

PERSPECTIVE

<https://doi.org/10.1038/s41467-020-14314-z>

OPEN

Organizing genome engineering for the gigabase scale

Bryan A. Bartley ^{1,4}, Jacob Beal ^{1,4*}, Jonathan R. Karr ^{2,4} & Elizabeth A. Strychalski^{3,4}

Genome-scale engineering holds great potential to impact science, industry, medicine, and society, and recent improvements in DNA synthesis have enabled the manipulation of megabase genomes. However, coordinating and integrating the workflows and large teams necessary for gigabase genome engineering remains a considerable challenge. We examine this issue and recommend a path forward by: 1) adopting and extending existing representations for designs, assembly plans, samples, data, and workflows; 2) developing new technologies for data curation and quality control; 3) conducting fundamental research on genome-scale modeling and design; and 4) developing new legal and contractual infrastructure to facilitate collaboration.

Engineering the entire genome of an organism promises to enable large-scale changes to its organization, function, and interactions with its environment, with broad potential for impacts across science, industry, medicine, and society¹. The past several decades have seen remarkable progress in our capability to synthesize DNA and modify genomes^{2–4}. Since Khorana created the first synthetic gene 40 years ago⁵, our capability to construct DNA sequences has doubled, approximately every 3 years (Fig. 1a), progressing from plasmids in the early 1990's^{6,7}, viruses in the early 2000's⁸, and gene clusters in the mid-2000's^{9,10}, to the first bacterial chromosome in 2008^{11,12}. Recently, several groups have re-engineered the 4 Mb genomes of *Escherichia coli*^{13,14} and *Salmonella typhimurium*¹⁵, and the Synthetic Yeast (Sc 2.0) project^{16,17} has nearly completed re-engineering an 11.4 Mb genome for *Saccharomyces cerevisiae*¹⁸. Looking ahead, in 2016 leaders from academia and industry formed Genome Project-Write¹ to initiate the engineering of the gigabase genomes of higher-order eukaryotes. The goals of the GP-Write consortium include engineering a virus-resistant, ultra-safe human-derived cell line for pharmaceutical production¹⁹.

From engineering genes to engineering genomes

Moving to the gigabase scale poses major technological and scientific challenges. Challenges related to DNA synthesis and editing have been discussed extensively in the literature^{20–23}. Significant attention has also been devoted to the challenges of modeling^{24,25}, designing^{17,26,27}, and testing²⁸ genomes. Less attention, however, has been devoted to the technologies, repositories, standards, and other resources needed to integrate these tasks into a cohesive workflow.

We contend that workflow integration is a first-class problem for gigabase-scale genome engineering. Over the last 40 years, the number of authors of pioneering genome engineering

¹Raytheon BBN Technologies, Cambridge, MA 02138, USA. ²Icahn Institute and Department of Genetics and Genomic Sciences, Icahn School of Medicine at Mount Sinai, New York, NY 10128, USA. ³National Institute of Standards and Technology, Gaithersburg, MD 20899, USA. ⁴These authors contributed equally: Bryan A. Bartley, Jacob Beal, Jonathan R. Karr, Elizabeth A. Strychalski. *email: jakebeal@ieee.org

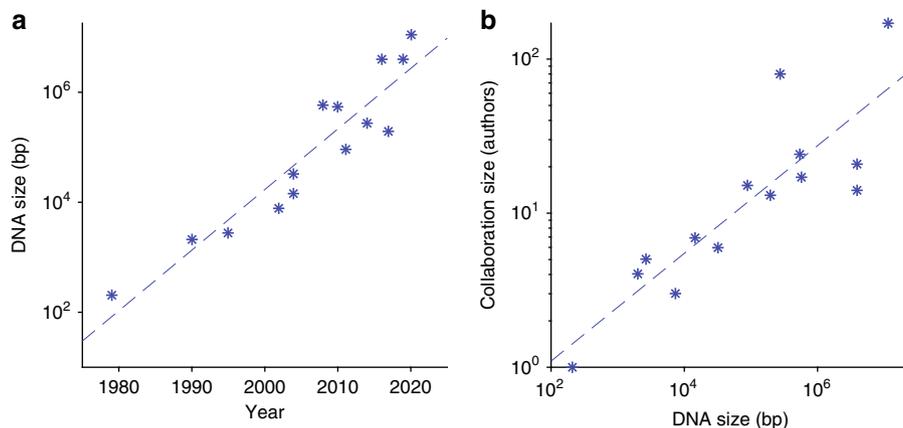


Fig. 1 As capabilities for genome engineering have advanced rapidly, the size of teams involved in each pioneering genome engineering project has also increased. **a** From 1980 to present, the size of the largest engineered genomes has grown exponentially, doubling approximately every 3 years. This trend suggests that gigabase engineering could become feasible by 2050. **b** The number of authors credited with producing these genomes has also grown exponentially. This trend suggests that engineering gigabase genomes will require the effort of ~500 individuals—either directly as part of a team or indirectly through an ecosystem of tools, services, automation, and other resources. The data for this figure are provided in Table 1.

Table 1 Year, genome size (bp), and the number of authors involved in pioneering genome engineering projects of the last 30 years.

Year	DNA size (bp)	Collaboration size (# authors)	Reference	Notes
1979	207	1	Khorana ⁵	First synthetic gene
1990	2050	4	Mandecki et al. ⁶	First synthetic plasmid
1995	2700	5	Stemmer et al. ⁷	Synthetic plasmid
2002	7.5000E+03	3	Cello et al. ⁸	Polio virus cDNA
2004	1.4600E+04	7	Tian et al. ⁹	rRNA genes
2004	3.1656E+04	6	Kodumal et al. ¹⁰	Gene cluster
2008	5.8297E+05	17	Gibson et al. ¹¹	Mycoplasma genitalium
2010	5.3100E+05	24	Gibson et al. ¹²	Mycoplasma mycoide, JCVI synthetic cell
2011	9.1010E+04	15	Dymond et al. ¹⁶	Sc 2.0 synIXR
2014	2.7287E+05	80	Annaluru et al. ¹⁷	Yeast chromosome synIII
2016	3.9700E+06	21	Ostrov et al. ¹³	Partially recoded <i>E. coli</i> , 62 K edits in genome
2017	2.0000E+05	13	Lau et al. ¹⁵	Salmonella typhimurium partial genome
2019	4.0000E+06	14	Fredens et al. ¹⁴	Recoded <i>E. coli</i>
2020	1.14E+07	172	Richardson et al. ¹⁸	Sc 2.0 estimated completion date; Genome size from Table 3 in reference; Collaboration size estimated from Sc 2.0 website

These data are plotted in Fig. 1

projects has risen markedly with genome size, suggesting that the complexity of genome engineering is also scaling with the size of the genome (Fig. 1b). If these trends continue, engineering a gigabase genome would be projected to become possible in ~2050 and require a team with the capabilities of around 500 investigators. To manage projects of such complexity without massive teams, we advocate for the development of an ecosystem of tools, services, automation, and other resources, which could enable a modestly sized team of bioengineers to indirectly access the equivalent capabilities of hundreds of people. To this end, we have examined the emerging design–build–test–learn workflow for genome engineering, identifying key interfaces and making recommendations for the adoption or development of technologies, repositories, standards, and frameworks.

