

COMMUNITY PAGE

SBOL Visual: A Graphical Language for Genetic Designs

Jacqueline Y. Quinn¹✉, Robert Sidney Cox III²✉, Aaron Adler³, Jacob Beal³, Swapnil Bhatia⁴, Yizhi Cai⁵, Joanna Chen^{6,7}, Kevin Clancy⁸, Michal Galdzicki⁹, Nathan J. Hillson^{6,7}, Nicolas Le Novère¹⁰, Akshay J. Maheshwari¹¹, James Alastair McLaughlin¹², Chris J. Myers¹³, Umesh P¹⁴, Matthew Pocock^{12,15}, Cesar Rodriguez¹⁶, Larisa Soldatova¹⁷, Guy-Bart V. Stan¹⁸, Neil Swainston¹⁹, Anil Wipat¹², Herbert M. Sauro^{20*}



CrossMark
click for updates

OPEN ACCESS

Citation: Quinn JY, Cox RS III, Adler A, Beal J, Bhatia S, Cai Y, et al. (2015) SBOL Visual: A Graphical Language for Genetic Designs. *PLoS Biol* 13(12): e1002310. doi:10.1371/journal.pbio.1002310

Published: December 3, 2015

Copyright: © 2015 Quinn et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Funding: The authors acknowledge funding from Autodesk, Inc., Stanford University, Brunel University, and the National Science Foundation Synthetic Biology Engineering Research Center and Grant EF-0850100. 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 US Department of Energy (Contract No. DE-AC02-05CH11231, to NJH). The portion of the work conducted by the University of Washington was supported by the National Library of Medicine (R41 LM010745), the National Human Genome Research Institute (R42 HG006737) and NSF awards: Theoretical Biology 0827592, Molecular and Cellular Biosciences 1158573. The portions of this work conducted by Newcastle University and Imperial College London were supported by the UK Engineering and Physical Sciences Research Council (Flowers Consortium, Grant No. EP/J02175X/1, to GBVS). The work conducted by the

1 Autodesk Research, Autodesk Inc., San Francisco, California, United States of America, **2** Chemical Science and Engineering, Kobe University, Kobe, Japan, **3** Information and Knowledge Technologies, Raytheon BBN Technologies, Cambridge, Massachusetts, United States of America, **4** Electrical and Computer Engineering, Boston University, Boston, Massachusetts, United States of America, **5** School of Biological Sciences, University of Edinburgh, Edinburgh, United Kingdom, **6** Fuels Synthesis and Technologies Divisions, Joint BioEnergy Institute, Emeryville, California, United States of America, **7** Lawrence Berkeley National Lab, Berkeley, California, United States of America, **8** Synthetic Biology Unit, ThermoFisher Scientific, Carlsbad, California, United States of America, **9** Arzeda Corp, Seattle, Washington, United States of America, **10** Babraham Institute, Cambridge, United Kingdom, **11** Stanford University School of Medicine, Stanford, California, United States of America, **12** School of Computing Science, Newcastle University, Newcastle upon Tyne, United Kingdom, **13** Department of Electrical and Computer Engineering, University of Utah, Salt Lake City, Utah, United States of America, **14** Department of Computational Biology & Bioinformatics, University of Kerala, Kerala, India, **15** Turing Ate My Hamster LTD, Newcastle upon Tyne, United Kingdom, **16** Department of Biomedical Sciences, College of Medicine, Florida State University, Tallahassee, Florida, United States of America, **17** Computer Science, Brunel University, London, United Kingdom, **18** Department of Bioengineering, Centre for Synthetic Biology and Innovation, Imperial College London, South Kensington Campus, London, United Kingdom, **19** Centre for Synthetic Biology of Fine and Specialty Chemicals (SYNBIOCHEM), University of Manchester, Manchester, United Kingdom, **20** Bioengineering, University of Washington, Seattle, Washington, United States of America

✉ These authors contributed equally to this work.

* hsauro@uw.edu

Abstract

Synthetic Biology Open Language (SBOL) Visual is a graphical standard for genetic engineering. It consists of symbols representing DNA subsequences, including regulatory elements and DNA assembly features. These symbols can be used to draw illustrations for communication and instruction, and as image assets for computer-aided design. SBOL Visual is a community standard, freely available for personal, academic, and commercial use (Creative Commons CC0 license). We provide prototypical symbol images that have been used in scientific publications and software tools. We encourage users to use and modify them freely, and to join the SBOL Visual community: <http://www.sbolstandard.org/visual>.

