Engineered gene circuits with reinforcement learning allow bacteria to master gameplaying

Adrian Racovita[1]†, Satya Prakash[2]†, Clenira Varela[2], Mark Walsh[2], Roberto Galizi[3], Mark Isalan[4], Alfonso Jaramillo[1,2]*

[1]*De novo* Synthetic Biology Lab, I2SysBio, CSIC-University of Valencia, Paterna, Spain.

[2]School of Life Sciences, University of Warwick, Coventry, UK.

[3]Centre for Applied Entomology and Parasitology, School of Life Sciences, Keele University, Keele, UK.

[4]Department of Life Sciences, Imperial College London, London, UK.

*Corresponding author. Email: Alfonso.Jaramillo@synth-bio.org

†These authors contributed equally to this work

Abstract:

The engineering of living cells able to learn algorithms by themselves, such as playing board games —a classic challenge for artificial intelligence— will allow complex ecosystems and tissues to be chemically reprogrammed to learn complex decisions. However, current engineered gene circuits encoding decision-making algorithms have failed to implement self-programmability and they require supervised tuning. We show a strategy for engineering gene circuits to rewire themselves by reinforcement learning. We created a scalable general-purpose library of *Escherichia coli* strains encoding elementary adaptive genetic systems capable of persistently adjusting their relative levels of expression according to their previous behavior. Our strains can learn the mastery of 3x3 board games such as tic-tac-toe, even when starting from a completely ignorant state. We provide a general genetic mechanism for the autonomous learning of decisions in changeable environments.

One-Sentence Summary:

We propose a scalable strategy to engineer gene circuits capable of autonomously learning decision-making in complex environments.

Animal brains are powerful decision-making devices able to learn by themselves using reinforcement learning. Computational design methods have been used to experimentally implement biological adaptive behaviors(*1-7*), but not advanced decision making, which was achieved artificially using physical and chemical systems by engineering memory units with neural network computational capabilities(*8-14*). Engineered gene circuits could endow living cells with decision making capabilities, although their reprogramming has focused on modifying the encoding DNA, such as mutating and recombining regulatory regions(*5, 15*). This adaptation requires the directed rewiring(*16, 17*) of gene circuits, demanding the precise adjustment of every interaction, as when an experimenter computes a steepest descent method(*13*) to infer the needed experimental adjustments. This hampers the engineering of large systems where the individual adjustment of parameters would be impractical.

To dramatically simplify the capability to train a gene network towards a complex behavior, irrespective of its size, the computation and implementation of the required adaptation should be encoded in the gene network itself. We therefore propose a genetic strategy where gene circuits autonomously rewire themselves towards a targeted behavior by shifting plasmid heteroplasmy. We test the capability for learning complex decision-making in living bacteria through the gameplaying of board games, a common decision-making benchmark in artificial intelligence(*18*).

Inducible antibiotic resistance markers allow the adaptation of co-encoded genes via shifting plasmid ratios

We created a library of plasmids carrying 9 possible inducible promoter systems (Fig. 1A), which we transformed into the *Escherichia coli* DH10B Marionette strain (providing 12 chemically-driven promoters with minimal cross-talk activation) after we cured it of its former chloramphenicol resistance(*19*).

We co-transformed the cells with a mixture of two almost identical multi-copy plasmids, P1 and P2, both maintained by an ampicillin resistance gene (AmpR). We designed the P1 and P2 plasmids to stabilize their copy number ratios within a cell, by having the same length(*20*) (including the length of the fluorescent markers), a common promoter and a translational insulator sequence, as well as the same medium copy replicon. P1 and P2 encoded inducible operons for fluorescent proteins followed polycistronically by fusions of antibiotic resistance proteins (Kan^R/Cm^R for kanamycin/chloramphenicol resistance) or corresponding non-functional "dead" forms ($dKan^R$/dCm^R) (Fig. 1A). The Cm^R and Kan^R genes allow cellular antibiotic selection for higher P1 or P2 DNA-copy number (denoted by *a* and *b*, respectively) (as confirmed by flow cytometry experiments, Fig. S1). Cm^R or Kan^R selection thus shifts the P1:P2 plasmid ratios because the total plasmid copy number (*a*+*b*) is conserved.

We define the fraction of the P1 plasmid in cells co-transformed with both P1 and P2 plasmids (*a*/(*a*+*b*)) as *weight* (Fig. 1A), analogously to artificial neural networks (ANN). The distinguishable fluorescent proteins in P1 and P2 allow the convenient utilization of their ratio for an accurate estimation of the weight from fluorescence alone (confirmed via qPCR, R^2=0.94, and mixed-read Sanger sequencing(*21*), R^2=0.99, Fig. S2). We have therefore chosen fluorescence measurements to measure *weight* and referred to the corresponding values as *F-weight*.

The strains co-transformed with the plasmids P1 and P2 realize a minimal gene circuit, which we call *memregulon* (contraction for *memory regulon*, analogous to the memristor element used in electronic circuits with neuromorphic behavior(*22*)), able to adjust its DNA levels according to

its promoter activation and antibiotic. A memregulon's red fluorescence agrees with its weight multiplied by the corresponding P1-only cells' red fluorescence (Fig. 1B and Fig. S16A). For instance, the mCherry expression of cells co-transformed with a 0.5 memregulon weight is expected to have 50% of the red fluorescence per cell than cells only transformed with the plasmid P1. Usually this would require reengineering the mCherry promoter with suitable mutations. Fig. 1B shows that the 0.5 weight cells have indeed a red fluorescence per cell significantly identical to half of the induced and non-induced values of cells transformed only with the P1 plasmid ($p<0.01$), which contrasts with the inability of transcription regulation to lower the non-induced values.

In the following, we grow and maintain bacterial cultures in agar plates. We can measure the weights in a memregulon culture using fluorescence or DNA sequencing, where we always copy the cultures with a replica plating to preserve the original plate (Fig. 1C). We show that the *weight* remains constant at the population level for many days and consecutive replica plating procedures (Fig. 1D), effectively functioning as a genetic memory system(*20*).

Memregulons produce gene circuits able to adapt their expression levels by self-modifying their DNA copies

Memregulon weight can be altered by culturing cells with specific antibiotics and promoter inducers. For example, kanamycin or chloramphenicol respectively decreases or increases the *weight,* corresponding to reduction or increment of mCherry fluorescence levels. In agar plates, we produce a new parental plate through a replica plating where the destination plate will contain selection antibiotic, ampicillin and the cognate inducers (Fig. 1E). We stop the antibiotic selection after a calibrated time with a subsequent replica plating using only ampicillin. We consider the memregulon as activated when its promoter has been fully induced by the cognate inducer; only in this state can the memregulon show fluorescence and change its weight significantly ($p<0.01$, Fig. S3).

