PORTLAND PRESS

Review Article

Genome-driven cell engineering review: *in vivo* and *in silico* metabolic and genome engineering

Sophie Landon[1,2,*], Joshua Rees-Garbutt[1,3,*], Lucia Marucci[1,2,4,†] and (iD) Claire Grierson[1,3,†]

[1]BrisSynBio, University of Bristol, Bristol BS8 1TQ, U.K.; [2]Department of Engineering Mathematics, University of Bristol, Bristol BS8 1UB, U.K.; [3]School of Biological Sciences, University of Bristol, Life Sciences Building, Bristol BS8 1TQ, U.K.; [4]School of Cellular and Molecular Medicine, University of Bristol, Bristol BS8 1UB, U.K.

Correspondence: Claire Grierson (claire.grierson@bristol.ac.uk) or Lucia Marucci (lucia.marucci@bristol.ac.uk)

Producing 'designer cells' with specific functions is potentially feasible in the near future. Recent developments, including whole-cell models, genome design algorithms and gene editing tools, have advanced the possibility of combining biological research and mathematical modelling to further understand and better design cellular processes. In this review, we will explore computational and experimental approaches used for metabolic and genome design. We will highlight the relevance of modelling in this process, and challenges associated with the generation of quantitative predictions about cell behaviour as a whole: although many cellular processes are well understood at the subsystem level, it has proved a hugely complex task to integrate separate components together to model and study an entire cell. We explore these developments, highlighting where computational design algorithms compensate for missing cellular information and underlining where computational models can complement and reduce lab experimentation. We will examine issues and illuminate the next steps for genome engineering.

Introduction

Synthetic biology is the rational design and engineering of cells and cellular systems using genetic manipulations [1,2]. It is divided into three fields [3]: DNA-based device construction (production of functioning biological components to be inserted into cells), synthetic protocell development (construction of rudimentary representations of living cells), and genome-driven cell engineering. For more about DNA-based device construction principles see Brophy and Voigt [4], and for an introduction to protocell development see Dzieciol and Mann [5]. In this review, we will focus on genome-driven cell engineering (see Box 1 for key terms).

Genome-driven cell engineering encompasses both metabolic engineering (control of cellular production processes) and genome engineering (production of minimal genomes, recoded genomes, and cellular chassis/factories). It encompasses diverse types and scales of genetic modifications and underscores the genome as the major driver of cellular events [3].

Metabolic engineering attempts to improve titre, accumulation rate, and yield of a specific metabolite, often from microorganisms in an industrial setting [6]. Genome engineering attempts to: understand (comprehending biological systems by trying to engineer them [7], e.g. minimal genomes), reduce risks (restricting bacteria to specific media [8], e.g. recoded genomes), and improve metabolite production (e.g. 'optimal' chassis cell development for metabolite production [9]).

Here, we review metabolic engineering and genome engineering from both biological and computational perspectives. Metabolic engineering, with established *in silico* design and simulation (hundreds of models, tens of algorithms) [10,11] and *in vivo* construction methodologies (hundreds of strains of several bacterial species) [12], could inform the future of genome engineering, given the development of whole-cell mathematical models [13], genome design algorithms [14,15], and CRISPR-cas9 gene editing techniques [16–18]. Finally, we examine issues and the next steps for genome engineering.

*Co-first authors.
†Co-last authors.

> **Box 1 Key terms**
>
> **Genome engineering:** Extensive and intentional genetic modification of a replicating system for a specific purpose [19].
>
> **Minimal genomes:** Reduced genomes containing only the genetic material essential for survival, with an appropriately rich medium and no external stresses. No single gene can be removed without loss of viability [20].
>
> **Recoded genomes**: Genomes with codon/s that have been freed, substituting codons for synonymous codons that encode the same amino acid, so that they can be assigned to new functions [21,22].
>
> **Platform cell/cell factory/chassis (interchangeable):** A bacterial species that can efficiently convert raw materials into a product of interest, through genome engineering or hosting genetic components [6,23–26].
>
> **Multiplex gene editing:** Simultaneous introduction of multiple distinct modifications to a genome [27].
>
> **Algorithm:** Series of steps or rules to attempt to solve a problem, often implemented in a computer.
>
> **Model:** Mathematical description of a system.
>
> **Metabolic flux:** Metabolic reaction rate (i.e. turnover of molecules through a metabolic reaction).
>
> **Flux vector:** A vector where each element corresponds to the metabolic flux of a reaction in the model.
>
> **Genome-scale biological models:** Category of models containing: metabolic models, transcription regulatory networks, protein–protein interaction networks, integrated cellular models, and whole-cell models [11].
>
> **Genome-scale metabolic models:** Models representing all active reactions in a cell/organism as a matrix of stoichiometric coefficients of each reaction, and linking reactions with gene products that catalyse them. Abbreviated to GSMMs or GEMs [28,29].
>
> **Whole-cell models:** Describe the life cycle of a single cell, modelling individual molecules and interactions, and includes the function of every known gene product [13].

Metabolic engineering *in vivo*

Metabolic engineering enhances the production of native or introduced metabolites, often in a microbial strain [6]. Genetic edits are used to introduce or modify the required pathway and take control of core metabolism, cellular regulation, and stress responses [6,12]. Applications are wide ranging, including fuels, feed additives, and pharmaceuticals [12,30], and determine the most appropriate microorganism for production (see Table 1).

Only a small number of microorganisms are 'industry ready', such as: *Escherichia coli (E. coli)*, *Bacillus subtilis (B. subtilis)*, *Streptomyces sp.*, *Pseudomonas putida*, *Corynebacterium glutamicum* [6], *Saccharomyces cerevisiae* and *Aspergillus niger* [12]. Requirements for industrial microorganisms are simple nutritional needs, fast and efficient growth, high resistance to extreme physical and chemical conditions, and efficient secretion systems [6]. Also required are sufficient genetic and metabolic knowledge and a range of genetic tools (e.g. promoters and terminators with varying expression levels, and well-characterised plasmids for precise manipulations). Due to the development of CRISPR-cas9 gene editing tools [31,32], a number of novel bacterial species are now usable, including *Vibrio natriegens* (has the shortest known doubling time, at 15 min), and *Roseobacter* and *Halomonas* (marine species with salt tolerance) [6]. Metabolic engineering has recently been reviewed for *E. coli* [33] and *B. subtilis* [34].

The metabolic production pathway is constructed, reconstructed, or tweaked in the strain, and can then be iterated upon to produce improvements in titre, rate, and yield. There are six strategies [34] for improving these: (i) modular pathway engineering, which divides up the production pathway to produce and combine modules with different expression levels [35]; (ii) cofactor engineering, in which metabolic flux to the desired products is enhanced through gene edits that alter non-protein cofactor levels [36]; (iii) scaffold-guided protein engineering, where the spatial locations of proteins in the cell are modified to increase local concentrations of intermediates [37]; (iv) transporter engineering, which improves the import of substrates [38] and export of products [39]; (v) dynamic pathway analysis, which identifies unknown network interactions and promotes or suppresses them to increase levels of product [40]; and (vi) evolutionary engineering, which mimics natural evolutionary approaches to produce greater amounts of product [41–43].

The development of an 'industry ready' strain takes several years and is costly. Strains for Artemisinin and 1,3-propanediol production took 10 years and $50 million, and 15 years and $130 million, to develop respectively

Table 1 A selection of microorganisms used for metabolite production

Microorganism	Primary feature	Applications	Product examples	Strain examples
Escherichia coli	Variety of tools/knowledge	Exploratory production, established industrial strain	1-3-Propanediol, 1-4-Butanediol, butanol, insulin, limonene, L-threonine, L-serine, PHAs, propane, succinate	Based on K-12 and B ancestor strains. Derivatives of MG1655, W3110, BW25113. Specific strains: BL21 Rosetta, DH1, ATCC 31884, DH10B
Bacillus subtilis	Efficient secretion systems	Protein production	Amylases, bacitracin, biotin, cellulosome, chiral stereoisomers, cobalamin, glucanases, guanosine, laccases monophosphate, riboflavin, subtilisin, vitamin B_6	Protease-defective mutants: WB600, WB800. Specific strains: 168, RH33, BSUL08, 1A1, E8, KU303
Pseudomonas putida	Chemical resistance	Harsh conditions and toxic product production	3-methyl-catechol, anthranilate, cinnamic acid, PHAs, phenol, o-cresol, styrene, terpenoids, vanillate	Specific strains: KT2440, EM42, Gpo1, S12
Cyanobacteria	Photosynthetic	Light-driven production	1-butanol, 1,3-propanediol, bisabolene, ethanol, farnesene, isoprene, isopropanol, PHAs	Specific strains: PCC- 6803, PCC-7942. PCC-7002

Information collated from Nielsen and Keasling [12] Calero and Nikel [6], Gu et al. [34], and Pontrelli et al. [33].

[12], though sales of metabolic products are expected to reach $6.2 billion by 2020 [44]. The time and cost is due to complex interactions and regulation in metabolism. Metabolite intermediates and products can cause toxicity and act as inhibitors of other reactions, or be misrouted or modified by unrelated enzyme reactions, leading to decreasing titre, rate, and yield [6].

Recently, the availability of accurate, genome scale metabolic models, refined with data captured using omics technologies, has begun to overcome these limitations and support rounds of *in silico* design and *in vivo* construction. [6,34].

Metabolic engineering *in silico*
Constraint-based metabolic models

Recent advances have allowed the reconstruction of genome-scale metabolic networks, and subsequently *in silico* models which can predict cellular phenotypes. Metabolic models are based on biological knowledge and experimental data; metabolic kinetic models describe how metabolites vary in time using differential equations, while metabolic constraint-based models formalise behaviour at steady-state (i.e. metabolite production is equal to metabolite loss) [45]. There are currently very few genome-scale kinetic models, due to the lack of experimental data for enzyme rate parameters, but the static nature of constraint-based models means they require significantly fewer parameters to construct, and so are more widely used. For this reason, we will mainly review constraint-based models.

Genome-scale metabolic models (GSMMs/GEMs) aim to form a solution space of flux values for each reaction in a metabolic model (see Orth et al. [46] for more details), or can give insight into the behaviour of the system through network analysis. These models consist of a stoichiometric matrix and a set of biologically feasible constraints for reactions (Figure 1).

There are currently 113 bacteria, 57 eukaryote and 8 archaea curated GEMs available at UCSD Systems Biology [47], see Figure 2. Tools for automatically generating GEMs have been developed, first modelSEED [48] and the most recently CarveMe [49]. CarveMe begins with a universal model consisting of 2383 metabolites and 4383 reactions, formulated from the BiGG database [50,51]. This can be stripped down to become a metabolic model for any specific organism, using its annotated genome. There are multiple other automation tools available: AuReMe [52], Merlin [53], MetaDraft [54], Pathway Tools [55], and Raven [56], which have been recently reviewed [57].

