Bioengineering

http://www.aimspress.com/

Volume 2, Issue 3, 93-103.
DOI: 10.3934/bioeng.2015.3.93
Received date 30 May 2015,
Accepted date 28 June 2015,
Published date 06 July 2015

Review

Integrative metabolic engineering

George H McArthur IV [2], Pooja P Nanjannavar [2], Emily H Miller [2], and Stephen S Fong [1,2,*]

[1] Center for the Study of Biological Complexity, Virginia Commonwealth University, Richmond, VA, USA
[2] Dept. of Chemical and Life Science Engineering, Virginia Commonwealth University, Richmond, VA, USA

* **Correspondence:** Email: ssfong@vcu.edu; Tel: +804-827-7038;
Fax: +804-838-3846.

Abstract: Recent advances in experimental and computational synthetic biology are extremely useful for achieving metabolic engineering objectives. The integration of synthetic biology and metabolic engineering within an iterative design-build-test framework will improve the practice of metabolic engineering by relying more on efficient design strategies. Computational tools that aid in the design and *in silico* simulation of metabolic pathways are especially useful. However, software helpful for constructing, implementing, measuring and characterizing engineered pathways and networks should not be overlooked. In this review, we highlight computational synthetic biology tools relevant to metabolic engineering, organized in the context of the design-build-test cycle.

Keywords: metabolic engineering; synthetic biology; computational biology; in silico

1. Introduction

Metabolic engineering is a purpose-driven, application-oriented discipline rooted in molecular biology and chemical reaction engineering [1]. For perspective, metabolic engineers often think of cells as microscopic chemical factories. As such, the explicit goal of metabolic engineering is the economical production of biomolecules (primarily small molecules) through the rational, often model-guided, modification of native and/or non-native metabolic pathways—a distinct departure from the random mutagenesis and screening approach historically used in industrial microbiology. However, metabolic engineering is generally less about *how* it is practiced and much more about *what* it is trying to accomplish, meaning that the field is highly outcome-focused.

At times, there has been confusion about how research areas such as synthetic biology and systems biology relate to metabolic engineering [2–5]. It is important to understand that synthetic and systems biology are both defined by *how* research is carried out, not by *what* the research goal is. In other words, these two fields are agnostic to the scientific question being asked or the engineering problem trying to be solved. Metabolic engineering objectives, conversely, can be accomplished without using synthetic biology or systems biology approaches (e.g., using traditional recombinant DNA technology and applied biochemistry instead). However, the field stands to benefit from experimental and computational advances in synthetic and systems biology [6].

Indeed, the ability to systematically construct and modify biological systems from well-understood components will significantly enhance the scope and depth of biological inquiry and enable the elucidation of biological design principles necessary for next-generation metabolic engineering [7]. The ultimate end-point of the field will be the routine target-driven design and construction of entire microorganisms for industrial processes [8]. Getting to that point requires a framework similar to those used to manufacture complex engineered products such as electronics, aircraft and chemical factories, which we have previously discussed and are now updating [9]. One of the biggest remaining challenges is efficient *design*. How can we design and simulate the behavior of genetically encoded metabolic pathways *in silico* before building the physical DNA and testing the system *in vivo*?

Here, we offer our perspective on the relationship between synthetic biology and metabolic engineering and how they can be integrated within an iterative design-build-test framework for engineering microbial metabolism (Figure 1). The canonical design-build-test cycle is commonly used to describe the metabolic engineering workflow. This cycle can sometimes include an explicit analysis/learning step that is associated with the feedback component of iterative design improvements (feedback arrow, Figure 1). The “learn” phase of the cycle can be invaluable to understanding *in vivo* cellular function and can involve a wide array of experimental data and analytical tools (e.g. statistics or modeling) often associated with systems biology. To limit the focus and scope of this review, we will focus on highlighting advances in the practice of metabolic engineering specifically made possible by new computational tools that support experimental approaches in synthetic biology (Table 1) centered around the design-build-test steps. Other recent reviews discuss the latest progress in systems biology and its role in metabolic engineering [10–11].

