SBOL: A community standard for communicating designs in synthetic biology

Authors: Michal Galdzicki, Kevin P. Clancy, Ernst Oberortner, Matthew Pocock, Jacqueline Quinn, Cesar A. Rodriguez, Nicholas Roehner, Mandy L. Wilson, 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, Jeffrey D. Johnson, Allan Kuchinsky, Matthew Lux, Goksel Misirli, Jean Peccoud, Hector A. Plahar, Evren Sirin, Guy-Bart Stan, Alan Villalobos, Anil Wipat, John H. Gennari, Chris J. Myers, Herbert M. Sauro

Date: November 8, 2013

Correspondence should be addressed to Herbert Sauro (hsauro@uw.edu)

Information exchange technology utilized

The *core data model* is described in the specification document[1] using the *Unified Modeling Language* (UML). To enable communication between software tools, however, SBOL must specify a consistent serialization mechanism. 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. Design tools can then 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 part descriptions, 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.

The core data model serves as connecting point for extensions. Optional modules can be added and are ignored by default in software applications that do not support a specific extension. This behavior allows an SBOL compliant software tool to gracefully bypass an unsupported extension. The proposed extensions in context of the core data model are illustrated in Supplementary Figure 2.

We have implemented SBOL 1.1 in the form of XML/RDF serialization and provided software libraries and specification documentation for developers. For any standard, tool development that makes

adoption easy is critical; we therefore provide the reference software library implementation for developers to use in their tools or base their own implementation on. The libSBOLj library can be found on the GitHub source code repository here: https://github.com/SynBioDex/libSBOLj

The specific revision of the source code at the time of the writting can be found here: https://github.com/SynBioDex/libSBOLj/tree/NBT

Supplementary Table 1. Glossary

Term	Definition
assembly [extension]	Physical construction of DNA molecules.
Collection	"Organizational container, a group of DnaComponents."[1]
context [extension]	Physical and experimental context necessary to replicate the function of a synthetic biological organism
core data model	"Representation of computer system objects together with their properties and relationships" (http://en.wikipedia.org/wiki/Data_model). We use this term in SBOL to distinguish the core from extensions and to emphasize its fundamental role in the standard.
data standard	Standardized data model for a subject domain, i.e. synthetic biology. In this manuscript we describe a data standard in contrast to other standards being developed in the field of synthetic biology, such as reference, functional, and composition standards[2,3].
devices	(Re)usable portion of functional or behavioral design, enabling annotation, composition and experimental characterization.
DNA component	"A DNA component represents a segment of DNA that serves to abstract the DNA sequence as an individual object, which can then be manipulated, combined, and reused in engineering new biological systems."[1]
DNA segments	Within SBOL, we consider DNA segments as elements of design for DNA circuits4, analogous to electrical circuits5. This conceptualization of DNA segments as an element of design is a level of abstraction used to form the basis of engineering synthetic biological systems[4,5].
experimental data [extension]	Data and information about measurements to test performance of a device or system.
extensions	Optional modules added to the core data model.
modeling [extension]	Qualitative and quantitative behavior of devices and systems, especially the interactions between the components used in a design.
precedes	relation specifies the relative order of SequenceAnnotations for a given

