Title

Machine learning for metabolic engineering: A review

Authors

Lawson, Christopher E

Martí, Jose Manuel

Radivojevic, Tijana

et al.

Publication Date

2021

DOI

10.1016/j.ymben.2020.10.005

Peer reviewed

Metabolic Engineering 63 (2021) 34–60

Contents lists available at ScienceDirect

Metabolic Engineering

journal homepage: www.elsevier.com/locate/meteng

Machine learning for metabolic engineering: A review

Christopher E. Lawson [a,b], Jose Manuel Martí [a,b,c], Tijana Radivojevic [a,b,c], Sai Vamshi R. Jonnalagadda [a,b,c], Reinhard Gentz [a,b,i], Nathan J. Hillson [a,b,c], Sean Peisert [a,i,j], Joonhoon Kim [b,e], Blake A. Simmons [a,b,c], Christopher J. Petzold [a,b,c], Steven W. Singer [a,b], Aindrila Mukhopadhyay [a,b,g], Deepti Tanjore [a,f], Joshua G. Dunn [h], Hector Garcia Martin [a,b,c,d,g,*]

[a] Biological Systems and Engineering, Lawrence Berkeley National Laboratory, Berkeley, CA, 94720, USA
[b] Joint BioEnergy Institute, Emeryville, CA, 94608, USA
[c] DOE Agile BioFoundry, Emeryville, CA, 94608, USA
[d] Basque Center for Applied Mathematics, 48009, Bilbao, Spain
[e] Pacific Northwest National Laboratory, Richland, 99354, WA, USA
[f] Advanced Biofuels and Bioproducts Process Development Unit, Emeryville, CA, 94608, USA
[g] Environmental Genomics and Systems Biology Division, Lawrence Berkeley National Laboratory, USA
[h] Ginkgo Bioworks, Boston, MA, 02210, USA
[i] Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA, 94720, USA
[j] University of California Davis, Davis, CA, 95616, USA

ARTICLE INFO

Keywords:
Machine Learning
Metabolic Engineering
Synthetic Biology
Deep Learning

ABSTRACT

Machine learning provides researchers a unique opportunity to make metabolic engineering more predictable. In this review, we offer an introduction to this discipline in terms that are relatable to metabolic engineers, as well as providing in-depth illustrative examples leveraging omics data and improving production. We also include practical advice for the practitioner in terms of data management, algorithm libraries, computational resources, and important non-technical issues. A variety of applications ranging from pathway construction and optimization, to genetic editing optimization, cell factory testing, and production scale-up are discussed. Moreover, the promising relationship between machine learning and mechanistic models is thoroughly reviewed. Finally, the future perspectives and most promising directions for this combination of disciplines are examined.

1. Introduction

Metabolic engineering is enjoying an auspicious moment, when its potential is becoming evident in the form of many commercially available products with undeniable impact on society. This discipline has produced: synthetic silk for clothing (Hahn, 2019; Johansson et al., 2014), meatless burgers that taste like meat because of bioengineered heme ("Meat-free outsells beef," 2019), synthetic human collagen for cosmetic purposes ("Geltor unveils first biodesigned human collagen for skincare market", 2019), antimalarial and anticancer drugs (Ajikumar et al., 2010; Paddon and Keasling, 2014), the fragance of recovered extinct flowers (Kiedaisch, 2019), biofuels (Hanson, 2013; Peralta-Yahya et al., 2012), hoppy flavored beer produced without hops (Denby et al., 2018), and synthetic cannabinoids (Dolgin, 2019; Luo et al., 2019), among others. Since the number of possible metabolites is enormous, we can only expect these successes to significantly increase in number in the future.

Traditional approaches, however, limit metabolic engineering to the usual 5–15 gene pathway, whereas full genome-scale engineering holds the promise of much more ambitious and rigorous biodesign of organisms. Genome-scale engineering involves multiplex DNA editing that is not limited to a single gene or pathway, but targets the full genome (Bao et al., 2018; Esvelt and Wang, 2013; Garst et al., 2017; Liu et al., 2015; Si et al., 2017). This approach can open the field of metabolic engineering to stunning new possibilities: engineering of microbiomes for therapeutic or bioremediation uses (Lawson et al., 2019), designing of multicellular organisms as biomaterials that match a specification (Islam et al., 2017), ecosystem engineering (Hastings et al., 2007), and perhaps

* Corresponding author. Biological Systems and Engineering, Lawrence Berkeley National Laboratory, Berkeley, CA, 94710, USA.
E-mail address: hgmartin@lbl.gov (H. Garcia Martin).

https://doi.org/10.1016/j.ymben.2020.10.005
Received 10 August 2020; Received in revised form 22 October 2020; Accepted 31 October 2020
Available online 19 November 2020

even fusion of physical and biological systems. None of these examples are likely to become reality through a traditional trial-and-error approach: the number of genetic part combinations that could produce these outcomes is a vanishingly small fraction of the total possible. For example, engineering a microbiome to produce a medical drug involves not only introducing and balancing the corresponding pathway in one or more of the microbiome species, but also modifying internal regulatory networks so as to keep the community stable and robust to external perturbations. Even for the case of single pathways and teams of highly-trained experts, the trial-and-error approach is hardly sustainable, since it results in very long development times: for example, it took Amyris an estimated 150 person-years of effort to produce the immediate precursor of the antimalarial artemisinin, and Dupont 575 person-years to generate propanediol (Hodgman and Jewett, 2012). An approach that pinpoints the designs that match a desired specification is needed.

The main challenge in more sophisticated biodesign is, arguably, our inability to accurately predict the outcomes of bioengineering (Carbonell et al., 2019; Lopatkin and Collins, 2020). New technologies provide markedly easier ways to make the desired DNA changes, but the final result on cell behavior is usually unpredictable (Gardner, 2013). If metabolic engineering is "the science of rewiring the metabolism of cells to enhance production of native metabolites or to endow cells with the ability to produce new products" (Nielsen and Keasling, 2016), the ability to engineer a cell to a specification (e.g. a given titer, rate and yield of a desired product) is critical for this purpose. Only the ability to accurately predict the performance of a genetic design can avoid an arduous trial-and-error approach to reach that specification.

Moreover, while the flourishing offshoots of the genomic revolution provide powerful new capabilities to discover new DNA sequences, understand their function, and modify them, it is not trivial to harness these technologies productively. The genomic revolution has provided the DNA code as a condensed set of cell instructions that constitutes the main engineering target, and functional genomics to understand the cell behavior. Furthermore, the cost for these data is rapidly decreasing: sequencing cost decreases faster than Moore's law, transcriptomic data grow exponentially (Stephens et al., 2015), and high-throughput workflows for proteomics and metabolomics are slowly becoming a reality (Chen et al., 2019; Zampieri et al., 2017). But many researchers find themselves buried in this "deluge of data": there seems to be more data than time to analyze them. Furthermore, data come in many different types (genomics, transcriptomics, proteomics, metabolomics, protein interaction maps, etc), complicating their analysis. As a result, analysis of functional genomics data often does not yield sufficient insights to infer actionable strategies to manipulate DNA for a desired phenotype. Moreover, CRISPR-based tools (Doudna and Charpentier, 2014; Knott and Doudna, 2018) provide easy DNA editing and metabolic perturbations (e.g. CRISPRi (Tian et al., 2019)). These tools provide the potential to perform genome-wide manipulations in model systems (Wang et al., 2018), and a growing number of hosts (Peters et al., 2019). However, it is not clear how to prioritize the possible targets. Rational engineering approaches have proven useful in the past (George et al., 2015; Kang et al., 2019; Tian et al., 2019), but the detailed knowledge of a pathway can produce on the order of tens of targets, whereas CRISPR-based tools can reach tens of thousands of genome sites (Bao et al., 2018; Bassalo et al., 2018; Garst et al., 2017; Gilbert et al., 2014).

Machine learning (ML) is a possible solution to these problems. Machine learning can systematically provide predictions and recommendations for the next steps to be implemented through CRISPR (or other methods (Paschon et al., 2019; Reyon et al., 2012; Wang et al., 2019)), and it can use the exponentially growing amounts of functional genomics data to systematically improve its performance. Machine learning has already proven its utility in many other fields: self-driving cars (Duarte and Ratti, 2018), automated translation (Wu et al., 2016) , face recognition (Voulodimos et al., 2018), natural language parsing (Kreimeyer et al., 2017), tumor detection (Paeng et al., 2016), and explicit content detection in music lyrics (Chin et al., 2018), among others. It has the potential to produce similar breakthroughs in metabolic engineering.

However, a change in perspective is required regarding the relative importance of molecular mechanisms. Whereas the machine learning paradigm concentrates on enabling predictive power, metabolic engineers typically define scientific value around the understanding of genetic or molecular mechanisms (see section 4.0). Nonetheless, the biological sciences (including computational biology) have been particularly challenged to make accurate quantitative predictions of complex systems from known and tested mechanisms. Hence, if accurate quantitative predictions are needed for a more transformative metabolic engineering, it may be desirable to shift some of the emphasis from identifying molecular mechanisms into enabling data-driven approaches. This apparent detour may, in the end, more efficiently produce mechanistic models, if we combine the predictive power of machine learning with the insight of molecular mechanisms (Heo and Feig, 2020).

In this review we provide an explanation of machine learning in metabolic engineering terms, in the hopes of providing a bridge between both disciplines. We explore the promises of machine learning, as well as its current pitfalls, provide examples of how it has been used so far, as well as auspicious future uses. In short, we will make the case that machine learning can take metabolic engineering to the next step in its maturation as a discipline, but it requires a conscious choice to understand its limitations and potential.

2. Demystifying machine learning for bioengineers

2.1. What is machine learning?

Machine learning is a subdiscipline of Artificial Intelligence (AI), which attempts to emulate how a human brain understands, and interacts with, the world (Fig. 1). A fully functioning AI would enable us to perform the same processes as human metabolic engineers: choose the best molecules to produce, suggest possible pathways to produce it, select the right pathway design to obtain the desired titers, rates and yield, and interpret the resulting experimental data to troubleshoot the metabolic engineering effort. A fully functioning AI would of course be useful for many other tasks such as: fully autonomous cars and planes, recommending medical treatments, directing agricultural practices, reading and summarizing texts like a human, automating translations from different human languages, and producing music and movies. Obviously, we do not yet have full functioning AIs (or strong AI or artificial general intelligence as it is often referred to (Pei et al., 2019; Walch, 2019)), and it is a continuing debate whether we will ever have them (Melnyk, 1996), but AI approaches have been quite successful in some bounded tasks such as playing chess and Go better than humans (Silver et al, 2016, 2018), or predicting protein structures from sequence (AlQuraishi, 2019). Since AI and machine learning are generally applicable tools, some of these partial successes can be very useful for metabolic engineering (see section 3 for examples).

Machine learning is the study of computer algorithms that seek to improve automatically through experience (i.e. learning), often by training on supervised examples (Fig. 2), also known as supervised machine learning. This works by statistically linking an input to its associated response for several different examples: e.g. promoter choice for a pathway and the corresponding final production, protein sequence and its function, etc (Figs. 2 and 3). It is important to realize that the emphasis is set in predicting the response, rather than produce mechanistic understanding. In fact, the algorithm linking input and response is *not* meant to represent a mechanistic understanding of the underlying processes: for example, modeling the full process of promoters causing the expression of proteins that code enzymes which then catalyze reactions that transform metabolites and result in a predicted production. Rather, the algorithm is chosen to be as expressive as possible to be able

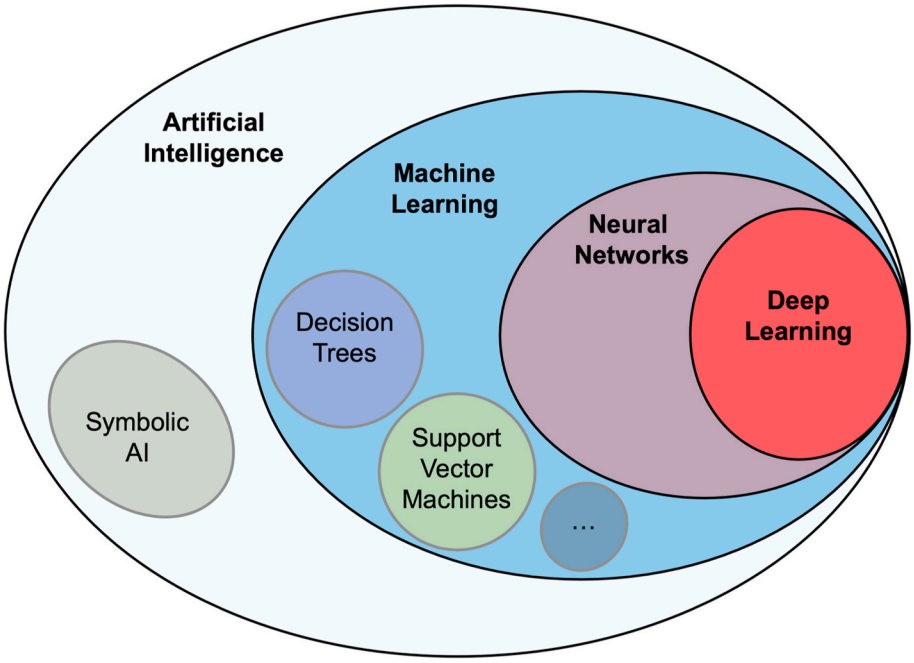


Fig. 1. Machine learning vs Artificial intelligence vs. Deep learning. Machine learning is a subdiscipline of Artificial Intelligence, which attempts to reproduce how human brains think. Symbolic AI (or Good Old Fashioned AI, or GOFAI), is a part of AI devoted to reproduce thought through symbolic representations of the world. In contrast, machine learning mimics thought using algorithms that learn a task (e.g., identify a dog) through learning from data. GOFAI was dominant in the early states of AI (50s–80s) but has now lost relative influence. Machine learning, however, is now the dominant branch of AI and focuses on improving performance through the acquisition of experience in terms of data. Among the many possible algorithmic approaches in machine learning, neural networks (Fig. 7) have become most popular since ~2010 because their performance seems not to saturate as easily as other methods (Fig. 8). Neural networks with many layers (Fig. 8) are called deep neural networks, and constitute the basis for Deep Learning.

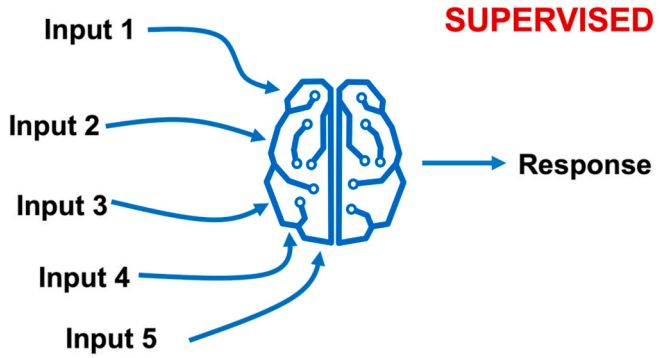


Fig. 2. Machine learning basics. Supervised Machine learning algorithms define learning in a narrow way: the ability to predict a response (e.g. the target compound production) from a set of inputs (e.g. protein concentrations for a pathway). The inputs (or features) and response (or output) can be numbers (e.g. protein concentrations) or categories (e.g. different available promoters). All supervised machine learning algorithms follow this general architecture. Because the algorithm linking input and response does not include mechanistic information, but is rather chosen to be as expressive as possible, machine learning can predict relationships between really diverse inputs and outputs: e.g., production and enzyme choice (see section 2.2.2), metabolite rate change and multiomics measurements (section 2.2.1), or protein sequence and protein function. The supervision consists in providing training data consisting of the input and the associated response. This labeling of the input data to teach the algorithm the right associations is the step that is most arduous and costly, particularly for large data sets. This has prompted AI researchers to develop methods that do not require this step (Fig. 6).

to learn any relationship between input and response. Hence, none of the biological information is encoded in the algorithm; all the biological information is provided by the training data, which must be carefully selected (supervised) so the algorithm can learn the desired relationship (promoters to production, protein sequence to function, etc.), generalize it, and be able to predict it for new inputs that were not in the training set (Fig. 3). This difference is crucial with respect to traditional metabolic engineering and microbiology, where understanding the mechanism is considered of paramount importance (see section 2.2.1 for a specific example). In machine learning, we can see the situation in which we can predict that, e.g., a given promoter choice will have the best production, but we cannot explain the metabolic mechanism that provides that optimal production (Zhang et al., 2020). This state of affairs has its pros and cons, and efforts have been made to introduce biological prior knowledge in the algorithms (see section 4).

There is a continuous interplay between the complexity of a supervised machine learning algorithm and the amount of data available to train it (Fig. 4). If the model/algorithm is not expressive enough (not enough parameters), it will be unable to describe the data accurately (underfitting). If the model displays much more parameters than data instances are available, it will just "memorize" the training data set rather than grasp the underlying general patterns required to predict new inputs (overfitting). In this case, the algorithm will produce exceedingly good results for the training set, but very poor ones for any new input that is used as a test (Figs. 3 and 4). Cross validation (Fig. 3) provides an effective way to choose the number of parameters: both overfitting and underfitting result in very poor predictions.

There are many supervised machine learning algorithms available in the public domain: linear regressions, quadratic regressions, random forest, support vector machines, neural networks, Gaussian process regressors, gradient boosting regressors (the popular library scikit-learn provides a good starting point with an extensive list and explanations (Pedregosa et al., 2011)). To give a concrete example, a classic machine learning algorithm is the decision tree, that can be used, for example, to predict which protein expression levels result in high production (Fig. 5). As can be observed, this algorithm represents a high-level abstraction of how humans are believed to think. Because no single algorithm is best for every learning task (Wolpert, 1996), a significant endeavor when applying machine learning is choosing the optimal algorithm for your problem (and its hyperparameters, see Fig. 5). Ensemble modeling is an alternative approach that sidesteps the challenge of model selection (Radivojević et al., 2020). Ensemble modeling takes the input of various different models and has them "vote" for a particular prediction. Based on their performance, a different weight is assigned to each algorithm. The examples of the random forest algorithm (Ho, 1995) or the super learner algorithm (van der Laan et al., 2007) have demonstrated that even very simple models can increase their performance significantly by using an ensemble of them (e.g., several decision trees in a random forest algorithm).

Learning without supervision also constitutes an important part of

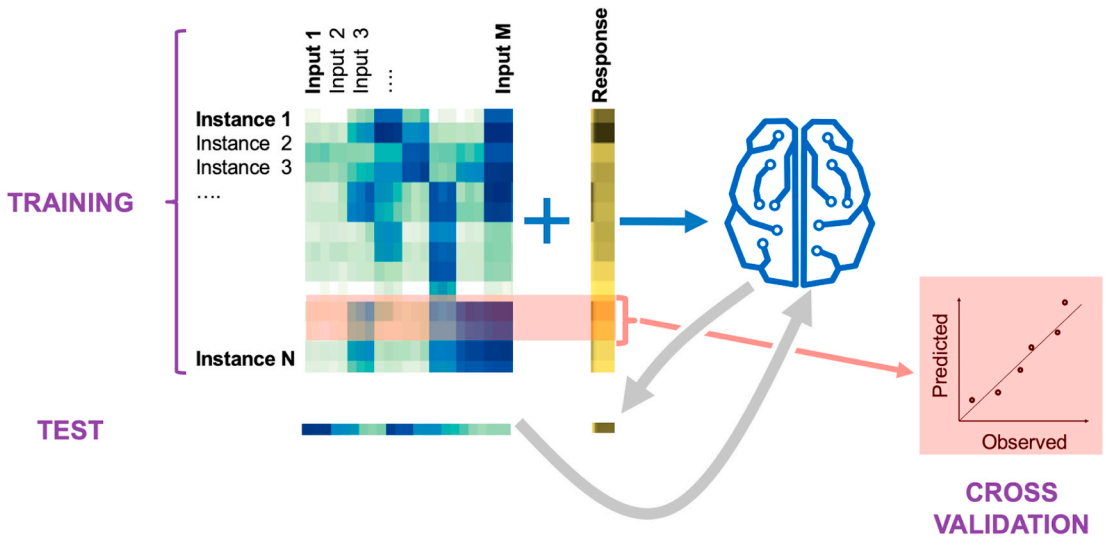


Fig. 3. Machine learning terminology. The standard workflow for supervised machine learning involves first using a *training* data set (including the inputs, or features, and the corresponding responses, or labels) to train the chosen algorithm. The training data set is composed of *instances* or examples of the inputs and response to be learnt. Instances depend on the problem to be learnt: they could be different strains and conditions (section 2.2.2 example), time points (section 2.2.1 example) or different proteins. The goal is for the algorithm to be able to predict the response for inputs that it has never seen before (i.e. were not in the training set), which is the ultimate *test* of its performance. A way to foresee how the algorithm will perform under such a test is to use only part of the training data set (all data except red overlay) for training, and then check the predictions for the remaining inputs (red overlay), to be compared with the known responses. This procedure is called *validation* and, if performed several times by randomly holding out a fraction of the training data set, it takes the name of *cross validation*. A 10-fold cross-fold validation, for example, randomly holds out 10% of the training set to test predictions for several draws. Cross validation is a good way to determine the needed algorithm complexity needed (Fig. 4). (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

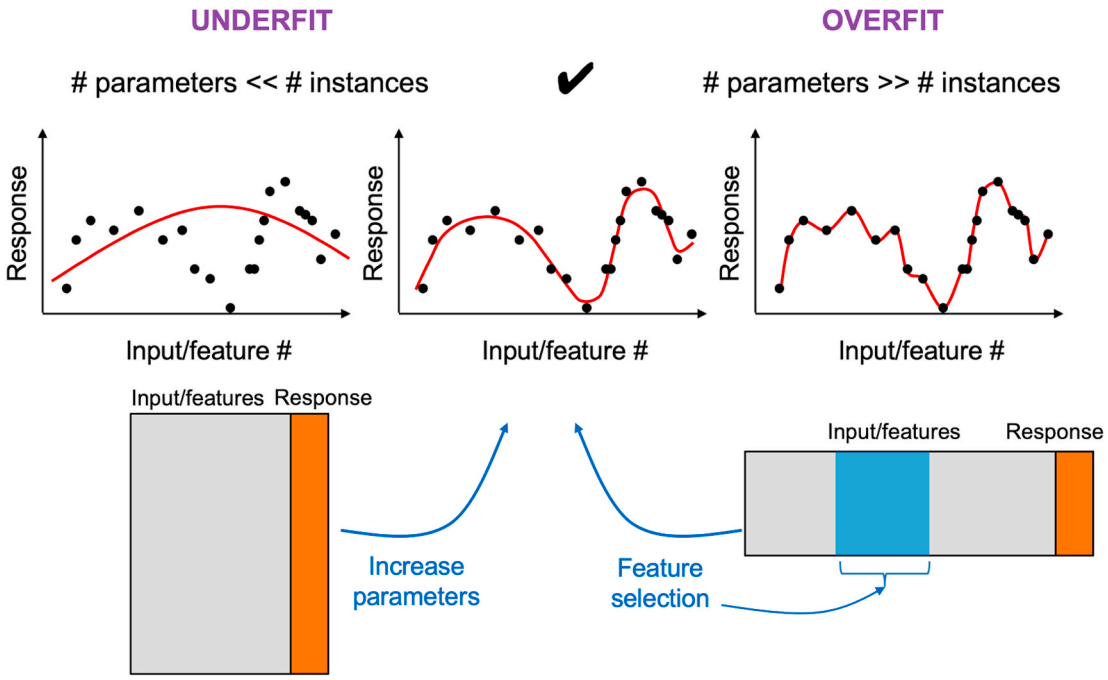


Fig. 4. Model complexity vs data availability. The number of parameters (model characteristics that can be changed to fit data, see Fig. 5) in a model provides an idea of its complexity (more parameters → more complex). If the number of parameters is much smaller than the number of instances, the model cannot hope to describe the training data (underfit model). This can happen with "long and skinny" training data: few inputs and many instances. If the number of parameters is much bigger than the instances, the model but will be unable to generalize beyond the training set. The solution for the underfitting case is straightforward: increase the complexity of the model (number of parameters). The solution for the overfitting case is reducing the model parameters. However, if the number of inputs/features is high, it may be impossible to do so. This is often the case in metabolic engineering, where omics data sets displaying tens of thousands of features are available, but only for ~100 instances ("short and fat" training data). It becomes imperative then to choose the most informative features through the feature selection methods provided by unsupervised learning (Fig. 6). This feature selection is needed to avoid the "curse of dimensionality": i.e., the amount of data needed to support results in a statistically sound fashion often grows exponentially with the dimensionality. Poor cross validation scores (Fig. 3) can help identify both overfitting and underfitting.

A

Instances	Strain ID	Protein 1 (units/cell)	Protein 2 (units/cell)	Production (Low/High)
		Input features		Response
	s1	2	8	Low
	s2	3	6	Low
	s3	9	6	Low
	s4	5	10	High
	s5	10	10	Low
	s6	5	1	Low
	s7	6	2	Low
	s8	9	1	Low
	s9	6	7	High
	s10	1	1	Low
	s11	5	4	Low
	s12	2	4	Low
	s13	6	12	High
	s14	9	3	Low
	s15	4	7	?