An emerging workflow for genome engineering

Recently, a number of groups have proposed or developed workflows for organism engineering^{3,18,27–32}, converging toward a common engineering cycle consisting of the four stages shown in Fig. 2. These stages are (1) Design: bioengineers use models

and design heuristics to specify a genome with an intended phenotype; (2) Build: genetic engineers construct the desired DNA sequence in a target organism; (3) Test: experimentalists assay molecular and behavioral phenotypes of the engineered organism; (4) Learn: modelers analyze the discrepancies between the desired and observed phenotypes to develop improved models and design heuristics. The process is repeated until an organism with the desired phenotype is identified. This incremental approach enables engineering despite our incomplete understanding of the complexities of biology.

The inner loop in Fig. 2 indicates the workflow used by many current genome engineering projects, which have primarily focused on “top-down” refactoring of existing genomes, e.g., by rewriting codons or reducing genomes to essential sequences. In the longer term, one of the key aims of synthetic biology is to engineer organisms that have novel phenotypes by “bottom-up” assembly of modular parts and devices³³. At a much smaller scale, organism engineers are already beginning to use this approach to engineer novel metabolic pathways for commercial production of high-value chemicals^{34–36}. For gigabase genome engineering, this

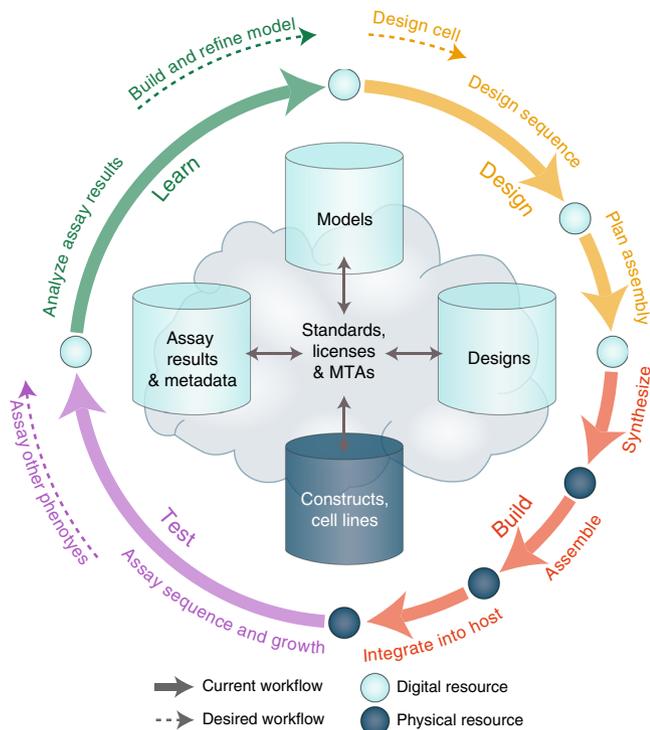


Fig. 2 The emerging design–build–test–learn workflow for genome engineering is shown schematically with current (solid arrows) and likely future (dashed arrows) tasks, interfaces (circles), and repositories (cylinders), either digital (light) or physical (dark).

approach will likely require more complex workflows that utilize more sophisticated design tools, phenotypic assays, data analytics, and models (outer loop of Fig. 2).

Executing these multistep workflows requires exchanging a wide range of materials, information, and other resources between numerous tools, people, institutions, and repositories. The design phase must communicate genome designs to the build phase, the build phase must deliver DNA constructs and cell lines to the test phase, the test phase must transmit measurements to the learn phase, the learn phase must provide models and design heuristics to the design phase, and workflows must be applied to coordinate the interaction and execution of tools across all of these stages.

In addition to these technical challenges, genome engineering must also address a number of safety, security, legal, contractual, and ethical issues. Throughout genome engineering workflows, bioengineers must pay careful attention to biosafety, biosecurity, and cybersecurity. To execute genome engineering workflows across multiple institutions, bioengineers must navigate materials transfer agreements, copyrights, patents, and licenses.

Every aspect of this genome engineering workflow must be scaled up to handle gigabase genomes. Ultimately, much or all of each step should be automated, and each interface between steps should be formalized to facilitate machine reasoning, removing the ad hoc and human-centric aspects of genome engineering as much as possible. In many cases, this can be facilitated by adopting or extending solutions from smaller-scale genome engineering, as well as solutions from related fields such as systems biology, genomics, genetics, bioinformatics, software engineering, database engineering, and high-performance computing. Other challenges of gigabase genome engineering, however, are likely to require the development of novel systems or additional fundamental research.

Identifying and closing gaps in the state of the art

In this section, we discuss the integration challenges identified in the previous section, reviewing the state of the art in technologies and standards with respect to the emerging needs of gigabase genome engineering. Instead of focusing on specific evolving protocols and methods, which are likely to advance rapidly, we consider the information that must be communicated to enable protocols or methods to be composed into a comprehensive workflow. Through this analysis, we identify critical gaps and opportunities, where additional technologies and standards would facilitate workflows that can effectively deliver gigabase engineered genomes. Table 2 summarizes the potential solutions that we have identified, which are detailed in the following subsections.

Genome refactoring and design

Current genome engineering projects have focused primarily on refactoring genomes while preserving their cellular function. For example, three recent projects have involved eliminating non-essential elements²⁷, reordering genes¹⁷, and inserting metabolic pathways³⁷. At this level, two critical challenges for scaling are accessing well-annotated source genomes and representing and exchanging designs for modified genomes. More complex changes of organism function will pose additional challenges related to composing parts to produce novel cellular functions.

Currently, genome design generally involves modifying pre-existing organism sequences, such as those available in the public archives of the International Nucleotide Sequence Database Collaboration (INSDC)³⁸, which currently contains ~10⁵ bacterial genomes and hundreds of eukaryotic genomes^{39–43}. Functional annotation is key, as genome engineers will need to consider tissue-specific expression patterns, regulatory elements, structural elements, replication origins, clinically significant sites of DNA recombination and instability, etc. The consistency of annotations is a key challenge, as many genomes have been annotated by different toolchains that produce significantly different annotations. For example, the human reference genomes generated by the RefSeq and GENCODE projects have notable differences^{44,45} with likely engineering consequences, such as ability to predict loss-of-function from interaction with alternative splicings. Much of this knowledge is also dispersed among different resources, though annotations can be integrated with the aid of services such as NCBI Genome Viewer⁴⁶, WebGestalt⁴⁷, and DAVID⁴⁸. For moving to the gigabase scale, improved annotation APIs will be valuable, as would estimates of the confidence and reliability of annotations, such as the RefSeq database does with the Evidence and Conclusion Ontology⁴⁹.

The gigabase scale poses challenges for the representation and exchange of genome designs as well. Common formats such as GenBank and EMBL are monolithic in their treatment of sequences, which makes it difficult to integrate or harmonize editing across multiple concurrent users, and can even cause difficulties in simply transferring the data. Two formats better suited for genome engineering are the Generic Feature Format (GFF) version 3 and the Synthetic Biology Open Language (SBOL) version 2⁵⁰. GFF3 allows hierarchical organization of sequence descriptions (e.g., genes may be organized into clusters, and clusters into chromosomes), uses the Sequence Ontology⁵¹ to annotated sequences, and has already been used in the Sc 2.0 genome engineering project¹⁸. SBOL 2 is also routinely used for hierarchical description of edited genomes⁵² and can interoperate with GFF3 (though GFF3 only represents a subset of SBOL)⁵³. SBOL provides a richer design-centric language, including support for variants, libraries, and partial designs (e.g., identifying genes in a cluster, but not yet particular variants or cluster

Table 2 Potential approaches for integrating the emerging gigabase engineering workflow, labeled for reference.