	DnaComponent. It is a constraint on the order of subComponents when there is not enough information to specify exact positions. For example, to say the promoter SequenceAnnotation precedes the CDS SequenceAnnotation.
promoter	"A regulatory_region composed of the TSS(s) and binding sites for TF_complexes of the basal transcription machinery." [6]
protein coding sequence (CDS)	"a contiguous sequence which begins with and includes a start codon and ends with and includes a stop codon."[6]
RDF standard	"Standard model for data interchange on the Web. RDF has features that facilitate data merging even if the underlying schemas differ, and it specifically supports the evolution of schemas over time without requiring all the data consumers to be changed" (http://www.w3.org/RDF/).
Sequence Ontology (SO)	"Set of terms and relationships used to describe the features and attributes of biological sequence" (http://www.sequenceontology.org/)[6,7]
standard exchange format	Standardized data format to convey information in a consistent, computer-readable format so that it can be read directly by applications. SBOL defines a strict XML format, compatible with RDF, to enable exchange of synthetic biology designs.
standards	Set of agreed upon and adhered to technical definitions and guidelines created with the goal to increase industrial and scientific productivity[8].
Synthetic Biology Open Language (SBOL)	SBOL is a data standard in synthetic biology used to represent designs in such a way that is compatible with a variety of applications to foster easy exchange. SBOL provides both a standard data model and exchange format to represent and consistently transfer designs among scientists. SBOL visual is used to diagram such designs.
systems	Structure, behavior, and relationships of interacting components forming an integrated whole.
Systems Biology Markup Language (SBML)	"Interchange format for computer models of biological processes". (http://sbml.org)
Systems Biology Ontology (SBO)	"Set of controlled, relational vocabularies of terms commonly used in Systems Biology, and in particular in computational modeling". (http://www.ebi.ac.uk/sbo/main/)
transcriptional terminator	"The sequence of DNA located either at the end of the transcript that causes RNA polymerase to terminate transcription." [6]
Unified Modeling Language (UML)	"Standardized (ISO/IEC 19501:2005), general-purpose modeling language in the field of software engineering. The Unified Modeling Language includes a set of graphic notation techniques to create visual models of object-oriented

	software-intensive systems." (http://en.wikipedia.org/wiki/Unified_Modeling_Language)	
visualization [extension]	Graphical notation that supports the description and specification for communicating designs.	
XML serialization	Process of converting data to an XML format that can be stored in a file or transmitted across a computer network. For more details see (http://en.wikipedia.org/wiki/Serialization).	
XML-Schema (XSD)	"XML Schemas express shared vocabularies and allow machines to carry out rules made by people. They provide a means for defining the structure, content and semantics of XML documents." (http://www.w3.org/XML/Schema)	

Supplementary Table 2. Differences and similarities of SBOL and GenBank flat file format

Both GenBank format and SBOL		
Describe a contiguous piece of sequence		
Indicate areas of biological significance, such as coding regions, transcription units, etc.		
Can be used as a file to export and import sequence		
GenBank format	**SBOL**	**Comment**
Existing molecule sequenced.	New designed sequence.	The use cases each was developed to satisfy are different. GenBank format was made for the results of DNA sequencing, while SBOL is made for designing a new DNA sequence. For example, SBOL DNA Components are re-used to specify a new design while retaining the identity and information about properties of its subcomponents.
Maintained by established consortium, the International Nucleotide Sequence Database	New development and maintenance by the SBOL Developers Group, to support data exchange	While, it is possible to add new conventions within GenBank formatted files, *ad hoc* additions would make the approach fragile. For example, software which does not

