Peer Review File

Manuscript Title: Nonlinear decision-making with enzymatic neural networks

Reviewer Comments & Author Rebuttals

Reviewer Reports on the Initial Version:

Referees' comments:

Referee #1 (Remarks to the Author):

In this manuscript, the authors developed neuromorphic enzymatic networks that could perform nonlinear decision making. Compared to DNA-only networks, this design with DNA-processing enzymes allows the implementation of low-leakage, high-sensitivity and nonlinear molecular perceptron. Using the enzymatic reaction networks, the authors demonstrated linear classification with sharp decision margins. Composed of two linear classifiers, a molecular perceptron with a hidden layer and an output layer to perform non-linear classification was realized for the first time. The concentration-scanning microfluidics offers massively parallel test for DNA-based computing in small volumes, which appears to be very attractive for simple preparation of DNA circuits and panoramic presentation of the computing performance. The design and the display of this work are very impressive, which make it suitable to be published in Nature with appropriate revisions.
1) My main concern lies in the clarification of the novelty of the work. As described by the authors, one main advantage of enzymatic reaction over DNA-only reactions is the low leak, which allows cascading of neurons to build deep networks. However, the drain-based activation strategy was proposed and demonstrated in a previous study, Nat. Commu. 7, 13474 (2016). Encapsulating reaction networks in droplets and high-parallel displaying massive input combinations has also been shown in Nat. Chem. 8, 760–767 (2016). Both papers were cited but they should better show the novelty over these published works.
2) The deep network that the authors showed only contained two layers with three neurons and the realized function is logic NOR, which is not attractive enough as the function is very easy to realize with DNA-based logic gates. The output layer was achieved by polymerase induced strand displacement, which is very different from the hidden layer. Lacking weighted summation and autocatalytic activation, this layer performed similar to a reporting device rather than a neuron.
The followings are a list of relatively minor issues that the authors may want to address:
1) The fitting algorithm for separatrix should be more clearly described. And was the fitted line deviated from the desired separatrix? Any calibrations needed?
2) The linearity and decision margin for negative-weighted classifiers are worse than the positive ones. The authors should include a more detailed discussion about this.
3) It is unfair to compare the Hill coefficients with Lopez's work as they used RNA as input, not "DNA-only networks". Maybe a comparison with Qian's work (Nature 559, 370, 2018) is more reasonable?
4) In the majority voting part, the author included a comparison of their neuromorphic network to DNA logic circuits. Actually, the advantages of neural networks over logic circuits have been discussed previously (Nature 559, 370, 2018). As the novelty of this manuscript lies in the use of enzymatic reactions, discussions on realizing the multi-bit majority voting with enzymatic networks as compared to the DNA-only networks may provide better insights.

Referee #2 (Remarks to the Author):

The authors present a linear classifier, a classifier using decision tree and a majority logic with ten inputs using DNA and enzymes. Their functionality is validated by experimental results at temperatures ranging from 40-49 degrees Celsius. Comments on the paper are listed below.

1. In the Abstract and P. 2, the authors state "molecular classifications equivalent to a multilayer neural network (i.e. nonlinear partitioning of a concentration space) have been elusive." Recent papers have demonstrated feasibility of perceptrons and artificial neural networks using DNA including activation functions such as sigmoid, ReLU and softmax. See for example,

https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9028230

http://www.nature.com/articles/s41598-018-26709-6

2. The claim that the paper has demonstrated "deep neuromorphic" in the title and "equivalent to a multilayer perceptron in ~25,000 cell-sized droplets" are partially true. The networks may not be considered deep. Equivalence with multi-layer perceptron may be overstated. Multilayer functions can synthesize many functions all of whom may not be realizable by the proposed experimental framework.

3. In Fig. 1, it appears that the input is regenerated. Is the input not consumed?

4. For Extended Data Fig. 4b, what condition can guarantee the crossover between autocatalytic production and removal?
5. The scatter plots showing decision boundaries are good. It would be useful for readers to see the classification accuracy including sensitivity and specificity in tabular form and/or a confusion matrix form.

6. The data generated by the experiment could be fit into a multilayer perceptron to see the difference between the multilayer perceptron and the nonlinear enzymatic network.

7. There is sufficient variation when the temperature is varied. Some form of robustness or error analysis with respect to experimental parameters such as input concentration and temperature etc. may be useful.

