In silico, in vitro, and in vivo machine learning in synthetic biology and metabolic engineering

Jean-Loup Faulon, Léon Faure

Title: *In silico*, *in vitro* and *in vivo* Machine Learning in Synthetic Biology and Metabolic Engineering

Jean-Loup Faulon and Léon Faure

MICALIS Institute, INRAE / University of Paris-Saclay, Jouy-en-Josas, France. Jean-Loup.Faulon@inrae.fr, www.jfaulon.com

Abstract

Among the main learning methods reviewed in this paper and used in synthetic biology and metabolic engineering are supervised learning, reinforcement and active learning, and *in vitro* or *in vivo* learning.

In the context of biosynthesis, supervised machine learning is being exploited to predict biological sequence activities, predict structures and engineer sequences, and optimize culture conditions.

Active and reinforcement learning methods use training sets acquired through an iterative process generally involving experimental measurements. They are applied to design, engineer and optimize metabolic pathways and bioprocesses.

The nascent but promising developments with *in vitro* and *in vivo* learning comprise molecular circuits performing simple tasks like pattern recognition and classification.

1. Introduction

We have seen the past few years a growing interest in using machine learning for chemistry and biology, synthetic biology and metabolic engineering making no exception to this trend [1]. This paper reviews three main techniques used when engineering biological systems. In section 2, we present an overview of supervised and semi-supervised machine learning techniques, providing examples on searching for promiscuous enzyme activities. In section 3, we discuss active and reinforcement learning methods, which are generally based on supervised learning, with training sets acquired on-the-fly in an iterative process. These methods are particularly amendable to the Design-Build-Test-Learn synthetic biology cycle. Examples are provided in the context of predicting enzymatic activities, optimizing metabolic pathways, and performing retro-biosynthesis. Engineering information processing devices in living systems is a long-standing venture of synthetic biology. Yet, the problem of engineering devices that perform basic operations found in machine learning remains largely unexplored. Section 4 presents attempts to construct *in vitro* and *in vivo* perceptron which are the basic units of all artificial neural networks.

2. Supervised and semi-supervised learning

Supervised learning is one of the main machine learning method that is being used in biology, and in particular in bioinformatics where it has been extensively developed [2]. Focusing on biosynthesis, and to name a few, supervised learning enables one to predict enzyme activities [3][4][5][6], to propose protein structures [7], to engineer sequences (DNA, RNA, protein) [8][9][10][11], to complete metabolome [12], to optimize culture conditions [13], and to perform more unexpected tasks like predicting the lab-of-origin of engineered DNA [14]. The supervised learning workflow starts by compiling a training set where each object being studied (promoter sequence, RBS sequence, protein sequence, pathways,..) has been labeled (with strength, activity, flux,…). In life sciences, labels generally correspond to experimental measurements. Disregarding the machine learning technique used, the workflow is composed of two main steps: (1) training and (2) validation (cf. Figure 1). Training is not performed on the entire dataset but a fraction of it, the rest being set aside for validation.

The core of supervised learning is of course the learning step, where a mapping between the objects and the labels is established. Several mapping techniques have been used in synthetic biology and metabolic engineering including support vector machine [3] (SVM for classification or SVR for regression), random forests [15] (RF) , Gaussian Processes [16] (GP) and artificial neural networks (ANNs) including deep learning networks [4]-[13]. While Figure 1 illustrates the process with ANNs, with all machine learning techniques one must first transform the objects into vectors of integers or reals. Features extraction is generally the method preferred to compute these vectors. For biological sequences, string spectrum [17] (count of kmers top left side of Figure 1), motifs counts [4] (like Pfam domains), and one-hot encoding and embeddings [5] are common features that are used. With feature vectors in hands, all machine learning techniques mentioned above search for a linear or non-linear function mapping features to labels. When using an ANN (right side of Figure 1), the function is a recursive weighted sum starting with the feature vector

(input layer), propagating to hidden layers to reach an output layer composed of only one node. As we wish a 0 or 1 answer the value of the last layer is generally calculated through a sigmoid function. Learning here consists in finding the weights (w_i) for each weighted sum minimizing the difference between the values (0, 1) calculated at the last layer and the values in the training set. During validation, values are predicted using the trained network, and true or false positives or negatives (TP, FP, TN, FN) are recorded. ROC curves (bottom left side of Figure 1) can also be calculated on the validation set changing the threshold between positives and negatives. The number provided in Figure 1 are areas under the ROC curves that have been reported in the past when using Support Vector Machines [3] and Gaussian Processes [16].

In some instances, it is necessary to merge different type of features together, for instance when building a classifier to determine if a given sequence will metabolize a given substrate. In order to merge biological and chemical information, one strategy is to compile feature vectors separately for each object and then merge these into a tensor product [3], the tensor representing the interactions between the objects (sequence-chemical). Such a strategy has shown to outperform other techniques for drug-target interactions [18] and enzyme-substrate interactions [3].

When dealing with classification problems, like for instance finding if a particular sequence is a promiscuous enzyme [19], we need positive and negative examples in the training set. Yet, very often we are faced with the issue that only positives can be found, as failures are hardly reported in the literature. Comes therefore the problem of generating negative examples. In the past, this issue has been tackled using ad-hoc methods to generate negatives arbitrarily, for instance, randomly drawing sequences that are not annotated as those in the training set, or sequences and chemicals that are distant (similarity wise) to those in the training set [19]. A more thorough method is using semi-supervised learning [20]. In semi-supervised learning, the data set is composed of two classes, a class of labeled examples (either positive or negative when performing classification) for which measurement have been carried out and a class of unlabeled examples. The learning process consists in finding the best partition between positives and negatives by shuffling unlabeled data points either in the positive class or in a negative class.