Stage	Interface between workflow stages	Recommendation	Rationale
Design	Import design materials	Extend: annotation methods, ECO	Current genome annotation tools have problems of scale, consistency, and integration that can be addressed by better leveraging ontologies like the Evidence and Conclusion Ontology
Design	Design sequence → Plan assembly, Sharing designs	Extend: GFF3 and/or SBOL with chromosomal coordinates, BpForms	GFF3 and SBOL are both effective formats for genome design specifications, but index-based coordinates are fragile; BpForms will also allow annotation of sequences with non-standard bases and modifications
Design	Plan assembly → Build	Adopt: FASTA/GenBank/GFF, migrating to SBOL (function & plan for manufacturability flexibility)	FASTA, GenBank, GFF and SBOL formats all readily encode sequences to build. SBOL can also encode expected function and assembly plans, allowing greater flexibility with respect to manufacturing constraints.
Design	Build and refine model → Design cell	Research: integrated CAD across multiple models	Current methods are not close to the needed capabilities.
Build	Synthesize → Assemble → Integrate into host	Extend: FASTQ, GVF, and/or SBOL with quality metrics & requirements	FASTQ and GVF encode variation, but not requirements; SBOL can encode both, but is not a native format for current instruments. Informal practices regarding metrics and requirements need to be organized and formalized.
Build	Integrate into host → Assay sequence		
Build	Sharing of methods, constructs, and cell lines	Adopt: biomanufacturing best practices and LIMS	Many applicable systems and methods exist, and tradeoff for selection will depend on details of project and facilities involved.
Test	Integrate into host → Assay growth and phenotype	Develop: fitness metrics and associated specification language	Informal practices regarding metrics and requirements need to be organized and formalized.
		Research: integrate omics, other measurements, phenotypic specification	Current methods are not close to the needed capabilities.
Test	Comparison of assay results	Extend: process control and calibration standards	Some assays (e.g., RNaseq, flow cytometry) already have established practices for ensuring data validity; others need to have such practices developed.
Test	Sharing of assay results	Extend: ontologies and curation tools	Standardized vocabularies are available in the Open Biological and Biomedical Ontology (OBO) foundry; additional terms will be needed, along with better curation tools to reduce friction in usage.
Learn	Assay results → Analyze assay results	Extend: SBOL + OBO ontologies, additional metadata and knowledge curation tools	SBOL can link designs, assay plans, execution traces, and data, using the standardized vocabularies of the Open Biological and Biomedical Ontology (OBO) foundry; better curation tools are needed to reduce friction in usage.
Learn	Analyze assay results → Build and refine model	Research: automated and scalable model generation and verification	Current methods are not close to the needed capabilities.
Learn	Model composition and sharing	Extend: SBML, CellML, BioModels, and related COMBINE standards	The Computational Modeling in Biology Network (COMBINE) has already been developing a set of interoperable tools and standards for model sharing and composition, which provide a solid foundation for complex model-building.
		Research: standards for describing multi-scale models	Current methods are not close to the needed capabilities.
Cross-cutting	Workflow management	Adopt: CWL, PROV-O, SBOL 2.2 Design/Build/Test, container tools, team coordination tools	SBOL can represent elements at all stages of the cycle and PROV-O can link them; existing workflow languages and team coordination tools can manage machine and human workflows; existing container tools support portability.
		Develop: interoperable laboratory automation languages	Many applicable systems and methods exist, but capture different aspects of workflows and protocols and are not compatible with one another.
Cross-cutting	Database federation	Adopt: existing open DBMS solutions, FAIRDOMhub, EDD, or similar	Database federation tools are well developed, including domain-specific tools.
Cross-cutting	IP tracking and composition	Develop: framework based on OSI / CC / Science Commons, PROV-O	Prior efforts of the Open Source Initiative, Creative Commons, and Science Commons address much but not all of this challenge; PROV-O can be used for automation-assisted tracing of intellectual property usage.
Cross-cutting	Transfer of constructs and cell lines	Develop: legal/contractual framework building on OpenMTA	The OpenMTA supports materials transfers, but there are additional issues to address regarding legal/regulatory compliance and patents.
Cross-cutting	Management of sensitive or controlled information	Develop: framework based on cross-domain information sharing, PROV-O	Existing tools for information tracing and sharing need domain adaptation.

For each interface in the emerging workflow, our recommendations fall into one of three categories: adopt or extend relatively mature existing methods (green), develop new solutions or expand nascent methods (yellow), and conduct additional fundamental research (red)

arrangement), other elements and cellular functions (e.g., proteins, metabolic pathways, regulatory interactions). SBOL also interoperates with models encoded in the Systems Biology Markup Language (SBML)^{54,55}. Both GFF3 and SBOL, however, would benefit from more stable specifications of sequence positions within chromosomes, as sequence index is fragile to changes and sequence uncertainties. SBOL supports (and GFF3 could be extended to support) expression of nonstandard bases and sequence modifications in an enhanced sequence encoding language such as BpForms⁵⁶.

Representations of genome designs also need to express design constraints and policies, such as removal of restriction sites, separation of overlapping features, replacement of codons, and optimization for DNA synthesis. Projects such as Sc 2.0 have implemented this with a combination of guidelines for human hand-editing and custom software tools, and DNA synthesis providers provide interfaces to check for manufacturability constraints. At the gigabase scale, however, it will be beneficial to adopt more powerful and expressive languages for describing design policies, such as rule-based ontologies^{57,58}, and to include assembly and transformation plans in design representations to simplify adjustments for manufacturability. JGI's BOOST tool provides a prototype in this direction⁵⁹. SBOL is well-suited for this task, though GenBank and GFF3 could also, at least in principle, be extended to encode such information.

Modeling will become increasingly important as genome engineering moves beyond refactoring and recoding into more complex changes to an organism's function. Genome-scale metabolic models^{60,61} and whole-cell models⁶² can be

constructed by combining biochemical and genomic information from multiple databases, such as BioCyc⁶³ and the SEED⁶⁴. Models will also need to predict the behavior of organisms that are composed of separately characterized genetic parts, devices, pathways, and genome fragments. Substantial fundamental research still needs to be conducted to make such models practical at the gigabase scale.

Building engineered genomes

Technology and protocols for building engineered genomes are advancing rapidly, with potential paths to the gigabase scale discussed, for example, in ref. 1 and ref. 23. Depending on the specific host and intended function of the engineered organism, there are numerous potential approaches and protocols for DNA synthesis, assembly, and delivery. Currently, there is an unmet need for guidance on best practices for measuring, tracking, and sharing information regarding engineered genomes and intermediate samples.