Exploring and analysing the steady-state solution space

There are numerous ways to simulate and analyse metabolic models, depending on the desired information. Elementary flux modes (EFMs) can be found, based on the stoichiometric matrix—these are the set of non-decomposable reactions that trace input metabolites to output metabolites which can be used to break a metabolic network into its component pathways [58]. In the context of metabolic engineering, these can be analysed to choose reactions to disrupt in order to direct cell resources towards specific metabolites. Alternatively, the fluxes through each reaction
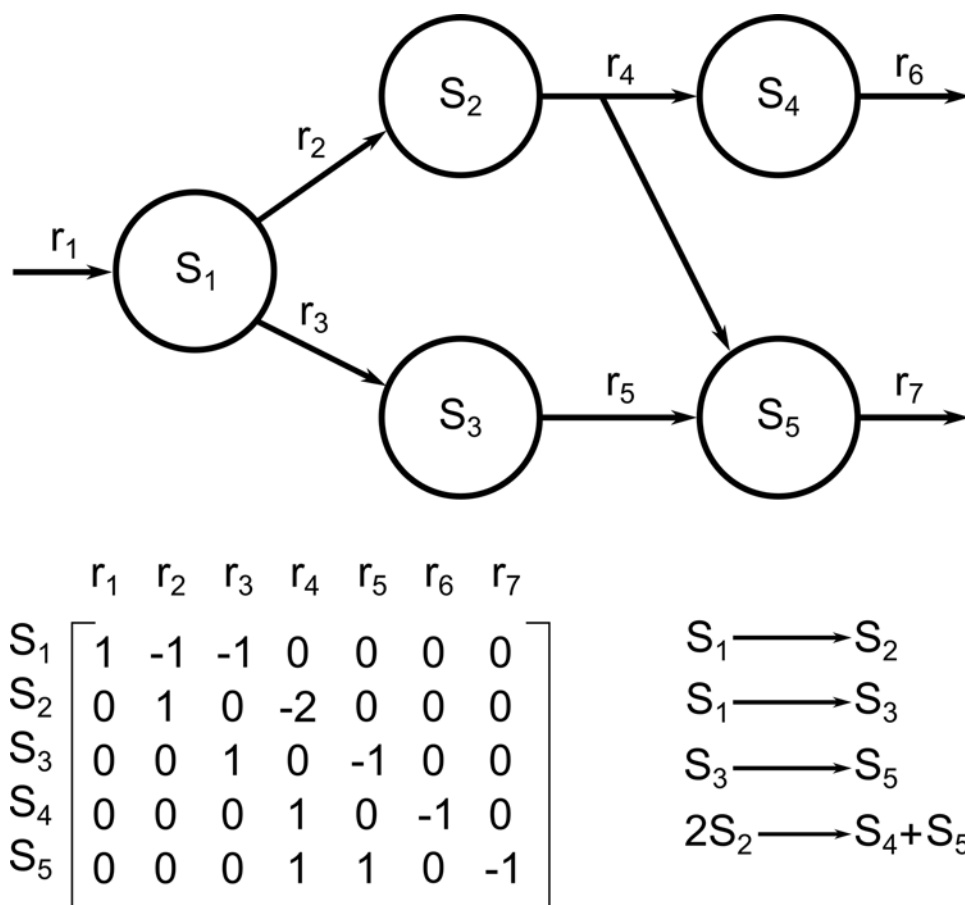
Figure 1. A toy metabolic network

S_i are the substrates and r_i are the reaction rates. The network can be represented as a stoichiometric matrix (whose columns and rows correspond to reactions and metabolites, respectively), and a system of equations.
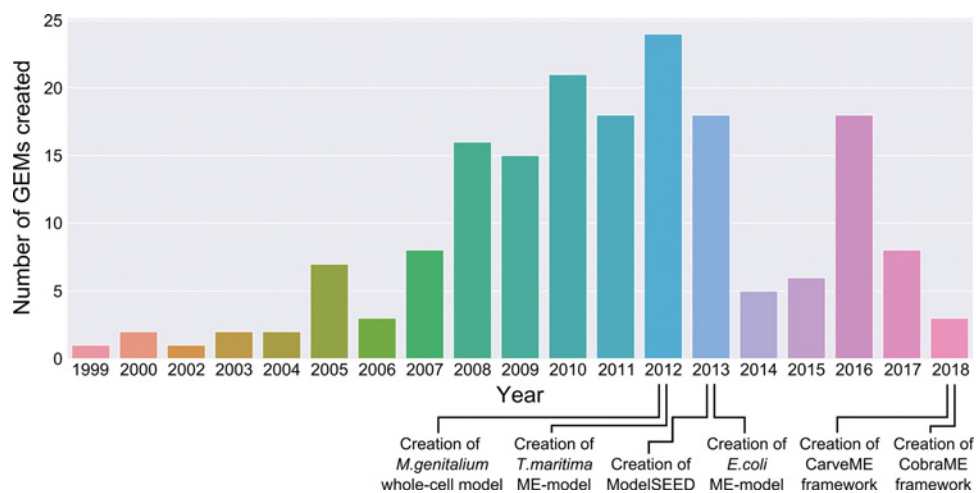

Figure 2. Creation and timeline of bacterial GEMs over the past two decades

More complex genome-scale computational models (such as metabolic and macromolecular expression (ME) models and the first whole-cell model), modelling automation tools (ModelSEED and CarveME) and the ME software frameworks COBRAMe are also included. 2001 did not see any models created.
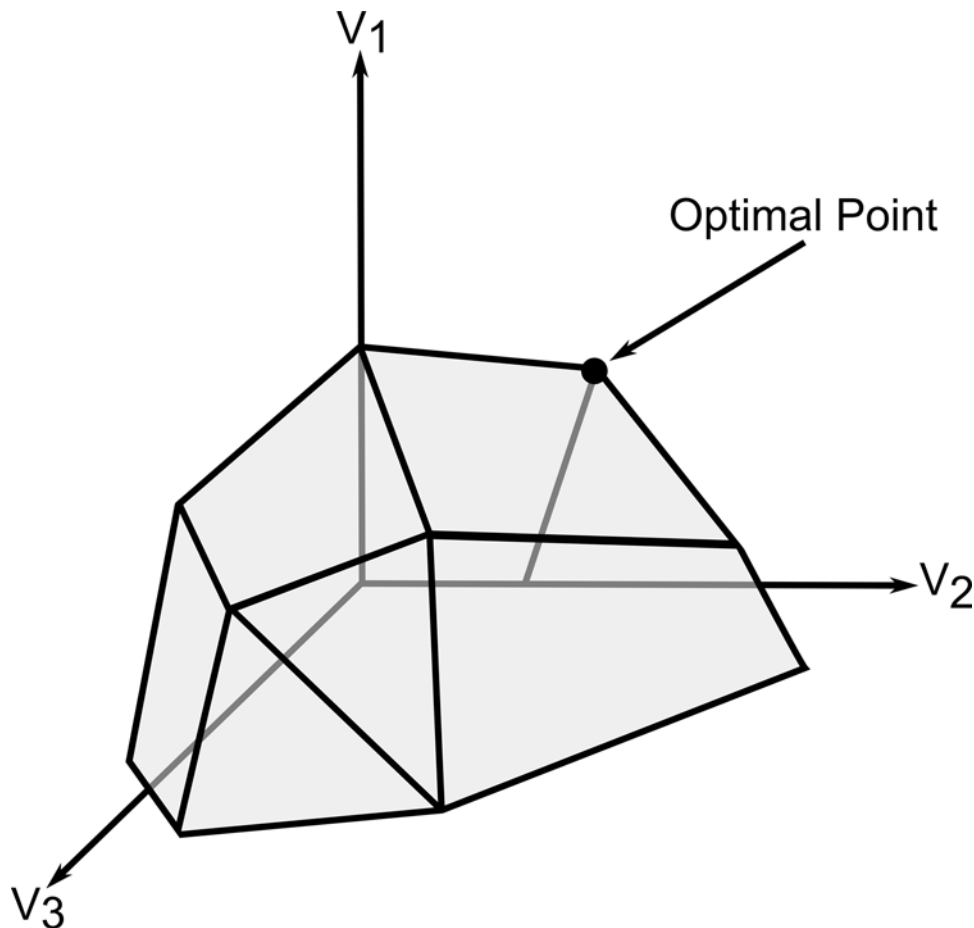
Figure 3. A schematic of the feasible region found through constraint-based modelling
Where v_i are fluxes of the system and form a flux polyhedron. The flux values that optimise the objective function can be found by looking at the extreme edges of the polyhedron, and selecting the point that fits the optimisation criteria.

when the system is at steady state can be found, either through Monte Carlo sampling [59] or, more commonly, by flux balance analysis (FBA). The solution space of the system at steady-state can be found by combining the constraints on the system to form a region which can be analysed, as shown in Figure 3.

To perform FBA, first an objective function is defined that can maximise or minimise the flux through any reaction in the system. When simulating a wild-type unicellular organism in the exponential growth phase, the rate of biomass production is maximised as a proxy for cellular fitness [60], and used as the objective function. This is formulated as a 'pseudoreaction', meant as the sum of the different biomass components (e.g. amino acids, fatty acids, vitamins, and cofactors); the flux through which is maximised. Other objective functions can be used for different purposes: for example, minimisation of ATP production [61], or optimisation of several reactions in parallel.

FBA optimisation generates flux vectors that are used to optimise a defined objective function. The flux vectors give insight into the dynamics of the system when equilibrium is reached, indicate which pathways the metabolites are involved in, and can also predict the behaviour of the simulated cell when grown in different culture conditions.

FBA is available in the COBRA (constraint-based reconstruction and analysis) toolbox for Python or MATLAB numerical computing environments. For an overview of the COBRA ecosystem see Lewis et al. [62].

Alternative methods to optimise the solution space

GEMs can be used for metabolite optimisation by analysing the effects of adding or removing genes. FBA can be used to directly calculate fluxes in cells with gene knockouts, or FBA wild-type fluxes can be used as input for other methods to calculate fluxes after gene knockouts: MOMA (minimisation of metabolic adjustment [63]) and ROOM (regulatory on/off minimisation of metabolic flux [64]). While FBA picks a solution that optimises a given objective

function, MOMA and ROOM output a solution which minimises the distance between the wild-type and the altered metabolism fluxes, or the number of changes in flux respectively. Given that a strain after a knockout is not a result of evolution, the assumption of the FBA objective function mimicking evolution may no longer be relevant and so both MOMA and ROOM account for the cell behaviour immediately after *in vivo* knockouts, which can be different from cell behaviour over a longer time scale [65].

