Special Issue: Industrial Biotechnology

Feature Review

Harnessing QbD, Programming Languages, and Automation for Reproducible Biology

Michael I. Sadowski,[1] Chris Grant,[1] and Tim S. Fell[1,*]

Building robust manufacturing processes from biological components is a task that is highly complex and requires sophisticated tools to describe processes, inputs, and measurements and administrate management of knowledge, data, and materials. We argue that for bioengineering to fully access biological potential, it will require application of statistically designed experiments to derive detailed empirical models of underlying systems. This requires execution of large-scale structured experimentation for which laboratory automation is necessary. This requires development of expressive, high-level languages that allow reusability of protocols, characterization of their reliability, and a change in focus from implementation details to functional properties. We review recent developments in these areas and identify what we believe is an exciting trend that promises to revolutionize biotechnology.

Biology, Context Sensitivity, and Reproducibility

The 21st century will be the century of biotechnology [1,2] – the economic contribution of biotech is set to grow substantially [3,4] and it has been suggested that biotechnology offers solutions to a variety of crises faced by humanity such as climate change [5], the supply of food [6], energy via biofuels [7], and tackling escalating healthcare costs.

However, particularly with regard to the cost of healthcare, a number of authors have argued that the pharma sector is itself in crisis [8–11]. They warn of a looming danger of significantly diminishing returns on R&D expenditure in the pharmaceutical industry [12], a phenomenon somewhat facetiously known as ‘Eroom's law’. This ‘law’ is purportedly the inverse of Moore's law (which describes an exponential increase in integrated circuit transistor density over time) and denotes an exponential decrease in ROI (return on investment) for investment in pharmaceutical R&D. In an analysis of the underlying causes, Cooke [11] suggested that, in addition to the economic factors, there are substantial problems with the fundamental framework of biological research.

Specifically, following comments made by Lord Winston [13] and echoing the rallying cry of systems biology, it is suggested that the basis of pharmaceutical research is too narrow, linear, and gene-centric, and fails to properly account for the true complexity of biological systems.

There may be other signs of impending problems: several recent studies have reported low rates of reproducibility across several areas of science including drug discovery [14], psychology [15], synthetic biology [16], medicine [17,18], and cancer research [19]. One recent study estimated that the annual economic cost of irreproducible research in the life sciences is $28 billion [20].

Trends

Biological complexity is a barrier to fulfilling the potential of biotechnology.

Large numbers of complex experiments are required to overcome this barrier.

Performing such complex experiments requires sophisticated software and hardware.

New programming languages and software tools for this are developing quickly.

Low-cost automation and sensors promise to unlock these techniques for all.

[1] Synthace Limited, London Bioscience Innovation Centre, 2 Royal College St, London NW1 0NH, UK

*Correspondence: t.fell@synthace.com (T.S. Fell).

http://dx.doi.org/10.1016/j.tibtech.2015.11.006

Analyses of the possible causes of these problems and suggested solutions include a need for improved statistical practice and better management structures [21,22], non-identifiability of research resources [23], basic flaws in experimental practice either deliberate or not [24–26], and intrinsic variability in key workhorses such as antibodies [27]. A recent article by Ioannidis highlights a need for openness, quality engineering, and proper statistical control as critical to achieving reproducible research suitable for development purposes [28].

One recent report of non-misconduct-related article retractions in PubMed found that contamination and errors in analytical procedures, such as sequencing and cloning, were the major sources of error, with contamination of diminishing and analytical errors of increasing importance over time [29]. Contamination has further been identified as a major source of errors in large sequencing studies [30]. Of particular importance is the ability to define unambiguously the initial conditions prevailing when the process was performed [31].

These issues preventing reproducibility can be viewed as problems in the infrastructure of research in biology on several levels: at the lowest level there is a philosophical and methodological failure to properly embrace the highly complex and context-sensitive nature of biological systems. This may be at least partly the consequence of a methodological attachment to an overly strict interpretation of the **OFAT (one factor at a time)** (see Glossary) philosophy commonly taught as fundamental to the scientific method, but a much greater problem is the sheer difficulty of implementation that comes with the substantially greater experimental complexity that is required.

Compounding this, the languages that are used to describe and exchange protocols and workflows are imprecise, being based almost entirely on natural language, and are only executable by highly trained expert humans. These means of exchange are poorly suited to engineering problems of such complexity and scope as are necessary to properly tackle biological engineering. The reliance on human execution is an unfortunate consequence of this and traps us in a situation of low-throughput, high-variance experimentation by workers who are massively overqualified for much of their day-to-day work. Data collection requires substantial manual efforts leading to the crucial details of context going unrecorded and contributing to unexplained variance in processes.

Finally, there is a lack of tools for designing experiments, managing data, and efficiently consolidating the huge amounts of data that are necessary to work in such a complex domain. Experimental plans are generated principally in terms of the details of execution, leading to a lack of reusability and an excessive focus on the low-level details.

Automated execution of high-dimensional experimental designs, facilitated by novel programming languages for biological data representation and protocol definitions powering sophisticated design and analysis tools, promises to help solve many of these problems. We review these developments in the following sections.

Design of Experiments for High-Dimensional Experimentation

Design of experiments (DoE) is a blanket term for a body of techniques for experimental search and optimization originally devised at Rothamsted by Fisher and developed by many others [32–34]. The principal feature of DoE is the use of structured experimentation to simultaneously investigate the effects of making many changes at once. A key application of these techniques is the method of **response surface optimization**, widely applied in optimizing the productivity and robustness of industrial processes.

The main benefit of this is the ability to detect **interactions** between experimental **factors**, allowing context sensitivity of the effects of experimental changes to be addressed. This is in

Glossary

Categorical factor: a type of factor in which no relation between the different levels exists. Examples might be choice of growth medium, promoter type, or choice of coding sequence. (cf. **continuous factor**.)

Continuous factor: a type of factor in which there is an intrinsic quantitative relationship between different levels. Examples might be temperatures, lengths of time, or concentrations of media components. (cf. categorical factor.)

Critical material attribute (CMA): in QbD, a key property of a material input that affects the quality of the output of the production process.

Critical process parameter (CPP): in QbD, a key variable affecting the quality of either the output of a production process or a critical quality attribute.

Critical quality attribute (CQA): in QbD, a key attribute that can be measured and that correlates with the quality of the output of the production process.

Factor: any controlled experimental variable tested in an experiment. Usually grouped into two types: categorical or continuous (qqv).

Interaction: the combined effect of several factors as distinct from what would be expected only from the **main effects** of those factors. Interactions are defined to have an order that quantifies the number of factors involved. For example, two-factor interactions quantify the effect of the level of one factor upon the change induced by varying the other.

Level: an individual setting of a particular factor. Most experimental designs assign the same number of levels to all factors. Two, three, and five levels are the most common choices.

Main effect: the effect of a single factor on the **response**.

OFAT/OVAT: one factor (variable) at a time. Terms used by DoE initiates to describe the traditional definition of scientific practice by contrast to the DoE approach of changing multiple factors simultaneously.