Manipulating DNA during assembly offers ample opportunities for reduced yield, breakage, error, and other sources of uncertainty in achieving the designed DNA sequence. Protocols and commercial kits to assemble shorter DNA fragments into larger constructs often involve amplification, handling, purification, transformation, or other storage and delivery steps that can increase uncertainty in the quality and quantity of the DNA. Assembled DNA may also include added sequences that are not biologically active, as in the case for some methods using restriction enzymes, or scars, such as occur may occur with

Golden Gate Assembly⁶⁵ or MoClo⁶⁶. Gibson Assembly⁶⁷ is scarless, but the yield and specific results may depend on the secondary structure of the DNA fragments. Thus, in addition to sequence information, workflows will likely need extended representations that can also track the full range of information likely to affect assembly products, including DNA secondary structure, assembly method, sequences required for assembly and their location along the DNA molecule (e.g., landing pads or sequences for compatibility with protocol-hosting strains of *E. coli* or yeast), and intended epigenetic modifications. The results verifying both intermediate and final sequence construction are typically produced in the FASTQ format⁶⁸, which is generally sufficient for smaller constructs. To operate on large-scale genomes, however, more comprehensive descriptions of a genome and its variations may be made with representations such as GVF⁶⁹ or SBOL⁷⁰.

Suitable options for the delivery of large, assembled DNA constructs and whole genomes are generally lacking. The yield of existing processes, such as electrical and chemical transformation or genome transplantation, could be improved significantly to increase their utility, and a broader range of approaches should be developed for use with any organism and cell type. This may also require identifying new cell-free environments or cell-based chassis for assembling and manipulating DNA that also have compatibility with genome packaging and delivery systems into host organisms. To facilitate such development, delivery protocols and their associated information regarding number of biological and technical replicate experiments, methods, measurements, etc. should be available in a machine-readable format. This should include information regarding the host cell, such as its genotype, which is often not fully verified. The adoption of best practices from industrial biomanufacturing settings and implementation of laboratory information management systems (LIMS) could provide a path forward toward integrating appropriate measurements, process controls, and information handling, as well as the tracking and exchange of samples. Advancing the use of automation to support the build step of the genome engineering workflow requires evaluating which steps may reduce costs and speed results, the availability of automated methods, ways to effectively share those methods and adapt them across platforms and manufacturers, and ways to more simply integrate and tune automated workflows.

Testing the function of engineered genomes

Strain fitness and other phenotypes can be assessed via a wide range of biochemical and omics measurements, the details of which are beyond the scope of this discussion. In all cases, however, collaborating organizations will need to agree on specific measurements, along with control and calibration measurements, to ensure that the results can be compared and used across the participating laboratories.

DNA constructs are often evaluated for their associated growth phenotypes to determine the nature and extent of unexpected consequences for cell function and fitness due to the revised genome sequence. Engineered cell lines should also be evaluated for robustness to changes in the environmental context that the cells are likely to experience during typical use in the intended application, as well as stability over relevant timescales to evolution or adaptation. This is complicated by the need for shared definitions and measurements for fitness, metabolic burden, and other phenotypic properties.

Standard protocols, reference cell lines, and the use of experimental design are examples of tools available to increase the rigor and confidence in conclusions that can be drawn from testing. It will likely also be useful to develop standards and measurement

assurance for testing engineered genomes. Such foundations can be used to help identify relationships between genotype and phenotype or determine the contributions of biological stochasticity and measurement uncertainty to the overall variability in a measured trait, though comprehensive methods of this sort are likely to require significant fundamental research.

Calibration of biological assays aids in comparing results both within a single laboratory and across different laboratories. Recent studies, for example, for fluorescence^{71,72}, absorbance⁷³, and RNAseq⁷⁴ measurements, demonstrate the possibility of realizing scalable and cost-effective comparability in biological measurements. Organism engineering is likely to be facilitated by the development of additional calibrated measurement methods and absolute quantitation of an organism's properties.

Establishing shared representations and practices for metadata, process controls, and calibration will also be critical. Automation-assisted integration and comparison of the data, metadata, process controls, and calibration across laboratories will facilitate both the testing process and learning through modeling and simulation. Some existing ontologies can be leveraged for this purpose, such as the Experimental Conditions Ontology⁷⁵ (ECO), the Experimental Factor Ontology⁷⁶ (EFO), and the Measurement Method Ontology⁷⁵ (MMO). In addition, appropriate LIMS tooling and curation assistance software (e.g., RightField⁷⁷) will be vital for enabling such metadata to be created consistently, correctly, and in a timely fashion, by limiting the required input from human investigators.

Learning systematically from test results

As genome engineering affects systems throughout an organism, comprehensive models are needed that can help to both predict and interpret the relationship between genotype and phenotype. Although some models have been constructed for a whole cell⁶² or whole organism⁷⁸, developing and tuning such models is extremely challenging. To scale to gigabase genomes, it will be valuable to develop improved capabilities for creating, calibrating, and verifying models.

The first challenge in learning from the data is discovering and marshaling the data needed. Partial solutions exist, such as the workflow model introduced in SBOL 2.2⁵⁰, and ontologies such as the Open Biological and Biomedical Ontology⁷⁹, the Experimental Factor Ontology⁷⁶, the Systems Biology Ontology⁸⁰, and phenotype ontologies^{81,82}. These will need to be integrated and extended to cover the full range of needs for genome engineering.

Automation-assisted generation and verification of models at scale, however, still have many open fundamental research challenges, including addressing the combinatorial complexity of biology and the multiple scales between genomes and organismal behavior, high-performance simulation of large models, model verification, and representation of model semantic meaning and provenance^{24,25}.

Until we have comprehensive predictive models, engineers will likely rely on ad hoc combinations of predictive models of parts of organisms, data-driven models, and heuristic design rules. For example, constraint-based models are often used in metabolic engineering³⁴, PSORTb⁸³ can be used to help target proteins to specific compartments, and GC-content optimization can be used to improve host compatibility⁸⁴. Gigabase-scale genome engineering will require applying many such models simultaneously, and thus will benefit from adopting existing standard formats designed to facilitate biological model sharing and composition, such as SBML⁸⁵, CellML⁸⁶, NeuroML⁸⁷, and other standards in the Computational Modeling in Biology Network (COMBINE)⁸⁸. Large numbers of models in these formats can already be found in public databases, such as BioModels⁸⁹, the NeuroML

database⁹⁰, Open Source Brain⁹¹, and the Physiome Model Repository⁹². Similarly, repositories such as Kipoi⁹³ and the DockerHub repository⁹⁴ can already be used to share data-driven models. Further extensions to such formats, however, will be valuable for automating the learning process, including associating semantic meaning with model components, capturing the provenance of model elements (e.g., data sources, assumptions, and design motivations), and capturing information about their predictive capabilities and applicable scope.

To increase automation in learning such models from data, it will likely be valuable to develop new repositories of models of individual biological parts that can be composed into models of entire organisms^{95,96}; new methods for generating model variants that explain new observations by incorporating models of additional parts, alternative kinetic laws, or alternative parameter values; and new model selection techniques for nonlinear multiscale models⁹⁷.

Coordination and sharing in complex workflows

Tasks in isolation are not enough: efficient operation of the design–build–test–learn cycle for engineering gigabase genomes will require coordinating all of the numerous heterogeneous tasks discussed into clear, cohesive, reproducible workflows^{98,99} for software interactions, for laboratory protocols, and for management of tasks and personnel. Automating workflows also provides opportunities to implement best practices for cybersecurity, cyberbiosecurity, and biosecurity.