Predicting gene essentiality using GEMs and algorithms

Metabolic models can be used also to predict gene essentiality—gene knockouts can be simulated and then cell survival assessed based on the end production of biomass (i.e., if the simulation results in zero biomass production, the cell is presumed to be dead and therefore the knocked-out gene is essential). This has successfully been shown for *E. coli* strains such as MG1655, where gene deletions simulated *in silico* correctly predicted the essentiality of 86% of single gene deletions [66]. Similar gene essentiality testing has been performed with FBA models of other organisms, including *Helicobacter pylori* [67], *Saccharomyces cerevisiae* [68], *B. subtilis* [69], and *C. glutamicum* [70], showing the accuracy of these models.

Further Development of GEMs

GEMs can also act as a springboard for more detailed cellular models that take into account transcription processes. These extended models, still in early stages of development, have not yet improved the accuracy of standard FBA models [49]. More recently, macromolecular expression (ME) has been incorporated (ME-models) to integrate tRNA charging, transcription, and translation reactions with metabolic reactions. The metabolic reactions are coupled with the macromolecular synthesis reactions of the enzymes that catalyse them, and the synthesis reactions for transcription and translation components (e.g. mRNA and proteins) are formed from the metabolic biomass production. The COBRAMe framework [71] aids generation of ME models, for example an *E. coli* model (iJL1678b-ME) that is more efficient than the first *E. coli* ME model (iOL1650-ME [72]), containing 1/6 variables and solving in 1/36 of the time.

The main limitation to this approach is the lack of well-curated databases: while genome and gene product information can be retrieved (e.g. using KEGG [73] and Genbank [74]), no single database contains rate parameters for transcription and translation, necessary for ME model parameterisation [75].

Metabolic engineering applications: constraint-based modelling and metabolic network analysis

As discussed above, GEMs can be used to study wild-type cell behaviour, as well as investigate the effects of gene knockouts. Another application is for metabolic engineering, where algorithms can use GEMs to predict genes within the model to knockout, amplify or inhibit, in order to produce a pre-defined goal of overproduction of some metabolite.

Metabolic engineering using Elementary Flux Modes (EFMs)

As well as providing scope for analysis of metabolic networks, EFMs can be used to isolate pathways that can be disrupted to force a cell to overproduce a metabolite. As EFMs find minimal pathways through the metabolic network, the paths from an input substrate to a chosen metabolite and its efficiency (i.e. the stoichiometry and length of the chain of reactions) can be found. Competing reaction pathways can then be found and removed, thereby producing a streamlined strain with minimal functionality. Although this process involves significant computational power, especially for genome-scale models, it has been shown to have success in improving lysine production in *C. glutamicum* [76].

Metabolic engineering using nested linear programming-based methods

Whereas FBA, MOMA and ROOM take gene modifications as their input and output a corresponding flux distribution, other algorithms designed specifically for metabolic engineering take a metabolite (other than biomass) as their input, and output a set of gene modifications that optimise its production.

For example, OptKnock maximises the production of a specified metabolite and biomass by deleting genes to re-route metabolites through certain reaction pathways [77]. Genome designs for the overproduction of succinate and lactate *in silico* using OptKnock were consistent with laboratory results [77].

Several algorithms for metabolite optimisation use linear programming and couple cell growth and biochemical production using bilevel mixed-integer linear program (MILP), a nested framework where an outer optimisation problem (e.g. maximise metabolite) is constrained by another inner optimisation problem (e.g. maximise biomass),
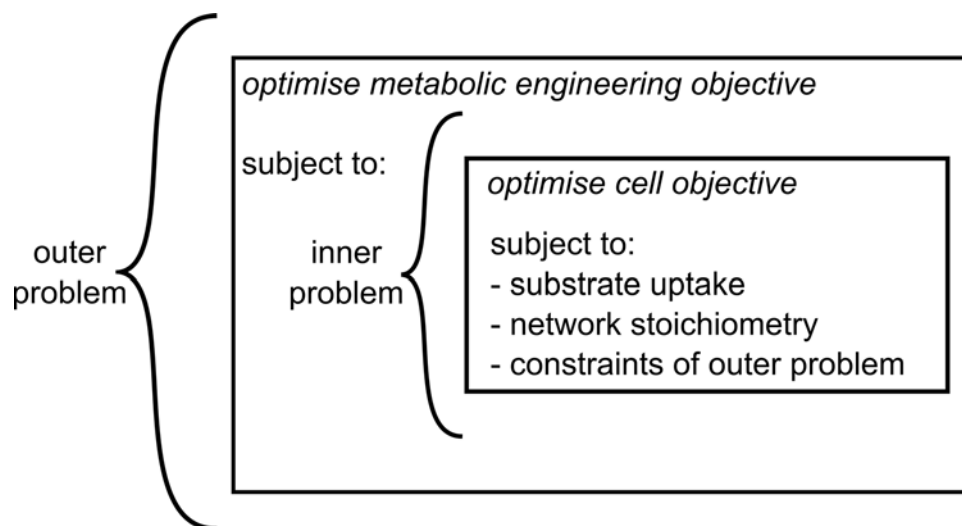
Figure 4. Bilevel linear programming
The nested structure of the bilevel linear programming algorithms, where the inner problem optimises for a cellular objective function and the outer problem optimises for some metabolic engineering objective.

Table 2 Metaheuristic algorithms for analysis of metabolic models and metabolic engineering

OptGene (Patil et al., 2005) [86]	Uses a genetic algorithm (the outer problem) to iteratively run FBA (the inner problem) with different knockout combinations to maximise metabolite production
RegKnock (Xu, 2018) [87]	Uses a genetic algorithm and a regulatory FBA model [88] for the inner problem, where extra constraints are placed on the system to model gene regulation events to maximise chosen metabolite production
FOCuS (Mutturi, 2017) [89]	Divides the total reactions into smaller groups, which are individually evaluated, as a pre-processing step, followed by a combination of flower-pollination algorithm [90] and clonal selection algorithm [91] to maximise metabolite production
GACOFBA (Salleh et al., 2015) [92]	Uses a combination of ant colony optimisation and a genetic algorithm as the outer problem to maximise metabolite production

as shown in Figure 4. This two-stage optimisation problem can be intractable; therefore, OptKnock [77], OptORF [78], and RobustKnock [79] reduce the bilevel problem to a single level problem using duality theory (i.e. an approach that enables optimal solutions of two problems to be found by setting their objectives equal to one another).

Alternatively, ReacKnock [80] uses Lagrangian multipliers (a process for non-linear optimisation) to allow specifically equality constraints, which can then reformulate the bilevel problem into a single level problem [81].

EMILiO [82] also uses linear programming, but iteratively: it begins by pruning the metabolic network to select a subset of the flux constraints that will maximise the metabolite production rate, then it prunes the subsets to minimise the number of reaction modifications. It will then output knockout, activation or inhibition modifications to produce the desired metabolite overproduction.

Optimising metabolism using reaction flux regulation

OptReg [83] and OptForce [84] output reactions to be up-regulated or down-regulated to create a desired flux distribution. They first calculate upper and lower bounds for every reaction flux in the system by iteratively changing the objective function to maximise and minimise each reaction, then compare these ranges to the flux distribution of a metabolism that overproduces a targeted metabolite. It is possible to identify the reactions that require regulation to transform their behaviour into that of the system that overproduces the targeted metabolite. OptForce has the addition of predicting knockouts as well as regulation changes, and also minimises the amount of interventions needed to achieve metabolite overproduction.

Optimising metabolism using metaheuristic algorithms

Other approaches are based on metaheuristic algorithms [85] (Table 2), which are high-level methods used to search a solution space. They are particularly powerful when sampling a large solution space using incomplete information, and often use optimisation methods that contain a degree of stochasticity. Multiple *E. coli* strain GEMs contain
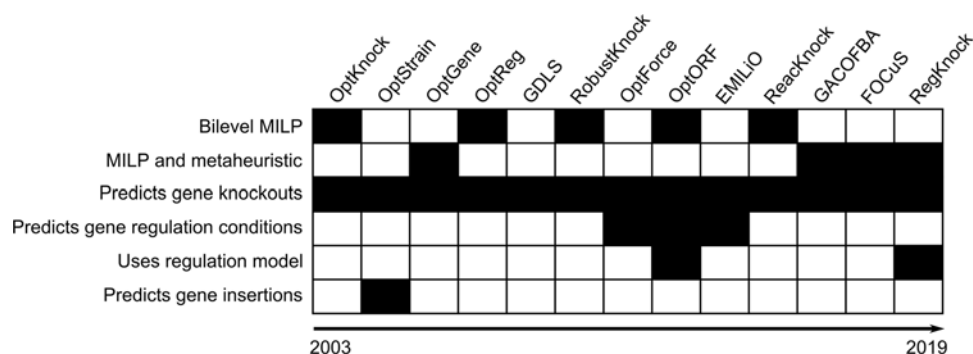
Figure 5. Comparison of metabolic engineering algorithms/frameworks features
Black rectangles indicate feature presence, white rectangles indicate absence.

over 2000 reactions, but the possible combinations for only five gene knockouts is 10^{15}, making the solution space huge. However, metaheuristic algorithms do not guarantee a globally optimal solution and they require significant computational power.

Optimising metabolism using non-native reactions and neighbourhood searching

Additional approaches for algorithmic metabolite optimisation include OptStrain [93], which searches through the KEGG database to find non-native reactions to add to a GEM to optimise metabolite production.

Genetic Design through Local Search (GDLS) [94] iteratively searches through possible solutions (e.g. knockout sets) that differ from the starting conditions using a neighbourhood search (a metaheuristic method for searching over the solution space by exploring solutions in the 'neighbourhood' of the current solution), and stores the best solutions in each iteration. Whereas bilevel linear programming approaches scale exponentially, the runtime for GDLS scales linearly with the number of knockouts, making it more efficient.

Metabolic models and algorithms summary

Choosing an algorithm to use (Figure 5) has to take into account the experimental methodologies available (i.e. the number of knockouts which can be performed), and the available computational power (significantly higher for metaheuristic algorithms). Also, the validity of results has to be critically considered given the accuracy of reaction databases used by some algorithms (e.g. OptStrain), and possible unrealistic results in simulated flux distributions when using entirely stoichiometric representations of metabolic pathways.

Genome engineering *in vivo*

Genome engineering is the production of modified genomes using either a prescriptive genome design or a clear laboratory-based algorithm to design gene edits, and accurate genetic tools that can be used repeatedly.

Genome engineering builds on historical gene essentiality research (see Figure 6). The sequencing of small bacterial genomes [95,96] led to comparative genomics, initially between pairs of bacteria [97], then including greater numbers of bacteria as genome sequencing increased, which led to the development of minimal gene sets [97–99]. However, as the number of microorganisms increased, the number of shared genes decreased: by the thousandth genome sequenced only four genes were shared across all sequenced bacteria [100]. This trend is true even among closely related species, 20 *Mycoplasma* strains were found to share only 196 genes [101]. This is due to non-orthologous gene displacements (NOGDs), independently evolved or diverged proteins that perform the same function but are not recognisably related [20,97]. This comparative work continues to be built on computationally, analysing the growing number of genomic datasets for key features that could match NOGDs (see persistent gene concept [102]); nevertheless, genome engineering has moved to a species-specific focus.