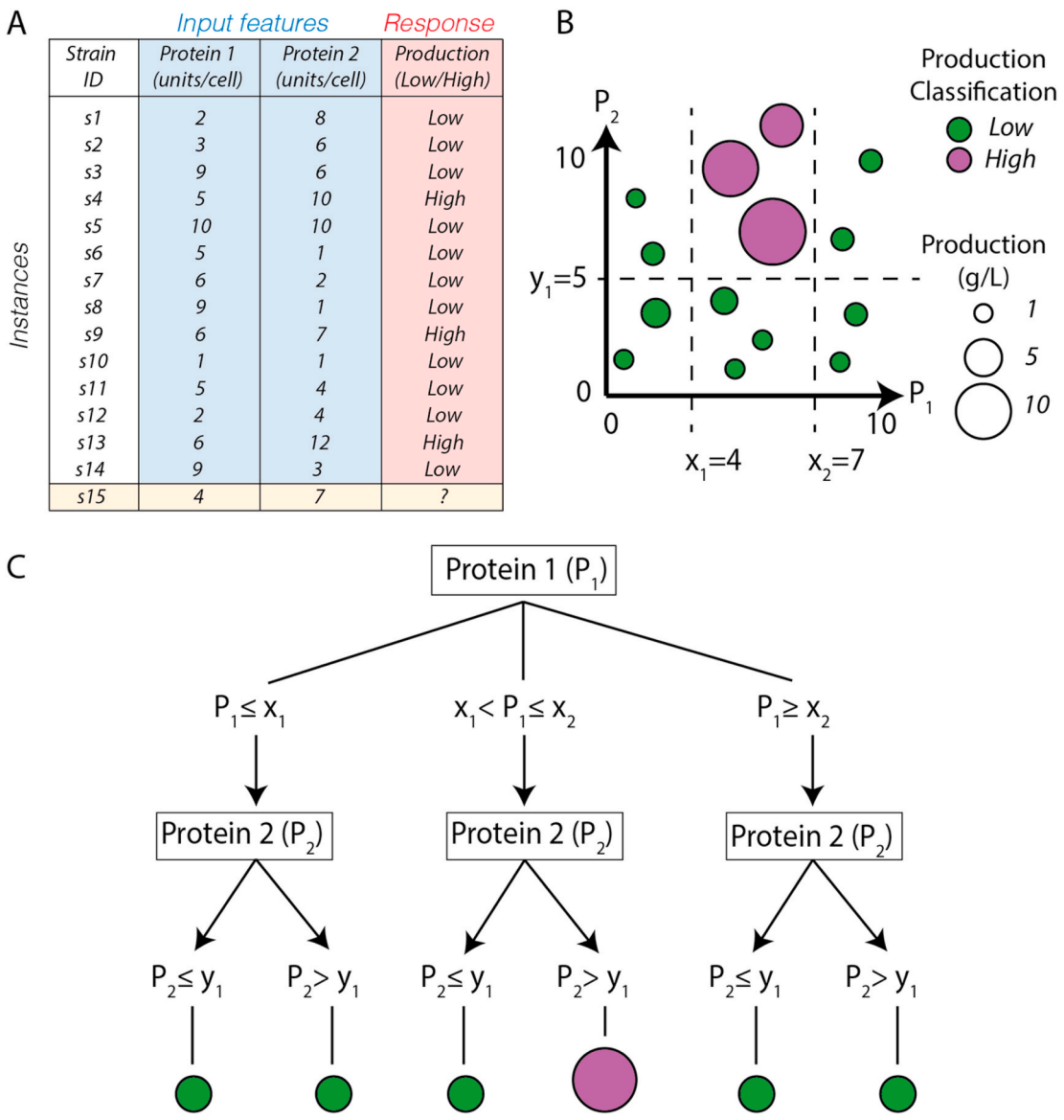


Fig. 5. Example of a supervised machine learning algorithm: a decision tree. Decision trees come from an abstracted view of how human learning works, rather than a mechanistic understanding. Decisions trees automatically build a decision "flowchart" that, in this case, predicts high or low production based on the protein expression levels. An example training data set and corresponding decision tree are shown in panels A and B, respectively, based on a set of strains (instances) and their production (response) depending on different protein expression levels (input features). Using the training data set, the algorithm decides on the optimal split points (x_1, x_2 and y_1) to predict the production based on the input features. The split points are the parameters of the algorithm: more parameters will allow the algorithm to describe more instances. The algorithm also has a number of "hyperparameters" which are set before training, including the maximum tree depth, and the minimum number of instances required to split a node, among others (see scikit learn library for more details). Decision trees form the base for one of the most popular algorithms: the random forest. The random forest algorithm is just an ensemble of decision trees.

machine learning, given the significant effort involved in creating labeled data sets. The areas of machine learning focused on this challenge are unsupervised learning and reinforcement learning. Unsupervised learning searches for patterns in a data set with no pre-existing labels, requires only minimal human supervision, and often attempts to create clusterings or representations that aid human understanding or reduce dimensionality (Fig. 6). Examples of unsupervised machine learning algorithms include Principal Component Analysis (PCA), K-means clustering (Sculley, 2010), and Single Value Decomposition (Manning et al., 2008). Familiar examples in metabolic engineering include identifying patterns in metabolomics profiles that distinguish between different types of cells: healthy vs. sick (Sajda, 2006), stressed vs. non-stressed (Luque de Castro and Priego-Capote, 2018; Mamas et al., 2011), or high-producing vs low-producing (Alonso-Gutierrez et al., 2015). Reinforcement learning represents a different paradigm regarding learning from experience that posits that humans learn not from properly labeled examples, but rather from interacting and probing their environment. Hence, the aim of reinforcement learning is to use experience and data to update an internal policy that optimizes a desired goal (Fig. 6). A prime example of this approach (Treloar et al., 2020) is controlling a bioreactor which contains a co-culture (environment), through manipulations of the concentration of auxotrophic nutrients flowing into the reactor (actions), and informed by the relative abundances (measurements), to ensure a specified co-culture composition (goal). Perhaps the most known example of reinforcement learning are the Hidden Markov Models (HMMs) that are commonly used to annotate genes and align sequences (Yoon, 2009). Reinforcement learning has also been applied to suggest pathways for specific molecules (Koch et al., 2020) or molecules that fit desired properties (Popova et al., 2018), as well as to optimize large-scale bioreactor fermentations using online

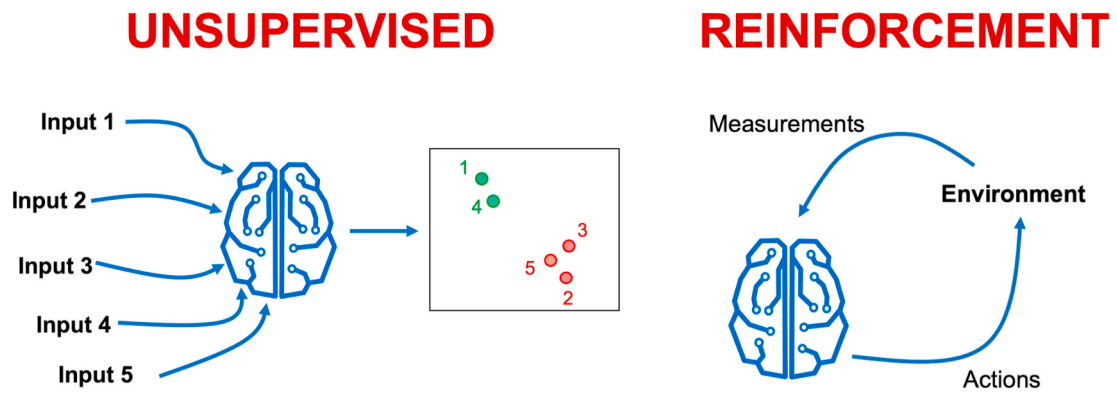
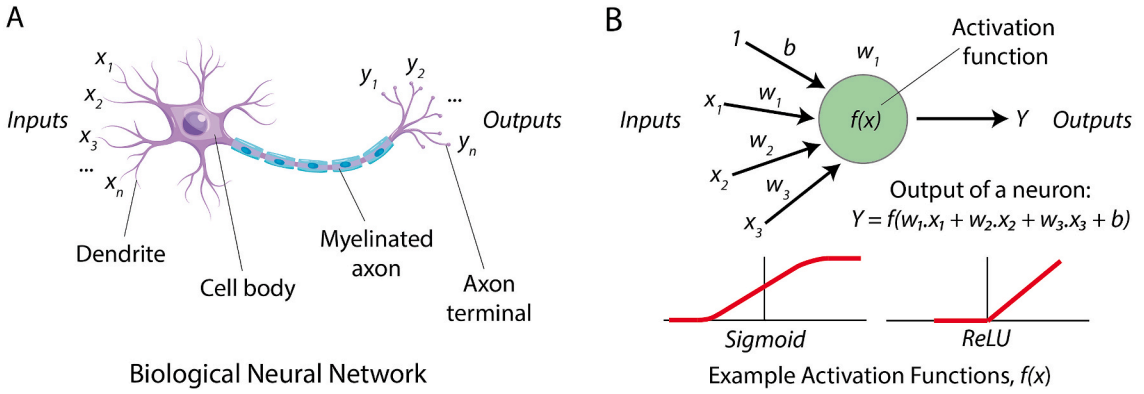


Fig. 6. Other types of machine learning that do not require labeled data. Unsupervised methods and reinforcement learning were created to avoid the cumbersome process of labeling data for supervised methods. Unsupervised machine learning methods often search for patterns that aid human understanding or reduce dimensionality (e.g. PCA). For example, in this case the algorithm projected the five inputs (e.g. metabolomics data) into a two dimensional plane that groups them according to similarity. This type of dimensionality reduction can be very useful for feature selection (Fig. 4). Reinforcement learning methods attempt to achieve a goal through a continuous interaction with an environment from which they learn through a variety of measurements, and on which they can act through a menu of actions. The result of the actions as viewed by the measurements is used to iteratively update an internal policy that dictates future actions.



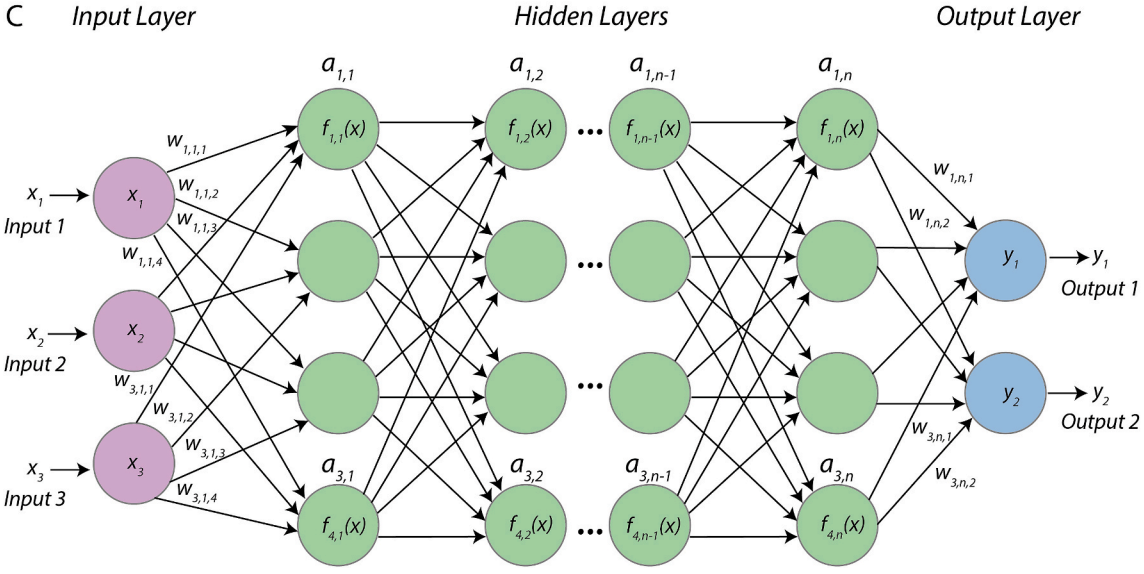


Fig. 7. Artificial neural networks are a particular type of machine learning algorithms that loosely mimic how neurons work (Fig. 1). Neurons are modeled as having a set of inputs (dendrites) and a single long axon that serves as output (A). Artificial neural network cells mimic that: several outputs combined linearly and a non-linear output (B). The output is combined to other cell inputs, creating an artificial neural network (ANN). Here we see a fully connected network where all cells from each layer are connected to all cells in the next layer (C). This type of architecture results in many parameters (w_{ij} and b_j), which requires large amounts of data to determine (Fig. 4). Deep neural networks are ANNs with many layers (Fig. 8). Given the original biological origin of ANNs, there is a significant interest in the AI field in obtaining further inspiration from biomimicry.

continuous process data (see section 3.3). However, there is still generally a dearth of reinforcement learning examples in metabolic engineering, which represents an opportunity for this type of machine learning.

Deep learning (DL, Fig. 7 (LeCun et al., 2015)) is a specific type of machine learning algorithm that has been particularly successful in the past decade (Fig. 8). This algorithm has been shown to improve performance with the amount of training data when other methods plateau. Deep learning is based in Artificial Neural Networks, which attempt to mimic how neurons work (Fig. 7). In the last decade, deep learning has been the basis of the most celebrated AI achievements. However, compared to more classical machine learning methods, deep learning generally requires far larger amounts of data for training: 10,000s or millions of instances, as opposed to hundreds or thousands (although that depends on the number of inputs, see section 2.2.2). The reason for these large data sets hunger is that deep neural networks can include thousands to millions of parameters, which need to be determined from the data (see Fig. 4). In metabolic engineering, the use of deep learning has been sparse for this reason: the data sets tend to be small (<100 instances), with the notable exception of sequence data. Deep learning has been most useful with sequence data: e.g., to predict protein function (Ryu et al., 2019), or translation initiation sites (Clauwaert et al., 2019) (see section 3.0). However, this is expected to change as more high-throughput methods to characterize cellular components become available, provided that data are structured consistently and stored appropriately (see section 2.4). Indeed, techniques to generate high quality omics data are improving rapidly and the cost per sample is decreasing (Stephens et al., 2015), so application of deep learning to metabolic engineering might become commonplace soon.

2.2. A couple of illustrative examples of machine learning in metabolic engineering

We will now illustrate how machine learning algorithms work through two different applications that elucidate particularly important points: predicting the kinetics of a metabolic network, and optimizing cell-free butanol production. We have focused on these examples because we believe they most relate to the day-to-day activities of metabolic engineers: leveraging omics data and improving production.

2.2.1. Kinetic learning: relearning Michaelis-Menten dynamics through machine learning

Our first example uses machine learning to tackle a commonly encountered problem in bioengineering: predicting the kinetics of a metabolic pathway. Predicting pathway dynamics can enable a much more efficient pathway design by allowing us to foresee in advance which pathway designs will meet our specifications (e.g., titers, rates and yields). Classic kinetic models predict the rate of change of a given metabolite based on an explicit functional relationship between substrate/product concentrations (metabolites) and enzymes (protein abundance, substrate affinity, maximum substrate turnover rate). Michaelis-Menten kinetic models (Costa et al., 2010; Heinrich and Schuster, 1996) have historically been the most common choice. In reality, the true functional relationship between metabolites and enzymes are typically unknown for most reactions due to gaps in our understanding of the mechanisms involved, resulting in poor prediction capabilities.

Costello et al. (Costello and Martin, 2018) showed that supervised machine learning (Fig. 2) can offer an alternative approach, where the relationship between metabolites and enzymes can be directly "learnt" from time series of protein and metabolite concentration data. In a sense, this approach involves relearning the equivalent of Michaelis-Menten based purely on data. This is a prime example of how a machine learning approach ignores mechanism in favor of predicting power: there is no intention that the function predicting metabolite change rate from proteins and metabolites describes a mechanism, but it offers the best prediction of the final limonene/isopentenol, which is what we require for our engineering. In this case, the *inputs* (Fig. 3) were the exogenous pathway protein and metabolite concentrations, and the *response* was the rate of change of the metabolite. The *instances* involved each of the time points for which the metabolite rate of changes was learnt.

This approach outperformed a classic kinetic model in predictive power using very little data: only three time series of protein and metabolite measurements of 7 time points each (for two different pathways). While it would be desirable to have hundreds of time point measurements, the high cost and time associated with performing multi-omic experiments typically constrains data sets to less than 10 time points/samples, which is too sparse for training accurate models. Critical to its success, hence, was the use of data augmentation to increase the number of available instances from the initial 7 time points to the final 200 used for learning. Data augmentation simulates additional instances by modifying or interpolating actual data. In this case, data augmentation involved first smoothing the data (via a Savitzky-Golay filter) and then interpolating new data points from the fitted curve. This augmentation scheme only assumed continuity and smoothness between time points, but provided sufficient data to train a machine learning model using data from only 2 time series that accurately predict pathway dynamics of the "unseen" third strain. The final predictions of metabolite concentrations for the exogenous pathways, although not perfect by any measure, were more accurate than equivalent predictions by a hand-crafted kinetic model. More importantly, while the kinetic model took weeks to produce through arduous literature search, the kinetic learning approach can be systematically applied to any pathway, product and host with no extra overhead.

An opportunity to improve the machine learning model predictions of Costello et al. would of course be to collect more data, but deciding which data to collect is not always clear. For example, instead of using protein and metabolite data only from the exogenous pathways as input features, protein and metabolite measurements from the full host metabolism could be added (surely, host metabolic effects like ATP supply must be relevant). However, using these extra data would not necessarily improve machine learning predictions. This is because many machine learning algorithms suffer from the "curse of dimensionality": that is, the amount of data needed to support results in a statistically sound fashion often grows exponentially with the dimensionality of the input (Fig. 4). Hence, machine learning algorithms may struggle to learn from data sets that have many measurements or "input features" (columns), but few instances (rows). Adding host proteins and metabolites will increase the number of inputs without increasing the number of instances. Unfortunately, most multi-omic data sets used in metabolic engineering fit this description, containing more than 5000 measurements (e.g. proteins or metabolites abundances), but only tens to hundreds of instances (e.g. different time points, strains, or growth conditions, depending on what your algorithm is attempting to learn) (Fig. 4). Therefore, collecting as many instances as possible should be emphasized early on during experimental design (see section 2.4).

In the absence of being able to generate more data, algorithms that reduce the number of input features to the most important ones can be performed, a process known as feature selection. Feature selection (Pedregosa et al., 2011) was used in Costello et al. (Costello and Martin, 2018) to identify a subset of the input features based on their contribution to the model's error. This, more limited, curated set of features was then used to predict metabolite dynamics. The idea behind this is to remove non-informative or redundant input features from the model. An additional approach used was dimensionality reduction, where "synthetic features" are created that transform the original input features into fewer ones (or "lower dimensions") based on their contribution to explaining the data's variability (for example, via principal component analysis). Similar to feature selection, these algorithms simplify the data set in order to better fit a machine learning model. These approaches were integrated into a machine learning pipeline using the tree-based

pipeline optimization tool (TPOT) (Olson et al., 2016; Olson and Moore, 2019), which automatically selected the best combination of feature preprocessing steps and machine learning models from the scikit-learn library (Pedregosa et al., 2011) to maximize prediction performance.

2.2.2. Artificial neural networks to improve butanol production in cell-free systems

Our second example involves using deep neural networks to optimize cell-free butanol production (Karim et al., 2020). Here, the authors provide an example of how machine learning can accelerate the design-build-test-learn (DBTL) cycles used in metabolic engineering (Nielsen and Keasling, 2016), by effectively guiding pathway design. In this study, the authors optimized a six-step pathway for producing n-butanol, an important solvent and drop-in biofuel, using a cell-free prototyping approach (iPROBE). iPROBE reduces the overall time to build pathways from weeks or months to a few days (around five in this case), providing the quick turnaround and large numbers of enzyme combinations that can enable successful use of machine learning. Several pathway variants were constructed in vitro and scored based on their measured butanol production through a TREE score which combines titer, rate, and enzyme expression. The challenge, however, lies in analyzing the sheer number of pathway combination possibilities. Testing only six homologs for the first four pathway steps at 3 different enzyme concentrations would result in 314,928 pathway combinations (strain genotypes). Even with the increased turnover provided by the cell-free approach, it would take years for typical analytical pipelines to exhaustively test the landscape of possible combinations. Therefore, a data-driven design-of-experiments approach was implemented using neural networks to predict optimized pathway designs (homolog sets and enzyme ratios) from an initial data set that could subsequently be tested. In this case the *input* for the neural network was the enzyme homologs used for each of the reaction steps and their corresponding concentrations. The *response* was the TREE score, and each *instance* was a pathway design.

The pathways predicted from the neural network model were able to improve butanol production scores over fourfold (~2.5 times higher titer, 58% increase in rate) compared to the base-case pathway. An initial data set of 120 instances (pathway designs) was used to train and test different neural network architectures consisting of 5–15 fully connected hidden layers and 5 to 15 nodes per layer. Genetic algorithms were used to suggest combinations of network architectures, and ten-fold cross validation was used to select the best. Once the model was built, the authors used a nonlinear optimization algorithm (Nelder-Mead simplex) to recommend pathway designs that optimized butanol production through the maximization of the TREE score. These machine learning recommendations resulted in 5 of the 6 top performing pathways, and outperformed 18 expert determined pathways selected based on prior knowledge, demonstrating the power of a data-driven design approach for cases in which design choices are numerous.

While the study by Karim et al. only reported 1 DBTL cycle, multiple cycles would have likely resulted in even better production pathways, and also provided more data instances for model training. Indeed, the neural network of 5–15 hidden layers developed by Karim et al. was relatively small compared to state-of-the-art deep neural networks, but this design was limited by having only 120 instances (pathway designs) to train on. If more data were to become available through more DBTL cycles, the neural network could have been made more complex by expanding its depth (hundreds of hidden layers), which would improve prediction performance (Fig. 4). This improved performance, however, comes at a cost: as the number of layers increases, the time to train the network (i.e. learning model weights and parameters) increases considerably. Moreover, the dense hidden layers of deep neural networks render them very difficult to interpret and infer possible mechanisms from. Hence a significant research thrust in machine learning involves new approaches to make models "explainable" (see Section 5.3) (Gunning, 2016; Gunning et al., 2019). The use of only 1–2 DBTL cycles seems to be the most common case in published projects (Denby et al., 2018; Alonso-Gutierrez et al., 2015; Opgenorth et al., 2019; Zhang et al., 2020). In our experience, this happens not because more DBTL cycles are not expected to be useful, but because results from a single DBTL cycle are often enough for a publication. Often, in the academic world, there is little incentive (or resources) to continue further.

2.3. Requirements for machine learning in metabolic engineering

Here we provide a practical guide on the immediate prerequisites to applying machine learning to metabolic engineering, in the next section we will discuss some practical considerations for experimental design once the machine learning project is in progress, and, in section 5.1, we discuss long term hurdles for the development of the discipline as a whole. In essence, four requirements need to be aligned for a successful application: data, algorithms, computing power and an interdisciplinary environment. Each of them is critical for a real impact.

Data needs to be abundant, non-sparse, high quality, and well organized. Training data needs to be abundant because machine learning algorithms depend critically on training data to be predictive. There is no prior biological knowledge embedded in them. In general, the more training data, the more accurate the algorithm predictions will be. Data augmentation (see section 2.2.1) can certainly help, and should be routinely used in metabolic engineering due to the scarcity of large data sets, but it is no substitute for experimental data. There is, however, no way to know a priori how much data will be enough. Different problems present different difficulty levels to being "learnt" (Radivojević et al., 2020), and this difficulty level can only be assessed empirically. A scaling plot of predictive accuracy vs. instances can be very helpful in this regard. Training data can be abundant but still sparse, depending on the phase space (Fig. 9) considered. A total of a hundred instances can be enough if only two input features are considered, or completely insufficient if a thousand input features are considered. The "curse of dimensionality" implies that the amount of data needed to support results in a statistically sound and reliable fashion often grows exponentially with the dimensionality (Fig. 4). The data must be high-quality in the sense that it must avoid biases due to inconsistent protocols and provide quantification for repeatability (see section 2.4). Both goals can be systematically achieved through automation (see section 5.2). Data needs to be well organized, following standards and ontologies, and must include the corresponding metadata (see section 2.4). The alternative is that data analysts will spend 50–80% of their effort organizing the data and metadata for analysis, mining their efforts (Lohr, 2014). Since data analysts might be the most effective effort multiplier in your team (Nielsen and Keasling, 2016), and possibly the most expensive (Metz, 2018), it is very useful to optimize their effort.