For integrating informational tasks, computational workflow engines enable specification, reproducible execution, and exchange of complex workflows involving multiple software programs and computing environments. Current workflow tools include both general tools, such as the Common Workflow Language (CWL)¹⁰⁰, the Dockstore¹⁰¹ and MyExperiment¹⁰² sharing environments, and the PROV ontology for tracking information provenance¹⁰³ (which is already being applied to link design–build–test–learn cycles in SBOL⁵⁰). There are also a number of bioinformatics-focused engines, including Cromwell¹⁰⁴, Galaxy¹⁰⁵, NextFlow¹⁰⁶, and Toil¹⁰⁷. These can be readily adopted for gigabase engineering through steps such as including CWL files in COMBINE archives¹⁰⁸, developing REST or other programmatic interfaces for databases used in genome engineering, containerization¹⁰⁹ of genome engineering computational tools, and depositing these containers to a registry such as DockerHub⁹⁴. Other enhancements likely to be useful include the development of graphical workflow tools for genome engineering, an ontology for annotating the semantic meaning of workflow tasks, and the application of issue tracking systems, such as GitHub issues¹¹⁰ or Jira¹¹¹, to help coordinate teams on the complex tasks involved in designing genomes that require human intervention.

For experimental protocols, a number of technologies have already been developed to automate and integrate experimental workflows as well. Laboratory automation systems can greatly improve both reproducibility and efficiency¹¹² and can also be integrated with LIMS¹¹³ to help track workflows and reagent stocks. A number of automation languages and systems have been developed, including Aquarium¹¹⁴, Antha¹¹⁵, and Autoprotocol¹¹⁶. Although these have not been widely adopted, they have been successfully applied to genetic engineering (e.g., ref. ¹¹⁷), and gigascale genome engineering would benefit from standardization and integration of such systems for application to build and test protocols.

Once links are established across different portions of a workflow, unified access to information in databases for various institutions and stages of the workflow can be accomplished using

standard federation methods and any of the various mature open tools for database management systems (DBMS). Scalable sharing would be further enhanced by adoption of the FAIR (findable, accessible, interoperable, reproducible) data management principles¹¹⁸, which puts specific emphasis on automation friendliness of data sharing. Repositories that support these principles and are applicable to genome engineering include FAIR-DOMHub¹¹⁹, Experimental Data Depot (EDD)¹²⁰, and SynBioHub¹²¹.

Contracts, intellectual property, and laws

Large-scale genome engineering also poses novel challenges in coordinating legal and contractual interactions. When using digital information, both humans and machines need to know the accompanying copyright and licensing obligations. Systematic licensing regimes have been developed for software by the Open Source Initiative (OSI) and other software organizations¹²² and for media and other content with the Creative Commons (CC) family of licenses¹²³, both of which readily allow either a user or a machine to determine if a digital object can be reused, if its reuse is prohibited, or if more complicated negotiation or determination is required. Such systems can be applied to much of the digital information in genome engineering. Care will need to be taken, however, regarding sensitive personal information and European Union database protection rights, which these do not address.

Transfer of physical biological materials was first standardized in 1995 with NIH's Uniform Biological Materials Transfer Agreement (UBMTA), which is used extensively by organizations such as Addgene. Broader and more compatible systems have been developed in the form of the Science Commons project¹²⁴ and the OpenMTA¹²⁵. There are still significant open problems regarding compliance with local regulatory and legal systems, however, particularly when materials cross international borders. Moreover, material transfer agreements generally do not address the intellectual property for materials, which is typically governed through patent law. No publicly available system yet supports automation for patent licensing. Development of automation-friendly intellectual property management might be supported by defining tiered levels that are simultaneously intelligible for the common user, legal experts, and computer systems—though establishing which material or usages can be classified into which tiers may be a difficult process of legal interpretation. Effective use in automation-assisted workflows will also require recording information about which inputs are involved in the production of results, using mechanisms such as the PROV ontology¹⁰³.

Finally, organizations will also need to manage the level of exposure of information, whether due to issues of privacy, safety, publication priority, or other similar concerns. Again, no current system exists, but a basis for developing one may be found in the cross-domain information sharing protocols that have been developed in other domains^{126,127}.

Recommendations and outlook

In summary, scaling up to gigabase genomes presents a wide range of challenges (Table 2). We observe that these challenges cluster into four general themes, each with a different set of needs and paths for development.

The first theme is representing and exchanging designs, plans, data, metadata, and knowledge. Managing information for gigabase genome design requires addressing many challenges regarding scale, representation, and standards. Relatively mature technologies exist to address most individual needs, as well as to assist with the integration of workflows. The practical implementation of effective workflows will require significant investment in building

infrastructure and tools that adopt these technologies, including domain-specific extensions and refinements.

The second theme is sharing and integrating data quality and experimental measurements. Sharing and integrating information arising from measurements of biological material poses significant challenges. It remains unclear what information would be advantageous to share, given the difficulty of obtaining and interpreting measurements of biological systems and the expense and unfavorable scaling of data curation. However, effective integration depends on associating reproducible measurement data with well-curated knowledge and metadata in compatible representations. A number of potential solutions exist for each of these, but significant investment will be needed to investigate how the state of the art can be extended to address these needs.

The third theme is integration of modeling and design at the gigabase scale. Considerable challenges surround efforts to develop a deeper understanding of the relationship between genotype and phenotype, regarding both the interpretation of experimental data and the application of that data to create and validate models, which may be applied in computer-assisted design. Long-term investment in fundamental research is needed, and the suite of biological systems of varying complexity, from cell-free systems to minimal and synthetic cells to natural living systems, may offer suitable experimental platforms for learning the relationship between genotype and phenotype.

Finally, the fourth theme is technical support for Ethical, Legal, and Societal Implications (ELSI) and Intellectual Property (IP) at scale. At the gigabase scale, computer-assisted workflows will be necessary to manage contracts, intellectual property, materials transfers, and other legal and societal interactions. Such workflows will need to be developed by interdisciplinary teams involving experts in law, ELSI issues, software engineering, and knowledge representation. Moreover, it will be critical to address these issues early, to minimize the potential for problematic entanglements associated with the reuse of resources.

In short, engineering gigabase-scale genomes presents significant challenges that will require coordinated investment to overcome. Because many other areas of bioscience face similar challenges, solutions to these challenges will likely also benefit the broader bioscience community. Importantly, the challenges of scale, integration, and lack of knowledge faced in genome engineering are not fundamentally different in nature than those that have been overcome previously in other engineering ventures, such as aerospace engineering and microchip design, which required organizing humans and sharing information across many institutions over time. Thus, we expect to be able to adapt solutions from these other fields for genome engineering.

Investment in capabilities for genome engineering workflows is critical to move from a world in which genome engineering is a heroic effort to one in which genome engineering is routine, safe, and reliable. Investment in workflows for genome engineering will support and enable a vast number of projects, including many not yet conceived, as was the case for reading the human genome. As workflow technologies improve, we anticipate that the trends of expanding team size will eventually reverse, enabling high-fidelity whole-genome engineering at a modest cost and supporting a wide range of medical and industrial applications.