Single gene knockout studies (implemented by systematic removal, inactivation, transposon mutagenesis, and antisense RNA [103]) are still used to provide an initial assessment of gene essentiality, but further work is required, as gene essentiality has been shown to depend both on the environmental context (i.e. how cells are grown) [104] and genomic context (i.e. what other genes present) [105].

PORTLAND PRESS

1953	DNA Double Helix published (Watson and Crick)
1958	Central Dogma of Biology published (Crick)
1977	Sanger sequencing and first genome sequence published (Sanger et al)
1984	*Mycoplasmas* proposed as models for understanding life (Morowitz)
1990	BLAST (Basic Local Alignment Search Tool) published (Altschul et al)
1995	*Haemophilus influenzae* genome sequenced (Fleischmann et al)
	Mycoplasma genitalium genome sequenced (Fraser et al)
1996	*Saccharomyces cerevisiae* genome sequenced (Goffeau et al)
	Comparison of *Haemophilus influenzae* and *Mycoplasma genitalium* genomes (Mushegian and Koonin)
1997	*Escherichia coli* genome sequenced (Blattner et al)
	Bacillus subtilis genome sequenced (Kunst et al)
1998	Lambda Red homologous recombination (recombineering) proteins inserted into *E.coli* (Murphy)
1999	Global transposon mutagenesis of *Mycoplasma genitalium* (Hutchison et al)
	E-Cell: software environment for whole cell simulation published (Tomita et al)
2000	KEGG: Kyoto encyclopedia of genes and genomes published (Kanehisa and Goto)
2001	*E.coli* minimal genome factory project launched
2004	SB1.0: the first international conference for synthetic biology
2006	*E.coli* single gene knockout library (Keio collection) published (Baba et al)
	E.coli MDS43 (15% reduction) published (Posfai et al)
2007	*Saccharomyces cerevisiae* version 2.0 project launched
	Genome transplantation in *Mycoplasmas* published (Lartigue et al)
	Assembly of *Haemophilus influenzae* genome inside of *E.coli* (Holt et al)
2008	Synthetic *M.genitalium* genome assembled inside of *S.cerevisiae* (Gibson et al)
	E.coli double gene knockout library published (Butland et al)
2009	Gibson Assembly published (Gibson et al)
	Genome transplantation of *Mycoplasma mycoides* into *Mycoplasma capricolum* (Lartigue et al)
	MAGE: multiplex automated genome engineering published (Wang et al)
2010	1000th genome sequenced (Lagesen et al)
	Mycoplasma mycoides JCVI-syn1.0 produced (Gibson et al)
2011	Discovery of tracrRNA in *Streptococcus pyogenes* (Deltcheva et al)
	CAGE: conjugative assembly genome engineering and first genome recoded *E.coli* published (Isaacs et al)
	E.coli Δ33a (38.9% reduction) published (Iwadate et al)
	Synthetic *S.cerevisiae* chromosome arm published (Dymond et al)
2012	Fusion of tracrRNA and crRNA to produce CRISPR-Cas9 guidance system (Jinek et al)
	M.genitalium whole-cell model published (Karr et al)
2013	RNA-guided editing of *E.coli* by combining CRISPR-Cas9 and recombineering published (Jiang et al)
	E.coli DGF-298 (35% reduction) published (Hirokawa et al)
	Direct cell to cell transfer of bacterial genomes into yeast (Karas et al)
2014	MEGA method for genome reduction published (Xue et al)
	Synthetic *S.cerevisiae* full chromosome published (Annaluru et al)
2016	*Mycoplasma mycoides* JCVI-syn3.0 (50% reduction) published (Hutchison et al)
	CasHRA method and *E.coli* MGE-syn1.0 (77% reduction) in *S.cerevisiae* published (Zhou et al)
	CRMAGE: method combining CRISPR-Cas9 and MAGE published (Ronda et al)
	RNA-guided editing of *E.coli* using CRISPR-Cas9 and non-homologous end-joining published (Su et al)
	RNA-guided editing of *Mycoplasma* genome within *S.cerevisiae* using CRISPR-Cas9 (Tsarmpopoulos et al)
	rE.coli-57 (57-codon genome) published (Ostrov et al)
2017	*B.subtilis* PG10 and PS38 (36% reductions) published (Reuß et al)
	Five synthetic *S.cerevisiae* chromosomes published (Richardson et al)
2018	MinGenome: genome design algorithm for genome-scale metabolic models published (Wang and Maranas)
	GAMA + Minesweeper: genome design algorithms for whole-cell models published (Rees and Chalkley et al)
2019	*E.coli* Syn61 (synthetic 61-codon genome) published (Fredens et al)

Figure 6. An incomplete history of genome engineering in microorganisms

Consequently, non-essential and essential classifications have been expanded to no essentiality, low essentiality, high essentiality, and complete essentiality [105], with other important classifications for genome engineering including quasi-essential (removal reduces growth rate substantially [106]), synthetic lethal (removal can kill the cell depending on the presence/absence of related genes [107,108]), and synthetic rescue (multiple genes that are essential individually, that can be removed together [109,110]). This redefinition of essentiality has underlined the existence

Table 3 Genome-driven cell engineering examples

Genome Reductions

Microbe	Reduction	Benefits
JCVI-Syn3.0 (Hutchison et al., 2016) [106]	50%	Smallest genome of any autonomously replicating cell. Has a doubling time of ~180 min, four to five times faster than *M. genitalium* (12–15 h [20])
E. coli Δ33a (Iwadate et al., 2011) [113]	39%	-
E. coli DGF-298 (Hirokawa et al., 2013) [9]	35%	Better growth fitness and cell yield, in a rich medium, than the wild-type strain, and has a more stable genome
B. subtilis PG10 and PS38 (Reuß et al., 2017) [112]	36%	Subsequently used for production purposes, as has traits that are favourable for producing 'difficult-to-produce proteins', overcoming several bottlenecks (secretion process and unstable product) [122]
E. coli Δ16 (Hashimoto et al., 2005) [123]	30%	-
B. subtilis MGIM (Ara et al., 2007) [124]	24%	Little reduction in growth rate and comparable enzyme productivity
E. coli MGF-01 (Mizoguchi et al., 2008) [114]	22%	Better growth rate resulting in 1.5-fold cell density and 2.4-fold greater threonine production compared with the wild-type strain
B. subtilis MBG874 (Morimoto et al., 2008) [125]	20%	Extracellular cellulase and protease production were 1.7- and 2.5-fold higher. Production period was elongated and carbon utilisation improved
E. coli MS56 (Park et al., 2014) [126]	23%	Insertion sequence free, making it more genomically stable, predicted to increase production of recombinant proteins
E. coli MDS43 (Posfai et al., 2006) [127]	15%	Showed genome stabilisation and increased electroporation efficiency, comparable with *E. coli* DH10B. Subsequently used for production purposes: 83% increase in L-threonine production, compared with *E. coli* MG1655 with the same metabolic engineering [116]

Genome Recoding

Microbe	Modifications
32 *E. coli* strains (Isaacs et al., 2011) [8]	Replaced 314 TAG (stop) codons with TAA
E. coli MG1655 (Lajoie et al., 2013) [22]	Replaced 321 UAG (stop) codons with UAA
r*E. coli*-57 (Ostrov et al., 2016) [119]	Replaced 62214 instances of seven codons (UAG (stop), AGG and AGA (Arg), AGC and AGU (Ser), UUG and UUA (Leu))
E. coli C123 (Napolitano et al., 2016) [128]	Replaced 123 rare AGA and AGG (Arg) codons from essential genes with 110 CGU conversions and 13 optimised codon substitutions
E. coli MDS42 (Wang et al., 2016) [129]	Tested 1468 codon changes using REXER technology and GENESIS method
S. cerevisiae Sc2.0 (Richardson et al., 2017) [130]	Replaced TAG (stop) codons with TAA
E. coli Syn61 (Fredens et al., 2019) [131]	Replaced 18214 codons, TCG with AGC, TCA with AGT, TAG with TAA, using REXER technology and GENESIS method

of multiple minimal genomes for individual bacterial species, depending on environmental conditions [26,105], and the selection of redundant genetic pathways in the cell [14].

Research for understanding (minimal genomes) and production (chassis development) (see Table 3) both involve large numbers of gene/base pair deletions and use similar genetic tools. However, they differ in intent: no single gene can be removed without loss of viability in minimal genomes [20], whereas the cellular growth rate is maintained or promoted in chassis development. Additionally, minimal genome research focuses on protein-coding genes ignoring: essential promoter regions, tRNAs, small non-coding RNAs [26], regulatory non-coding sequences [103], and the physical layout of the genome [103,111], which are of interest to chassis development. Finally, bacterial species that do not have a use industrially are of use in minimal genome research. *Mycoplasma genitalium* only synthesises DNA, RNA, and proteins from imported precursors, in order to replicate itself [20], which it does slowly in a stress-free laboratory environment [106]; useful for understanding but not for industry.

Of the largest scale reductions to date (see Table 3): *JCVI-Syn3.0* [106], and *B. subtilis* PG10 and PS38 [112] were produced for the purposes of understanding, and *E. coli* Δ 33a [113] and *E. coli* DGF-298 [9] were produced as chassis cells for production. Regardless of original intent, minimal genome reduction strains can have emergent beneficial properties [114,115] (see Table 3) in addition to the lower metabolic burden and increased metabolic efficiency produced by reducing gene numbers [116]. Additionally, the reduced internal biochemistry may interfere less with introduced external pathways [117], making for improved chassis cells. Two minimal genome reduction strains have been subsequently used for production purposes (see Table 3).

Research for reducing risks (genome recoding) substitutes synonymous codons (encoding the same amino acid) across an entire genome resulting in: virus resistance (viral replication relies on all 64 codons [21]), prevention of gene transfer [118], and increased translation efficiency [8]. It also produces a blank codon that can be repurposed for a novel function not commonly found in nature [8,21,119]. This incorporation of non-standard amino acid (NSAA) is

276

**PORTLAND
PRESS**

Table 4 Features of an optimal chassis for a wide range of applications

Feature	Description
Genetically stable	Removal of mobile DNA elements (e.g. insertion elements, transposases, phages, integrases, site-specific recombinases) [132]
Genomically recoded	Substitute codons to create blank codons for inclusion of new, non-natural amino acids [8], decreased likelihood of viral infection [21], and horizontal gene transfer [118]
Genome minimised	Removal of competing and unwanted metabolic pathways that divert the resources of the cell away from desired end products [19], resulting in increased capacity for and reduced impact of cellular burden [133,134], and greater robustness and energy efficiency [135]. Also reducing transcriptional regulatory interactions resulting in lower resistance to engineering efforts [132]. Additionally, allows exploitation of larger and optimal precursor pools [136]
Production efficiency	Simple nutritional needs, fast and efficient growth, and efficient secretion systems [6]
Robustness	Tolerance for extreme conditions [6] i.e. strength of cell membrane or wall and appropriate coping mechanisms [26]
Well understood	Sufficient knowledge of the organism's genome and metabolism to produce accurate mathematical models and modularisation of metabolic pathways [26].
Developed tools	A range of established genetic tools for manipulation, including promoters and terminators with varying expression levels, and well-characterised plasmids, to enable titre, rate, and yield improvements and rapid and efficient tuning of genetic components [19]

a form of biocontainment, further reducing risk, as the organism is engineered to be dependent upon the presence of the synthetic NSAA to survive.