Referee #3 (Remarks to the Author):

A majority voting algorithm and a 3-layer perceptron were implemented using the so-called PEN TOOLBOX. They have succeeded in visualizing the characteristics of the obtained circuit by making a large number of droplets using a microfluidic device and comprehensively investigating the whole concentration space of the input chemical substances.

Although some studies on neural networks by DNA computing have been conducted so far, it has been difficult to construct a multi-layered circuit since the concentration can only be a positive value. By adopting PEN, the logic can be greatly simplified, and it is highly evaluated that the three-layer perceptron could be implemented by chemical reaction.

Comprehensive characterization using the droplet system has made it possible to completely visualize the characteristics of the constructed system, which is a result supported by extremely reliable experimental technology.

Some suggestions:
The expression "deep neuromorphic network" in the title is generally considered to refer to a large scale multi-layered neural network. I think it is an exaggeration to call a perceptron with 2 inputs, 3 layers, and 1 output a "deep" neural network.

The expression "demonstrating computations equivalent to multilayer perceptron in ~ 25000 cell-

sized droplets” at the end of the abstract is misleading.
This expression is misunderstood as if many droplets form a network. In reality, each droplet contains a complete network.

For me, the significance of majority voting with veto is not clear.
Is the correct answer rate 100% when the veto right is given to other than X3?

In Materials section, please explain why you used a special exonuclease and what makes it different from commercial one.

Author Rebuttals to Initial Comments:

Table of Contents

Referee #1 (Remarks to the Author):

In this manuscript, the authors developed neuromorphic enzymatic networks that could perform nonlinear decision making. Compared to DNA-only networks, this design with DNA-processing enzymes allows the implementation of low-leakage, high-sensitivity and nonlinear molecular perceptron. Using the enzymatic reaction networks, the authors demonstrated linear classification with sharp decision margins. Composed of two linear classifiers, a molecular perceptron with a hidden layer and an output layer to perform non-linear classification was realized for the first time. The concentration-scanning microfluidics offers massively parallel test for DNA-based computing in small volumes, which appears to be very attractive for simple preparation of DNA circuits and panoramic presentation of the computing performance. The design and the display of this work are very impressive, which make it suitable to be published in Nature with appropriate revisions.

We are immensely grateful to the reviewers for their time on the manuscript and their expert reading. We used their comments to revise and vastly improve the manuscript, and we hope that they will be satisfied with the revised version.

To summarize at a high level:

- We have clarified, disambiguated and expanded various points, stressing why our paper is novel and original. We added references to better explain background research on molecular neuromorphic computation and possible future developments.
- We discuss what it means for a network to be deep. We added discussion about two possible applications of our networks (DNA storage and molecular diagnosis of cancer) to show the depths that would be needed for practical computations
- We revamped figures, making them clearer, more visually appealing and more streamlined. We now make hybrid raw/smooth plots that retain their experimental feeling while being easier to read and saving space.
- We performed new analysis to back our claims - confirming the excellent performance of our networks over the state-of-the-art, and demonstrating that our network cannot be satisfyingly captured by a single-layer perceptron.
- We vastly extended the experimental demonstration to broaden the appeal of the paper, implementing a fully tunable multilayer perceptron that computed a boxcar function. We also extended majority voting and implemented a new functionality: majority right

1) My main concern lies in the clarification of the novelty of the work. As described by the authors, one main advantage of enzymatic reactions over DNA-only reactions is the low leak, which allows cascading of neurons to build deep networks. However, the drain-based activation strategy was proposed and demonstrated in a previous study, Nat. Commu. 7, 13474 (2016). Encapsulating reaction networks in droplets and high-parallel displaying massive input combinations has also been shown in Nat. Chem. 8, 760–767 (2016). Both papers were cited but they should better show the novelty over these published works.

We thank the reviewers for their comments and their thorough reading of the manuscript.

To summarize: (details are given below)

- **We highlight the new methodological developments needed for this paper: the thermal gradient platform, the silicon chambers, the strategies for positive and negative weighting, the NOR gate, and the protocol to equalize enzymatic activities for majority voting**
- **We emphasize the novelty of our networks for linear classification over the state-of-the-art papers of Qian and Seelig (e.g. we discuss the scalability for majority voting of enzymatic networks and DNA-only network)**
- **We reinforce the main novelty of the paper - nonlinear classification - with a new set of experiments where we synthesize rectangular functions with multilayer perceptrons**

To address this comment, we would like first to highlight the novelty over our own published work, and then show the novelty over other published works. Briefly put, the novelty of our paper lies more in what we have achieved (classification of nonlinearly separable regions and majority voting on 10 bits) than the tools we have used to do that (enzymatic networks and microfluidics). The two papers that are cited by the reviewer are means to an end: they reported a chemical primitive (the drain strategy) and a methodology (the droplet platform) on which we build to demonstrate neuromorphic computation, but they did not themselves demonstrate any neuromorphic computation, nor did they demonstrate any advanced logic computation like majority voting for that matter.