Received: 24 August 2019; Accepted: 18 December 2019;

References

- Boeke, J. D. et al. The genome project-write. *Science* **353**, 126–127 (2016).
- Carlson, R. H. *Biology Is Technology: The Promise, Peril, and New Business of Engineering Life* (Harvard University Press, 2011).
- Hughes, R. A. & Ellington, A. D. Synthetic DNA synthesis and assembly: putting the synthetic in synthetic biology. *Cold Spring Harb. Perspect. Biol.* **9**, a023812 (2017).
- Chari, R. & Church, G. M. Beyond editing to writing large genomes. *Nat. Rev. Genet.* **18**, 749 (2017).
- Khorana, H. G. Total synthesis of a gene. *Science* **203**, 614–625 (1979).
- Mandecki, W., Hayden, M. A., Shallcross, M. A. & Stotland, E. A totally synthetic plasmid for general cloning, gene expression and mutagenesis in *Escherichia coli*. *Gene* **94**, 103–107 (1990).
- Stemmer, W. P. C., Cramer, A., Ha, K. D., Brennan, T. M. & Heyneker, H. L. Single-step assembly of a gene and entire plasmid from large numbers of oligodeoxyribonucleotides. *Gene* **164**, 49–53 (1995).
- Cello, J., Paul, A. V. & Wimmer, E. Chemical synthesis of poliovirus cDNA: generation of infectious virus in the absence of natural template. *Science* **297**, 1016–1018 (2002).
- Tian, J. et al. Accurate multiplex gene synthesis from programmable DNA microchips. *Nature* **432**, 1050 (2004).
- Kodumal, S. J. et al. Total synthesis of long DNA sequences: synthesis of a contiguous 32-kb polyketide synthase gene cluster. *Proc. Natl Acad. Sci. USA* **101**, 15573–15578 (2004).
- Gibson, D. G. et al. Complete chemical synthesis, assembly, and cloning of a *Mycoplasma genitalium* genome. *Science* **319**, 1215–1220 (2008).
- Gibson, D. G. et al. Creation of a bacterial cell controlled by a chemically synthesized genome. *Science* **329**, 52–56 (2010).
- Ostrov, N. et al. Design, synthesis, and testing toward a 57-codon genome. *Science* **353**, 819–822 (2016).
- Fredens, J. et al. Total synthesis of *Escherichia coli* with a recoded genome. *Nature* **569**, 514 (2019).
- Lau, Y. H. et al. Large-scale recoding of a bacterial genome by iterative recombineering of synthetic DNA. *Nucleic Acids Res.* **45**, 6971–6980 (2017).
- Dymond, J. S. et al. Synthetic chromosome arms function in yeast and generate phenotypic diversity by design. *Nature* **477**, 471 (2011).
- Annaluru, N. et al. Total synthesis of a functional designer eukaryotic chromosome. *Science* **344**, 55–58 (2014).
- Richardson, S. M. et al. Design of a synthetic yeast genome. *Science* **355**, 1040–1044 (2017).
- The GP write Leadership Group. Ultra-safe cells resistant to natural viruses announced as first gp-write grand-scale project <https://www.engineeringbiologycenter.org/press/may2018.pdf> (2018).
- Ellis, T., Adie, T. & Baldwin, G. S. DNA assembly for synthetic biology: from parts to pathways and beyond. *Integr. Biol.* **3**, 109–118 (2011).
- Esvelt, K. M. & Wang, H. H. Genome-scale engineering for systems and synthetic biology. *Mol. Syst. Biol.* **9**, 641 (2013).
- Kosuri, S. & Church, G. M. Large-scale de novo DNA synthesis: technologies and applications. *Nat. Methods* **11**, 499 (2014).
- Ostrov, N. et al. Technological challenges and opportunities for writing genomes. *Science* **366**, 310–312 (2019).
- Goldberg, A. P. et al. Emerging whole-cell modeling principles and methods. *Curr. Opin. Biotechnol.* **51**, 97–102 (2018).
- Szigeti, B. et al. A blueprint for human whole-cell modeling. *Curr. Opin. Syst. Biol.* **7**, 8–15 (2018).
- Nielsen, A. A. K. et al. Genetic circuit design automation. *Science* **352**, aac7341 (2016).
- Hutchison, C. A. et al. Design and synthesis of a minimal bacterial genome. *Science* **351**, aad6253 (2016).
- Carbonell, P. et al. An automated design-build-test-learn pipeline for enhanced microbial production of fine chemicals. *Commun. Biol.* **1**, 66 (2018).
- Appleton, E., Densmore, D., Madsen, C. & Roehner, N. Needs and opportunities in bio-design automation: four areas for focus. *Curr. Opin. Chem. Biol.* **40**, 111–118 (2017).
- Gill, R. T., Halweg-Edwards, A. L., Clauset, A. & Way, S. F. Synthesis aided design: the biological design-build-test engineering paradigm? *Biotechnol. Bioeng.* **113**, 7–10 (2016).
- Poust, S., Hagen, A., Katz, L. & Keasling, J. D. Narrowing the gap between the promise and reality of polyketide synthases as a synthetic biology platform. *Curr. Opin. Biotechnol.* **30**, 32–39 (2014).
- Pouvreau, B., Vanhercke, T. & Singh, S. From plant metabolic engineering to plant synthetic biology: the evolution of the design/build/test/learn cycle. *Plant Science* **273**, 3–12 (2018).
- Purnick, P. E. M. & Weiss, R. The second wave of synthetic biology: from modules to systems. *Nat. Rev. Mol. Cell Biol.* **10**, 410 (2009).
- Kim, H. U., Kim, T. Y. & Lee, S. Y. Metabolic flux analysis and metabolic engineering of microorganisms. *Mol. Biosyst.* **4**, 113–120 (2008).
- Woolston, B. M., Edgar, S. & Stephanopoulos, G. Metabolic engineering: past and future. *Annu. Rev. Chem. Biomol. Eng.* **4**, 259–288 (2013).