While numerous papers are making use of machine learning in the life sciences, in the context of biosynthesis and bioengineering only a few studies have triggered experimental validation. One can cite a work making use of a semi-supervised GP [16], which predicted three native *E. coli* BL21 capable of synthesizing L-acetyl-Leucine. These enzymes (ECBD0907, ECBD4067 and ECDB4269) are known to transform glutamate into acetyl-glutamate , ornithine into acetyl-ornithine and glutamate and methyl-oxovalerate into oxoglutarate and leucine. When overexpressed, two (ECBD4067 and ECDB4269) of the three enzymes increased the production of acetyl-leucine. Not only this study demonstrated that machine learning could be used to find promiscuous enzyme activities, but also revealed that acetyl-leucine was produced in *E. coli*, which was not known prior to that study. In our second example, the DeepEC deep learning method [5] was used to find alternative EC numbers for YgbJ, an L-threonate dehydrogenase (annotated 1.1.1.411 in Uniprot). For YgbJ, DeepEC predicted an oxidoreductase activity on D-glycerate (1.1.1.60). A follow up enzymatic assay revelated that YgbJ was indeed able to metabolize both D-glycerate and L-threonate.

3. Active Learning and Reinforcement Learning

Active learning (AL) is a special case of supervised machine learning, where a learner (any learning algorithm mentioned in the previous section) can interactively query an oracle (a human, a robot, a computer simulation) to ask new data points to be labeled [21]. The process is iterative and the training set is acquired and growing on the fly. Since the learner chooses the examples to be labeled, the number of examples can be made lower than the number required in normal supervised learning while maintaining performances. For instance, searching novel substrates for a small set of four promiscuous enzymes, it was shown that SVMs trained on a set of substrates selected by AL performed with 80% accuracies using 33% fewer compounds than when trained on the whole set of substrates [22]. AL is particularly appealing in the context of bioengineering since it reduces the number of experiments to be performed. Additionally, AL perfectly fits the Design-Build-Test and Learn (DBTL) cyclic process developed in synthetic biology as it proposes a solution to the Learning step of the cycle [23].

AL is illustrated in Figure 2 to search alternative substrates metabolized by promiscuous enzymes. The process starts by asking to label (i.e. perform measurements) an initial set of data points (enzyme x substrate pairs). Each data point is described by features, one can use for instance chemical fingerprints for substrates and string spectrum or one-hot encoding for sequences. The initial set can be generated by choosing enzyme x substrate pairs at random, or better using fractional factorial design [24] to evenly sample the space of possibilities. Measurements are then acquired, eventually using robotic screening equipment, and the pairs with activity measurements are added to the labeled dataset. Next, a machine learning algorithm is trained on the labeled dataset and used to predict labels from features for all the pairs in the unlabeled set (or a sample of pairs if the whole unlabeled set is too large). Methods mentioned in section 2 like the tensor product can be utilized to perform the predictions. Based on predictions carried out on the unlabeled dataset, a new set of enzyme x substrate pairs is selected for the next round of measurement. The selected pairs are screened, the new measurements are added to the growing training set and the trained model is retained. The process is iterated until the performances of the learner cannot be improved.

The AL process illustrated in Figure 2 is generic, it can and has been applied to other biosynthesis and metabolic engineering relevant problems like finding the expression level of enzymes in a pathway producing a target molecule [25][26][27][28] or finding buffer composition maximizing cell-free productivity [29]. As an example, coupling robotic equipment with AL, HamediRad *et al*. [27] optimized lycopene biosynthetic pathway evaluating less than 1% of possible variants while outperforming random screening by 77%. One critical step in AL is the selection of new data points to be labeled. AL makes use of two selection modes, exploitation and exploration. In exploitation mode and when maximizing an objective, AL is seeking predicted label values (μ) higher than those already in the training set, while in exploration mode AL is searching for predictions having high variances (σ), these corresponding to data points that are far away from those being in the training set. As shown in Borkowski *et al*. [29], the combination of exploitation and exploration via an Upper Confidence Bound (UCB) formula like $\mu + k\,\sigma$ (where k is a constant)

is efficient in large combinatorial spaces. Indeed, that paper demonstrated that less than 10 AL iterations were sufficient for a 34-fold cell-free productivity increase, while optimizing buffer composition in a combinatorial space > 4 10^6.

Reinforcement Learning (RL) is another technique that can be coupled with simulations or experimental measurements. RL was popularized by the Google DeepMind AlphaGO program [30]. It has since been used for retro-biosynthesis [31], synthesis planning of synthetic pathways with green process [32], and bioprocess optimization [33][34]. The Monte Carlo Tree Search (MCTS) RL method, developed for the AlphaGO program [30], is outlined in Figure 3.A in the context of retro-biosynthesis. Retro-biosynthesis consists of finding heterologous enzymatic reactions transforming the native metabolites of a host strain into a target molecule. Traditional breadth-first search retrosynthesis algorithms [35] proceed from the target (source) to the strain (sink) applying retro reaction rules (rules for reactions that have been reversed). The process is iterated layer by layer until a pathway is found ending in the sink. MCTS does not proceed breadth-first but instead makes use of 4 phases (selection, expansion, rollout and backpropagation), repeated until a number of iterations is reached. During expansion and rollout policy networks can be used to return the most appropriate transformations for the set of molecules of the selected node. Supervised or semi-supervised methods described in section 2 can be used to train these policy networks. Figure 3.B shows that number of targets successfully retrieved by RL (reported in [31]) is larger than the number obtained with a classical breadth-first algorithm [35].

