An engineering theory of evolution

Simeon D. Castle¹, Claire S. Grierson¹,² and Thomas E. Gorochowski¹,²,*

¹ School of Biological Sciences, University of Bristol, Tyndall Avenue, Bristol, BS8 1TQ, UK
² BrisSynBio, University of Bristol, Tyndall Avenue, Bristol, BS8 1TQ, UK

* Correspondence should be addressed to T.E.G. (thomas.gorochowski@bristol.ac.uk)

Keywords: evolution; evotype; engineering theory; genotype-phenotype maps; directed evolution; selection; synthetic biology; bioengineering; evolvability; self-adaptive systems
Abstract

Biological technologies are fundamentally unlike any other because biology evolves. Bioengineering therefore requires novel design methodologies with evolution at their core. Knowledge about evolution is currently applied to the design of biosystems *ad hoc*. Unless we have a unified engineering theory of evolution, we will neither be able to meet evolution’s potential as an engineering tool, nor understand or limit its unintended consequences on our biological designs. Our concept of the evotype offers a conceptual framework for engineering the evolutionary potential of biosystems. We show how a biosystem’s evolutionary properties might be designed by engineering genetic variation, the production of function, and selection. This idea could apply to all biosystems – from individual proteins to communities of whole-cells or even entire ecosystems – whether the goal is to direct evolution in the design process, or to limit its impacts during application. These principles could even be used beyond the realm of bioengineering to design entirely synthetic auto-adaptive technologies.
Engineering evolution

The past few decades have seen a revolution in our ability to engineer biology and create living systems with novel functions\(^1\). Yet, several hurdles still hinder our capability to harness biology’s full potential\(^2\). These predominantly stem from the fact that engineering the stuff of life is not the same as engineering its properties because life evolves. Evolution makes engineering living systems a radically different challenge to engineering other mediums. To be effective, we cannot simply apply traditional engineering design principles to biology and deal with evolution as a secondary thought. If nothing in biology makes sense except in the light of evolution\(^3\), then evolution must be a central part of an engineering theory of biology.

Evolution poses both a challenge and an opportunity when designing biosystems. On one hand it is a detrimental force that can unpick the meticulous plans of an engineer through genetic variation\(^4\). Designed biosystems cannot escape evolution when used, and loss of function is a particular concern for engineers, especially as there are often selection pressures working against the design’s function\(^5,6\). It is essential that we learn to build *evolutionarily stable* biosystems that can continue to operate under unavoidable evolutionary forces.

On the other hand, evolution is an extremely effective problem solver and engineers have exploited this fact for decades\(^7\text--}^{10}\). For example, directed evolution can be used to optimise or even generate completely novel traits in proteins\(^11\) or cells\(^12\). However, these methods rely on the ability of evolution to find solutions in a reasonable length of time. For most systems, the search space is so vast that the starting point in this process must have the potential to generate useful phenotypes relatively quickly.

Evolution may even be employed as a feature of the system during operation. For example, adaptive systems that evolve in response to environmental cues, or evolvable genetic circuits that can be designed with specific classes of phenotype that are reached as necessary through evolutionary change. To create such systems, it is critical that the biological design is *specifically evolvable*. This means it must have the potential to generate the types of phenotypes desired by the engineer from a single starting point in a reasonable time frame.

Even more critical is our moral obligation to develop a deeper understanding of how synthetic biosystems will continue to evolve if deployed into our bodies or the wider environment\(^13\). The field has rightly made efforts to develop tools to reduce and mitigate evolution\(^14\), with fail-safes such as kill switches\(^15\) or metabolic dependencies\(^16\). However, without a good theoretical understanding of how synthetic biosystems might continue to evolve once deployed, we risk these technologies developing unexpected faults with dire, but avoidable, consequences. Even breeding has at times had dire consequences. Notably, the inadvertent creation of the hyper-aggressive Africanised bee, which has had a severe impact...
on humans and ecology. As we develop technologies capable of even more rapid genetic change, such as gene drives, these concerns will become even more salient.

Central to many of these issues is the view in traditional engineering disciplines that the engineered artefact is a final destination in the design process. This view breaks for biology. Instead, we believe that a new perspective is needed for a truly effective engineering of biology; one that sees a designed biosystem as a starting point in a lineage of possibilities.

Whilst much of evolutionary biology has concerned itself with organisms’ histories, bioengineers must consider the future, and specifically how a biosystem will continue to evolve when used. Here, we describe a framework that enables this transition and offers a way to specify, test and conceive the properties of biosystems in terms of their evolutionary potential, and not just their phenotype (Figure 1). This provides a way to reimagine biological engineering so it that works in harmony with life’s ability to evolve.

The design type and the evotype

To better harness the capabilities of biology, we need a way of thinking about the evolutionary properties of engineered biosystems. We must design for potential evolutionary change and not just the immediate functionalities displayed by a system (i.e. its phenotypic traits). Though these are properties of populations yet to exist, they can still be considered in the context of an individual biosystem. We consider the design type as the system that has been engineered consisting of a single genotype. The design type could be any biosystem capable of evolution: a protein, genetic circuit, virus, cell, animal or plant, or even an ecosystem. We introduce the concept of the evotype to capture the evolutionary properties of that system. The evotype is the set of evolutionary dispositions of the design type, analogous to genotype and phenotype being sets of genes and traits, respectively. Unlike a trait, a disposition is not a directly observable property, rather it is a potential property of the system. For example, a protein may have the evolutionary disposition of instability where its structure may change dramatically when mutated. Designing the dispositions of the evotype is a challenge fundamental to engineering biology.

For all but the very simplest biosystems, it is impractical to enumerate every potential evolutionary disposition, just as it is impossible to consider every trait of the phenotype. Instead, an appropriate sample of the evotype must be used for the purpose at hand, just as samples of traits are used when describing the phenotype. How we take this sample, and thus the scope of the evotype covered, should be determined by knowledge of the design type, its intended function, and the context in which it will be used. This could include the size of population, environment, and required number of generations over which the system must operate reliably.
Broadly speaking, we may wish to seek one of two goals when designing the evotype: the first is that of *evolutionary stability*, where a system changes its function as little as possible as it evolves during use, and the second is *specific evolvability*, where the system can easily evolve new phenotypes of a specific class (i.e. the classes of function specified by the engineer), or adapt to changes in the environment (i.e. continuing to produce a desired chemical product).

Specific evolvability requires an element of robustness: core functions of the phenotype must remain unperturbed throughout sequence space so that new phenotypes can be explored. This is analogous to natural evolvability \(^1\) where the ability to generate novel phenotypes alone is insufficient, as they must also be adaptive. It also relates to the concept of plasticity, which is the ability to generate new features without total loss of function \(^2\). The relationship between robustness and evolvability in natural evolution has been explored in detail in prior literature\(^3\). For example, a genetic circuit may have been specified to produce an OR logic function in response to two inputs. That is, it expresses an output protein if either one of the two input chemicals are present. A population with an evolutionarily stable version of this circuit is likely to maintain an OR function during use. A specifically evolvable version of the circuit on the other hand might be designed to readily produce other logic functions when evolved (e.g. AND, NOR, NOT), without simply destroying existing functionality or causing lethality to the host cell.

Whether evolutionary stability or specific evolvability is the goal, it can be achieved through engineering genetic variation, the production of function from genotype, and both natural and artificial selection. How these processes interact to constrain and bias evolution can be understood by describing a landscape surrounding the design type in sequence space. This extends and generalises the fitness landscape concept as applied to natural systems\(^4\) by accounting for the roles of variation, production of function, and selection in engineered biosystems. The bioengineer’s goal is to sculpt this evotype landscape to their specification to ensure it has a structure in line with their requirements.