University of Manchester was supported by the UK Biotechnology and Biosciences Research Council (Centre for Synthetic Biology of Fine and Speciality Chemicals (SYNBIOCHEM; BB/M017702/1). The work conducted by the University of Edinburgh was supported by a Chancellor's Fellowship and BBSRC grant (Building national hardware and software infrastructure for UK DNA Foundries; BB/M025640/1). Finally, SBOL is supported by the National Science Foundation under Grant Nos. DBI-1356041 to CJM, and DBI-1355909 to HMS. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or our other funding agencies. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: I have read the journal's policy and have the following conflicts: SB is co-founder of Lattice Automation, a company that develops software tools for synthetic biology.

Abbreviations: BBF RFC, BioBrick Foundation Request for Comments; CAD, computer aided design; CDS, coding sequence; CFP, cyan fluorescent protein; CSS, Cascading Style Sheets; GFP, green fluorescent protein; IEEE, Institute of Electrical and Electronics Engineers; IPTG, isopropyl-beta-D-thiogalactopyranoside; RFP, red fluorescent protein; SBOL, Synthetic Biology Open Language; SBGN, Systems Biology Graphical Notation; SVG, Scalable Vector Graphics; YFP, yellow fluorescent protein.

Background

By the 1970s, molecular biologists had already developed many variations in the language used to describe functional regions of DNA, or genetic sequence *features*, with different terms used to describe similar features in different organisms. A protein-coding DNA sequence might be called a *coding sequence* (CDS), an *open reading frame*, an *exon*, or simply a *gene*, depending on the organism and method of study. To address such concerns, the *Sequence Ontology* [1] maintains a standard set of terms for describing different genetic features. This effort helped unify annotation efforts during the rise of high-throughput genome sequencing in the last decade.

Across that same decade, *synthetic biology* has advanced capabilities for forward engineering of complex genetic systems with multiple sequence features. This has increased the need for consistent terminology and representations of genetic designs. A visual representation of genetic sequence elements and their arrangement can quickly communicate adjacency, contiguity, repetition, and uniqueness. These properties become relevant as genetic designs become more complex, with multiple promoters, CDSs, etc. This is especially true for genetic designs expressed heterologously and when a system is engineered first in one organism (e.g., [2]), then moved to a different host (e.g., [3,4]).

Standards are enabling technologies for communication: standard symbols have had a profound impact in other engineering disciplines, such as the Institute of Electrical and Electronics Engineers (IEEE) standards for representing electronic components and circuits [5,6], or computer-aided design (CAD) standards for representing architecture and mechanical engineering [7,8]. Standard symbols simplify figures and user interfaces, enhance familiarity, and streamline the design process. SBOL Visual aims to have a similar salutary effect for the engineering of biological systems.

SBOL Visual Specification

Synthetic Biology Open Language (SBOL) Visual is the product of an ongoing community effort to develop and standardize a graphical language for synthetic biology and biological engineering, focusing initially on symbols for commonly used sequence features [9]. In its current form, SBOL Visual is a set of symbols that correspond to sequence features encoded by a DNA molecule. The meaning of each symbol is established by association with terms in the Sequence Ontology (S1 and S2 Tables). SBOL Visual builds on the Sequence Ontology's ten years of work on standardizing precise definitions of genetic sequence features, and the success of this work ensures that SBOL Visual symbols are well aligned with established scientific vocabulary. The mapping to Sequence Ontology terms also connects SBOL Visual to the SBOL data exchange standard, enabling automatic mapping from data to a graphical representation [9].