As the promoters might have had small crosstalks with noncognate inducers, we measured the change in *weight* in the presence of cognate and non-cognate inducers, which showed significant variation in *weight* ($p< 0.01$) only for the cognate inducer (Fig. 1F, S4 and S5). This allows culturing different memregulons together, where each memregulon can independently adjust its weight. Instead of manually modifying the mCherry expression levels through external manipulations of the plasmid DNA copy number, we allow the mCherry operon to persistently set its own P1 copy number levels (*i.e.* mCherry expression levels) via selection pressures from kanamycin/chloramphenicol and cognate inducer activation. This enables the local and unsupervised training of *weights*, a desired feature in the training of ANN(*23*).

We can use the combined output (*e.g.* fluorescence) of a set of active memregulons for decision making. If the output is not desired, we then reproduce the environmental condition, activating memregulons in the presence of kanamycin/chloramphenicol to downregulate/upregulate their expression, thus contributing to decision making. This allows training by self-programming of the decision-making, by a stepwise downregulation/upregulation of the memregulons' contribution to wrong/correct decisions.

Choosing the highest weight among independent memregulon cocultures allows an experimental reinforcement learning algorithm to find the optimal path in decision trees

The distributed multicellular circuits (DMC)(*24, 25*) strategy allows us to explore whether a coculture of strains, containing memregulons, can learn complex decision-making. For this, we

initially challenged the cultures with a mathematical problem equivalent to solving a maze (Fig. 2A), where a "rat" must find the path to the exit without backtracking. Although this corresponds to one of the simplest decision trees, it will allow defining the methodology to be used for more complex problems. Paths encounter crossings with 3 possible diversions each. Among the maze's four crossings, the one encountered first is designated the inducer L-arabinose (Ara) and the others the inducer 3-hydroxytetradecanoyl-homoserine lactone (OHC14). The optimal path follows the diversions **1** and **2** at the crossings **a** and **b**. To choose the diversion at each crossing, we set up three cocultures of two strains at a 1:1 cell ratio. Each coculture contains strains with the pBAD and pCin memregulons of different weights, encoding Marionette promoters(*19*), inducible under the chemicals Ara and OHC14 respectively. The 'chosen' diversion at the **a** and **b** crossings is defined as the number of the coculture with the highest pBAD and pCin weight, allowing generating an integer value from a vector of analog values. The starting cultures were picked such that they had weights where their initial decisions were the furthest from the optimal path. The weights are measured after replica plating measurement of the red and green fluorescence, adding the chemical inducer designated to the crossing (Fig. 2B). If the two decisions (**a**, **b**) do *not* correspond to the unique path towards the exit, we apply a "punishment" selection using a destination plate containing kanamycin, ampicillin, and the inducers Ara and OHC14. We repeat this cycle of measurement and negative reinforcement learning twice until no more kanamycin selection is needed because the memregulons have modified their weights to encode the output of the optimal path (Fig. 2C). Controls where the learning is done with either swapped antibiotics or swapped inducers show no change in decisions (Fig. S6).

Generalizing the experimental reinforcement learning algorithm allows finding the optimal strategy in the tic-tac-toe game

Because memregulon cocultures maintain stable their constituent memregulon weights (Fig. S7), we investigated whether the use of additional promoters could scale up the complexity of problems by challenging cocultures of memregulon strains to learn mastering a board game. As done with the early computers, we chose the familiar tic-tac-toe game, a two-player game on a 3x3 board, where the two players ("X" and "O") alternately occupy one vacant board position; the winner is the first player obtaining 3 matching symbols on any row, column, or diagonal. This game was studied recently using DNA computing(*11, 26*), which required implementing custom 3-input logic gates with catalytic DNA. However, it is not necessary to implement combinatorial gates to implement expert players if decisions are made by choosing the highest weight (called winner-take-all, WTA, strategy) even when using linear positive weights (*27*) (Fig. S8).

It is also useful to define a measure of the general skill level at a game, alike to the Elo ranking(*28*). Thanks to the small size of a 3x3 board game, we can use a computer simulation to play all possible games. For this, we input the measured F-weights into a simulation parametrized with our experimental data (supplementary text), where we evaluate the percentage of won or drawn games (called *expertise*) when playing all possible matches.

As an example of how reinforcement learning can automatically train the weights to achieve a complex computation, we generalized our previous experimental learning algorithm (Fig. 2B) to two-player games. We now consider one of the players to be a trainer (player X) and the other a bacterial player. The O player consists of a set of cocultures, one for each of the board positions, excluding the central (played first by player X) (Fig. 3A, left). We assigned a chemical inducer to each of the 9 board positions (Fig. 3A, right). The cells play a match against an opponent by reading their F-weights through replica plating fluorescence measurements (Fig. 3B). The

experimental algorithm is as follows (see Fig. 3C): As in the maze example, the chemicals activate the memregulons' promoters involved in a decision vertex (acting as a "leaf" selector in the decision tree), but now the simultaneous use of more than one inducer to measure the F-weights allows the identification of all the opponent's positions. The highest "multi-inducer" F-weight, among memregulon cocultures at unoccupied positions, "chooses" the bacteria's next move. After several rounds, the match finishes and, if the O player loses, we apply a negative reinforcement learning operation to the O cocultures, assigned to positions occupied by the O player (Fig. 3C). This updates the parental cultures and we play new matches until the player O achieves mastery (100% expertise).

An example of a match is overviewed in Fig. 3D: After player X starts at the center (round 0), player O could move at any of the other 8 unoccupied positions and, therefore, we consider cocultures at all of them. We do replica F-weight measurements to the cocultures by inducing them with 3-oxohexanoyl-homoserine lactone (OC6, inducer assigned to the center position, where X has moved) and then we choose the position where its coculture had the highest F-weight. In the next round, X makes another move (corresponding to the position assigned to Ara) and we inquire about O's move by inducing the 6 cocultures (at unoccupied positions) with OC6 and Ara (the two positions currently occupied by X), and measuring the highest F-weight among them. O loses in round 3, so we apply a negative reinforcement operation with kanamycin selection (Fig. 1E), in the cocultures at positions previously occupied by O (Fig. 3D, encircled in green), adding all the inducers corresponding to X's moves before round 3 (OC6, Ara, & OHC14), which lead to the losing decisions of O. After this learning, we have updated 3 cocultures, which become new parental plates for replica measurements, together with the unchanged 5. Bacteria play new matches until a match ends in a draw and the O player achieves mastery.

