The challenges of informatics in synthetic biology: from biomolecular networks to artificial organisms

Gil Alterovitz, Taro Muso and Marco F. Ramoni
Submitted: 5th August 2009; Received (in revised form): 23rd September 2009

Abstract
The field of synthetic biology holds an inspiring vision for the future; it integrates computational analysis, biological data and the systems engineering paradigm in the design of new biological machines and systems. These biological machines are built from basic biomolecular components analogous to electrical devices, and the information flow among these components requires the augmentation of biological insight with the power of a formal approach to information management. Here we review the informatics challenges in synthetic biology along three dimensions: in silico, in vitro and in vivo. First, we describe state of the art of the in silico support of synthetic biology, from the specific data exchange formats, to the most popular software platforms and algorithms. Next, we cast in vitro synthetic biology in terms of information flow, and discuss genetic fidelity in DNA manipulation, development strategies of biological parts and the regulation of biomolecular networks. Finally, we explore how the engineering chassis can manipulate biological circuitries in vivo to give rise to future artificial organisms.

Keywords: informatics; synthetic biology; systems biology; networks

INTRODUCTION
The processing and management of information is a critical part of synthetic biology, a field that approaches the design of biologically based machines from a systems engineering perspective, as a complement to systems biology. Whereas systems biology studies how biological parts give rise to the emergent properties and functions of a unified organism, the main goal of synthetic biology is to start with a set of functions and properties, and build a suitable system out of biological components. In other words, systems biology and synthetic biology represent two sides of the same coin: analysis and design [1].

The development of biologically based solutions to human problems is as old as mankind. For thousands of years, man has been breeding plants for agriculture, horses for transportation and pets for companionship. Genetic engineering pioneered the use of natural genes to modify organisms. Synthetic biologists also alter natural systems for human consumption, but with a different approach: they engineer biological systems starting from artificial components. As in systems engineering, biological modules could be developed from an eclectic set of natural sources and rapidly combined to arrive at innovations that would be far beyond incremental, time-consuming adjustments of natural organisms. The imminent departure from traditional biological engineering inspires novel ways to solve age-old problems, such as those in alternative energy [2], drug manufacture [3, 4], therapeutics [5] and green chemistry [6]. In other words, synthetic biology opens the door to unprecedented biochemical flexibility—a marked departure from an incremental pattern of progress.

Corresponding author. Taro Muso. E-mail: muso@alum.mit.edu
Gil Alterovitz is a Harvard Medical School faculty member in the Children’s Hospital Informatics Program at the Harvard/MIT Division of Health Sciences and Technology (HST).
Taro Muso is a graduate of the Harvard/MIT Division of Health Sciences and Technology (HST) and an affiliate of the Partners Healthcare Center for Personalized Genetic Medicine.
Marco F. Ramoni is the Associate Professor of Pediatrics and Medicine at Harvard Medical School, and the Director of the Biomedical Cybernetics Laboratory at the Partners Healthcare Center for Personalized Genetic Medicine.

© The Author 2009. Published by Oxford University Press. For Permissions, please email: journals.permissions@oxfordjournals.org
In theory, the synthetic biologist should be able to start with a set of desired features, design a biological circuitry that meets those requirements, and implement that design in vivo. The reality is not so straightforward (Figure 1). The current practice of producing complex biological systems usually requires an iterative optimization, partly because biological parts are subject to apoptosis, crosstalk, mutations and perturbations. In addition, a biological component can exhibit context dependence—it can stop working when it is transplanted from its native context into another cell type. Synthesized biological circuitry also suffers from biological noise and undesirable initial conditions. The issues inherent in this field become most apparent when one considers biological components, when put together, give rise to emergent properties in the whole. The existence of emergent properties indicates that our biological knowledge and design capabilities are not yet at the level of sophistication needed for a priori design and production of a prototype with a fair shot at success.