4. *In vitro* and *in vivo* Learning

In all the applications we have seen so far, learning is performed *in silico*. In this section we are interested in performing learning *in vitro* or *in vivo*, the main challenge is therefore to be able to construct molecular devices processing information the same way as the basic blocks of machine learning programs. Two main goals motivate this innovative learning approach. The first, rather theoretical, is to probe to which extent cellular networks can be engineered to learn. The second more pragmatic is to develop diagnostics tools for pollutants or diseases [45] making use of *in vitro* or *in vivo* molecular circuits performing learning tasks like classification. Constructing electronic-like devices *in vivo* has been a long-standing endeavor of synthetic biology and many logic gates [36], switches [37], amplifiers [38], latches [39] and memory devices [40] can be found in the literature. Quite complex logic circuits can now be constructed [39] but building a circuit that would mimic the behavior of a machine learning code is still out of reach. Timing consideration is also a major issue as it takes generally half an hour (the time taken to transcribe and translate genes) to pass information from one circuit layer to another when implemented *in vivo*. Considering the time already taken to train an *in silico* machine learning model, trying to do this *in vivo* appears unreasonable. One strategy to overcome the complexity and timing issues is to train the circuits *in silico* and to construct *in vitro* or *in vivo* devices that reproduce the behavior of the trained circuits. That strategy has actually been followed to engineer molecular perceptrons, which are the basic units found in artificial neural networks. In a pioneer work, Quian *et al*. [41], built a 4 inputs Hopfield network (a recurrent neural network) using DNA strand displacement. This Hopfield network was trained *in silico* to remember four input patterns: 0110,

1111, 0011 and 1000. Weight were implemented changing the concentrations of the DNA strands used in the circuits.

In a more recent work, presented in Figure 4, Pandi *et al*. [42] trained a 4 inputs perceptron to classify 16 input patterns. Figure 4 shows a 16 input patterns perceptron implemented through a metabolic network expressed in cell-free systems. The input patterns are based on the presence or absence of four input metabolites (hippurate, cocaine, benzamide, and biphenyl-2,3-diol). In Figure 4.A**,** a targeted behavior is chosen arbitrary for a classification into positives and negatives. An *in silico* perceptron is trained via simple logistic regression to determine the weights best matching the targeted behavior. Figure 4.B shows enzymes metabolizing the input metabolites into benzoate, which is an activator of the transcription factor BenR. BenR is then used to express a Green Fluorescent Protein (GFP). In Figure 4.C a kinetics model (cf. Pandi *et al*. [42] for details) is used to determine the enzyme DNA concentrations corresponding to the weights calculated in panel (A). Finally**,** the perceptron is constructed in Figure 4.D using the enzyme determined in panel (B) and the concentrations of the enzyme DNA calculated in panel (C). The observed behavior (relative fluorescent unit, RFU) matches well the targeted behavior and the kinetics model predictions (red circles). Other classifiers can be constructed using the same setup simply changing the weights and the corresponding concentrations of enzyme DNA (cf. Pandi *et al*. [42] for other examples).

As the last example, a trained perceptron was constructed *in vivo* to classify 12 input patterns [43]. The trained perceptron was implemented engineering two *E. coli* strains: a sender and a receiver. The sender produced quorum molecules (acyl-homoserine lactone 3OHC14:1-HSL) and the receiver was engineered to respond upon detection of these molecules by expressing a fluorescent reporter. The perceptron weights were instantiated by varying the promoter strength, affecting the production level of the quorum molecules in the sender strains.

5. Conclusion and perspectives

The use of machine learning in biology will continue to grow. In fact, a search on bioRxiv with the key words “deep learning”, returns about 450 manuscripts deposited each month for the last year and that number nearly doubled between march 2020 (370) and march 2021 (682). However, the number of published papers actually prompting design of experiments and new experimental finding is much smaller. That number will undoubtfully increase as machine learning techniques are being interfaced with robotized workstations allowing to automate engineering as currently being done in chemistry with synthesis planning [44]. One area of particular interest to synthetic biology is the development of molecular devices enabling *in vitro* or *in vivo* learning like the perceptrons presented in section 4. Aside from finding practical applications with biomarkers detection and decision making for medical diagnostics [45], such devices could also be used to probe to what extent molecular and cellular networks can handle problems currently solved *in silico* and even shed some lights on how cognition could emerge from basic molecular circuits, a fundamental and long-standing question [46].

The neural networks of our brains inspired the development of artificial neural networks, perhaps artificial neural networks can now prompt the discovery and engineering of new learning molecular devices in living systems.

ACKNOWLEDGMENTS

J.-L.F. would like to acknowledge funding provided by the ANR funding agency grant numbers ANR-15-CE21-0008, ANR-17-CE07-0046, and ANR-18-CE44-0015. L.F. is supported by INRAE's MICA department INRAE's metaprogram BIOLPREDICT.

REFERENCES

[1] P. Carbonell, T. Radivojevic, and H. García Martín, "Opportunities at the Intersection of Synthetic Biology, Machine Learning, and Automation," *ACS Synth. Biol.*, vol. 8, no. 7, pp. 1474–1477, Jul. 2019, doi: 10.1021/acssynbio.8b00540.