While there are many machine learning algorithms to choose from (Fig. 10), there is no clear best algorithm for every situation. Indeed there is a famous theorem (the no free lunch theorem, NFLT) that proves (under some conditions) that no single algorithm is most effective for every type of problem (Wolpert, 1996). While the utility of the NFLT for machine learning has been cast in doubt (Giraud-Carrier and Provost, 2005), the standard approach remains to try as many algorithms as possible and compare their results. In this effort, it is very useful to count on libraries that collect a large variety of algorithms and have standardized input, output and other standard procedures (e.g. cross-validation). The most popular among them is, without a doubt, scikit-learn (Pedregosa et al., 2011), a python library that comprises a very wide selection of machine learning methods, is well documented, and easy to use (Fig. 10). These features combined with its open source nature, and its compatibility with Jupyter notebooks (Kluyver, 2016), which facilitate reproducibility and communication, make it our top recommendation for beginners. Furthermore, the open source nature and wide use of scikit-learn means that there are several tools that leverage it to combine and test methods. Tree-based pipeline optimization tool (TPOT), for example, automatically combines all the

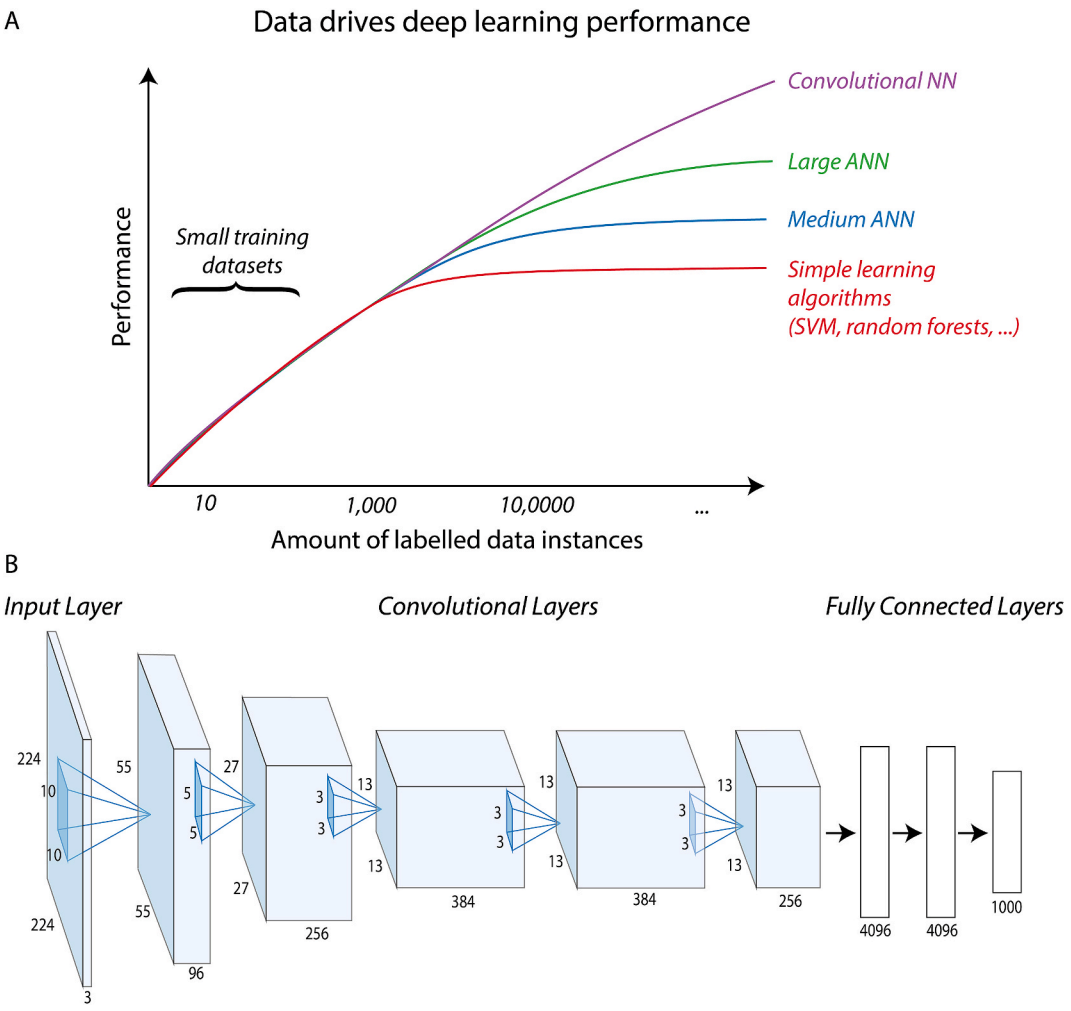


Fig. 8. Deep Learning involves artificial neural networks many layers deep. (A) Deep learning methods have been shown to improve performance with the amount of training data when other methods plateau (B) This is Alexnet, one of the first ANNs that leveraged the network depth to improve performance and win the ImageNet image classification contest in 2012. Deep networks lower the amount of parameters by sparsely using fully connected layers, which require many parameters. The first five layers in Alexnet are convolutional layers (Rawat and Wang, 2017), which only take input from a limited number of cells in the previous layer. Many architectures are possible for deep learning, and finding the optimal one is more of an art than a science. See Lecun et al. (LeCun et al., 2015) for more details.

available algorithms and preprocessing steps in scikit-learn to choose the best option (Olson et al., 2016). Another example is the Automated Recommendation Tool (ART), which leverages scikit-learn, ensemble modeling, and bayesian inference to provide uncertainty quantification for predictions (Radivojević et al., 2020). A proprietary alternative is to use Matlab, for which a machine learning toolbox is available (Ciaburro, 2017), with possible educational discounts. For artificial neural networks, the best supported (and free) frameworks are TensorFlow and Pytorch, backed by Google and Facebook respectively. Keras, a framework focused on providing a simple interface for neural networks, is now the official high-level front-end for TensorFlow (Géron, 2019). Keras has its own hyperparameter tuner, Keras Tuner (O'Malley et al., 2019), and an extremely simple interface for DL with Keras and TensorFlow, AutoKeras (Jin et al., 2019).

Computation is another key element, particularly for large amounts of data. Whereas the libraries above (Scikit-learn, Matlab toolbox, Tensorflow, Pytorch) can be run on a standard laptop (e.g. 2018 Macbook Pro, 3.5 Ghz Intel Core i7, 16 GB RAM), as more training data is added this may be insufficient. This is particularly the case for deep neural networks using Tensorflow or Pytorch, which will benefit from the parallelization obtained through Graphics Processing Units (GPUs). The need to scale up all these Python frameworks for high performance computing (HPC) or deployment on cloud computing environments (e.g. Amazon EC2, Microsoft Azure, and Google's Cloud Platform) has promoted the development of several parallel and distributed computing backends for data analysis and machine learning, such as Ray, Spark, and Dask (Rocklin, 2015). Furthermore, as the general applicability of AI has become more evident, new processor architectures are being created specifically for neural network machine learning, including Google's Tensor Processing Unit (TPU), Nvidia's V100 and A100, Graphcore's Intelligence Processing Unit (IPU), and a variety of FPGA-based solutions.

Since very few people master both machine learning and metabolic engineering, interdisciplinary collaborations are truly necessary. Machine learning practitioners and metabolic engineers are trained very differently, however, and this can produce significant friction (see section 5.1). Both disciplines profess different cultures, which are reflected in how they solve problems, but also which problems are prioritized. It is, hence, very important to foster an inclusive work environment that integrates and values contributors with very different skills, and does not penalize knowledge gaps. It is also important to be very clear about the interfaces: which exchanges (e.g., data, designs, predictions) are expected, and when, in order for both sides to be effective.

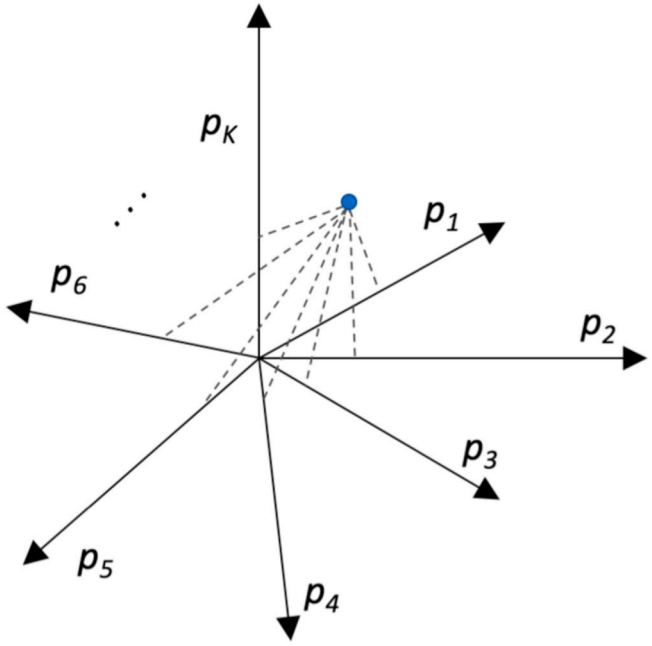


Fig. 9. The input phase space. It is a multidimensional space composed by all possible configurations of a system. Each axis represents, for example, the expression level for a protein p_i (or any other variable such as nucleobase for each position, transcription level, promoter, fermentation condition etc.) required to specify the input state of a system. Hence, a point in the space (blue) corresponds to a unique possible state of the system, consisting of e.g. expression levels for each protein in the pathway considered. The volume of this space, representing all possible states, grows exponentially with the number of variables. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

2.4. Practical considerations for implementing machine learning

As in the case of a genetic selection or screen, machine learning requires careful experimental planning to make it effective. An experimental design that ignores its basic assumptions (e.g., instances are independent and identically distributed) will result in a random walk over possible designs with the same (or even worse) results as a trial-and-error approach.

Here, we offer a succinct list of recommendations to consider when planning to use machine learning to guide bioengineering:

- **Choose the right objective/response.** When a response for the algorithm is chosen, you are entering a Faustian bargain with your algorithm: it will try to optimize it to the detriment of everything else (Riley 2019) . For example, setting final titer as the response might provide high titers in the end for a production strain, but at rates so slow that the result is of little practical use. In the case of Karim et al., (see section 2.2.2), the response was a carefully selected mixture of titer, rate, and enzyme expression precisely for this reason. Deciding on the right response is a bit of an art, and less trivial than often assumed. Be careful what you ask the algorithm for, because you may get it!
- **Choose inputs that truly predict your response.** Performing small, directed experiments in the lab to verify that the response of interest (e.g. a phenotype) is affected by a given input (e.g. a treatment) can save a significant amount of time and headaches later in the DBTL cycle, by limiting the number of inputs (and the overall complexity of the model) to terms that matter. Omitting this step might give rise to a frustrating chase of a red herring in the form of statistical noise, or cause serious challenges to the interpretability of the model.
- **Choose actionable inputs that can be measured.** The machine learning process will require you to change your inputs in order to achieve the desired goal (e.g. increase production). Hence, these inputs need to be experiment variables that can be easily manipulated. Since you will need to assess whether you indeed reached the recommended targets, it is highly desirable that these inputs can be

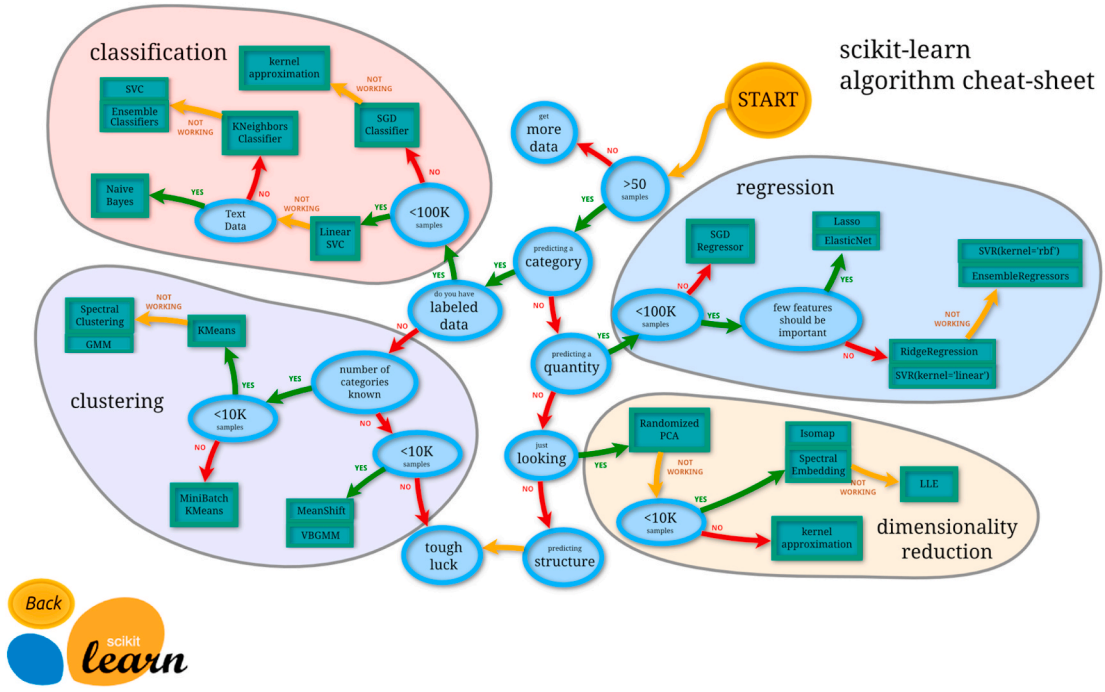


Fig. 10. The scikit-learn library is our top recommendation for machine learning beginners. This library provides a wide range of supervised and unsupervised algorithms, as well as practical advice on how to choose among them. Image obtained from scikit-learn github repository (https://github.com/scikit-learn/scikit-learn, Pedregosa et al., 2011).

easily measured. For example, it is generally better to use as inputs promoter (Zhang et al., 2020) or enzyme choices (Karim et al., 2020), rather than protein levels (Opgenorth et al., 2019). Promoter or enzyme choices are entirely under the metabolic engineer's control, and their effects on expression may be verified via sequencing; whereas certain target protein levels may be difficult to reach, and usually require specialized mass spectrometry methods to verify.

- **Choose very carefully how many experiment variables you would like to explore.** Choosing too many variables (i.e. input features, Fig. 3) can make the corresponding phase space too large for machine learning to explore in a reasonable amount of DBTL cycles. Choosing too few variables might mean missing important system configurations (e.g. if protein X is not chosen and it needs to be downregulated to improve production, it will be impossible to find the optimum). As a *very crude* rule of thumb, you should budget for around at least 100 instances per 5–10 variables. This, of course, depends on the difficulty presented by the problem being learnt: more difficult problems will need more instances per variable, whereas easier problems will require less instances per variable.
- **Verify that your experiment variables can be independently acted upon.** Whole-operonic effects can make this unexpectedly difficult (Opgenorth et al., 2019). For example, if recommendations require protein A concentration to be increased three-fold and protein B to be decreased by a factor of two to improve production, but a strong promoter for protein A also produces an increase in protein B, it will be difficult to reach the target protein profile. Hence, modular pathway designs (Boock et al., 2015) that ensure that the full input phase space can be fully explored are highly recommended. Systematic part characterization involving large promoter libraries with a variety of tested relative strengths are a fundamental tool in this endeavor.
- **Design your experiment to start with ~100 instances for the initial DBTL cycle.** Although there are examples of success stories with less than a hundred instances as starting points (Radivojević et al., 2020), this outcome cannot be guaranteed. Actual success depends on the complexity of the problem (Radivojević et al., 2020), and this complexity can only be gauged by testing predictive accuracy as data sets increase. By starting with ~100 instances, one ensures some progress even if predictions are not accurate: this amount of instances goes a long way to ensure statistical convergence. The alternative is a non-predictive model and little understanding whether the problem is lack of data (instances), or other design problems (Opgenorth et al., 2019). Consider automating as much of your process as possible so as to guarantee enough instances. This automation may seem an unnecessary hassle, but it will pay off in the long run.
- **Sample the initial phase space as widely as possible.** Ensure that you cover wide ranges for both input and response variables. Strive to include both bad (e.g. low production) and intermediate results as well as good ones (e.g. high production), since this is the only way that the algorithms can learn to distinguish the inputs needed to reach any of these regimes. The Latin Hypercube (McKay et al., 1979) is a good choice to choose starting points, but other options are also available.
- **Consider uncertainty, as well as predicted response, when choosing next steps.** As the need to quantify prediction uncertainty becomes more recognized in the biological sciences (Begoli et al., 2019), more algorithms provide it along with response predictions (Radivojević et al., 2020). Using this information can improve the whole process. Choose some recommendations with the lowest possible uncertainty even if the predicted outcome is not so desirable (e.g. low production), so as to establish trust in the approach (see sociological hurdles in section 5.1). Choose some recommendations with large uncertainty even if the predicted outcome is not desirable so as not to miss unexpected opportunities. In addition, to obtain an empirical view of how uncertainty in the data affects the accuracy of predictions, it may be instructive to create simulated, in silico "ground truth" data sets displaying different levels of noise in order to test the performance of the machine learning algorithm.
- **Avoid biases created through inconsistent protocols and beware of hidden variables.** Machine learning algorithms learn to map an input to a response (Fig. 3). If different DBTL cycles produce different results for reasons that are not reflected in the input (hidden variables (Riley 2019)), the algorithms will provide poor predictions. Such uncontrolled variables can easily arise in biological data due to lab temperature or climate fluctuations, reagent batch differences, undetected culture mutations, "edge effects" in plate-based assays, and equipment drift. These effects should be assessed and eliminated as part of the experimental design, and is one of the key topics of communication for bench and computational scientists to empower downstream data analysis and predictions. Machine learning can also help by performing simple checks: if an algorithm can predict which well or batch sample the data came from, that means they unduly influence the response. Lack of repeatability is the main stumbling block of machine learning.
- **Add experimental controls to test for repeatability.** Since ensuring repeatability is among the top requirements for machine learning to be successful, it is important to test and quantify it often. Batch, instrument, and operator effects are often the first principal component of data. These effects can be detected by including a few controls of known response in every experiment (e.g., 2–3 base strains in every DBTL cycle). While this approach consumes valuable analytical resources, it ensures that the data can be trusted and does not need to be discarded, saving substantial labor during modeling and analysis.
- **Plan for several DBTL cycles.** Machine learning algorithms shine when they can dynamically probe your system, since they are designed to learn from data interactively. While results can be obtained using two DBTL cycles, they are not comparable to what >5 cycles can provide (Radivojević et al., 2020). If only a limited budget of, e.g. 100 instances, is available, it is better to start with a strong first cycle and several weaker ones (e.g. 40 instances for cycle 1, then six 10 instance cycles) than the usual two DBTL cycle study (e.g. 60 instances for first cycle, 40 instances for the second one).
- **Standardize your data and metadata.** Taking machine learning for metabolic engineering seriously requires large amounts of high quality data. Hence, it is advisable to store it in a standardized manner. There are a variety of data repositories available for this purpose: e.g., the Experiment Data Depot (Morrell et al., 2017), the Inventory of Composable Elements (ICE) (Ham et al., 2012), DICOM-SB (Sainz de Murieta et al., 2016), SynBioHub (McLaughlin et al., 2018), ProteomeXchange (Vizcaíno et al., 2014), MetaboLights (Haug et al., 2013), BioGraph-IIn (Gonzalez-Beltran et al., 2013), the Nature Scientific Data journal ("Open for business," 2017), to name a few. Moreover, a labeled data set of high quality is a significant resource for the community, and is more likely to be cited.
- **Be careful about how you split your data for cross-validation.** Cross-validation of your model (Fig. 3), assumes data sets are independent and identically distributed (iid). This assumption is basic for machine learning, and presumes that both validation and training sets stem from the same generative processes and have no memory of past generated samples. However, it can be violated in practice due to temporal effects on biological systems or group effects during sample processing (Riley 2019). In these cases, alternatives to random splitting need to be considered. Sheridan (2013), for example, showed that randomly splitting compound libraries used for drug discovery overestimated their model's ability to successfully predict drug candidates. The reason for this difference is that compounds added to the public record at particular dates shared higher structural similarity, resulting in models that had already "seen" compounds in the test set when randomly split. Similar considerations need to be made when sample generation occurs in a biased

manner, which is quite common in biological experiments. For example, "batch effects" can be avoided by splitting the data first by group (e.g. each batch) to ensure the same group is not represented in both testing and training sets (see scikit-learn group k-fold). Do only worry about this effect if you have a large data set (>100 instances).

Perhaps the best way to get familiar with machine learning, and its potential and limitations, is to experiment with it in a tutorial. The recently published Automated Recommendation Tool (Radivojević et al., 2020) includes three synthetic data sets, three real data sets and a software package that can be used for this purpose. Furthermore, some of these cases are explained in detail in several Jupyter notebooks contained in the github repository (https://github.com/JBEI/ART/tree/master/notebooks), and can be used as tutorials.

3. Applications of machine learning to metabolic engineering

Although application of machine learning in metabolic engineering is nascent, early studies have already shown its potential use for accelerating bioengineering. Here, we highlight examples where machine learning is being used to improve different stages of the metabolic engineering development cycle: gene annotation and pathway design, pathway optimization, pathway building, performance testing, and production scale-up (Table 1). We focus on prime examples that best epitomize the potential of machine learning in metabolic engineering, rather than an exhaustive list of applications. The reason for this decision is that this list is quickly growing and might be outdated soon, and there are recent reviews on the topic that provide that information (Kim et al., 2019; Presnell and Alper, 2019; Volk et al., 2020). We also discuss key challenges and opportunities when applying machine learning for metabolic engineering, with particular focus on practices that could formalize data-driven approaches.

3.1. Machine learning for design

The goal of metabolic engineering design is to develop DNA parts and assembly instructions to synthesize metabolic pathways and produce a desired molecule (Nielsen and Keasling, 2016; Woolston et al., 2013). This requires completion of several tasks, including gene annotation, pathway reconstruction and design, as well as metabolic flux optimization, which currently rely heavily on domain expertise and enjoy little standardization (Nielsen and Keasling, 2016). Application of machine learning can improve the accuracy and speed of these tasks, offering a standardized approach that fully leverages experimental data.

3.1.1. Pathway reconstruction and design

Locating and annotating protein encoding genes in a genome sequence is essential for metabolic pathway reconstruction and design. This is conventionally done bioinformatically, for example using Hidden Markov Models (HMMs) (Finn et al., 2011; Kelley et al., 2012; Yoon, 2009). Initially, genes are identified in a genome by searching for known protein coding signatures (e.g. Shine-Dalgarno sequences), and this is followed by annotation based on sequence homology searches against a database of previously characterized proteins. More recently, however, deep learning approaches have been used to identify and functionally annotate protein sequences in genomes by leveraging large high-quality experimental data sets (Armenteros et al., 2019; Clauwaert et al., 2019; Ryu et al., 2019). DeepRibo, for example, uses high-throughput ribosome profiling coverage signals and candidate open reading frame sequences (input features) to train deep neural networks to delineate expressed open reading frames (response is part of predicted ORF or not for every nucleotide) (Clauwaert et al., 2019). This approach showed more robust performance compared to a similar tool, REPARATION (Ndah et al., 2017), that uses a random forest classifier instead of deep neural networks. DeepRibo also improved prediction of protein coding sequences in different bacteria (e.g. *Escherichia coli* and *Streptomyces coelicolor*) compared to RefSeq annotations, including higher identification of novel small open reading frames commonly missed by sequence alignment algorithms. Another example is DeepEC, which takes a protein sequence as input and predicts enzyme commission (EC) numbers as output with high precision and throughput using deep neural networks (Ryu et al., 2019). A data set containing 1,388,606 expert curated reference protein sequences and 4669 enzyme commission numbers (Swiss-Prot (Bairoch and Apweiler, 2000) and TrEMBL (UniProt Consortium, 2015) data sets) was used to train the deep neural networks, which improved EC number prediction accuracy and speed compared to 5 alternative EC number predictions tools, including CatFam (Yu et al., 2009), DETECT v2 (Nursimulu et al., 2018), ECPred (Dalkiran et al., 2018), EFICAz2.5 (Kumar and Skolnick, 2012), and PRIAM (Claudel-Renard et al., 2003). DeepEC was also shown to be more sensitive in predicting the effects of protein sequence domain and binding site mutations compared to these tools, which could improve the accuracy of annotating homologous proteins that have mutations with previously unknown effects on function (e.g. from metagenomic data sets).

The design of metabolic pathways involves identifying a series of chemical reactions that produce a desired product from a starting substrate, and selecting different enzymes that catalyze each reaction. While nature has evolved many pathways for producing diverse molecules, the known and characterized biochemical pathways can still be insufficient to produce certain molecules of interest, especially non-natural compounds or secondary metabolites. Therefore, retrosynthesis methods that start with a desired chemical and suggest a set of chemical reactions that could produce it from cellular metabolite precursors are being pursued to design new metabolic pathways (Lin et al., 2019; Lee et al., 2019). The latest and most sophisticated of these methods use generalized reaction rules to describe possible biochemical transformations (Delépine et al., 2018; Kumar et al., 2018). However, the number of possible reaction combinations is intractable since it grows combinatorially with the number of reactions. Choosing the right reaction combination is a non-trivial problem, which is typically tackled via optimization or heuristic methods. A possible solution to this search problem comes from solving the same problem in organic synthesis, through the use of deep neural networks (Segler et al., 2018). Segler et al. preprocessed 12.4 million reaction rules from the Reaxys chemistry database to train three deep neural networks implemented within a Monte Carlo tree search (heuristic search algorithm used in decision making) to discover retrosynthesis routes for small molecules. This deep learning approach found pathways for twice as many molecules, thirty times faster than traditional computer-aided searches (Segler et al., 2018). The predicted synthesis routes better adhered to known chemical principles than traditional computer-aided searches and could not be differentiated by expert organic chemists compared to synthesis routes taken from the literature, highlighting the potential of deep learning to be applied for metabolic retrosynthesis (or retrobiosynthesis). Indeed, a similar Monte Carlo Tree Search method has recently been extended to predict synthetic pathways within biological systems (RetroPath RL), enabling systematic pathways design for metabolic engineering (Koch et al., 2020).

Pathways designed via retrosynthesis still face the difficult challenge of finding enzymes for novel biochemical reactions, for which no enzyme is known. In this case, the solution involves enzymes that may catalyze the novel reaction through enzyme promiscuity, or new enzyme functions must be designed or evolved that perform the desired chemistry. While chemoinformatic techniques (e.g. density functional theory, DFT, and partitioned quantum mechanics and molecular mechanics, QM/QM) can be used to predict the interaction between metabolites and proteins in silico (Alderson et al., 2012), these techniques are computationally intensive and require substantial domain expertise. Therefore, the task of searching for promiscuous enzymes is increasingly being performed using more general and computationally efficient techniques

Table 1
Machine learning applications for metabolic engineering.