Principal component analysis (PCA): one of a large number of closely related techniques for multivariate data analysis that apply matrix decomposition methods to arrays derived from high-dimensional data with the aim of finding lower-dimensional representations that contrast to the more traditional prescription of changing only one factor at a time, which cannot identify interaction effects. In the face of the very high complexity of bioprocesses, this is a serious limitation. A particularly attractive feature of this mode of experimentation is very high efficiency in a number of experimental **runs**: sophisticated designs for exploring very high-dimensional spaces have been developed, allowing the number of factor effects investigated to approach or exceed the number of experimental runs [35–38].

Despite these advantages, uptake of methods from DoE in many areas of basic biological research has been slow, with a few examples scattered across application areas [39–42] and some useful work recently published on the utility of different designs in some real case studies [43]. Contrastingly, these techniques are routine in the more industrialized and manufacturing-oriented sectors of biotechnology [44–47]. The situation is improving, however, with some interesting recent applications in protein engineering [48], strain engineering [49–51], development of whole cell biocatalysis [52], and genetic part characterization [53].

DoE techniques are exceptionally useful in improving the speed and efficiency of R&D; however, arguably their most important application is in the body of practices known variously as **quality by design (QbD)** or **process analytical technology (PAT)**. The purpose of QbD is to ensure that the outputs of production processes are as insensitive as possible to variations in the quality of inputs or process parameters such as environmental conditions [54]. A related idea is that of design for manufacture (DfM), which aims to model economic consequences of choices made at the design stage that will be incurred at the manufacturing stage, attempting to choose designs that will be cheapest for manufacturing within required quality bounds. Use of these techniques requires detailed knowledge of interactions between these sources of process sensitivity and controllable parameters and therefore relies on the use of techniques such as the above that are able to estimate these interactions.

Developments of QbD approaches relevant to the pharmaceutical industry are reviewed in [55–58]. A framework for how this may be applied to biosciences research is presented in Figure 1. The majority of recent applications of QbD in biotechnology research are directed towards improvement of analytics [59–62].

Of critical importance to the practical realization of QbD/PAT and DfM is the ability to identify **critical quality attributes (CQAs)**. This creates an emphasis on measurement, requiring interpretation of highly complex sets of data from related experiments. Typically, these very high-dimensional datasets contain much correlated information with a relatively small number of underlying features driving the observed variation. Techniques from machine learning and multivariate statistics are required to untangle these complex interdependencies and we briefly summarize these in the following section.

Analysis of High-Dimensional Data

There exist many statistical methods for dealing with high-dimensional legacy data such as **principal component analysis (PCA)**, partial least squares (PLS) regression, independent component analysis (ICA), ridge regression, lasso regression, canonical correlation, structural equation modeling (SEM), etc. Modern machine learning techniques such as neural networks, decision trees, and support vector machines provide another means to capture complex nonlinear relationships between a set of observed inputs and observed outputs.

The relationship between these methods and the experimental design techniques and implementation tools we describe is interesting and potentially extraordinarily rich and practically useful. All of the methods used above provide alternative means to analyze large datasets with high apparent dimensionality, usually by defining a lower-dimensional subspace of the inputs,

capture the majority of the observed variation.

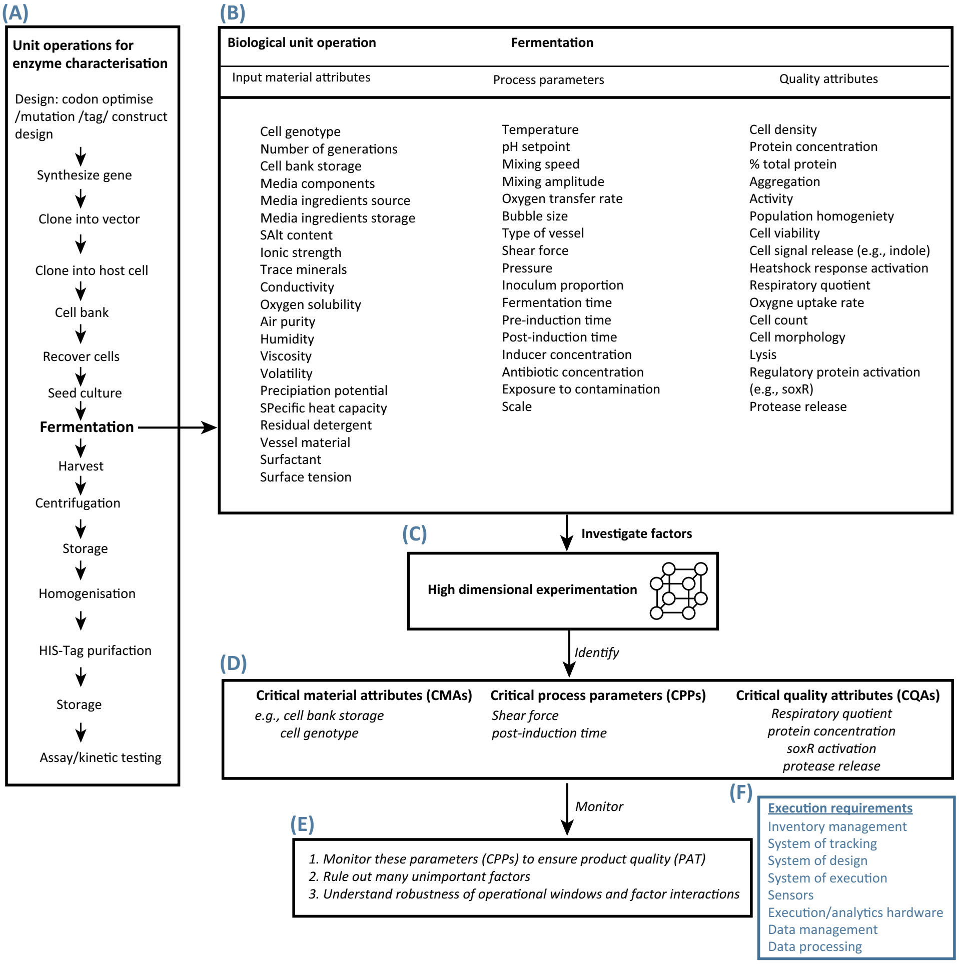
Process analytical technology (PAT): is a mechanism to design, analyze, and control pharmaceutical manufacturing processes through the measurement of critical process parameters that affect critical quality attributes.

Quality by design (QBD): a quality framework for process development, definition, and quality assurance. Typically this comprises (i) risk assessment (or failure mode and effects analysis) to identify potential causes of failure in the process; (ii) design space characterization (usually by DoE) to identify and model critical effectors of product quality; and (iii) defining methods to monitor and control the identified effectors of product quality.

Response: an experimental output of interest, determined by measurement and usually the subject of optimization or quality targets.

Response surface methodology (RSM): an optimization technique as a result of Box and others combining hill-climbing procedures with experimental designs aimed at modeling responses with second-order polynomial functions.

Run: a combination of factor settings with one or more measured responses, a unit of a designed experiment.