[2] P. Larranaga *et al.*, "Machine learning in bioinformatics," *Brief. Bioinform.*, vol. 7, no. 1, pp. 86–112, Mar. 2006, doi: 10.1093/bib/bbk007.

[3] J.-L. Faulon, M. Misra, S. Martin, K. Sale, and R. Sapra, "Genome scale enzyme–metabolite and drug–target interaction predictions using the signature molecular descriptor," *Bioinformatics*, vol. 24, no. 2, pp. 225–233, 2008. * The study presents for the first time a tensor vector product merging biological sequences and chemical structures. The paper shows that accuracies are higher with the tensor product than when learning is performed separately on biological or chemical information.

[4] Y. Li *et al.*, "DEEPre: sequence-based enzyme EC number prediction by deep learning," *Bioinformatics*, vol. 34, no. 5, pp. 760–769, Mar. 2018, doi: 10.1093/bioinformatics/btx680.

[5] J. Y. Ryu, H. U. Kim, and S. Y. Lee, "Deep learning enables high-quality and high-throughput prediction of enzyme commission numbers," *PNAS*, vol. 116, no. 28, pp. 13996–14001, Jul. 2019, doi: 10.1073/pnas.1821905116. * This study is one of the first to make use of deep learning in the context of EC number prediction.

[6] A. Sureyya Rifaioglu, T. Doğan, M. Jesus Martin, R. Cetin-Atalay, and V. Atalay, "DEEPred: Automated Protein Function Prediction with Multi-task Feed-forward Deep Neural Networks," *Sci Rep*, vol. 9, no. 1, p. 7344, Dec. 2019, doi: 10.1038/s41598-019-43708-3.

[7] A. W. Senior *et al.*, "Improved protein structure prediction using potentials from deep learning," *Nature*, vol. 577, no. 7792, Art. no. 7792, Jan. 2020, doi: 10.1038/s41586-019-1923-7. ** The paper presents a deep learning strategy to predict protein structures with high accuracies. The method is exemplified with results obtained for the CASP13 competition (Critical Assessment of Protein Structure Prediction).

[8] Y. Wang, H. Wang, L. Liu, and X. Wang, "Synthetic Promoter Design in *Escherichia coli* based on Generative Adversarial Network," Bioinformatics, preprint, Feb. 2019. doi: 10.1101/563775.

[9] J. A. Valeri *et al.*, "Sequence-to-function deep learning frameworks for engineered riboregulators," *Nat Commun*, vol. 11, no. 1, p. 5058, Dec. 2020, doi: 10.1038/s41467-020-18676-2.

[10] N. M. Angenent-Mari, A. S. Garruss, L. R. Soenksen, G. Church, and J. J. Collins, "A deep learning approach to programmable RNA switches," *Nat Commun*, vol. 11, no. 1, p. 5057, Dec. 2020, doi: 10.1038/s41467-020-18677-1.

[11] J. Wang, H. Cao, J. Z. H. Zhang, and Y. Qi, “Computational Protein Design with Deep Learning Neural Networks,” *Sci Rep*, vol. 8, no. 1, p. 6349, Dec. 2018, doi: 10.1038/s41598-018-24760-x.
[12] A. Zelezniak *et al.*, “Machine Learning Predicts the Yeast Metabolome from the Quantitative Proteome of Kinase Knockouts,” *Cell Systems*, vol. 7, no. 3, pp. 269-283.e6, Sep. 2018, doi: 10.1016/j.cels.2018.08.001.
[13] W. Peng *et al.*, “The artificial neural network approach based on uniform design to optimize the fed-batch fermentation condition: application to the production of iturin A,” *Microbial Cell Factories*, vol. 13, no. 1, p. 54, Apr. 2014, doi: 10.1186/1475-2859-13-54.
[14] A. A. K. Nielsen and C. A. Voigt, “Deep learning to predict the lab-of-origin of engineered DNA,” *Nature Communications*, vol. 9, no. 1, Art. no. 1, Aug. 2018, doi: 10.1038/s41467-018-05378-z.
[15] P. Chen, J. Z. Huang, and X. Gao, “LigandRFs: random forest ensemble to identify ligand-binding residues from sequence information alone,” *BMC Bioinformatics*, vol. 15, no. Suppl 15, p. S4, 2014, doi: 10.1186/1471-2105-15-S15-S4.
[16] J. Mellor, I. Grigoras, P. Carbonell, and J.-L. Faulon, “Semisupervised Gaussian Process for Automated Enzyme Search,” *ACS Synth Biol*, vol. 5, no. 6, pp. 518–528, 17 2016, doi: 10.1021/acssynbio.5b00294. ** This the first paper where semi-supervised learning is used to predict enzymatic activity. The study also prompted experimental validation to discover a novel *E. coli* native metabolite .
[17] S. Martin, D. Roe, and J.-L. Faulon, “Predicting protein-protein interactions using signature products,” *Bioinformatics*, vol. 21, no. 2, pp. 218–226, Jan. 2005, doi: 10.1093/bioinformatics/bth483.
[18] H. Yabuuchi *et al.*, “Analysis of multiple compound–protein interactions reveals novel bioactive molecules,” *Mol Syst Biol*, vol. 7, p. 472, Mar. 2011, doi: 10.1038/msb.2011.5.
[19] P. Carbonell and J.-L. Faulon, “Molecular signatures-based prediction of enzyme promiscuity,” *Bioinformatics*, vol. 26, no. 16, pp. 2012–2019, Aug. 2010, doi: 10.1093/bioinformatics/btq317.
[20] L. Käll, J. D. Canterbury, J. Weston, W. S. Noble, and M. J. MacCoss, “Semi-supervised learning for peptide identification from shotgun proteomics datasets,” *Nature Methods*, vol. 4, no. 11, Art. no. 11, Nov. 2007, doi: 10.1038/nmeth1113.
[21] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, “Active Learning with Statistical Models,” *jair*, vol. 4, pp. 129–145, Mar. 1996, doi: 10.1613/jair.295.
[22] D. A. Pertusi, M. E. Moura, J. G. Jeffryes, S. Prabhu, B. Walters Biggs, and K. E. J. Tyo, “Predicting novel substrates for enzymes with minimal experimental effort with active learning,” *Metabolic Engineering*, vol. 44, pp. 171–181, Nov. 2017, doi: 10.1016/j.ymben.2017.09.016. ** This paper presents the first attempt to use active learning in the context of enzymatic activity prediction.
[23] J. Nielsen and J. D. Keasling, “Engineering Cellular Metabolism,” *Cell*, vol. 164, no. 6, pp. 1185–1197, Mar. 2016, doi: 10.1016/j.cell.2016.02.004.
[23] J. Nielsen and J. D. Keasling, “Engineering Cellular Metabolism,” *Cell*, vol. 164, no. 6, pp. 1185–1197, Mar. 2016, doi: 10.1016/j.cell.2016.02.004.
[24] G. Hanrahan and K. Lu, “Application of Factorial and Response Surface Methodology in Modern Experimental Design and Optimization,” *Critical Reviews in Analytical Chemistry*, vol. 36, no. 3–4, pp. 141–151, Dec. 2006, doi: 10.1080/10408340600969478.
[25] A. J. Jervis *et al.*, “Machine Learning of Designed Translational Control Allows Predictive Pathway Optimization in Escherichia coli,” *ACS Synth Biol*, vol. 8, no. 1, pp. 127–136, 18 2019, doi: 10.1021/acssynbio.8b00398.
[26] P. Opgenorth *et al.*, “Lessons from Two Design–Build–Test–Learn Cycles of Dodecanol Production in *Escherichia coli* Aided by Machine Learning,” *ACS Synth. Biol.*, vol. 8, no. 6, pp. 1337–1351, Jun. 2019, doi: 10.1021/acssynbio.9b00020.