It is clear that the acknowledgement of the existence of emergent properties implies the need for a better understanding of systems biology. What is less obvious is that efficiently building a robust infrastructure for synthetic biology requires a careful management of relevant information by the research community. Such information would include biological device data exchanged by collaborators, network models exported by software and signals transduced from one biological device to another. The complexity and amount of information needed implies an opportunity for synergy through standardized communication. However, reviews on synthetic biology from an informatics perspective are rare. This review addresses this gap in the literature.

IN SILICO
Computer-based design and simulation are key elements of synthetic biology, and there is a need for efficient communication between both human beings and software programs. Taken together, these facts imply the need for standardization of synthetic biology data in silico.

Information standards
Most of the efforts in synthetic biology computer data standardization can be grouped into two areas. One starts with a network perspective, and the other has a 'bottom-up' approach that emphasizes the fundamental building block of synthetic biology, or the biological part. The dominant parts format appears to be the BioBrick Standard [7] (Figure 2), which is used by the Registry of Standard Biological Parts (http://partsregistry.org) and the international Genetically Engineered Machine (iGEM) competition [8]. The Biobricks Standard is a set of rules that define features of a DNA sequence so that each BioBrick can be easily combined into larger compositions in vitro. In other words, each BioBrick is an easily clonable DNA sequence which codes for a biological part. While the ease of DNA construction is addressed, extending the format to support

**Figure 1:** The synthetic biology infrastructure. Solid lines indicate the components of synthetic biology and the connections among them. Bold solid lines emphasize the main path from given requirements to finished product. Boxes with thin solid lines indicate support structures that need to be developed in order to make synthetic biology a practical reality. The cycles within the graph convey that the current technology requires an iterative approach to arrive at a useful biological system. In a larger context, synthetic biology (design) and systems biology (analysis) feed into each other (see dashed lines). For example, in vivo tests result in data which feed back into synthetic biology. The Database box represents organized knowledge from systems biology and quantitative data on biological parts.
the functional composition of these modules remains an important challenge [9]. The BioBrick format bases its parts characterization on promoter structure and sequences, and this is not easily translated into functional characterization within the context of interacting networks [10]. Sequence-based descriptions of parts would be appropriate in designing small systems where potential interactions could be intuitively processed (for example, by ignoring ‘nonessential’ DNA segments), but this becomes impractical for the design of large networks. This is because even ‘nonessential’ portions of biological sequences still affect functional efficiency in DNA promoters, RNA, and proteins [11]. (This paper [11] not only published new biological parts but also proposed a general strategy that addresses problems of emergent properties and design inaccuracy. This paper convincingly argued for a new way to develop and characterize components, and will likely influence the way future biological parts are presented in databases and publications.) Minor changes of nonessential sequences affect individual components in minor amounts that are only quantitatively noticeable, but small changes to one component can still have a dramatic impact on network behavior. Therefore, quantitative characterizations of component functions are necessary for efficient network design. Canton et al. [12] (this paper proposed to augment the BioBricks documentation standards) proposed to extend the Biobrick Standard by adding quantified descriptions formatted into datasheets akin to those common in electrical engineering. However, different biological parts may require different types of information [9]. In other words, the Registry may require more than one datasheet format.

Other enhancements of the BioBrick Standard have also been proposed. Recent experimental tests to confirm the validity of plasmid inserts for a collection of clones have resulted in unexpected discrepancies, so a quality control scheme has been proposed [13] (This paper proposes a quality control scheme for the Registry of Biological Parts). A provisional BioBrick language (PoBoL) was created to define a data exchange standard (http://pobol.org) [14]. More specifically, PoBoL aims to define minimal information requirements for BioBricks, provide annotation methods for BioBricks, maintain interlinking possibilities and set the stage for further language extensions.