Trends in Biotechnology

Figure 1. Challenges and Process of How a Quality by Design (QbD) Framework Could be Applied to Research in the Biosciences. (A) The train of unit operations involved in a typical biosciences experiment, which may be undertaken in a research laboratory, in this case expression and characterization of an enzyme. This closely matches the flow of unit operations that would be involved in the manufacturing of a regulated pharmaceutical and, in principle, similar quality approaches used to ensure pharmaceutical quality could be applied to less-regulated 'grass-roots' research in the biosciences. (B) For one of these unit operations (fermentation) the variables that could affect product quality are listed and categorized into input material attributes, process parameters, and quality attributes of the process output. (C) High-dimensional experimentation based on Design of Experiments (DoE) is applied to identify which of the parameters are critical to overall product quality. (D) List of **critical material attributes (CMAs)**, **critical process parameters (CPPs)**, and critical quality attributes (CQAs) of the intermediates or final product, which have been identified through the process. (E) The actionable outputs of this process. To ensure product quality one should try to ensure monitoring (ideally continuously) of the critical parameters identified using process analytical technology (PAT). This also has the advantage that the unimportant factors (not critical to product quality) are identified and it is possible to generate prevalidated process models and windows of operating conditions whereby product quality is ensured. (F) List of requirements needed to perform this process adequately.

differing in their range of applicability, underlying mechanics, or where in the process they are most useful. The crucial feature of the relationship between DoE as practiced and multivariate methods such as the above is that by designing an experiment in a statistically optimal way the application of any of these methods will likely benefit. Briefly put, the first fundamental principle of experimental design is to ensure there is no correlation between any pair of effects (taken in the broadest sense to include higher-order combinations of input factors). This essentially obviates a large part of exploratory data analysis, which largely aims to remove correlations and collinearities between input data points – these are highly problematic for most machine learning and statistical methods. Simply getting all the data into a common framework to have a large enough set to do inference on can require dropping many potentially interesting factors and takes a lot of work. The mindset of DoE and the tools that are in development to facilitate its adoption in this domain as well as automated logging of inputs and outputs will be a huge step towards rectifying this situation by providing a large supply of clean data.

One key point about the difference in using this approach is that most machine learning and multivariate methods are designed to cope as much as possible with noisy, highly correlated legacy input data, a situation that DoE strives to avoid. Less effort has therefore been expended on how to operate in the situation where it is possible to make more data of better quality and more on the question of how to cope with the noisy, limited data available. The field of active learning and any areas relating to robotics are useful sources of insight into these problems but they are away from the mainstream of machine learning.

In particular, the focus in DoE on trying to include as many factors as may be of interest and engaging substantial domain knowledge in the task should remove the issue of sources of variation that have not been measured. The expansion of sensor technology to provide information on uncontrolled factors potentially of interest is another useful development in this regard.

The question of how to design for a machine learning approach is of great interest: there is a practical reason for the adoption of linear models in the traditional use of DoE, which is largely based on assumptions of the difficulty of experimental implementation. Given the independence of effects up to a particular order in the design, it is likely that even though these models are not the same as those that are being fitted, they nonetheless will provide substantially cleaner data than is usually available. As experiments become easier it is possible that the use of polynomial basis functions may no longer be sufficient for more sophisticated functions and alternative basis sets such as splines must be chosen. In the limit of very cheap experiments, the use of uniform designs gives optimal robustness to model misspecification [63], although these do come at a high experimental cost.

DoE, Systems Biology, and Screening

High-throughput screening approaches are very common and popularly used in biological systems. In terms of the DoE approaches we have described, these typically amount to an undirected search of a small number of factors, each with a large number of **levels**.

Although attractive for many reasons, including the simplicity of the experimental setup, there are many difficulties with this as a method for determining biological context dependence for three reasons:

(i) significant biases in the underlying population being screened, leading to
(ii) a low rate of overall information processing, mandating
(iii) the necessity for high-throughput amenable assays. In most cases, these will reduce the overall information determined per run, and in some cases imperfect proxies are used that can mislead (e.g., the use of a ‘similar’ fluorescent substrate in place of the true substrate).

There is clearly a place for screening in addressing questions that require large numbers of levels of one or more **categorical factors**, particularly when a digital output is acceptable (i.e., when trying to find an enzyme that has activity on a substrate under fixed conditions). Indeed, in some cases screening is pragmatically necessary and can be highly effective (e.g., to screen a library of 10^{10} antibody variants for binding to a cell receptor using a cell sorter). However, as a counterpoint it is preferable to apply multivariate methods in a process of factor characterization to attempt to convert these large categorical factors into a smaller number of continuous dimensions that can be more efficiently searched. Decreased reliance on large numbers of experiments leads to the potential use of much better assays and a higher aggregate amount of information processing.

Contrastingly, systems biology and DoE can be a very natural fit. Although it has been argued that prediction and explanation are often conflicting statistical goals [64]. There is a natural alliance between the two: the better an explanatory, mechanistic model captures important underlying features of the system, the better that model will predict. This essentially relates to a question of what might be called 'experimental phrasing' – the choice of factors for experimental manipulation. If the underlying components of the system and their relationships are known it is much easier to avoid including correlated or anticorrelated factors *a priori*, allowing better focused experimental questions to be asked.

A simple example of this might be found in mixture models. If we specify the components of a solution as masses and volumes and operate with a fixed total amount then certain runs end up actually being duplicated, despite being coded differently, because the final concentrations of the input components are the same. This wastes experimental runs and can confuse the analysis. Coding the system in terms of these concentrations instead leads to much greater efficiency.

By striving to understand what the underlying components are, in terms of the systems of life, and how they relate to one another, the systems biology approach has a great deal to offer the development of both predictive and explanatory models for bioprocess development.

Computational Languages for Biology

At present, the vast majority of biological knowledge is communicated in variants of natural languages such as English. This applies whether we are describing a procedure for doing an experiment, the results of an experiment, or our interpretation and semantic model of the underlying system. However, natural languages are imprecise and inherently ambiguous, leading to misinterpretation and errors in implementation. Expressing certain concepts in this way is often very difficult, leading to incompleteness. Development of appropriate languages to express biological concepts and protocols is a highly active area of research, and one that is crucial for enabling many of the approaches necessary for tackling highly complex systems.

Ontologies

Biological systems comprise an enormous number of parts and behaviors that must be catalogued if they are to be meaningfully examined and manipulated. However, it is not sufficient merely to label things, for communication and interpretability by both humans and computers the system of labels must be given a structure that defines how labels are related to one another. The tree of life is a very well-known example of this phenomenon. The term used by computer scientists for capturing such domain knowledge is 'ontology'.

Many large-scale efforts at defining biological ontologies have been made, including defining gene function [65], protein structures [66,67], biological sequences [68], models of biological processes and pathways [69,70], specific biological processes in detail [71], and specific techniques such as RT-qPCR [72]. Another important issue is the definition of equipment and its capabilities. Many pieces of equipment are not highly complex and would not require a sophisticated ontology, but there are clearly examples, such as mass spectrometers, which need to be carefully described to capture their precise capabilities. Indeed such an ontology has been created as part of a larger effort to delineate standards for proteomics analysis [73]. The Purdue ontology is another very useful example that embeds both high-level and low-level concepts as they relate to pharmaceutical development [74].

Reasoning about protocols is another domain for which computational assistance is hugely beneficial. One ontology for this, the experimental factor ontology [75], has been created, but in this case there is a better solution. Although ontologies are very useful for codifying certain types of domain knowledge, they only represent the underlying structure indirectly. If experimental factors can be defined by embedding them into protocols in a way that ties them directly to the implementation, then their meaning is not merely labeling but a consequence of how they are used in that protocol. This amounts to developing a high-level programming language for biological protocols, an ambitious goal that has already been the subject of some development. To better understand higher level languages, first we will introduce an intermediate concept, domain-specific language, which is also of great importance.