2. Design

An increasing number of genetic parts (promoters, protein coding sequences, etc.) are being mined from nature, characterized, refined and standardized – some are even being designed *de novo*. Computer-aided design (CAD) tools have been developed to take advantage of these parts databases and sequence generators for the model-guided engineering of higher-order systems such as metabolic pathways. The design phase will likely become a bottleneck in the engineering process as new technologies dramatically increase the accessibility and throughput of working through the build and test phases. In addition, the integration of characterization data from standardized measurements made in the test phase will be critical to the success of this approach to engineering biological systems. Therefore, software tools that aid in the system design and redesign will become increasingly important.

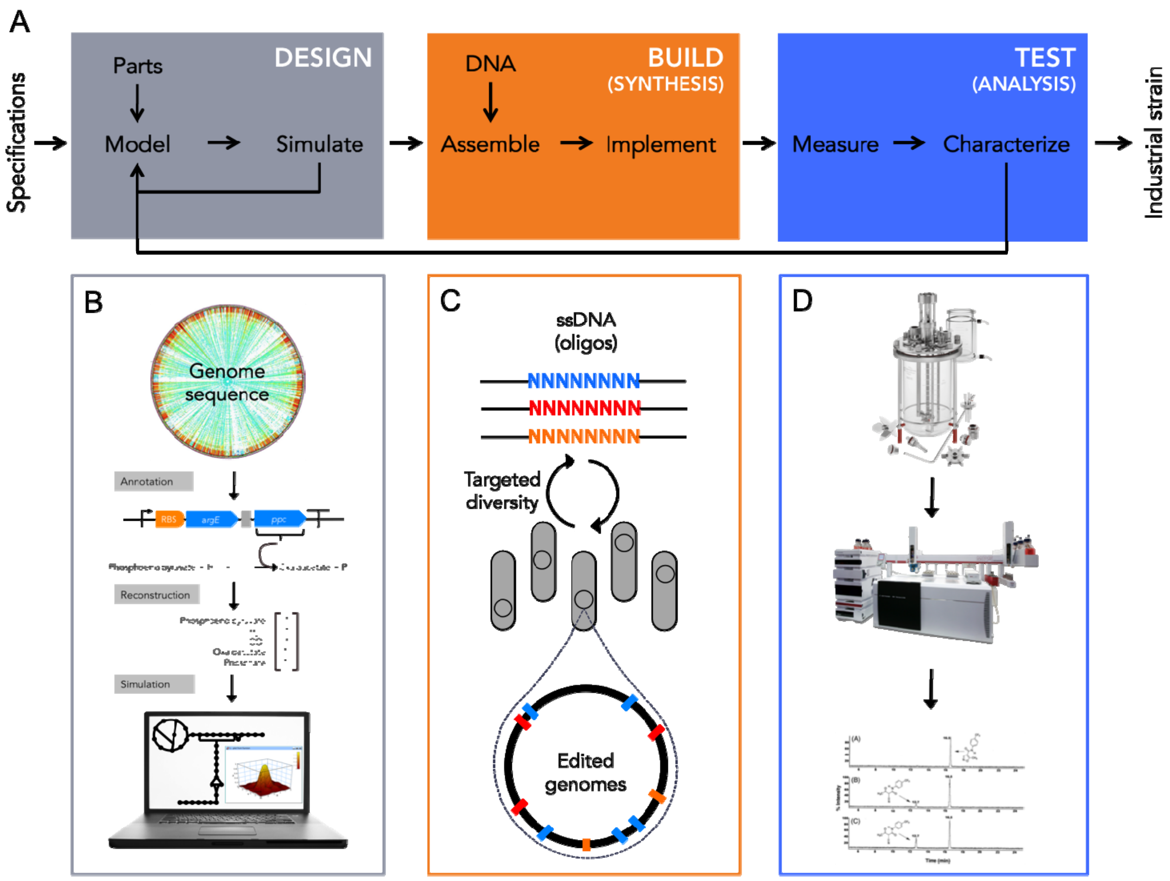

Figure 1. Iterative design-build-test cycle for industrial strain engineering. A) The design, build and test phases of the cycle can be broken down into discrete actionable steps such as *assemble* and *implement* for the *Build* phase. Often an explicit learn step is included in this cycle that here is represented by the feedback arrow from the *Test* phase back to the *Design* phase. Software tools exist to support each of these steps. For example, j5 [30] is a useful tool for designing DNA assembly reactions and CRISPR Design [37] is helpful for generating gRNAs necessary for CRISPR-Cas9-mediated genome editing. B) Computational synthetic biology tools exist to aid in the design of genetic systems such as metabolic pathways [24,25]. These tools can be integrated with existing systems biology modeling approaches. C) Multiplex automated genome engineering (MAGE) has been demonstrated to be a useful genome editing method for metabolic engineering applications. MODEST [40] has been developed to quickly and reliably design the oligonucleotides needed for genomic changes facilitated by MAGE. D) Data collected from experiments need to be integrated with the *Design* phase. This remains a challenge because the data are often specific to a particular application.