Though SBOL Visual makes use of Sequence Ontology terms, the two projects address objectives of differing scope. The Sequence Ontology provides a controlled vocabulary for all functional genetic features, while SBOL Visual focuses on facilitating the communication of engineered genetic designs. One driving need for SBOL Visual is to abstract and simplify the full complexity of sequence features that may be represented with a single symbol, e.g., the promoter, which is composed of many functional subsequences [10]. For example, the "bent arrow" symbol used by SBOL Visual for a promoter (SO:0000167) could also be used to denote the transcription start (SO:0000315) when describing the substructure of the same promoter. Such usage is common in the scientific literature, and unambiguous since the descriptions are at different levels of detail. As such, each SBOL Visual symbol can map to one or more terms in the Sequence Ontology. For features common in engineered systems but not yet in the Sequence

Ontology, SBOL Visual has contributed back to the ongoing development of Sequence Ontology by contributing several new terms (e.g., `restriction_enzyme_assembly_scar`, SO:0001953).

In developing these symbols, the community initially selected a set of commonly used genetic parts and a set of features relevant for the assembly of DNA molecules. The symbols were chosen based on common depiction of genetic designs in molecular, systems, and synthetic biology publications. The 12 core features ([S1 Table](#)) include typical sequences needed for the proper functioning of a gene: such as signals for DNA replication, RNA transcription, and protein translation. We specify the simple rectangle symbol to be a “user-defined” catchall for special genetic features not currently part of SBOL Visual. We also include nine symbols to describe common methods of DNA assembly, such as restriction sites ([S2 Table](#)). Together these 21 symbols were proposed and ratified as SBOL Visual 1.0 Specification.

SBOL Visual was designed to serve a range of formats, including whiteboard discussions, slide presentations, scientific publications, and *computer-aided design* [11]. Some symbols were adjusted to account for communication goals, particularly to increase visual distinctiveness, to decrease feature orientation ambiguity, and to facilitate rapid design. The SBOL Visual symbols can thus be drawn quickly by hand, used as stencil art with computer illustration software, rendered in a web browser, used as image primitives for building software tools, or applied as a formal symbology for communication and instruction. Inspired by the clarity, simplicity, and usefulness of electronic symbols, stylistic features such as scaling, line-width, color, and use of text labels are left explicitly unspecified in SBOL Visual. This built-in flexibility facilitates variation in style for functional or aesthetic purposes, which allows genetic designers and software developers to differentiate their images and interfaces to express additional information (e.g., module structure, protein type, origin of components) or artistic style while still benefiting from the standard. Thus, SBOL Visual provides a set of readily distinguishable basic shapes, which can be refined for individual applications.

Current Use in Publications and Software

Several scientific publications have already adopted SBOL Visual (e.g., [12–16]). To illustrate the potential benefits of SBOL Visual in publication, consider the development of the multi-color genetic reporter system shown in [Fig 1](#). This system was initially built with three fluorescent protein genes to measure genetic regulation and noise in *Escherichia coli* [17], and it was later modified to use as a test bed for measuring combinations of promoters and ribosome binding sites [18], the effect of codon bias on the initiation of translation [19], and to analyze the effects of carbon and nitrogen metabolism on synthetic circuit performance [20]. When originally published, each of these publications used modified forms of the genetic system and slightly different symbols to depict the genetic features ([Fig 1A–1C](#) left-side images). Unifying such variant genetic visual depictions with a common symbology ([Fig 1A–1C](#) right-side images) makes their relationship much more immediately apparent, and clearly communicates their structural differences. We expect the use of SBOL Visual to disseminate new results, inspire new genetic designs, and communicate ideas across human language barriers.

Adoption of SBOL Visual by software tools (see [Box 1](#)) has further helped to drive use of SBOL Visual in publication. For example, early versions of the SBOL Visual symbols in Clotho Spectacles [22] were used to produce schematics for a genetic system able to convert ambient nitrogen to ammonia (nitrogen fixation) ([S1 Fig](#)) [13]. The SBOL Visual symbols have also been used to represent software-generated combinatorial designs of transcriptional cascades and feed-forward circuits [23], and in layered transcriptional circuit devices derived from CRISPR for use in mammalian cells [15]. In sum, SBOL Visual is becoming an accepted standard for communicating genetic designs, both in software and in the scientific literature.