Task	Application	Input features	Algorithm	Response	Ref.
Identify ORF	Identify signal peptide (SignalP 5.0)	20,758 protein amino acid sequence	deep RNN	presence or absence of signal peptide	Armenteros et al. (2019)
	ORF prediction (DeepRibo)	626,708 candidate ORF DNA sequences and ribo-seq signal (alignment file) from 7 species	RNN and CNN	translation initiation site and translated open reading frames	Clauwaert et al. (2019)
	ORF prediction (REPARATION)	67,158 candidate ORF DNA sequences and ribo-seq signal (alignment file) from 4 species	random forest	translation initiation site and translated open reading frames	Ndah et al. (2017)
Annotate ORF	Annotate enzyme (DeepEC)	1,388,606 protein sequences and 4669 EC numbers	CNN	enzyme commission (EC) numbers	Ryu et al. (2019)
	Annotate enzyme (ECPred)	11,018 protein amino acid sequences, subsequence extraction and peptide physicochemical properties	ensemble classifier (BLAST-kNN, Pepstats-SVM and SPMap)	EC numbers	Dalkiran et al. (2018)
	Annotate enzyme (DEEPre)	22,168 protein amino acid sequence	CNN and RNN	EC numbers	Le et al., 2018
	Annotate enzyme (EnzyNet)	63,558 protein sequences represented as voxel-based protein spatial structure	CNN	EC numbers	Amidi et al., 2018
Enzyme & Pathway design	Automated enzyme search	10,951 compounds and 6556 reactions. Features represented as reaction signature and enzyme amino acid sequence	support vector machines	positive or negative enzyme-reaction pairs	Faulon et al. (2008)
	Automated enzyme search	7318 reactions and 9001 enzymes. Features represented as reaction signature and enzyme amino acid sequence pair	gaussian process model	positive or negative enzyme-reaction pairs, and Michaelis constant KM (substrate affinity)	Mellor et al. (2016)
	Directed evolution	805 protein amino acid sequence variants	linear, kernel, neural network, and ensemble methods	protein fitness	Wu et al. (2019)
	Directed evolution	585,199 protein amino acid sequence variants	partial least-squares linear regression and multivariate optimization	bacterial halohydrin dehalogenase productivity	Fox et al. (2007)
	Directed evolution	218 protein amino acid sequence variants	gaussian process	fluorescent protein color	Saito et al. (2018)
	Directed evolution	4716 protein amino acid sequence variants	gaussian process	protein thermostability	Romero et al. (2013)
	Rational protein design (UniRep)	~24 millon protein amino acid sequence	RNN	protein feature representation	Alley et al. (2019)
	Rational protein design	96 protein amino acid sequence variants (UniRep encoding)	UniRep pretraining + linear regression (ridge, lasso-lars, ensemble)	protein fitness	Biswas et al. (2020)
	Rational protein design (BioSeqVAE)	protein amino acid sequence	residual neural networks	protein representation	Costello and Garcia Martin, 2019
	Synthetic pathway design (RetroPath RL)	N/A	Monte Carlo Tree Search reinforcement learning	metabolic pathway	Koch et al. (2020)
Pathway optimization	Promoter design	675,000 constitutive and 327,000 inducible promoter sequences	CNN	gene expression activity	Kotopka and Smolke, 2020
	Promoter design	100 mutated promoter and RBS sequences	neural network	promoter strength	Meng et al. (2013)
	Promoter design	promoter amino acid sequence	neural network	promoter strength	Tunney et al., 2018
	Riboswitch design	biophysical properties (entropy, stem melting temperature, GC content, length, free energy, etc) from 96 riboswitch aptamer sequences	Random forest and CNN	dynamic range of gene expression between ON/OFF states	Groher et al. (2019)
	Plasmid design (SelProm)	120 plasmid sequences	partial least-squares regression	promoter strength, induction time, inducer concentration	Jervis et al. (2019a)
	multi-gene pathway optimization (MiYA)	24 strains, different promoter combinations	neural network	β-carotene and violacein production	Zhou et al. (2018)
	multi-gene pathway optimization	156 strains, different RBS sequences	support vector machines and neural network	limonene production titer	Jervis et al. (2019b)
	multi-gene pathway optimization (BioAutomata)	136 strains, different promoter and RBS sequences	gaussian process and Bayesian optimization	lycopene production titer	HamediRad et al. (2019)
	(Automated Recommendation Tool (ART)	promoter combinations, multi-omics data, etc	Bayesian ensemble model	chemical production titer, rate, yield	Radivojević et al. (2020)
	multi-gene pathway optimization	250 strains, different promoter combinations	probabilistic ensemble model (ART) and Bayesian optimization (EVOLVE algorithm)	tryptophan production titer	Zhang et al. (2020)
	Multi-gene Pathway optimization	12 strains, 4 biological replicates per strain	Ensemble model (random forest, polynomial, multilayer perceptron, TPOT meta-learner)	Dodecanol production	Opgenorth et al. (2019)
CRISPR	sgRNA Scorer 2.0, CRISPR activity	430 sgRNA sequences	support vector machine	sgRNA on-target activity	Chari et al. (2017)

(*continued on next page*)

Table 1 (*continued*)

Task	Application	Input features	Algorithm	Response	Ref.
	Azimuth, CRISPR activity	4390 sgRNA sequences	support vector machine with logistical regression	sgRNA on-target and off-target activity	Doench et al. (2016)
	Seq-DeepCpf1, CRISPR activity	16,292 sgRNA sequences	CNN	sgRNA on-target activity	Kim et al. (2018)
	Elevation, CRISPR activity	299,387 sgRNA–target pairs	gradient boosted regression trees	sgRNA off-target activity	Listgarten et al. (2018)
	CRISPR activity	294,534 sgRNA-target pairs	CNN and deep feedforward neural network	sgRNA off-target activity	Lin and Wong (2018)
	DeepCRISPR, CRISPR activity	0.68 billion sgRNA sequences (unlabeled pre-training data) ~160,000 sgRNA-target pairs (off-target data sets) ~200,000 sgRNA seq uences (on-target data set)	deep convolutional denoising neural network and CNN	sgRNA on-target and off-target activity	Chuai et al. (2018)
Outlier Detection	Novelty and Outlier Detection	Any training data set	isolation forest, local outlier factor, one-class SVM, and elliptic envelope	outlier identification	https://scikit-learn.org/stable/modules/outlier_detection.html
Omics Data Processing	Prosit, peptide identification	550,000 tryptic peptides	bi-directional RNN	peptide chromatographic retention time and tandem mass spectra	Gessulat et al. (2019)
	Peakonly, metabolite peak detection	4000 regions of interest, labeled as noise, one or more peaks, or uncertain peak	CNN	peak detection + integration (peak area)	Melnikov et al. (2020)
Bioprocess Control & optimization	Bioprocess optimization	69 fed-batch fermentations, 13 process features (fermentation conditions, inoculum conditions, media variables)	three-step optimization method using decision trees, neural network, and hybrid genetic algorithm	maximum cell concentration, product concentration, and productivity	Coleman et al. (2003)
	Bioprocess optimization	25 fed-batch fermentations, 11 process features (temperature, induction strength, growth rate, process variables) + spectroscopic information	data preprocessing, random forest and neural network	cell dry mass, recombinant soluble protein conc., inclusion bodies conc.	Melcher (2015)
	Bioprocess optimization	27 batch fermentations, 7 process features (time, pH, temperature, kLa, biomass, xylose, glycerol)	regression and neural network coupled to genetic algorithm for optimization	xylitol production	Pappu and Gummadi (2017)
	Bioprocess control & real-time optimization	continuous bioreactor, 24 h duration with measurement/action every 5 min	Neural network fitted Q-learning algorithm (reinforcement learning)	control species biomass ratio; maximize product yield	Treloar et al. (2020)
	Process control	600 temperature measurement/action timesteps (episodes)	Model Predictive Control (MPC) guided deep deterministic policy gradient (reinforcement learning). Policy parameterized by neural network	Reactor (CSTR) temperature control	Xie et al., 2020
	Real-time process optimization	500 measurement/action episodes	Policy gradient parameterized by a recurrent neural network. Transfer learning from offline training on mechanistic model	Maximize product yield (phycocyanin)	Petsagkourakis et al., 2020
	Process control	21 measurement/action episodes	multi-step action Q-learning controller based on fuzzy k selector	ethanol concentration control	Li et al., 2011

Notes: RNN = recurrent neural network; CNN = convolutional neural network.

from machine learning. For example, given a reaction and enzyme pair instance, Support Vector Machines (Faulon et al., 2008) and Gaussian Processes (Mellor et al., 2016) have been developed to predict whether the enzyme catalyzes the reaction, with the latter model having the added benefit of providing uncertainty quantification. These models predict positive or negative enzyme reaction pairs from protein sequences (e.g. K-mers) and reaction signatures (e.g. functional groups, chemical transformation properties) (Carbonell and Faulon, 2010) by learning patterns about promiscuous enzyme activities through training. They can also be applied to predict substrate affinity for proteins (K_m values) (Mellor et al., 2016), an important kinetic parameter for determining enzyme activity, which is difficult and time consuming to measure experimentally. This is critical for pathway design as sequences with the most desirable kinetic properties can be selected when multiple candidates catalyzing a given reaction are available.

In the case that no enzyme can be found for a target reaction, new enzymes may be designed or discovered through protein engineering. A common laboratory method for protein engineering is directed evolution, where beneficial mutations accumulate in a protein through iterative experimental rounds of mutation and selection until the desired protein function is achieved (Yang et al., 2019). In essence, a series of local searches (via sequence mutation and screening) are performed on an enormous and highly complex functional landscape with the hope of finding a local optima (i.e. protein variant with desired properties). However, experimental approaches can only explore an infinitesimal part of this landscape and computational approaches are needed to guide experimental efforts. Machine learning can be used to guide directed evolution and decrease the number of experimental iterations needed to obtain a protein with the desired function. This is achieved by leveraging previous screening data to learn a protein's sequence-function landscape and predict new sequence libraries that contain variants with higher fitness. For example, instead of experimentally performing sequential single point mutations or recombining mutations found in best variants (common directed evolution approaches), Wu et al. (2019) trained a machine learning model to perform in silico evolution rounds that ranked new protein variants by predicted fitness for experimental testing. Instead of relying on a single machine learning method, multiple models (linear, kernel, neural network, and ensemble) were trained in parallel, and the ones showing the highest accuracy were used to perform in silico evolution rounds (Wu et al., 2019). This enabled deeper exploration of the possible variant functional landscape, resulting in the successful evolution of an

immunoglobulin-binding protein and a putative nitric oxide dioxygenase from *Rhodothermus marinus*. ML-assisted directed evolution has also been used to maximize enzyme productivity (Fox et al., 2007), change the color of fluorescent proteins (Saito et al., 2018), and optimize protein thermostability (Romero et al., 2013) making it a promising approach for searching large sequence-function spaces in an efficient manner for proteins variants with desired properties.

In addition to directed evolution, deep learning has also recently been applied for the rational design of proteins (Alley et al., 2019; Biswas et al., 2020; Costello and Garcia Martin, 2019). For example, Alley et al. (2019) developed UniRep, which uses recurrent neural networks to learn an internal statistical representation of proteins that contained physicochemical, organism, secondary structure, evolutionary and functional information, by training on 24 million UniRef50 (Suzek et al., 2015) amino acid sequences (instances). The resulting representation was applied to train models (random forest or sparse linear model) using UniRep encoded proteins that predicted the stability of a large collection of de novo designed proteins and also the functional consequence of single point mutations on wild-type proteins. UniRep encoding was also used to optimize the function of two fundamentally different proteins (to wild-type), a eukaryotic green fluorescent protein from *Aequorea victoria*, and a prokaryotic β-lactam hydrolyzing enzyme from *Escherichia coli*, highlighting the generalizability of this approach for rational protein engineering (Biswas et al., 2020). Other generative models based on deep learning have been used to suggest protein sequences with desired functionality and location (Costello and Garcia Martin, 2019).

3.1.2. Pathway optimization

Following pathway design, metabolic flux optimization is required to maximize product titers, rates, and yields (TRY). In this endeavor, machine learning provides an orthogonal approach to computational approaches leveraging flux analysis and genome-scale models, which have been successfully used in the past to increase TRY (Maia et al., 2016). The combination of both approaches has the potential to be more effective than each of them separately (see section 4 for a discussion).

A common approach to increase TRY involves fine tuning gene expression through the modification of promoter and ribosome binding site (RBS) sequences. Despite decades of progress in understanding the regulatory mechanisms controlling gene expression (Snyder et al., 2014), quantitative prediction of gene expression based on sequence information remains challenging. While computational models do exist to predict gene expression (Leveau and Lindow, 2001; Salis et al., 2009; Rhodius and Mutalik, 2010), they rely on a comprehensive understanding of transcription and translation processes. This knowledge is often unavailable, especially for non-model organisms. Therefore, many gene expression optimization efforts rely on trial-and-error experimental approaches based on promoter and RBS library screening (Choi et al., 2019), that also suffer from the large combinatorial space of possible sequences.

Machine learning has also guided the design of promoter and RBS sequences in a data-driven manner for improved control of gene expression. In particular, neural networks have been used to predict gene expression output from input promoter sequences or coding regions (Kotopka and Smolke, 2020; Meng et al., 2013; Tunney et al., 2018). Meng et al. (2013) used a simple neural network trained with 100 mutated promoter and RBS sequences as inputs to predict promoter strength (response). This machine learning model outperformed mechanistic models based on position weight matrix or thermodynamics methods (Leveau and Lindow, 2001; Salis et al., 2009; Rhodius and Mutalik, 2010), and was able to optimize heterologous expression of a small peptide BmK1 (used in traditional Chinese medicine) and the *dxs* gene involved in the isoprenoid production pathway (Meng et al., 2013). Additionally, optimization of promoter strength and inducer concentration/time has been achieved using partial least squares regression (Jervis et al., 2019a), whereas prediction of riboswitch dynamic range from aptamer sequence biophysical properties has been achieved using a combination of random forests and neural networks (Groher et al., 2019). In this latter riboswitch design example, instead of directly using sequence information to train the random forest, the authors calculated known riboswitch biophysical properties from aptamer sequences (entropy, stem melting temperature, GC content, length, free energy, etc.) and used these as input features for model training, in order to predict switching behavior. This allowed for the interpretation of which input features were most important to the model prediction using variable importance (e.g. melting temperature was more important than free energy), enabling inferences on possible mechanisms.

More recently, machine learning models have been used to optimize multi-step pathways for chemical production (Zhou et al., 2020). For example, Zhou et al. (2018) used neural network ensembles to improve a 5-step pathway for violacein production (pharmaceutical) by selecting promoter combinations to tune gene expression. Using an initial training set of only 24 strains (out of a possible 500) containing different promoters for each gene, the model predicted a new strain that improved violacein titer by 2.42-fold after only 1 DBTL iteration. Their ensemble approach allowed top producing strains to be predicted from a combination of over 1000 ANN, which improved model accuracy and also allowed optimization of violacein based on both titer and purity. In another example, Opgenorth et al., (2019) used an ensemble of four different models (random forest, polynomial, multilayer perceptron, TPOT meta-learner) to optimize a 3-step pathway for dodecanol production from two DBTL cycles. The model was trained using data generated from 12 strains (48 data points total) with different RBS sequences for each gene, where an optimization step was used to recommend improved strain designs to build and test in the second cycle. Additional machine learning models have guided the optimization of multi-gene pathways, including limonene production in E. coli using support vector regression (Jervis et al., 2019b), lycopene synthesis in E. coli using gaussian processes (HamediRad et al., 2019), and tryptophan production in S. cerevisiae using ensemble models (Zhang et al., 2020). Together, these examples highlight the potential of systemically leveraging high-throughput strain construction, testing, and machine learning to optimize multi-step pathway expression for improving product TRY.

To enable broader use of ML-driven pathway optimization and design by the metabolic engineering community, Radivojevic et al. (Radivojević et al., 2020) developed the Automated Recommendation Tool (ART). ART is specifically tailored to the needs of the metabolic engineering field: effective methods for small training data sets and uncertainty quantification. ART's ability to quantify uncertainty enables a principled way to explore areas of the phase space that are least known, and is of critical importance to gauge the reliability of the recommendations. We expect that further development of tools tailored to the specific needs of the field will enable broader application of machine learning.

3.2. Machine learning for building and testing cellular factories

Machine learning can also be used to improve the tools that build and test cellular factories. A major challenge in gene editing using CRISPR-Cas technologies, for example, is predicting the on-target knockout efficacy and off-target profile of single-guide RNA (sgRNA) designs. Several approaches exist to make these predictions, including alignment-based methods (Aach et al., 2014), hypothesis-driven methods (Heigwer et al., 2014; Hsu et al., 2013), and classic machine learning algorithms (i.e. non-deep learning) (Chari et al., 2017; Doench et al., 2016). However, their generalizability has been limited by the small size and low quality (high noise) of the training data. Higher-throughput screening methods combined with deep learning have recently improved the accuracy and generalizability of sgRNA activity prediction tools. For example Kim et al (Kim et al., 2018), developed DeepCpf1, which predicts on-target knockout efficacy (indel frequencies) using deep neural networks trained on large-scale sgRNA

(AsCpf1) activity data sets. While previous machine learning tools had been trained on medium-scale data (1251 target sequences), the authors high-throughput experimental approach generated a data set of indel frequencies for over 15,000 target sequence compositions, which was sufficient to train deep neural networks. Seq-DeepCpf1 was shown to outperform conventional ML-based algorithms, and steadily increased in performance as training data size increased, highlighting the value of data sets with >10,000 high-quality training instances. Seq-DeepCpf1 was also extended by considering input features other than target sequence composition known to affect sgRNA activity (in this example, chromatin accessibility (Jensen et al., 2017)) that further improved prediction accuracy and performance on independently collected data sets from other cell types (a metric of model generalizability). This highlights the value of expanding input features beyond the obvious choice.

In addition to predicting on-target knockout efficacy, the off-target profile of sgRNA activity is also important to forecast, in order to prevent undesirable perturbations that result in genomic instability or functional disruption of otherwise normal genes. This has been performed using both regressive models and deep neural networks (Listgarten et al., 2018,Lin and Wong, 2018). To combine on-target knockout efficacy and off-target profile predictions into one tool Chuai et al (Chuai et al., 2018), developed DeepCRISPR. DeepCRISPR uses both an unsupervised deep representation learning technique and deep neural networks to maximize on-target efficacy (high sensitivity), while minimizing off-target effects (high specificity). Unsupervised representation learning allows DeepCRISPR to automatically discover the best representation of input features from billions of genome-wide unlabeled sgRNA sequences, instead of specifying what input features should look like (e.g. target sequence composition). This sgRNA representation was then used when training a deep neural network using labeled data consisting of target sequences and epigenetic information (input features) to predict both on-target and off-target activities (responses). Overall, DeepCRISPR outperformed classic machine learning methods and exhibited high generalizability to other cell types, highlighting the value of unsupervised representation learning to automate feature identification.

Machine learning methods could also be used to optimize the DNA assembly and transformation protocols critical for building engineered strains. Although DNA and strain construction has traditionally been accomplished empirically (Chan et al., 2013) or guided by rule-of-thumb approaches (Engler and Marillonnet, 2014), the ability to assemble and test DNA constructs and their transformation efficiencies under different conditions in high-throughput could enable data-driven optimization. For example, machine learning could leverage comprehensive overhang ligase fidelity data sets (Potapov et al., 2018) to expand the identification of high-fidelity overhang sets for gibson assembly, potentially allowing more DNA fragments to be assembled in a single reaction. Machine learning could also leverage large data sets that examine transformation efficiency under a range of different conditions (e.g. media compositions, temperatures, incubation times, electroporation conditions, plasmid designs) to improve plasmid delivery and expression. This would be particularly useful for expanding genetic systems to a broader range of host organisms that have potential for industrial applications (Brophy et al., 2018; Wang et al., 2019).

Once cell factories are built their performance needs to be tested. Cell factories can be assayed for various components such as target molecules, transcripts, proteins, and metabolites. The throughput of these assays varies greatly from over 10,000 samples per day to fewer than 20 samples per day (Petzold et al., 2015). Together, the data from these assays provide a comprehensive picture of how the engineered cells function. However, constructing large numbers of strains followed by high-throughput screening often produces noisy data sets arising from several factors, including small plate-based formats (e.g. edge effects), analytical measurement errors, and laboratory handling errors and biases. One way to reduce these errors is manual inspection, but this approach is not scalable for large data sets and often not reproducible due to person-to-person variability. Therefore, machine learning methods that predict outliers and biases from data and perform data processing in a standardized and reproducible manner are desirable. For this, the use of unsupervised learning algorithms that do not depend on "good" and "bad" labeled data examples have been used, such as clustering analysis methods (Fig. 6). The sci-kit learn library implemented in python has a set of machine learning tools available to perform outlier detection, including Isolation Forest, Local Outlier Factor, One-Class SVM, and Elliptic Envelope, that can be integrated into workflows to provide rapid and robust data quality processing. Additionally, supervised learning approaches based on deep neural networks have been applied to improve multi-omics data processing, for example protein identification from tandem mass spectra (Gessulat et al., 2019) and peak detection during metabolomic data processing (Melnikov et al., 2020). Given the large volume of data generated overtime from lab workflows and analytical instruments, further efforts to standardized data processing using machine learning should result in improved data sets for cellular factory design and analysis.

3.3. Machine learning for scaling up cellular factories

One of the largest challenges in metabolic engineering is maintaining the performance of laboratory strains when scaling up to commercial production plants (Chubukov et al., 2016; Wehrs et al., 2019). The typical procedure consists of cultivating lab strains in successively larger fermentation systems from bench-scale (~250mL-5L), to pilot-scale (~20–200L), to full-scale processes (>1000L). Critical to successful scale up is understanding how process variables (feed rate, pH, temperature, fermentation time, mixing regime, media composition, aeration rate, etc.) impact host physiology, cell growth, and product TRY. Accordingly, a central task of bioprocess scale-up is to identify and fine tune these process variables to maintain robust and stable production of the desired chemical. This process is often heuristic, and scale-up process development is often seen as more of an art than a science (Crater and Lievense, 2018; Humphrey, 1998). The fundamental reasons for this, is that large scale fermentations are expensive and difficult to predict. A fermentation is a massively multiparametric process that can be affected by the slightest change in any of the number of factors involved in bioreactor conditions. For example, a change in feedstock or water source, inoculation volume, or even altitude of the bioreactor can impact the progress of the fermentation process. Performing thorough fermentation optimization studies in bioreactors is not only expensive, but also time consuming. Each 2L bioreactor test can cost over 1000 USD and last over a week. Hence, scientific methods are needed to accelerate fermentation process development in bioreactors, beyond the current artisanal procedure. Fortunately, modern fermentation systems used during scale up and at commercial plants contain sophisticated process controls, comprehensive data collection and archiving systems, and automation, which can be leveraged for training machine learning algorithms.

The use of machine learning to mine the wealth of online and offline bioprocess data to shed light on the cause of scale-up process failures, and to improve process outcomes, is common (Charaniya et al., 2008; Baughman and Liu, 2014). For example, Coleman et al. (2003) used historical process data to develop a three-step optimization method using decision trees, an ANN ensemble, and a genetic algorithm to identify which process input variables were most important for fermentation modeling, and to select input values that increased product output. To avoid overfitting, process inputs (different fermentation, media, and inoculum conditions – 13 total) were sub-selected using decision tree analysis on a data set of 69 fed-batch fermentations, which identified inputs that best corresponded with each process output (biomass density, product concentration, and productivity). This feature selection preprocessing step is common for bioprocess data sets to remove highly correlated or redundant process inputs prior to model

training to prevent overfitting (Melcher et al., 2015; Coleman et al., 2003; Charaniya et al., 2008). The subsetted inputs were then used to train ANN ensembles to quantitatively predict each process output. This resulted in a data-driven process model that was used to identify novel input conditions that maximized process outputs via optimization (genetic algorithm). A similar approach combining ANN modeling followed by optimization using a genetic algorithm was taken by Pappu et al. (Pappu and Gummadi, 2017) to optimize fermentation parameters for producing xylitol. The model accurately predicted xylose consumption, biomass density, and xylitol production following training on 27 fermentation batches with multiple inputs, and was used to select new process inputs (pH, agitation speed, and aeration rate) that increased xylitol titers from 59.4 to 65.7 g/L. These examples highlight the ability to generate predictive process models in a data-driven fashion, providing an alternative to more traditional physical-based kinetic models (e.g. Monod or Droop model) that often fail to capture poorly understood relationships between microbial growth and multiple process variables (Kovárová-Kovar and Egli, 1998).

Bioprocess data is highly heterogeneous and requires appropriate data pre-processing to be used for machine learning. Many bioprocess parameters are collected online as continuous measurements (optical (cell) density, pH, dissolved oxygen, oxygen uptake rate, flow rate, off-gas production, etc.) while others (e.g., chemical concentrations, substrate consumption rates) are measured offline at discrete time intervals. Additionally, some parameters, such as product concentrations, are only measured at the final time point, while others are categorical or binary (e.g. ON/OFF nutrient feed setting). This results in highly heterogeneous data sets with respect to time and between fermentation runs that require pre-processing to extract temporal trends that compactly and smoothly represent the data, preventing model overfitting (i.e. many more features than instances). For example, instead of using each time point measurement for model training, first and second order derivatives can be used to more compactly represent temporal trends (Cheung and Stephanopoulos, 1990a); (Cheung and Stephanopoulos, 1990b), as can wavelet decomposition methods (Bakshi and Stephanopoulos, 1994), which outperforms more classical smoothing approaches such as Savitzky-Golay. For low and very low signal-to-noise ratios, more recent methods of denoising can be applied, such as mean envelope filter (Merino et al., 2015) or spectral noise reduction by vector casting (Gebrekidan et al., 2020). Other approaches, including discrete Fourier transform and symbolic aggregate approximation (SAX) can be applied, which represent temporal trends as representative segments (e.g. mean over time window) instead of the entire time-series (Charaniya et al., 2008). In addition to reducing the number of timepoints used for model training, temporal offsets between data sets can arise, for example, due to lag phases in growth between fermentation batches. This can be corrected using dynamic time warping strategies that align time profiles between data sets to avoid incorrect comparisons (Chakrabarti et al., 2002); (Keogh and Ratanamahatana, 2005).