**Engineering genetic variation**

Not all point mutations are equally likely; transversions and transitions differ in their likelihoods \(^5\), and methylation \(^6\), genomic context \(^7\), and species \(^8\) all influence local and global mutation rates. Furthermore, *algorithmic mutations* \(^9\) may occur. These are mutations that result in changes of several nucleotides in one event (thus an algorithm can describe the change) and can be thought of as shortcuts through sequence space (Figure 1B, left). The likelihood of an algorithmic mutation may be much greater than the summed likelihoods of the equivalent sequence of individual point mutations. For example, the chance of an insertion of the two-base motif ‘AC’ into a tandem repeat region due to slipped strand mispairing may be more...
likely than two insertion events of ‘A’ and ‘C’ occurring independently. Recombination and mobile genetic elements are other examples of biological processes capable of producing algorithmic mutations.

Sequence space is therefore not explored in a uniformly random way, even discounting for the role of selection. Instead, the paths evolution can take are determined by the variation operator set, which defines all the different point and algorithmic mutations that can occur in the system. Each variation operator in this set has an associated probability distribution that represents the likelihood of arriving at a given sequence from another (i.e. by this operator acting on the design type). The distributions of the variation operator set combine to produce the variation probability distribution. This describes the chance of arriving at any given sequence from the design type due to all the biochemical and physical processes capable of causing genetic variation that are present in the system (Figure 1B, right). The variation operator set defines the rate and the likely directions in sequence space a design will explore during evolution. As a design type evolves, the variation probability distribution changes as further dispositions become available.

The variation operator set depends on the specifics of the biosystem being engineered, and the set to be applied in practice is dependent on available knowledge of the system. For example, the variation operator set of a design type biosystem may be said to include transition mutations, transversion mutations, and recombination, each associated with a unique probability that varies across the design type’s sequence. A sample population can be generated by applying the operator set to the design type. This population, with the design type at its centre, may be named a quasispecies; as is used for the related concept in viral evolution. The variation probability distribution can be considered in all stages of the engineering process. Global and local mutation rates could be specified in a design, and standardised mutation rates could be listed in part datasheets. It is likely that improvements to the prediction of mutation probabilities will be made with the increasing availability of sequence data and associated computational methods. Design rules for influencing local genetic variability are already known (e.g. avoiding the reusing of parts and repetitive sequences to reduce homologous recombination and indel mutations), and global mutation rates can also be rationally engineered and manipulated.

A large toolkit for altering for controlling genetic variation has already been created by bioengineers, which could be used to improve evolutionary stability or increase specific evolvability. New tools will doubtless be developed from the diverse mechanisms that generate genetic variation in nature. The variation probability distribution of the design type can be modified by either adding or removing variation operators (e.g. by adding or removing DNA modifying enzymes), or by modulating existing operators in the system across the
genotype. This may be through altering DNA sequence properties (e.g. avoiding simple sequence repeats to reduce the chance of indels through slipped-strand mispairing\(^\text{35}\)). Variation operators can be highly targeted like the DNA methylation of specific bases to increase likelihood of mutation through spontaneous deamination \(^\text{26}\), or may have a global effect such as the removal of error prone polymerases from a host organism \(^\text{39}\). Orthogonal mutation systems that modulate genetic variation of a specific plasmid or region of DNA can be used to overcome genomic error-thresholds, increasing the potential for directed evolution \(^\text{40}\).

Larger-scale genetic variation can be achieved through mechanisms such as site-specific recombination, which can be used for inserting, removing, duplicating, inverting, or shuffling large segments of DNA, exemplified by the SCRaMbLE system used in the synthetic yeast Sc2.0 \(^\text{41}\). Finally, acquisition of foreign DNA either from other organisms in the population through sex, horizontal gene transfer, or from free oligonucleotides in the environment\(^\text{12}\) may also be engineered. By combining these and other biochemical tools, it may eventually be possible to precisely design the variation operator set to produce complex combinations of genetic variation. For example, the variation operator set of a genetic circuit may be engineered by avoiding repeat parts (removing the homologous recombination operator), using a host with a high-fidelity DNA polymerase (globally reducing probability of point mutations), and by incorporating DNA recombination sites (adding an operator for specific DNA recombination, perhaps to be used for future directed evolution). Table 1 provides examples of methods for controlling variation operators that have been developed so far.

**Engineering the production of function**

Each genotype produces a phenotype via the processes of gene expression, growth, and development. However, due to the constraints and biases of these processes \(^\text{42}\) phenotypes are not necessarily distributed evenly throughout sequence space, and not all conceivable phenotypes may be possible. Many systems have shown similar properties in the structure of their mapping from genotype to phenotype. Namely, redundancy (there are many more genotypes than phenotypes), and bias (a small fraction of phenotypes are over-represented). This has been shown both through simulation \(^\text{43}\) and experimentally in RNA and protein structures \(^\text{43}\), and DNA binding sites \(^\text{44}\). How these principles apply to more complex biosystems is a major challenge due to their vast genotype spaces. Nevertheless, there will be a statistical structure in the mapping of genotype to phenotype. If this structure is sufficiently well understood it could offer a powerful way of engineering the evotype.

Engineered biosystems have phenotypic traits that influence reproductive fitness, and traits that influence *function* – the behaviour or properties specified by the designer (though
these are not mutually exclusive). The structure of the mapping from genotype space to function space is therefore a key part of the evotype. Function space may be discrete, or it may be continuous (Figure 1C). Designed functions could be literal mathematical functions, physical characteristics like colour or size, or combinations of several properties.

Any designed system has a degree of utility – the extent to which the system fulfils the specified function. The sole goal of a traditional engineering design process is to maximise the utility of the design type. However, the topology of the function landscape surrounding the design type is also important. It may be rugged and highly variable with the function rapidly changing across sequence space, or it may be smooth and have large neutral regions where function changes little or remains constant. These properties are a key design consideration: what is the functional range to be covered by the design type’s evotype? Should the variation be large for increased evolvability, or limited, for evolutionary stability? Most likely the function landscape should be smooth and predictable, but how is this best achieved? Which regions of function space must be avoided, and which can be tolerated? For example, it may be necessary to reduce irrelevant or harmful functions as much as possible in a diagnostic application where regions of function space cause false negatives, whilst regions causing false positives can be tolerated.

Designs may have identical functionality but occupy regions of function space with very different topological properties. If a system is designed without considering its surrounding function landscape, a design with an undesirable evotype may be a likely outcome. Systems with identical phenotypes yet differing function landscapes were demonstrated by Schaerli et al., who designed two genetic circuits, both with the same strip-generating function. It was found that each produced a different spectrum of new phenotypes when mutated due to differences in the regulatory mechanisms used \(^45\). We are only just beginning to understand what influences the structure of the mapping from genotype to phenotype. However, there are some general principles which seem to hold across scales and contexts. Fortunately, many of these principles are already familiar to engineering.

**Prevalent phenotypes**

Phenotypes that are more prevalent in sequence space can be both more robust (as they are more likely to be in genotype networks sharing the same phenotype), and more evolvable (as this allows a wider search of genotype space, increasing access to more novel phenotypes)\(^{23}\). Therefore, choosing prevalent components may aid both evolutionary stability and specific evolvability. For example, if designing a protein the codon chosen may influence evolvability: for a leucine residue, if UAA is chosen, its 1-mutant neighbourhood has a lower prevalence for leucine than any other codon (two versus four respectively). Therefore, UAA may have lower evolutionary stability, but higher specific evolvability than other codons (since it is able
to generate a wider range of non-polar amino acids). Remapping the genetic code itself has been suggested as a way of altering its evolvability \(^{46}\). Other examples of applying phenotypic prevalence include choosing RNA or protein structures that are highly represented in sequence space \(^{47,48}\). An interesting question is how the phenotypic prevalence of a system’s parts relates to the overall phenotype compared to higher-order properties. Is the robustness of a genetic circuit’s parts or its network topology a greater determinant of its overall robustness?

