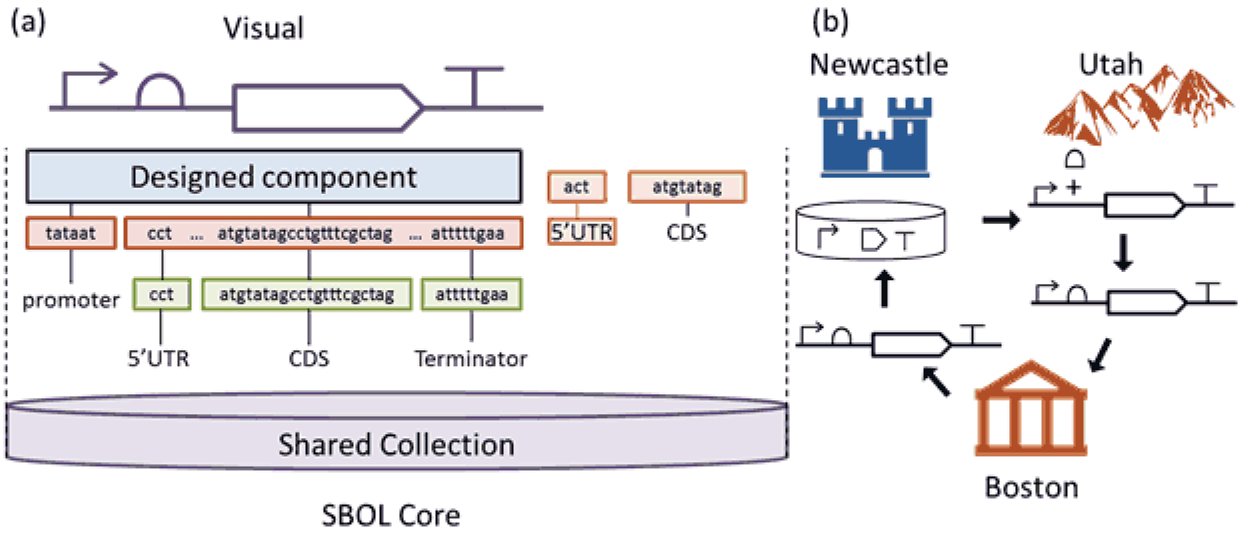

Figure 1. The SBOL core data model and demonstration of inter-institutional exchange. (a) The core data model defines DNA components and other fundamental objects of synthetic biology designs. A DNA component defines the design of a segment of DNA in terms of its required sub-components, their sequential arrangement (e.g., that one component must precede another) and, where known, its DNA sequence. This strategy allows us to specify: 1. designs in which the sequence is undefined, partially defined, or fully defined; 2. hierarchical compositions of components; 3. unambiguously defined component types using Sequence Ontology (SO) terms[18,19]; and 4. collections of components for distribution to recipients. SBOL Visual[20] enables the depiction of genetic designs in a standard graphical notation. **(b)** Demonstration of exchange from Newcastle to Utah to Boston, and back to Newcastle.

SBOL Core provides a specification document, use cases, and software support. The specification[21] describes in detail the data model and the requirements of the standard. The use cases describe the stakeholder goals for data exchange of synthetic biology designs. The software support consists of libSBOLj, a Java library designed for developers to easily incorporate SBOL support into their tools. (A library for the C language is also under development.) See Table 1 for a list of software tools that support SBOL.

Table 1. List of tools that support SBOL.