[27] M. HamediRad, R. Chao, S. Weisberg, J. Lian, S. Sinha, and H. Zhao, “Towards a fully automated algorithm driven platform for biosystems design,” *Nat Commun*, vol. 10, no. 1, p. 5150, Dec. 2019, doi: 10.1038/s41467-019-13189-z. ** The paper presents an automated platform coupled with active learning to optimize the production of metabolic pathways.
[28] Y. Zhou, G. Li, J. Dong, X. Xing, J. Dai, and C. Zhang, “MiYA, an efficient machine-learning workflow in conjunction with the YeastFab assembly strategy for combinatorial optimization of heterologous metabolic pathways in Saccharomyces cerevisiae,” *Metabolic Engineering*, vol. 47, pp. 294–302, May 2018, doi: 10.1016/j.ymben.2018.03.020.
[29] O. Borkowski *et al.*, “Large scale active-learning-guided exploration for in vitro protein production optimization,” *Nat Commun*, vol. 11, no. 1, p. 1872, Apr. 2020, doi: 10.1038/s41467-020-15798-5. ** This study is making use of artificial neural networks coupled with active learning to boost the productivity of cell-free systems. The optimized system has a productivity substantially higher than those obtained using standard protocols.
[30] D. Silver *et al.*, “Mastering the game of Go with deep neural networks and tree search,” *Nature*, vol. 529, no. 7587, Art. no. 7587, Jan. 2016, doi: 10.1038/nature16961.
[31] M. Koch, T. Duigou, and J.-L. Faulon, “Reinforcement Learning for Bioretrosynthesis,” *ACS Synth Biol*, vol. 9, no. 1, pp. 157–168, Jan. 2020, doi: 10.1021/acssynbio.9b00447. * Reinforcement learning is used in the context of retro-biosynthesis, solutions are scored making use of enzyme availability among other criteria.
[32] X. Wang *et al.*, “Towards efficient discovery of green synthetic pathways with Monte Carlo tree search and reinforcement learning,” *Chem. Sci.*, vol. 11, no. 40, pp. 10959–10972, Oct. 2020, doi: 10.1039/D0SC04184J.
[33] B. J. Pandian and M. M. Noel, “Control of a bioreactor using a new partially supervised reinforcement learning algorithm,” *Journal of Process Control*, vol. 69, pp. 16–29, Sep. 2018, doi: 10.1016/j.jprocont.2018.07.013.
[34] P. Petsagkourakis, I. O. Sandoval, E. Bradford, D. Zhang, and E. A. del Rio-Chanona, “Reinforcement learning for batch bioprocess optimization,” *Computers & Chemical Engineering*, vol. 133, p. 106649, Feb. 2020, doi: 10.1016/j.compchemeng.2019.106649.
[35] B. Delépine, T. Duigou, P. Carbonell, and J.-L. Faulon, “RetroPath2.0: A retrosynthesis workflow for metabolic engineers,” *Metab. Eng.*, vol. 45, pp. 158–170, 2018, doi: 10.1016/j.ymben.2017.12.002.
[36] A. A. K. Nielsen *et al.*, “Genetic circuit design automation,” *Science*, vol. 352, no. 6281, Apr. 2016, doi: 10.1126/science.aac7341.
[37] A. A. Green, P. A. Silver, J. J. Collins, and P. Yin, “Toehold switches: de-novo-designed regulators of gene expression,” *Cell*, vol. 159, no. 4, pp. 925–939, Nov. 2014, doi: 10.1016/j.cell.2014.10.002.
[38] J. Bonnet, P. Yin, M. E. Ortiz, P. Subsoontorn, and D. Endy, “Amplifying genetic logic gates,” *Science*, vol. 340, no. 6132, pp. 599–603, May 2013, doi: 10.1126/science.1232758.
[39] L. B. Andrews, A. A. K. Nielsen, and C. A. Voigt, “Cellular checkpoint control using programmable sequential logic,” *Science*, vol. 361, no. 6408, Sep. 2018, doi: 10.1126/science.aap8987.
[40] F. Farzadfard and T. K. Lu, “Synthetic biology. Genomically encoded analog memory with precise in vivo DNA writing in living cell populations,” *Science*, vol. 346, no. 6211, p. 1256272, Nov. 2014, doi: 10.1126/science.1256272.
[41] L. Qian, E. Winfree, and J. Bruck, “Neural network computation with DNA strand displacement cascades,” *Nature*, vol. 475, no. 7356, Art. no. 7356, Jul. 2011, doi: 10.1038/nature10262. ** In this pioneer work the authors make use of DNA strand displacement to encode a Hopfield (recurrent) neural network enabling to distinguish input patterns.
[42] A. Pandi *et al.*, “Metabolic perceptrons for neural computing in biological systems,” *Nat Commun*, vol. 10, no. 1, p. 3880, 28 2019, doi: 10.1038/s41467-019-11889-0. ** The authors engineer the