Of equal importance to biological parts standardization is an agreement on how network designs should be described. To model biological systems, it seems logical to start with conventions developed in the systems biology community, such as the Systems Biology Markup Language (SBML) [15–18], Cellular Markup Language (CellML) [19, 20], MIRIAM [21], Systems Biology Graphical Notation (SBGN) [22, 23] [(23) formally presents a set of conventions in graphical notation

---

**Figure 2:** The BioBrick Standard [7]. (a) Basic sequence template of the BioBrick Standard. The insert of the BioBrick is flanked upstream and downstream with restriction sites. EcoRI and XbaI restriction sites are at the 5’-end (prefix). SpeI and PstI restriction sites are at the 3’-end (suffix). Each insert is a genetic component that can code for a promoter, ribosome-binding site, open reading frame, transcription termination sequence, or any combination of these. Restriction site sequences are not allowed within the genetic component. (b) Schematic of the joining process. To attach insert 1 upstream of insert 2, use restriction enzymes SpeI and XbaI respectively. The ends can then be joined together to form a scar, which cannot be cleaved again by either one of the restriction enzymes.
that will help biologists communicate clearly and efficiently) and BioPAX [24].

SBML was developed to exchange biological process information in the systems biology community [15–18]. It can be used to model a variety of phenomena, such as metabolic pathways, gene regulation and cell signaling pathways. Its success can be attributed to a number of factors. First, SBML has incorporated a number of other useful standards: MathML 2.0 [25], which provides a common mathematical expression language; the Resource Description Framework (RDF) [26], which allows for machine-readable metadata; and the Systems Biology Ontology (SBO) [27, 28] is a set of six controlled vocabularies. Second, SBML provides community-driven software support [29] (http://sbml.org/SBML_Software_Guide). A particularly useful software platform is an application programming interface (API) library called LibSBML [30], which makes SBML file manipulation accessible to scripting languages. Current translation scripts have bridged SBML-structured data and other formats [31]. Third, the SBML format is used in the BioModels Database [32] (http://www.ebi.ac.uk/biomodels-main/). Recent developments demonstrate both language extensions and applications. Its utility has been extended for stochastic simulations [33]. SBML has been used in the analysis of iron metabolism [34] and the RB/E2F pathway [35].

CellML, an alternative to SBML, is an extensible markup language that models the cell as a set of ordinary differential equations [19, 20]. Its more modular structure is convenient for multi-scale modeling and reuse of parts but has less emphasis on the biochemistry. CellML also incorporates MathML and RDF. It also has some community-driven software support [36] (http://www.cellml.org/tools). There are translators that bridge SBML and CellML [37]. Community adoption of this standard has resulted in the CellML Model Repository, which is a publicly accessible database of curated biological models [38] (this paper [38] presents the current state of the model repository). CellML’s flexibility stems from its ability to represent biological phenomena through mathematical and model building constructs, but sometimes it is useful to have explicit biological descriptions. To this end Wimalaratne et al. [39] have developed a biophysical annotation framework.

MIRIAM, or minimal information requested in the annotation of biochemical networks, is a scheme to provide extensive documentation in the model file in a structured manner [21]. Models can only be useful if there is enough annotation. Controlled annotations are achieved with the help of uniform resource identifiers (URIs) [40]. The MIRIAM approach provides a common annotation format as well as controlled vocabularies and databases [41].

SBGN is a recent attempt at standardizing the visual representation of biological networks (Figure 3) [22]. Recently, automatic equation generation for SBML from SBGN diagrams was made possible [42].

BioPAX is an effort to represent pathway data with ontological annotations [24, 43]. BioPAX complements formats like CellML and SBML because it focuses on the integration of large qualitative pathways rather than on mathematical modeling [10, 44].

The synthetic biology community also has other approaches that border on standardization. For example, Pedersen et al. [45] introduced a formal language called Genetic Engineering of Cells (GEC), which allows a modular modeling of interactions between potentially undetermined proteins and genes.