	Application	SBOL	Description	Affiliation	URL	Citation
1	GenoCAD	visual	Design of DNA sequences using a grammar-based methodology.	VBI	http://www.genocad.org	24
2	DeviceEditor	visual	A visual biological CAD canvas, front-end for j5.	JBEI/LBNL	http://j5.jbei.org	33
3	Pigeon	visual	Design visualizer	BU	http://pigeoncad.org/	34
4	Proto BioCompiler	visual, core	Automated design of genetic regulatory networks from high-level programs.	BBN	http://synbiotools.bbn.com/	15
5	Graphviz	visual	Graph visualization software	AT&T Research	http://www.graphviz.org/	25
6	SBPkb	core	Semantic information retrieval from Registry of Standard Biological Parts.	UW	http://www.sbolstandard.org/sbol-in-use/sbpkb	35
7	Eugene	core	Language for the composition of parts into novel biological devices.	BU	http://www.eugenecad.org	36
8	Hermes	core	Import and export of SBOL for the Clotho platform.	BU		30
9	j5	core	Automates the design of DNA assembly protocols.	JBEI/LBNL	http://j5.jbei.org	37
10	GenBank Converter	core	Interconverts SBOL and GenBank format files.	JBEI/LBNL	http://j5.jbei.org/bin/sbol_converter_entry_form.pl	
11	JBEI-ICE	core	Repository for DNA sequences, microbial strains, and Arabidopsis seeds.	JBEI/LBNL	https://public-registry.jbei.org	31
12	iBioSim	core	Automates design, simulation, checking, and abstraction of genetic circuit models.	University of Utah	http://www.async.ece.utah.edu/iBioSim/	38
13	Gene Designer	core	DNA design tool.	DNA2.0	https://www.dna20.com/genedesigner2/	39
14	BacilloBricks	core	Catalog of parts and their composable models.	Newcastle University	http://www.bacillobricks.co.uk	
15	MoSeC	core	Automates the derivation of DNA sequences from models.	Newcastle University	http://intbio.ncl.ac.uk/?projects=mosec	40
16	Registry of Standard Biological Parts	core	Collection of standardized genetic parts, via SBOL Converter.	MIT	http://partsregistry.org/	
17	VectorEditor	core	Viewing, annotating and *in silico* cloning of sequences	JBEI/LBNL	https://public-registry.jbei.org/static/vesa/VectorEditor.html	31
18	Tinker Cell	visual, core	CAD tool for synthetic biology, via WikiDust.	UW		23
19	GSL	core	Internal language	Amyris		41
20	Vector NTI®	core	Sequence analysis and design tools for molecular biology data	Life Tech	http://www.invitrogen.com/site/us/en/home/Products-and-Services/Applications/Cloning/v	

					ector-nti-software.html
21	SBOL Designer	visual, core	Create SBOL designs using SBOL visual icons and Geneious plugin	Clark & Parsia	http://clarkparsia.github.io/sbol

The *core data model* is an abstract model that can (for example) be described via *Unified Modeling Language* (UML), see the SBOL specification document[21]. However, to enable communication between systems, SBOL must also specify a consistent serialization. Currently, the community has adopted a strictly defined *XML serialization* (http://www.sbolstandard.org/initiatives/serialization), which also conforms to the *RDF standard* (http://www.w3.org/RDF/). Our goal is to allow both *XML-Schema* (XSD) compliant tools and RDF tools to use the same serialization. Necessarily, this means SBOL uses a subset of the standard RDF format that can also be expressed in XSD. Our use of RDF enables the unique identification of DNA elements, annotations, and collections across the World Wide Web. In particular, by associating world-wide unique IDs, via URIs, information about designs can be linked across institutions[22]. Association of unique IDs, via URIs, with information about DNA components means that design information from different institutions can be linked. Therefore, design tools can automatically aggregate part information from various sources. For example, an institute could add quality control and characterization data to a parts collection while retaining their connection to the original parts description, thereby removing problems of maintaining coherence between mirrored or forked data repositories. URIs and the SBOL serialization also make it easier to re-use a DNA component in a new design while retaining its identity and information about its properties.

SBOL Extensions

SBOL Core represents the structure and function of DNA sequences. A complete description of a synthetic biology design also needs to represent other perspectives on the design, such as the dynamic behavior of the overall system, and the context of the host organism into which the design is introduced. To this end, SBOL Core is being augmented with a set of optional modules, called *extensions,* intended to augment the core with additional data models for *assembly, modeling, visualization, experimental data, context*, and other information about a synthetic biology design necessary for its fabrication, verification, and deployment. The extension mechanism allows the standard to evolve as synthetic biology grows, and it provides specific capabilities for different aspects of the synthetic biology workflow. At present, the SBOL Developers Group has identified three key extensions that SBOL requires. Additionally, working groups of interested stakeholders have been formed to investigate the scope and requirements of the extensions.

The **SBOL Visual Extension** (http://www.sbolstandard.org/community/sbol-working-groups/visual) provides a graphical representation of the information encoded in SBOL Core, in the form of a set of icons that can be used in graphical tools. SBOL Visual is the most mature of the extensions; Version 1.0.0[20] of SBOL Visual is already in use in a number of software tools, such as Tinkercell[23], GenoCAD[24], and GraphViz[25]. For a complete list see Table 1.