2.1. Parts

One of the key drivers of synthetic biology over the past ten years has been the attempt to establish large collections or registries of genetic parts. After all, it is difficult to practice parts-based design of whole, higher-order genetic systems without the constituent parts. The Registry of Standard Biological Parts remains the preeminent collection and catalog of physical DNA samples and their associated sequence and characterization data [12]. Similar DNA repositories have been established including the JBEI Registry [13] and the BIOFAB collection [14] (made available via AddGene.org). Recently, a collection of parts derived from commonly used plasmids (GenoLIB) was published [15]. The work conducted to generated the GenoLIB database highlights one of the major problems with parts-based genetic engineering: more often than not, databases lack consistency in the way DNA sequences are identified and described. Furthermore, it is not always clear what characterization data should be associated with genetic parts (other than enzyme coding sequences, perhaps) that would be relevant for the design of larger systems that might utilize combinations of DNA sequences to achieve diverse functions.

Table 1. Computational tools.

Software name	Description	Website	Ref.
Design			
BIOFAB	Parts collection	http://biofab.synberc.org/	14
JBEI Registry	Parts collection	https://public-registry.jbei.org/	13
iGEM Registry	Parts collection	http://parts.igem.org/	12
GenoLIB	Parts collection	http://genocad.com/	15
antiSMASH	Database mining	http://antismash.secondarymetabolites.org/	16
Syntax Inspector	Database mining	http://andersonlab.qb3.berkeley.edu/Software/EDSSI/	17
RBS Calculator	Part generator	https://www.denovodna.com/software/	18
R2o	Part generator	http://www.r2odna.com/	19
GenoCAD	System design	http://genocad.com/	25
EugeneCAD	System design	http://eugenecad.org/	24
SynBioSS	Simulator	http://synbioss.sourceforge.net/	28
iBioSim	Simulator	http://www.async.ece.utah.edu/iBioSim/	29
Build			
j5	DNA assembly	https://j5.jbei.org	30
GeneDesign	DNA assembly	http://54.235.254.95/cgi-bin/gd/gdOlapDes.cgi	33
Primer3 Plus	Sequencing	http://primer3plus.com/	41
GenoREAD	Sequencing	http://www.genoread.org/	42
CRISPR Design	Genome editing	http://crispr.mit.edu/	37
CHOPCHOP	Genome editing	https://chopchop.rc.fas.harvard.edu/	38
sgRNA Designer	Genome editing	http://www.broadinstitute.org/rnai/public/analysis-tools/sgrna-design	39
MODEST	Genome editing	http://modest.biosustain.dtu.dk/	40
Test			
GenoSIGHT	Measurement	http://genosight.sourceforge.net/	44
TASBE	Characterization	https://synbiotools.bbn.com/	45

In addition to curated databases of previously characterized parts, the genetic wealth and number of potential parts contained in raw sequence databases is extraordinary. A number of tools have been developed to help experimental synthetic biologists mine these databases for useful parts. For example, the web-based antiSMASH tool automatically identifies secondary metabolite biosynthetic pathways and gene clusters within genomic sequences, allowing a huge diversity of enzyme-mediated chemistry to be accessed [16]. However, community-populated sequence databases are not usually thoroughly curated and may contain errors. The Engineered DNA Sequence Syntax Inspector is a bioinformatic tool that solves this problem by identifying syntax errors in sequence information using three steps: 1) predicting protein coding sequences using GeneMark, 2) retrieving homologous sequences using BLAST and 3) predicting syntax errors using the SIFT algorithm [17]. DNA sequences that contain point errors, structural misannotations and unannotated coding sequences can therefore be identified and corrected before they are incorrectly incorporated into a larger construct.