The availability of continuous online bioreactor data has also enabled control and optimization of bioprocesses through reinforcement learning. Currently, bioprocesses are controlled manually or using proportional–integral–derivative (PID) controller or model predictive control (MPC) (Qin and Badgwell, 2003) methods that automatically modulate one or more process variables (e.g. feeding rate) to control an output (e.g. temperature, production concentration). While these techniques have been widely used for complex multivariable control applications, they are built upon fixed models of the environment that do not get continuously updated and improved as they see more data. Therefore, there is growing interest in using model-free reinforcement learning methods to learn, through trial and error, the best control algorithm from large online data, and to optimize process operations (for a detailed overview see (Shin et al., 2019)). For example, a control policy was learned from online ethanol data to control final ethanol titers during yeast fermentations that had a lower overshoot, faster tracking, shorter transition, and smoother control signal than an advanced PID controller (Li et al., 2011). Reinforcement learning methods have also been demonstrated in simulated systems to control co-culture species biomass abundances and optimize product yields (Treloar et al., 2020), control reactor temperatures (Xie et al., 2020), and to optimize a downstream product separation unit (Hwangbo and Sin, 2020). However, current reinforcement learning methods alone still suffer from requiring large amounts of data for complex multivariable processes, and are often impractical or too costly to implement in real world applications (Shin et al., 2019). Therefore, approaches to improve the sample efficiency of reinforcement learning methods are needed; promising examples include combining them with model-based controllers (Xie et al., 2020) or through transfer learning, where offline model simulations are initially used to train control policies followed by the efficient adaptation of these policies with real online data (Petsagkourakis et al., 2020).

In sum, despite the challenges of high experimental cost and unpredictable nature of fermentations, the wealth of data generated in a single fermentation makes application of machine learning to scale-up an appealing proposal. Machine learning can be used to identify optimal fermentation parameters (i.e. selecting the most appropriate process conditions) and recommend appropriate responses during process upsets (via adaptive process monitoring and control) using the large amount of data that is available**.** This area may benefit significantly from coupling machine learning with mechanistic modelling (see next section) such as computational fluid dynamics simulations (Haringa et al., 2016, 2017).

4. Machine learning and mechanism

4.1. Two paradigms at odds

Whereas the machine paradigm concentrates on enabling predictive power, metabolic engineers typically define scientific value around the understanding of mechanism, because it is perceived to be the road to better performance. Mechanisms are defined as the causally related set of processes and parts that result in the observed phenomena. Understanding these mechanisms has been crucial in the history of microbiology because it results in knowledge that can indeed be leveraged to predict the behavior of a biological system (pathways, strains, products, etc.) and can also be transferred to different systems where the same mechanism is involved. For example, if fosmidomycin is toxic and inhibits 1-deoxy-d-xylulose-5-phosphate reductoisomerase (DXR) in the mevalonate pathway in *E. coli*, you would expect fosmidomycin to inhibit DXR in another host (Murkin et al., 2014). The kinetics of this inhibition mechanism can also be used to quantitatively predict the corresponding changes in mevalonate pathway flux, based on a Michaelis Menten equation that relates fosmidomycin concentration and DXR reaction rate (i.e. inhibitory dissociation constant, K_i).

While there are a variety of different mechanistic mathematical models that are useful for guiding design, including gene expression models (Ay and Arnosti, 2011), genome-scale models (GSM) (King et al., 2016,Thiele and Palsson, 2010), kinetic models (O. D. O.D. Kim et al., 2018), whole cell models (Karr et al., 2012; Macklin et al., 2020), and process models (Koutinas et al., 2012), many of them fail to provide the accurate quantitative predictions needed to systematically drive metabolic engineering projects in practice. For example, predicting metabolic flux changes due to gene knockouts with GSM remains challenging (O'Brien et al., 2015), even after attempts to improve prediction accuracy by deriving constraints or objective functions from experimental data such as transcriptomics (Machado and Herrgård, 2014). Moreover, kinetic model predictions based on assumed quantitative relationship between inputs (e.g. fosmidomycin concentration) and outputs (e.g. DXR reaction flux) often do not hold in reality (Costa et al., 2010; Heijnen, 2005) and are nearly impossible to parameterize for every enzyme across all growth conditions. A key reason why these models fail is because their mathematical relationships between inputs and outputs

are based on ideal conditions (e.g. in vitro for Michaelis Menten equation) that do not capture the complexity of the intracellular environment (e.g. regulation). They also lack the ability to automatically leverage more data to learn and improve prediction performance. If the model predictions fail, it takes a human head to creatively figure out how to correct the model, which often happens too slowly, leaving design to rely on trial-and-error experimental approaches. Therefore, new quantitative prediction frameworks are needed to drive the commercial success of metabolic engineering projects in industry, and bring about the field's full potential (as discussed in the introduction).

Machine learning's flexible data-driven framework can help overcome the challenges facing predictive biology. Machine learning links inputs and outputs (Fig. 2) without needing to understand what happens in between (i.e. the mechanism). Instead of using knowledge-derived mathematical relationships, machine learning models empirically derive input/output relationships (equations) through training on data that can be collected in a higher throughput manner (titers, rates, yields, expression levels, etc.) and can automatically improve prediction performance as more data becomes available. Of course, machine learning approaches have their own limits. They require a large amount of data that is expensive to collect, and which constitutes currently the largest practical bottleneck (see Section 5.1). Moreover, most machine learning algorithms, particularly deep neural networks, are black boxes and difficult to interpret, although this is also improving (see Section 5.3). Therefore, troubleshooting machine learning models to try and achieve further predictive power once performance has plateaued is challenging, especially since a clear connection to mechanism is not available. Accordingly, the preferred type of model is both predictive and mechanistic, and it is by leveraging machine learning with mechanistic models that these types of models can be created.

4.2. Integrating biological knowledge and machine learning

It is by integrating machine learning and mechanistic models that the benefits of both approaches can be combined: predictability that systematically increases as more data is available, and mechanistic insight. It is not entirely clear how to proceed about reaching this goal, but there are some budding attempts (Fig. 11). A more comprehensive list of approaches that integrate data- and knowledge-based models can be found in the review by Zampieri et al. (2019).

One interesting avenue to explore is whether machine learning can be used to parameterize mechanistic models. A couple of studies (Andreozzi et al., 2016; Heckmann et al., 2018) demonstrated the potential for this by leveraging a set of machine learning models to predict enzyme catalytic turnover numbers from input features composed of network properties, enzyme structural properties, biochemistry, and assay conditions. Enzyme turnover numbers were then used to parametrize genome-scale models which improved proteome predictions. Similarly, Chakrabarti et al. (2013) used a machine learning approach to identify feasible kinetic parameters for an ORACLE (optimization and risk analysis of complex living entities) kinetic model of metabolism. More generally, deriving biological knowledge from machine learning methods would enable an efficient way to advance scientific understanding from the increasing data deluge coming from multi-omic approaches. While it is not obvious how an actual mechanism can be learnt from purely data-driven machine learning approaches that are based on correlations rather than causation, some recent examples have demonstrated promising results in identifying relationships that are candidates for follow-up experiments to distill mechanisms. For example, Ma et al. (2018) developed a visible neural network (VNN), which couples the model's inner workings to those of a real system, by incorporating knowledge from gene ontologies into a VNN to simultaneously simulate cell hierarchical structure and function. The resulting VNN was optimized for functional prediction (e.g., growth rate) while respecting biological structure (subsystem hierarchy) and was capable of identifying subsystem activity patterns. Another study (Zelezniak et al., 2018) leveraged metabolic network information to predict metabolite concentrations (response) from protein levels (input) in *S. cerevisiae* mutants through a multilinear regression: metabolite concentrations were expressed as a function of expression levels of the closest enzyme neighbors in the metabolic network. A more general approach called explainable artificial intelligence, XAI (see Section 5.3), presents enourmous potential for providing mechanistic insights within data-driven machine learning models. An algorithm of this type was able to detect enhancer activity in the Drosophila embryo and alternative splicing in human-derived cell lines by systematically capturing high-order interactions between features that are predictive of the response (Basu et al., 2018).

Another possible approach is to incorporate input features derived from mechanistic models into machine learning models to improve their predictive power. For example, Culley et al. (2020) developed a machine learning pipeline for predicting *S. cerevisiae* growth rate that leveraged transcriptomic data and genome-scale model predicted fluxes as input features. They show that using fluxes predicted from parsimonious flux balance analysis (pFBA) as features combined with transcriptomics data improved the predictive power of neural networks over using transcriptomics data alone. In a similar direction, it would be worthwhile exploring whether synthetic data augmentation based on mechanistic simulations can increase predictive accuracy of machine learning models while learning hypothesized mechanisms underlying the data. Also, mechanistic models can be a useful tool for feature selection for machine learning models. It has been shown that GSMs can be fruitfully

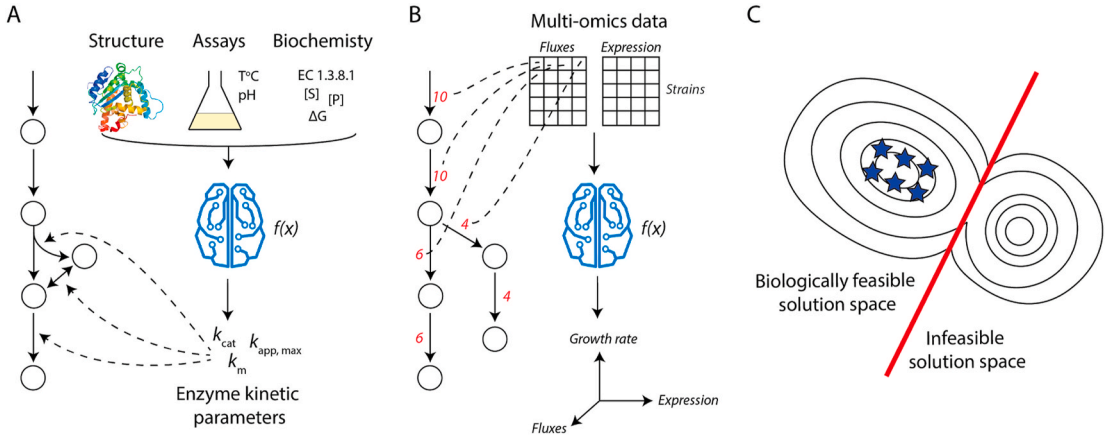


Fig. 11. Integrating machine learning and mechanistic models. (A) Parametrizing GSMs using machine learning predictions; (B) Using GSMs to derive input features for machine learning; (C) constraining the machine learning solution space with mechanism.

leveraged to identify a subset of reactions to then be optimized through machine learning methods (Zhang et al., 2020).

Finally, incorporating known physical or biological constraints on the solution space of machine learning algorithms can ensure biologically meaningful solutions or rule out possible machine learning solutions that are known to be biologically infeasible. In a study by Wu et al. (Wu et al., 2016) various machine learning algorithms were used to predict central carbon metabolic fluxes measured through ^{13}C Metabolic Flux Analysis (response) from culture and genetic information (input). The best performing machine learning model flux predictions were then changed as minimally as possible to satisfy the stoichiometric constraints provided by a GSM. Similarly, machine learning was used to reconcile empirical genetic interaction data with FBA model predictions (Szappanos et al., 2011).

4.3. Inspiring new machine learning from metabolic engineering

Metabolic engineering can also provide inspiration for new machine learning and AI algorithms. Biomimicry was the inspiration for neural networks (Fig. 7), so it is not unreasonable to think that biology can be the inspiration for more and better algorithms. Gene regulatory networks, for example, involve a sophisticated network of molecular interactions that regulate and determine the cell behaviour to sense and react to environmental cues and optimize survival. A full mechanistic understanding of the general principles of how this is achieved for different cells, environments, and threats, could provide valuable insights for new machine learning approaches.

Indeed, metabolic engineering is in a better situation to inspire new machine learning algorithms than other disciplines. While there is no hope that understanding how a neural network identifies a cat will reveal physiologically meaningful information on the brain identification processes, in metabolic engineering we are quite close to the mechanism. Indeed, some of the mechanistic models provide predictions that may not be completely accurate, but are qualitatively acceptable (Lerman et al., 2012; Karr et al., 2012; Macklin et al., 2020). We believe using machine learning to complement the parts of mechanistic models that are less tested can significantly increase their accuracy. These hybrid models can lead to new inspiration for new machine learning architectures and general approaches.

5. Perspectives for machine learning in metabolic engineering

5.1. Major bottlenecks for further application

While the need for improved predictive power fosters the further application of machine learning in metabolic engineering, there are some fundamental obstacles to a wider application. These obstacles are both technical (data and algorithmic challenges) and sociological.

The foremost challenge is undoubtedly the scarcity of the large data sets needed to train machine learning algorithms. The majority of metabolic engineering projects typically involve much less than 100 strains/conditions. Whereas training instances can be multiplied by shrewd data augmentation (see section 2.2.1), it seems unlikely that the current status quo will be able to provide the amount of data routinely found in other fields (several million instances/images in ImageNet (Deng et al., 2009)). This will undoubtedly limit the benefit that metabolic engineering can leverage from machine learning. Another challenge is data quality, which is as important as quantity. High repeatability and low uncertainty are critical characteristics of high-quality data: an experiment must produce similar responses under identical inputs, or there is little hope that an algorithm can be predictive. Furthermore, data sharing is often hampered by the lack of biological data standards needed for this exchange. For example, in the case of multiomics data, there are databases for genes (e.g. Genbank IDs (Benson et al., 2011)), proteins (e.g. Uniprot IDs (The UniProt Consortium, 2017)), metabolites (pubchem IDs (Kim et al., 2016)), and reactions (e.g. BIGG database (King et al., 2016)), but these databases are often not comprehensive (e.g. not all proteins are submitted to Uniprot) and are not fully interlinked (e.g. BIGG metabolites not always have a pubchem entry). While there are efforts to solve this problem (e.g. Metanet X (Moretti et al., 2016), or BioCyc (Karp et al., 2019)), this issue rarely reaches the high profile needed to attract the investment required to completely solve it. Moreover, if the state of data standardization is not good, metadata standardization is in an even worse state. Without an investment in this piece of infrastructure, there is little hope for a disruptive impact of machine learning in metabolic engineering (Fig. 12). A possible solution to several of these problems involves automation (see section 5.2).

A second hurdle involves the adaptation of machine learning algorithms to the special needs of metabolic engineering. Uncertainty quantification is one of the needs of a discipline with small training data sets that is beginning to be met (Radivojević et al., 2020). Explainable AI (XAI) involves creating models such that the reasons for their predictions can be understood by humans (see section 5.3). This is particularly important in metabolic engineering, where we often have, or can easily investigate, the mechanism responsible for a given response. This investigation is much more complicated for other fields like, e.g., artificial vision or astrophysics. The integration of prior biological knowledge into machine learning algorithms, and its extraction from machine learning results is also an area that could provide significant advances in both metabolic engineering and machine learning, as discussed in section 4.

Another, often overlooked, obstacle involves the sociological challenge of having two very different groups working together: machine learning researchers and metabolic engineers. These two crowds are typically trained very differently and there is little intersection among them. Communication is, hence, often complicated by these differences. Furthermore, they are different not only in their skill toolbox, but also in which problems arouse their interest. This creates problems in aligning interests and managing projects. Interaction is, however, necessary: it is becoming impossible even for machine learning researchers to keep abreast of the literature on their field, and the new metabolic engineering tools (e.g., CRISPR-based gene editing, cell-free engineering) are posing a similar challenge in this field. Only through an interdisciplinary effort can the best of both disciplines be combined to create something bigger than the sum of the parts.

5.2. Integrating machine learning and synthetic biology with automation

As indicated above, the training data for machine learning must be high-quality, in the sense that it must avoid biases due to inconsistent protocols and provide quantification for repeatability (see section 2.4). Both goals can be systematically achieved through automation, which is one of the main reasons the intersection of machine learning, synthetic biology, and automation is thriving (Carbonell et al., 2019). Biological and chemical sciences data are nowadays growing at an unprecedented pace, but the databases aggregating biological and chemical findings are usually biased (Rodrigues, 2020). To avoid this bias, it is highly desirable to start veering away from the traditional approach of one entire PhD per molecule or one scientist performing the full metabolic engineering process, in order to adopt the creation and maintenance of integrated engineering pipelines (Fig. 13). This is the path embodied by biofoundries (Chao et al., 2017; Hillson et al., 2019). This goal can be achieved by extending current automation pipelines for machine learning (Olson and Moore, 2018). Pipelines are fully or semi-automated infrastructure that realize a procedure in a systematic manner: e.g., phenotyping through proteomics, strain construction, fermentation. Automated pipelines facilitate consistent protocols and reproducibility in synthetic biology (Jessop-Fabre and Sonnenschein, 2019), and have the capability to produce the amount of data required by machine learning. Fully automated and integrated DBTL pipelines have already been successfully adopted for the identification and optimization of

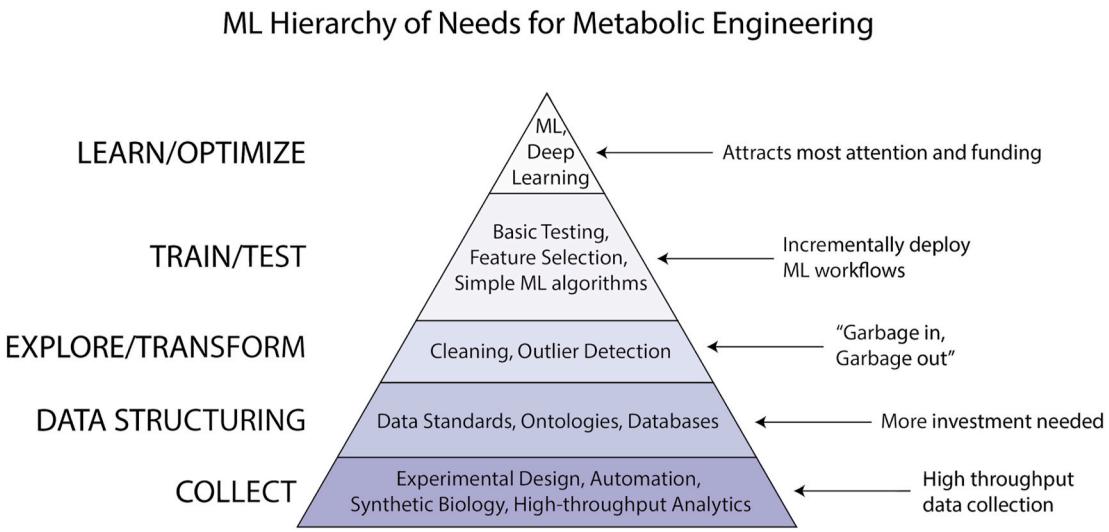


Fig. 12. The hierarchy of needs for leveraging machine learning in metabolic engineering. It is futile to rely on machine learning to guide metabolic engineering without first establishing the basic infrastructure that it depends on. The very base consists on creating the infrastructure to physically collect large amounts of high quality data. The next step is to have the databases, standard and ontologies to structure and store the data appropriately. Data cleaning and outlier detection follow. The base for simple machine learning algorithms (linear regression), feature selection and algorithm training is at this point set. It is only at this stage that sophisticated machine learning and deep learning can significantly improve the metabolic engineering practice. Adapted from Rogati (Rogati, 2017).

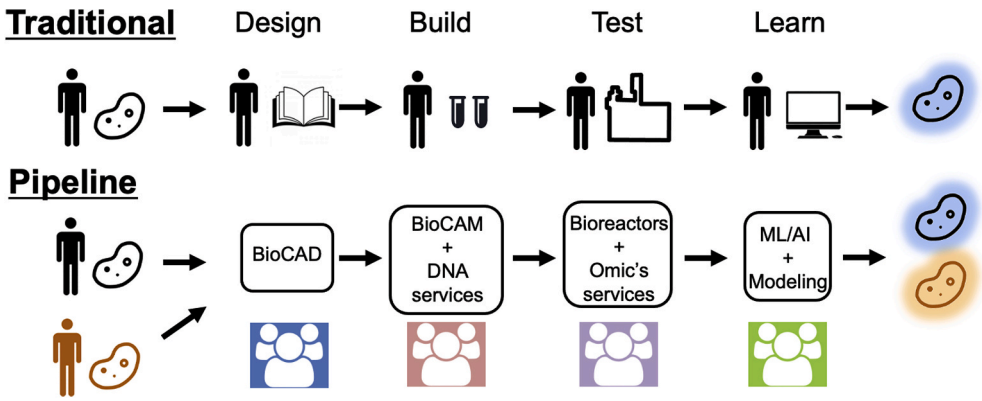


Fig. 13. Traditional metabolic engineering vs pipeline. The traditional metabolic engineering process involves a single researcher doing all phases of the project from pathway choice to strain building, fermentation, and data analysis. The pipeline approach instead focuses resources on creating a single, flexible, semi automated, pipeline consisting of different connected services supported by specialized teams. The pipeline approach favors repeatability, data quality and the stream of data required by machine learning. Furthermore, the pipeline allows for simultaneous development of multiple strains, so knowledge obtained from one design can immediately be leveraged for all others. BioCAD: Biological Computer-Aided Design; BioCAM: Biological Computer-Aided Manufacturing of Synthetic DNA (Oberortner et al., 2020).

biosynthetic pathways (Carbonell et al., 2018). In general, we expect machine learning, biochemical analytical techniques and automation to follow a path of parallel development and keep symbiotically interacting in pipelines so that machine learning will be a pillar in every step of biosystems design (Volk et al., 2020).

Automating metabolic engineering often involves multiplexing the bioengineering efforts to parallelize a set of combinatorial experiments. For example, digital microfluidics is a high-throughput liquid handling technique able to quickly automate diverse biological experiments at micro and nanoscales, thus accelerating the DBTL cycles and making synthetic biology programmable (Gach et al., 2017; Kothamachu et al., 2020). The combination of microfluidics with nanofluidics and optoelectronics has been used for the automated growth and analysis of thousands of cell lines in parallel on a single chip (Le et al., 2018). Another implementation of these technologies enables the parallel construction and optical screening of tens of thousands of synthetic microbial communities per day (Kehe et al., 2019). Other efforts focus on parallelizing experiments while maintaining their potential to efficiently scale up. That is the case of automated workflows for media optimization, induction profiling, or microbial bioprocess optimization leveraging the Biolector, a microtiter plate-based cultivation device (Rohe et al., 2012). Another example of productive scale up involves the Automated Microscale Bioreactor (Ambr 250), which can generate comparable cell growth and protein production profiles comparable to those obtained in 1000-L bioreactor industrial scale fermentations (Xu et al., 2017).

Some automation technologies focus on the human-to-system interface and embrace AI to further accelerate the experimentation processes. Robotic Process Automation (RPA) is an alternative approach that provides agents (bots) that operate on user interfaces in the way a human would do (van der Aalst et al., 2018). RPA is meant to replace humans in repetitive work that is frequent enough to make fully automation economically feasible. Intelligent RPA (IRPA) is the current effort to fuse RPA with advanced AI methods to drastically extend its scope (Syed et al., 2020). Combining experimentation platforms with AI to accelerate experimental research is at the core of the so-called self-driving laboratories (Häse et al., 2019), which typically use multi-objective optimization techniques (Häse et al., 2018) and iterate over the design, execution, and learning steps of the experiments with complete autonomy (MacLeod et al., 2020). The use of AI-driven automation

technologies with hardware robotics represents a step further. In that sense, a very recent automation effort has used state of the art robotics to completely move the focus from automating the instruments to automate the researcher (Burger et al., 2020).

Cloud labs are tools based on cloud technologies (Xu, 2012) that allow a scientist to remotely conduct biological research through robotic control, by using a high-level interface to ease the requirement for any programming knowledge. As an added benefit, researchers usually get all the intermediate and final results stored on the cloud in digital formats prepared for downstream analysis by local or cloud computing (Mell and Grance, 2011). Past years have seen an emergence of cloud labs (Check Hayden, 2014) and tools (Bates et al., 2017), which has been recently boosted by the social distancing requirements of the SARS-CoV2 pandemic. Thus, a remote or distributed manner of experimentation is arising as an alternative to the local or centralized classic model.