Genome recoding is possible due to the development of MAGE (multiplex automated genome engineering) [120] and CAGE (conjugative assembly genome engineering) [8,121], and subsequently REXER [129]. MAGE cyclically targets many genetic locations to conduct mismatches, insertions, deletions in a single cell or across a population of cells, maintaining high efficiency of up to ten targets at a time [120]. This leads to rapid and continuous generation of genetic diversity for strain and pathway engineering. CAGE is a complementary method, assembling modified genomic modules from individual cells into a single genome through cell to cell transfer, and has been used in combination with MAGE to systematically recode codons [8,121].

Combining genome engineering research together can give insights into what an 'optimal' cellular chassis could look like (see Table 4) and suggest research pathways going forward.

Genome engineering *in silico*
Whole-cell models

The first whole-cell model [137] has been produced recently and is an important development for *in silico* cellular research, as the first integration of mathematical models to simulate an entire cell's components. The recency is due to the immense complexity of individual cells. There are many well-characterised cellular subsystem models, such as ordinary differential equation (ODE) or network models for protein interactions [138], but combining different subsystems together has only been feasible in the last decade.

The *Mycoplasma genitalium* whole-cell model [13] consists of 28 linked submodels that simulate different cellular processes e.g. metabolism using FBA and cell division using ODEs. The model is implemented in MATLAB and produces large amounts of output data. Genes can be knocked out of the model, and environmental variables altered, so the cellular behaviour can be examined in various different situations.

A recent application of the *M. genitalium* whole-cell model is *in silico* genome reduction [14]. This is due to the ease and low cost of simulations (with the appropriate computational infrastructure) compared with *in vivo* experiments. Although modelling is never 100% accurate, it can help to shed light on unexplained phenomena and guide the design of lab experiments, producing research more efficiently [139]. However, even with a genome as small as *M. genitalium* (525 genes), the number of possible gene knockout combinations at genome-scale is of the order of ten [127], making simulating every knockout set unattainable due to time and computational constraints.

Algorithms to reduce the solution space can be used. Minesweeper and GAMA (Guess Add Mate Algorithm) [14] identified up to 165 genes that can be removed from the *M. genitalium* genome while still producing a dividing *in silico* cell. Minesweeper approximates a divide-and-conquer algorithm by knocking out gene sets of varying sizes, then combining sets that produced a dividing cell, generating knockout sets of greater size (thus a smaller genome). GAMA begins similarly, knocking out gene sets of varying sizes, followed by a genetic algorithm to combine these sets iteratively over multiple generations, to make the genome smaller.

These approaches are similar to the metaheuristic algorithms used for metabolic engineering, they are purely input/output dependent. These algorithms could potentially be applied to any model, regardless of its formulation, so theoretically scalable from the GEM level to the whole-cell model level. OptGene and GAMA both use a genetic algorithm to achieve their results, except with a different objective function. It is plausible that GAMA could be modified to maximise a metabolite at the whole-cell level, and equally possible that FOCuS and GACOFBA could be applied to whole-cell models for similar purposes or for genome minimisation.

Whole-cell models are a vital new approach for genome design. Combined with flexible algorithms (e.g. genetic algorithms [140] and ant colony optimisation [141]) they can suggest genetic modifications to produce organisms designed for specific purposes, while producing greater understanding of cellular processes and genetic interactions.

Issues

There is a clear need for greater species-specific understanding of the metabolism and the genome. Even well-studied organisms (*B. subtilis* and *E. coli*) have genes with unknown functions and essentiality; bacterial genomes have on an average 33% genes of unknown function [142]. Of the genes with known functions, in most cases we only understand essentiality at the single or double knockout level [143,144]. Current genome reductions have had to identify synthetic lethal interactions as part of their reduction efforts, rather than being able to design around them. If we had a greater grasp of gene product interactions, enabling them to be accurately modelled, this could be avoided. We would also be taking steps towards a proposed end goal of genome design, combining modular components of different bacteria in a novel cell [24,145].

Another approach for genome-driven cell engineering, constructing bacterial genomes from scratch and inserting them into a host cell, is not currently possible in the majority of bacteria due to economic and technological constraints. Economically, bacterial genome production is too expensive for most institutes. Producing JCVI-Syn1.0 was estimated to cost $\sim$\$40,000,000 [146]. Technologically, megabase-sized genomes can be constructed in yeast [147,148], however successful genome transplantation has only been demonstrated in a few *Mycoplasmas* [149–151].

Development of whole-cell models for genome engineering is time and cost intensive. The *M. genitalium* whole-cell model took 10-person years to build [152], resulting in the Karr Lab at Mount Sinai developing automation tools (Datanator and WC-Lang [153]) along the lines of automated tools for producing GSMMs.

Currently, genome engineering has not combined computational and biological research, due to how recently the required tools were developed [13–16] and the difficulty of working with *M. genitalium* in the lab [106]. This is set to change with the upcoming publication of an *E. coli* whole-cell model, as well as other whole-cell models [154]. In combination with compatible genome design algorithms [14] this might allow integrated *in silico* and *in vivo* genome engineering for the first time.

Conclusions

Combining *in silico* and *in vivo* research will soon be possible in genome engineering. With the release of increasingly refined whole-cell models, genome engineers will have an appropriate model, computational design algorithms, and scalable genetic editing technologies. Following the path of metabolic engineering, better design, and construction processes *in silico* and *in vivo* could be applied to a larger scale problem, replacing large quantities of lab work.

The next steps for genome engineering are: (i) the production and publication of new whole-cell models [154]; (ii) the implementation of computational standards to keep the field cohesive and prevent fragmentation [155]; (iii) the testing of *in silico* designs *in vivo* [14]; (iv) and the establishment of routine procedures for *in vivo* genome reductions for species that will soon have whole-cell models.

Summary

- Metabolic engineering has an established process of *in silico* design and *in vivo* construction.

- *In silico* design informing *in vivo* construction is the future of genome engineering.

- Whole-cell models and algorithms for genome design will widen the field of genome engineering.

- Testing current *in silico* predictions *in vivo*, and uniting *in silico* and *in vivo* research in *E. coli*, are the next steps in genome engineering.

PORTLAND
PRESS

Acknowledgments

Competing Interests

The authors declare that there are no competing interests associated with the manuscript.

Data Access Statement

The present study did not generate any new data.

Funding

Author Contribution

C.G., J.RG., S.L., and L.M. were involved in ideation and editing of the paper. S.L. was responsible for Metabolic Engineering *in silico*, Genome Engineering *in silico*, Figures 1–5, and contributed to Issues. J.RG. was responsible for Introduction, Metabolic Engineering *in vivo*, Genome Engineering *in vivo*, Figure 6, Issues, and Conclusion.

Abbreviations

CAGE, conjugative assembly genome engineering; COBRA, constraint-based reconstruction and analysis; FBA, flux balance analysis; FOCuS, flower pollination coupled clonal selection algorithm; GAMA, guess add mate algorithm; GACOFBA, genetic ant colony optimisation flux balance analysis; GDLS, genetic design through local search; GSMM/GEM, genome-scale metabolic model; KEGG, Kyoto encyclopedia of genes and genomes; MAGE, multiplex automated genome engineering; ME, macromolecular expression; MOMA, minimisation of metabolic adjustment; NOGD, non-orthologous gene displacement; NSAA, non-standard amino acid; ODE, ordinary differential equation; ROOM, regulatory on/off minimisation of metabolic flux.

References

1   Serrano, L. (2007) Synthetic biology: promises and challenges. *Mol. Syst. Biol.* **3**, 158, https://doi.org/10.1038/msb4100202
2   Huang, W.E. and Nikel, P.I. (2019) The Synthetic Microbiology Caucus: from abstract ideas to turning microbes into cellular machines and back. *Microb. Biotechnol.* **12**, 5–7, https://doi.org/10.1111/1751-7915.13337
3   O'Malley, M.A., Powell, A., Davies, J.F. and Calvert, J. (2008) Knowledge-making distinctions in synthetic biology. *Bioessays* **30**, 57–65, https://doi.org/10.1002/bies.20664
4   Brophy, J.A.N. and Voigt, C.A. (2014) Principles of genetic circuit design. *Nat. Methods* **11**, 508–520, https://doi.org/10.1038/nmeth.2926
5   Dzieciol, A.J. and Mann, S. (2012) Designs for life: protocell models in the laboratory. *Chem. Soc. Rev.* **41**, 79–85, https://doi.org/10.1039/C1CS15211D
6   Calero, P. and Nikel, P.I. (2019) Chasing bacterial chassis for metabolic engineering: a perspective review from classical to non-traditional microorganisms. *Microb. Biotechnol.* **12**, 98–124, https://doi.org/10.1111/1751-7915.13292
7   Endy, D. (2005) Foundations for engineering biology. *Nature* **438**, 449–453, https://doi.org/10.1038/nature04342
8   Isaacs, F.J. (2011) Precise manipulation of chromosomes *in vivo* enables genome-wide codon replacement. *Science* **333**, 348–353, https://doi.org/10.1126/science.1205822
9   Hirokawa, Y. (2013) Genetic manipulations restored the growth fitness of reduced-genome Escherichia coli. *J. Biosci. Bioeng.* **116**, 52–58, https://doi.org/10.1016/j.jbiosc.2013.01.010
10 Kim, T.Y., Sohn, S.B., Kim, Y.B., Kim, W.J. and Lee, S.Y. (2012) Recent advances in reconstruction and applications of genome-scale metabolic models. *Curr. Opin. Biotechnol.* **23**, 617–623, https://doi.org/10.1016/j.copbio.2011.10.007
11 Xu, N., Ye, C. and Liu, L. (2018) Genome-scale biological models for industrial microbial systems. *Appl. Microbiol. Biotechnol.* **102**, 3439–3451, https://doi.org/10.1007/s00253-018-8803-1
12 Nielsen, J. and Keasling, J.D. (2016) Engineering cellular metabolism. *Cell* **164**, 1185–1197, https://doi.org/10.1016/j.cell.2016.02.004
13 Karr, J.R. (2012) A whole-cell computational model predicts phenotype from genotype. *Cell* **150**, 389–401, https://doi.org/10.1016/j.cell.2012.05.044
14 Rees, J., Chalkley, O., Purcell, O., Marucci, L. and Grierson, C. (2018) Designing genomes using design-simulate-test cycles. *bioRxiv*, https://doi.org/10.1101/344564
15 Wang, L. and Maranas, C.D. (2018) MinGenome: an in silico top-down approach for the synthesis of minimized genomes. *ACS Synth. Biol.* **7**, 462–473, https://doi.org/10.1021/acssynbio.7b00296