Domain-Specific Languages

Domain-specific languages (DSLs) are computing languages that aim to simplify problem solving in a restricted domain by creating a programming language dedicated to that domain. The key point is that DSLs are designed to effectively express solutions to problems of a particular type, in contrast to general purpose programming languages that can, less efficiently, express solutions to any problem. The benefit of biasing a language in this way is that programs can be expressed in a more problem-specific manner, making programming simpler and less error-prone.

A small but growing number of DSLs for biological problem domains exist, due largely to efforts in synthetic biology. An important example is Eugene, a DSL for specifying constructs to be assembled in synthetic biological workflows [76]. This allows the underlying logic of a genetic construct to be described for the purposes of specifying a desired construct and semantic checking.

Another key DSL in biology is SBML; this defines an open standard for exchanging details of metabolic simulations [70], a similar framework has also been developed as CELLML [77], which is of particular importance in relation to physiomics. Other DSLs include examples for genetic regulatory mechanisms [78], GENOCAD, another construct description language [79], and Proto [80,81] – a language originally conceived for distributed spatial computing but that is readily applicable to the cellular domain.

General Purpose Protocol Languages

DSLs are by definition restricted in scope. Each can only provide access to a small area of biological space. A few groups have recently begun to work on high-level languages that are intended to be general purpose descriptions of biology by combining both computational and experimental aspects in a single description that is executable on automated hardware.

The languages that have been developed so far are extensions of general purpose programming languages with additions to define experimental procedures. These consist primarily of liquid handling operations, operations to change the environment such as gassing, shaking, and temperature control, and measurement technologies such as spectrophotometry to measure absorbance, fluorescence, and emittance, or sophisticated analyses such as mass spectrometry. Other additions include definitions of scientific units and attempts to represent important biological concepts, often by incorporating an existing ontology or DSL.

The first language of this kind was the BioCoder language [82], which is built on top of the C/C++ language. The design and philosophy of this language – which aims for platform independence, execution of arbitrary protocols, and suitably designed domain-specific abstraction of the protocol space – are absolutely correct although the language has not to our knowledge been widely adopted. BioCoder has however been highly influential on the design of subsequent languages, in particular the efforts to standardize design and operation of microfluidics devices [83,84].

A similar but more limited idea is seen in Transcriptic's Autoprotocol. Defining an instruction set for programming laboratory automation, Autoprotocol marries the high-level Python programming language with a set of low-level commands embodied as JSON data structures detailing pipetting operations, heating or cooling and data collection operations such as measuring absorbance. Autoprotocol then provides a means for generating streams of such operations using all the tools of the Python language, with tracking of samples and results happening automatically. The purpose of this, created by cloud lab company Transcriptic, is to provide high-level access to their services. Similar company specific interfaces exist for other automation systems, including Tecan Script (and a C# based interface as well) by Tecan, CyBio XML for CyBio equipment, a C based interface for Perkin Elmer equipment, and the Venus Scripting language for Hamilton systems. In all cases, the level of abstraction is fairly low but, crucially, the protocol is not integrated with complex data processing and logic. Once written, a protocol remains a static sequence of commands with no possibility that the exact instruction set could vary based on the results of measurements.

Our own language, Antha, is perhaps the first *bona fide* attempt to create a high-level protocol language for general purpose computation in biology. Built atop Google's Go language [85], Antha is an open source high-level language that combines a complete, fully featured programming language with a number of domain-specific features such as liquid handling planning to allow specification not only of the most complex manual protocols but also to incorporate sophisticated logic and algorithms within protocols, enabling experiments of an entirely new level of complexity to be defined. The Antha language is the central component of AnthaOS, a service-oriented architecture providing device integration, experimental logging, stock management, and network interfaces for external code and services.

Finally, the Klavins lab from the University of Washington recently released Aquarium, which offers another proprietary language and execution environment, primarily for providing inventory management and step-by-step instructions for biological workflows to lab technicians.

In addition to facilitating execution of complex statistical experimental designs discussed in the first section, the ability to express logic in a high-level language that combines design, execution, and processing allows potential for integration of other useful algorithms to improve industrial translation. These may include testing mechanistic models (such as multistep enzyme kinetic models [86]), engineering parameter identification for process design [87] and process scale-up [88], economic [89] and environmental constraints [90], and the facilitation of metrology principles [91] to improve reproducibility.

These languages are all relatively new and there are many open questions about what the best language representation will be and what features are really important. Many of the advantages of these approaches do not depend on these choices, however: precision and economy of expression, integration of automation, code reuse, code generation, seamless incorporation of informatics pipelines, and the ability to use software engineering tools to do biology are all extremely powerful capabilities not accessible by traditional means. However, applying these to biology is essentially a trade-off between one type of complexity and another. New tools and metaphors are required if these sophisticated techniques are to be made accessible.

Tools for Automation, Visualization, and Experimental Design

Managing complexity via abstraction is in itself somewhat of a complex task. Biological CAD (computer aided design) systems are at an early stage of development with various components of a full-featured workflow editor present but no unified solution. Essentially, all the tools in this section are developed by systems and synthetic biologists.

At the lowest level, that of pure implementation, two systems for cross-platform liquid handling with some degree of abstraction have been developed: Pr-Pr (formerly Par-Par) from JBEI, (Joint Bioenergy Institute), which aims to provide standardized liquid handling using a high level of abstraction and high user-friendliness with outputs to Tecan liquid handling platforms, human execution, and a custom microfluidics platform [92,93]. Pr-Pr is fully open source and available on the web.

The Clotho platform is a large-scale project aimed to produce a platform with potentially user-developed plugins to provide a unified CAD system for synthetic biology applications [94]. This includes aspects of construct assembly, workflow design, and additionally contains its own implementation layer, Puppeteer, which is a web-based tool for designing liquid handling implementations in the underlying cross-platform language CHRIS (Common Human-Robot Instruction Set).

One level up from the implementation layer are tools that permit a user to design a construct for assembly and eventual transformation into a host chassis. Several commercial offerings exist for plasmid or construct design, but few offer significant functionality in the case of generating large libraries of constructs. One option that incorporates both design and implementation is the j5 system from JBEI [95,96], which is free for non-commercial use. This is integrated with other JBEI tools including DeviceEditor [97] for visual design of large-scale devices, the ICE (inventory of composable elements) part registry [98], and the Eugene DSL for specification of construct designs [76].

The Raven system [99] provides a solution for designing essentially arbitrary numbers of constructs by a rule-based system. The user defines input parts and an assembly method and identifies constraints on the system in the form of required, forbidden, recommended, or discouraged intermediates. The system also designs primers for PCR-based validation of the assemblies. Raven is available free for academic use. Visualization of designed constructs is facilitated using the pigeon platform, which provides a simple textual language for automated production of SBOL (synthetic biology open language) visual diagrams depicting constructs [100]. This is also freely available for academics.

Design and simulation of potential biological parts and devices is an area of overlap between systems and synthetic biology and has therefore seen significant software developed for a variety of modeling situations. A selection will be covered here; for further information and background, see [101,102].