first metabolic network reproducing the behavior of a perceptron. The network is used to classify samples with different metabolic profiles.
[43] X. Li, L. Rizik, and R. Daniel, “Synthetic neural-like computing in microbial consortia for pattern recognition,” In Review, preprint, Sep. 2020. doi: 10.21203/rs.3.rs-82365/v1. * This paper presents an implementation of a perceptron *in vivo* using sender and receiver engineered strains.
[44] C. W. Coley *et al.*, “A robotic platform for flow synthesis of organic compounds informed by AI planning,” *Science*, vol. 365, no. 6453, p. eaax1566, Aug. 2019, doi: 10.1126/science.aax1566.
[45] P. L. Voyvodic *et al.*, “Plug-and-play metabolic transducers expand the chemical detection space of cell-free biosensors,” *Nat Commun*, vol. 10, no. 1, p. 1697, 12 2019, doi: 10.1038/s41467-019-09722-9.
[46] I. Tagkopoulos, Y. C. Liu, and S. Tavazoie, “Predictive behavior within microbial genetic networks,” *Science*, vol. 320, no. 5881, pp. 1313–7, Jun. 2008, doi: 10.1126/science.1154456.

FIGURES

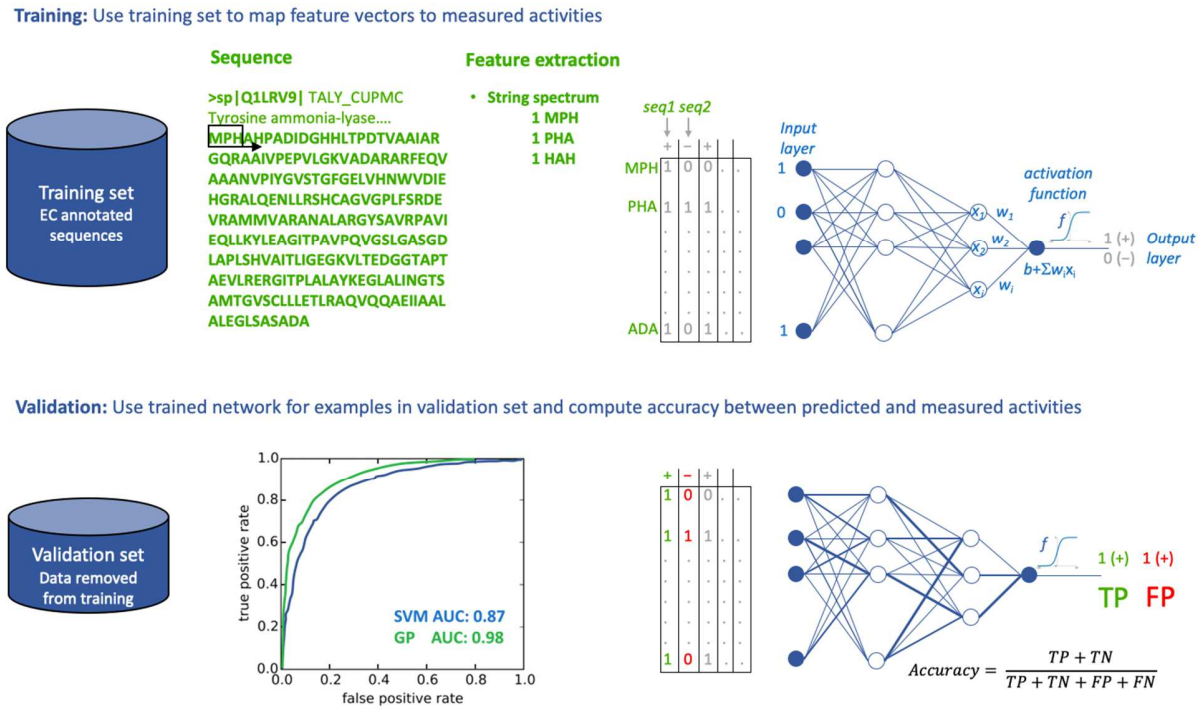

Figure 1. Typical Machine learning process to predict and validate enzyme activity from sequence. We assume here we have collected a set of sequences having a given EC number (at any level of the EC nomenclature) along with a set of sequences not having that EC number. We wish to learn if any given sequence belong to the chosen EC class or not. This is a classification problem. See main text for additional details.