We started this project of neuromorphic computation soon after publishing those two papers, 5 years ago. In the meantime, it took three students and two postdocs to integrate these technologies, debug inevitable problems and optimize the network and eventually demonstrate neuromorphic computations.

Although the system works seamlessly now, it was not trivial at first. There was no experimental guarantee in 2016, when we started the project, that our enzymatic system could emulate neuromorphic computations. There has been a persistent reality gap in DNA computing between what we ought to be able to do, and what we are actually able to do. Consider for instance the universal approximation theorem of Soloveichik et al.[1] (a landmark paper in the field). In principle DNA strand displacement could emulate arbitrary chemical reaction networks, and the authors simulated two DNA-only oscillators that oscillated perfectly

in their paper. Yet turning these simulations into experimental traces has proved much harder than expected[2]: it took five years of work to a talented graduate student (N. Srinivas) to get a DNA-only system to oscillate, and the quality of the experimental oscillations was nowhere near the simulated traces. Soloveichik recently acknowledged this gap in one of his recent paper[3]:

> *A programmable chemical process called DNA strand displacement can in principle (and, to some extent, experimentally) implement arbitrary, rationally designed CRNs*

We overcame the reality gap and emulated neuromorphic computations thanks to the strong foundation of the two papers cited by the reviewer. We *used* the droplet platform to map a continuous input space and expedite the design of the networks. Without microfluidics, we would never have pinpointed the optimal concentrations so quickly - nor would we have gained the kind of broad insights that come from *actually* seeing the bifurcation diagram of an enzymatic system (as opposed to guessing it). But that does not mean we did not have to improve the platform. At some point in the project, we needed to optimize the temperature - a parameter crucial for any enzymatic reaction. So we designed from scratch silicon chambers, a material with excellent thermal and mechanical properties. And we built a new temperature module to incubate droplets in a controlled thermal gradient and scan intervals of temperatures. This understanding of thermal effects helped us to optimize the working of the network.

We *used* the drain-based activation strategy as a computational primitive for our chemical neurons. Without the drain, we would never have built chemical neurons that are so robust and nonlinear. We knew the drain was robust, but we had not deployed it so far in molecular computing, and had no clear idea of its nonlinearity when connected to a converter template - a parameter crucial for making a chemical neuron. So we measured the Hill coefficient of the neuron with the droplet platform, and found that it performed better than non-enzymatic strategies.

We also had to integrate more templates than we had ever handled (10 templates for majority voting). To that end we developed a protocol to equalize enzymatic activities and guarantee the equibalance of votes (Extended Data Figure 7). We also developed a strategy for negative weights and a NOR gate to connect neural and logical computations. Although these methodological developments were essential, we did not delve on them because we did not see them as central to the paper. Again, they were a means to an end: demonstrating nonlinear classifications with neuromorphic molecular networks and majority voting on many bits.

In line with the comment from the reviewer, we now better highlight these technical advances in the paper. For instance, we now explicitly mention the development of the Silicon chambers and the thermal platform in the section on the linear classifier:

We then moved to cell-sized compartments and used concentration-scanning microfluidics[46,47] to prepare tens of thousands of ~55 µm droplets containing the same linear classifier, but with inputs spanning the concentration plane. We immobilized a monolayer of droplets in a silicon chamber - a material with excellent optical, thermal and mechanical properties - and incubated the chamber in a thermal platform (Extended Data Fig. 2). Overall, this microfluidic setup offers

a superb control of concentrations and temperatures, and enables a precise visualization of the decision boundary (Fig. 2 c,d).

We also mention that we developed a protocol for equalizing the enzymatic activities of the templates in the majority voting, which is required to ensure the equibalance of the inputs (section on Majority voting):

We thus developed an *ad-hoc* protocol that equalizes the production rate of signal strand for 10 converter templates, in order to ensure the equibalance of each vote (Extended Data Fig. 7).

The protocol was already present in Extended Data, but we had referred to it only in the figure, not in the text.