16   Jiang, W.Y., Bikard, D., Cox, D., Zhang, F. and Marraffini, L.A. (2013) RNA-guided editing of bacterial genomes using CRISPR-Cas systems. *Nat. Biotechnol.* **31**, 233–239, https://doi.org/10.1038/nbt.2508

17   Zhao, D. (2017) CRISPR/Cas9-assisted gRNA-free one-step genome editing with no sequence limitations and improved targeting efficiency. *Sci. Rep.* **7**, 16624, https://doi.org/10.1038/s41598-017-16998-8

18   Zerbini, F. (2017) Large scale validation of an efficient CRISPR/Cas-based multi gene editing protocol in Escherichia coli. *Microb. Cell Fact.* **16**, 68, https://doi.org/10.1186/s12934-017-0681-1

19   Carr, P.A. and Church, G.M. (2009) Genome engineering. *Nat. Biotechnol.* **27**, 1151–1162, https://doi.org/10.1038/nbt.1590

20   Glass, J.I., Merryman, C., Wise, K.S., Hutchison, III, C.A. and Smith, H.O. (2017) Minimal cells-real and imagined. *Cold Spring Harb. Perspect. Biol.* **9**, a023861, https://doi.org/10.1101/cshperspect.a023861

21   Kohman, R.E., Kunjapur, A.M., Hysolli, E., Wang, Y. and Church, G.M. (2018) From designing the molecules of life to designing life: future applications derived from advances in DNA technologies. *Angew. Chem. Int. Ed. Engl.* **57**, 4313–4328, https://doi.org/10.1002/anie.201707976

22   Lajoie, M.J. et al. (2013) Genomically recoded organisms expand biological functions. *Science* **342**, 357–360, https://doi.org/10.1126/science.1241459

23   Adams, B.L. (2016) The next generation of synthetic biology chassis: moving synthetic biology from the laboratory to the field. *ACS Synth. Biol.* **5**, 1328–1330, https://doi.org/10.1021/acssynbio.6b00256

24   Vickers, C.E., Blank, L.M. and Krömer, J.O. (2010) Grand challenge commentary: Chassis cells for industrial biochemical production. *Nat. Chem. Biol.* **6**, 875–877, https://doi.org/10.1038/nchembio.484

25   Foley, P.L. and Shuler, M.L. (2010) Considerations for the design and construction of a synthetic platform cell for biotechnological applications. *Biotechnol. Bioeng.* **105**, 26–36, https://doi.org/10.1002/bit.22575

26   Xavier, J.C., Patil, K.R. and Rocha, I. (2014) Systems biology perspectives on minimal and simpler cells. *Microbiol. Mol. Biol. Rev.* **78**, 487–509, https://doi.org/10.1128/MMBR.00050-13

27   Thompson, D.B. (2018) The future of multiplexed eukaryotic genome engineering. *ACS Chem. Biol.* **13**, 313–325, https://doi.org/10.1021/acschembio.7b00842

28   Feist, A.M. and Palsson, B.Ø. (2008) The growing scope of applications of genome-scale metabolic reconstructions using Escherichia coli. *Nat. Biotechnol.* **26**, 659–667, https://doi.org/10.1038/nbt1401

29   Thiele, I. and Palsson, B.Ø. (2010) A protocol for generating a high-quality genome-scale metabolic reconstruction. *Nat. Protoc.* **5**, 93–121, https://doi.org/10.1038/nprot.2009.203

30   Paddon, C.J. et al. (2013) High-level semi-synthetic production of the potent antimalarial artemisinin. *Nature* **496**, 528–532, https://doi.org/10.1038/nature12051

31   Deltcheva, E. (2011) CRISPR RNA maturation by trans-encoded small RNA and host factor RNase III. *Nature* **471**, 602, https://doi.org/10.1038/nature09886

32   Jinek, M. (2012) A programmable dual-RNA-guided DNA endonuclease in adaptive bacterial immunity. *Science* **337**, 816–821, https://doi.org/10.1126/science.1225829

33   Pontrelli, S. (2018) Escherichia coli as a host for metabolic engineering. *Metab. Eng.* **50**, 16–46, https://doi.org/10.1016/j.ymben.2018.04.008

34   Gu, Y. (2018) Advances and prospects of *Bacillus subtilis* cellular factories: from rational design to industrial applications. *Metab. Eng.* **50**, 109–121, https://doi.org/10.1016/j.ymben.2018.05.006

35   Biggs, B.W., De Paepe, B., Santos, C. N.S., De Mey, M. and Kumaran Ajikumar, P. (2014) Multivariate modular metabolic engineering for pathway and strain optimization. *Curr. Opin. Biotechnol.* **29**, 156–162, https://doi.org/10.1016/j.copbio.2014.05.005

36   Chen, X., Li, S. and Liu, L. (2014) Engineering redox balance through cofactor systems. *Trends Biotechnol* **32**, 337–343, https://doi.org/10.1016/j.tibtech.2014.04.003

37   Conrado, R.J. (2012) DNA-guided assembly of biosynthetic pathways promotes improved catalytic efficiency. *Nucleic Acids Res.* **40**, 1879–1889, https://doi.org/10.1093/nar/gkr888

38   Gu, Y. (2017) Rewiring the glucose transportation and central metabolic pathways for overproduction of N-acetylglucosamine in *Bacillus subtilis*. *Biotechnol. J.* **12**, 1700020, https://doi.org/10.1002/biot.201700020

39   Hemberger, S. (2011) RibM from *Streptomyces davawensis* is a riboflavin/roseoflavin transporter and may be useful for the optimization of riboflavin production strains. *BMC Biotechnol.* **11**, 119, https://doi.org/10.1186/1472-6750-11-119

40   Liu, Y. (2016) A dynamic pathway analysis approach reveals a limiting futile cycle in N-acetylglucosamine overproducing Bacillus subtilis. *Nat. Commun.* **7**, 11933, https://doi.org/10.1038/ncomms11933

41   Portnoy, V.A., Bezdan, D. and Zengler, K. (2011) Adaptive laboratory evolution–harnessing the power of biology for metabolic engineering. *Curr. Opin. Biotechnol.* **22**, 590–594, https://doi.org/10.1016/j.copbio.2011.03.007

42   Wisselink, H.W., Toirkens, M.J., Wu, Q., Pronk, J.T. and van Maris, A.J.A. (2009) Novel evolutionary engineering approach for accelerated utilization of glucose, xylose, and arabinose mixtures by engineered Saccharomyces cerevisiae strains. *Appl. Environ. Microbiol.* **75**, 907–914, https://doi.org/10.1128/AEM.02268-08

43   Wright, J. (2011) Batch and continuous culture-based selection strategies for acetic acid tolerance in xylose-fermenting Saccharomyces cerevisiae. *FEMS Yeast Res.* **11**, 299–306, https://doi.org/10.1111/j.1567-1364.2011.00719.x

44   Singh, R., Kumar, M., Mittal, A. and Mehta, P.K. (2016) Microbial enzymes: industrial progress in 21st century. *3 Biotech* **6**, 174, https://doi.org/10.1007/s13205-016-0485-8

45   Kim, O.D., Rocha, M. and Maia, P. (2018) A review of dynamic modeling approaches and their application in computational strain optimization for metabolic engineering. *Front. Microbiol.* **9**, 1690, https://doi.org/10.3389/fmicb.2018.01690

46   Orth, J.D., Thiele, I. and Palsson, B.O. (2010) What is flux balance analysis? *Nat. Biotechnol.* **28**, 245–248, https://doi.org/10.1038/nbt.1614

47 Systems Biology UCSD (2018) Systems Biology UCSD Database. *Systems Biology UCSD*,
    http://systemsbiology.ucsd.edu/InSilicoOrganisms/OtherOrganisms

48 Devoid, S. (2013) Automated genome annotation and metabolic model reconstruction in the SEED and Model SEED. *Methods Mol. Biol.* **985**, 17–45,
    https://doi.org/10.1007/978-1-62703-299-5˙2

49 Machado, D., Andrejev, S., Tramontano, M. and Patil, K.R. (2018) Fast automated reconstruction of genome-scale metabolic models for microbial
    species and communities. *Nucleic Acids Res.* **46**, 7542–7553, https://doi.org/10.1093/nar/gky537

50 BiGG Models (2018) BiGG Models: A platform for integrating, standardizing, and sharing genome-scale models. *Systems Biology UCSD*,
    http://bigg.ucsd.edu/

51 King, Z.A. (2015) BiGG Models: a platform for integrating, standardizing and sharing genome-scale models. *Nucleic Acids Res.* **44**, D515–D522,
    https://doi.org/10.1093/nar/gkv1049

52 Aite, M. (2018) Traceability, reproducibility and wiki-exploration for 'à-la-carte' reconstructions of genome-scale metabolic models. *PLoS Comput.
    Biol.* **14**, e1006146, https://doi.org/10.1371/journal.pcbi.1006146

53 Dias, O., Rocha, M., Ferreira, E.C. and Rocha, I. (2010) Merlin: metabolic models reconstruction using genome-scale information. *IFAC Proc. Vol.* **43**,
    120–125, https://doi.org/10.3182/20100707-3-BE-2012.0076

54 Olivier, B.G. (2018) MetaToolkit: MetaDraft. https://systemsbioinformatics.github.io/cbmpy-metadraft/

55 Karp, P.D. (2016) Pathway Tools version 19.0 update: software for pathway/genome informatics and systems biology. *Brief. Bioinform.* **17**, 877–890,
    https://doi.org/10.1093/bib/bbv079

56 Wang, H. (2018) RAVEN 2.0: a versatile toolbox for metabolic network reconstruction and a case study on Streptomyces coelicolor. *PLoS Comput. Biol.*
    14, e1006541, https://doi.org/10.1371/journal.pcbi.1006541

57 Mendoza, S.N., Olivier, B.G., Molenaar, D. and Teusink, B. (2019) A systematic assessment of current genome-scale metabolic reconstruction tools.
    bioRxiv, https://doi.org/10.1101/558411

58 Zanghellini, J., Ruckerbauer, D.E., Hanscho, M. and Jungreuthmayer, C. (2013) Elementary flux modes in a nutshell: properties, calculation and
    applications. *Biotechnol. J.* **8**, 1009–1016, https://doi.org/10.1002/biot.201200269