Box 1. SBOL Visual Adoption in Academic and Commercial Software Tools Enables Clear Design Specification and Accurate Sequence Annotation

- Pigeon [24] is a synthetic biology design visualizer that generates SBOL Visual figures from terse strings similar to classical genotype notation with a great deal of flexibility, including the ability to color-code, invert, and add text descriptions to the symbols.
- The ICE genetic design repository software platform [25] includes an SBOL Visual view (created automatically using Pigeon) with DNA component types (CDS, terminator, etc.), encouraging more accurate sequence annotation. This view is automatically generated from the uploaded annotated sequence file.
- The GraphViz graph visualization package [26] includes SBOL visual symbols, which can be used to visualize genetic constructs and include them as components in arbitrary graph diagrams.
- SBOL Visual Web Widgets (see Table 1) allow for dynamically rendering the symbols in Scalable Vector Graphics (SVG) and their styling using Cascading Style Sheets (CSS). The current set of available widgets include CDS, promoter, terminator, and restriction site.
- The VisBOL Web-based design visualizer (see Table 1) is an open-source tool enabling the dynamic, automated generation of SBOL Visual designs from SBOL documents in the browser, while DasBOL (see Table 1) provides a web service for querying relationships between SBOL Visual and Sequence Ontology terms.
- SBOL Visual symbols are used as interactive abstractions of genetic parts in a number of genetic design applications, including BioCompiler [16], Clotho [22], DeviceEditor [27], GenoCAD [28], SBOL Designer [29] and its Geneious extension [30], TeselaGen Design Editor [31], Tinkercell [32], and VectorNTI Express Designer [33].

New Symbol Adoption and Symbol Variations Process

Although SBOL Visual can express many useful features of genetic designs, it is not complete. SBOL Visual is developed through an open and ongoing community process, and any synthetic biology practitioner is encouraged to propose new symbols and modifications to existing symbols. This is done by submitting a proposal to the SBOL Visual working group by email, containing the proposed name, symbol, associated Sequence Ontology term, and motivation for addition of the symbol. In cases in which no appropriate Sequence Ontology term exists, the SBOL Visual working group requests one; we have so far submitted ten such terms to the Sequence Ontology. The proposed symbol then goes through an endorsement process, a trial period, and finally adoption by open community vote.

Using the same process, a symbol may be proposed as a *variant* of an existing symbol. Symbol variants share the same Sequence Ontology term, but may be more useful in different contexts. For example, an asymmetric variant of the symbol for transcriptional terminator has been proposed in order to better capture the often directional behavior of these genetic parts. While a more complex shape, it allows for more explicit specification of directionality when

Table 1. Links to SBOL Visual resources and related tools.

SBOL Visual website: Project page with information about SBOL Visual and download links	http://sbolstandard.org/visual/
Sequence Ontology: Terms used to establish SBOL Visual symbol meanings	http://sequenceontology.org/
Pigeon: Web tool for fast and easy scripting of SBOL Visual diagrams (open public web server)	http://pigeoncad.org/
ICE Genetic Repository: Repository for parts, generates SBOL Visual from annotated sequence files (open source but requires login account to use public web server)	https://public-registry.jbei.org
Web Widgets: JavaScript SBOL Visual web renderer (open source with live demo)	http://drdozer.github.io/sbolv/
VisBOL: Open-source web-based SBOL Visual design visualizer in JavaScript, with support for visualizing Genbank and SBOL standard file formats	http://visbol.org/design/
DasBOL: A web service which looks up SBOL Visual terms associated with Sequence Ontology codes	http://dasbol.org/
DeviceEditor: A visual biological CAD canvas (not open source and requires registration)	https://j5.jbei.org/
TeselaGen Design Editor: A visual biological CAD canvas (commercial software, requires registration, free for academic use with registration)	https://teselagen.com/

doi:10.1371/journal.pbio.1002310.t001

significant. Variant symbols may also be proposed by software tools, to allow for style differences between user interfaces. Variants also provide a path for improvement of symbols and deprecation of obsolete versions over time, and they allow contextual details and styles to be recognized by the SBOL Visual standard.

Collaboration and Future Work

SBOL Visual can be combined with other graphical languages, such as symbols from the *Systems Biology Graphical Notation* (SBGN), a standard for depiction of biological regulatory networks and molecular interactions [34]. SBOL and SBGN are both core standards of the COMBINE community project [35]. The SBGN Activity Flow language has codified the common use of network interactions diagrams between biological components, and can display regulatory interactions between genetic parts represented by SBOL Visual (S1A Fig). The SBOL Visual and SBGN development groups are collaborating with the aim of enabling seamless and unambiguous use of both standards. With such collaborations and the development of further SBOL Visual symbols, we hope to enable better depiction at the various levels of organization required to represent engineered biological systems.