Perhaps the most powerful approach to obtaining parts is to simply generate them from first principles, if possible. An incredibly useful example of this is the widely used RBS calculator, which can 1) generate novel RBS sequences for a desired translation initiation rate or 2) predict translation initiation rates from given RBS sequences [18]. Salis et al. built the RBS calculator by integrating an optimization algorithm with a biophysical model of translation initiation that quantifies the thermodynamic interactions between an mRNA transcript and the 30S ribosomal subunit. Another de novo genetic part generator is R2o, which helps researchers design biologically neutral orthogonal DNA spacer sequences [19]. These orthogonal synthetic DNA sequences are intentionally designed to not look like other functional parts and are used to reduce part dependence on genetic context. As we learn more about the interactions between various biological components (e.g., protein-DNA, protein-RNA), we will be able to generate even more—possibly all—genetic parts from scratch.

The next step advance for parts-based approaches will be to design and construct novel, non-native parts. A recent groundbreaking study takes a step in this direction through the computational design of a novel enzyme (formolase) and associated metabolic pathway [20]. The binding pocket of the designed formolase responsible for facilitating a carboligation reaction that directly fixes one-carbon units into three-carbon units was designed using RosettaDesign and Foldit calculations. Biochemical function of the formolase was demonstrated using an in vitro pathway. This study demonstrated a proof of principle approach of how computational enzyme design could become the cornerstone of novel metabolic pathway engineering. In another recently published study, researchers working with synthetic RNA were able to design a completely new functional class of small RNAs that are able to activate transcription, an activity not yet seen in nature [21]. These small transcription activating RNAs (STARs) were used to construct novel RNA-based biochemical logic gates that are faster than their protein-driven counterparts. This new kind of genetic circuitry will be particular useful for metabolic engineers who are interested in building dynamically regulated metabolism [22,23].

2.2. Modeling and simulation for in silico *testing*

Composing large metabolic pathways and eventually entire genomes from basic, well-characterized genetic parts should provide engineers the ability to mathematically describe these systems and simulate their behavior in the cell. Some of the difficulty in achieving this is: 1) there is a limited number of well-characterized parts available (a problem that is being addressed by curated collections of parts, as discussed above) and 2) it is often not clear how to go about constructing a system from parts. Sophisticated synthetic biology design tools based on parts and rules are beginning to address the latter problem.

Currently, the two primary rule-based design tools for synthetic biology are EugeneCAD [24] and GenoCAD [25]. They are particularly useful for combinatorial pathway design. There are two main differences between the two software tools with respect to the user interface. First, EugeneCAD is operated via a command line editor whereas GenoCAD provides a graphical user interface to compose rules. Second, EugeneCAD is a combinatorial design tool by default, providing straightforward commands to constrain a design space to a reasonable size (e.g., limiting the orientation of a specific genetic part, fixing the order of particular genetic parts). GenoCAD approaches design differently, decoupling the design of grammars (the collection of rules used for a particular design environment) from the design of the construct itself, a product of the predetermined grammar and the collection of genetic parts associated with that particular grammar. The advantage of the GenoCAD approach is two-fold. First, it limits unnecessary exploration of the design space by leveraging existing biological knowledge. Second, once a grammar has been established and a library of parts has been associated with that grammar, it can be saved and used again for other construct designs.

EugeneCAD appears to be an appropriate tool for the extensive exploration of genetic design space. For example, a recent metabolic engineering effort built and tested hundreds of genetic permutations of the refactored nitrogen fixation (nif) cluster from Klebsiella oxytoca (103 genetic parts, 16 genes, 5 operons), exploiting a massively parallel design-build-test cycle [26]. The design was accomplished using EugeneCAD, which allows some usually assumed constraints, such as “a terminator must occur at the end of each gene or operon,” to be relaxed. As a result, this allowed the characterization of non-standard or non-intuitive construct architectures (e.g., pseudo-operons that contain a promoter between two genes but not a terminator) that exhibited high activity. Interestingly, despite the large number of designs generated, generalized architectural rules could not be derived as several genetic architectures were able to achieve the same functional activity.