Below we also discuss novelty with respect to two pioneering works on molecular neuromorphic computation: the papers of Qian and Seelig.

We feel that DNA computing has matured to the point that the community is now combining the vast troves of mechanisms published in the past decades to make new molecular programs - not unlike the switch from electrical engineering (the design of electronic circuits at low level) to software engineering (the design of programs at a high level). Advances in the writing, reading and handling of DNA make it possible to build large scale molecular programs with DNA and put them to do useful work for other scientific communities, like those of molecular diagnosis or data storage. So the next breakthroughs in DNA nanotechnology will not necessarily come from new mechanisms or tricks at a low level, but rather from the ingenious combination of existing technologies at a high level.

The pioneering work of Cherry and Qian in 2018 is a case in point[4]. It was not pioneering because it reported a new mechanism: after all, at a low-level their DNA classifier worked with seesaw gates (reported almost a decade earlier in two papers[5,6]), and with the cooperative hybridization of Zhang, also reported in 2011[7]. The importance of the 2018 paper comes from the integration of seemingly disparate technologies and ideas: of course, the seesaw gates and cooperative hybridization, but also the architecture of competitive molecular networks[8], as well as the use of a commercial acoustic liquid handler to manage the large number of DNA strands that come with encoding images. None of these ideas in itself is radically new, but their combination is - making it one of the best papers in DNA computing of the decade. This paper has been inspirational, and encouraged us to push the limit of our enzymatic network even further.

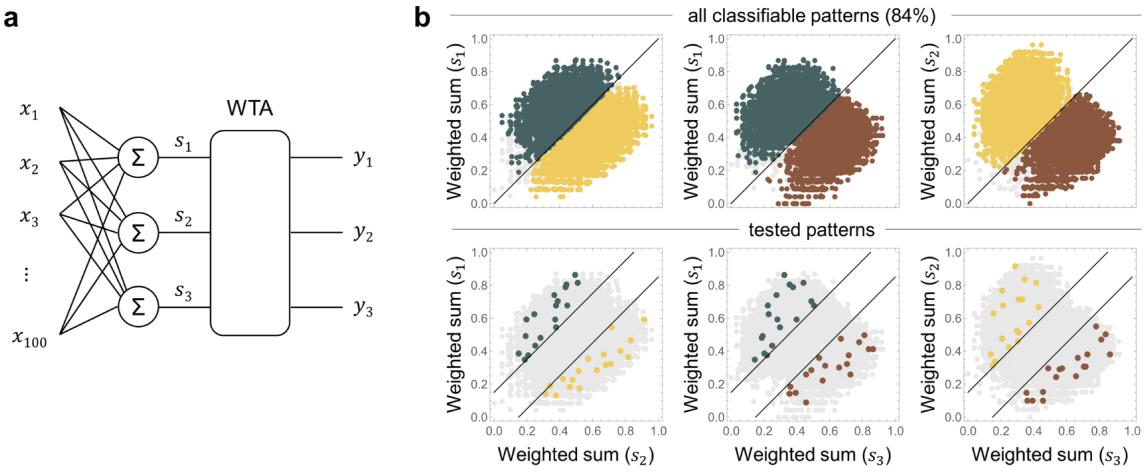
The 2018 paper of Seelig was also important to us because they demonstrated linear classification on natural RNA[9]. They did introduce a new mechanism in their paper, but it was for the linear part of the classifier (the weighted summation of concentrations of mRNA), and the nonlinear part also borrowed the cooperative hybridization of Zhang from 2011. Yet the paper of Seelig was seminal because they set out to work with natural sequences (like us but unlike Qian's paper), and reported new ideas to deal with the secondary structures of mRNA (thermal or chemical annealing).

These nonenzymatic networks of Qian and Seelig were a constant yardstick for us. They defined the state-of-the-art and set the bar for other papers. And we believe we have overcome it with our enzymatic networks. First, we demonstrate a classifier that is many times more sensitive and nonlinear (in our new set of experiments, we now demonstrate nonlinear classification with RNA inputs at 10~50 pM, which is about three orders of magnitude more sensitive than Seelig). Secondly, we used the sharpness of our classifier to compute the majority with veto on 10 bits - while it is unlikely that nonenzymatic networks could compute the majority of more than 3 bits because of their limited decision margin (See SI discussion on Majority Voting).