Parallel efforts are underway to extend the language with higher-level concepts like genetic devices, genetic systems, and cellular chassis. At the sequence level, many genetic features become strongly dependent on the host organism. We envision that several of the 12 core design SBOL Visual symbols could be expanded in detail to more detailed visual languages.

For example, the CDS symbol could display more detailed information about the protein it encodes. For rational protein engineering, a common design task is to fuse two different protein functional domains into a single molecule, such as the repressor-YFP fusion depicted Fig 1B. Often, designs use a flexible linker sequence in between two functional domains. One future goal is to develop a systematic visual representation of protein coding sequences that can display the design choices in making such fusions. This protein design language could also include distinctive symbols representing design elements such as cellular localization signals, protein degradation signals, phosphorylation sites, protein cleavage sites, and purification tags. Many

functional protein domains have been annotated by SwissProt and UniProtKB [36] and provide a rich source of sequences for rational protein design. For example, an inducible eukaryotic transcription factor was rationally designed by fusing domains for DNA binding, estrogen response, and promoter activation [37]. We hope to extend SBOL Visual to be able to visually depict such feats of engineering at the sequence design level.

The SBOL Visual Website and Data Distribution

The SBOL Visual website hosts prototypical symbol images in several formats, including vector files, an Omnigraffle stencil set, a set of web widgets, and a TrueType SBOL Visual font for word processors. The website also provides human-readable mappings between our terms and the Sequence Ontology terms, descriptions of how these terms are being used in CAD tools, and instructions for submitting new symbols and symbol variants. A machine-readable mapping of SBOL Visual symbols to common GenBank feature keys, Sequence Ontology terms, and Pigeon and Graphviz codes are provided as [S1 Data](#) and are also available via the SBOL Visual website. The SBOL Visual specification is published as a BioBrick Foundation Request for Comments (BBF RFC) upon approval by the SBOL Developers Group. SBOL Visual version 1.0.0, was published as BBF RFC 93 [38]. To make suggestions, ask questions, or join the SBOL Visual working group, visit the SBOL Visual website or email visual@sbolstandard.org.

Supporting Information

S1 Data. Machine-readable mapping of SBOL Visual symbols to common GenBank feature keys, Sequence Ontology terms, and Pigeon and Graphviz codes in CSV format.
(CSV)

S1 Fig. SBOL Visual diagrams can be created by multiple methods for various use cases and levels of formality. A gene regulatory network for production of T7 RNA Polymerase as an output to isopropyl-beta-D-thiogalactopyranoside (IPTG) AND aTc logic as presented by Temme et al. is shown: (A) informally sketched on a whiteboard, as might be done in design brainstorming, and (B) generated using Pigeon. These illustrate how different stages of the design process use the symbols differently. For example, (A) and (B) include regulation arcs while the figure in the original manuscript includes origin of replication and resistance markers. Also illustrated here is how symbol use influences the development of the standard. Use of the circle around an “x” for spacer by Temme et al. led to the inclusion of the symbol in Pigeon, and the symbol is currently going through the new symbol adoption process. Parts (A) and (B) demonstrate depiction of regulation alongside SBOL Visual. For example, “*repressor tetR* represses *promoter Ptet*” is represented by a line linking the tetR CDS symbol with the Ptet promoter symbol. This is formally described by SBGN Activity Flow as an *inhibitory arc*. In such combined diagrams, SBOL Visual depicts the genetic parts while SBGN depicts the network of interactions between biological components. SBGN is particularly compatible with SBOL Visual, and the vast majority of regulation maps in synthetic biology publications can be described using the SBGN Activity Flow language. In Activity Flow, nodes represent “activities” (e.g., gene activity) while arcs represent the effect of one activity on another. Therefore, SBOL Visual symbols can represent Activity Flow nodes, with arcs representing the regulation logic.
(TIFF)

S1 Table. SBOL Visual symbols for genetic design.
(PDF)

S2 Table. SBOL Visual symbols used for cloning and sequencing.
(PDF)

Acknowledgments

The authors thank all members of the SBOL effort, past and present (<http://www.sbolstandard.org/community/sbol-developers>), for their support of and contributions to SBOL Visual, and the members of the Computational Modeling in Biology Network (<http://co.mbine.org/>) for welcoming SBOL into the consortium and for their continued collaborative efforts. The authors thank Reshma Shetty for publishing the first SBOL Visual Omnigraph stencil. The authors would like to particularly thank Suzie Bartram, Anusuya Ramasubramanian, and Drew Endy for publishing the first SBOL Visual RFC and inspiring the formation of the SBOL Visual project.