Interestingly, COVID-19 has also promoted the do-it-yourself (DIY) approach to lab automation. In 1981, IBM introduced the personal computer (PC), democratizing computing with an open architecture model (Miller, 1989; O'Regan, 2012), and producing a paradigm shift. An equivalent shift for automation seems to be in motion, due to the combination of the maturity of the open source model with the rise of free open scientific hardware (FOSH), now accelerated by the SARS-CoV2 pandemic (Maia Chagas et al., 2020). This trend in automation emerged from the use of 3D printing for a growing number of scientific and engineering applications in the laboratory (Silver, 2019). This pure DIY approach has already produced successful high-throughput automation platforms for bioengineering (Wong et al., 2018) and is susceptible to improvement by machine learning techniques such as deep reinforcement learning (Treloar et al., 2020). Some companies, such as Opentrons, are taking advantage of this new market niche and are providing open automation solutions halfway between the extreme DIY and the classical automation (May, 2019) based on proprietary and expensive equipment and consumables (Maia Chagas, 2018). There are already some open automation systems built on top of Opentrons liquid handling robots and devoted to synthetic biology applications, such as the DNA-BOT for automated DNA assembly (Storch et al., 2020). On the other hand, a low-cost modular FOSH liquid handler has been recently combined with machine learning for automatizing droplet experiments with AI-enabled computer vision (Faiña et al., 2020). Considering all of the above, it seems that there are technological developments quickly converging towards open hardware and software automation solutions based on machine learning and specific for synthetic biology.

5.3. Novel machine learning techniques to watch

Deep learning, with applications using several interconnected layers of ANNs (see Figs. 7 and 8), has been the subfield of machine learning driving the recent boost of AI. The number of such layers of ANNs is the depth of the neural network. With increasing depths, deep neural networks often have a large number of parameters. For example, a state-of-the-art system for natural language processing (NLP) (Manning, 1999), the autoregressive language model GPT-3 (Brown et al., 2020), has almost one hundred layers and 175 billion parameters. These DL systems are intricate black boxes making decisions that are not easily interpretable from a human perspective. If a prediction deviates from the expected answer, it is generally not easy to understand why it failed, or how to correct the issue. These algorithms are only as good as the data they are trained with, so biases in the data have a significant impact on the predictions (Rodrigues, 2020), with a growing need for developing bias quantification metrics along with methods for overfitting detection and data debiasing (Ellingson et al., 2020).

The lack of interpretability has prevented machine learning in general and DL in particular from expanding in some fields that require trust in the underlying technology, such as in defense, healthcare, and other sensitive applications. Different novel approaches are under active research to overcome this critical drawback. Some of these try to make classic machine learning methods such as random forests more interpretable without a loss of efficacy (Basu et al., 2018). Another technique is even able to extract explicit physical relations by applying symbolic regression to components of a Graph Neural Network (GNN) trained by encouraging sparse latent representations in a supervised setting (Cranmer et al., 2020). In drug discovery, the lack of transparent and reproducible workflows has hindered widespread adoption of machine learning models, but this is being solved by novel scalable pipelines with traceable models stressing uncertainty quantification (Minnich et al., 2020).

In 2017, DARPA launched its explainable artificial intelligence (XAI) program as a comprehensive strategy to tackle the machine learning interpretability problem. DARPA's XAI aims at developing superior AI systems able to have a symbiotic relationship with humans (Gunning and Aha, 2019). A recent evolution on top of the XAI paradigm is the concept of Responsible AI (Barredo Arrieta et al., 2020), which imposes further constraints on the implementation of AI systems, like transparency, accountability, and ethics. However, the movement towards greater interpretability involves significant trade-offs in terms of performance, with a toll on fidelity and accuracy (Gunning et al., 2019). Ultimately, that compromise could be rendered unnecessary by advances in high performance computing (HPC), since AI and HPC are converging in approaching the exascale era (Gwynne, 2019). Indeed, the joint effort of XAI developments with exascale computing, by bridging the gaps between cutting-edge research and sustainable policies, could pave the way for designing practical solutions to global challenges such as climate change (Streich et al., 2020).

XAI has numerous applications in unraveling the profound mechanics of natural or artificial systems, such as the molecular mechanisms underlying genome biology (Basu et al., 2018). A related DL framework is the use of physics-informed neural networks (PINN), which are trained to solve supervised forward and inverse problems involving nonlinear partial differential equations (PDE), thus supporting the union of data-driven and mathematical models (Raissi et al. 2019, 2020). In the case of very noisy data, Bayesian Neural Networks can be combined with PINNs (called then B-PINNs) to both avoid overfitting and quantify uncertainty (Yang et al., 2020).

Transfer learning (TL) (Ando and Zhang, 2005; Caruana, 1997; Pan and Yang, 2010) is the technique of knowledge transfer from a domain with enough training data to another related domain of interest that lacks such data. This transfer considerably enhances the learning performance by avoiding costly data-labeling efforts. This area is under rapid expansion but already offers many consolidated models from which to choose carefully depending on the type of application and its data (Zhuang et al., 2020). For example, TL has been used to tackle the problem of predicting associations between genotype and phenotype (Petegrosso et al., 2017). Clearly, TL could be key for different metabolic engineering projects if used to transfer predictive capabilities from one organism to another, avoiding the cost and time expenses of getting large multiomics data sets from scratch. Finally, TL can be combined with XAI methods, for instance, for gathering pathway and metabolic information in model organisms and translate it to others so as to get comprehensive genome-scale metabolic models in an efficient manner.

6. Conclusion

Machine learning provides an opportunity to make metabolic engineering more predictable and efficient. In this review, we have attempted to provide an introduction to this discipline in terms that are relatable to metabolic engineers, as well as providing illustrative examples along the traditional phases of metabolic engineering (from pathway choice and construction to scaling). We have also included practical advice including library suggestions and experimental design recommendations. Finally, we have examined the perspectives for this

combination of disciplines, which are particularly relevant and difficult to predict, given the current explosive growth of both machine learning and synthetic biology.

In our opinion, metabolic engineering could take two courses in the future: incremental or disruptive. In one, traditional methods prevail, progress is incremental, and more molecules are arduously brought into commercial use at an increasing rate. In another one, metabolic engineering fully embraces and integrates the possibilities afforded by automation and machine learning. This choice leads to a disruptive change that makes production of new molecules a relatively easy task dwarfed by the more ambitious goals enabled by the new predictive capabilities. Metabolic engineering is used to engineer microbiomes, create new biomaterials, provide fundamental understanding of emergent properties and evolution, and suggest new artificial intelligence approaches.

The fundamental challenges for the disruptive path involve enabling streams of high-quality data, developing new algorithms to integrate the advantages of data-driven and mechanistic approaches, and fully leveraging novel tools in machine learning and synthetic biology. In our view, solving these challenges is only possible through a multidisciplinary collaboration of scientists including metabolic engineers, biochemists, microbiologists, computer scientists, electrical engineers, chemical engineers, mathematicians, statisticians, and physicists, among others. We hope to have provided in this review a helpful resource for that multidisciplinary collaboration.

CRediT authorship contribution statement

Christopher E. Lawson: Writing - review & editing, Writing - original draft, All authors contributed to the writing of this review. **Jose Manuel Martí:** Writing - review & editing, Writing - original draft, All authors contributed to the writing of this review. **Tijana Radivojevic:** Writing - review & editing, Writing - original draft, All authors contributed to the writing of this review. **Sai Vamshi R. Jonnalagadda:** Writing - review & editing, Writing - original draft, All authors contributed to the writing of this review. **Reinhard Gentz:** Writing - review & editing, Writing - original draft, All authors contributed to the writing of this review. **Nathan J. Hillson:** Writing - review & editing, Writing - original draft, All authors contributed to the writing of this review. **Sean Peisert:** Writing - review & editing, Writing - original draft, All authors contributed to the writing of this review. **Joonhoon Kim:** Writing - review & editing, Writing - original draft, All authors contributed to the writing of this review. **Blake A. Simmons:** Writing - review & editing, Writing - original draft, All authors contributed to the writing of this review. **Christopher J. Petzold:** Writing - review & editing, Writing - original draft, All authors contributed to the writing of this review. **Steven W. Singer:** Writing - review & editing, Writing - original draft, All authors contributed to the writing of this review. **Aindrila Mukhopadhyay:** Writing - review & editing, Writing - original draft, All authors contributed to the writing of this review. **Deepti Tanjore:** Writing - review & editing, Writing - original draft, All authors contributed to the writing of this review. **Joshua G. Dunn:** Writing - review & editing, Writing - original draft, All authors contributed to the writing of this review. **Hector Garcia Martin:** Writing - review & editing, Writing - original draft, All authors contributed to the writing of this review.

Acknowledgements

This work was part of the the Agile BioFoundry (http://agilebiofoundry.org) and the DOE Joint BioEnergy Institute (http://www.jbei.org), supported by the U. S. Department of Energy, Energy Efficiency and Renewable Energy, Bioenergy Technologies Office, and the Office of Science, through contract DE-AC02-05CH11231 between Lawrence Berkeley National Laboratory and the U. S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan). This research is also supported by the Basque Government through the BERC 2014–2017 program and by the Spanish Ministry of Economy and Competitiveness MINECO: BCAM Severo Ochoa excellence accreditation SEV-2013-0323.

References

Aach, J., Mali, P., Church, G.M., 2014. CasFinder: flexible algorithm for identifying specific Cas9 targets in genomes. BioRxiv. https://doi.org/10.1101/005074.

Ajikumar, P.K., Xiao, W.-H., Tyo, K.E.J., Wang, Y., Simeon, F., Leonard, E., Mucha, O., Phon, T.H., Pfeifer, B., Stephanopoulos, G., 2010. Isoprenoid pathway optimization for Taxol precursor overproduction in *Escherichia coli*. Science 330, 70–74. https://doi.org/10.1126/science.1191652.

Alderson, R.G., De Ferrari, L., Mavridis, L., McDonagh, J.L., Mitchell, J.B.O., Nath, N., 2012. Enzyme informatics. Curr. Top. Med. Chem. 12, 1911–1923. https://doi.org/10.2174/156802612804547353.

Alley, E.C., Khimulya, G., Biswas, S., AlQuraishi, M., Church, G.M., 2019. Unified rational protein engineering with sequence-based deep representation learning. Nat. Methods 16, 1315–1322. https://doi.org/10.1038/s41592-019-0598-1.

Alonso-Gutierrez, J., Kim, E.-M., Batth, T.S., Cho, N., Hu, Q., Chan, L.J.G., Petzold, C.J., Hillson, N.J., Adams, P.D., Keasling, J.D., Garcia Martin, H., Lee, T.S., 2015. Principal component analysis of proteomics (PCAP) as a tool to direct metabolic engineering. Metab. Eng. 28, 123–133. https://doi.org/10.1016/j.ymben.2014.11.011.

AlQuraishi, M., 2019. AlphaFold at CASP13. Bioinformatics 35, 4862–4865. https://doi.org/10.1093/bioinformatics/btz422.

Amidi, A., Amidi, S., Vlachakis, D., Megalooikonomou, V., Paragios, N., Zacharaki, E.I., 2018. EnzyNet: enzyme classification using 3D convolutional neural networks on spatial representation. PeerJ 6, e4750.

Ando, R.K., Zhang, T., 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. J. Mach. Learn. Res. 6, 1817–1853.

Andreozzi, S., Miskovic, L., Hatzimanikatis, V., 2016. iSCHRUNK–In silico approach to characterization and reduction of uncertainty in the kinetic models of genome-scale metabolic networks. Metab. Eng. 33, 158–168. https://doi.org/10.1016/j.ymben.2015.10.002.

Armenteros, J.J.A., Tsirigos, K.D., Sønderby, C.K., Petersen, T.N., Winther, O., Brunak, S., von Heijne, G., Nielsen, H., 2019. SignalP 5.0 improves signal peptide predictions using deep neural networks. Nat. Biotechnol. 37, 420–423. https://doi.org/10.1038/s41587-019-0036-z.

Ay, A., Arnosti, D.N., 2011. Mathematical modeling of gene expression: a guide for the perplexed biologist. Crit. Rev. Biochem. Mol. Biol. 46, 137–151. https://doi.org/10.3109/10409238.2011.556597.

Bairoch, A., Apweiler, R., 2000. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. Nucleic Acids Res. 28, 45–48. https://doi.org/10.1093/nar/28.1.45.

Bakshi, B.R., Stephanopoulos, G., 1994. Representation of process trends—III. Multiscale extraction of trends from process data. Comput. Chem. Eng. 18, 267–302. https://doi.org/10.1016/0098-1354(94)85028-3.

Bao, Z., HamediRad, M., Xue, P., Xiao, H., Tasan, I., Chao, R., Liang, J., Zhao, H., 2018. Genome-scale engineering of Saccharomyces cerevisiae with single-nucleotide precision. Nat. Biotechnol. 36, 505–508. https://doi.org/10.1038/nbt.4132.

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., Herrera, F., 2020. Explainable Artificial Intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. Inf. Fusion 58, 82–115. https://doi.org/10.1016/j.inffus.2019.12.012.

Bassalo, M.C., Garst, A.D., Choudhury, A., Grau, W.C., Oh, E.J., Spindler, E., Lipscomb, T., Gill, R.T., 2018. Deep scanning lysine metabolism in *Escherichia coli*. Mol. Syst. Biol. 14, e8371 https://doi.org/10.15252/msb.20188371.

Basu, S., Kumbier, K., Brown, J.B., Yu, B., 2018. Iterative random forests to discover predictive and stable high-order interactions. Proc. Natl. Acad. Sci. U.S.A. 115, 1943–1948. https://doi.org/10.1073/pnas.1711236115.

Bates, M., Berliner, A.J., Lachoff, J., Jaschke, P.R., Groban, E.S., 2017. Wet lab accelerator: a web-based application democratizing laboratory automation for synthetic biology. ACS Synth. Biol. 6, 167–171. https://doi.org/10.1021/acssynbio.6b00108.

Begoli, E., Bhattacharya, T., Kusnezov, D., 2019. The need for uncertainty quantification in machine-assisted medical decision making. Nat. Mach. Intell. 1, 20–23. https://doi.org/10.1038/s42256-018-0004-1.

Benson, D.A., Karsch-Mizrachi, I., Lipman, D.J., Ostell, J., Sayers, E.W., 2011. GenBank. Nucleic Acids Res. 39, D32–D37. https://doi.org/10.1093/nar/gkq1079.

Biswas, S., Khimulya, G., Alley, E.C., Esvelt, K.M., Church, G.M., 2020. Low-N protein engineering with data-efficient deep learning. BioRxiv. https://doi.org/10.1101/2020.01.23.917682.

Boock, J.T., Gupta, A., Prather, K.L., 2015. Screening and modular design for metabolic pathway optimization. Current Opinion in Biotechnology 36, 189–198.

Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Amodei, D., 2020. Language Models Are Few-Shot Learners. arXiv. https://arxiv.org/abs/2005.14165.

Burger, B., Maffettone, P.M., Gusev, V.V., Aitchison, C.M., Bai, Y., Wang, X., Li, X., Alston, B.M., Li, B., Clowes, R., Rankin, N., Harris, B., Sprick, R.S., Cooper, A.I., 2020. A mobile robotic chemist. Nature 583, 237–241. https://doi.org/10.1038/s41586-020-2442-2.

Carbonell, P., Faulon, J.-L., 2010. Molecular signatures-based prediction of enzyme promiscuity. Bioinformatics 26, 2012–2019. https://doi.org/10.1093/bioinformatics/btq317.

Carbonell, P., Jervis, A.J., Robinson, C.J., Yan, C., Dunstan, M., Swainston, N., Vinaixa, M., Hollywood, K.A., Currin, A., Rattray, N.J.W., Taylor, S., Spiess, R., Sung, R., Williams, A.R., Fellows, D., Stanford, N.J., Mulherin, P., Le Feuvre, R., Barran, P., Goodacre, R., Scrutton, N.S., 2018. An automated Design-Build-Test-Learn pipeline for enhanced microbial production of fine chemicals. Commun. Biol. 1, 66. https://doi.org/10.1038/s42003-018-0076-9.

Carbonell, P., Radivojevic, T., García Martín, H., 2019. Opportunities at the intersection of synthetic biology, machine learning, and automation. ACS Synth. Biol. 8, 1474–1477. https://doi.org/10.1021/acssynbio.8b00540.

Caruana, R., 1997. Multitask Learning. Springer Science and Business Media LLC. https://doi.org/10.1023/a:1007379606734.

Chakrabarti, A., Miskovic, L., Soh, K.C., Hatzimanikatis, V., 2013. Towards kinetic modeling of genome-scale metabolic networks without sacrificing stoichiometric, thermodynamic and physiological constraints. Biotechnol. J. 8, 1043–1057. https://doi.org/10.1002/biot.201300091.

Chakrabarti, K., Keogh, E., Mehrotra, S., Pazzani, M., 2002. Locally adaptive dimensionality reduction for indexing large time series databases. ACM Trans. Database Syst. 27, 188–228. https://doi.org/10.1145/568518.568520.

Chao, R., Mishra, S., Si, T., Zhao, H., 2017. Engineering biological systems using automated biofoundries. Metab. Eng. 42, 98–108. https://doi.org/10.1016/j.ymben.2017.06.003.

Charaniya, S., Hu, W.-S., Karypis, G., 2008. Mining bioprocess data: opportunities and challenges. Trends Biotechnol. 26, 690–699. https://doi.org/10.1016/j.tibtech.2008.09.003.

Chari, R., Yeo, N.C., Chavez, A., Church, G.M., 2017. sgRNA scorer 2.0: a species-independent model to predict CRISPR/cas9 activity. ACS Synth. Biol. 6, 902–904. https://doi.org/10.1021/acssynbio.6b00343.

Check Hayden, E., 2014. The automated lab. Nature 516, 131–132. https://doi.org/10.1038/516131a.

Chen, Y., Guenther, J.M., Gin, J.W., Chan, L.J.G., Costello, Z., Ogorzalek, T.L., Tran, H. M., Blake-Hedges, J.M., Keasling, J.D., Adams, P.D., García Martín, H., Hillson, N.J., Petzold, C.J., 2019. Automated "cells-to-peptides" sample preparation workflow for high-throughput, quantitative proteomic assays of microbes. J. Proteome Res. 18, 3752–3761. https://doi.org/10.1021/acs.jproteome.9b00455.

Cheung, J.T.Y., Stephanopoulos, G., 1990a. Representation of process trends—Part I. A formal representation framework. Comput. Chem. Eng. 14, 495–510. https://doi.org/10.1016/0098-1354(90)87023-I.

Cheung, J.T.Y., Stephanopoulos, G., 1990b. Representation of process trends—Part II. The problem of scale and qualitative scaling. Comput. Chem. Eng. 14, 511–539. https://doi.org/10.1016/0098-1354(90)87024-J.

Chin, H., Kim, J., Kim, Y., Shin, J., Yi, M.Y., 2018. Explicit content detection in music lyrics using machine learning. In: 2018 IEEE International Conference on Big Data and Smart Computing (BigComp). Presented at the 2018 IEEE International Conference on Big Data and Smart Computing. BigComp), IEEE, pp. 517–521. https://doi.org/10.1109/BigComp.2018.00085.

Choi, K.R., Jang, W.D., Yang, D., Cho, J.S., Park, D., Lee, S.Y., 2019. Systems metabolic engineering strategies: integrating systems and synthetic biology with metabolic engineering. Trends Biotechnol. 37, 817–837. https://doi.org/10.1016/j.tibtech.2019.01.003.

Chuai, G., Ma, H., Yan, J., Chen, M., Hong, N., Xue, D., Zhou, C., Zhu, C., Chen, K., Duan, B., Gu, F., Qu, S., Huang, D., Wei, J., Liu, Q., 2018. DeepCRISPR: optimized CRISPR guide RNA design by deep learning. Genome Biol. 19, 80. https://doi.org/10.1186/s13059-018-1459-4.

Chubukov, V., Mukhopadhyay, A., Petzold, C.J., Keasling, J.D., Martín, H.G., 2016. Synthetic and systems biology for microbial production of commodity chemicals. NPJ Syst. Biol. Appl. 2, 16009. https://doi.org/10.1038/npjsba.2016.9.

Ciaburro, G., 2017. Matlab for Machine Learning. Packt Publishing Ltd.

Claudel-Renard, C., Chevalet, C., Faraut, T., Kahn, D., 2003. Enzyme-specific profiles for genome annotation: PRIAM. Nucleic Acids Res. 31, 6633–6639. https://doi.org/10.1093/nar/gkg847.

Clauwaert, J., Menschaert, G., Waegeman, W., 2019. DeepRibo: a neural network for precise gene annotation of prokaryotes by combining ribosome profiling signal and binding site patterns. Nucleic Acids Res. 47, e36. https://doi.org/10.1093/nar/gkz061.

Coleman, M.C., Buck, K.K.S., Block, D.E., 2003. An integrated approach to optimization of Escherichia coli fermentations using historical data. Biotechnol. Bioeng. 84, 274–285. https://doi.org/10.1002/bit.10719.

Costa, R.S., Machado, D., Rocha, I., Ferreira, E.C., 2010. Hybrid dynamic modeling of Escherichia coli central metabolic network combining Michaelis-Menten and approximate kinetic equations. Biosystems 100, 150–157. https://doi.org/10.1016/j.biosystems.2010.03.001.

Costello, Z., Garcia Martin, H., 2019. How to Hallucinate Functional Proteins. arXiv. https://arxiv.org/abs/1903.00458.

Costello, Z., Martin, H.G., 2018. A machine learning approach to predict metabolic pathway dynamics from time-series multiomics data. NPJ Syst. Biol. Appl. 4, 19. https://doi.org/10.1038/s41540-018-0054-3.

Cranmer, M., Sanchez-Gonzalez, A., Battaglia, P., Xu, R., Cranmer, K., Spergel, D., Ho, S., 2020. Discovering Symbolic Models from Deep Learning with Inductive Biases. arXiv. https://arxiv.org/abs/2006.11287.

Crater, J.S., Lievense, J.C., 2018. Scale-up of industrial microbial processes. FEMS Microbiol. Lett. 365 https://doi.org/10.1093/femsle/fny138.

Culley, C., Vijayakumar, S., Zampieri, G., Angione, C., 2020. A mechanism-aware and multiomic machine-learning pipeline characterizes yeast cell growth. Proc. Natl. Acad. Sci. U.S.A. 117, 18869–18879. https://doi.org/10.1073/pnas.2002959117.

Dalkiran, A., Rifaioglu, A.S., Martin, M.J., Cetin-Atalay, R., Atalay, V., Doğan, T., 2018. ECPred: a tool for the prediction of the enzymatic functions of protein sequences based on the EC nomenclature. BMC Bioinf. 19, 334. https://doi.org/10.1186/s12859-018-2368-y.

Delépine, B., Duigou, T., Carbonell, P., Faulon, J.-L., 2018. RetroPath2.0: a retrosynthesis workflow for metabolic engineers. Metab. Eng. 45, 158–170. https://doi.org/10.1016/j.ymben.2017.12.002.

Denby, C.M., Li, R.A., Vu, V.T., Costello, Z., Lin, W., Chan, L.J.G., Williams, J., Donaldson, B., Bamforth, C.W., Petzold, C.J., Scheller, H.V., Martin, H.G., Keasling, J.D., 2018. Industrial brewing yeast engineered for the production of primary flavor determinants in hopped beer. Nat. Commun. 9, 965. https://doi.org/10.1038/s41467-018-03293-x.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. Presented at the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops). IEEE, pp. 248–255. https://doi.org/10.1109/CVPR.2009.5206848.

Doench, J.G., Fusi, N., Sullender, M., Hegde, M., Vaimberg, E.W., Donovan, K.F., Smith, I., Tothova, Z., Wilen, C., Orchard, R., Virgin, H.W., Listgarten, J., Root, D.E., 2016. Optimized sgRNA design to maximize activity and minimize off-target effects of CRISPR-Cas9. Nat. Biotechnol. 34, 184–191. https://doi.org/10.1038/nbt.3437.

Dolgin, E., 2019. Scientists brew cannabis using hacked beer yeast. Nature. https://doi.org/10.1038/d41586-019-00714-9.

Doudna, J.A., Charpentier, E., 2014. Genome editing. The new frontier of genome engineering with CRISPR-Cas9. Science 346, 1258096. https://doi.org/10.1126/science.1258096.

Duarte, F., Ratti, C., 2018. The impact of autonomous vehicles on cities: a review. J. Urban Technol. 25, 3–18. https://doi.org/10.1080/10630732.2018.1493883.

Ellingson, S.R., Davis, B., Allen, J., 2020. Machine learning and ligand binding predictions: a review of data, methods, and obstacles. Biochim. Biophys. Acta Gen. Subj. 1864, 129545. https://doi.org/10.1016/j.bbagen.2020.129545.

Esvelt, K.M., Wang, H.H., 2013. Genome-scale engineering for systems and synthetic biology. Mol. Syst. Biol. 9, 641. https://doi.org/10.1038/msb.2012.66.

Faiña, A., Nejati, B., Stoy, K., 2020. EvoBot: an open-source, modular, liquid handling robot for scientific experiments. Appl. Sci. 10, 814. https://doi.org/10.3390/app10030814.

Faulon, J.-L., Misra, M., Martin, S., Sale, K., Sapra, R., 2008. Genome scale enzyme-metabolite and drug-target interaction predictions using the signature molecular descriptor. Bioinformatics 24, 225–233. https://doi.org/10.1093/bioinformatics/btm580.

Finn, R.D., Clements, J., Eddy, S.R., 2011. HMMER web server: interactive sequence similarity searching. Nucleic Acids Res. 39, W29–W37. https://doi.org/10.1093/nar/gkr367.

Fox, R.J., Davis, S.C., Mundorff, E.C., Newman, L.M., Gavrilovic, V., Ma, S.K., Chung, L. M., Ching, C., Tam, S., Muley, S., Grate, J., Gruber, J., Whitman, J.C., Sheldon, R.A., Huisman, G.W., 2007. Improving catalytic function by ProSAR-driven enzyme evolution. Nat. Biotechnol. 25, 338–344. https://doi.org/10.1038/nbt1286.