59 Schellenberger, J. and Palsson, B.Ø. (2009) Use of randomized sampling for analysis of metabolic networks. *J. Biol. Chem.* **284**, 5457–5461,
    https://doi.org/10.1074/jbc.R800048200

60 Varma, A. and Palsson, B.O. (1994) Metabolic flux balancing: basic concepts, scientific and practical use. *Biotechnology* **12**, 994–998,
    https://doi.org/10.1038/nbt1094-994

61 Knorr, A.L., Jain, R. and Srivastava, R. (2007) Bayesian-based selection of metabolic objective functions. *Bioinformatics* **23**, 351–357,
    https://doi.org/10.1093/bioinformatics/btl619

62 Lewis, N.E., Nagarajan, H. and Palsson, B.O. (2012) Constraining the metabolic genotype-phenotype relationship using a phylogeny of in silico
    methods. *Nat. Rev. Microbiol.* **10**, 291–305, https://doi.org/10.1038/nrmicro2737

63 Segre, D., Vitkup, D. and Church, G.M. (2002) Analysis of optimality in natural and perturbed metabolic networks. *Proc. Natl. Acad. Sci. U.S.A.* **99**,
    15112–15117, https://doi.org/10.1073/pnas.232349399

64 Shlomi, T., Berkman, O. and Ruppin, E. (2005) Regulatory on/off minimization of metabolic flux changes after genetic perturbations. *Proc. Natl. Acad.
    Sci. U.S.A.* **102**, 7695–7700, https://doi.org/10.1073/pnas.0406346102

65 Fong, S.S. and Palsson, B.Ø. (2004) Metabolic gene–deletion strains of *Escherichia coli e*volve to computationally predicted growth phenotypes. *Nat.
    Genet.* **36**, 1056, https://doi.org/10.1038/ng1432

66 Edwards, J.S. and Palsson, B.O. (2000) The *Escherichia coli* MG1655 in silico metabolic genotype: Its definition, characteristics, and capabilities. *Proc.
    Natl. Acad. Sci. U.S.A.* **97**, 5528–5533, https://doi.org/10.1073/pnas.97.10.5528

67 Schilling, C.H. (2002) Genome-scale metabolic model of *Helicobacter pylori* 26695. *J. Bacteriol.* **184**, 4582–4593,
    https://doi.org/10.1128/JB.184.16.4582-4593.2002

68 Förster, J., Famili, I., Palsson, B.Ø. and Nielsen, J. (2003) Large-scale evaluation of *in silico* gene deletions in Saccharomyces cerevisiae. *OMICS* **7**,
    193–202, https://doi.org/10.1089/153623103322246584

69 Oh, Y.-K., Palsson, B.O., Park, S.M., Schilling, C.H. and Mahadevan, R. (2007) Genome-scale reconstruction of metabolic network in *Bacillus subtilis*
    based on high-throughput phenotyping and gene essentiality data. *J. Biol. Chem.* **282**, 28791–28799, https://doi.org/10.1074/jbc.M703759200

70 Shinfuku, Y. (2009) Development and experimental verification of a genome-scale metabolic model for Corynebacterium glutamicum. *Microb. Cell
    Fact.* **8**, 43, https://doi.org/10.1186/1475-2859-8-43

71 Lloyd, C.J. (2018) COBRAme: a computational framework for genome-scale models of metabolism and gene expression. *PLoS Comput. Biol.* **14**,
    e1006302, https://doi.org/10.1371/journal.pcbi.1006302

72 O'Brien, E.J., Lerman, J.A., Chang, R.L., Hyduke, D.R. and Palsson, B.Ø. (2013) Genome-scale models of metabolism and gene expression extend and
    refine growth phenotype prediction. *Mol. Syst. Biol.* **9**, 693

73 Kanehisa, M. and Goto, S. (2000) KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.* **28**, 27–30,
    https://doi.org/10.1093/nar/28.1.27

74 Benson, D.A. (2012) GenBank. *Nucleic Acids Res.* **41**, D36–D42, https://doi.org/10.1093/nar/gks1195

75 Hameri, T., Fengos, G., Ataman, M., Miskovic, L. and Hatzimanikatis, V. (2019) Kinetic models of metabolism that consider alternative steady-state
    solutions of intracellular fluxes and concentrations. *Metab. Eng.* **52**, 29–41, https://doi.org/10.1016/j.ymben.2018.10.005

76 Becker, J., Zelder, O., Häfner, S., Schröder, H. and Wittmann, C. (2011) From zero to hero—design-based systems metabolic engineering of
    Corynebacterium glutamicum for l-lysine production. *Metab. Eng.* **13**, 159–168, https://doi.org/10.1016/j.ymben.2011.01.003

77 Burgard, A.P., Pharkya, P. and Maranas, C.D. (2003) Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial
    strain optimization. *Biotechnol. Bioeng.* **84**, 647–657, https://doi.org/10.1002/bit.10803

78 Kim, J. and Reed, J.L. (2010) OptORF: optimal metabolic and regulatory perturbations for metabolic engineering of microbial strains. *BMC Syst. Biol.* **4**, 53, https://doi.org/10.1186/1752-0509-4-53

79 Tepper, N. and Shlomi, T. (2009) Predicting metabolic engineering knockout strategies for chemical production: accounting for competing pathways. *Bioinformatics* **26**, 536–543, https://doi.org/10.1093/bioinformatics/btp704

80 Xu, Z., Zheng, P., Sun, J. and Ma, Y. (2013) ReacKnock: identifying reaction deletion strategies for microbial strain optimization based on genome-scale metabolic network. *PLoS ONE* **8**, e72150, https://doi.org/10.1371/journal.pone.0072150

81 Ye, J.J. and Zhu, D.L. (1995) Optimality conditions for bilevel programming problems. *Optimization* **33**, 9–27, https://doi.org/10.1080/02331939508844060

82 Yang, L., Cluett, W.R. and Mahadevan, R. (2011) EMILiO: a fast algorithm for genome-scale strain design. *Metab. Eng.* **13**, 272–281, https://doi.org/10.1016/j.ymben.2011.03.002

83 Pharkya, P. and Maranas, C.D. (2006) An optimization framework for identifying reaction activation/inhibition or elimination candidates for overproduction in microbial systems. *Metab. Eng.* **8**, 1–13, https://doi.org/10.1016/j.ymben.2005.08.003

84 Ranganathan, S., Suthers, P.F. and Maranas, C.D. (2010) OptForce: an optimization procedure for identifying all genetic manipulations leading to targeted overproductions. *PLoS Comput. Biol.* **6**, e1000744, https://doi.org/10.1371/journal.pcbi.1000744

85 Glover, F.W. and Kochenberger, G.A. (2006) *Handbook of Metaheuristics*, Springer Science & Business Media

86 Patil, K.R., Rocha, I., Förster, J. and Nielsen, J. (2005) Evolutionary programming as a platform for in silico metabolic engineering. *BMC Bioinformatics* **6**, 308, https://doi.org/10.1186/1471-2105-6-308

87 Xu, Z. (2018) RegKnock: identifying gene knockout strategies for microbial strain optimization based on regulatory and metabolic integrated network. *bioRxiv*, https://doi.org/10.1101/438168

88 Covert, M.W. and Palsson, B.Ø. (2002) Transcriptional regulation in constraints-based metabolic models of Escherichia coli. *J. Biol. Chem.* **277**, 28058–28064, https://doi.org/10.1074/jbc.M201691200

89 Mutturi, S. (2017) FOCuS: a metaheuristic algorithm for computing knockouts from genome-scale models for strain optimization. *Mol. Biosyst.* **13**, 1355–1363, https://doi.org/10.1039/C7MB00204A

90 Yang, X.-S. (2012) Flower pollination algorithm for global optimization. *International Conference on Unconventional Computing and Natural Computation*, pp. 240–249, Springer

91 De Castro, L.N. and Von Zuben, F.J. (2000) The clonal selection algorithm with engineering applications. *Proc. GECCO* **2000**, 36–39

92 Salleh, A.H.M (2015) Gene knockout identification for metabolite production improvement using a hybrid of genetic ant colony optimization and flux balance analysis. *Biotechnol. Bioprocess Eng.* **20**, 685–693, https://doi.org/10.1007/s12257-015-0276-9

93 Pharkya, P., Burgard, A.P. and Maranas, C.D. (2004) OptStrain: a computational framework for redesign of microbial production systems. *Genome Res* **14**, 2367–2376, https://doi.org/10.1101/gr.2872004

94 Lun, D.S. (2009) Large-scale identification of genetic design strategies using local search. *Mol. Syst. Biol.* **5**, 296, https://doi.org/10.1038/msb.2009.57

95 Fleischmann, R.D. (1995) Whole-genome random sequencing and assembly of Haemophilus influenzae Rd. *Science* **269**, 496–512, https://doi.org/10.1126/science.7542800

96 Fraser, C.M. (1995) The minimal gene complement of mycoplasma-genitalium. *Science* **270**, 397–403, https://doi.org/10.1126/science.270.5235.397

97 Mushegian, A.R. and Koonin, E.V. (1996) A minimal gene set for cellular life derived by comparison of complete bacterial genomes. *Proc. Natl. Acad. Sci. U.S.A.* **93**, 10268–10273, https://doi.org/10.1073/pnas.93.19.10268

98 Gil, R., Silva, F.J., Pereto, J. and Moya, A. (2004) Determination of the core of a minimal bacterial gene set. *Microbiol. Mol. Biol. Rev.* **68**, 518–537

99 Shuler, M.L., Foley, P. and Atlas, J. (2012) Modeling a minimal cell. In *Microbial Systems Biology* (Navid, A., ed.), vol. **881**, pp. 573–610, Humana Press

100 Lagesen, K., Ussery, D.W. and Wassenaar, T.M. (2010) Genome update: the 1000th genome - a cautionary tale. *Microbiology* **156**, 603–608, https://doi.org/10.1099/mic.0.038257-0

101 Liu, W. (2012) Comparative genomics of Mycoplasma: analysis of conserved essential genes and diversity of the pan-genome. *PLoS ONE* **7**, e35698, https://doi.org/10.1371/journal.pone.0035698

102 Acevedo-Rocha, C.G., Fang, G., Schmidt, M., Ussery, D.W. and Danchin, A. (2013) From essential to persistent genes: a functional approach to constructing synthetic life. *Trends Genet.* **29**, 273–279, https://doi.org/10.1016/j.tig.2012.11.001

103 Gil, R. (2006) The minimal gene-set machinery. In *Encyclopedia of Molecular Cell Biology and Molecular Medicine* (Meyers, R.A., ed.), Wiley-VCH Verlag GmbH & Co. KGaA

104 D'Elia, M.A., Pereira, M.P. and Brown, E.D. (2009) Are essential genes really essential? *Trends Microbiol.* **17**, 433–438, https://doi.org/10.1016/j.tim.2009.08.005