Visual tools that blend design and simulation of biological devices include TinkerCell [103,104]. These offer ODE-based (ordinary differential equation) simulations and integration with parts catalogs hosted locally or remotely.

At the highest level of design, with almost no user input, are tools such as AutoBioCAD, which promise to design genetic circuits for *Escherichia coli* with no user interaction except the specification of design goals [105].

A crucial step for which many sophisticated tools exist is that of integrating the information into knowledge, which is the domain of knowledge management systems. The increasingly computational nature of research and development we anticipate is likely to play well with these systems. For a good recent review, see Herwig *et al*. [106].

The full power of these techniques is dependent on having access to automation. Traditionally expensive and single purpose, automation has remained out of reach of many researchers outside of large organizations. Recent trends in low-cost, open source hardware are set to change this trend.

Trends in Automation

Advances in automation will be key to effective implementation of these approaches. At the high end, recent product offerings such as TAP's Ambr250 parallel bioreactor system that comprises up to 24 parallel disposable bioreactors serviced by a liquid handling robot are starting to enable more ambitious bioprocess development practices towards those approaches discussed in this article. Laboratory automation has been established for decades now in pharma [107] and bioprocessing [108], but further take-up has been slow, particularly in academia. The reason may be partly cultural but we speculate that this is also partly due to the fact that robots tend to be too expensive, too hard to program, and typically set-up to do one job repeatedly rather than be flexible general use tools.

There have been recent developments that are likely to transform this trend. Companies such as Transcriptic, Emerald Cloud Lab, and Arcturus Biocloud have emerged, which allow users to execute experiments on remote, lab infrastructure. An increasing number of vendors are releasing professional grade lower cost liquid handlers (less than $20 000) such as the Gilson Pipetmax and Cybio Gene-theatre. This could drop further still with the launch of open source liquid handlers from OpenTrons and Modular Science, pricing units at less than $5000. Other open hardware has also been produced to enable the growing DIYbio [109] community such as thermocyclers (openPCR and openQPCR), electrophoresis tanks (Open gelbox), or combined hardware kits (Bentobox, Amino). These are partially enabled by the use of cheap off-the-shelf parts (such as computer fans for the thermocycler) or include 3D printing design files and assembly instructions.

Development of cheap analytics equipment could be driven by the need for cheap diagnostics for the developing world. Many affordable point-of-care diagnostics take advantage of the ubiquity and technology available in mobile phones [110]. Smartphone spectrophotometers, fluorescence detectors [111–113], and microscopes [114] have already been produced. Cheap sensors and wireless sensor networks will also play an important role. Widely available and programmable via Rasberry Pi and Arduino, these will enable cheap bioprocess control and monitoring of silent variables such as ambient temperature and humidity to be monitored more effectively. Use of these sensors in open controllers has been demonstrated for related fields such as plant cultivation [115], bee keeping [116], and wine fermentation [117].

The availability of cheap cameras presents an inadequately tapped opportunity to apply machine vision as a means to correct errors, simplify programming (at least for the user), and enable the use of cheaper hardware. Recently, some liquid handlers have been released that take advantage of this concept (Andrews Alliance).

These trends in cheap, open, and increasingly sophisticated hardware will play a key role in realizing the benefits discussed in this article.

Concluding Remarks and Future Perspectives

High-dimensional experimentation, using one or more of the techniques discussed in this review, is essential for harnessing biological potential, but the experiments required are often complex and much more difficult to set up than a brute-force screen. When experimenting with very large numbers of factors, which is most likely necessary, experiments are beyond the limit of what humans can reliably perform. Solving this problem ultimately requires automation, but the current state of the art in laboratory automation is very much geared towards the need to do a single process repeatedly and is not well suited to implementing designed experiments.

Outstanding Questions

What is the best way to combine high-throughput screening with model-based experimentation?

What is the best structure for a high-level programming language for biology?

What are the most appropriate design goals for such a language: readability, expressive power, or some blend of the two?

How will we keep on top of all the data that can now be collected and generated?

How will security of data and automated facilities be maintained?

New developments in software approaches to protocol specification and implementation, design tools, and a new generation of low-cost liquid handlers are taking place that may be expected to provide solutions.

Of the three important technologies we have discussed, we believe the most crucial is the development and adoption of high-level machine languages for executable bioprocesses. The reasons for this can be simply enumerated:

(i) They provide an unambiguous medium of exchange and facilitate reuse
(ii) They remove the artificial separation between description of a process and its executable form
(iii) Abstraction and domain language based terminology enhance readability and remove unnecessary implementation details

(iv) Machine writing of processes is facilitated, enables automated optimization and avoids human error
(v) Practices for managing complexity in software development become available: version control, automated testing and validation, code validity checking, and others [118,119].

One notable feature in common with these languages is that they are all extensions of existing programming languages: C/C++ (BioCoder), Python (Autoprotocol and Go (Antha) – this is a necessary choice given the huge effort required to generate a full-featured programming language and is further justified by the need for it to be readily usable without significant effort to encourage adoption.

However, there are many features of the physical domain that are not easily represented by these traditional programming languages. The most appropriate way to structure a language for this purpose is not yet apparent and has not been widely discussed ([120] is a rare exception) and remains an interesting outstanding question that will only be answered when such languages are much more widely used.

Many initial attempts to link biological experimentation and computation have now appeared in the literature [121–123]. One identified cause of the productivity decline in the pharmaceutical industry, the reliance on brute-force screening, is clearly amenable to the solutions proposed using automated optimization [12]. Other reviewers have similarly argued that more advanced modeling will be a key part in improving the competitiveness of biotechnology [89].

At least one example in biotechnology has recently been published [124]. A number of recent developments in Bayesian experimental design may be useful in this regard. Liepe *et al*. [125] present a useful framework in which predictive accuracy is directly optimized, for instance, building on much other recent work in the area [126–130]. An alternative approach applied to systems biology has also recently been described [131]. Hybrid approaches that combine mechanistic and non-mechanistic models offer a very attractive framework for addressing the difficulties of combining the wealth and diversity of data that arise from conducting such complex experiments [132–134].

Linking biotechnology more firmly with computation will bring many challenges beyond the technical ones of making it all work correctly. Andreas *et al.* [135], in a stimulating article, discuss the substantial challenges involved in keeping track of all the data merely derived from experiments – particularly acute are issues relating to data security in every sense. However, the security issues involved in making laboratory equipment and reagents accessible across the internet are more substantial still even without the obvious technical problems.

Adoption of these methods remains a significant challenge (see Outstanding Questions). Will expert molecular biologists and biotechnologists be willing to give up their pipettes and accept a more abstracted, less hands-on view of their experiments? In some ways this is difficult to imagine; however, one of us remembers a time before kits were commonplace in molecular biology and similar questions being asked, a fact which seems amusing in hindsight given their present ubiquity [136]. It is to be hoped that these new technologies will be embraced as enthusiastically and to such positive effect.