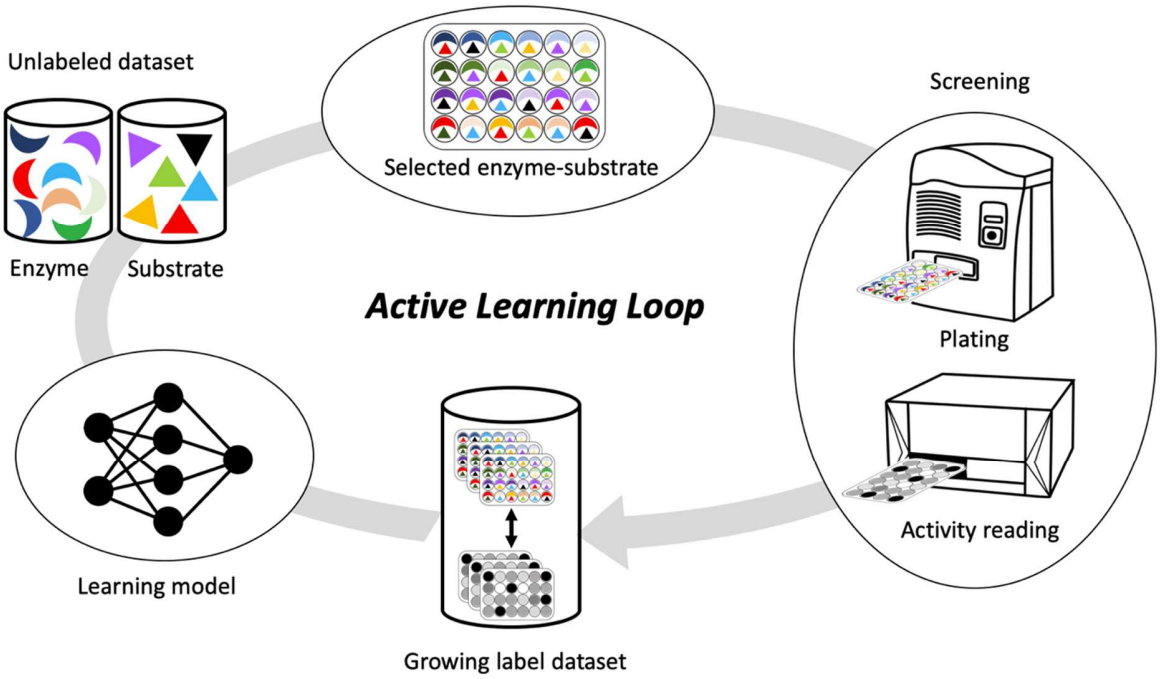

Figure 2. Active learning process applied to search for alternative substrates of promiscuous enzymes.