On the other hand, GenoCAD appears to be an excellent tool for capturing and formalizing domain-specific biological knowledge that can be easily reused by other non-expert researchers. For example, a GenoCAD grammar was recently developed to guide the design of synthetic transcription factors (sTFs) for use in eukaryotic cells [27], which is incredibly useful for generating synthetic or orthogonal transcription factors that can be integrated into larger systems. Future work will include extending this grammar to include promoters recognized by the DNA-binding domain of the sTF, allowing genetic engineers to quickly design synthetic gene networks derived from these interactions. The complementary nature of these two design tools (EugeneCAD for exploring the genetic design space, GenoCAD for capturing and formalizing the optimal design grammar) should empower researchers to elucidate new biological design rules.

These tools are powerful and they will become even more helpful when the simulation of the systems they help to design becomes routine. To achieve the biological equivalent of a chemical factory, in silico models of the designed systems generated by GenoCAD or EugeneCAD need to be simulated and evaluated prior to being built. A multi-level in silico platform from genetic design to prediction of function will reduce the need to massively parallelize the design-build-test cycle, which is limited by cost and the throughput of the assays used to characterize the system’s performance. To that end, GenoCAD has incorporated a simulation module powered by COPASI that allows designs associated with an attribute grammar to be simulated within the web application. Alternatively, researchers can use a variety of standalone simulation tools including SynBioSS [28] and iBioSim [29].

3. Build

We are nearing the DNA synthesis pricing of $0.01 per base pair for lengths of several Mb (entire bacterial chromosomes). At this price point, the majority of in-house cloning becomes unnecessary and the need for DNA assembly largely disappears. For example, a 2 kb plasmid would be synthesized for $20, equivalent to the current cost of long primers used for popular overlap-based DNA assembly methods. Academic labs and startup companies will be able to afford whole-genome synthesis. How does this radical shift in cost change the way we think about building and implementing synthetic metabolic pathways?

DNA fabrication will eventually become a completely outsourced task, just as specialist companies around the world manufacture the components of cars and planes or integrated circuits and other electronic components are manufactured in fabs before being used in larger device designs. However, there is currently a need for in-house DNA assembly and a number of computational tools exist to support this tedious task.

A common tool in the synthetic biology community, j5 is used for designing optimal DNA assembly strategies [30]. The manual experimental design of multipart DNA assembly is time-consuming, laborious and error-prone. Automating this process step with software mitigates these issues and provides cost-effective DNA assembly protocols for a variety of methods including Gibson's hugely popular Isothermal DNA Assembly [31,32]. Furthermore, the ability to generate machine-readable protocols for automating the actual physical assembly of DNA using liquid-handling machines makes j5 particularly attractive to organizations with existing robotic infrastructure. For the construction of individual genetic parts, DNA synthesis is likely the way to go. However, polymerase chain assembly (PCA) may be appropriate, especially if the goal is to build very large libraries of part variants that would be too expensive to outsource. GeneDesign provides an online tool that designs assembly oligonucleotides from a user-provided DNA sequence (e.g., a protein coding sequence) [33].

Beyond assembling DNA, there is a pressing need for computational tools that aid in the implementation of these synthetic DNA constructs. In particular, synthetic DNA intended for genome editing using approaches such as CRISPR-Cas9 [34] and MAGE [35,36] require design tools to be executed efficiently. In the case of CRISPR, tools have been developed to help design guide RNAs necessary for genomic targeting by Cas9. CRISPR Design [37], CHOPCHOP [38] and sgRNA Designer [39] are just a few tools available now – and many more are currently in development. Another tool, called MODEST [40], has been made to assist in the design of oligonucleotides for oligo-mediated genome engineering and recombineering approaches (e.g., MAGE). Widespread adoption of these tools will accelerate genome engineering for a variety of applications and will eventually be integrated with the CAD tools described in the Design section, paving the way for next-generation computational tools appropriate for whole-genome design and synthetic genomics. Other computational tools for sequencing, including primer design (Primer3 Plus) [41] and construct sequence verification (GenoREAD) [42], are generally useful for day-to-day lab work.

4. Test

The measurement and characterization of synthetic (or natural) biological systems is difficult to standardize – and therefore difficult to build computational tools to support. Determining the relevant parameters to be measured is complicated by dependencies on the objective or application at hand. For example, if the research goal were to optimize a nitrogen fixation pathway in an engineered bacterium, then it would make sense to use a nitrogenase activity assay to measure the performance of the candidate pathways [26]. On the other hand, general functional genomics measurements (e.g., RNA-Seq) could be easily standardized such that system behavior would be described in the context of an integrated metabolic model, which is beginning to be possible [43]. The full marriage of systems and synthetic biology will be empowering, but it will be some time before cell-wide characterization becomes the standard evaluation of newly designed systems. However, it is worth noting that this is already taking place in the context of genome editing in which entire genomes are being re-sequenced after each CRISPR-mediated modification.