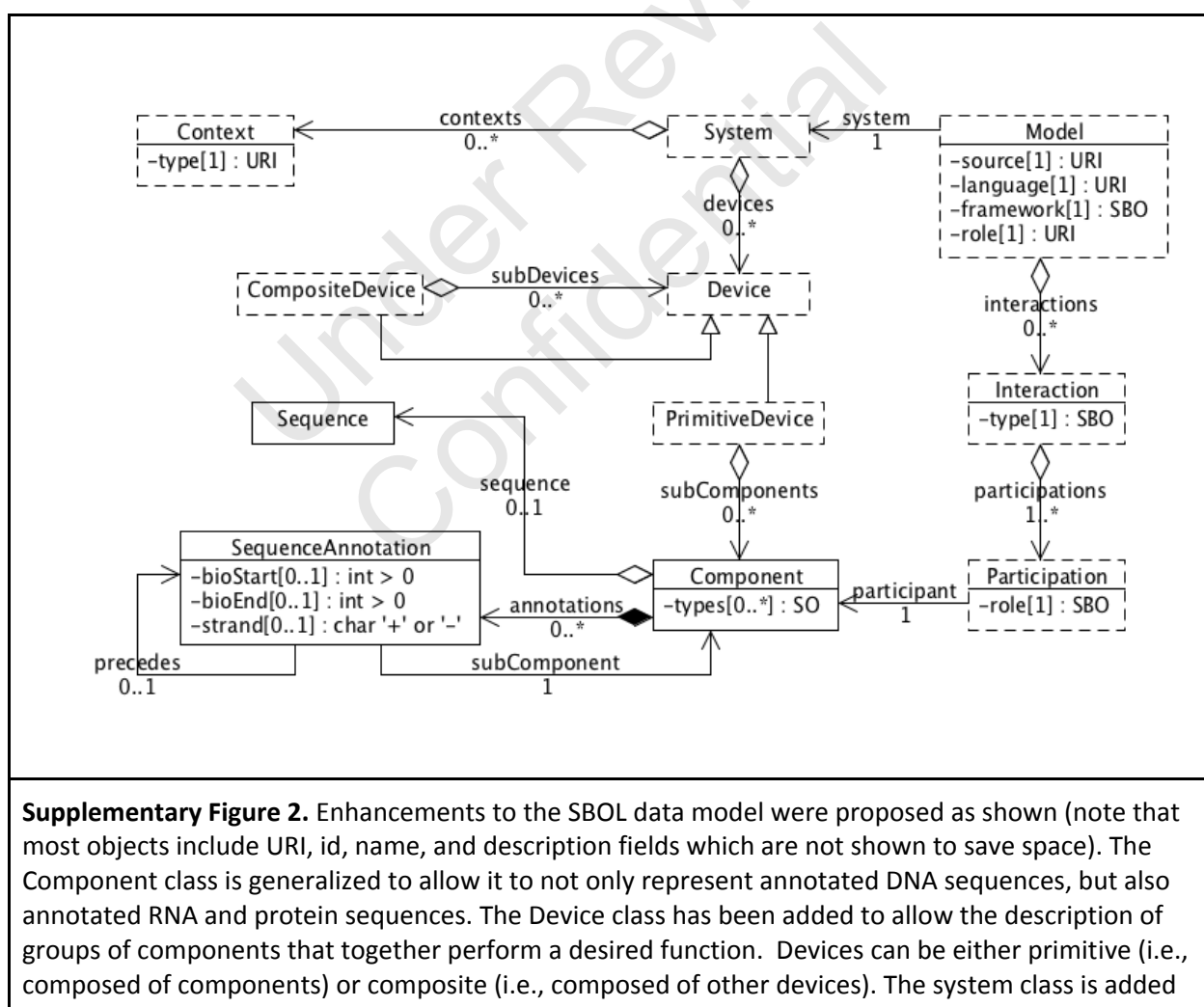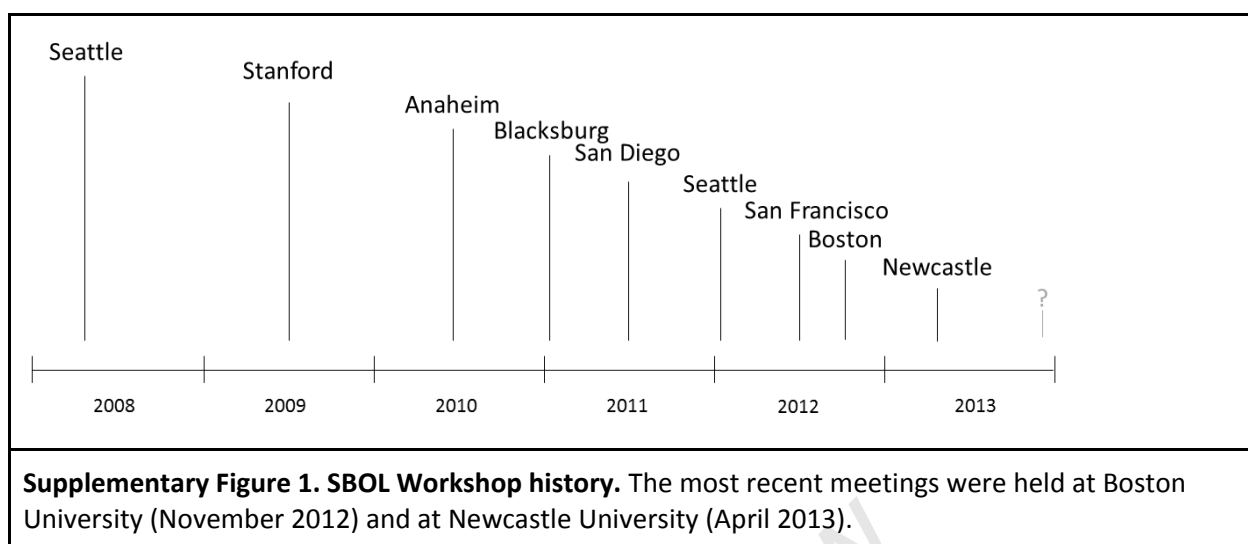|		
Collaboration (INSDC), to support aggregation in centralized databases. Consortium is made up of members of each of the databases' advisory bodies.	among synthetic biologists, the software tools they use, and repositories. Membership is open to all affected interests.	conform to the modification would not read the file properly. It is less likely that these formats would be altered to satisfy evolving requirements for synthetic biology. For example, information about computational models, devices, and systems are outside of the scope of GenBank format files.
Primarily defined to serve as a human readable format to display GenBank entries. A secondary consideration is its machine readability and consistent interpretation.	Defined as a data exchange standard, with a specified format that is machine readable by SBOL compliant software and any software capable of reading RDF. To avoid ambiguity SBOL leverages terms defined by the Sequence Ontology (SO).	The GenBank format has been used extensively outside of its original goal to serve the DDBJ/EMBL/GenBank repositories. Interpretation of files generated by other sources can be ambiguous when read by software tools. For example, software tools such as, A plasmid Editor (ApE)[9], use the 'misc_feature' Feature Key by default and result in this ambiguous feature's extensive use in many files generated by the software.
Assumes the DNA molecule already exists, as a single, complete molecule. Any unknown segments are the result of incomplete sequencing; intentional or not.	Accounts for partial designs, intermediate stages of design, by allowing the DNA sequence of a DNA component to be optional.	One could develop a convention for GenBank files interpreting sequences with N, as incomplete. However such a convention would create ambiguity as to whether the sequence is intentionally omitted, unspecified, or the result of incomplete sequencing.
Features are defined in terms of their position on the sequence specified by the sequence data field; these do not specify a hierarchical relationship among each other.	The explicit hierarchical composition allows for a combination of DNA segments, subcomponents, to make up a complete design.	Results of DNA sequencing, such as in GenBank format, do not have a natural hierarchy. In SBOL, a new DNA sequence can be composed from multiple subcomponents. Since a DNA component with subcomponents can be used in a new design, multiple levels of hierarchical composition are possible.
No explicit mechanism for groups of GenBank entries.	Collection data structure, which allows DNA components to be grouped by the synthetic biologist into meaningful libraries or catalogues of components.	Multiple GenBank entries can be concatenated in a single file. However, no explicit semantics are attached to such groupings.