36. King, Z. A., Lloyd, C. J., Feist, A. M. & Palsson, B. O. Next-generation genome-scale models for metabolic engineering. *Curr. Opin. Biotechnol.* **35**, 23–29 (2015).
37. Steen, E. J. et al. Metabolic engineering of *Saccharomyces cerevisiae* for the production of n-butanol. *Microbial. Cell Factories* **7**, 36 (2008).
38. Cochrane, G., Karsch-Mizrachi, I. & Takagi, T., International Nucleotide Sequence Database Collaboration. The international nucleotide sequence database collaboration. *Nucleic Acids Res.* **44**, D48–D50 (2015).
39. Cunningham, F. et al. Ensembl 2019. *Nucleic Acids Res.* **47**, D745–D751 (2018).
40. Kersey, P. J. et al. Ensembl genomes 2016: more genomes, more complexity. *Nucleic Acids Res.* **44**, D574–D580 (2015).
41. O’Leary, N. A. et al. Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation. *Nucleic Acids Res.* **44**, D733–D745 (2015).
42. Haft, D. H. et al. RefSeq: an update on prokaryotic genome annotation and curation. *Nucleic Acids Res.* **46**, D851–D860 (2017).
43. Mashima, J. et al. Dna data bank of japan (DDBJ) progress report. *Nucleic Acids Res.* **44**, D51–D57 (2015).
44. Frankish, A. et al. Comparison of gencode and refseq gene annotation and the impact of reference geneset on variant effect prediction. *BMC Genomics* **16**, S2 (2015).
45. McCarthy, D. J. et al. Choice of transcripts and software has a large effect on variant annotation. *Genome Med.* **6**, 26 (2014).
46. National Center for Biotechnology Information. Genome data viewer, <https://www.ncbi.nlm.nih.gov/genome/gdv/> (2019).
47. Liao, Y., Wang, J., Jaehng, E. J., Shi, Z. & Zhang, B. WebGestalt 2019: gene set analysis toolkit with revamped UIs and APIs. *Nucleic Acids Res.* **47**, W199–W205 (2019).
48. Huang, D. W., Sherman, B. T. & Lempicki, R. A. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nat. Protoc.* **4**, 44 (2009).
49. Chibucos, M. C., Siegel, D. A., Hu, J. C. & Giglio, M. The evidence and conclusion ontology (ECO): supporting GO annotations. In (Dessimoz C. & Škunca N. eds.) *The Gene Ontology Handbook* 245–259 (Humana Press, New York, 2017).
50. Cox, R. S. et al. Synthetic biology open language (SBOL) version 2.2. 0. *J. Integr. Bioinform.* **15**, 20180001 (2018).
51. Eilbeck, K. et al. The sequence ontology: a tool for the unification of genome annotations. *Genome Biol.* **6**, R44 (2005).
52. SBOL Community. Synbiohub 1.4, <https://github.com/SynBioHub> (2019).
53. SBOL Community. libsbolj 2.5-prerelease, <https://github.com/SynBioDex/libSBOLj> (2019).
54. Nguyen, T., Roehner, N., Zundel, Z. & Myers, C. J. A converter from the systems biology markup language to the synthetic biology open language. *ACS Synth. Biol.* **5**, 479–486 (2016).
55. Roehner, N., Zhang, Z., Nguyen, T. & Myers, C. J. Generating systems biology markup language models from the synthetic biology open language. *ACS Synth. Biol.* **4**, 873–879 (2015).
56. Lang, P. F., Chebaro, Y. & Karr, J. R. BpForms: a toolkit for concretely describing modified DNA, RNA and proteins. Preprint at arXiv preprint arXiv:1903.10042, (2019).
57. Venkatachalam, A. R., Mellichamp, J. M. & Miller, D. M. A knowledge-based approach to design for manufacturability. *J. Intell. Manuf.* **4**, 355–366 (1993).
58. Abrantes, R. et al. Rule ontology for automatic design verification application to PCB manufacturing and assembly. In *IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society* 3403–3409 (IEEE, 2017).
59. Oberortner, E., Cheng, J. F., Hillson, N. J. & Deutsch, S. Streamlining the design-to-build transition with build-optimization software tools. *ACS Synth. Biol.* **6**, 485–496 (2016).
60. Swainston, N. et al. Recon 2.2: from reconstruction to model of human metabolism. *Metabolomics* **12**, 109 (2016).
61. Monk, J. M. et al. iml1515, a knowledgebase that computes *Escherichia coli* traits. *Nat. Biotechnol.* **35**, 904 (2017).
62. Karr, J. R. et al. A whole-cell computational model predicts phenotype from genotype. *Cell* **150**, 389–401 (2012).
63. Karp, P. D. et al. The biocy collection of microbial genomes and metabolic pathways. *Brief. Bioinform.* **20**, 1085–1093 (2017).
64. Overbeek, R. et al. The SEED and the Rapid Annotation of microbial genomes using subsystems technology (RAST). *Nucleic Acids Res.* **42**, D206–D214 (2013).
65. Engler, C., Gruetznern, R., Kandzia, R. & Marillonnet, S. Golden gate shuffling: a one-pot dna shuffling method based on type IIs restriction enzymes. *PLoS ONE* **4**, e5553 (2009).
66. Weber, E., Engler, C., Gruetznern, R., Werner, S. & Marillonnet, S. A modular cloning system for standardized assembly of multigene constructs. *PLoS ONE* **6**, e16765 (2011).
67. Gibson, D. G. et al. Enzymatic assembly of dna molecules up to several hundred kilobases. *Nat. Methods* **6**, 343 (2009).
68. Cock, P. J. A., Fields, C. J., Goto, N., Heuer, M. L. & Rice, P. M. The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic Acids Res.* **38**, 1767–1771 (2009).
69. Reese, M. G. et al. A standard variation file format for human genome sequences. *Genome Biol.* **11**, R88 (2010).
70. Roehner, N. et al. Sharing structure and function in biological design with SBOL 2.0. *ACS Synth. Biol.* **5**, 498–506 (2016).
71. Wang, L. & Hoffman, R. A. Standardization, calibration, and control in flow cytometry. *Curr. Protoc. Cytom.* **79**, 1–3 (2017).
72. Beal, J. et al. Quantification of bacterial fluorescence using independent calibrants. *PLoS One* **13**, e0199432 (2018).
73. Stevenson, K., McVey, A. F., Clark, I. B. N., Swain, P. S. & Pilizota, T. General calibration of microbial growth in microplate readers. *Sci. Rep.* **6**, 38828 (2016).
74. Lee, H., Pine, P. S., McDaniel, J., Salit, M. & Oliver, B. External RNA controls consortium beta version update. *J. Genomics* **4**, 19 (2016).
75. Shimoyama, Mary et al. Three ontologies to define phenotype measurement data. *Front. Genet.* **3**, 87 (2012).
76. Malone, J. et al. Modeling sample variables with an experimental factor ontology. *Bioinformatics* **26**, 1112–1118 (2010).
77. Wolstencroft, K. et al. RightField: embedding ontology annotation in spreadsheets. *Bioinformatics* **27**, 2021–2022 (2011).
78. Sarma, G. P. et al. OpenWorm: overview and recent advances in integrative biological simulation of *Caenorhabditis elegans*. *Philos. Trans. Royal Soc. B* **373**, 20170382 (2018).
79. Smith, B. et al. The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nat. Biotechnol.* **25**, 1251 (2007).
80. Courtot, M. et al. Controlled vocabularies and semantics in systems biology. *Mol. Syst. Biol.* **7**, 543 (2011).
81. Chibucos, M. C. et al. An ontology for microbial phenotypes. *BMC Microbiol.* **14**, 294 (2014).
82. Köhler, S. et al. The human phenotype ontology in 2017. *Nucleic Acids Res.* **45**, D865–D876 (2016).
83. Yu, N. Y. et al. PSORTb 3.0: improved protein subcellular localization prediction with refined localization subcategories and predictive capabilities for all prokaryotes. *Bioinformatics* **26**, 1608–1615 (2010).
84. Villalobos, A., Ness, J. E., Gustafsson, C., Minshull, J. & Govindarajan, S. Gene designer: a synthetic biology tool for constructing artificial dna segments. *BMC Bioinformatics* **7**, 285 (2006).
85. Hucka, M. et al. The Systems Biology Markup Language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* **19**, 524–531 (2003).
86. Cuellar, A. et al. The CellML 1.1 specification. *J. Integr. Bioinform.* **12**, 4–85 (2015).
87. Gleeson, P., Crook, S., Silver, A. & Cannon, R. Development of NeuroML version 2.0: greater extensibility, support for abstract neuronal models and interaction with systems biology languages. *BMC Neurosci.* **12**, P29 (2011).
88. Hucka, M. et al. Promoting coordinated development of community-based information standards for modeling in biology: the COMBINE initiative. *Front. Bioeng. Biotechnol.* **3**, 19 (2015).
89. Glont, M. et al. BioModels: expanding horizons to include more modelling approaches and formats. *Nucleic Acids Res.* **46**, D1248–D1253 (2017).
90. Crook, S. M. & Dietrich, S. Model exchange with the NeuroML model database. *BMC Neurosci.* **15**, P171 (2014).
91. Gleeson, P. et al. Open source brain: a collaborative resource for visualizing, analyzing, simulating, and developing standardized models of neurons and circuits. *Neuron* **103**, 395–411 (2019).
92. Yu, T. et al. The physiome model repository 2. *Bioinformatics* **27**, 743–744 (2011).
93. Avsec, Ž. et al. The kipoi repository accelerates community exchange and reuse of predictive models for genomics. *Nat. Biotechnol.* **37**, 592–600 (2019).
94. Docker Inc. DockerHub, <https://hub.docker.com/> (2019).
95. Cooling, M. T. et al. Standard virtual biological parts: a repository of modular modeling components for synthetic biology. *Bioinformatics* **26**, 925–931 (2010).
96. Cowan, A. E., Mendes, P. & Blinov, M. L. Modelbricks: modules for reproducible modeling improving model annotation and provenance. *NPJ Syst. Biol. Appl.* **5**, 1–6 (2019).
97. Kirk, P., Thorne, T. & Stumpf, M. P. H. Model selection in systems and synthetic biology. *Curr. Opin. Biotechnol.* **24**, 767–774 (2013).
98. Myers, C. J. et al. A standard-enabled workflow for synthetic biology. *Biochem. Soc. Trans.* **45**, 793–803 (2017).
99. Moreno, A. G. et al. An implementation-focused bio/algorithmic workflow for synthetic biology. *ACS Synth. Biol.* **5**, 1127–1135 (2016).
100. Amstutz, P. et al. Common Workflow Language, v1.0 <https://www.commonwl.org/> (2016).
101. O’Connor, B. D. et al. The Dockstore: enabling modular, community-focused sharing of Docker-based genomics tools and workflows. *F1000Research* **6**, 52 (2017).