Gach, P.C., Iwai, K., Kim, P.W., Hillson, N.J., Singh, A.K., 2017. Droplet microfluidics for synthetic biology. Lab Chip 17, 3388–3400. https://doi.org/10.1039/c7lc00576h.

Gardner, T.S., 2013. Synthetic biology: from hype to impact. Trends Biotechnol. 31, 123–125. https://doi.org/10.1016/j.tibtech.2013.01.018.

Garst, A.D., Bassalo, M.C., Pines, G., Lynch, S.A., Halweg-Edwards, A.L., Liu, R., Liang, L., Wang, Z., Zeitoun, R., Alexander, W.G., Gill, R.T., 2017. Genome-wide mapping of mutations at single-nucleotide resolution for protein, metabolic and genome engineering. Nat. Biotechnol. 35, 48–55. https://doi.org/10.1038/nbt.3718.

Gebrekidan, M.T., Knipfer, C., Braeuer, A.S., 2020. Vector casting for noise reduction. J. Raman Spectrosc. 51, 731–743. https://doi.org/10.1002/jrs.5835.

"Geltor unveils first biodesigned human collagen for skincare market", 2019. PRnewswire. https://www.prnewswire.com/news-releases/geltor-unveils-first-biodesigned-human-collagen-for-skincare-market-300819885.html.

George, K.W., Thompson, M.G., Kang, A., Baidoo, E., Wang, G., Chan, L.J.G., Adams, P. D., Petzold, C.J., Keasling, J.D., Lee, T.S., 2015. Metabolic engineering for the high-yield production of isoprenoid-based C_5 alcohols in E. coli. Sci. Rep. 5, 11128. https://doi.org/10.1038/srep11128.

Géron, A., 2019. Hands-on Machine Learning with Scikit-Learn, Keras, and Tensorflow: Concepts, Tools, and Techniques to Build Intelligent Systems, second ed. O'reilly Media.

Gessulat, S., Schmidt, T., Zolg, D.P., Samaras, P., Schnatbaum, K., Zerweck, J., Knaute, T., Rechenberger, J., Delanghe, B., Huhmer, A., Reimer, U., Ehrlich, H.-C., Aiche, S., Kuster, B., Wilhelm, M., 2019. Prosit: proteome-wide prediction of peptide tandem mass spectra by deep learning. Nat. Methods 16, 509–518. https://doi.org/10.1038/s41592-019-0426-7.

Gilbert, L.A., Horlbeck, M.A., Adamson, B., Villalta, J.E., Chen, Y., Whitehead, E.H., Guimaraes, C., Panning, B., Ploegh, H.L., Bassik, M.C., Qi, L.S., Kampmann, M.,

Weissman, J.S., 2014. Genome-scale CRISPR-mediated control of gene repression and activation. Cell 159, 647–661. https://doi.org/10.1016/j.cell.2014.09.029.
Gonzalez-Beltran, A., Maguire, E., Georgiou, P., Sansone, S.-A., Rocca-Serra, P., 2013. Bio-GraphIIn: a graph-based, integrative and semantically-enabled repository for life science experimental data. EMBnet j 19, 46. https://doi.org/10.14806/ej.19.B.728.
Groher, A.-C., Jager, S., Schneider, C., Groher, F., Hamacher, K., Suess, B., 2019. Tuning the performance of synthetic riboswitches using machine learning. ACS Synth. Biol. 8, 34–44. https://doi.org/10.1021/acssynbio.8b00207.
Gunning, D., 2016. Explainable Artificial Intelligence (XAI) (No. DARPA-BAA-16-53). Defense Advanced Research Projects Agency.
Gunning, D., Aha, D., 2019. Darpa's explainable artificial intelligence (XAI) program. AI Mag. 40, 44–58. https://doi.org/10.1609/aimag.v40i2.2850.
Gunning, D., Stefik, M., Choi, J., Miller, T., Stumpf, S., Yang, G.-Z., 2019. XAI—explainable artificial intelligence. Sci. Robot. 4, eaay7120 https://doi.org/10.1126/scirobotics.aay7120.
Gwynne, P., 2019. Exascale supercomputer intiative launched. Phys. World 32. https://doi.org/10.1088/2058-7058/32/5/12, 11–11.
Hahn, J., 2019. Spiber and North Face Japan create first readily-available spider silk jacket. https://www.dezeen.com/2019/10/24/spiber-moon-parka-spider-silkthe-north-face-japan/ accessed 2.19.20.
HamediRad, M., Chao, R., Weisberg, S., Lian, J., Sinha, S., Zhao, H., 2019. Towards a fully automated algorithm driven platform for biosystems design. Nat. Commun. 10, 5150. https://doi.org/10.1038/s41467-019-13189-z.
Ham, T.S., Dmytriv, Z., Plahar, H., Chen, J., Hillson, N.J., Keasling, J.D., 2012. Design, implementation and practice of JBEI-ICE: an open source biological part registry platform and tools. Nucleic Acids Res. 40, e141. https://doi.org/10.1093/nar/gks531.
Hanson, Chris, 2013. "Amyris ships first commerical order of Biofene from Brazil plant". Biomass Magazine. http://biomassmagazine.com/articles/8610/amyris-ships-first-commerical-order-of-biofene-from-brazil-plant.
Haringa, C., Tang, W., Deshmukh, A.T., Xia, J., Reuss, M., Heijnen, J.J., Mudde, R.F., Noorman, H.J., 2016. Euler-Lagrange computational fluid dynamics for (bio)reactor scale down: an analysis of organism lifelines. Eng. Life Sci. 16, 652–663. https://doi.org/10.1002/elsc.201600061.
Haringa, C., Tang, W., Wang, G., Deshmukh, A.T., van Winden, W.A., Chu, J., van Gulik, W.M., Heijnen, J.J., Mudde, R.F., Noorman, H.J., 2017. Computational fluid dynamics simulation of an industrial P. chrysogenum fermentation with a coupled 9-pool metabolic model: towards rational scale-down and design optimization. Chem. Eng. Sci. 175, 12–24. https://doi.org/10.1016/j.ces.2017.09.020.
Häse, F., Roch, L.M., Aspuru-Guzik, A., 2018. Chimera: enabling hierarchy based multi-objective optimization for self-driving laboratories. Chem. Sci. 9, 7642–7655. https://doi.org/10.1039/c8sc02239a.
Häse, F., Roch, L.M., Aspuru-Guzik, A., 2019. Next-generation experimentation with self-driving laboratories. Trends in Chemistry 1, 282–291. https://doi.org/10.1016/j.trechm.2019.02.007.
Hastings, A., Byers, J.E., Crooks, J.A., Cuddington, K., Jones, C.G., Lambrinos, J.G., Talley, T.S., Wilson, W.G., 2007. Ecosystem engineering in space and time. Ecol. Lett. 10, 153–164. https://doi.org/10.1111/j.1461-0248.2006.00997.x.
Haug, K., Salek, R.M., Conesa, P., Hastings, J., de Matos, P., Rijnbeek, M., Mahendraker, T., Williams, M., Neumann, S., Rocca-Serra, P., Maguire, E., González-Beltrán, A., Sansone, S.-A., Griffin, J.L., Steinbeck, C., 2013. MetaboLights–an open-access general-purpose repository for metabolomics studies and associated meta-data. Nucleic Acids Res. 41, D781–D786. https://doi.org/10.1093/nar/gks1004.
Heckmann, D., Lloyd, C.J., Mih, N., Ha, Y., Zielinski, D.C., Haiman, Z.B., Desouki, A.A., Lercher, M.J., Palsson, B.O., 2018. Machine learning applied to enzyme turnover numbers reveals protein structural correlates and improves metabolic models. Nat. Commun. 9, 5252. https://doi.org/10.1038/s41467-018-07652-6.
Heigwer, F., Kerr, G., Boutros, M., 2014. E-CRISP: fast CRISPR target site identification. Nat. Methods 11, 122–123. https://doi.org/10.1038/nmeth.2812.
Heijnen, J.J., 2005. Approximative kinetic formats used in metabolic network modeling. Biotechnol. Bioeng. 91, 534–545. https://doi.org/10.1002/bit.20558.
Heinrich, R., Schuster, S., 1996. The Regulation of Cellular Systems. Springer US, Boston, MA. https://doi.org/10.1007/978-1-4613-1161-4.
Heo, L., Feig, M., 2020. High-accuracy protein structures by combining machine-learning with physics-based refinement. Proteins 88, 637–642. https://doi.org/10.1002/prot.25847.
Hillson, N., Caddick, M., Cai, Y., Carrasco, J.A., Chang, M.W., Curach, N.C., Bell, D.J., Le Feuvre, R., Friedman, D.C., Fu, X., Gold, N.D., Herrgård, M.J., Holowko, M.B., Johnson, J.R., Johnson, R.A., Keasling, J.D., Kitney, R.I., Kondo, A., Liu, C., Martin, V.J.J., Freemont, P.S., 2019. Building a global alliance of biofoundries. Nat. Commun. 10, 2040. https://doi.org/10.1038/s41467-019-10079-2.
Hodgman, C.E., Jewett, M.C., 2012. Cell-free synthetic biology: thinking outside the cell. Metab. Eng. 14, 261–269. https://doi.org/10.1016/j.ymben.2011.09.002.
Ho, T.K., 1995. Random decision forests. Proceedings of 3rd international conference on document analysis and recognition 1, 278.
Hsu, P.D., Scott, D.A., Weinstein, J.A., Ran, F.A., Konermann, S., Agarwala, V., Li, Y., Fine, E.J., Wu, X., Shalem, O., Cradick, T.J., Marraffini, L.A., Bao, G., Zhang, F., 2013. DNA targeting specificity of RNA-guided Cas9 nucleases. Nat. Biotechnol. 31, 827–832. https://doi.org/10.1038/nbt.2647.
Humphrey, A., 1998. Shake flask to fermentor: what have we learned? Biotechnol. Prog. 14, 3–7. https://doi.org/10.1021/bp970130k.
Islam, M.R., Tudryn, G., Bucinell, R., Schadler, L., Picu, R.C., 2017. Morphology and mechanics of fungal mycelium. Sci. Rep. 7, 13070. https://doi.org/10.1038/s41598-017-13295-2.
Jensen, K.T., Fløe, L., Petersen, T.S., Huang, J., Xu, F., Bolund, L., Luo, Y., Lin, L., 2017. Chromatin accessibility and guide sequence secondary structure affect CRISPR-Cas9 gene editing efficiency. FEBS Lett. 591, 1892–1901. https://doi.org/10.1002/1873-3468.12707.
Jervis, A.J., Carbonell, P., Taylor, S., Sung, R., Dunstan, M.S., Robinson, C.J., Breitling, R., Takano, E., Scrutton, N.S., 2019a. SelProm: a queryable and predictive expression vector selection tool for Escherichia coli. ACS Synth. Biol. 8, 1478–1483. https://doi.org/10.1021/acssynbio.8b00399.
Jervis, A.J., Carbonell, P., Vinaixa, M., Dunstan, M.S., Hollywood, K.A., Robinson, C.J., Rattray, N.J.W., Yan, C., Swainston, N., Currin, A., Sung, R., Toogood, H., Taylor, S., Faulon, J.-L., Breitling, R., Takano, E., Scrutton, N.S., 2019b. Machine learning of designed translational control allows predictive pathway optimization in Escherichia coli. ACS Synth. Biol. 8, 127–136. https://doi.org/10.1021/acssynbio.8b00398.
Jessop-Fabre, M.M., Sonnenschein, N., 2019. Improving reproducibility in synthetic biology. Front. Bioeng. Biotechnol. 7, 18. https://doi.org/10.3389/fbioe.2019.00018.
Jin, H., Song, Q., Hu, X., 2019. Auto-keras: an efficient neural architecture search system. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. Presented at the KDD '19: the 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. ACM, New York, NY, USA, pp. 1946–1956. https://doi.org/10.1145/3292500.3330648.
Johansson, J., Hedhammar, M., Rising, A., Nordling, K., 2014. Method of producing polymers of spider silk proteins 2010. US Patent 8642734B2, filed 2010. and issued. https://patents.google.com/patent/US20120041177.
Kang, A., Mendez-Perez, D., Goh, E.-B., Baidoo, E.E.K., Benites, V.T., Beller, H.R., Keasling, J.D., Adams, P.D., Mukhopadhyay, A., Lee, T.S., 2019. Optimization of the IPP-bypass mevalonate pathway and fed-batch fermentation for the production of isoprenol in Escherichia coli. Metab. Eng. 56, 85–96. https://doi.org/10.1016/j.ymben.2019.09.003.
Karim, A.S., Dudley, Q.M., Juminaga, A., Yuan, Y., Crowe, S.A., Heggestad, J.T., Garg, S., Abdalla, T., Grubbe, W.S., Rasor, B.J., Coar, D.N., Torculas, M., Krein, M., Liew, F.E., Quattlebaum, A., Jensen, R.O., Stuart, J.A., Simpson, S.D., Köpke, M., Jewett, M.C., 2020. In vitro prototyping and rapid optimization of biosynthetic enzymes for cell design. Nat. Chem. Biol. 16, 912–919. https://doi.org/10.1038/s41589-020-0559-0.
Karp, P.D., Billington, R., Caspi, R., Fulcher, C.A., Latendresse, M., Kothari, A., Keseler, I. M., Krummenacker, M., Midford, P.E., Ong, Q., Ong, W.K., Paley, S.M., Subhraveti, P., 2019. The BioCyc collection of microbial genomes and metabolic pathways. Briefings Bioinf. 20, 1085–1093. https://doi.org/10.1093/bib/bbx085.
Karr, J.R., Sanghvi, J.C., Macklin, D.N., Gutschow, M.V., Jacobs, J.M., Bolival, B., Assad-Garcia, N., Glass, J.I., Covert, M.W., 2012. A whole-cell computational model predicts phenotype from genotype. Cell 150, 389–401. https://doi.org/10.1016/j.cell.2012.05.044.
Kehe, J., Kulesa, A., Ortiz, A., Ackerman, C.M., Thakku, S.G., Sellers, D., Kuehn, S., Gore, J., Friedman, J., Blainey, P.C., 2019. Massively parallel screening of synthetic microbial communities. Proc. Natl. Acad. Sci. U.S.A. 116, 12804–12809. https://doi.org/10.1073/pnas.1900102116.
Kelley, D.R., Liu, B., Delcher, A.L., Pop, M., Salzberg, S.L., 2012. Gene prediction with Glimmer for metagenomic sequences augmented by classification and clustering. Nucleic Acids Res. 40, e9. https://doi.org/10.1093/nar/gkr1067.
Keogh, E., Ratanamahatana, C.A., 2005. Exact indexing of dynamic time warping. Knowl. Inf. Syst. 7, 358–386. https://doi.org/10.1007/s10115-004-0154-9.
Kiedaisch, Jill, 2019. "You Can Now Smell a Flower That Went Extinct a Century Ago". Popular Mechanics. https://www.popularmechanics.com/science/environment/a27155735/smell-flower-extinct/.
Kim, G.B., Kim, W.J., Kim, H.U., Lee, S.Y., 2019. Machine learning applications in systems metabolic engineering. Curr. Opin. Biotechnol. 64, 1–9. https://doi.org/10.1016/j.copbio.2019.08.010.
Kim, H.K., Min, S., Song, M., Jung, S., Choi, J.W., Kim, Y., Lee, S., Yoon, S., Kim, H.H., 2018. Deep learning improves prediction of CRISPR-Cpf1 guide RNA activity. Nat. Biotechnol. 36, 239–241. https://doi.org/10.1038/nbt.4061.
Kim, O.D., Rocha, M., 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.
Kim, S., Thiessen, P.A., Bolton, E.E., Chen, J., Fu, G., Gindulyte, A., Han, L., He, J., He, S., Shoemaker, B.A., Wang, J., Yu, B., Zhang, J., Bryant, S.H., 2016. PubChem substance and compound databases. Nucleic Acids Res. 44, D1202–D1213. https://doi.org/10.1093/nar/gkv951.
King, Z.A., Lu, J., Dräger, A., Miller, P., Federowicz, S., Lerman, J.A., Ebrahim, A., Palsson, B.O., Lewis, N.E., 2016. 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.
Kluyver, Thomas, 2016. Jupyter Notebooks—a publishing format for reproducible computational workflows. Positioning and Power in Academic Publishing: Players, Agents and Agendas. IOS Press Ebooks. http://ebooks.iospress.nl/publication/42900.
Knott, G.J., Doudna, J.A., 2018. CRISPR-Cas guides the future of genetic engineering. Science 361, 866–869. https://doi.org/10.1126/science.aat5011.
Koch, M., Duigou, T., Faulon, J.-L., 2020. Reinforcement learning for bioretrosynthesis. ACS Synth. Biol. 9, 157–168. https://doi.org/10.1021/acssynbio.9b00447.
Kothamachu, V.B., Zaini, S., Muffatto, F., 2020. Role of digital microfluidics in enabling access to laboratory automation and making biology programmable. SLAS Technol, 2472630320931794. https://doi.org/10.1177/2472630320931794.
Kotopka, B.J., Smolke, C.D., 2020. Model-driven generation of artificial yeast promoters. Nature Communications 11 (1), 1–13.
Koutinas, M., Kiparissides, A., Pistikopoulos, E.N., Mantalaris, A., 2012. Bioprocess systems engineering: transferring traditional process engineering principles to industrial biotechnology. Comput. Struct. Biotechnol. J. 3, e201210022 https://doi.org/10.5936/csbj.201210022.

Kovárová-Kovar, K., Egli, T., 1998. Growth kinetics of suspended microbial cells: from single-substrate-controlled growth to mixed-substrate kinetics. Microbiol. Mol. Biol. Rev. 62, 646–666.
Kreimeyer, K., Foster, M., Pandey, A., Arya, N., Halford, G., Jones, S.F., Forshee, R., Walderhaug, M., Botsis, T., 2017. Natural language processing systems for capturing and standardizing unstructured clinical information: a systematic review. J. Biomed. Inf. 73, 14–29. https://doi.org/10.1016/j.jbi.2017.07.012.
Kumar, A., Wang, L., Ng, C.Y., Maranas, C.D., 2018. Pathway design using de novo steps through uncharted biochemical spaces. Nat. Commun. 9, 184. https://doi.org/10.1038/s41467-017-02362-x.
Kumar, N., Skolnick, J., 2012. EFICAz2.5: application of a high-precision enzyme function predictor to 396 proteomes. Bioinformatics 28, 2687–2688. https://doi.org/10.1093/bioinformatics/bts510.
Lawson, C.E., Harcombe, W.R., Hatzenpichler, R., Lindemann, S.R., Löffler, F.E., O'Malley, M.A., García Martín, H., Pfleger, B.F., Raskin, L., Venturelli, O.S., Weissbrodt, D.G., Noguera, D.R., McMahon, K.D., 2019. Common principles and best practices for engineering microbiomes. Nat. Rev. Microbiol. 17, 725–741. https://doi.org/10.1038/s41579-019-0255-9.
LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444. https://doi.org/10.1038/nature14539.
Lee, S.Y., Kim, H.U., Chae, T.U., Cho, J.S., Kim, J.W., Shin, J.H., Kim, D.I., Ko, Y.-S., Jang, W.D., Jang, Y.-S., 2019. A comprehensive metabolic map for production of bio-based chemicals. Nat. Catal. 2, 18–33. https://doi.org/10.1038/s41929-018-0212-4.
Lerman, J.A., Hyduke, D.R., Latif, H., Portnoy, V.A., Lewis, N.E., Orth, J.D., Schrimpe-Rutledge, A.C., Smith, R.D., Adkins, J.N., Zengler, K., Palsson, B.O., 2012. In silico method for modelling metabolism and gene product expression at genome scale. Nat. Commun. 3, 929. https://doi.org/10.1038/ncomms1928.
Leveau, J.H., Lindow, S.E., 2001. Predictive and interpretive simulation of green fluorescent protein expression in reporter bacteria. J. Bacteriol. 183, 6752–6762. https://doi.org/10.1128/JB.183.23.6752-6762.2001.
Le, K., Tan, C., Gupta, S., Guhan, T., Barkhordarian, H., Lull, J., Stevens, J., Munro, T., 2018. A novel mammalian cell line development platform utilizing nanofluidics and optoelectro positioning technology. Biotechnol. Prog. 34, 1438–1446. https://doi.org/10.1002/btpr.2690.
Lin, G.-M., Warden-Rothman, R., Voigt, C.A., 2019. Retrosynthetic design of metabolic pathways to chemicals not found in nature. Curr. Opin. Struct. Biol. https://doi.org/10.1016/j.coisb.2019.04.004.
Lin, J., Wong, K.-C., 2018. Off-target predictions in CRISPR-Cas9 gene editing using deep learning. Bioinformatics 34, i656–i663. https://doi.org/10.1093/bioinformatics/bty554.
Listgarten, J., Weinstein, M., Kleinstiver, B.P., Sousa, A.A., Joung, J.K., Crawford, J., Gao, K., Hoang, L., Elibol, M., Doench, J.G., Fusi, N., 2018. Prediction of off-target activities for the end-to-end design of CRISPR guide RNAs. Nat. Biomed. Eng. 2, 38–47. https://doi.org/10.1038/s41551-017-0178-6.
Liu, R., Bassalo, M.C., Zeitoun, R.I., Gill, R.T., 2015. Genome scale engineering techniques for metabolic engineering. Metab. Eng. 32, 143–154. https://doi.org/10.1016/j.ymben.2015.09.013.
Lohr, Steve, 2014. The New York Times. https://www.nytimes.com/2014/08/18/technology/for-big-data-scientists-hurdle-to-insights-is-janitor-work.html.
Lopatkin, A.J., Collins, J.J., 2020. Predictive biology: modelling, understanding and harnessing microbial complexity. Nat. Rev. Microbiol. https://doi.org/10.1038/s41579-020-0372-5.
Luo, X., Reiter, M.A., d'Espaux, L., Wong, J., Denby, C.M., Lechner, A., Zhang, Y., Grzybowski, A.T., Harth, S., Lin, W., Lee, H., Yu, C., Shin, J., Deng, K., Benites, V.T., Wang, G., Baidoo, E.E.K., Chen, Y., Dev, I., Petzold, C.J., Keasling, J.D., 2019. Complete biosynthesis of cannabinoids and their unnatural analogues in yeast. Nature 567, 123–126. https://doi.org/10.1038/s41586-019-0978-9.
Luque de Castro, M.D., Priego-Capote, F., 2018. The analytical process to search for metabolomics biomarkers. J. Pharmaceut. Biomed. Anal. 147, 341–349. https://doi.org/10.1016/j.jpba.2017.06.073.
Ma, J., Yu, M.K., Fong, S., Ono, K., Sage, E., Demchak, B., Sharan, R., Ideker, T., 2018. Using deep learning to model the hierarchical structure and function of a cell. Nat. Methods 15, 290–298. https://doi.org/10.1038/nmeth.4627.
Machado, D., Herrgård, M., 2014. Systematic evaluation of methods for integration of transcriptomic data into constraint-based models of metabolism. PLoS Comput. Biol. 10, e1003580 https://doi.org/10.1371/journal.pcbi.1003580.
Macklin, D.N., Ahn-Horst, T.A., Choi, H., Ruggero, N.A., Carrera, J., Mason, J.C., Sun, G., Agmon, E., DeFelice, M.M., Maayan, I., Lane, K., Spangler, R.K., Gillies, T.E., Paull, M.L., Akhter, S., Bray, S.R., Weaver, D.S., Keseler, I.M., Karp, P.D., Morrison, J.H., Covert, M.W., 2020. Simultaneous cross-evaluation of heterogeneous E. coli datasets via mechanistic simulation. Science 369. https://doi.org/10.1126/science.aav3751.
MacLeod, B.P., Parlane, F.G.L., Morrissey, T.D., Häse, F., Roch, L.M., Dettelbach, K.E., Moreira, R., Yunker, L.P.E., Rooney, M.B., Deeth, J.R., Lai, V., Ng, G.J., Situ, H., Zhang, R.H., Elliott, M.S., Haley, T.H., Dvorak, D.J., Aspuru-Guzik, A., Hein, J.E., Berlinguette, C.P., 2020. Self-driving laboratory for accelerated discovery of thin-film materials. Sci. Adv. 6, eaaz8867 https://doi.org/10.1126/sciadv.aaz8867.
Maia Chagas, A., 2018. Haves and have nots must find a better way: the case for open scientific hardware. PLoS Biol. 16, e3000014 https://doi.org/10.1371/journal.pbio.3000014.
Maia Chagas, A., Molloy, J.C., Prieto-Godino, L.L., Baden, T., 2020. Leveraging open hardware to alleviate the burden of COVID-19 on global health systems. PLoS Biol. 18, e3000730 https://doi.org/10.1371/journal.pbio.3000730.
Mamas, M., Dunn, W.B., Neyses, L., Goodacre, R., 2011. The role of metabolites and metabolomics in clinically applicable biomarkers of disease. Arch. Toxicol. 85, 5–17. https://doi.org/10.1007/s00204-010-0609-6.
Manning, C.D., 1999. Foundations of Statistical Natural Language Processing, first ed. Mit Press, Cambridge, Mass.
Manning, C.D., Raghavan, P., Schutze, H., 2008. Introduction to Information Retrieval. Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511809071.
May, M., 2019. A DIY approach to automating your lab. Nature 569, 587–588. https://doi.org/10.1038/d41586-019-01590-z.
McKay, M.D., Beckman, R.J., Conover, W.J., 1979. Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 21, 239–245. https://doi.org/10.1080/00401706.1979.10489755.
McLaughlin, J.A., Myers, C.J., Zundel, Z., Mısırlı, G., Zhang, M., Ofiteru, I.D., Goñi-Moreno, A., Wipat, A., 2018. SynBioHub: a standards-enabled design repository for synthetic biology. ACS Synth. Biol. 7, 682–688. https://doi.org/10.1021/acssynbio.7b00403.
Meat-free outsells beef. Nat. Biotechnol. 37, 2019. https://doi.org/10.1038/s41587-019-0313-x, 1250–1250.
Melcher, M., Scharl, T., Spangl, B., Luchner, M., Cserjan, M., Bayer, K., Leisch, F., Striedner, G., 2015. The potential of random forest and neural networks for biomass and recombinant protein modeling in Escherichia coli fed-batch fermentations. Biotechnol. J. 10, 1770–1782. https://doi.org/10.1002/biot.201400790.
Mellor, J., Grigoras, I., Carbonell, P., Faulon, J.-L., 2016. Semisupervised Gaussian process for automated enzyme search. ACS Synth. Biol. 5, 518–528. https://doi.org/10.1021/acssynbio.5b00294.
Mell, P., Grance, T., 2011. The NIST Definition of Cloud Computing. National Institute of Standards and Technology, Gaithersburg, MD. https://doi.org/10.6028/NIST.SP.800-145.
Melnikov, A.D., Tsentalovich, Y.P., Yanshole, V.V., 2020. Deep learning for the precise peak detection in high-resolution LC-MS data. Anal. Chem. 92, 588–592. https://doi.org/10.1021/acs.analchem.9b04811.
Melnyk, A., 1996. Searle's abstract argument against strong AI. Synthese 108, 391–419. https://doi.org/10.1007/BF00413696.
Meng, H., Wang, J., Xiong, Z., Xu, F., Zhao, G., Wang, Y., 2013. Quantitative design of regulatory elements based on high-precision strength prediction using artificial neural network. PloS One 8, e60288. https://doi.org/10.1371/journal.pone.0060288.
Merino, M., Gómez, I.M., Molina, A.J., 2015. Envelope filter sequence to delete blinks and overshoots. Biomed. Eng. Online 14, 48. https://doi.org/10.1186/s12938-015-0046-0.
Metz, Cade, 2018. A.I. Researchers Are Making More Than $1 Million, Even at a Nonprofit. The New York Times. https://www.nytimes.com/2018/04/19/technology/artificial-intelligence-salaries-openai.html.
Miller, E.K., 1989. The computer revolution. IEEE Potentials 8, 27–31. https://doi.org/10.1109/45.31594.
Minnich, A.J., McLoughlin, K., Tse, M., Deng, J., Weber, A., Murad, N., Madej, B.D., Ramsundar, B., Rush, T., Calad-Thomson, S., Brase, J., Allen, J.E., 2020. AMPL: a data-driven modeling pipeline for drug discovery. J. Chem. Inf. Model. 60, 1955–1968. https://doi.org/10.1021/acs.jcim.9b01053.
Moretti, S., Martin, O., Van Du Tran, T., Bridge, A., Morgat, A., Pagni, M., 2016. MetaNetX/MNXref–reconciliation of metabolites and biochemical reactions to bring together genome-scale metabolic networks. Nucleic Acids Res. 44, D523–D526. https://doi.org/10.1093/nar/gkv1117.
Morrell, W.C., Birkel, G.W., Forrer, M., Lopez, T., Backman, T.W.H., Dussault, M., Petzold, C.J., Baidoo, E.E.K., Costello, Z., Ando, D., Alonso-Gutierrez, J., George, K.W., Mukhopadhyay, A., Vaino, I., Keasling, J.D., Adams, P.D., Hillson, N.J., Garcia Martin, H., 2017. The experiment data Depot: a web-based software tool for biological experimental data storage, sharing, and visualization. ACS Synth. Biol. 6, 2248–2259. https://doi.org/10.1021/acssynbio.7b00204.
Murkin, A.S., Manning, K.A., Kholodar, S.A., 2014. Mechanism and inhibition of 1-deoxy-D-xylulose-5-phosphate reductoisomerase. Bioorg. Chem. 57, 171–185. https://doi.org/10.1016/j.bioorg.2014.06.001.
Ndah, E., Jonckheere, V., Giess, A., Valen, E., Menschaert, G., Van Damme, P., 2017. REPARATION: ribosome profiling assisted (re-)annotation of bacterial genomes. Nucleic Acids Res. 45, e168. https://doi.org/10.1093/nar/gkx758.
Nielsen, J., Keasling, J.D., 2016. Engineering cellular metabolism. Cell 164, 1185–1197. https://doi.org/10.1016/j.cell.2016.02.004.
Nursimulu, N., Xu, L.L., Wasmuth, J.D., Krukov, I., Parkinson, J., 2018. Improved enzyme annotation with EC-specific cutoffs using DETECT v2. Bioinformatics 34, 3393–3395. https://doi.org/10.1093/bioinformatics/bty368.
O'Brien, E.J., Monk, J.M., Palsson, B.O., 2015. Using genome-scale models to predict biological capabilities. Cell 161, 971–987. https://doi.org/10.1016/j.cell.2015.05.019.
Others O'Malley, T., Bursztein, E., Long, J., Chollet, F., Jin, H., Invernizzi, L., 2019. Keras Tuner.
O'Regan, G., 2012. Revolutions in the 1980s and 1990s. In: A Brief History of Computing. Springer London, London, pp. 63–69. https://doi.org/10.1007/978-1-4471-2359-0_5.
Olson, R.S., Moore, J.H., 2018. Identifying and harnessing the building blocks of machine learning pipelines for sensible initialization of a data science automation tool. In: Riolo, R., Worzel, B., Goldman, B., Tozier, B. (Eds.), Genetic Programming Theory and Practice XIV, Genetic and Evolutionary Computation. Springer International Publishing, Cham, pp. 211–223. https://doi.org/10.1007/978-3-319-97088-2_14.
Olson, R.S., Moore, J.H., 2019. TPOT: a tree-based pipeline optimization tool for automating machine learning. In: Hutter, F., Kotthoff, L., Vanschoren, J. (Eds.), Automated Machine Learning: Methods, Systems, Challenges, the Springer Series on