In principle, a trainer can always choose strategies avoiding draws, although we do not impose any condition on the trainer. Two bacterial players can even learn together by playing each other. To test this, we set up 2 cocultures of 2 memregulon strains, both chosen to have some knowledge of the game (having X and O expertises of 90% and 48% respectively) and able to achieve mastery in few learnings. We performed a tournament of memregulon cocultures playing among themselves and applying positive or negative reinforcement to the players winning or losing matches. Both cocultures reached mastery after one match (Fig. S9).

A random player of 9-memregulon cocultures learns to master tic-tac-toe by playing using reinforcement learning

We asked if our experimental algorithm could train a naïve bacterial player O (playing uniformly random) to learn mastering the tic-tac-toe game. We chose bacterial cultures to be second player because the naïve player had a low starting expertise (20%). The starting cocultures (denoted by O_0) consist of the same 9 memregulon strains at equal cell ratios and equal weights at all 8 positions (Fig. 4A); this experimentally implements a random player because all positions have the same cultures and interrogating for the highest weight would give a random position. We performed all the experiments in 3 biological replicates:

O plays a tournament against a trainer player X (decision matrix in Table S1). The first match lasts for 5 rounds and ends with O losing. We show in Fig. 4B the F-weights of the cocultures at allowed positions as filled red circles (containing the F-weight value multiplied by 100). Their highest value represents O's decision (O's move in the next round). The match ends and player X wins the match, which triggers a negative reinforcement (L1) of the O_0 cocultures at the 4

positions occupied by O in round 4, to produce O_1. The weight decreases at those positions and the measurement of O_1 in round 0 shows that the position with the highest F-weight has changed, implying a different decision (Fig. 4B). After each learning, we also compute the expertise of each of the biological replicates (Fig. 4C). Tables S2-S11 detail the computation of the O player's expertise after each learning, by showing the results of using the measured F-weights to play every possible tic-tac-toe match. The cocultures continue the matches by losing each time in a different way, and suffering a negative reinforcement learning (L2 to L7) each time, which further changes the cocultures (O_2 to O_8). The expertise did not increase monotonously, but it reached 100% for all replicates in O_8. We also validated the mastery by letting the cocultures play against an expert automaton (Fig. 4B).

Although the O_8 cultures acquired their mastery by playing 8 games, they have the capability to win arbitrary matches (Fig. 4C, Fig. S10). As a positive control, we performed a single steepest-descent-like operation to manually train the weights, according to a computational calculation to obtain an expert player (O_{sd}) (supplementary text) (Fig. S11). Two alternative learning tournaments were performed as negative control, starting from O_7; using either negative reinforcement with a swapped inducer (O_7^a) or using chloramphenicol instead of kanamycin (O_7^b) did not improve the expertise, as the player lost against the expert automaton (Fig. 4A). We also verified that the cocultures maintained their expertise in time after cold storage (at 4 °C or -80 °C) (Fig. 4D, S12). Reinforcement learning also allowed naïve bacterial cocultures to reach mastery when acting as a first player (Fig. S13).

Memregulon cocultures can also learn mastering arbitrary 3x3 board games

To explore the capacity of a consortium of memregulon strains to learn arbitrary algorithms, we performed computer simulations of cocultures of 9 memregulons at every position of a 3x3 board except the center, showing that they can learn in less than 35 cycles (Fig. 4E) 98% of the possible games in this board (Fig. S14A). Moreover, trying to push the limits of learning, they could even learn how to simultaneously master more than one game at the same time, although not always (Fig. S14B). In some cases, we found that such repeated learning tournaments required enough negative reinforcement steps that some weights vanished (Fig. S15A). If a weight vanishes, the P1 plasmid is lost, and so is its ability to store a memory, because it is not possible to have a P1 and P2 mixture anymore. To rescue this, we add to the experimental algorithm an operation that we call *memregulon fusion*. For this, we mix each memregulon strain culture with another one that contains the same memregulon with a weight of 0.5. This mixture operation changes all weights by averaging each of them with 0.5. This averaging maintains the position with the highest weight, and therefore the player's expertise (Fig. 4F), while increasing the weights smaller than 0.5 (Fig. S15B).

To allow our experimental learning optimization to converge towards mastery on arbitrary games, we need to avoid getting non-expert players trapped in draws where no more learning occurs. For this, we further extended the experimental learning algorithm by applying a reinforcement learning using chloramphenicol (instead of kanamycin) for selection. After the last match where a negative reinforcement was applied, we incubated the cocultures with chloramphenicol and the inducers used in the match. We call this reinforcement "unlearning", mirroring a similar concept from machine learning(*29*). After one round of unlearning, the bacteria altered their decisions and therefore their expertise also changed, thus avoiding getting trapped in draws (Fig. 4F).

Discussion

We can better appreciate the computational power of our memregulon cocultures by identifying them with a single-layer artificial neural network of three 2-input neurons (maze example) or nine 9-input neurons (3x3 board games), with the only non-linearity coming from a winner-take-all (WTA) interaction among the neurons (decision on the highest weight). Such networks can be universal function approximators, even when using positive weights exclusively (*27*). The change of a weight only when a memregulon is active is central to learning. This follows Hebb's idea(*30*) that the changes in synaptic strength (weight) should be proportional to the presynaptic cell activity and to a function of the postsynaptic cell activity. Long-term potentiation and long-term depression would correspond to a weight increasing and decreasing respectively(*31*). Moreover, similarly to neuromodulated synaptic plasticity, because their change of weights requires the memregulon activity together with either kanamycin or chloramphenicol, these antibiotics act as neuromodulatory signals(*32*).

Memregulons also allow for the construction of gene circuits with predefined behaviors because the red fluorescence per cell linearly correlates with its weight (Fig. S16A). Although positive and negative reinforcement learning could be thought to be equivalent to positive and negative selections in directed evolution(*33*), here we do not have mutations, which allows for a smoother, faster and reversible traversing of the phenotypic landscape. Memregulons maintained their weight in solid cultures across many days, suggesting the possibility of using them in ecosystem-level gene circuits(*34*). It could be possible to enrich the computation capability by using different promoters in P1 and P2 (Fig. S16B), providing a mechanism to adapt the topology of gene circuits(*35*). Further developments could involve genetically encoding the computation of the maximum output among positions(*36*), negative selection markers(*37*), CRISPR to cleave(*20*) or regulate(*38*) the plasmid copies, engineered RNA replicons(*39*), engineered microbial ecosystems(*40*), as well as adding an extra memregulon library to each player, designed to receive the output of the first library through a cell-cell communication system, mimicking a hidden layer in a neural network. This would enable the processing of more complex information and, therefore, learning more advanced algorithms.