The **SBOL Context Extension** (http://www.sbolstandard.org/community/sbol-working-groups/hostcontext) describes the host organism used to realize the synthetic biology design, and the environment under which it must operate in order to achieve a functional device. The context extension provides information about the physical context, including the strain of the host, the medium in which the host resides, the container in which the medium is stored, the environmental conditions, and the measurement device used to study the context. Precise details about the experimental context are essential to the reproducibility of laboratory results.

The **SBOL Modeling Extension** (http://www.sbolstandard.org/community/sbol-working-groups/modelling) provides a mechanism for linking computational models to SBOL designs[26]. In this way, the modeling extension leverages the significant work done in the development of standards for modeling biological systems, such as the *Systems Biology Markup Language* (SBML)[27]. The extension identifies the modeling language (SBML[27], CellML[28], MATLAB, BNGL[29], etc.) of the linked model, as well as its modeling framework (ODE, Stochastic, Boolean, etc.). Additionally, the extension can document qualitative interactions between components in a design, e.g. the interaction of a transcription factor with a promoter. Each interaction includes terms from the *Systems Biology Ontology* (SBO) to specify its type (repression, activation, etc.) and the roles (repressor, activator, etc.) played by its participating components.

In order to connect these extensions with SBOL Core, the SBOL Developers Group has proposed extending the core with additional data structures for *devices* and *systems,* as well as, generalizing the notion of components to encompass protein and RNA components, in addition to DNA components. Devices gather components and sub-devices on the basis of shared function, while systems gather devices on the basis of shared context and models for their behavior. Figure 2 summarizes these proposed extensions and how they connect with SBOL Core.

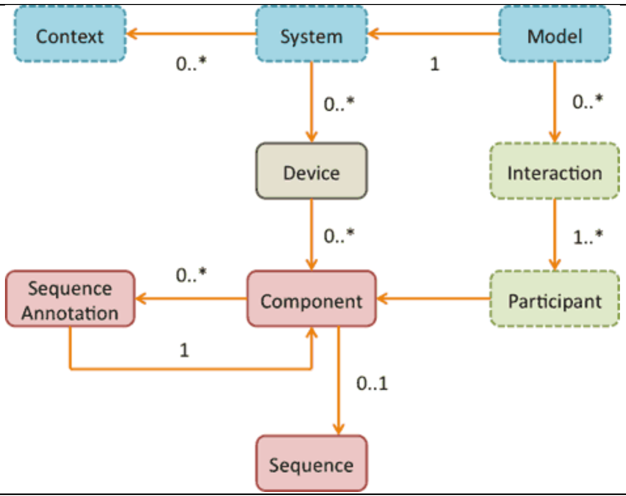

Figure 2. Extensions to the core data model and how they connect to each other. This diagram indicates the relationship between the different classes of information in a SBOL representation. At the head of the diagram is the 'System' class. Boxes in dotted outline represent proposed additions to the standard. The numerals indicate the number of possible relationships between classes. For example, a 'Model' must belong to one and only 'System', whereas a 'Component' can have none or any number of 'Sequence Annotations' attached to it. See main text for further details.

SBOL Demonstration

As a proof-of-concept, the SBOL Developers Group demonstrated an initial, relatively simple use of the SBOL standard to exchange data among three groups of researchers, each at separate university environment, and each using separate tools. This demonstration provided an important community experience by demonstrating buy-in and reassuring all that there is support for transmitting SBOL designs across environments and tools. The three teams and tools are iBioSim[16] at the University of Utah, Clotho[30] at Boston University, and Bacillo Bricks (http://www.bacillobricks.co.uk) at Newcastle University. The first step is that each of these tools has to incorporate SBOL into their software; a current list of tools that support SBOL is presented in Table 1.

Starting with design, the Utah team planned to produce a simple expression cassette: a promoter, an RBS, a coding sequence, and a terminator. As a first step, they selected three components from the Bacillo Bricks repository. These were transmitted using the SBOL standard from Newcastle to Utah, thereby demonstrating export and import capabilities from software at both Utah and Newcastle. Within iBioSim, the Utah team designed an appropriate RBS DNA component to create a complete cassette as a design template. Next, iBioSim exported this design template via SBOL to the Clotho tool at Boston. The Boston team added additional sequence information needed for construction, and then the completed cassette was sent back to Newcastle's Bacillo Bricks library, thereby storing the improved, completed cassette for others to use and build from. Figure 1(b) shows a diagrammatic view of this round-trip information transfer.