Ideally, a synthetic biology design approach would have the versatility to employ both network- and component-centric standards so that multiple levels of detail could be considered at the same time. In addition to importing publicly accessible data in common formats, the workflow would integrate problem-specific data and formats as well. Integration of the network and component perspectives is occurring or anticipated on multiple fronts. The BioBricks format is expected to support the design of ever more complex networks by incorporating integration approaches akin to BioPAX [10] that allow for ontological annotations. In contrast, standards like CellML and SBML that already allow mathematical network modeling would benefit from extending their formalisms to leverage synthetic biology constructs, such as DNA sequences and device-level information [10]. A third front is composed of integration efforts not though explicit dialogue on standards but with software development. OpenCell (PCEnv), a CellML-based platform, can model both quantitative networks and synthetic biology constructs [19].
The result of these efforts would be a comprehensive description framework, but the classic tradeoff between detail-driven accuracy and analytical efficiency will persist. Because a tradeoff naturally implies numerous possible approaches to addressing both accuracy and efficiency, each subgroup within synthetic biology may opt to pursue their own specialized formats for data management. For example, a network that depends on transcriptional regulation and a model that depends on protein–protein interaction may have different description requirements for modules and control kinetics equations. Such specializations may be easily achieved through the custom tag facility of XML [46], which is already familiar to developers of SBML and CellML.

Figure 3: SBGN network example of inter-cellular signaling near the neuromuscular junction [22, 23]. Biological concepts are organized with glyphs, or named containers. Some glyphs represent entity pool nodes, each of which is a population of entities that are not distinguished from one another in the current SBGN framework. Circular (not ellipsoid) glyphs represent ‘simple molecules’ like ATP and calcium ions. Rectangles with four rounded corners represent ‘macromolecules’ such as myosin. Glyphs can be adorned with additional information, such as the nicotinic acetylcholine receptors (nAChR), which are attached to the ‘state variable’ glyphs ‘open’ and ‘closed’. Note that the transition from ‘closed’ to ‘open’ is designated by an arrow with the ‘transition’ glyph, a small square. Another used process glyph here is ‘association’, where two lines converge to form one arrow, and a filled disk is placed downstream of the connection. By having a carefully planned set of conventions for depicting biological processes, collaborators can communicate with each other with minimal ambiguity in graphical notation.
Databases and software tools
No single data standard in synthetic biology has yet achieved the scope necessary to account for all useful information, such as epigenetic data [9]. Nevertheless, the current data formats are still useful for organizing biological information in databases and software. Synthetic Biology Software Suite (SynBioSS), designed for modeling synthetic genetic constructs, uses the Registry of Standard Biological Parts as well as a kinetic parameter database [47]. GenoCAD aims to streamline the design of synthetic DNA sequences [48]. This program appears to imply a debate in the synthetic biology community about the need for well-formatted ends for easy connection of coding sequences. The software takes advantage of the BioBrick-formatted DNA registry, but it also aims to do away with the standardization of the means by which the parts are connected. This implies a BioBrick-independent, general means of producing long stretches of error-free DNA (discussed later). CellML has software support through OpenCell (formerly PCEnv), Cellular Open Resource (COR) [49], InsilicoIDE and JSIM [19]. Cytoscape can visualize and analyze complex networks for biological research [50]. Plug-ins, which confer additional features, are actively being developed [51–54]. Funahashi’s CellDesigner [55], an editor for SBML, was designed as a tool to model network dynamics. It has a plug-in facility that enables third parties to extend the software capability. CellDesigner’s utility has been extended for stochastic simulations [33] and automatic equation generation from SBGN diagrams [42]. CellDesigner has been used in the analyses of iron metabolism [34] and the RB/E2F pathway [35]. The Process Modeling Tool (ProMoT) is a ‘drag and drop’ design platform [29]. Other software developments can be found at format-specific resource pages [29, 36, 56]. In short, concurrent with the efforts to reach consensus on information standards are attempts to employ data and standards in the design of synthetic networks.