Thirdly, and most importantly, we demonstrated computations equivalent to a multilayer perceptron, that is a nonlinear classification. We partitioned three non-linearly separable and continuous regions of the concentration plane. In addition, we have now performed a new class of experiments where we wire 3 full chemical neurons to compute a parametric rectangular function on the concentration of a RNA input - further backing our claim of equivalence to a multilayer perceptron. **Such nonlinear classification is a significant achievement over the literature**.

The main achievement of the paper of Cherry and Qian was to propose a strategy that has the potential to recognize up to 20% of MNIST, a gold standard in machine learning. Doing molecular computations on a standard dataset of machine learning is a tour-de-force that should be lauded. Yet it may have escaped the notice of many readers that the paper of Cherry and Qian[4] *did not* claim nonlinear classification, that is a classification equivalent to that done by a multilayer perceptron. Due to the limited decision margin of their classifier, they chose sets of examples that were separated by a comfortable decision margin (see original figure below). But at the end, **all the examples of fully molecular classification on MNIST presented by Cherry *et al.* (Fig. 3, Extended Data Fig. 5, Extended Data Fig. 7 of their paper[4]) were linearly separable, i.e., they could have been classified by a single-layer perceptron.**

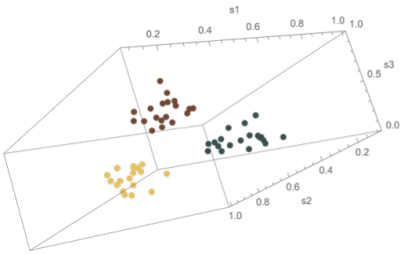
Take for instance Extended Data Fig. 7 in their paper:



[Extract of Cherry, Nature 2018] Extended Data Fig. 7 A winner-take-all DNA neural network that recognizes 100-bit patterns as one of three handwritten digits. a, Circuit diagram. b, Choosing the test input patterns on the basis of their locations in the weighted-sum space. c, Overlap between the three memories: '2', '3' and '4'. d,

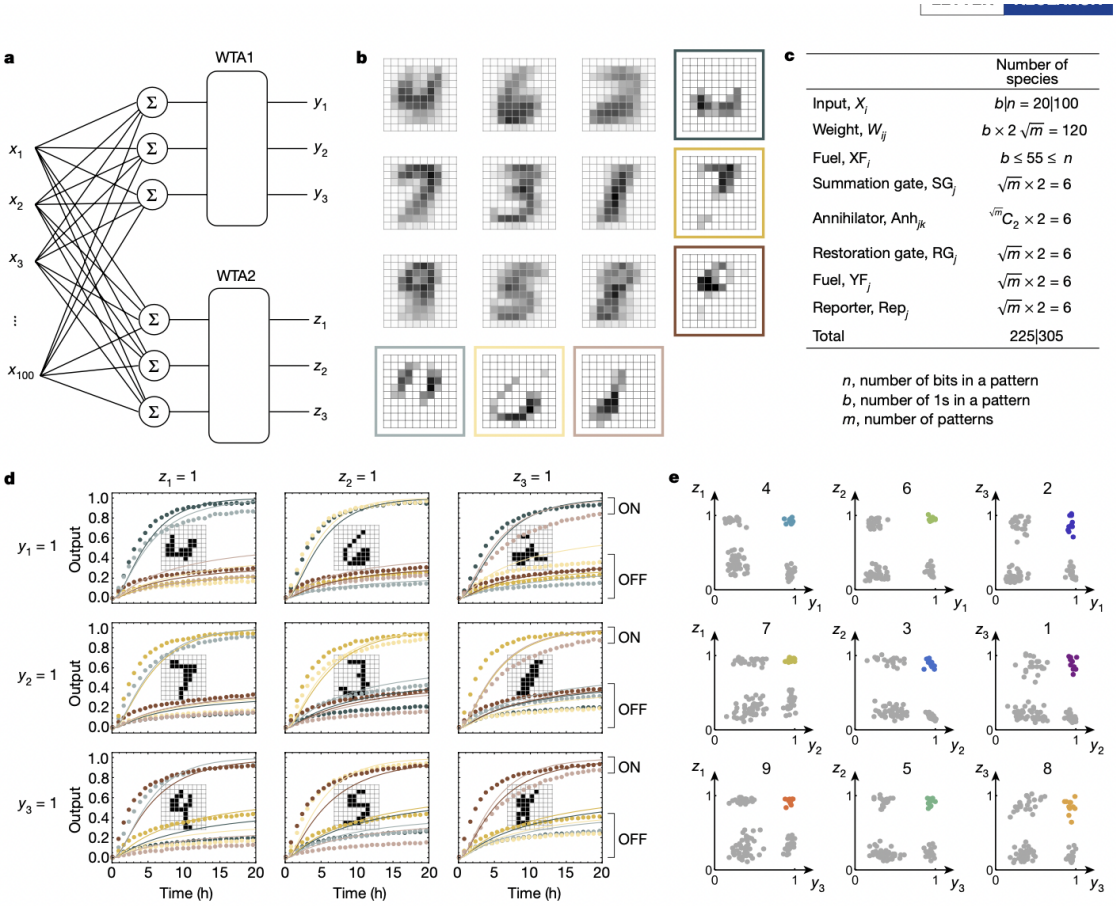
Recognizing handwritten digits with up to 28 flipped bits compared to the ‘remembered’ digits. Dotted lines indicate fluorescence kinetics data and solid lines indicate simulation. The standard concentration is 100 nM. Initial concentrations of all species are listed in Extended Data Fig. 10. The input pattern is shown in each plot. Note that 40 is the maximum number of flipped bits because all patterns have exactly 20 1s.