References

1. Isaacson, W. (1999) The biotech century. *Time* 22 March
2. Rifkin, J. (1999) *The Biotech Century: Harnessing the Gene and Remaking the World,* Jeremy P. Tarcher/Putnam
3. Cantor, C.R. (2000) Biotechnology in the 21st century. *Trends Biotechnol.* 18, 6–7
4. McGloughlin, M. (1999) Ten reasons why biotechnology will be important to the developing world. *Agbioforum* 2, 163–174
5. Denmark, W.W.F. (2009) *Industrial Biotechnology: More Than Green Fuel in a Dirty Economy?* WWF Denmark
6. Swaminathan, M.S. (2010) Achieving food security in times of crisis. *New Biotechnol.* 27, 453–460
7. Sexton, S. *et al.* (2009) The role of biotechnology in a sustainable biofuel future. *Agbioforum* 12, 130–140
8. Mitchell, P. (2008) US credit crunch impacts biotech across the globe. *Nat. Biotech.* 26, 359–360
9. Smith, G. *et al.* (2009) Wasting cash – the decline of the British biotech sector. *Nat. Biotechnol.* 27, 531–537
10. Kessel, M. (2011) The problems with today's pharmaceutical business – an outsider's view. *Nat. Biotechnol.* 29, 27–33
11. Cooke, P. (2013) Are biotechnology and its clusters in crisis? *Technol. Anal. Strat. Manag.* 25, 785–798
12. Scannell, J.W. *et al.* (2012) Diagnosing the decline in pharmaceutical R&D efficiency. *Nat. Rev. Drug Discov.* 11, 191–200
13. Whipple, T. (2012) Gene 'revolution' has stalled, says Winston. *The Times* 9 June, 3
14. Prinz, F. *et al.* (2011) Believe it or not: how much can we rely on published data on potential drug targets? *Nat. Rev. Drug Discov.* 10, 712
15. Baker, M. (2015) First results from psychology's largest reproducibility test. *Nature* Published April 30, 2015. http://dx.doi.org/10.1038/nature.2015.17433
16. Kwok, R. (2010) Five hard truths for synthetic biology. *Nature* 463, 288–290
17. Glasziou, P. *et al.* (2008) What is missing from descriptions of treatment in trials and reviews? *BMJ* 336, 1472–1474
18. Steward, O. *et al.* (2012) Replication and reproducibility in spinal cord injury research. *Exp. Neurol.* 233, 597–605
19. Begley, C.G. and Ellis, L.M. (2012) Drug development: raise standards for preclinical cancer research. *Nature* 483, 531–533
20. Freedman, L.P. *et al.* (2015) The economics of reproducibility in preclinical research. *PLoS Biol.* 13, e1002165
21. Button, K.S. *et al.* (2013) Power failure: why small sample size undermines the reliability of neuroscience. *Nat. Rev. Neurosci.* 14, 365–376
22. Ioannidis, J.P. *et al.* (2014) Increasing value and reducing waste in research design, conduct, and analysis. *Lancet* 383, 166–175
23. Vasilevsky, N.A. *et al.* (2013) On the reproducibility of science: unique identification of research resources in the biomedical literature. *PeerJ.* 1, e148
24. Fang, F.C. *et al.* (2012) Misconduct accounts for the majority of retracted scientific publications. *Proc. Natl. Acad. Sci. U.S.A.* 109, 17028–17033
25. Begley, C.G. (2013) Reproducibility: six red flags for suspect work. *Nature* 497, 433–434
26. McDermott, J.E. (2013) Reproducibility: two more red flags for suspect work. *Nature* 499, 284
27. Baker, M. (2015) Reproducibility crisis: blame it on the antibodies. *Nature* 521, 274–276
28. Ioannidis, J.P.A. (2014) How to make more published research true. *PLoS Med.* 11, e1001747
29. Casadevall, A. *et al.* (2014) Sources of error in the retracted scientific literature. *FASEB J.* 28, 3847–3855
30. Salter, S.J. *et al.* (2014) Reagent and laboratory contamination can critically impact sequence-based microbiome analyses. *BMC Biol.* 12, 87
31. Djulbegovic, B. and Hozo, I. (2014) Effect of initial conditions on reproducibility of scientific research. *Acta Inform. Med.* 22, 156
32. Elfving, G. (1952) Optimum Allocation in Linear Regression Theory. *Ann Math. Stat.* 23, 255–262
33. Fisher, R.A. (1974) *The Design of Experiments,* Hafner Press
34. Box, G. and Wilson, K. (1951) On the experimental attainment of optimum conditions. *J. R. Stat. Soc. Ser. B* 13, 1–45
35. Haaland, P.D. (1989) *Experimental Design in Biotechnology,* Marcel Dekker
36. Mukerjee, R. and Wu, C.F.J. (2007) *A Modern Theory of Factorial Design,* Springer
37. Gilmour, S. (2006) Factor screening via supersaturated designs. In *Screening* (Dean, A. and Lewis, S., eds), pp. 169–190, Springer
38. Jones, B. and Majumdar, D. (2014) Optimal supersaturated designs. *J. Am. Stat. Assoc.* 109, 1592–1600
39. Viader-Salvadó, J.M. *et al.* (2013) Optimization of five environmental factors to increase beta-propeller phytase production in *Pichia pastoris* and impact on the physiological response of the host. *Biotechnol. Prog.* 29, 1377–1385
40. Nagashima, H. *et al.* (2013) Application of a quality by design approach to the cell culture process of monoclonal antibody production, resulting in the establishment of a design space. *J. Pharm. Sci.* 102, 4274–4283
41. Rajeswari, P. *et al.* (2014) Characterization of saltern based *Streptomyces* sp. and statistical media optimization for its improved antibacterial activity. *Front. Microbiol.* 5, 753
42. Roessl, U. *et al.* (2015) Design of experiments reveals critical parameters for pilot-scale freeze-and-thaw processing of L-lactic dehydrogenase. *Biotechnol. J.* 10, 1390–1399
43. Kumar, V. *et al.* (2014) Design of experiments applications in bioprocessing: concepts and approach. *Biotechnol. Prog.* 30, 86–99
44. Weuster-Botz, D. (2000) Experimental design for fermentation media development: statistical design or global random search? *J. Biosci. Bioeng.* 90, 473–483
45. Collins, N. (2003) Culture medium optimization and scale-up for microbial fermentations. In *Handbook of Industrial Cell Culture Mammalian, Microbial, and Plant Cells* (Vinci, V. and Parekh, S. R., eds), pp. 171–193, Humana Press
46. Mandenius, C-F. and Brundin, A. (2008) Bioprocess optimization using design-of-experiments methodology. *Biotechnol. Prog.* 24, 1191–1203
47. Gurunathan, B. and Sahadevan, R. (2011) Design of experiments and artificial neural network linked genetic algorithm for modeling and optimization of L-asparaginase production by Aspergillus terreus MTCC 1782. *Biotechnol. Bioprocess Eng.* 16, 50–58
48. Govindarajan, S. *et al.* (2015) Mapping of amino acid substitutions conferring herbicide resistance in wheat glutathione transferase. *ACS Synth. Biol.* 4, 221–227
49. Weski, J. and Ehrmann, M. (2012) Genetic analysis of 15 protein folding factors and proteases of the *Escherichia coli* cell envelope. *J. Bacteriol.* 194, 3225–3233

50. Farasat, I. *et al.* (2014) Efficient search, mapping, and optimization of multi-protein genetic systems in diverse bacteria. *Mol. Syst. Biol.* 10, 731