Another approach that could be standardized within the design-build-test cycle is fluorescence-based measurements of synthetic biological systems such as genetic circuits or gene networks that use GFP or other fluorescent proteins as readable output. The advantage here is simply that once you have a measurement setup in place, there is little needed to change for subsequent experiments because you are measuring the same signals (wavelengths of light). This is especially true for automated, software-controlled measurement that significantly reduces user error. For example, an adaptive imaging platform called GenoSIGHT processes time-lapse fluorescence microscopy data as they are recorded and makes real-time adjustments to experimental conditions in response to the data collected [44]. This adaptive imaging approach has been shown to improve the reproducibility of gene expression data, resulting in more accurate measurements of gene network parameters. In addition, experimental time was cut to one-tenth of that required for its manual counterpart.

The data from these measurements need to be characterized and integrated with the original model in order to close the loop and support redesign. For fluorescent measurements, some progress has been made with the “Tool-chain to Accelerate Synthetic Biological Engineering” (TASBE) project and practical tools collectively called the TASBE Analysis Service are available on the web [45]. TASBE provides implementations of all of the necessary data processing, including filters, compensation techniques, etc. and supports a bead-based standard that provides fluorescence data in absolute units instead of commonly reported units of relative fluorescence. Absolute units are essential to integrating quantitative characterization data with the mathematical models used for design.

5. Conclusion

In the not too distant future, metabolic engineering will be mostly design-driven with researchers spending most of their effort on system design and data analysis, rather than on cloning and troubleshooting. This vision of the future is made possible by the incredible advances made recently in synthetic biology, both experimental and computational. Synthetic biologists are managing the biological complexity of their designs by abstracting across various levels of biological organization [46]. They are deliberately decoupling design from fabrication because how the DNA is built should not affect its design [47]. DNA synthesis has become affordable and methods for assembling and implementing synthetic DNA are efficient and accessible [48]. Quantitative models of engineered systems that can be simulated prior to building and testing in the lab are being developed and disseminated [49]. Advances in all these areas equally enable progress in target-specific projects and broader conceptual projects such as re-factoring natural systems [50,51] or modularizing pathways (e.g. multivariate modular metabolic engineering [52]).

In short, biology is rapidly becoming designable. As the need for DNA assembly and in-house cloning is gradually replaced by de novo synthesis, so will the need to perform common laboratory protocols. Just as molecular biology vendors removed the need for routine in-house purification of enzymes and preparation of reagents, many laboratory measurements and simple biological analysis will be outsourced to centralized characterization facilities (also known as robotic cloud labs). Computational tools for the design of biological systems such as metabolic pathways will become increasingly important as the Design phase inevitably becomes the bottleneck of the design-build-test cycle and the need for predictable, reliable design becomes critical. We anticipate that both new and existing computational synthetic biology tools, including those discussed in this review (Table 1), will enable the further engineering of novel or improved metabolic capabilities.

Acknowledgments

The authors would like to thank Adam B. Fisher and other members of the Fong Lab for helpful discussions and for comments on the manuscript.

Conflict of Interest

George H. McArthur has a financial interest in GenoCAD.