Adaptive gene circuits could already exist in prokaryotic or eukaryotic systems as a non-Darwinian adaptation tool(*41*). Heterozygotic mutations in multicopy plasmids(*42*), polyploid Archea(*43*) or in mitochondrial DNA (microheteroplasmy)(*44*) maintain the ratios of wild-type to intra-cellular mutations. As a mutation in a growth-altering gene under a regulation could suffice to set up a reinforcement learning, it may be possible to infer memregulons in nature by identifying a mapping among environmental conditions, genes, inducible promoters, and selection markers with their inactivating mutations. This mapping would establish in fact a language for "teaching" algorithms to these cells. Reinforcement learning with memregulons provides a strategy for the unsupervised adaptation of complex gene circuits with a large, unknown number of interactions, which will allow for the engineering of genetically encoded general-purpose computational devices capable of self-learning, opening the way to the engineering of synthetic living artificial intelligence.

Acknowledgments:

We acknowledge M. Kushwaha, M. Fuegger and T. Nowak for discussions.

Funding:

Ministerio de Ciencia e Innovacion PID2020-118436GB-I00 (AJ)

BBSRC BB/P020615/1 (MI, AJ)

EPSRC-BBSRC grant BB/M017982/1 (AJ)

EU grant 610730 (AJ)

School of Life Sciences departmental allocation, Keele University (RG)

Volkswagen Foundation grant LIFE 93 065 (MI)

Author contributions:

Conceptualization: AR, AJ

Software: AR, AJ

Formal analysis: AR, AJ

Methodology: AR, SP, AJ

Investigation: AR, SP, CV, MW, RG, AJ

Visualization: AR, AJ

Supervision: AJ

Writing – original draft: AR, AJ

Writing – review & editing: AR, SP, CV, MW, RG, MI, AJ

Competing interests:

Authors declare that they have no competing interests.

Data and materials availability:

All data are available in the main text or the supplementary materials.

Supplementary Materials

Materials and Methods

Supplementary Text

Figs. S1 to S19

Tables S1 to S13

References (*1–44*)