**Redundancy**

Redundancy is used in classical engineering and by evolution. It can add robustness by allowing variation of parts of a system without overall loss of function and can aid in evolvability by enabling redundant parts to mutate and thus explore new regions of function space. This can be seen in serial homology, where repeated parts such as limbs or teeth enable evolution of specialised functions \(^{49}\), in gene duplications \(^{50}\), and in the scale-free structure of genetic networks where most nodes can be removed without altering overall function \(^{51}\). Note that redundant parts may either be repeats of the same part, or different parts that can produce the same function.

**Modularity, regularity, and hierarchy**

The organisational properties of a biosystem are a major influence on its evotype. These can be summarised by modularity, regularity, and hierarchy \(^{52}\). Modularity is the division of a system into subsystems (modules), where each has a high degree of internal connectivity, but little interdependency between subsystems. Examples of this can be seen in the connectivity of protein and regulatory networks, in RNA structures, and in limb development \(^{53}\). Regularity is the use of patterns, repetitions, and symmetries (e.g. serial homology and animal body-plans). Hierarchy is the recursive arrangement of a system into subsystems (that are themselves composed of subsystems, etc.) \(^{54}\) For example, an organism is composed of organs, which is in turn composed of tissues, cells, etc. Hierarchy is also seen in gene regulatory networks. For example, only nine proteins regulate half of all genes of *E. coli* \(^{40}\).

These principles are distinct but often work together. For example, identical modules are often repeated in regular patterns, and modules are arranged in a hierarchical structure. These principles may each promote evolvability in different ways. Modularity allows parts of a system to mutate and change function with a reduced negative impact on the rest of the system. Regularity reduces the information required to describe the system (e.g. its genotype), essentially reducing the size of the search space. Hierarchy allows the progressive increase in the complexity of a system from the bottom up \(^{54}\). Though the widespread use of these principles in both biology and technology clearly demonstrates their importance, how and
where these principles should be applied is context specific. This can be illustrated with an imagined example.

Consider two biosensor circuits that each use red, green, and blue (RGB) fluorescent proteins to produce a white output. In circuit A, the overall output of RGB should be as high as possible when the input is positive (e.g. high sensitivity is required), the whiteness of the signal is less critical. In circuit B, it is important that the positive signal remains precisely white (e.g. other colours represent other input conditions), and overall output level is less critical.

Circuit A would benefit from a modular arrangement of red, green, and blue because a mutation in any one of these genes does not affect the other two, thus reducing the impact on overall output. However, for circuit B, a less modular design would be preferable: though a mutation would have triple the effect on overall output, all colours would be impacted equally conserving the overall hue. The nature of the design problem therefore relates to how modularity should be used. In fact, Kashtan et al. showed that modular architectures evolve in gene regulatory networks in response to modular environmental selection pressures, and themselves prove to be more evolvable. Similar relationships between how hierarchy and regularity should relate to the design problem no doubt exist. However, we are far from having design principles for their application, particularly for more complex problems.

**Environmental robustness**

Principles that improve robustness to environmental change or noise may also improve robustness to genetic change. For example, if a genetic circuit is robust to noise in the concentration of a regulatory protein, it may also be more robust to mutations that change the promoter’s expression level. Similarly, proteins that are more thermodynamically stable may also be more evolvable. Systems could be buffered against environmental and genetic perturbations through the use of negative feedback.

**Designing parameter space**

The structure of the parameter space of a system plays a large role in how function changes under genetic variation. If the behaviour of a system can be modelled or inferred against the variation of key parameters, this can provide information about which functions may be accessible and most likely throughout sequence space. For example, by modelling the regulatory mechanisms of two genetic circuits, Schaeli et al. explain why they produce different distributions of functions when subjected to point mutations. Similarly, with a simple mathematical model of equilibrium binding, Mayo et al. showed that the cis-regulatory region of the **lac**-operon in *E. coli* is incapable of accessing some input functions via point mutation. Parameter spaces are analogous to the morpho-spaces of evolutionary-developmental biology, which provide constraints on organismal form.
Other principles

System specific principles may also provide design rules for the evotype. For example, RNA gene regulation may be less evolvable than transcriptional regulation so could determine whether the regulation is applied at the transcriptional or translational level. Physical and chemical processes of self-organisation may even be able to reduce a function’s dependency on the genotype. Perhaps ideas from developmental biology and morphogenesis could be recast into engineering terms, such as concepts from the theory of facilitated variation, particularly as bioengineering progresses to multicellular organisms. Metaheuristic design approaches will also no doubt become an increasingly powerful tool: machine learning approaches may be able to predict the evolvability of biological networks, and genetic algorithms have been used to evolve more robust genetic networks in silico.

Engineering natural and artificial selection

Selection is the force that gives the otherwise random (but constrained) processes of genetic variation a ‘direction’ by driving a population up the slopes of the adaptive landscape. Uniquely, an engineered biosystem is a result of two forms of selection: natural selection and the design process. Natural selection acts on reproductive fitness of the biosystem, and the design process can be thought of as a sophisticated form of artificial selection acting on its utility. Fitness and utility both form part of the evotype and understanding the interplay between these two processes is critical for effective evotype design since there is often a tension between the two (Figure 1D). If fitness and utility are uncorrelated, then natural selection is likely to undo the work of the engineer. However, if fitness and utility are highly correlated, then natural selection will also drive up utility. For example, one might design a cell in a bioreactor, or a plant crop to produce a chemical product. Perhaps it uses control circuitry to maintain optimal metabolic fluxes to maximise yield in fluctuating environmental conditions, thus resulting in high utility. However, this will inevitably have a fitness effect on the organism (e.g. due to metabolic burden of the circuit, or toxicity of the product), and thus natural selection will favour mutants where this functionality is repressed. It should be noted that natural selection here is meant as the process that acts on the reproductive ability of the biosystem. Neither the environment nor biosystem need to be natural (e.g. the organisms could be engineered to make use of non-canonical amino acids and grown within a bioreactor). The critical distinction is that natural selection acts on survival of the biosystem without the input of the engineer.

The aim of a bioengineer then, is to maximise fitneity – defined as a function that combines the derivatives of utility and fitness (Figure 1D). Ultimately, evotype engineering is
controlling how fitneity changes throughout sequence space: it is the sculpting of fitneity landscape. Exactly what form the fitneity function should take, and the best way to mathematically describe the fitneity landscape to effectively capture the interaction between these two forms of selection is not yet clear. Defining, modelling, and characterising the fitneity landscapes of designed biological systems is a future avenue of research ripe with potential. Nevertheless, the concept can already help in thinking about how to improve the fitneity of designs on an intuitive level.