105 Rancati, G., Moffat, J., Typas, A. and Pavelka, N. (2018) Emerging and evolving concepts in gene essentiality. *Nat. Rev. Genet.* **19**, 34–49, https://doi.org/10.1038/nrg.2017.74

106 Hutchison, C.A. et al. (2016) Design and synthesis of a minimal bacterial genome. *Science* **351**, aad6523, https://doi.org/10.1126/science.aad6253

107 Kaelin, Jr, W.G. (2005) The concept of synthetic lethality in the context of anticancer therapy. *Nat. Rev. Cancer* **5**, 689–698, https://doi.org/10.1038/nrc1691

108 Nijman, S.M.B. (2011) Synthetic lethality: general principles, utility and detection using genetic screens in human cells. *FEBS Lett.* **585**, 1–6, https://doi.org/10.1016/j.febslet.2010.11.024

109 Smalley, D.J., Whiteley, M. and Conway, T. (2003) In search of the minimal *Escherichia coli* genome. *Trends Microbiol.* **11**, 6–8, https://doi.org/10.1016/S0966-842X(02)00008-2

PORTLAND
PRESS

110 Motter, A.E., Gulbahce, N., Almaas, E. and Barabási, A.-L. (2008) Predicting synthetic rescues in metabolic networks. *Mol. Syst. Biol.* **4**, 168, https://doi.org/10.1038/msb.2008.1

111 Mushegian, A. (1999) The minimal genome concept. *Curr. Opin. Genet. Dev.* **9**, 709–714, https://doi.org/10.1016/S0959-437X(99)00023-4

112 Reuß, D.R. (2017) Large-scale reduction of the *Bacillus subtilis* genome: consequences for the transcriptional network, resource allocation, and metabolism. *Genome Res.* **27**, 289–299, https://doi.org/10.1101/gr.215293.116

113 Iwadate, Y., Honda, H., Sato, H., Hashimoto, M. and Kato, J.-I. (2011) Oxidative stress sensitivity of engineered *Escherichia coli* cells with a reduced genome. *FEMS Microbiol. Lett.* **322**, 25–33, https://doi.org/10.1111/j.1574-6968.2011.02331.x

114 Mizoguchi, H., Sawano, Y., Kato, J. and Mori, H. (2008) Superpositioning of deletions promotes growth of *Escherichia coli* with a reduced genome. *DNA Res.* **15**, 277–284, https://doi.org/10.1093/dnares/dsn019

115 Moya, A. (2009) Toward minimal bacterial cells: evolution vs. design. *FEMS Microbiol. Rev.* **33**, 225–235, https://doi.org/10.1111/j.1574-6976.2008.00151.x

116 Lee, J.H. (2009) Metabolic engineering of a reduced-genome strain of *Escherichia coli* for L-threonine production. *Microb. Cell Fact.* **8**, 2, https://doi.org/10.1186/1475-2859-8-2

117 Martínez-García, E. and de Lorenzo, V. (2016) The quest for the minimal bacterial genome. *Curr. Opin. Biotechnol.* **42**, 216–224, https://doi.org/10.1016/j.copbio.2016.09.001

118 Wang, L. (2018) Synthetic genomics: from DNA synthesis to genome design. *Angew. Chem. Int. Ed. Engl.* **57**, 1748–1756, https://doi.org/10.1002/anie.201708741

119 Ostrov, N. (2016) Design, synthesis, and testing toward a 57-codon genome. *Science* **353**, 819–822, https://doi.org/10.1126/science.aaf3639

120 Wang, H.H. (2009) Programming cells by multiplex genome engineering and accelerated evolution. *Nature* **460**, 894–898, https://doi.org/10.1038/nature08187

121 Ma, N.J., Moonan, D.W. and Isaacs, F.J. (2014) Precise manipulation of bacterial chromosomes by conjugative assembly genome engineering. *Nat. Protoc.* **9**, 2285–2300, https://doi.org/10.1038/nprot.2014.081

122 Aguilar Suarez, R., Stülke, J. and van Dijl, J.M. (2019) Less is more: towards a genome-reduced *Bacillus* cell factory for 'difficult proteins'. *ACS Synth. Biol.* **8**, 99–108, https://doi.org/10.1021/acssynbio.8b00342

123 Hashimoto, M. (2005) Cell size and nucleoid organization of engineered *Escherichia coli* cells with a reduced genome. *Mol. Microbiol.* **55**, 137–149, https://doi.org/10.1111/j.1365-2958.2004.04386.x

124 Ara, K. (2007) Bacillus minimum genome factory: effective utilization of microbial genome information. *Biotechnol. Appl. Biochem.* **46**, 169–178, https://doi.org/10.1042/BA20060111

125 Morimoto, T. (2008) Enhanced recombinant protein productivity by genome reduction in Bacillus subtilis. *DNA Res.* **15**, 73–81, https://doi.org/10.1093/dnares/dsn002

126 Park, M.K. (2014) Enhancing recombinant protein production with an Escherichia coli host strain lacking insertion sequences. *Appl. Microbiol. Biotechnol.* **98**, 6701–6713, https://doi.org/10.1007/s00253-014-5739-y

127 Posfai, G. (2006) Emergent properties of reduced-genome Escherichia coli. *Science* **312**, 1044–1046, https://doi.org/10.1126/science.1126439

128 Napolitano, M.G. (2016) Emergent rules for codon choice elucidated by editing rare arginine codons in Escherichia coli. *Proc. Natl. Acad. Sci. U.S.A.* **113**, E5588–97, https://doi.org/10.1073/pnas.1605856113

129 Wang, K. (2016) Defining synonymous codon compression schemes by genome recoding. *Nature* **539**, 59–64, https://doi.org/10.1038/nature20124

130 Richardson, S.M. (2017) Design of a synthetic yeast genome. *Science* **355**, 1040–1044, https://doi.org/10.1126/science.aaf4557

131 Fredens, J. (2019) Total synthesis of *Escherichia coli* with a recoded genome. *Nature* **569**, 514–518, https://doi.org/10.1038/s41586-019-1192-5

132 Henry, C.S., Overbeek, R. and Stevens, R.L. (2010) Building the blueprint of life. *Biotechnol. J.* **5**, 695–704, https://doi.org/10.1002/biot.201000076

133 Borkowski, O. (2018) Cell-free prediction of protein expression costs for growing cells. *Nat. Commun.* **9**, 1457, https://doi.org/10.1038/s41467-018-03970-x

134 Ceroni, F. and Ellis, T. (2018) The challenges facing synthetic biology in eukaryotes. *Nat. Rev. Mol. Cell Biol.* **19**, 481–482, https://doi.org/10.1038/s41580-018-0013-2

135 Mol, M., Kabra, R. and Singh, S. (2018) Genome modularity and synthetic biology: Engineering systems. *Prog. Biophys. Mol. Biol.* **132**, 43–51, https://doi.org/10.1016/j.pbiomolbio.2017.08.002

136 Park, S.Y., Yang, D., Ha, S.H. and Lee, S.Y. (2018) Metabolic engineering of microorganisms for the production of natural compounds. *Adv. Biosys.* **2**, 1700190, https://doi.org/10.1002/adbi.201700190

137 Goldberg, A.P. (2018) Emerging whole-cell modeling principles and methods. *Curr. Opin. Biotechnol.* **51**, 97–102, https://doi.org/10.1016/j.copbio.2017.12.013

138 Pržulj, N., Corneil, D.G. and Jurisica, I. (2004) Modeling interactome: scale-free or geometric. *Bioinformatics* **20**, 3508–3515, https://doi.org/10.1093/bioinformatics/bth436

139 Waltemath, D. and Wolkenhauer, O. (2016) How modeling standards, software, and initiatives support reproducibility in systems biology and systems medicine. *IEEE Trans. Biomed. Eng.* **63**, 1999–2006, https://doi.org/10.1109/TBME.2016.2555481

140 Davis, L. (1991) *Handbook of Genetic Algorithms*, 1st Edition, Van Nostrand Reinhold

141 Dorigo, M. and Di Caro, G. (1999) Ant colony optimization: a new meta-heuristic. *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, vol. **2**, pp. 1470–1477, IEEE

142 Price, M.N. (2018) Mutant phenotypes for thousands of bacterial genes of unknown function. *Nature* **557**, 503–509, https://doi.org/10.1038/s41586-018-0124-0

143 Baba, T. (2006) Construction of *Escherichia coli* K-12 in-frame, single-gene knockout *mutants*: the Keio collection. *Mol. Syst. Biol.* **2**, 1–11, https://doi.org/10.1038/msb4100050

144 Butland, G. (2008) eSGA: E. coli synthetic genetic array analysis. *Nat. Methods* **5**, 789–795, https://doi.org/10.1038/nmeth.1239

145 Gibson, D.G. (2014) Programming biological operating systems: genome design, assembly and activation. *Nat. Methods* **11**, 521–526, https://doi.org/10.1038/nmeth.2894

146 Sleator, R.D. (2010) The story of Mycoplasma mycoides JCVI-syn1.0: the forty million dollar microbe. *Bioeng. Bugs* **1**, 229–230, https://doi.org/10.4161/bbug.1.4.12465

147 Gibson, D.G. (2008) Complete chemical synthesis, assembly, and cloning of a *Mycoplasma genitalium* genome. *Science* **319**, 1215–1220, https://doi.org/10.1126/science.1151721

148 Zhou, J., Wu, R., Xue, X. and Qin, Z. (2016) CasHRA (Cas9-facilitated Homologous Recombination Assembly) method of constructing megabase-sized DNA. *Nucleic Acids Res.* **44**, e124, https://doi.org/10.1093/nar/gkw475

149 Lartigue, C. (2007) Genome transplantation in bacteria: changing one species to another. *Science* **317**, 632–638, https://doi.org/10.1126/science.1144622

150 Lartigue, C. (2009) Creating bacterial strains from genomes that have been cloned and engineered in yeast. *Science* **325**, 1693–1696, https://doi.org/10.1126/science.1173759

151 Baby, V. (2018) Cloning and transplantation of the Mesoplasma florum genome. *ACS Synth. Biol.* **7**, 209–217, https://doi.org/10.1021/acssynbio.7b00279

152 Szigeti, B. (2018) A blueprint for human whole-cell modeling. *Curr. Opin. Syst. Biol.* **7**, 8–15, https://doi.org/10.1016/j.coisb.2017.10.005

153 Karr, J. (2018) Tools for building, simulating, analyzing whole-cell models. *Whole Cell Model.*, https://www.wholecell.org/tools/

154 Karr, J. (2018) Comprehensive whole-cell computational models of individual cells. *Whole Cell Model.*, https://www.wholecell.org/models/

155 Waltemath, D. (2016) Toward community standards and software for whole-cell modeling. *IEEE Trans. Biomed. Eng.* **63**, 2007–2014, https://doi.org/10.1109/TBME.2016.2560762