Data S1 to S15

Figure 1

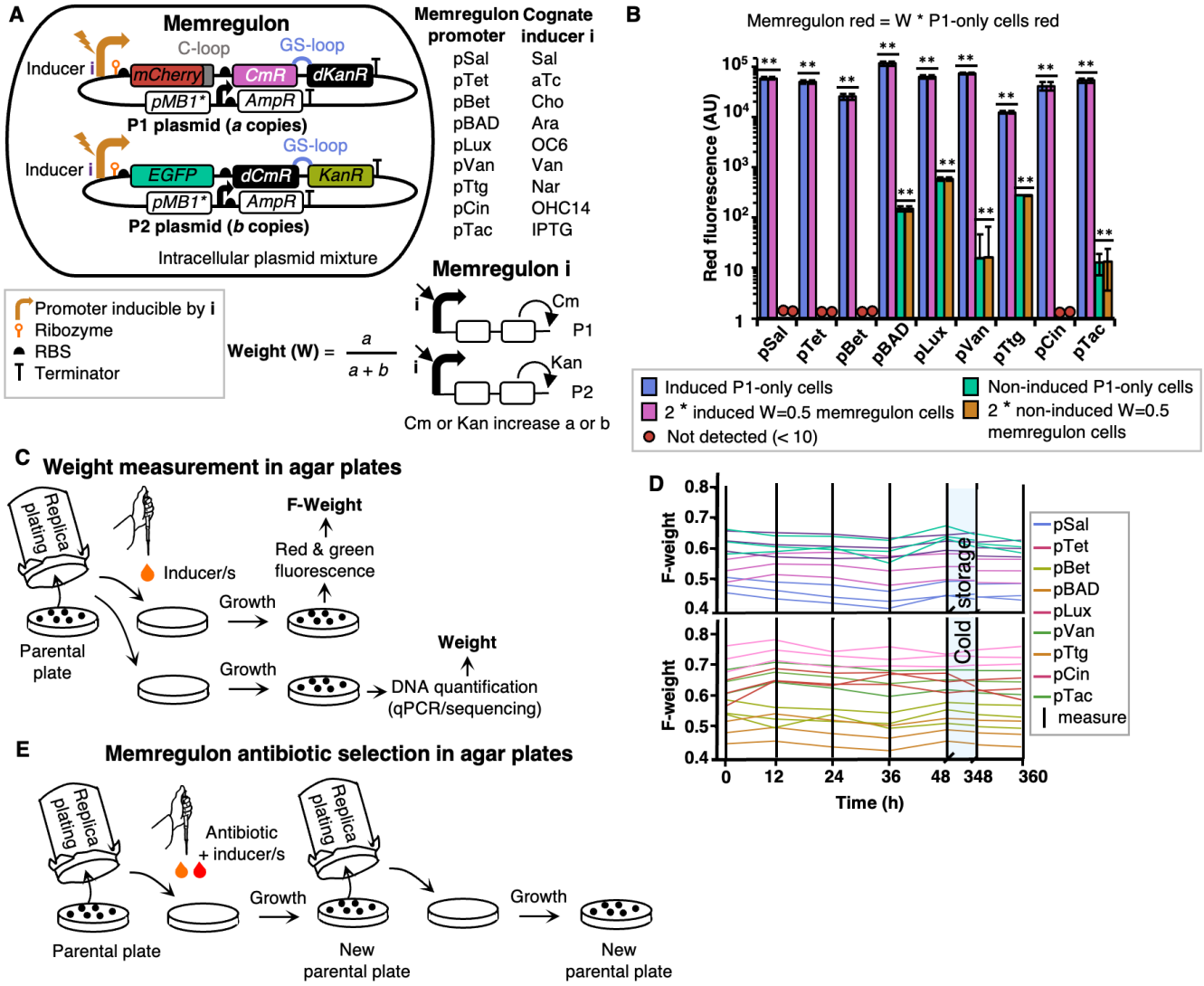

Fig. 1. Memregulons stably adapt their gene expression by varying plasmid copy-number ratios, when their promoter is activated by its cognate inducer. (**A**) Design of a *memregulon*, a stable heterozygotic plasmid system of two quasi-identical co-transformed plasmids P1 and P2, controlled by an inducible promoter (brown). Plasmids are designed such that, in the presence of ampicillin maintenance antibiotic, the number of DNA copies of plasmids P1 and P2 (*a* and *b* respectively) remains constant in a bacterial population, maintaining the P1 plasmid fraction (weight) stable. We engineered a library of 9 memregulons by using orthogonal inducible promoters (right), shown with their cognate inducers (Sal, Sodium salicylate; aTc, Anhydrotetracycline HCl; Cho, Choline chloride; Ara, L-Arabinose; OC6, 3OC6-AHL; Van, Vanillic acid; Nar, Naringenin; OHC14, 3OHC14:1-AHL; IPTG, Isopropyl-beta-D-thiogalactoside). The memregulons are designed so the use of chloramphenicol or kanamycin resistance gene (CmR or KanR), in the presence of the cognate chemical inducer and appropriate selection antibiotic, implements a positive feedback on the number of copies of the P1 or P2 plasmid. (**B**) Red fluorescence of cells containing only P1 plasmids is twice the fluorescence of their respective memregulon systems with 0.5 weight (W), after full induction (or no induction) with their cognate inducers. More generally, a memregulon's red fluorescence is found to agree with W multiplied by the corresponding P1-only cells' red fluorescence (Fig. S16A). A double asterisk denotes a statistically significant ($p<0.01$) identity in fluorescence value. (**C**) The weight of memregulon cocultures grown on agar plates is measured by either fluorescence characterization (F-weight) or DNA quantification (weight). We use a replica plating procedure to copy the cultures in the parental plate to a copy plate, where they are incubated at 37 °C. For fluorescence characterization (top) we add the appropriate inducers. One inducer for a single F-weight calculation or several inducers for the sum of the corresponding memregulon's F-weights. The plate is then photographed in a blue light transilluminator, and red and green fluorescence values are obtained through image analysis software to compute the F-weight (see supplementary text). While the copy plate is discarded after measurement, the parental plate is incubated at 37 °C for its regrowth. No inducers are added for DNA quantification (bottom). F-weights correlate well with the weights obtained by Sanger sequencing (see supplementary text). Error bars indicate SD from n = 3 biological population replicates obtained on 3 different days. Data shown for a library of 9 memregulons. (**D**) F-weights stay stable despite subsequent replica plating procedures (vertical bars). Moreover, fluorescence weights remain stable when the memregulon strains are stored for 2 weeks in 4 °C (cold storage blue band). (**E**) Memregulon weights can be changed by selection in the presence of Kanamycin/Chloramphenicol + inducers during exponential growth on agar plates. The concentrations of antibiotics are included in Table S13.

Figure 2

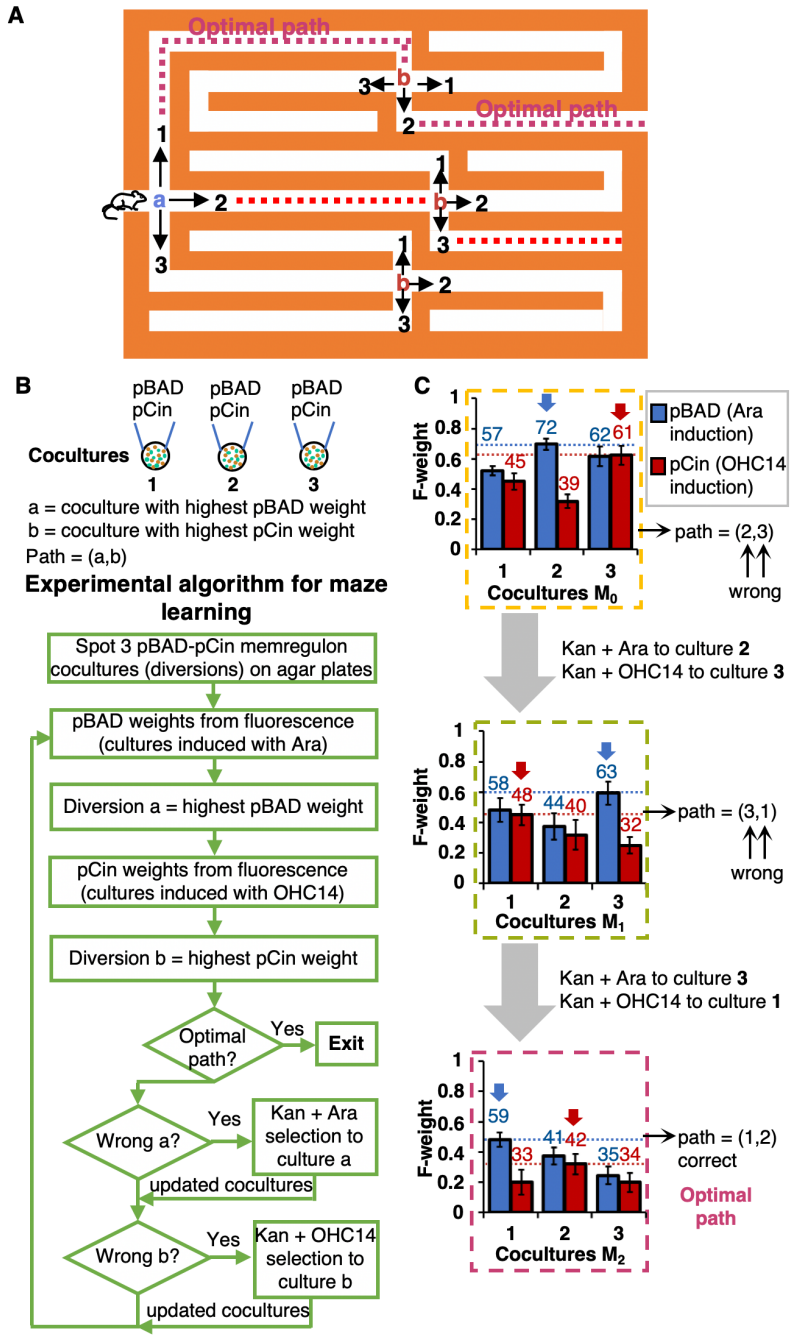

Fig. 2 Cocultures of two strains with different memregulons stably learn the decision tree of a maze by a negative reinforcement wet-algorithm. (**A**) We challenge the bacteria with a problem equivalent to finding the single optimal path in a maze of 4 crossings (1 of type **a** and 3 of type **b**), and 3 diversions each, where no backtracking is allowed. The paths are described as (*a*,*b*), where *a* is the diversion chosen at the **a** crossing and *b* is the diversion chosen at the **b** crossing. (**B**) Description of the experimental setup. Top: Setup of 3 cocultures, each composed of pBAD and pCin memregulon strains, at equal volumetric ratios but different weights. Bottom: flowchart with the experimental algorithm to find the optimal decisions at each crossing. *a* and *b* correspond to the coculture number with the highest F-weight when inducing with Ara and OHC14 respectively. This corresponds to measuring the pBAD and pCin memregulon weights. The algorithm stops when the optimal path (1,2) is found, otherwise if $a \neq 1$ we apply a kanamycin selection with Ara to the culture number *a*. If $b \neq 2$ we apply a kanamycin selection with OHC14 to the coculture number *b*. (**C**) F-weights measurement of the starting cocultures (M_0) obtained by inducing with Ara and OHC14, giving the pBAD and pCin memregulon weights respectively. Top: Measurement of the highest F-weights gives the (2,3) path, leading to wrong *a* and *b* values. We therefore punish the cocultures 2 and 3 with kanamycin + Ara and kanamycin + OHC14, respectively, to create the cocultures M_1. Middle: Highest F-weights of M_1 give the (3,1) path, with again incorrect *a* and *b* values. We again punish the cocultures 3 and 1 with kanamycin + Ara and kanamycin + OHC14, respectively, to create the cocultures M_2. Bottom: Highest F-weights of M_2 give the (1,2) path, solving the maze.