References

1. Eilbeck K, Lewis SE, Mungall CJ, Yandell M, Stein L, Durbin R, et al. The Sequence Ontology: a tool for the unification of genome annotations. *Genome Biol.* BioMed Central Ltd; 2005; 6: R44.
2. Martin VJJ, Pitera DJ, Withers ST, Newman JD, Keasling JD. Engineering a mevalonate pathway in *Escherichia coli* for production of terpenoids. *Nat Biotechnol.* 2003; 21: 796–802. PMID: [12778056](#)
3. Ro D-K, Paradise EM, Ouellet M, Fisher KJ, Newman KL, Ndungu JM, et al. Production of the antimalarial drug precursor artemisinic acid in engineered yeast. *Nature.* Nature Publishing Group; 2006; 440: 940–943.
4. Paddon CJ, Westfall PJ, Pitera DJ, Benjamin K, Fisher K, McPhee D, et al. High-level semi-synthetic production of the potent antimalarial artemisinin. *Nature.* 2013; 496: 528–532. doi: [10.1038/nature12051](#) PMID: [23575629](#)
5. IEEE Graphic Symbols for Logic Functions (Includes IEEE Std 91A-1991 Supplement, and IEEE Std 91-1984). IEEE Std 91a-1991 \ IEEE Std 91-1984. 1991; 0_1-. 10.1109/IEEESTD.1991.81068
6. IEEE Standard American National Standard Canadian Standard Graphic Symbols for Electrical and Electronics Diagrams (Including Reference Designation Letters). IEEE Std 315-1975 (Reaffirmed 1993). 1993; i-244. 10.1109/IEEESTD.1993.93397
7. Schley M, Buday R, Sanders K, Smith D. AIA CAD layer guidelines. Washington, DC: The American Institute of Architects Press; 1997.
8. Institution BS. BS 1192:2007 Collaborative production of architectural, engineering and construction information. Code of practice. British Standards Institution; 2007.
9. Galdzicki M, Clancy KP, Oberortner E, Pocock M, Quinn JY, Rodriguez CA, et al. The Synthetic Biology Open Language (SBOL) provides a community standard for communicating designs in synthetic biology. *Nat Biotechnol.* 2014; 32: 545–550. doi: [10.1038/nbt.2891](#) PMID: [24911500](#)
10. Rydenfelt M, Garcia HG, Cox RS 3rd, Phillips R. The influence of promoter architectures and regulatory motifs on gene expression in *Escherichia coli*. *PLoS One.* Public Library of Science; 2014; 9: e114347.
11. Toyoda T. Methods for open innovation on a genome-design platform associating scientific, commercial, and educational communities in synthetic biology. *Methods Enzymol.* 2011; 498: 189–203. doi: [10.1016/B978-0-12-385120-8.00009-7](#) PMID: [21601679](#)
12. Ceroni F, Algar R, Stan G-B, Ellis T. Quantifying cellular capacity identifies gene expression designs with reduced burden. *Nat Methods.* 2015; 12: 415–418. doi: [10.1038/nmeth.3339](#) PMID: [25849635](#)
13. Temme K, Zhao D, Voigt CA. Refactoring the nitrogen fixation gene cluster from *Klebsiella oxytoca*. *Proc Natl Acad Sci U S A.* 2012; 109: 7085–7090. doi: [10.1073/pnas.1120788109](#) PMID: [22509035](#)
14. Guo Y, Dong J, Zhou T, Auxillos J, Li T, Zhang W, et al. YeastFab: the design and construction of standard biological parts for metabolic engineering in *Saccharomyces cerevisiae*. *Nucleic Acids Res.* 2015; 43: e88. doi: [10.1093/nar/gkv464](#) PMID: [25956650](#)
15. Kiani S, Beal J, Ebrahimkhani MR, Huh J, Hall RN, Xie Z, et al. CRISPR transcriptional repression devices and layered circuits in mammalian cells. *Nat Methods.* 2014; 11: 723–726. doi: [10.1038/nmeth.2969](#) PMID: [24797424](#)
16. Beal J, Lu T, Weiss R. Automatic compilation from high-level biologically-oriented programming language to genetic regulatory networks. *PLoS One.* Public Library of Science; 2011; 6: e22490.
17. Cox RS 3rd, Dunlop MJ, Elowitz MB. A synthetic three-color scaffold for monitoring genetic regulation and noise. *J Biol Eng.* BioMed Central Ltd; 2010; 4: 10.
18. Kosuri S, Goodman DB, Cambray G, Mutalik VK, Gao Y, Arkin AP, et al. Composability of regulatory sequences controlling transcription and translation in *Escherichia coli*. *Proc Natl Acad Sci U S A.* National Academy of Sciences; 2013; 110: 14024–14029.