51. Zhou, H. *et al.* (2015) Algorithmic co-optimization of genetic constructs and growth conditions: application to 6-ACA, a potential nylon-6 precursor. *Nucleic Acids Res.* Published online October 30, 2015. http://dx.doi.org/10.1093/nar/gkv1071

52. Grant, C. *et al.* (2012) Tools for characterizing the whole-cell bio-oxidation of alkanes at microscale. *Biotechnol. Bioeng.* 109, 2179–2189

53. Mutalik, V.K. *et al.* (2013) Quantitative estimation of activity and quality for collections of functional genetic elements. *Nat. Methods* 10, 347–353

54. Montgomery, D.C. (2013) *Statistical Quality Control: A Modern Introduction.* (7th edn), Wiley

55. Lawrence, X.Y. (2008) Pharmaceutical quality by design: product and process development, understanding, and control. *Pharm. Res.* 25, 781–791

56. Rathore, A.S. (2009) Roadmap for implementation of quality by design (QbD) for biotechnology products. *Trends Biotechnol.* 27, 546–553

57. Yu, L.X. *et al.* (2014) Understanding pharmaceutical quality by design. *AAPS J.* 16, 771–783

58. Mercier, S.M. *et al.* (2014) Multivariate PAT solutions for biopharmaceutical cultivation: current progress and limitations. *Trends Biotechnol.* 32, 329–336

59. Jiang, C. *et al.* (2010) Defining process design space for a hydrophobic interaction chromatography (HIC) purification step: application of quality by design (QbD) principles. *Biotechnol. Bioeng.* 107, 985–997

60. Bhambure, R. and Rathore, A.S. (2013) Chromatography process development in the quality by design paradigm. I: establishing a high-throughput process development platform as a tool for estimating ‘characterization space’ for an ion exchange chromatography step. *Biotechnol. Prog.* 29, 403–414

61. Lie, A. *et al.* (2013) Design of experiments and multivariate analysis for evaluation of reversed-phase high-performance liquid chromatography with charged aerosol detection of sucrose caprate regioisomers. *J. Chromatogr. A* 1281, 67–72

62. Wagdy, H.A. *et al.* (2013) Determination of the design space of the HPLC analysis of water-soluble vitamins. *J. Sep. Sci.* 36, 1703–1710

63. Fang, K. *et al.* (2006) *Design and Modeling for Computer Experiments,* Chapman & Hall/CRC

64. Shmueli, G. (2010) To explain or to predict? *Stat. Sci.* 25, 289–310

65. Gene Ontology Consortium (2015) Gene Ontology Consortium: going forward. *Nucleic Acids Res.* 43, D1049–D1056

66. Andreeva, A. *et al.* (2014) SCOP2 prototype: a new approach to protein structure mining. *Nucleic Acids Res.* 42, D310–D314

67. Sillitoe, I. *et al.* (2015) CATH: comprehensive structural and functional annotations for genome sequences. *Nucleic Acids Res.* 43, D376–D381

68. Mungall, C.J. *et al.* (2011) Evolution of the Sequence Ontology terms and relationships. *J. Biomed. Inform.* 44, 87–93

69. Courtot, M. *et al.* (2014) Controlled vocabularies and semantics in systems biology. *Mol. Syst. Biol.* 7, 543

70. Galdzicki, M. *et al.* (2014) The Synthetic Biology Open Language (SBOL) provides a community standard for communicating designs in synthetic biology. *Nat. Biotechnol.* 32, 545–550

71. Antezana, E. *et al.* (2009) The Cell Cycle Ontology: an application ontology for the representation and integrated analysis of the cell cycle process. *Genome Biol.* 10, R58

72. Lefever, S. *et al.* (2009) RDML: structured language and reporting guidelines for real-time quantitative PCR data. *Nucleic Acids Res.* 37, 2065–2069

73. Deutsch, E.W. *et al.* (2015) Development of data representation standards by the human proteome organization proteomics standards initiative. *J. Am. Med. Inform. Assoc.* 22, 495–506

74. Hailemariam, L. and Venkatasubramanian, V. (2010) Purdue ontology for pharmaceutical engineering: part I. Conceptual framework. *J. Pharm. Innov.* 5, 88–99

75. Malone, J. *et al.* (2010) Modeling sample variables with an Experimental Factor Ontology. *Bioinformatics* 26, 1112–1118

76. Bilitchenko, L. *et al.* (2011) Eugene – a domain specific language for specifying and constraining synthetic biological parts, devices, and systems. *PLoS ONE* 6, e18882

77. Miller, A.K. *et al.* (2010) An overview of the CellML API and its implementation. *BMC Bioinformatics* 11, 178

78. Sedlmajer, N. *et al.* (2011) GReg: a domain specific language for the modeling of genetic regulatory mechanisms. *BioPPN 2011* 724, 21–35

79. Cai, Y. *et al.* (2007) A syntactic model to design and verify synthetic genetic constructs derived from standard biological parts. *Bioinformatics* 23, 2760–2767

80. Beal, J. and Bachrach, J. (2006) Infrastructure for engineered emergence on sensor/actuator networks. *IEEE Intell. Syst.* 21, 10–19

81. Beal, J. and Bachrach, J. (2008) Cells are plausible targets for high-level spatial languages. In *Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops*, pp. 284–291, IEEE

82. Ananthanarayanan, V. and Thies, W. (2010) Biocoder: a programming language for standardizing and automating biology protocols. *J. Biol. Eng.* 4, 13

83. Grissom, D. *et al.* (2015) An open-source compiler and PCB synthesis tool for digital microfluidic biochips. *Integr. VLSI J.* 51, 169–193

84. McDaniel, J. *et al.* (2013) Automatic synthesis of microfluidic large scale integration chips from a domain-specific language. In *Biomsedical Circuits and Systems Conference (BioCAS)*, pp. 101–104, IEEE

85. Donovan, A.A.A. and Kernighan, B.W.A. (2015) *The Go Programming Language,* Addison-Wesley

86. Rios-Solis, L. *et al.* (2015) Modelling and optimisation of the one-pot, multi-enzymatic synthesis of chiral amino-alcohols based on microscale kinetic parameter determination. *Chem. Eng. Sci.* 122, 360–372

87. Tufvesson, P. *et al.* (2013) Advances in the process development of biocatalytic processes. *Org. Process Res. Dev.* 17, 1233–1238

88. Micheletti, M. and Lye, G.J. (2006) Microscale bioprocess optimisation. *Curr. Opin. Biotechnol.* 17, 611–618

89. Van Dien, S. (2013) From the first drop to the first truckload: commercialization of microbial processes for renewable chemicals. *Curr. Opin. Biotechnol.* 24, 1061–1068

90. Lima-Ramos, J. *et al.* (2014) Application of environmental and economic metrics to guide the development of biocatalytic processes. *Green Process. Synth.* 3, 195–213

91. Beal, J. (2015) Bridging the gap: a roadmap to breaking the biological design barrier. *Front. Bioeng. Biotechnol.* 2, 87