Figure 3

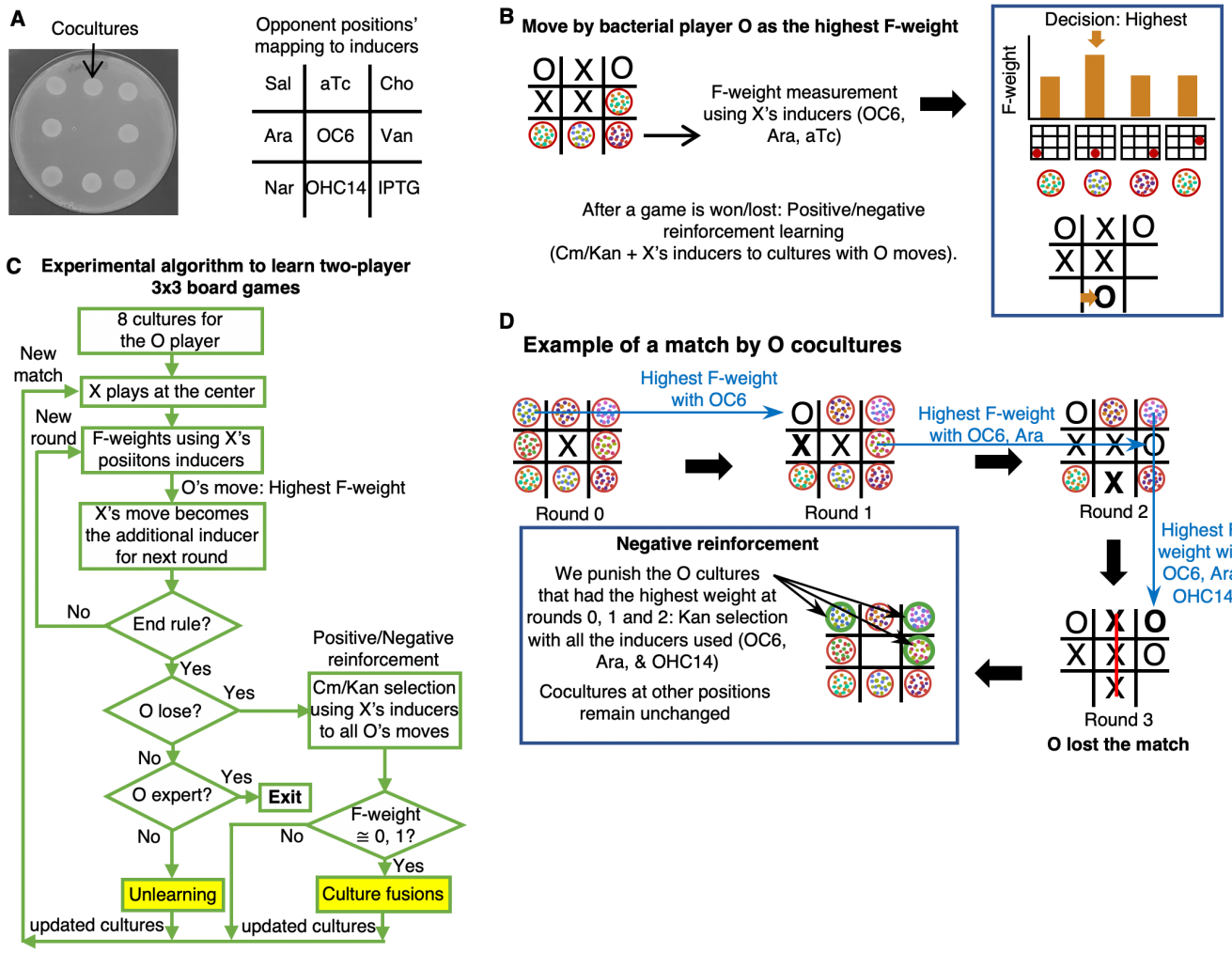

Fig. 3. A general wet-algorithm, implementing positive and negative reinforcement learning, allowing memregulon cocultures to learn two-player 3x3 board games. (**A**) Left: Memregulon coculture arrangement in a plate for a bacterial O player. We assign cocultures at each board position except the center. Right: Mapping to inducers of the opponent positions. (**B**) A move is determined by the position of the coculture having the highest measured F-weight (Fig. 1C), when incubating the cocultures with the inducers associated to the X's moves. After the move, the parental plates are interrogated again with the new X's move. This is repeated for several rounds until the game ends. If the player O wins/loses, we apply a chloramphenicol/kanamycin selection, adding the inducers of X's positions only to the O cultures involved in the played match. (**C**) General experimental algorithm for training cocultures to learn how to play two-player 3x3 board games. The steps highlighted in yellow are not needed for tic-tac-toe. (**D**) Example of a tic-tac-toe match showing the role played by the inducers in communicating the X's positions to the O cocultures and the selective punishment of the cocultures deciding a move in the match. X plays first at the center position and we induce all the 8 cocultures with the inducer mapped to the the X's position (OC6), to compute the highest F-weight. Player O moves at the position corresponding to the coculture with the highest F-weight, here assumed to be at the top-left corner. In the following rounds, the cocultures at unoccupied positions are successively induced with all the inducers corresponding to X's positions, until the game finishes with X winning. Afterwards, we apply a negative reinforcement operation to only the cocultures at the positions that O played. This is done by a kanamycin selection in presence of the inducers of X's moves at all rounds before winning (OC6, Ara, OHC14).