19. Goodman DB, Church GM, Kosuri S. Causes and effects of N-terminal codon bias in bacterial genes. *Science*. 2013; 342: 475–479. doi: [10.1126/science.1241934](https://doi.org/10.1126/science.1241934) PMID: [24072823](https://pubmed.ncbi.nlm.nih.gov/24072823/)
20. Cardinale S, Joachimiak MP, Arkin AP. Effects of genetic variation on the E. coli host-circuit interface. *Cell Rep*. 2013; 4: 231–237. doi: [10.1016/j.celrep.2013.06.023](https://doi.org/10.1016/j.celrep.2013.06.023) PMID: [23871664](https://pubmed.ncbi.nlm.nih.gov/23871664/)
21. Dunlop MJ, Cox RS 3rd, Levine JH, Murray RM, Elowitz MB. Regulatory activity revealed by dynamic correlations in gene expression noise. *Nat Genet*. 2008; 40: 1493–1498. doi: [10.1038/ng.281](https://doi.org/10.1038/ng.281) PMID: [19029898](https://pubmed.ncbi.nlm.nih.gov/19029898/)
22. Xia B, Bhatia S, Bubenheim B, Dadgar M, Densmore D, Anderson JC. Developer's and user's guide to Clotho v2.0 A software platform for the creation of synthetic biological systems. *Methods Enzymol*. 2011; 498: 97–135. <https://books.google.com/books?hl=en&lr=&id=9uPvZWibr4C&oi=fnd&pg=PA97&dq=Developer%92s+and+user%27s+guide+to+Clotho+&ots=4Lpm4OcVIC&sig=szSWptATb8FfiEfSMgiHYO3l-Ng> doi: [10.1016/B978-0-12-385120-8.00005-X](https://doi.org/10.1016/B978-0-12-385120-8.00005-X) PMID: [21601675](https://pubmed.ncbi.nlm.nih.gov/21601675/)
23. Davidsohn N, Beal J, Kiani S, Adler A, Yaman F, Li Y, et al. Accurate Predictions of Genetic Circuit Behavior from Part Characterization and Modular Composition. *ACS Synth Biol*. 2015; 4: 673–681. doi: [10.1021/sb500263b](https://doi.org/10.1021/sb500263b) PMID: [25369267](https://pubmed.ncbi.nlm.nih.gov/25369267/)
24. Bhatia S, Densmore D. Pigeon: a design visualizer for synthetic biology. *ACS Synth Biol*. 2013; 2: 348–350. doi: [10.1021/sb400024s](https://doi.org/10.1021/sb400024s) PMID: [23654259](https://pubmed.ncbi.nlm.nih.gov/23654259/)
25. Ham TS, Dmytriv Z, Plahar H, Chen J, Hillson NJ, Keasling JD. Design, implementation and practice of JBEI-ICE: an open source biological part registry platform and tools. *Nucleic Acids Res*. Oxford University Press; 2012; 40: e141. doi: [10.1093/nar/gks531](https://doi.org/10.1093/nar/gks531) PMID: [22718978](https://pubmed.ncbi.nlm.nih.gov/22718978/)
26. Ellson J, Gansner ER, Koutsoufios E, North SC, Woodhull G. Graphviz and dynagraph—static and dynamic graph drawing tools. *Graph drawing software*. Springer; 2004. pp. 127–148. http://link.springer.com/chapter/10.1007/978-3-642-18638-7_6
27. Ajikumar PK, Xiao W-H, Tyo KEJ, Wang Y, Simeon F, Leonard E, et al. Isoprenoid Pathway Optimization for Taxol Precursor Overproduction in Escherichia coli. *Science*. BioMed Central Ltd; 2010; 330: 70–74.