Supplementary Table 3. SBOL Demonstration Resources

Participant	Resource	Description
All	Galdzickietal_SBOL_Demonstration.zip	Zip archive which contains the files described in the rows below.
Washington	washington.zip----------------------------------partial_pIKE_left_cassette.xml-----------partial_pIKE_right_cassette.xml---------partial_pTAK_left_cassette.xml----------partial_pTAK_right_cassette.xml-	Zip archive has four SBOL files, each containing a partial design for a left or right cassette of a pTAK or pIKE class toggle switch.
Boston	boston.zip--pIKE_pTAK_left_right_cassettes.xml	Zip archive has a single SBOL file containing complete designs for four variants of each cassette type (left pTAK, right pTAK, left pIKE, and right pIKE).
Utah	utah.zip--pIKE_pTAK_toggle_switches.xml--------toggle_switch_models-----------------	Zip archive has a SBOL file containing complete designs for sixteen variants of each toggle switch class (pTAK and pIKE), and a directory with 32 SBML files, each containing a SBML model describing the behavior for a different toggle switch variant (specific variant is indicated via annotation of SBML model with its URI).
Newcastle	http://www.virtualparts.org/parts/property/publication/sbol-nbt	URL pointing to storage place in the Virtual Parts Repository for all fully specified SBOL DNA components and SBML models generated during the demonstration.
Joint BioEnergy Institute	https://public-registry.jbei.org/#page=collections;id=103	URL pointing to storage place in JBEI-ICE for all toggle switch DNA components generated during the demonstration.