Figure 4

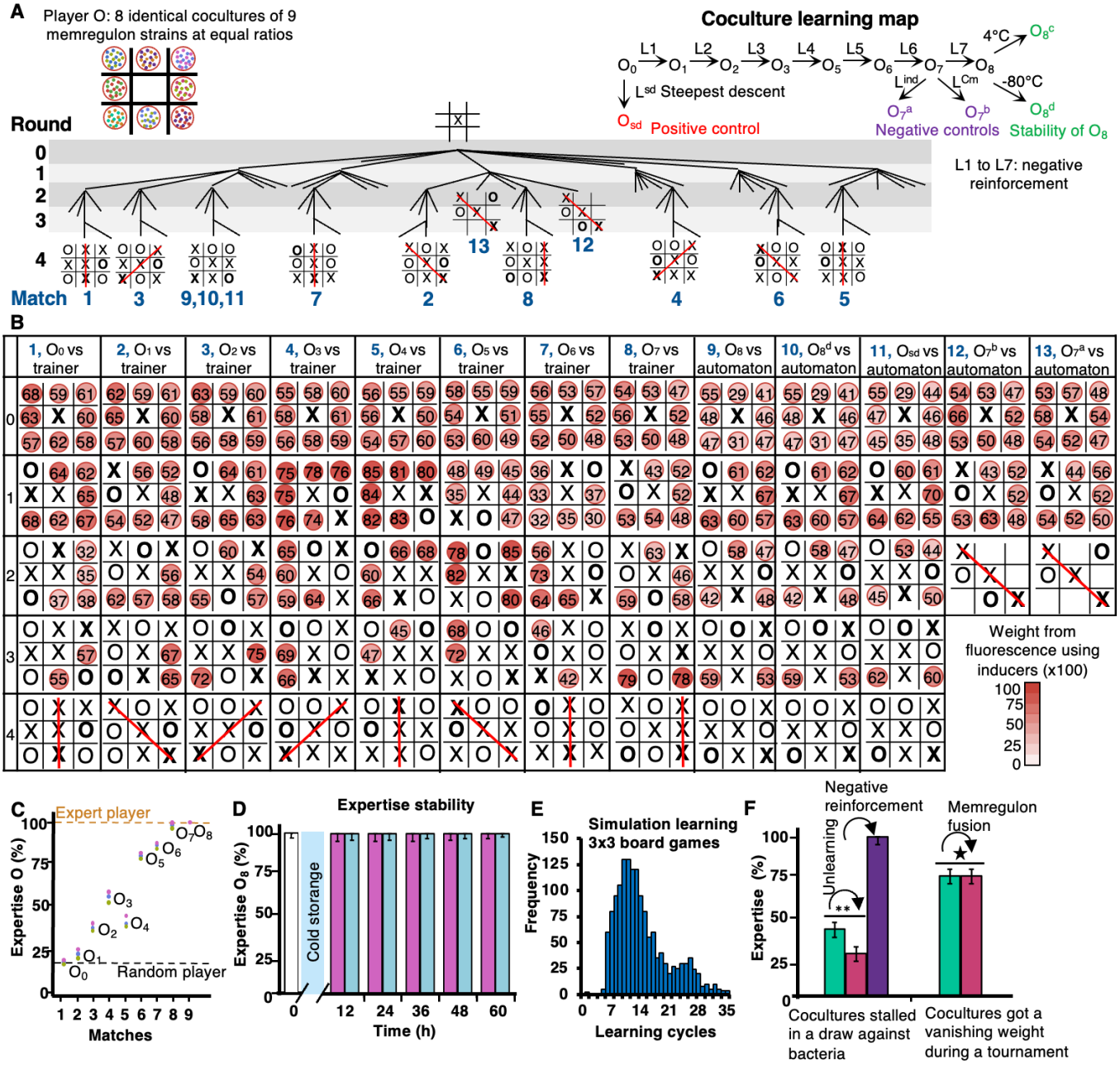

Fig. 4. Uniform memregulon cocultures with random expertise learn to master the tic-tac-toe game. Cocultures (player O) play a tournament against a trainer (player X) designed (see suppl. Table S1) to always try to win. (**A**) Top left: cocultures are initially setup (O_0) at each of the 8 positions by mixing (at equal volumetric ratios) memregulon strains for each of the 9 inducible promoters, and in three biological replicates. Top right: The cocultures lose at every match, and a negative reinforcement learning (L1 to L7) is applied to them, creating in succession the cocultures O_1 to O_8, until O_8 reaches 100% expertise. We use as negative controls a kanamycin selection with swapped inducers (L_{ind}) or swapped antibiotic (L_{Cm}) to the cocultures O_7, to create O_7^a and O_7^b respectively. As a positive control, we implement a weight training experimentally mimicking steepest descent (L^{sd}), where the cocultures O_{sd} have learned in one step without using antibiotic selection. (**B**) Detail of the F-weights of the matches played, shown inside red-colored circles (multiplied by 100). We challenged the obtained cocultures (O_8, O_8^d, O_{sd}, O_7^a and O_7^b) to play against an expert player automaton, verifying that their matches ended in draws except for the cocultures from the negative controls. (**C**) We computed a coculture's expertise (defined as percentage of wins and draws after playing any possible match) by a computer simulation using the measured F-weights. We use a different color for each of the 3 biological replicates**.** Dashed lines indicate the random and expert players. (**D**) The mastery (100% expertise) of player O is stable in time, even after cold storage of the plates for 4 days. (**E**) Computer simulation of the O_0 cocultures playing negative reinforcement learning tournaments in 1,500 random games at 3x3 boards, achieving mastery in 98% of cases shown as a distribution of the length of the learning cycles. (**F**) Experimental testing of the operations highlighted in yellow in Fig. 3C, allowing extending the reinforcement learning algorithm to arbitrary games. An *unlearning operation* (see main text), based on a reinforcement learning with swapped antibiotic (chloramphenicol instead of kanamycin), avoids getting stuck at suboptimal (expertise < 100%) draws, at the cost of decreasing the expertise. Expertise was then dramatically increased after a subsequent negative reinforcement learning (double asterisk, $p<0.01$). A *memregulon fusion* (see main text) maintains the expertise (star, $p < 0.01$) and allows for learning to continue without the risk of a weight reaching a value of 0 or 1 (which makes them unable to learn anymore). Error bars indicate SD from n = 3 biological population replicates, obtained on 3 different days.