References

1. Stephanopoulos G (2012) Synthetic Biology and Metabolic Engineering. *ACS Synth Biol* 1: 514–525.
2. Klein-Marcuschamer D, Yadav VG, Ghaderi A, et al. (2010) De novo metabolic engineering and the promise of synthetic DNA. *Adv Biochem Eng Biotechnol* 120: 101–131.
3. Yadav VG, Stephanopoulos G (2010) Reevaluating synthesis by biology. *Curr Opin Microbiol* 1–6.
4. Nielsen J, Fussenegger M, Keasling J, et al. (2014) Engineering synergy in biotechnology. *Nature Publishing Group* 10: 319–322.
5. Church GM, Elowitz MB, Smolke CD, et al. (2014) Realizing the potential of synthetic biology. *Nat Rev Mol Cell Biol* 15: 289–294.
6. Bayer TS (2010) Transforming biosynthesis into an information science. *Nat Chem Biol* 6: 859–861.
7. Yeh BJ, Lim WA (2007) Synthetic biology: lessons from the history of synthetic organic chemistry. *Nat Chem Biol* 3: 521–525.
8. Keasling JD (2010) Manufacturing Molecules Through Metabolic Engineering. *Science* 330: 1355–1358.
9. McArthur IV GH, Fong SS (2010) Toward Engineering Synthetic Microbial Metabolism. *J Biomed Biotechnol* 2010: 1–10.
10. Lee JW, Na D, Park JM, et al. (2012) Systems metabolic engineering of microorganisms for natural and non-natural chemicals. *Nat Chem Biol* 8: 536–546.
11. Fong SS (2014) Computational approaches to metabolic engineering utilizing systems biology and synthetic biology. *CSBJ* 11: 28–34.
12. Smolke CD (2009) Building outside of the box: iGEM and the BioBricks Foundation. *Nat Biotechnol* 27: 1099–1102.
13. Ham TS, Dmytriv Z, Plahar H, et al. (2012) Design, implementation and practice of JBEI-ICE: an open source biological part registry platform and tools. *Nucleic Acids Res* 40: e141–e141.
14. Mutalik VK, Guimaraes JC, Cambray G, et al. (2013) Precise and reliable gene expression via standard transcription and translation initiation elements. *Nat Methods* 1–15.
15. Adames NR, Wilson ML, Fang G, et al. (2015) GenoLIB: a database of biological parts derived from a library of common plasmid features. *Nucleic Acids Res*.

16. Weber T, Blin K, Duddela S, et al. (2015) antiSMASH 3.0--a comprehensive resource for the genome mining of biosynthetic gene clusters. *Nucleic Acids Res*.
17. Hsiau TH-C, Anderson JC (2014) Engineered DNA Sequence Syntax Inspector. *ACS Synth Biol* 3: 91–96.
18. Salis HM, Mirsky EA, Voigt CA (2009) Automated design of synthetic ribosome binding sites to control protein expression. *Nat Biotechnol* 27: 946–950.
19. Casini A, Christodoulou G, Freemont PS, et al. (2014) R2oDNA Designer: Computational Design of Biologically Neutral Synthetic DNA Sequences. *ACS Synth Biol* 3: 525–528.
20. Siegel JB, Smith AL, Poust S, et al. (2015) Computational protein design enables a novel one-carbon assimilation pathway. *Proc Natl Acad Sci USA* 112: 3704–3709.
21. Chappell J, Takahashi MK, Lucks JB (2015) Creating small transcription activating RNAs. *Nat Chem Biol* 11: 214–220.
22. Stevens JT, Carothers JM (2015) Designing RNA-Based Genetic Control Systems for Efficient Production from Engineered Metabolic Pathways. *ACS Synth Biol* 4: 107–115.
23. Carothers JM, Goler JA, Juminaga D, et al. (2011) Model-Driven Engineering of RNA Devices to Quantitatively Program Gene Expression. *Science* 334: 1716–1719.
24. Oberortner E, Densmore D (2014) Web-Based Software Tool for Constraint-Based Design Specification of Synthetic Biological Systems. *ACS Synth Biol* 141216122616002.
25. Czar MJ, Cai Y, Peccoud J (2009) Writing DNA with GenoCAD. *Nucleic Acids Res* 37: W40–W47.
26. Smanski MJ, Bhatia S, Zhao D, et al. (2014) Functional optimization of gene clusters by combinatorial design and assembly. *Nat Biotechnol* 32: 1241–1249.
27. Purcell O, Peccoud J, Lu TK (2014) Rule-Based Design of Synthetic Transcription Factors in Eukaryotes. *ACS Synth Biol* 3: 737–744.
28. Kaznessis YN (2011) SynBioSS-Aided Design of Synthetic Biological Constructs, 1st ed., vol. 498. Elsevier Inc. 137–152.
29. Myers CJ, Barker N, Jones K, et al. (2009) iBioSim: a tool for the analysis and design of genetic circuits. *Bioinformatics* 25: 2848–2849.
30. Hillson NJ, Rosengarten RD, Keasling JD (2012) j5 DNA Assembly Design Automation Software. *ACS Synth Biol* 1: 14–21.
31. Gibson DG, Young L, Chuang R-Y, et al. (2009) Enzymatic assembly of DNA molecules up to several hundred kilobases. *Nat Methods* 6: 343–345.
32. Gibson DG, Smith HO, Hutchison CA, et al. (2010) Chemical synthesis of the mouse mitochondrial genome. *Nat Methods* 7: 901–903.
33. Richardson SM, Nunley PW, Yarrington RM, et al. (2010) GeneDesign 3.0 is an updated synthetic biology toolkit. *Nucleic Acids Res* 38: 2603–2606.
34. Jakočiūnas T, Bonde I, Herrgård M, et al. (2015) Multiplex metabolic pathway engineering using CRISPR/Cas9 in Saccharomyces cerevisiae. *Metab Eng* 28: 213–222.
35. Wang HH, Isaacs FJ, Carr PA, et al. (2009) Programming cells by multiplex genome engineering and accelerated evolution. *Nature* 1–6.
36. Wang HH, Kim H, Cong L, et al. (2012) Genome-scale promoter engineering by coselection MAGE. *Nat Methods* 9: 591–593.
37. Hsu PD, Scott DA, Weinstein JA, et al. (2013) DNA targeting specificity of RNA-guided Cas9 nucleases. *Nat Biotechnol* 31: 827–832.
38. Montague TG, Cruz JM, Gagnon JA, et al. (2014) CHOPCHOP: a CRISPR/Cas9 and TALEN web tool for genome editing. *Nucleic Acids Res* 42: W401–W407.