To design for evolutionary stability, it is sufficient to limit or neutralise the impact of natural selection. This can be achieved one of three ways. Firstly, the fitness of the design type and its immediate neighbours can be increased to create a local peak or plateau. This could be done through adaptive evolution after the design phase, reducing or dynamically controlling burden, or by reducing toxicity of the associated function. Secondly, the fitness of neighbouring genotypes can be decreased to flatten the surrounding fitness landscape, for example by using organisms with a reduced genome that may be less fit than wild type organisms, but with freed-up metabolic resources. Thirdly, the utility landscape can be flattened so that even if there is a natural selection pressure away from the design-type, it is less likely to impact the design’s function. Approaches for doing this have been outlined in the previous section. To ensure a specifically evolvable evotype, fitness and utility must correlate: both fitness and utility must slope in the same direction. An engineer could do this by coupling function to survival, perhaps through a toxin-anti-toxin system, or by coupling function to growth (e.g. by having the product of a system aid in metabolism of an energy source). Alternatively, an artificial environmental pressure, such as repeated screening, could be used to ensure utility and fitness correlate.

**Toward evotype engineering**

The evotype is a new way to think about the properties of engineered biosystems and how they relate to each other (Table 2). It is a framework for thinking about an important but often overlooked property: the role the biosystem itself plays in its future evolution. This is especially critical due to the impact of an intervention (e.g. a new mutational method) being closely linked to the composition of the system itself. For example, in a simple case, an identical protein could be encoded by very different sequences and so be impacted by a targeted mutating element in different ways. This is quite different to how engineers normally view systems. As engineered biosystems are the result of both human creativity and natural adaptation, a holistic consideration of both the roles of design and evolution is necessary. The evotype helps us do this by explicitly considering the intertwined effects that genetic variation, production of function, and multiple forms of selection will have on a design (Figure 1).
We can now design and build genotypes with great precision, but we must account for the inevitable processes of genetic variation that will follow. The statistical structure of variation is unique to each biosystem and something we have control over. Yet understanding the details of genetic variation is insufficient if we do not also understand how this will manifest in changes of the designed function of the biosystem. Even a system with low mutation rates can be evolutionarily unstable if function changes wildly with small sequence alterations. Directed evolution will not be successful, despite the mutation strategy, if desired functions are simply not accessible from the starting point. If the biosystem's utility (i.e. its success as a design) and its fitness (i.e. its success as a biological replicator) are at odds, well designed dispositions for variation or function might not save the design from the pressure of natural selection. This must also be understood as conflict between utility and fitness landscapes across sequence space surrounding the original design type. It is clear then that all three of the aspects of the evotype must be considered together, and all offer significant scope for engineering. For instance, imagine a large genetic circuit that places an unavoidably high metabolic burden on the host cell. If it is crucial that the function of the circuit is maintained over long periods of time, then redundancy could be used to accommodate unavoidable mutations. However, if the dent to reproductive fitness is severe, this may still not be enough. Therefore, combining redundancy in the design with a hyper-stable host cell (e.g. one where all mobile genetic elements have been deleted and efficient DNA repair mechanisms are present 71) might be the only way to achieve the desired goal for the system.

Designing biosystems with evolution in mind is a vital step towards a more complete engineering theory of biology. However, to be practical, supporting tools must exist that can provide key information regarding the genetic variation, genotype-function map, and selective pressures within a biosystem. Advances in sequencing offer a means to quantitatively measure millions of genotypes in parallel 72 and when combined with high-throughput techniques such as fluorescence activated cell sorting (FACS) make it possible to infer simplified genotype-function maps 73. The local function landscapes of the green fluorescent protein74, and transcription factor binding sites 44 have already been characterised experimentally with such methods. Detailed measurements of fitness in large populations of cells are also possible 75–77. By combining sequencing with expression and growth measurements, genetic variation, function, and fitness could be characterised simultaneously to provide a complete picture of the evotype.

Even so, the vastness of evotype landscapes and the need for functions calculated from many outputs of a system mean that new methods with greater throughputs are also necessary. There is a particular need for methods able to measure many characteristics of each cell simultaneously (e.g. via automated high-content microscopy 78 or high-throughput Raman spectroscopy 79). Parallel to these experimental methods, a promising direction to
bypass the need to directly measure these properties are the development of sufficiently comprehensive computational models (e.g. encompassing whole cells) to allow for a mechanistic understanding of the biases in processes related to variation and reproductive rate. In these cases, if they are sufficiently accurate, the evotype could be predicted and used within computer aided-design workflows to reduce the need to physically implement every possible design. Nevertheless, for systems of even moderate complexity, the evotype landscapes are much too vast to be exhaustively characterised or even modelled. It will therefore be of great importance to understand how they should be sampled, how large a region of the landscape needs to be characterised, and to what extent local landscape properties can be extrapolated. Use of machine learning is another method that holds great promise for increasing the ability to estimate the evotype landscape, albeit at the cost of mechanistic knowledge of the system.

In addition to characterising evotypes, tools for bioengineers to directly sculpt their landscapes must also be available (Figure 2). Here, we have touched upon the numerous repurposed biomolecular components that can alter the types of possible variation. However, there is a spectacular diversity of molecular machines dedicated to manipulating genetic information in the natural world, suggesting a need for an even larger toolkit to precisely modify genetic variation as needed. Likewise, principles for constraining and biasing the production of function and for controlling selection pressures have been suggested, but they are still poorly understood, and have barely been applied rigorously in an engineering setting.

An engineering theory of evolution is a new way of looking at evolutionary theory. It is also a new way of thinking about what it is that engineers do, and what the design process is in the context of bioengineering. The concept of the evotype, with some modifications, may also find use in evolutionary science, where it offers a framework for considering the mechanistic constraints of evolution and a way of talking about the evolutionary characteristics of organisms. It may also be applied beyond biological engineering fields to create new auto-adaptive technologies. Here the framework could be applied to ask how we design technologies to evolve, and not just how to engineer systems that already do.

Acknowledgements

S.D.C. was supported by the EPSRC/BBSRC Centre for Doctoral Training in Synthetic Biology grant EP/L016494/1. T.E.G. was supported by a Royal Society University Research Fellowship grant UF160357. T.E.G. and C.S.G. were supported by BrisSynBio, a BBSRC/EPSRC Synthetic Biology Research Centre grant BB/L01386X/1. This study did not involve any underlying data.
Author Contributions

Whilst T.E.G. and C.S.G. framed questions, suggested figures and tables and provided direction, the core concepts and new terminology here are mostly the work of S.D.C., who also wrote the first draft. S.D.C. and T.E.G. developed the figures with input from C.S.G. T.E.G. and C.S.G edited the manuscript and supervised the work.

Declaration of Interest

The authors declare no competing financial interests.
References

1. Andrianantoandro, E. et al. Synthetic biology: New engineering rules for an emerging discipline. *Molecular Systems Biology* 2, 2006.0028 (2006).

2. Grozinger, L. et al. Pathways to cellular supremacy in biocomputing. *Nature Communications* 1–11 (2019) doi:10.1038/s41467-019-13232-z.

3. Dobzhansky, T. Nothing in Biology Makes Sense except in the Light of Evolution. *The American Biology Teacher* 35, 125–129 (1973).

4. Renda, B. A., Hammerling, M. J. & Barrick, J. E. Engineering reduced evolutionary potential for synthetic biology. *Mol. BioSyst.* 10, 1668–1678 (2014).

5. Ellis, T. Predicting how evolution will beat us. *Microbial Biotechnology* 12, 41–43 (2019).

6. Fernandez-Rodriguez, J., Yang, L., Gorochowski, T. E., Gordon, D. B. & Voigt, C. A. Memory and Combinatorial Logic Based on DNA Inversions: Dynamics and Evolutionary Stability. *ACS Synthetic Biology* 4, 1361–1372 (2015).

7. Yokobayashi, Y., Weiss, R. & Arnold, F. H. Directed evolution of a genetic circuit. *Proceedings of the National Academy of Sciences of the United States of America* 99, (2002).