References

1. P. Mohammadi, N. Beerenwinkel, Y. Benenson, Automated Design of Synthetic Cell Classifier Circuits Using a Two-Step Optimization Strategy. *Cell Syst* **4**, 207-218.e214 (2017).
2. A. Didovyk *et al.*, Distributed Classifier Based on Genetically Engineered Bacterial Cell Cultures. *ACS Synthetic Biology* **4**, 72-82 (2015).
3. C. T. Fernando *et al.*, Molecular circuits for associative learning in single-celled organisms. *Journal of The Royal Society Interface* **6**, 463-469 (2009).
4. J. Macia, B. Vidiella, R. V. Sole, Synthetic associative learning in engineered multicellular consortia. *Journal of The Royal Society Interface* **14**, 20170158 (2017).
5. A. E. Friedland *et al.*, Synthetic Gene Networks That Count. *Science* **324**, 1199-1202 (2009).
6. L. B. Andrews, A. A. K. Nielsen, C. A. Voigt, Cellular checkpoint control using programmable sequential logic. *Science* **361**, eaap8987 (2018).
7. R. Zhu, J. M. del Rio-Salgado, J. Garcia-Ojalvo, M. B. Elowitz, Synthetic multistability in mammalian cells. *Science* **375**, eabg9765 (2022).
8. L. Qian, E. Winfree, J. Bruck, Neural network computation with DNA strand displacement cascades. *Nature* **475**, 368-372 (2011).
9. X. Lin *et al.*, All-optical machine learning using diffractive deep neural networks. *Science* **361**, 1004-1008 (2018).
10. P. Banda, C. Teuscher, D. Stefanovic, Training an asymmetric signal perceptron through reinforcement in an artificial chemistry. *Journal of The Royal Society Interface* **11**, 20131100 (2014).
11. R. Pei, E. Matamoros, M. Liu, D. Stefanovic, M. N. Stojanovic, Training a molecular automaton to play a game. *Nature nanotechnology* **5**, 773-777 (2010).
12. A. Pandi *et al.*, Metabolic perceptrons for neural computing in biological systems. *Nature Communications* **10**, 3880 (2019).
13. X. Li *et al.*, Synthetic neural-like computing in microbial consortia for pattern recognition. *Nature Communications* **12**, 3139 (2021).
14. K. Sarkar, D. Bonnerjee, R. Srivastava, S. Bagh, A single layer artificial neural network type architecture with molecular engineered bacteria for reversible and irreversible computing. *Chemical Science* **12**, 15821-15832 (2021).
15. L. Yang *et al.*, Permanent genetic memory with >1 byte capacity. *Nature methods* **11**, 1261-1266 (2014).
16. M. Isalan *et al.*, Evolvability and hierarchy in rewired bacterial gene networks. *Nature* **452**, 840-845 (2008).
17. J. Carrera, S. F. Elena, A. Jaramillo, Computational design of genomic transcriptional networks with adaptation to varying environments. *Proceedings of the National Academy of Sciences of the United States of America* **109**, 15277-15282 (2012).
18. J. Schrittwieser *et al.*, Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature* **588**, 604-609 (2020).
19. A. J. Meyer, T. H. Segall-Shapiro, E. Glassey, J. Zhang, C. A. Voigt, Escherichia coli "Marionette" strains with 12 highly optimized small-molecule sensors. *Nature chemical biology* **15**, 196-204 (2019).

20. W. Tang, D. R. Liu, Rewritable multi-event analog recording in bacterial and mammalian cells. *Science* **360**, eaap8992 (2018).
21. N. Cermak, M. S. Datta, A. Conwill, Rapid, Inexpensive Measurement of Synthetic Bacterial Community Composition by Sanger Sequencing of Amplicon Mixtures. *iScience* **23**, 100915 (2020).
22. L. Chua, Memristor-The missing circuit element. *IEEE Transactions on Circuit Theory* **18**, 507-519 (1971).
23. D. Krotov, J. J. Hopfield, Unsupervised learning by competing hidden units. *Proceedings of the National Academy of Sciences* **116**, 7723-7731 (2019).
24. S. Regot *et al.*, Distributed biological computation with multicellular engineered networks. *Nature* **469**, 207-211 (2011).
25. A. Tamsir, J. J. Tabor, C. A. Voigt, Robust multicellular computing using genetically encoded NOR gates and chemical 'wires'. *Nature* **469**, 212-215 (2011).
26. M. N. Stojanovic, D. Stefanovic, A deoxyribozyme-based molecular automaton. *Nature biotechnology* **21**, 1069-1074 (2003).
27. W. Maass, On the computational power of winner-take-all. *Neural Comput* **12**, 2519-2535 (2000).
28. A. E. Elo, *The rating of chessplayers, past and present*. (BT Batsford Limited, 1978).
29. Y. Cao, J. Yang, in *2015 IEEE Symposium on Security and Privacy (SP)*. (IEEE, 2015), pp. 463-480.
30. D. O. Hebb, *The organization of behavior: a neuropsychological theory*. (Science editions, 1949).
31. C. Koch, *Biophysics of computation: information processing in single neurons*. (Oxford university press, 2004).
32. N. Frémaux, W. Gerstner, Neuromodulated Spike-Timing-Dependent Plasticity, and Theory of Three-Factor Learning Rules. *Frontiers in Neural Circuits* **9**, (2016).
33. C. A. Voigt, S. Kauffman, Z. G. Wang, Rational evolutionary design: the theory of in vitro protein evolution. *Adv. Protein Chem.* **55**, 79-160 (2000).
34. W. Kong, D. R. Meldgin, J. J. Collins, T. Lu, Designing microbial consortia with defined social interactions. *Nature chemical biology* **14**, 821-829 (2018).
35. C. C. Guet, M. B. Elowitz, W. Hsing, S. Leibler, Combinatorial synthesis of genetic networks. *Science* **296**, 1466-1470 (2002).
36. K. M. Cherry, L. Qian, Scaling up molecular pattern recognition with DNA-based winner-take-all neural networks. *Nature* **559**, 370-376 (2018).
37. M. Ibba, P. Kast, H. Hennecke, Substrate specificity is determined by amino acid binding pocket size in Escherichia coli phenylalanyl-tRNA synthetase. *Biochemistry* **33**, 7107-7112 (1994).
38. L. S. Qi *et al.*, Repurposing CRISPR as an RNA-guided platform for sequence-specific control of gene expression. *Cell* **152**, 1173-1183 (2013).
39. Y. Li *et al.*, In vitro evolution of enhanced RNA replicons for immunotherapy. *Scientific Reports* **9**, 6932 (2019).
40. A. Jaramillo, Engineered stable ecosystems. *Nat Microbiol* **2**, 17119 (2017).
41. R. Mathis, M. Ackermann, Response of single bacterial cells to stress gives rise to complex history dependence at the population level. *Proceedings of the National Academy of Sciences of the United States of America* **113**, 4224-4229 (2016).
42. J. Rodriguez-Beltran *et al.*, Multicopy plasmids allow bacteria to escape from fitness trade-offs during evolutionary innovation. *Nat Ecol Evol* **2**, 873-881 (2018).

43. C. Hildenbrand, T. Stock, C. Lange, M. Rother, J. Soppa, Genome Copy Numbers and Gene Conversion in Methanogenic Archaea. *Journal of Bacteriology* **193**, 734-743 (2011).
44. J. Aryaman, I. G. Johnston, N. S. Jones, Mitochondrial Heterogeneity. *Frontiers in Genetics* **9**, 718 (2019).