This demonstration illustrates three use cases: (a) using SBOL to represent abstract DNA design templates for sharing, (b) adding sequence information to those templates, and then using SBOL to represent and share those completed designs, and (c) using SBOL for organizing and publishing repositories, or collections, of DNA components. This transfer of SBOL information across a variety of tools and institutions offers a clear vision for how the field could benefit from the broad adoption of the SBOL standard. First, SBOL allows researchers to retrieve information in a consistent and unambiguous manner from a variety of libraries—not only Bacillo Bricks, but other resources such as the Registry of Standard Biological Parts (http://partsregistry.org) or JBEI-ICE[31]. Second, researchers can work with a wide variety of software tools for a variety of steps in the workflow. As long as all tools support the SBOL standard, the researcher has great flexibility in collaborating with others and with tool use.

Community

SBOL is an open standard in that participation in standardization activities is unrestricted to all affected interests[32], essential information is publicly accessible on the web, and it can be used without cost. Additionally, as the needs of the community evolve, SBOL is also open to change. Community engagement, and a democratic decision-making process, steers the standard so that no one's interest dominates its development. It is developed and agreed upon by a diverse group of stakeholders from the synthetic biology community. SBOL has been under development by the Synthetic Biology community since 2008.

SBOL community engagement and outreach efforts have been inspired by the tremendous success of SBML. The SBOL Developers Group took advantage of some crucial "lessons learned" from SBML, and applied these to SBOL. Early engagement with young scientists and regular meetings are very important for building up the excitement and consensus within the community. The community holds a minimum of two meetings per year to encourage familiarity with the field and to develop trust among the participants. The most recent meetings were held at Boston University (November 2012) and at Newcastle University (April 2013).

At the heart of the SBOL community is the SBOL Developers Group, a diverse group of researchers, developers, and other stakeholders from academic, government, and commercial organizations. At this writing, the SBOL community has more than 80 delegates from 29 organizations (16 academic, 11 commercial, and 2 government labs), who work across organizational and international boundaries to set priorities and reach agreement on the standard.

To facilitate the ongoing standardization process and the development of extensions, the SBOL community has developed a formal governance structure. The SBOL effort is coordinated by a group of five elected editors under the guidance of an elected SBOL Chair. The editors represent the diverse backgrounds of the SBOL community, and serve two-year terms. They are responsible for documentation and community organization, while the SBOL Chair helps coordinate the overall process and funding. The SBOL Editors monitor changes, proposals, and other elements of interest, and process requests for revisions to the SBOL specifications. Other community members can also submit amendments to the SBOL Editors after discussion within the SBOL Developers Group.

Future Developments, Limitations, and Conclusions

This paper describes SBOL, a proposed standard for representing designs in synthetic biology. Since its inception in 2008, the SBOL community has grown to include academic, government, and commercial organizations. This paper reports on the latest iteration of SBOL, designated Version 1.1.0. This version allows engineers to specify an unambiguous description of a DNA design in a hierarchical and fully annotated form. However, the complete specification of a design requires much more information than simply the DNA sequence. For this reason, SBOL has been designed to be an extensible format allowing additional information to be included as the synthetic biology field develops. Several extensions are under active development, including a visual extension for graphical representation of genetic designs, a modeling extension for linking to computational models, and a context extension to specify strain, environmental conditions, etc. This last extension coupled with SBOL Core has the potential to greatly aid the reproduction of experimental results, a primary goal of this project.

The aim of SBOL is to increase productivity in the design, building, testing, and dissemination of synthetic biological systems. As one way to improve productivity, SBOL encourages and makes easier the description and sharing of designs via libraries such as Bacillo Bricks or JBEI-ICE[31]. A standard such as SBOL improves the reproducibility of results, since SBOL files could be provided as supplementary material to journal articles, which would allow other researchers to more easily build from prior work.