8. Giver, L., Gershenson, A., Freskgard, P.-O. & Arnold, F. H. Directed evolution of a thermostable esterase. *PNAS* 95, 12809–12813 (1998).

9. Boder, E. T., Midelfort, K. S. & Wittrup, K. D. Directed evolution of antibody fragments with monovalent femtomolar antigen-binding affinity. *Proceedings of the National Academy of Sciences* 97, 10701–10705 (2000).

10. Esvelt, K. M., Carlson, J. C. & Liu, D. R. A system for the continuous directed evolution of biomolecules. *Nature* 472, 499–503 (2011).

11. Guntas, G., Mansell, T. J., Kim, J. R. & Ostermeier, M. Directed evolution of protein switches and their application to the creation of ligand-binding proteins. *Proc. Natl. Acad. Sci. U.S.A.* 102, 11224–11229 (2005).

12. Wang, H. H. et al. Programming cells by multiplex genome engineering and accelerated evolution. *Nature* 460, 894–898 (2009).

13. Anderson, J. et al. Engineering and ethical perspectives in synthetic biology. Rigorous, robust and predictable designs, public engagement and a modern ethical framework are vital to the continued success of synthetic biology. *EMBO Rep* 13, 584–590 (2012).

14. Wright, O., Stan, G.-B. & Ellis, T. Building-in biosafety for synthetic biology. *Microbiology (Reading, Engl.)* 159, 1221–1235 (2013).

15. Chan, C., Lee, J., Cameron, E., Bashor, C. & Collins, J. ‘Deadman’ and ‘Passcode’ microbial kill switches for bacterial containment. *Nat Chem Biol* 12, 82–86 (2015).
16. Mandell, D. J. et al. Biocontainment of genetically modified organisms by synthetic protein design. *Nature* **518**, 55–60 (2015).

17. Winston, M. L. The Biology and Management of Africanized Honey Bees. *Annu. Rev. Entomol.* **37**, 173–193 (1992).

18. Oye, K. A. et al. Regulating gene drives. *Science* **345**, 626–628 (2014).

19. Pigliucci, M. Are ecology and evolutionary biology "soft" sciences? *Ann. Zool. Fennici* **39**, 87–98 (2002).

20. Bartley, B. A., Kim, K., Medley, J. K. & Sauro, H. M. Synthetic Biology: Engineering Living Systems from Biophysical Principles. *Biophys J* **112**, 1050–1058 (2017).

21. Kirschner, M. & Gerhart, J. Evolvability. *Proceedings of the National Academy of Sciences of the United States of America* **95**, 8420–8427 (1998).

22. Mayo, A. E., Setty, Y., Shavit, S., Zaslaver, A. & Alon, U. Plasticity of the cis-regulatory input function of a gene. *PLoS Biology* **4**, 555–561 (2006).

23. Wagner, A. Robustness and evolvability: a paradox resolved. *Proceedings of the Royal Society B: Biological Sciences* **275**, 91–100 (2008).

24. Wright, S. *The roles of mutation, inbreeding, crossbreeding, and selection in evolution*. vol. 1 (na, 1932).

25. Stoltzfus, A. & Norris, R. W. On the Causes of Evolutionary Transition: Transversion Bias. *Mol Biol Evol* **33**, 595–602 (2016).

26. Jones, P. A., Rideout, W. M., Shen, J. C., Spruck, C. H. & Tsai, Y. C. Methylation, mutation and cancer. *Bioessays* **14**, 33–36 (1992).

27. Zhu, Y. O., Siegal, M. L., Hall, D. W. & Petrov, D. A. Precise estimates of mutation rate and spectrum in yeast. *Proc. Natl. Acad. Sci. U.S.A.* **111**, E2310-2318 (2014).

28. Drake, J., Chariseworth, B., Chariseworth, D. & Crow, J. Rates of Spontaneous Mutation. *Genetics* **148**, 1667–1686 (1998).

29. Chaitin, G. *Proving Darwin: Making Biology Mathematical*. (Vintage Books, 2013).

30. Levinson, G. & Gutman, G. A. Slipped-strand mispairing: a major mechanism for DNA sequence evolution. *Mol. Biol. Evol.* **4**, 203–221 (1987).

31. Vos, M. Why do bacteria engage in homologous recombination? *Trends in microbiology* **17**, 226–32 (2009).

32. Frost, L. S., Leplae, R., Summers, A. O. & Toussaint, A. Mobile genetic elements: the agents of open source evolution. *Nat. Rev. Microbiol.* **3**, 722–732 (2005).

33. Eigen, M. On the nature of virus quasispecies. *Trends Microbiol.* **4**, 216–218 (1996).

34. Canton, B., Labno, A. & Endy, D. Refinement and standardization of synthetic biological parts and devices. *Nature biotechnology* **26**, 787 (2008).

35. Jack, B. R. *et al*. Predicting the Genetic Stability of Engineered DNA Sequences with the EFM Calculator. *ACS Synth Biol* **4**, 939–943 (2015).
36. Sleight, S. C. & Sauro, H. M. Visualization of Evolutionary Stability Dynamics and Competitive Fitness of Escherichia coli Engineered with Randomized Multigene Circuits. *ACS Synth. Biol.* 2, 519–528 (2013).

37. Hossain, A. et al. Automated design of thousands of nonrepetitive parts for engineering stable genetic systems. *Nature Biotechnology* 1–10 (2020) doi:10.1038/s41587-020-0584-2.

38. Geng, P., Leonard, S. P., Mishler, D. M. & Barrick, J. E. Synthetic Genome Defenses against Selfish DNA Elements Stabilize Engineered Bacteria against Evolutionary Failure. *ACS Synth. Biol.* 8, 521–531 (2019).

39. Csörgő, B., Fehér, T., Tímár, E., Blattner, F. R. & Pósfai, G. Low-mutation-rate, reduced-genome Escherichia coli: an improved host for faithful maintenance of engineered genetic constructs. *Microbial Cell Factories* 11, 11 (2012).

40. Ravikumar, A., Arzumanyan, G. A., Obadi, M. K. A., Javanpour, A. A. & Liu, C. C. Scalable, Continuous Evolution of Genes at Mutation Rates above Genomic Error Thresholds. *Cell* 175, 1946-1957.e13 (2018).

41. Dymond, J. & Boeke, J. The Saccharomyces cerevisiae SCRaMbLE system and genome minimization. *Bioeng Bugs* 3, 168–171 (2012).

42. Ahnert, S. E. Structural properties of genotype–phenotype maps. *Journal of The Royal Society Interface* 14, 20170275 (2017).

43. Ferrada, E. & Wagner, A. A Comparison of Genotype-Phenotype Maps for RNA and Proteins. *Biophys J* 102, 1916–1925 (2012).

44. Aguilar-Rodríguez, J., Payne, J. L. & Wagner, A. A thousand empirical adaptive landscapes and their navigability. *Nature Ecology & Evolution* 1, 0045 (2017).

45. Schaefer, Y. et al. Synthetic circuits reveal how mechanisms of gene regulatory networks constrain evolution. *Molecular Systems Biology* 14, 1–18 (2018).

46. Pines, G., Winkler, J. D., Pines, A. & Gill, R. T. Refactoring the Genetic Code for Increased Evolvability. *mBio* 8, mBio.01654-17, e01654-17 (2017).

47. Schaper, S. & Louis, A. A. The Arrival of the Frequent: How Bias in Genotype-Phenotype Maps Can Steer Populations to Local Optima. *PLOS ONE* 9, e86635 (2014).