28. Czar MJ, Cai Y, Peccoud J. Writing DNA with GenoCAD. *Nucleic Acids Res*. Oxford University Press; 2009; 37: W40–7. doi: [10.1093/nar/gkp361](https://doi.org/10.1093/nar/gkp361) PMID: [19429897](https://pubmed.ncbi.nlm.nih.gov/19429897/)
29. SBOL Designer by clarkparsia [Internet]. [cited 26 Feb 2015]. <http://clarkparsia.github.io/sbol/>
30. Kearse M, Moir R, Wilson A, Stones-Havas S, Cheung M, Sturrock S, et al. Geneious Basic: an integrated and extendable desktop software platform for the organization and analysis of sequence data. *Bioinformatics*. 2012; 28: 1647–1649. doi: [10.1093/bioinformatics/bts199](https://doi.org/10.1093/bioinformatics/bts199) PMID: [22543367](https://pubmed.ncbi.nlm.nih.gov/22543367/)
31. Hillson NJ. j5 DNA assembly design automation. *Methods Mol Biol*. 2014; 1116: 245–269. doi: [10.1007/978-1-62703-764-8_17](https://doi.org/10.1007/978-1-62703-764-8_17) PMID: [24395369](https://pubmed.ncbi.nlm.nih.gov/24395369/)
32. Chandran D, Bergmann FT, Sauro HM. Computer-aided design of biological circuits using TinkerCell. *Bioeng Bugs*. 2010; 1: 274–281. doi: [10.4161/bbug.1.4.12506](https://doi.org/10.4161/bbug.1.4.12506) PMID: [21327060](https://pubmed.ncbi.nlm.nih.gov/21327060/)
33. Vector NTI[®] Express Designer Software. <http://www.lifetechnologies.com/jp/ja/home/life-science/cloning/vector-nti-software/vector-nti-express-designer-software.html>
34. Le Novere N, Hucka M, Mi H, Moodie S, Schreiber F, Sorokin A, et al. The systems biology graphical notation. *Nat Biotechnol*. Nature Publishing Group; 2009; 27: 735–741. <http://www.nature.com/nbt/journal/v27/n8/abs/nbt.1558.html>
35. Hucka M, Nickerson DP, Bader GD, Bergmann FT, Cooper J, Demir E, et al. Promoting Coordinated Development of Community-Based Information Standards for Modeling in Biology: The COMBINE Initiative. *Front Bioeng Biotechnol*. 2015; 3: 19. doi: [10.3389/fbioe.2015.00019](https://doi.org/10.3389/fbioe.2015.00019) PMID: [25759811](https://pubmed.ncbi.nlm.nih.gov/25759811/)
36. Consortium UniProt. Reorganizing the protein space at the Universal Protein Resource (UniProt). *Nucleic Acids Res*. 2012; 40: D71–5. doi: [10.1093/nar/gkr981](https://doi.org/10.1093/nar/gkr981) PMID: [22102590](https://pubmed.ncbi.nlm.nih.gov/22102590/)
37. Mclsaac RS, Oakes BL, Wang X, Dummit KA, Botstein D, Noyes MB. Synthetic gene expression perturbation systems with rapid, tunable, single-gene specificity in yeast. *Nucleic Acids Res*. 2013; 41: e57. doi: [10.1093/nar/gks1313](https://doi.org/10.1093/nar/gks1313) PMID: [23275543](https://pubmed.ncbi.nlm.nih.gov/23275543/)
38. Rodriguez C, Bartram S, Ramasubramanian A, Endy D. Bbf rfc 16: Synthetic biology open language visual (sbolv) specification. 2009; <http://dspace.mit.edu/handle/1721.1/49523>