To facilitate adoption of SBOL, the community has developed a written specification document and associated software libraries to enable third-party developers to include SBOL in their workflow and software tools. As of this writing, SBOL has been adopted by twenty software tools that include both commercial and academic efforts. As SBOL continues to mature, the SBOL Developers Group will add extensions to handle an increasing range of the knowledge needed to reliably exchange and reproduce synthetic biology designs. For research to progress at full speed, the development of standards such as SBOL is essential for the translation of synthetic biology research into practice.

Acknowledgements

This work was partially supported by grants from the Microsoft Computational Challenges in Synthetic Biology Initiative (2006), AutoDesk, National Science Foundation (0527023, 1147158 and CCF-1218095), National Library of Medicine (R41 LM010745, T15 LM007442), National Human Genome Research Institute (R42 HG006737), Agilent Technologies' Applications and Core Technology University Research (ACT-UR) program, DARPA (HR0011-10-C- 0168), and the EPSRC-funded Flower Consortium project (EP/J02175X/1). The portion of this work conducted by the Joint BioEnergy Institute was supported by the Office of Science, Office of Biological and Environmental Research, of the U.S. Department of Energy (Contract No.DE-AC02-05CH11231). The views and conclusions contained in this document are those of the authors and not the U.S. Government or any agency thereof.

Contributions

All authors helped develop the SBOL standard by contributing to the specification document and by participating in workshops or on the SBOL Developers mailing list and/or wrote software that supports SBOL. See Supplementary Table 3 for a full list of author contributions.

References

1. Khalil, A. S. & Collins, J. J. Synthetic biology: applications come of age. *Nature reviews. Genetics* **11**, 367–79 (2010).
2. Keasling, J. The Promise of Synthetic Biology. *The Bridge* **35**, 18–21 (2005).
3. Benenson, Y. Biomolecular computing systems: principles, progress and potential. *Nature Reviews Genetics* (2012).
4. Woolston, B. M., Edgar, S. & Stephanopoulos, G. Metabolic Engineering: Past and Future. *Annual review of chemical and biomolecular engineering* (2013).doi:10.1146/annurev-chembioeng-061312-103312
5. Nandagopal, N. & Elowitz, M. B. Synthetic biology: integrated gene circuits. *Science* **333**, 1244–1248 (2011).
6. Mukherji, S. & Van Oudenaarden, A. Synthetic biology: understanding biological design from synthetic circuits. *Nat Rev Genet* **10**, 859–871 (2009).
7. Slattery, W. J. An index of U.S. voluntary engineering standards; covering those standards, specifications, test methods, and recommended practices issued by national standardization organizations in the United States. *United States. National Bureau of Standards. Office of Engineering Standards Services. Information Section* **no.329**, (1971).