Challenges in Machine Learning. Springer International Publishing, Cham, pp. 151–160. https://doi.org/10.1007/978-3-030-05318-5_8.
Olson, R.S., Urbanowicz, R.J., Andrews, P.C., Lavender, N.A., Kidd, L.C., Moore, J.H., 2016. Automating biomedical data science through tree-based pipeline optimization. In: Squillero, G., Burelli, P. (Eds.), Applications of Evolutionary Computation, Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 123–137. https://doi.org/10.1007/978-3-319-31204-0_9.
Open for business, 2017. Sci. Data 4, 170058. https://doi.org/10.1038/sdata.2017.58.
Opgenorth, P., Costello, Z., Okada, T., Goyal, G., Chen, Y., Gin, J., Benites, V., de Raad, M., Northen, T.R., Deng, K., Deutsch, S., Baidoo, E.E.K., Petzold, C.J., Hillson, N.J., Garcia Martin, H., Beller, H.R., 2019. Lessons from two design-build-test-learn cycles of dodecanol production in Escherichia coli aided by machine learning. ACS Synth. Biol. 8, 1337–1351. https://doi.org/10.1021/acssynbio.9b00020.
Paddon, C.J., Keasling, J.D., 2014. Semi-synthetic artemisinin: a model for the use of synthetic biology in pharmaceutical development. Nat. Rev. Microbiol. 12, 355–367. https://doi.org/10.1038/nrmicro3240.
Paeng, K., Hwang, S., Park, S., Kim, M., Kim, S., 2016. A Unified Framework for Tumor Proliferation Score Prediction in Breast Histopathology.
Pan, S.J., Yang, Q., 2010. A survey on transfer learning. IEEE Trans. Knowl. Data Eng. 22, 1345–1359. https://doi.org/10.1109/TKDE.2009.191.
Pappu, S.M.J., Gummadi, S.N., 2017. Artificial neural network and regression coupled genetic algorithm to optimize parameters for enhanced xylitol production by Debaryomyces nepalensis in bioreactor. Biochem. Eng. J. 120, 136–145. https://doi.org/10.1016/j.bej.2017.01.010.
Paschon, D.E., Lussier, S., Wangzor, T., Xia, D.F., Li, P.W., Hinkley, S.J., Scarlott, N.A., Lam, S.C., Waite, A.J., Truong, L.N., Gandhi, N., Kadam, B.N., Patil, D.P., Shivak, D. A., Lee, G.K., Holmes, M.C., Zhang, L., Miller, J.C., Rebar, E.J., 2019. Diversifying the structure of zinc finger nucleases for high-precision genome editing. Nat. Commun. 10, 1133. https://doi.org/10.1038/s41467-019-08867-x.
Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830.
Pei, J., Deng, L., Song, S., Zhao, M., Zhang, Y., Wu, Shuang, Wang, G., Zou, Z., Wu, Z., He, W., Chen, F., Deng, N., Wu, Si, Wang, Y., Wu, Y., Yang, Z., Ma, C., Li, G., Han, W., Li, H., Shi, L., 2019. Towards artificial general intelligence with hybrid Tianjic chip architecture. Nature 572, 106–111. https://doi.org/10.1038/s41586-019-1424-8.
Peralta-Yahya, P.P., Zhang, F., del Cardayre, S.B., Keasling, J.D., 2012. Microbial engineering for the production of advanced biofuels. Nature 488, 320–328. https://doi.org/10.1038/nature11478.
Petegrosso, R., Park, S., Hwang, T.H., Kuang, R., 2017. Transfer learning across ontologies for phenome-genome association prediction. Bioinformatics 33, 529–536. https://doi.org/10.1093/bioinformatics/btw649.
Petzold, C.J., Chan, L.J.G., Nhan, M., Adams, P.D., 2015. Analytics for metabolic engineering. Front. Bioeng. Biotechnol. 3, 135. https://doi.org/10.3389/fbioe.2015.00135.
Popova, M., Isayev, O., Tropsha, A., 2018. Deep reinforcement learning for de novo drug design. Sci. Adv. 4, eaap7885 https://doi.org/10.1126/sciadv.aap7885.
Presnell, K.V., Alper, H.S., 2019. Systems metabolic engineering meets machine learning: a new era for data-driven metabolic engineering. Biotechnol. J. 14, e1800416 https://doi.org/10.1002/biot.201800416.
Radivojević, T., Costello, Z., Workman, K., Garcia Martin, H., 2020. A machine learning Automated Recommendation Tool for synthetic biology. Nat. Commun. 11, 4879. https://doi.org/10.1038/s41467-020-18008-4.
Raissi, M., Perdikaris, P., Karniadakis, G.E., 2019. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. 378, 686–707. https://doi.org/10.1016/j.jcp.2018.10.045.
Reyon, D., Tsai, S.Q., Khayter, C., Foden, J.A., Sander, J.D., Joung, J.K., 2012. FLASH assembly of TALENs for high-throughput genome editing. Nat. Biotechnol. 30, 460–465. https://doi.org/10.1038/nbt.2170.
Rhodius, V.A., Mutalik, V.K., 2010. Predicting strength and function for promoters of the Escherichia coli alternative sigma factor, sigmaE. Proc. Natl. Acad. Sci. U.S.A. 107, 2854–2859. https://doi.org/10.1073/pnas.0915066107.
Riley, P., 2019. Three pitfalls to avoid in machine learning. Nature 572, 27–29. https://doi.org/10.1038/d41586-019-02307-y.
Rocklin, M., 2015. Dask: parallel computation with blocked algorithms and task scheduling. In: Huff, K., Bergstra, J. (Eds.), Proceedings of the 14th Python in Science Conference. Presented at the SciPy 2015, pp. 130–136.
Rodrigues, T., 2020. The good, the bad, and the ugly in chemical and biological data for machine learning. Drug Discov. Today Technol. https://doi.org/10.1016/j.ddtec.2020.07.001.
Rogati, Monica, 2017. The AI Hierarchy of Needs. Hackernoon. https://hackernoon.com/the-ai-hierarchy-of-needs-18f111fcc007.
Rohe, P., Venkanna, D., Kleine, B., Freudl, R., Oldiges, M., 2012. An automated workflow for enhancing microbial bioprocess optimization on a novel microbioreactor platform. Microb. Cell Factories 11, 144. https://doi.org/10.1186/1475-2859-11-144.
Romero, P.A., Krause, A., Arnold, F.H., 2013. Navigating the protein fitness landscape with Gaussian processes. Proc. Natl. Acad. Sci. U.S.A. 110, E193–E201. https://doi.org/10.1073/pnas.1215251110.
Ryu, J.Y., Kim, H.U., Lee, S.Y., 2019. Deep learning enables high-quality and high-throughput prediction of enzyme commission numbers. Proc. Natl. Acad. Sci. U.S.A. 116, 13996–14001. https://doi.org/10.1073/pnas.1821905116.
Sainz de Murieta, I., Bultelle, M., Kitney, R.I., 2016. Toward the first data acquisition standard in synthetic biology. ACS Synth. Biol. 5, 817–826. https://doi.org/10.1021/acssynbio.5b00222.
Saito, Y., Oikawa, M., Nakazawa, H., Niide, T., Kameda, T., Tsuda, K., Umetsu, M., 2018. Machine-learning-guided mutagenesis for directed evolution of fluorescent proteins. ACS Synth. Biol. 7, 2014–2022. https://doi.org/10.1021/acssynbio.8b00155.
Sajda, P., 2006. Machine learning for detection and diagnosis of disease. Annu. Rev. Biomed. Eng. 8, 537–565. https://doi.org/10.1146/annurev.bioeng.8.061505.095802.
Salis, H.M., Mirsky, E.A., Voigt, C.A., 2009. Automated design of synthetic ribosome binding sites to control protein expression. Nat. Biotechnol. 27, 946–950. https://doi.org/10.1038/nbt.1568.
Sculley, D., 2010. Web-scale k-means clustering. In: Proceedings of the 19th International Conference on World Wide Web - WWW '10. Presented at the the 19th International Conference. ACM Press, New York, New York, USA, p. 1177. https://doi.org/10.1145/1772690.1772862.
Segler, M.H.S., Preuss, M., Waller, M.P., 2018. Planning chemical syntheses with deep neural networks and symbolic AI. Nature 555, 604–610. https://doi.org/10.1038/nature25978.
Sheridan, R.P., 2013. Time-split cross-validation as a method for estimating the goodness of prospective prediction. J. Chem. Inf. Model. 53, 783–790. https://doi.org/10.1021/ci400084k.
Silver, A., 2019. Five innovative ways to use 3D printing in the laboratory. Nature 565, 123–124. https://doi.org/10.1038/d41586-018-07853-5.
Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., Hassabis, D., 2016. Mastering the game of Go with deep neural networks and tree search. Nature 529, 484–489. https://doi.org/10.1038/nature16961.
Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K., Hassabis, D., 2018. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. Science 362, 1140–1144. https://doi.org/10.1126/science.aar6404.
Si, T., Chao, R., Min, Y., Wu, Y., Ren, W., Zhao, H., 2017. Automated multiplex genome-scale engineering in yeast. Nat. Commun. 8, 15187. https://doi.org/10.1038/ncomms15187.
Snyder, L.R., Peters, J.E., Henkin, T.M., Champness, W., 2014. Molecular Genetics of Bacteria, fourth ed.fourth ed. ASM Press.
Stephens, Z.D., Lee, S.Y., Faghri, F., Campbell, R.H., Zhai, C., Efron, M.J., Iyer, R., Schatz, M.C., Sinha, S., Robinson, G.E., 2015. Big data: astronomical or genomical? PLoS Biol. 13, e1002195 https://doi.org/10.1371/journal.pbio.1002195.
Storch, M., Haines, M.C., Baldwin, G.S., 2020. DNA-BOT: a low-cost, automated DNA assembly platform for synthetic biology. Synth. Biol. https://doi.org/10.1093/synbio/ysaa010.
Streich, J., Romero, J., Gazolla, J.G.F.M., Kainer, D., Cliff, A., Prates, E.T., Brown, J.B., Khoury, S., Tuskan, G.A., Garvin, M., Jacobson, D., Harfouche, A.L., 2020. Can exascale computing and explainable artificial intelligence applied to plant biology deliver on the United Nations sustainable development goals? Curr. Opin. Biotechnol. 61, 217–225. https://doi.org/10.1016/j.copbio.2020.01.010.
Suzek, B.E., Wang, Y., Huang, H., McGarvey, P.B., Wu, C.H., Consortium, UniProt, 2015. UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. Bioinformatics 31, 926–932. https://doi.org/10.1093/bioinformatics/btu739.
Syed, R., Suriadi, S., Adams, M., Bandara, W., Leemans, S.J.J., Ouyang, C., ter Hofstede, A.H.M., van de Weerd, I., Wynn, M.T., Reijers, H.A., 2020. Robotic process automation: contemporary themes and challenges. Comput. Ind. 115, 103162. https://doi.org/10.1016/j.compind.2019.103162.
Szappanos, B., Kovács, K., Szamecz, B., Honti, F., Costanzo, M., Baryshnikova, A., Gelius-Dietrich, G., Lercher, M.J., Jelasity, M., Myers, C.L., Andrews, B.J., Boone, C., Oliver, S.G., Pál, C., Papp, B., 2011. An integrated approach to characterize genetic interaction networks in yeast metabolism. Nat. Genet. 43, 656–662. https://doi.org/10.1038/ng.846.
The UniProt Consortium, 2017. UniProt: the universal protein knowledgebase. Nucleic Acids Res. 45, D158–D169. https://doi.org/10.1093/nar/gkw1099.
Thiele, I., 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.
Tian, T., Kang, J.W., Kang, A., Lee, T.S., 2019. Redirecting metabolic flux via combinatorial multiplex CRISPRi-mediated repression for isopentenol production in Escherichia coli. ACS Synth. Biol. 8, 391–402. https://doi.org/10.1021/acssynbio.8b00429.
Treloar, N.J., Fedorec, A.J.H., Ingalls, B., Barnes, C.P., 2020. Deep reinforcement learning for the control of microbial co-cultures in bioreactors. PLoS Comput. Biol. 16, e1007783 https://doi.org/10.1371/journal.pcbi.1007783.
UniProt Consortium, 2015. UniProt: a hub for protein information. Nucleic Acids Res. 43, D204–D212. https://doi.org/10.1093/nar/gku989.
van der Aalst, W.M.P., Bichler, M., Heinzl, A., 2018. Robotic process automation. Bus. Inf. Syst. Eng. 60, 269–272. https://doi.org/10.1007/s12599-018-0542-4.
van der Laan, M.J., Polley, E.C., Hubbard, A.E., 2007. Super learner. Stat. Appl. Genet. Mol. Biol. 6, Article25 https://doi.org/10.2202/1544-6115.1309.
Vizcaíno, J.A., Deutsch, E.W., Wang, R., Csordas, A., Reisinger, F., Ríos, D., Dianes, J.A., Sun, Z., Farrah, T., Bandeira, N., Binz, P.-A., Xenarios, I., Eisenacher, M., Mayer, G., Gatto, L., Campos, A., Chalkley, R.J., Kraus, H.-J., Albar, J.P., Martinez-Bartolomé, S., Hermjakob, H., 2014. ProteomeXchange provides globally

coordinated proteomics data submission and dissemination. Nat. Biotechnol. 32, 223–226. https://doi.org/10.1038/nbt.2839.
Volk, M.J., Lourentzou, I., Mishra, S., Vo, L.T., Zhai, C., Zhao, H., 2020. Biosystems design by machine learning. ACS Synth. Biol. 9, 1514–1533. https://doi.org/10.1021/acssynbio.0c00129.
Voulodimos, A., Doulamis, N., Doulamis, A., Protopapadakis, E., 2018. Deep learning for computer vision: a brief review. Comput. Intell. Neurosci., 7068349 https://doi.org/10.1155/2018/7068349.
Walch, Kathleen, 2019. Rethinking Weak Vs. Strong AI. Forbes. https://www.forbes.com/sites/cognitiveworld/2019/10/04/rethinking-weak-vs-strong-ai/?sh=193057d76da3.
Wang, G., Björk, S.M., Huang, M., Liu, Q., Campbell, K., Nielsen, J., Joensson, H.N., Petranovic, D., 2019. RNAi expression tuning, microfluidic screening, and genome recombineering for improved protein production in Saccharomyces cerevisiae. Proc. Natl. Acad. Sci. U.S.A. 116, 9324–9332. https://doi.org/10.1073/pnas.1820561116.
Giraud-Carrier, C., Provost, F., 2005. Toward a Justification of Meta-Learning: Is the No Free Lunch Theorem a Show-Stopper. Proceedings of the ICML-2005 Workshop on Meta-Learning 12.
Wehrs, M., Tanjore, D., Eng, T., Lievense, J., Pray, T.R., Mukhopadhyay, A., 2019. Engineering robust production microbes for large-scale cultivation. Trends in microbiology 27 (6), 524–537.
Wolpert, D.H., 1996. The lack of A priori distinctions between learning algorithms. Neural Comput. 8, 1341–1390. https://doi.org/10.1162/neco.1996.8.7.1341.
Wong, B.G., Mancuso, C.P., Kiriakov, S., Bashor, C.J., Khalil, A.S., 2018. Precise, automated control of conditions for high-throughput growth of yeast and bacteria with eVOLVER. Nat. Biotechnol. 36, 614–623. https://doi.org/10.1038/nbt.4151.
Woolston, B.M., Edgar, S., Stephanopoulos, G., 2013. Metabolic engineering: past and future. Annu. Rev. Chem. Biomol. Eng. 4, 259–288. https://doi.org/10.1146/annurev-chembioeng-061312-103312.
Wu, S.G., Wang, Y., Jiang, W., Oyetunde, T., Yao, R., Zhang, X., Shimizu, K., Tang, Y.J., Bao, F.S., 2016. Rapid prediction of bacterial heterotrophic fluxomics using machine learning and constraint programming. PLoS Comput. Biol. 12, e1004838 https://doi.org/10.1371/journal.pcbi.1004838.
Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, Ł., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Dean, J., 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation.
Wu, Z., Kan, S.B.J., Lewis, R.D., Wittmann, B.J., Arnold, F.H., 2019. Machine learning-assisted directed protein evolution with combinatorial libraries. Proc. Natl. Acad. Sci. U.S.A. 116, 8852–8858. https://doi.org/10.1073/pnas.1901979116.
Xu, P., Clark, C., Ryder, T., Sparks, C., Zhou, J., Wang, M., Russell, R., Scott, C., 2017. Characterization of TAP Ambr 250 disposable bioreactors, as a reliable scale-down model for biologics process development. Biotechnol. Prog. 33, 478–489. https://doi.org/10.1002/btpr.2417.
Xu, X., 2012. From cloud computing to cloud manufacturing. Robot. Comput. Integrated Manuf. 28, 75–86. https://doi.org/10.1016/j.rcim.2011.07.002.
Yang, K.K., Wu, Z., Arnold, F.H., 2019. Machine-learning-guided directed evolution for protein engineering. Nat. Methods 16, 687–694. https://doi.org/10.1038/s41592-019-0496-6.
Yang, L., Meng, X., Karniadakis, G.E., 2020. B-PINNs: Bayesian Physics-Informed Neural Networks for Forward and Inverse PDE Problems with Noisy Data. arXiv. https://arxiv.org/abs/2003.06097.
Yoon, B.-J., 2009. Hidden Markov models and their applications in biological sequence analysis. Curr. Genom. 10, 402–415. https://doi.org/10.2174/138920209789177575.
Yu, C., Zavaljevski, N., Desai, V., Reifman, J., 2009. Genome-wide enzyme annotation with precision control: catalytic families (CatFam) databases. Proteins 74, 449–460. https://doi.org/10.1002/prot.22167.
Zampieri, G., Vijayakumar, S., Yaneske, E., Angione, C., 2019. Machine and deep learning meet genome-scale metabolic modeling. PLoS Comput. Biol. 15, e1007084 https://doi.org/10.1371/journal.pcbi.1007084.
Zampieri, M., Sekar, K., Zamboni, N., Sauer, U., 2017. Frontiers of high-throughput metabolomics. Curr. Opin. Chem. Biol. 36, 15–23. https://doi.org/10.1016/j.cbpa.2016.12.006.
Zelezniak, A., Vowinckel, J., Capuano, F., Messner, C.B., Demichev, V., Polowsky, N., Mülleder, M., Kamrad, S., Klaus, B., Keller, M.A., Ralser, M., 2018. Machine learning predicts the yeast metabolome from the quantitative proteome of kinase knockouts. Cell Syst 7, 269–283.e6. https://doi.org/10.1016/j.cels.2018.08.001.
Zhang, J., Petersen, S.D., Radivojevic, T., Ramirez, A., Pérez-Manríquez, A., Abeliuk, E., Sánchez, B.J., Costello, Z., Chen, Y., Fero, M.J., Martin, H.G., Nielsen, J., Keasling, J.D., Jensen, M.K., 2020. Combining mechanistic and machine learning models for predictive engineering and optimization of tryptophan metabolism. Nat. Commun. 11, 4880. https://doi.org/10.1038/s41467-020-17910-1.
Zhou, Y., Li, G., Dong, J., Xing, X.-H., Dai, J., Zhang, C., 2018. MiYA, an efficient machine-learning workflow in conjunction with the YeastFab assembly strategy for combinatorial optimization of heterologous metabolic pathways in Saccharomyces cerevisiae. Metab. Eng. 47, 294–302. https://doi.org/10.1016/j.ymben.2018.03.020.
Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., He, Q., 2020. A comprehensive survey on transfer learning. Proc. IEEE.