Supplementary Table 4. Author Contributions

Name	Affiliation	conceived the idea	wrote the manuscript	edited the manuscript	designed data model	fostered community	developed libSBOL	supervised libSBOL development	developed serialization	demonstrated use	supervised deployment
Michal Galdzicki	Biomedical and Health Informatics, University of Washington, 850 Republican Street, Building C, Seattle, WA, 98109-4714, USA	•	•	•	•	•	•	•	•		•
Mandy L. Wilson	Virginia Bioinformatics Institute, Virginia Tech, Bioinformatics Facility Phase 1 (0477), 1015 Life Science Cir, Blacksburg, VA, 24061-0477, USA		•	•	•	•		•			•
Cesar A. Rodriguez	Autodesk Research, Autodesk Inc., Pier 9, The Embarcadero, San Francisco, CA, 94111, USA	•			•	•		•			
Ernst Oberortner	Department of Electrical and Computer Engineering, Boston University, 8 Saint Mary's Street, Boston, MA, 02215, USA				•	•	•	•		•	•
Matthew Pocock	School of Computing Science, Newcastle University, Claremont Tower, Claremont Road, Newcastle upon Tyne, NE1 7RU, UK		•	•		•	•				•
Laura Adam	Virginia Bioinformatics Institute, Virginia Tech, Bioinformatics Facility Phase 1 (0477), 1015 Life Science Cir, Blacksburg, VA, 24061-0477, USA					•					
J. Christopher Anderson	Department of Bioengineering, University of California, Berkeley, 512E EBB, Berkeley, CA, 94720, USA	•		•		•					•
Bryan A. Bartley	Department of Bioengineering, University of Washington, Box 355061, Seattle, WA, 98195-5061, USA					•	•			•	
Jacob Beal	Raytheon BBN Technologies, 10 Moulton Street, Cambridge, MA, 2138, USA				•	•				•	•
Deepak Chandran	OCTO, Autodesk, 1290 Beethoven Cmn - 207, Fremont, 94538, USA	•			•	•					•
Joanna Chen	Fuels Synthesis Division, Joint Bioenergy Institute (JBEI), 1 Cyclotron Road MS978R4121, Berkeley, CA, 94720, USA									•	
Douglas Densmore	Electrical and Computer Eng., Boston University, 8 Saint Mary's St., Boston MA, 02215, USA	•		•		•					•
Drew Endy	Department of Bioengineering, Stanford University, Y2E2-269B, 473 Via Ortega, Stanford, CA, 94305-4201, USA	•		•		•					
Raik Grünberg	IRIC, University of Montreal, PO Box 6128 STN Centre-Ville, Montreal, QC, H3C 3J7, Canada	•		•	•				•		
Jennifer Hallinan	School of Computing Science, Newcastle University, Claremont Tower, Claremont Road, Newcastle upon Tyne, NE1 7RU, UK	•		•		•					•
Nathan J. Hillson	Fuels Synthesis Division, Joint Bioenergy Institute (JBEI), 1 Cyclotron Road MS978R4121, Berkeley, CA, 94720, USA				•	•					•
Haiyao Huang	Department of Electrical and Computer Engineering, Boston University, 8 Saint Mary's Street, Boston, MA, 02215, USA				•						
Jeffrey D. Johnson	Plant & Microbial Biology, University of California, Berkeley, 111 Koshland Hall, Berkeley, CA, 94720, USA						•			•	
Allan Kuchinsky	Molecular Tools Laboratory, Agilent Technologies, 5301 Stevens Creek Blvd, Santa Clara, CA, 95051, USA				•	•	•				
Matthew Lux	Virginia Bioinformatics Institute, Virginia Tech, Bioinformatics Facility Phase 1 (0477), 1015 Life Science Cir, Blacksburg, VA, 24061-0477, USA					•					
Goksel Misirli	School of Computing Science, Newcastle University, Claremont Tower, Claremont Road, Newcastle upon Tyne, NE1 7RU, UK			•	•	•	•			•	
Chris J. Myers	Dept. of Electrical & Computer Eng., University of Utah, 50 S. Central Campus Dr., Rm. 3280, Salt Lake City, UT, 84112, USA	•		•	•	•				•	•
Jean Peccoud	Virginia Bioinformatics Institute, Virginia Tech, Washington St, MC0477, Blacksburg, VA, 24061, USA	•		•	•	•					•
Hector A. Plahar	Fuels Synthesis Division, Joint Bioenergy Institute (JBEI), 1 Cyclotron Road MS978R4121, Berkeley, CA, 94720, USA						•			•	
Nicholas Roehner	Bioengineering, University of Utah, 50 S. Central Campus Dr., Rm. 3280, Salt Lake City, UT, 84112, USA			•	•		•			•	
Evren Sirin	Clark & Parsia, 180 Massachusetts Ave, Arlington, MA, 02474, USA						•	•	•	•	•
Guy-Bart Stan	Bioengineering, Imperial College London, South Kensington Campus, Exhibition Road, London, SW7 2AZ, UK	•		•	•						•
Alan Villalobos	DNA2.0, 1140 Obrien Dr. Ste A, Menlo Park, CA, 94107, USA									•	
Anil Wipat	School of Computing Science, Newcastle University, Claremont Tower, Claremont Road, Newcastle upon Tyne, NE1 7RU, UK			•	•	•				•	
John H. Gennari	Biomedical and Health Informatics, University of Washington, 850 Republican Street, Building C, Seattle, WA, 98109-4714, USA	•	•			•			•		•
Herbert M. Sauro	Department of Bioengineering, University of Washington, William H. Foege Building, 3720 15th Ave NE, Seattle, WA, 98195-5061, USA	•	•	•		•					•