8. Peccoud, J. *et al.* Essential information for synthetic DNA sequences. *Nature Biotechnology* **29**, 22 (2011).
9. Canton, B., Labno, A. & Endy, D. Refinement and standardization of synthetic biological parts and devices. *Nature Biotechnology* **26**, 787–93 (2008).
10. Kelly, J. R. *et al.* Measuring the activity of BioBrick promoters using an in vivo reference standard. *Journal of biological engineering* **3**, 4 (2009).
11. Müller, K. M. & Arndt, K. M. Standardization in synthetic biology. *Synthetic Gene Networks: Methods and Protocols, Methods in Molecular Biology* **813**, 23–43 (2012).
12. Pearson, W. R. & Lipman, D. J. Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences* **85** , 2444–2448 (1988).
13. The DDBJ/EMBL/GenBank Feature Table Definition Version 10.2. *INSDC* (2012).at <http://www.insdc.org/documents/feature-table>
14. Stein, L. *Generic Feature Format Version 3 (GFF3) Version: 1.21*. (2013).at <http://www.sequenceontology.org/gff3.shtml>
15. Beal, J., Lu, T. & Weiss, R. Automatic Compilation from High-Level Biologically-Oriented Programming Language to Genetic Regulatory Networks. *PLoS ONE* **6**, e22490 (2011).
16. Madsen, C. *et al.* Design and Test of Genetic Circuits using iBioSim. *Design & Test of Computers, IEEE* 1 (2011).
17. Beal, J. *et al.* An End-to-End Workflow for Engineering of Biological Networks from High-Level Specifications. *ACS Synthetic Biology* **1**, 317–331 (2012).
18. Eilbeck, K. *et al.* The Sequence Ontology: a tool for the unification of genome annotations. *Genome biology* **6**, R44 (2005).
19. Mungall, C. J., Batchelor, C. & Eilbeck, K. Evolution of the Sequence Ontology terms and relationships. *Journal of biomedical informatics* (2010).doi:10.1016/j.jbi.2010.03.002
20. Quinn, J. *et al.* Synthetic Biology Open Language Visual (SBOL Visual), version 1.0.0. *BBF RFC #93* (2013).doi:1721.1/78249
21. Galdzicki, M. *et al.* Synthetic Biology Open Language (SBOL) Version 1.1.0. *BBF RFC #87* (2012).doi:1721.1/73909
22. Bizer, C., Heath, T. & Berners-Lee, T. Linked data-the story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)* **5**, 1–22 (2009).
23. Chandran, D., Bergmann, F. T. & Sauro, H. M. TinkerCell: modular CAD tool for synthetic biology. *Journal of Biological Engineering* **3**, (2009).
24. Cai, Y., Wilson, M. L. & Peccoud, J. GenoCAD for iGEM: a grammatical approach to the design of standard-compliant constructs. *Nucleic Acids Research* **38**, 2637–2644 (2010).
25. Ellson, J., Gansner, E. R., Koutsofios, E., North, S. C. & Woodhull, G. Graphviz and dynagraph—static and dynamic graph drawing tools. *Graph Drawing Software* 127–148 (2004).
26. Cai, Y., Lux, M. W., Adam, L. & Peccoud, J. Modeling structure-function relationships in synthetic DNA sequences using attribute grammars. *PLoS computational biology* **5**, e1000529 (2009).
27. Hucka, M. *et al.* The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* **19**, 524–531 (2003).
28. Lloyd, C. M., Halstead, M. D. B. & Nielsen, P. F. CellML: its future, present and past. *Progress in biophysics and molecular biology* **85**, 433–50 (2004).
29. Faeder, J. R., Blinov, M. L. & Hlavacek, W. S. Rule-based modeling of biochemical systems with BioNetGen. *Systems Biology* 113–167 (2009).
30. Xia, B. *et al.* Developer’s and user's guide to Clotho v2.0 A software platform for the creation of synthetic biological systems. *Methods in enzymology* **498**, 97–135 (2011).
31. Ham, T. S. *et al.* Design, implementation and practice of JBEI-ICE: an open source biological part registry platform and tools. *Nucleic Acids Res* **40**, e141 (2012).

32. United States Standards Strategy Committee *The United States Standards Strategy*. (American National Standards Institute: 2010).

33. Chen, J., Densmore, D., Ham, T. S., Keasling, J. D. & Hillson, N. J. DeviceEditor visual biological CAD canvas. *Journal of biological engineering* **6**, 1 (2012).

34. Bhatia, S. & Densmore, D. Pigeon: A Design Visualizer for Synthetic Biology. *ACS synthetic biology* (2013).doi:10.1021/sb400024s

35. Galdzicki, M., Rodriguez, C., Chandran, D., Sauro, H. M. & Gennari, J. H. Standard Biological Parts Knowledgebase. *PLoS ONE* **6**, e17005 (2011).

36. Bilitchenko, L. *et al.* Eugene - A domain specific language for specifying and constraining synthetic biological parts, devices, and systems. *PLoS one* **6**, e18882 (2011).

37. Hillson, N. J., Rosengarten, R. D. & Keasling, J. D. j5 DNA Assembly Design Automation Software. *ACS Synthetic Biology* **1**, 14–21 (2011).

38. Myers, C. J. *et al.* iBioSim: a tool for the analysis and design of genetic circuits. *Bioinformatics* **25**, 2848–2849 (2009).

39. Villalobos, A., Ness, J. E., Gustafsson, C., Minshull, J. & Govindarajan, S. Gene Designer: a synthetic biology tool for constructing artificial DNA segments. *BMC bioinformatics* **7**, 285 (2006).

40. Misirli, G. *et al.* Model annotation for synthetic biology: automating model to nucleotide sequence conversion. *Bioinformatics (Oxford, England)* **27**, 973–9 (2011).

41. Platt, D. Personal Communication: Thanks! [re: San Francisco meeting]. sbol–dev@googlegroups.com (2012).