Algorithms and heuristics
Computer-based informatics also has the advantage of relatively low-cost, quick simulations prior to in vitro implementation. Loewe [57] proposed a framework that combined systems biology and evolutionary theory to simulate mutations whose effects are too subtle to be detected in vitro. Chen et al. [58] proposed a stochastic game theory-based approach to address complications due to uncertain initial conditions and extra-cellular disturbances. They also proposed managing uncertainties by addressing four design specifications [59]. Banga [60] has recently reviewed optimization in computational systems biology. Computational limits make model simplification a useful strategy. To this end enzyme kinetic models are translated in a number of formats to reduce the model complexity. Hadlich et al. [61] developed an algorithm to automate the process of kinetic format translation. Bentley [36] proposes methods called systemic computation (SC) and fractal proteins for improving the simulations of biological systems. OptCircuit is an optimization-based method for automatically identifying the required circuits from a database of components and kinetic parameters [62]; this method may work well with Ellis et al.’s strategy of designing networks from quantitatively characterized libraries of diversified components [11]. Cantone et al. [63] developed a small synthetic gene network to assess current modeling and reverse-engineering algorithms. Models based on ordinary differential equations and Bayesian networks were qualitatively accurate, but it is not yet clear if these conclusions are generalizable to the analysis of larger networks. We see that the need for an unambiguous, quantitative, and collaborative exchange of digital, computerized information is currently being addressed by a variety of standards, databases and software.

Improvements in algorithms for analyzing networks in synthetic and systems biology are needed, because our current, relatively simple models do not have the capacity to handle the abundant data acquired from complex biological systems [31]. Issues in network analysis are exemplified by the fact that inferences from small-sized networks cannot be simply extrapolated to larger networks, as Stumpf et al. [64] have shown that sub-networks of a scale-free network are not necessarily scale free. In general, a rigorous statistical analysis of network data is difficult because there are numerous correlations [31].

IN VITRO
The informatics approach can also reframe the in vitro aspects of synthetic biology. In this light, DNA synthesis from computer-aided design is essentially a format conversion from bytes
to basepairs. Biological parts development often involves a refinement of signal transduction, or data flow within a biological circuit. Protein complexes can be modeled as instances of noisy communication channels [65, 66]. Indeed, because information-processing devices such as logic gates have been already implemented in vitro (Figure 4). In other words, critical informatics technology

**Figure 4:** Transcription-based logic gates constructed from modular transcription units [67]. Electronic logic gates are the fundamental building blocks of computational ability. For each logic gate, the table presents the boolean logic (column 2), design a biological module (column 3) and emulate the electronic counterpart with an expression profile (column 4). Each network architecture represents a synthetically designed component.
in synthetic biology resides not only in computers but also in biological circuitry as well.

**DNA synthesis**
Following a successful simulation, the computer-based network design must be translated into an *in vitro* DNA sequence. BioBrick-formatted synthetic genes can provide a set of required, proofread sequences that one can splice together (Figure 5). Combined, the much longer sequence codes for the synthetic biological circuitry. On the other hand, doing away with the BioBrick parts connection formats can streamline the design of synthetic DNA sequences [48], as long as sequence proofreading can still be done. In other words, an approach independent of the build-by-parts strategy requires a high-fidelity method for writing the basepair sequence, because even a single basepair mutation has been shown to cause system-wide disorders such as sickle-cell anemia. Linshiz et al. [68] (this paper proposes a strategy to make large, error-free DNA target molecules) developed a method for writing long, error-free DNA from potentially faulty building blocks (Figure 6). Gibson et al. [69] (this paper demonstrates that it is possible to handle an entire *Mycoplasma* genome with high fidelity) developed a method for constructing large DNA molecules, such as a 582,970-basepair *Mycoplasma genitalium* genome.

**Biological component design**
Just as electrical circuits need devices that control data flow, biological networks need biological parts that modulate signal transduction. Informatics issues in components and the network overlap with
each other. We will start with components and transition into network informatics.

Synthetic biological devices are often made from natural devices with evolutionary optimization. Natural components may therefore have context dependence that precludes them from compatible connection points with other devices. One example is the codon mismatch that occurs when a biological
part is transferred from one organism to a host of a different kingdom [71]. In order to adapt natural parts to the needs of synthetic biology, they must be standardized. Lucks [72] proposed a set of general features to consider when developing a biological device. An ideal part would be independent, reliable, tunable, orthogonal and composable. In other words, it does not interfere with other circuitry, functions as intended (context independent), can function in a range of selectable modes, can be tuned so that it does not interfere with similar devices, and can be combined to function in a system predictably. In addition, DNA sequences must adhere to the rules of transcription control [73]. Suarez et al. [74] discuss the challenges in the computational design of proteins. Martin et al. [71] review guidelines for engineering synthetic enzymes. Recent synthetic biology devices include a cellular counter in Escherichia coli [75], a tunable synthetic mammalian oscillator [76], an aptazyme-based riboswitch [77], a tunable synthetic gene oscillator [78] and a double inversion recombination switch [79]. Incidentally, Tsai et al. [80] argue that biological oscillators sometimes contain positive feedback loops in order to achieve frequency control without amplitude change. Dawid et al. [81] designed synthetic RNA regulatory elements based on transcription attenuation control.