48. Li, H., Helling, R., Tang, C. & Wingreen, N. Emergence of Preferred Structures in a Simple Model of Protein Folding. *Science* 273, 666–669 (1996).

49. Carroll, S. Homology, Serial Homology, and Williston’s Law. in *Endless Forms Most Beautiful* 29–36 (Weidenfeld & Nicolson, 2006).

50. Zhang, J. Evolution by gene duplication: an update. *Trends in Ecology & Evolution* 18, 292–298 (2003).

51. Albert, R., Jeong, H. & Barabási, A.-L. Error and attack tolerance of complex networks | *Nature*. *Nature* 406, 268–382 (2001).
52. Lipson, H. Principles of modularity, regularity, and hierarchy for scalable systems. *Journal of Biological Physics and Chemistry* **2007**, *125–128* (2007).

53. Wagner, G., Pavlicev, M. & Cheverud, J. The road to modularity. *Focus on Evo-Devo* **8**, 921–931 (2007).

54. Simon, H. A. The Architecture of Complexity. in *Facets of Systems Science* (ed. Klir, G. J.) 457–476 (Springer US, 1991).

55. Kashtan, N. & Alon, U. Spontaneous evolution of modularity and network motifs. *PNAS* **102**, 13773–13778 (2005).

56. Kaneko, K. Evolution of robustness to noise and mutation in gene expression dynamics. *PLoS ONE* **2**, (2007).

57. Bloom, J. D., Labthavikul, S. T., Otey, C. R. & Arnold, F. H. Protein stability promotes evolvability. *PNAS* **103**, 5869–5874 (2006).

58. Afrooz, T. & Beisel, C. L. Understanding and exploiting feedback in synthetic biology. *Chemical Engineering Science* **103**, 79–90 (2013).

59. Brakefield, P. M. Evo-devo and constraints on selection. *Trends in Ecology & Evolution* **21**, 362–368 (2006).

60. Payne, J. L., Khalid, F. & Wagner, A. RNA-mediated gene regulation is less evolvable than transcriptional regulation. *PNAS* **115**, E3481–E3490 (2018).

61. Gerhart, J. & Kirschner, M. The theory of facilitated variation | *PNAS. Proceedings of the National Academy of Sciences* **104**, 8582–8589 (2007).

62. Kim, H., Muñoz, S., Osuna, P. & Gershenson, C. Antifragility Predicts the Robustness and Evolvability of Biological Networks through Multi-Class Classification with a Convolutional Neural Network. *Entropy* **22**, 986 (2020).

63. Noman, N., Monjo, T., Moscato, P. & Iba, H. Evolving Robust Gene Regulatory Networks. *PLoS One* **10**, e0116258 (2015).

64. Kauffman, S. A. *The Origins of Order: Self-Organization and Selection in Evolution*. (Oxford University Press, USA, 1993).

65. Wannier, T. M. *et al.* Adaptive evolution of genomically recoded *Escherichia coli*. *Proc Natl Acad Sci USA* **115**, 3090–3095 (2018).

66. Springman, R., Molineux, I. J., Duong, C., Bull, R. J. & Bull, J. J. Evolutionary Stability of a Refactored Phage Genome. *ACS Synth. Biol.* **1**, 425–430 (2012).

67. Ceroni, F. *et al.* Burden-driven feedback control of gene expression. *Nature Methods* **15**, 387–393 (2018).

68. Kurokawa, M., Seno, S., Matsuda, H. & Ying, B.-W. Correlation between genome reduction and bacterial growth. *DNA Res* **23**, 517–525 (2016).
69. Martínez-García, E., Nikel, P. I., Aparicio, T. & de Lorenzo, V. Pseudomonas 2.0: genetic upgrading of P. putida KT2440 as an enhanced host for heterologous gene expression. Microb Cell Fact 13, 159 (2014).

70. Stieber, D., Gabant, P. & Szpirer, C. Y. The art of selective killing: plasmid toxin/antitoxin systems and their technological applications. BioTechniques 45, 344–346 (2008).

71. Umenhoffer, K. et al. Genome-Wide Abolishment of Mobile Genetic Elements Using Genome Shuffling and CRISPR/Cas-Assisted MAGE Allows the Efficient Stabilization of a Bacterial Chassis. ACS Synth. Biol. 6, 1471–1483 (2017).

72. Reuter, J. A., Spacek, D. V. & Snyder, M. P. High-throughput sequencing technologies. Mol. Cell 58, 586–597 (2015).

73. Sharon, E. et al. Inferring gene regulatory logic from high-throughput measurements of thousands of systematically designed promoters. Nature Biotechnology 30, 521–530 (2012).

74. Sarkisyan, K. S. et al. Local fitness landscape of the green fluorescent protein. Nature 533, 397–401 (2016).

75. Nevozhay, D., Adams, R. M., Itallie, E. V., Bennett, M. R. & Balázsi, G. Mapping the Environmental Fitness Landscape of a Synthetic Gene Circuit. PLOS Computational Biology 8, e1002480 (2012).

76. Cira, N. J., Pearce, M. T. & Quake, S. R. Neutral and selective dynamics in a synthetic microbial community. Proc Natl Acad Sci USA 115, E9842–E9848 (2018).

77. van Opijnen, T., Bodi, K. L. & Camilli, A. Tn-seq: high-throughput parallel sequencing for fitness and genetic interaction studies in microorganisms. Nature Methods 6, 767–772 (2009).

78. Pepperkok, R. & Ellenberg, J. High-throughput fluorescence microscopy for systems biology. Nat. Rev. Mol. Cell Biol. 7, 690–696 (2006).

79. Goda, K. et al. High-throughput single-microparticle imaging flow analyzer. PNAS 109, 11630–11635 (2012).

80. Marucci, L. et al. Computer-Aided Whole-Cell Design: Taking a Holistic Approach by Integrating Synthetic With Systems Biology. Front. Bioeng. Biotechnol. 8, 942 (2020).

81. Nielsen, A. A. K. et al. Genetic circuit design automation. Science 352, aac7341–aac7341 (2016).

82. du Plessis, L., Leventhal, G. E. & Bonhoeffer, S. How Good Are Statistical Models at Approximating Complex Fitness Landscapes? Molecular Biology and Evolution 33, 2454–2468 (2016).

83. McDonald, J. I. et al. Reprogrammable CRISPR/Cas9-based system for inducing site-specific DNA methylation. Biology open 5, 866–74 (2016).
84. Nivina, A. et al. Structure-specific DNA recombination sites: Design, validation, and machine learning–based refinement. *Science Advances* **6**, eaay2922 (2020).

85. Romanini, D. W., Peralta-Yahya, P., Mondol, V. & Cornish, V. W. A Heritable Recombination System for Synthetic Darwinian Evolution in Yeast. *ACS Synth. Biol.* **1**, 602–609 (2012).

86. Reis, A. C. et al. Simultaneous repression of multiple bacterial genes using nonrepetitive extra-long sgRNA arrays. *Nature Biotechnology* **37**, 1294–1301 (2019).

87. Umenhoffer, K. et al. Reduced evolvability of Escherichia coli MDS42, an IS-less cellular chassis for molecular and synthetic biology applications. *Microbial Cell Factories* **9**, (2010).

88. Nyerges, Á. et al. CRISPR-interference-based modulation of mobile genetic elements in bacteria. *Synthetic Biology (Oxford, England* **4**, (2019).

89. Le Breton, Y., Mohapatra, N. P. & Haldenwang, W. G. In Vivo Random Mutagenesis of Bacillus subtilis by Use of TnYLB-1, a mariner-Based Transposon. *Appl Environ Microbiol* **72**, 327–333 (2006).