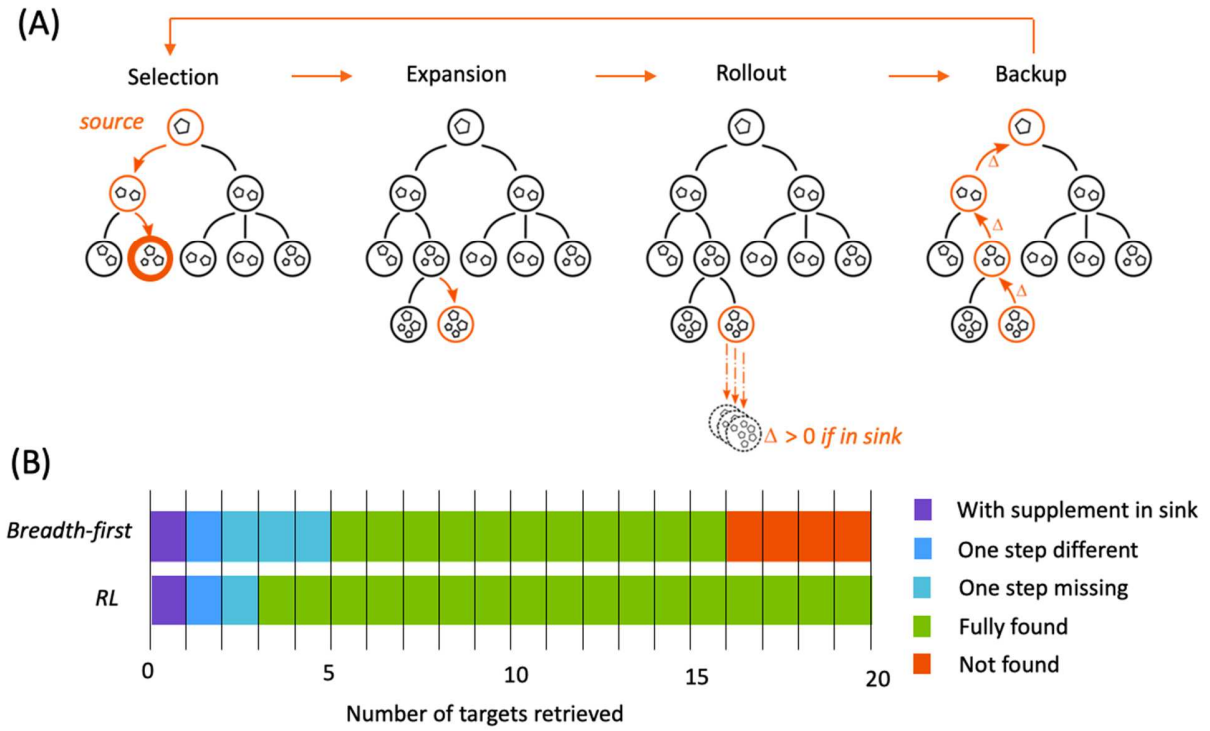

Figure 3. Reinforcement Learning (RL) with Monte-Carlo Tree Search (MCTS) illustrated with retro-biosynthesis. **(A) Monte Carlo Tree Search method (MCTS)**. Circles represent nodes and pentagons molecules. ***Selection***. Starting from the root node (here, a chemical state containing the target compound), the best child nodes are iteratively chosen until a leaf node is reached. Typical selection policies are based on exploitation and exploration. Exploitation is computed from a reward value (Δ) received in previous iterations of the algorithm (nodes with high values are favored) and exploration is based on the number of times a node has been visited (nodes with low number of visits are favored). ***Expansion***. Possible transformations are applied on the selected node generating new children. ***Rollout***. If the node is not terminal (the molecules are not in the sink or the maximum number of iterations is not reached) a transformation is sampled from available transformations and the process is repeated. If the node is terminal, a reward (if in sink) or penalty (if not in sink) is returned. Rollout is repeated until a maximum number of steps or the maximal depth of the tree is reached. ***Backpropagation or update***. The reward obtained after exploring the expanded node is returned to its parents to update their values (Δ) and visit counts. **(B) Performances for a Golden Set of 20 experimental pathways** (cf. [31]). With supplementation (purple) means a supplement has to be provided in the media to identify the correct experimental pathway. One step different (dark blue) means only one step differs from the described pathway, for example, by using a different co-substrate. One step missing (light blue) means the search algorithm found a pathway identical to the experimental one, except one step which was short-cut. Fully found (green) means the experimental pathway was found without restriction. Not found (orange) means the experimental pathway was not found.

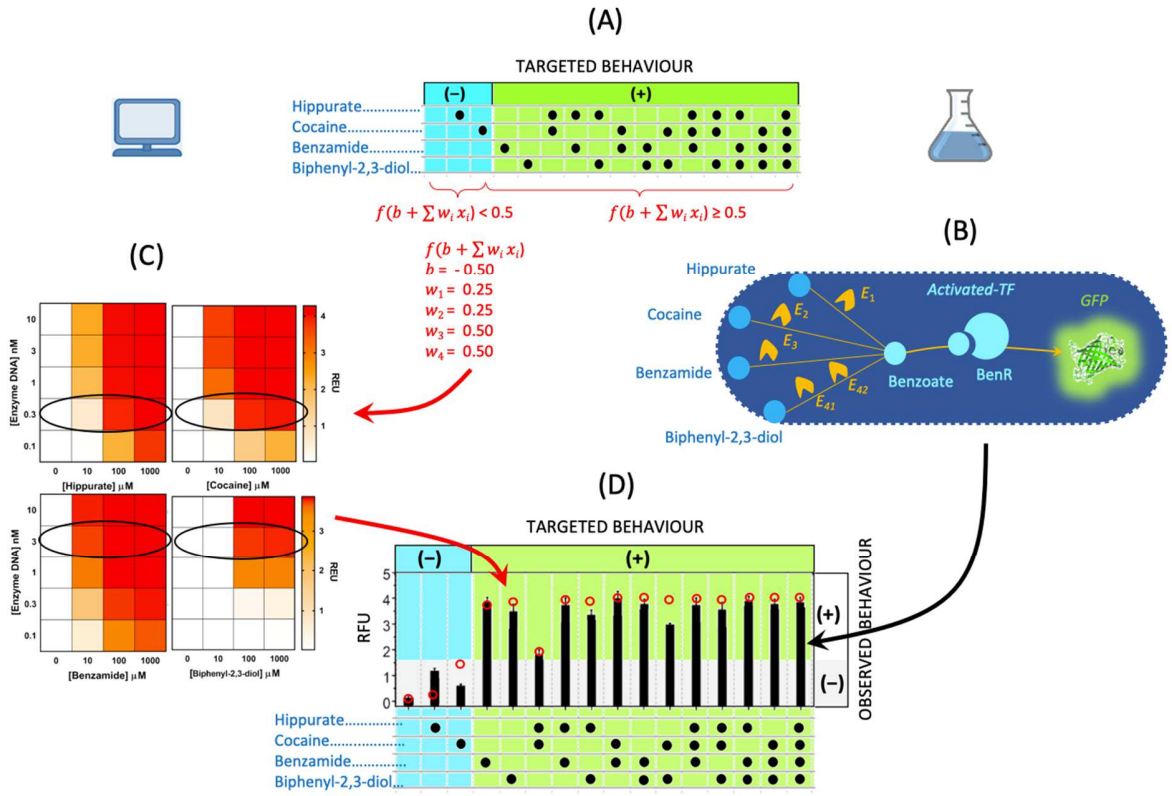

Figure 4. Building a metabolic perceptron. See main text for description.