92. Linshiz, G. *et al.* (2014) PR-PR: cross-platform laboratory automation system. *ACS Synth. Biol.* 3, 515–524

93. Linshiz, G. *et al.* (2013) PaR-PaR laboratory automation platform. *ACS Synth. Biol.* 2, 216–222

94. Xia, B. *et al.* (2011) Developer's and user's guide to Clotho v2.0. A software platform for the creation of synthetic biological systems. *Methods Enzym.* 498, 97–135

95. Hillson, N.J. *et al.* (2012) j5 DNA assembly design automation software. *ACS Synth. Biol.* 1, 14–21

96. Hillson, N.J. (2014) j5 DNA assembly design automation. In *DNA Cloning and Assembly Methods* (Valla, S. and Lale, R., eds), pp. 245–269, Humana Press

97. Chen, J. *et al.* (2012) DeviceEditor visual biological CAD canvas. *J. Biol. Eng.* 6, 1

98. Ham, T.S. *et al.* (2012) Design, implementation and practice of JBEI-ICE: an open source biological part registry platform and tools. *Nucleic Acids Res.* 40, e141

99. Appleton, E. *et al.* (2014) Interactive assembly algorithms for molecular cloning. *Nat. Methods* 11, 657–662

100. Bhatia, S. and Densmore, D. (2013) Pigeon: a design visualizer for synthetic biology. *ACS Synth. Biol.* 2, 348–350

101. MacDonald, J.T. *et al.* (2011) Computational design approaches and tools for synthetic biology. *Integr. Biol. Quant. Biosci. Nano Macro* 3, 97–108

102. Kelwick, R. *et al.* (2014) Developments in the tools and methodologies of synthetic biology. *Front. Bioeng. Biotechnol.* 2, 60

103. Chandran, D. *et al.* (2009) TinkerCell: modular CAD tool for synthetic biology. *J. Biol. Eng.* 3, 19

104. Chandran, D. *et al.* (2010) Computer-aided design of biological circuits using tinkercell. *Bioeng. Bugs* 1, 276–283

105. Rodrigo, G. and Jaramillo, A. (2013) AutoBioCAD: full biodesign automation of genetic circuits. *ACS Synth. Biol.* 2, 230–236

106. Herwig, C. *et al.* (2015) Knowledge management in the QbD paradigm: manufacturing of biotech therapeutics. *Trends Biotechnol.* 33, 381–387

107. Berridge, J.C. (1989) Advances in automation of pharmaceutical analysis. *J. Pharm. Biomed. Anal.* 7, 1313–1321

108. Lye, G.J. *et al.* (2003) Accelerated design of bioconversion processes using automated microscale processing techniques. *Trends Biotechnol.* 21, 29–37

109. Kuznetsov, S. *et al.* (2015) DIYbio things: open source biology tools as platforms for hybrid knowledge production and scientific participation. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 4065–4068, ACM

110. Kay, M. *et al.* (2011) *mHealth: New Horizons for Health Through Mobile Technologies,* World Health Organization

111. Zhu, H. *et al.* (2011) Cost-effective and compact wide-field fluorescent imaging on a cell-phone. *Lab Chip* 11, 315–322

112. Balsam, J. *et al.* (2015) Smartphone-based fluorescence detector for mHealth. *Methods Mol. Biol.* 1256, 231–245

113. Balsam, J. *et al.* (2013) Low-cost technologies for medical diagnostics in low-resource settings. *Expert Opin. Med. Diagn.* 7, 243–255

114. Markovic, N. and Markovic, O. Bioscion Inc. Bioscicon's cellphone camera-microscope universal adapter, US 20150036043 A1

115. Mainwaring, A. *et al.* (2002) Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 88–97, ACM

116. Fitzgerald, D.W. *et al.* (2015) Design and development of a smart weighing scale for beehive monitoring. In *26th Irish Signals and Systems Conference (ISSC)*, pp. 1–6, IEEE

117. Di Gennaro, S.F. *et al.* (2014) An open-source and low-cost monitoring system for precision enology. *Sensors* 14, 23388–23397

118. Sandve, G.K. *et al.* (2013) Ten simple rules for reproducible computational research. *PLoS Comput. Biol.* 9, e1003285

119. Ram, K. (2013) Git can facilitate greater reproducibility and increased transparency in science. *Source Code Biol. Med.* 8, 7

120. Brenner, S. (2010) Sequences and consequences. *Philos. Trans. R. Soc. B: Biol. Sci.* 365, 207–212

121. Zhang, X. *et al.* (2009) Metabolic evolution of energy-conserving pathways for succinate production in *Escherichia coli*. *Proc. Natl. Acad. Sci. U.S.A.* 106, 20180–20185

122. Zhang, Y. *et al.* (2015) ChiNet uncovers rewired transcription subnetworks in tolerant yeast for advanced biofuels conversion. *Nucleic Acids Res.* 43, 4393–4407

123. Ismail, M.A. *et al.* (2015) A Newton cooperative genetic algorithm method for in silico optimization of metabolic pathway production. *PLoS ONE* 10, e0126199

124. Pathak, L. *et al.* (2015) Artificial intelligence versus statistical modeling and optimization of cholesterol oxidase production by using *Streptomyces* sp. *PLoS ONE* 10, e0137268

125. Liepe, J. *et al.* (2013) Maximizing the information content of experiments in systems biology. *PLoS Comput. Biol.* 9, e1002888

126. Vanlier, J. *et al.* (2012) A Bayesian approach to targeted experiment design. *Bioinformatics* 28, 1136–1142

127. Vanlier, J. *et al.* (2012) An integrated strategy for prediction uncertainty analysis. *Bioinformatics* 28, 1130–1135

128. Huan, X. and Marzouk, Y.M. (2013) Simulation-based optimal Bayesian experimental design for nonlinear systems. *J. Comput. Phys.* 232, 288–317

129. Komorowski, M. *et al.* (2011) Sensitivity, robustness, and identifiability in stochastic chemical kinetics models. *Proc. Natl. Acad. Sci. U.S.A.* 108, 8645–8650

130. Apgar, J.F. *et al.* (2010) Sloppy models, parameter uncertainty, and the role of experimental design. *Mol. Biosyst.* 6, 1890

131. Pauwels, E. *et al.* (2014) A Bayesian active learning strategy for sequential experimental design in systems biology. *BMC Syst. Biol.* 8, 102

132. Zalai, D. *et al.* (2015) Combining mechanistic and data-driven approaches to gain process knowledge on the control of the metabolic shift to lactate uptake in a fed-batch CHO process. *Biotechnol. Prog.* Published online October 6, 2015. http://dx.doi.org/10.1002/btpr.2179

133. Ferreira, A.R. *et al.* (2014) Fast development of *Pichia pastoris* GS115 Mut+ cultures employing batch-to-batch control and hybrid semi-parametric modeling. *Bioprocess Biosyst. Eng.* 37, 629–639

134. Von Stosch, M. *et al.* (2014) Hybrid semi-parametric modeling in process systems engineering: past, present and future. *Comput. Chem. Eng.* 60, 86–101

135. Andreas, H. *et al.* (2014) Knowledge discovery and interactive data mining in bioinformatics-state-of-the-art, future challenges and research directions. *BMC Bioinformatics* 15, I1

136. Weiner, M.P. and Slatko, B.E. (2008) Kits and their unique role in molecular biology: a brief retrospective. *Biotechniques* 44, 701