Arkin [79] proposed developing a group of devices from a common core structure by altering a particular key property. Calling them a ‘family of parts’, Arkin argued that related devices are likely to share characterization protocols. Common protocols for a versatile set of devices would simplify the physical composition process, and this would have important ramifications on design strategies as well as parts organization within the Registry. However, it is important to keep in mind that similar devices raise the risk of crosstalk and interference with each other [10]. Unlike electrical circuits, the same ‘logic gate’ probably cannot be used in the same space.

Ellis et al. [11] proposed the development of libraries of diversified components—parts that are functionally equivalent but have differences in the nonessential sequences—for improving design strategy. Differences in nonessential sequences affect quantitative functional efficiencies of components, and this in turn can have a large impact on overall network behavior. If required documented libraries are established prior to design, then one can accurately simulate and fine-tune a system by picking the components with appropriate functional efficiencies. In other words, Ellis et al. [11] proposes to move component ‘tweaking’ to the front-end of the synthetic biology infrastructure and upstream of software-based network design. Such ‘diversified’ parts would address issues of emergent properties, biological noise and tunability. It may also address the need for compatible inputs and outputs in serial connectivity. Ellis et al. [11] successfully employed the above strategy in the development of a feed-forward loop network and a gene timer network. Establishment of such libraries will probably occur not only for DNA but RNA and proteins as well.

Biological noise presents problems for information flow through biological parts. A digital step-like interface between components may reduce the effect that noise would have on an analog system [82].

Network informatics

Information flow can also be addressed from the perspective of networks. The oldest synthetic biological circuits were based on transcriptional regulation. Within the transcriptional network, two genes were connected by having one gene code for the transcription factor of the promoter of the other gene. Carrera et al. [83] (this paper demonstrates a method to model and modify the transcription regulation network of E. coli) proposed to rewire the transcription regulation network by exchanging the endogenous promoters. Other biological circuit experiments have involved RNA-based regulation and metabolism [84]. Recently, Bashor et al. [85] [this paper introduces and demonstrates the idea of using protein scaffolds (and hence protein–protein interactions) to control synthetic regulatory networks] constructed a biological network through protein–protein interactions. Compared to translation-dependent regulatory circuits, protein-level connections have the potential for quicker response with lower cellular resource consumption rates. Engineering of protein–protein interactions becomes a tractable problem if system design leverages well-characterized protein domains [86] that enable a combinatorial strategy to generating synthetic proteins and signaling pathways. In anticipation of multi-cellular assembles with synthetic signaling requirements, Weber et al. developed a metabolite-controlled intercellular signaling
method [87]. To achieve transient system dynamics, Yin et al. [88] argued for augmenting target structure sequences with the capability to automatically construct self-assembly and disassembly pathways. Yin et al. [88] implemented such a system with a DNA hairpin motif.

Biological noise is also a problem at the network level. Studying noise in complex networks traditionally involves computational perturbation methods, because an in vitro implementation of an arbitrary noise source is not always trivial. To bridge this gap, Lu et al. [89] have developed a means of implementing simple in silico perturbation sources as in vitro molecular noise generators.