90. Greener, A., Callahan, M. & Jerpseth, B. An efficient random mutagenesis technique using an E. coli mutator strain. *Methods Mol Biol* **57**, 375–385 (1996).

91. Badran, A. H. & Liu, D. R. Development of potent in vivo mutagenesis plasmids with broad mutational spectra. *Nature Communications* **6**, 8425 (2015).

92. Halperin, S. O. et al. CRISPR-guided DNA polymerases enable diversification of all nucleotides in a tunable window. *Nature* **560**, 248–252 (2018).

93. Camps, M., Naukkarinen, J., Johnson, B. P. & Loeb, L. A. Targeted gene evolution in Escherichia coli using a highly error-prone DNA polymerase I. *PNAS* **100**, 9727–9732 (2003).

94. Vojta, A. et al. Repurposing the CRISPR-Cas9 system for targeted DNA methylation. *Nucleic Acids Research* **44**, 5615–5628 (2016).

95. Hess, G., Frésard, L., Han, K., Lee, C. & Bassik, M. Directed evolution using dCas9-targeted somatic hypermutation in mammalian cells | *Nature Methods. Nature Methods* **13**, 1036–1042 (2016).

96. Tyo, K. E. J., Ajikumar, P. K. & Stephanopoulos, G. Stabilized gene duplication enables long-term selection-free heterologous pathway expression. *Nature Biotechnology* **27**, 760–765 (2009).

97. Albert, R. Scale-free networks in cell biology. *J Cell Sci* **118**, 4947–4957 (2005).

98. Park, Y., Espah Boruji, A., Gorochowski, T. E., Shin, J. & Voigt, C. A. Precision design of stable genetic circuits carried in highly-insulated E. coli genomic landing pads. *Molecular Systems Biology* **16**, e9584 (2020).
99. Meyer, A. J., Ellefson, J. W. & Ellington, A. D. Directed Evolution of a Panel of Orthogonal T7 RNA Polymerase Variants for in Vivo or in Vitro Synthetic Circuitry. *ACS Synthetic Biology* **4**, 1070–1076 (2015).

100. Kyllilis, N., Tuza, Z. A., Stan, G.-B. & Polizzi, K. M. Tools for engineering coordinated system behaviour in synthetic microbial consortia. *Nature Communications* **9**, 2677 (2018).

101. Wei, S.-P. *et al.* Formation and functionalization of membraneless compartments in Escherichia coli. *Nature Chemical Biology* **16**, 1143–1148 (2020).

102. Xiang, N. *et al.* Using synthetic biology to overcome barriers to stable expression of nitrogenase in eukaryotic organelles. *PNAS* **117**, 16537–16545 (2020).

103. Richardson, S. M. *et al.* Design of a synthetic yeast genome. *Science* **355**, 1040–1044 (2017).

104. Steel, H. & Papachristodoulou, A. Low-Burden Biological Feedback Controllers for Near-Perfect Adaptation. *ACS Synth. Biol.* **8**, 2212–2219 (2019).

105. Gorochowski, T. E., Avci-Kucukgoze, I., Bovenberg, R. A. L., Roubos, J. A. & Ignatova, Z. A Minimal Model of Ribosome Allocation Dynamics Captures Trade-offs in Expression between Endogenous and Synthetic Genes. *ACS Synth. Biol.* **5**, 710–720 (2016).

106. Gorochowski, T. E., Van Den Berg, E., Kerkman, R., Roubos, J. A. & Bovenberg, R. A. L. Using synthetic biological parts and microbioreactors to explore the protein expression characteristics of escherichia coli. *ACS Synthetic Biology* (2014) doi:10.1021/sb4001245.

107. Mittal, P., Brindle, J., Stephen, J., Plotkin, J. B. & Kudla, G. Codon usage influences fitness through RNA toxicity. *Proc Natl Acad Sci USA* **115**, 8639–8644 (2018).

108. Abil, Z., Ellefson, J. W., Gollihar, J. D., Watkins, E. & Ellington, A. D. Compartmentalized partnered replication for the directed evolution of genetic parts and circuits. *Nature Protocols* **12**, 2493–2512 (2017).

109. Yang, G. & Withers, S. G. Ultrahigh-Throughput FACS-Based Screening for Directed Enzyme Evolution. *ChemBioChem* **10**, 2704–2715 (2009).

110. Smith, G. P. & Petrenko, V. A. Phage Display. *Chem. Rev.* **97**, 391–410 (1997).

111. Mozhaev, V. V. & Martinek, K. Structure-stability relationships in proteins: new approaches to stabilizing enzymes. *Enzyme and Microbial Technology* **6**, 50–59 (1984).

112. Archetti, M. Genetic robustness and selection at the protein level for synonymous codons. *Journal of Evolutionary Biology* **19**, 353–365 (2006).
### Engineering Genetic Variation

| Mechanism                      | Type of Variation            | Stability                          | Evolvability                     | Examples                                                                 |
|-------------------------------|-----------------------------|-----------------------------------|----------------------------------|--------------------------------------------------------------------------|
| Site specific recombination   | Inversions, Deletions, Duplications, Rearrangement | Remove host recombinases           | Targeted recombination systems   | CRISPR-recombinase, SCRAMBLE, De novo recombination sites               |
| Homologous recombination      | Rearrangement, Deletions, Duplications | Remove host recombinases           | Inducible recombination          | Heritable recombination system                                           |
|                               |                             | Avoid repeated sequences           | Design in repeated sequences      | EFM calculator, Automated design of non-repetitive parts                 |
| Mobile genetic elements       | Rearrangement               | Remove Transposons                 | Transposon mutagenesis           | Abolishment of Mobile Genetic Elements, CRISPR-controlled insertion sequences, In vivo transposon mutagenesis |
| DNA polymerase fidelity       | Global point mutation rate  | Eliminate error-prone polymerases  | Mutator strains                  | Low mutation E. coli, X11-Red Mutator strain, Mutagenesis plasmids, CRISPR-DNA polymerase |
|                               | Local point mutation rate   | --                                | Targeted DNA polymerase          | CRISPR-DNA polymerase                                                  |
|                               | Plasmid-specific mutation   | --                                | Orthogonal mutation plasmids      | Orthorep, Pol 1 mutagenesis system                                      |
|                               | rate                        |                                   |                                  |                                                                          |
| Slipped-strand Mismatching    | Targeted small indels       | Avoid simple sequence repeats      | Employ simple sequence repeats    | EFM calculator                                                        |
| DNA Methylation               | Point mutation              | Remove host DNA methyltransferase  | Targeted methylation             | CRISPR-methylation                                                     |
| Cytosine Deamination          | Point mutation              | --                                | Targeted deamination             | CRISPR-deamination                                                     |

### Engineering the Production of Function

| Principle                  | Stability                          | Evolvability                     | Examples                                                                 |
|----------------------------|-----------------------------------|----------------------------------|--------------------------------------------------------------------------|
| Prevalence                 | Use prevalent phenotypes          | Use prevalent phenotypes         | Designable protein structures                                            |
| Redundancy                 | Use multiple copies of genes/constructs | Use multiple copies of genes/constructs | High gene-copy system for E. coli                                      |
| Modularity                 | Insulate genetic parts            | Insulate function                | Ribozyme insulators, Insulated genetic landing pads                      |
| Use Orthogonal systems     | Use Orthogonal systems            |                                  | T7 RNA polymerase                                                        |
| Spatial/temporal separation| Spatial/temporal separation       |                                  | Microbial consortia, Targeting to a cell cycle stage, specific cell type or organelle |
| Regularity                 | Standardise parts across system   | Standardise parts across system  | Standardised architecture of the Yeast 2.0 genome                        |
| Hierarchy | Co-control related functions | Co-control related functions | – |
|-----------|----------------------------|----------------------------|---|
|           | Use networks with wide, shallow hierarchies | Use networks with wide, shallow hierarchies | – |
| Environmental Robustness | Feedback control | Feedback control | Feedback control of transcription and translation[^104] |
|           | Use larger tolerances for parts | Use larger tolerances for parts | Maximise dynamic range of genetic logic gates[^81] |
| Designing Parameter Space | Design a constrained parameter space | Design a diverse parameter space | Modelling phenotype distributions of genetic circuits[^45] |