39. Doench JG, Hartenian E, Graham DB, et al. (2014) Rational design of highly active sgRNAs for CRISPR-Cas9–mediated gene inactivation. *Nat Biotechnol* 1–8.
40. Bonde MT, Klausen MS, Anderson MV, et al. (2014) MODEST: a web-based design tool for oligonucleotide-mediated genome engineering and recombineering. *Nucleic Acids Res* 42: W408–W415.
41. Untergasser A, Nijveen H, Rao X, et al. (2007) Primer3Plus, an enhanced web interface to Primer3. *Nucleic Acids Res* 35: W71–W74.
42. Wilson ML, Cai Y, Hanlon R, et al. (2012) Sequence verification of synthetic DNA by assembly of sequencing reads. *Nucleic Acids Res* 41: e25–e25.
43. O'Brien EJ, Lerman JA, Chang RL, et al. (2013) Genome-scale models of metabolism and gene expression extend and refine growth phenotype prediction. *Mol Syst Biol* 9: 1–13.
44. Ball DA, Lux MW, Adames NR, et al. (2014) Adaptive Imaging Cytometry to Estimate Parameters of Gene Networks Models in Systems and Synthetic Biology. *PLoS ONE* 9: e107087.
45. Beal J, Weiss R, Densmore D, et al. (2012) An End-to-End Workflow for Engineering of Biological Networks from High-Level Specifications. *ACS Synth Biol* 1: 317–331.
46. Canton B, Labno A, Endy D (2008) Refinement and standardization of synthetic biological parts and devices. *Nat Biotechnol* 26: 787–793.
47. Endy D (2005) Foundations for engineering biology. *Nature* 438: 449–453.
48. Fisher AB, Canfield ZB, Hayward LC, et al. (2013) Ex vivo DNA assembly. *Front Bioeng Biotechnol* 1: 1–7.
49. Galdzicki M, Clancy KP, Oberortner E, et al. (2014) The Synthetic Biology Open Language (SBOL) provides a community standard for communicating designs in synthetic biology. *Nat Biotechnol* 32: 545–550.
50. Chan LY, Kosuri S, Endy D, et al. (2005) Refactoring bacteriophage T7. *Mol Syst Biol* 1: 0018.
51. Temme K, Zhao D, Voigt CA (2012) Refactoring the nitrogen fixation gene cluster from Klebsiella oxytoca. *P Natl Acad Sci U S A* 109: 7085–7090.
52. Yadav VG, De May M, Lim CG, et al. (2012) The Future of Metabolic Engineering and Synthetic Biology: Towards a Systematic Practice. *Metab Eng* 14: 233–241.