102. Goble, C. A. et al. myExperiment: a repository and social network for the sharing of bioinformatics workflows. *Nucleic Acids Res.* **38**, W677–W682 (2010).
103. Missier, P., Belhajjame, K. & Cheney, J. The W3C PROV family of specifications for modelling provenance metadata. In *Proc. 16th International Conference on Extending Database Technology 773–776* (ACM, 2013).
104. Broad Institute. The Workflow Description Language and cromwell, <https://github.com/broadinstitute/cromwell> (2019).
105. Goecks, J., Nekrutenko, A. & Taylor, J. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biol.* **11**, R86 (2010).
106. Tommaso, P. D. et al. Nextflow enables reproducible computational workflows. *Nat. Biotechnol.* **35**, 316 (2017).
107. Vivian, J. et al. Toil enables reproducible, open source, big biomedical data analyses. *Nat. Biotechnol.* **35**, 314 (2017).
108. Bergmann, F. T. et al. COMBINE archive and OMEX format: one file to share all information to reproduce a modeling project. *BMC Bioinform.* **15**, 369 (2014).
109. Soltész, S., Pötzl, H., Fiuczynski, M. E., Bavier, A. & Peterson, L. Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors. in *ACM SIGOPS Operating Systems Review*, Vol. 41, 275–287 (ACM, 2007).
110. GitHub, Inc. GitHub guides: Mastering issues, <https://guides.github.com/features/issues/> (2019).
111. Fisher, J., Koning, D. & Ludwigsen, A. P. Utilizing Atlassian Jira for large-scale software development management. Technical report (Lawrence Livermore National Lab. (LLNL), Livermore, CA, 2013).
112. Sadowski, M. I., Grant, C. & Fell, T. S. Harnessing QbD, programming languages, and automation for reproducible biology. *Trends Biotechnol.* **34**, 214–227 (2016).
113. Prasad, P. J. & Bodhe, G. L. Trends in laboratory information management system. *Chemometr. Intell. Lab. Syst.* **118**, 187–192 (2012).
114. Keller, B., Vrana, J., Miller, A., Newman, G. & Klavins, E. Aquarium: the laboratory operating system version 2.6.0. <https://github.com/klavinslab/aquarium> (2019).
115. Synthace. Antha. <https://github.com/antha-lang>
116. Miles, B. & Lee, P. L. Achieving reproducibility and closed-loop automation in biological experimentation with an IoT-enabled lab of the future. *SLAS Technol.* **23**, 432–439 (2018).
117. Yang, Y., Nemhauser, J. L. & Klavins, E. Synthetic bistability and differentiation in yeast. *ACS Synth. Biol.* **8**, 929–936 (2019).
118. Wilkinson, M. D. et al. The FAIR guiding principles for scientific data management and stewardship. *Sci. Data* **3**, 160018 (2016).
119. Wolstencroft, K. et al. FAIRDOMHub: a repository and collaboration environment for sharing systems biology research. *Nucleic Acids Res.* **45**, D404–D407 (2016).
120. Morrell, W. C. et al. The experiment data depot: a web-based software tool for biological experimental data storage, sharing, and visualization. *ACS Synth. Biol.* **6**, 2248–2259 (2017).
121. McLaughlin, J. A. et al. SynBioHub: a standards-enabled design repository for synthetic biology. *ACS Synth. Biol.* **7**, 682–688 (2018).
122. Opensource.org. Open Source Initiative. <https://opensource.org/> (2019).
123. Lessig, L. The creative commons. *Fla. L. Rev.* **55**, 763 (2003).
124. Nguyen, T. Science commons: material transfer agreement project. *Innov. Technol. Gov. Glob.* **2**, 137–143 (2007).
125. Kahl, L. et al. Opening options for material transfer. *Nat. Biotechnol.* **36**, 923 (2018).
126. Chandrasekaran, C., Simpson, W. & Trice, A. Cross-domain solutions in an era of information sharing. in *The 1st International Multi-conference on Engineering and Technological Innovation: IMET2008, Orlando, FL*, Vol. 1, 313–318 (2008).
127. Sun, J. & Fang, Y. Cross-domain data sharing in distributed electronic health record systems. *IEEE Trans. Parallel. Distrib. Syst.* **21**, 754–764 (2009).

Acknowledgements

This work was supported, in part, by NIH awards P41-EB023912 and R35-GM119771 and by NSF awards 1548123 and 1522074. We thank the reviewers for legal insights for Section “Contracts, intellectual property, and laws”, and thank Nicola Hawes for help illustrating Fig. 2. The views, opinions, and/or findings expressed are those of the author (s) and should not be interpreted as representing the official views or policies of these funding agencies or the U.S. Government. This document does not contain technology or technical data controlled under either U.S. International Traffic in Arms Regulation or U.S. Export Administration Regulations. Certain commercial equipment, instruments, or materials are identified to adequately specify experimental procedures. Such identification neither implies recommendation nor endorsement by the National Institute of Standards and Technology nor that the equipment, instruments, or materials identified are necessarily the best for the purpose.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to J.B.

Peer review information *Nature Communications* thanks David Grewal, Tom Ellis and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

Reprints and permission information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2020