**IN VIVO**

Whereas in vitro synthetic biology enables biochemical flexibility, *in vivo* synthetic biology endows large-scale production capacity to a biological network [90]. The first step in the transition from *in vitro* to *in vivo* is the insertion of the constructed DNA into a biological chassis where transcription and translation could take place, such as a bacterium’s genome. Itaya et al. [91] addressed physicochemical stability issues of large DNA by developing the *Bacillus subtilis* genome (BGM) vector, which accommodates large DNA as part of the *B. subtilis* genome, which might combine well with cell-free expression systems in the future [92]. Shao et al. [93] developed a method for assembling a 19 kb recombinant DNA molecule in *Saccharomyces cerevisiae*. Minaeva et al. [56] integrated two recombination methods—phages site-specific and Red/ET-mediated—into a straightforward, convenient protocol. This method, called the Dual-In/Out Strategy, was applied successfully on plasmid-less marker-less *E. coli*.

When a biological network is expressed by synthetic DNA sequences within the host, or engineering chassis, crosstalk between the host and synthetic circuitry can adversely affect performance. For example, endogenous carotenoid pathways in higher plants seem to resist synthetic alterations [94]. Emergent problems from crosstalk is not surprising, even for commonly studied organisms like *E. coli*, because significant portions of organismal gene regulatory networks are not yet known [95]. Hence, minimizing or at least controlling crosstalk is a desired goal in network information control.

One approach is to reach community consensus on a ‘standard’ organism in which developed ‘standard’ parts exhibit negligible crosstalk and other desired properties. The obvious candidates are those that already have methods for accommodating large DNA molecules: *S. cerevisiae* [93] and *E. coli* [56]. However, both species will probably require crosstalk reduction through numerous deletions of non-essential genes.

The logical endpoint of systematic nonessential gene deletion is the concept of the minimal cell [96, 97], which in theory is composed only of genetic material critical to survival. Natural minimal cells like *Pelagibacter ubique* that thrive in resource-deficient environments may also be good starting points for the development of a standard artificial organism [97]. The standard artificial organism, however, is not necessarily a minimal cell, because effective crosstalk elimination may occur before all nonessential genes are deleted. In addition, the genomes of parasitic minimal cells and artificially minimized cells may present fastidious habits and lack the reliability of a bulkier genome [82]. Synthetic biology needs a host that minimizes interference while providing robust cellular infrastructure, and minimal cells do not guarantee that.

Another way to address crosstalk is to develop orthogonal ribosomes and mRNA that interact only with each other and with neither the ribosome nor the genetic material of the host organism [98]. Evolved ribosome–mRNA pairs can then be used to construct cellular networks [99]. With this approach, a synthetic type 1 coherent feed-forward loop was developed in *E. coli* [100] (this paper demonstrates that synthetic circuits can based on orthogonal transcription–translation networks). With enough orthogonal components, it may be possible to build a parallel metabolism within the cell [101].

Ultimately however, it may be necessary to implement physicochemical partitions with the phospholipid bilayer, whose adoption in natural modules poses a convincing argument for its use in synthetic biology. The bilayer can form a liposome into which one can incorporate several biochemical modules [96], which roughly outline the series of steps needed. This is essentially a ‘ground-up’ approach to the minimal cell, and the option to use artificial, low-interference modules suggests a higher chance of success than the ‘top-down’
approach of multiple gene deletions. Recently, Kuruma et al. [102] (this paper represents the latest progress in the development of the liposome into a viable chassis) developed a liposome-based system that synthesizes phosphatidic acid, a major constituent of cell membranes. A cell-free translation system was encapsulated in a liposome, in which functional membrane enzymes were synthesized. This represents a significant step toward liposome-encapsulated phospholipid bilayer biosynthesis and points toward synthetic modules with autopoietic capabilities.

At the border of in vitro and in vivo synthetic biology is the cell-free system, a platform for implementing complex biological processes outside a cell membrane. Historically, it has been difficult to activate more than one biochemical network in a single platform, but Jewett et al. [103] (this paper represents the latest progress on integrating multiple biochemical networks in a single cell-free system) have recently developed a cell-free system capable of co-activating central catabolism, oxidative phosphorylation, and protein synthesis.