### Engineering Natural and Artificial Selection

| Principle | Stability | Evolvability | Examples |
|-----------|-----------|--------------|----------|
| Increase design type fitness | Evolution for fitness after design phase | – | Adaptive evolution of recoded E. coli[^65], Adaptive evolution of refactored phage genome[^66] |
| Reduce metabolic burden | – | | Modelling ribosome allocation to reduce burden[^105], Characterising burden of genetic parts[^106] |
| Dynamic control of burden | – | Burden-driven feedback control in E. coli[^69] |
| Reduce toxicity | – | mRNA toxicity[^107] |
| Decrease neighbouring fitness | Minimised chassis organism | – | Pseudomonas 2.0[^69] |
| Correlate fitness and utility | Couple function to reproduction | Couple function to reproduction | Phage assisted continuous evolution[^10], Compartmentalized partnered replication[^108] |
| | Couple function to survival | Couple function to reproduction | Toxin/antitoxin systems[^79] |
| | Artificial selection pressure | Artificial selection pressure | FACS screening[^109], Ribosome Display[^89], Phage Display[^110] |

---

[^104]: Reference to feedback control of transcription and translation
[^81]: Reference to maximising dynamic range of genetic logic gates
[^45]: Reference to modelling phenotype distributions of genetic circuits
[^65]: Reference to adaptive evolution of recoded E. coli
[^66]: Reference to adaptive evolution of refactored phage genome
[^105]: Reference to modelling ribosome allocation to reduce burden
[^106]: Reference to characterising burden of genetic parts
[^69]: Reference to Pseudomonas 2.0
[^10]: Reference to phage assisted continuous evolution
[^108]: Reference to compartmentalized partnered replication
[^79]: Reference to toxin/antitoxin systems
[^109]: Reference to FACS screening
[^89]: Reference to ribosome display
[^110]: Reference to phage display

---

1. Methods for production of function are suggestions; the details of how to apply these principles to influence the evotype in design type systems are still in the process of being understood.
Table 2: Role of genotype, phenotype and evotype when describing biological systems.

|                        | Genotype                               | Phenotype                          | Evotype                              |
|------------------------|----------------------------------------|------------------------------------|--------------------------------------|
| **Describes**          | Informational/ Hereditary properties   | Physical/Environmental properties  | Evolutionary properties              |
| **Is a set of...**     | Genes/Sequences                        | Traits                             | Dispositions                         |
| **Pig example**        | Hox gene                               | Four legs                          | Chance of evolving wings             |
| **Protein example**    | Codon                                  | Structural stability under temperature $^a$ | Structural stability under mutation $^b$ |
| **Biosensor circuit example** | Genetic circuit sequence               | Sensor sensitivity/specificity, Input-output function, Fluorescent output, etc. | Probability of: total failure, sensitivity loss, I/O function change, etc. |

a. For example, strengthened hydrophobic interactions in the interiors of thermostable proteins $^{111}$.
b. For example, robustness of the genetic code to amino acid mutations due to synonymous codons $^{112}$.
Figure 1: The evotype and its key properties. (A) The evotype visualised as a landscape surrounding the design type (red square), where fitness (the combined function of fitness and utility) is plotted as a vertical axis against a 2D plane of sequence space with the probability of evolution exploring regions of sequence space overlaid in grey. The properties of this landscape are determined by the interaction of three components: variation, function, and selection. (B) A variation probability distribution can be projected onto sequence space, which represents the likelihood of exploring a given sequence through genetic variation. Darker regions represent regions of higher probability. This is the sum of the distributions of the individual variation operators present in the system (variation operator set). For example, point mutation (bottom layer in set), recombination of homologous regions (middle layer in set), and slip-strand mutation (top layer in set). Red arrows in middle and bottom layers represent algorithmic and point mutations respectively. (C) How phenotypic functions are distributed in sequence space surrounding the design type is critical. Function space may be considered as
discrete (top), where the space may have high genotypic robustness (left grid) or high variability (right grid). A continuous utility space (bottom) plotted against a 1D projection of sequence space. The colour under the curve represents the discrete function associated with that region of sequence space and the utility that each has as a continuous value. For example, if the goal is to produce blue-like functions, dark blue may have the highest utility, followed by lighter variants in the spectrum. The bioengineer must define a minimal threshold (dashed line), below which the design is deemed to be a failure (e.g. non-desired function is exhibited). (D) Sequences differ in their reproductive fitness. This is the driver of natural selection and can be plotted across sequence space as an adaptive landscape (red dotted line). Utility (blue dashed line) may or may not correlate with reproductive fitness across sequence space. The fitneity (grey solid line) is the combination of the fitness and utility. Bioengineers must optimise fitneity both for the design type and throughout the landscape.
Figure 2: Engineering evotypes by sculpting their landscapes. Different biosystem designs may share the same phenotype but have very different evotypes (top row). Rational engineering approaches could be used to transform a naive design (middle column), where evotype has not been considered, into either evolutionarily stable (left column) or specifically evolvable (right column) evotypes, which are characterised by their fitness landscapes. Bioengineers can sculpt the evotype by modifying three major factors: genetic variation, production of function and selection. Genetic variation (green row): In a naive design, a mixture of variation operators may be in play. This might create a system that can reach many different regions of sequence space. It could be made more stable by reducing global mutation rates (e.g. host strain engineering), or by removing homologous regions to reduce the chance of recombination. Conversely, a naive design might be made more evolvable by increasing mutation rates in focused areas of sequence space (e.g. via methylation) and incorporating site-specific recombination or gene-shuffling (e.g. the SCRaMbLE system). Function (blue row): A naive design may have high utility, however, if its function changes rapidly and chaotically across sequence space, it may be inherently
unstable. A robust evotype has large neutral regions in function space. Conversely a design can be made more evolvable if it can access a large range of new phenotypes, of a specific class (e.g. produce a colour), and the landscape may be smoothed (e.g. through removing crosstalk between features) and thus made amenable to evolutionary search. Production of function may be engineered by using prevalent phenotypes, designing in redundancy, modularity, regularity and hierarchy, increasing environmental robustness, or by designing a system's parameter space. Selection (orange row): If, as in the naive design, reproductive fitness (red dotted line) and utility (blue dashed line) are highly uncorrelated, then the design type may have a strong selection pressure acting against it, and regions where both fitness and utility are maximised may be rare or non-existent, so high fitness (grey solid line), may not be achievable. For a stable design, one might act to reduce the effects of natural selection through global increases in fitness (e.g. through reducing metabolic burden of a genetic circuit), by reducing toxicity of gene products, or by reducing the fitness of neighbouring sequences (e.g. using minimised chassis organisms). A naive design can be made more evolvable by closely correlating fitness and utility (e.g. through coupling function to reproduction). This means natural selection will act to drive up the utility of the design: the precise goal of a directed evolution experiment.