conceived the idea: Proposed the need for a data standard at the first workshop or original online discussion; wrote the manuscript: Initial manuscript draft; edited the manuscript: Made corrections and provided critical feedback on the manuscript; designed data model: Participated in the discussions of the data model structure online or at workshops; fostered community: Organized workshops or online community; developed libSBOL: Contributed code to libSBOL; supervised libSBOL development: Management of developer and priorities of implementation; developed serialization: Contributed ideas to the structure of the serialization; demonstrated use: A demonstration of software support for SBOL in the paper; supervised deployment: Coordinated local effort with community

Supplementary Figure 1. SBOL Workshop history. The most recent meetings were held at Boston University (November 2012) and at Newcastle University (April 2013).

Supplementary Figure 2. Enhancements to the SBOL data model were proposed as shown (note that most objects include URI, id, name, and description fields which are not shown to save space). The Component class is generalized to allow it to not only represent annotated DNA sequences, but also annotated RNA and protein sequences. The Device class has been added to allow the description of groups of components that together perform a desired function. Devices can be either primitive (i.e., composed of components) or composite (i.e., composed of other devices). The system class is added

as a way to group a collection of devices with a description of their context under which they are to be used. The context class will be used to specify the experimental methods 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 system. The model class is a mechanism for attaching models to genetic designs as an external file using any modeling language of choice (SBML, CellML, matlab, BNGL, etc.).

References

1. Galdzicki, M.*, et al.* Synthetic Biology Open Language (SBOL) Version 1.1.0. in *BBF RFC #87* 1-26 (2012).
2. Kelly, J.R.*, et al.* Measuring the activity of BioBrick promoters using an in vivo reference standard. *Journal of biological engineering* **3**, 4 (2009).
3. Müller, K.M. & Arndt, K.M. Chapter 2 Standardization in Synthetic Biology. **813**(2012).
4. Savageau, M.A. Design principles for elementary gene circuits: Elements, methods, and examples. *Chaos* **11**, 142 (2001).
5. Kærn, M.*, et al.* The engineering of gene regulatory networks. *Annual review of biomedical engineering* **5**, 179-206 (2003).
6. Eilbeck, K.*, et al.* The Sequence Ontology: a tool for the unification of genome annotations. *Genome Biol* **6**, R44-R44 (2005).
7. Mungall, C.J., Batchelor, C. & Eilbeck, K. Evolution of the Sequence Ontology terms and relationships. *Journal of biomedical informatics* (2010).
8. Sage, A.P. & Rouse, W.B. *Handbook of systems engineering and management.*, (Wiley-Interscience, 2011).
9. Davis, M.W. ApE- A Plasmid Editor. (2009).