Once a synthetic network has been fully implemented in vivo, the combined host-guest network must be characterized for performance and potential crosstalk. However, experimental perturbations inevitably lead to data noise. In fact, for protein interactions networks the rate of false-positive and false-negative results may be as high as 40% [104, 105]. To address this problem Lappe and Holm [106] have devised a means of efficiently deriving interaction networks. Cantone et al. [63] found that reverse-engineering methods based on ordinary differential equations and Bayesian networks were effective at inferring the structure of a small, synthetic gene regulatory network.

**CONCLUSION**

The survey of the role of information processing in synthetic biology reveals how future developments may be influenced by current ones (Table 1). Consolidation of and additions to data exchange formats are needed to enable efficient communication between people and software. The likely improvement in quantitative precision of component functional data will reduce network design unpredictability and post hoc tweaking. Current hosts for in vivo synthetic biology include E. coli and S. cerevisiae, but future hosts may take a more minimalist approach and incorporate orthogonal metabolic systems.

Synthetic biology is the next step in the progress of engineering biological systems. The key informatics challenges (some of which overlap with those of systems biology) are standardization, development of appropriate statistical analysis methods, digital data integrity, biological noise control and limitation of crosstalk (Table 2). When these issues are properly addressed, the result will be artificial organisms unrivaled in their biochemical sophistication.

**Table 1**: Recent major developments in synthetic biology. For each development, the row indicates its immediate impact niche, and the column indicates the informatics scope. However, all items noted have the potential to deeply influence the progress of synthetic biology in the next decade

<table>
<thead>
<tr>
<th>Biological part</th>
<th>In silico</th>
<th>In vitro</th>
<th>In vivo</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBGN [23]</td>
<td>CellML Model Repository [38]</td>
<td>Increased size of high-fidelity DNA [68, 69] Synthetic networks based on protein-protein interactions [85]</td>
<td></td>
</tr>
<tr>
<td>Redesigning global transcription regulation [83]</td>
<td>Orthogonal transcription-translation networks [100]</td>
<td>Semi-synthetic minimal cells [102]</td>
<td>Integrated cell-free metabolic platform [103]</td>
</tr>
</tbody>
</table>
Table 2: Idealized recipe for synthetic biology. For each step, potentially useful tools are identified. All steps exhibit issues at this time. Emergent properties can be thought of as the result of biological noise. Note that many of the problems can be traced to just a couple of infrastructure issues. For example, having many choices for parts with very precise characterizations would address issues of emergent properties at the \textit{in vitro} level. If gene regulatory networks of the chassis were to be better documented, then crosstalk issues (and therefore \textit{in vivo} emergent properties) would probably pose less of a problem.

<table>
<thead>
<tr>
<th>Step(s)</th>
<th>Tools</th>
<th>Major issues</th>
<th>Proposals</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNA synthesis</td>
<td>BioBrick ligation scheme (Figure 2) [7]</td>
<td>The integrity of large DNA target molecules</td>
<td>High fidelity construction methods for large DNA molecules [68, 69] Streamlined sequence design method [48]</td>
</tr>
<tr>
<td>\textit{In vivo} testing</td>
<td>Fluorescent protein expression [75]</td>
<td>Mutations Apoptosis Emergent properties</td>
<td>Strategy to study subtle mutations [57] Strategy to derive interaction networks [106]</td>
</tr>
</tbody>
</table>
Key Points
- The main goal of synthetic biology is to start with a set of functions and properties, and build a suitable system out of biological components.
- Component data standards (such as BioBrick) will likely require extensions to account for quantitative performance data, so that network design can become more predictable.
- Data standards for networks and components will likely consolidate in order to increase the accuracy of design simulations and efficiency of collaborations.
- Biological parts development will likely employ the strategy of building quantitatively characterized libraries of diversified components, because these libraries will increase the accuracy of network-level simulations.
- Host interference of synthetic networks might be effectively addressed by gene deletions and the use of orthogonal protein expression systems.

FUNDING
This work was supported in part by the National Library of Medicine (NLM/NIH) under grant K99 LM009826 and the National Human Genome Research Institute (NHGRI/NIH) under grants 1R01HG003354 and 1R01HG004836.

References