It may not be obvious at first, but the set of chosen digits are linearly separable. It becomes evident if we project the images on the 3D space of weighted sum (s1,s2,s3):



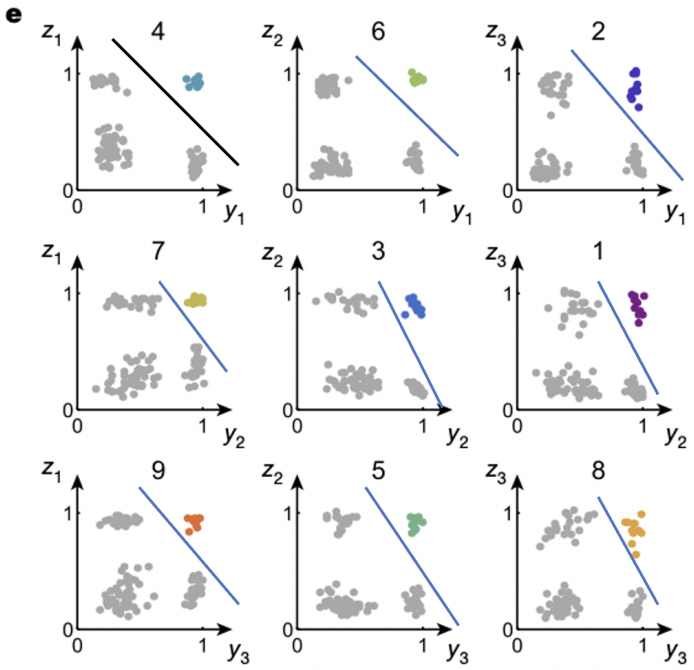
(Here each point is an image, the colour indicating the digit, data are taken from the supplementary materials of the paper). The images are clearly linearly separable from this 3D visualization, and this is confirmed by fitting a linear classifier on the data. (Again it is not surprising that a set of 3 digits can linearly classified, because most of MNIST can be linearly classified[10]

The last classification of Cherry *et al.*[4] (Fig. 4) is not fully molecular, because it lacks a final logical layer (a lookup table) that converts the code (y1,y2,y3,z1,z2,z3) into one category of digits among nine (for instance if y1 AND z1 are present, then output the signal for digit 4). As a matter of fact, two layers of computation are not even needed because the selected sets in Fig. 4 are also linearly separable and could have been fully classified by a single-layer perceptron.



[Extract of Cherry, Nature 2018] Figure 4 A winner-take-all DNA neural network that recognizes 100-bit patterns as one of nine handwritten digits. **a,** Circuit diagram for recognizing nine distinct patterns using a grouping approach. WTA1 and WTA2 are two separate winner-take-all functions that each yields a distinct set of outputs (yj and zj). b, Weights determined using an 'average then subtract' method. The average of 100 example digits from the MNIST database are shown grouped by rows (corresponding to outputs yj) and columns (corresponding to outputs zj). The weight matrix for each group (boxed patterns; colours correspond to the respective output trajectories in **d)** is the average of all in-group digits less the average of all out-of-group digits. Using this weight matrix, the fraction of experimentally feasible test patterns from all examples in the MNIST database was calculated for all possible ways of grouping the nine digits and the best grouping was chosen and shown here. **c,** Number of distinct species in the circuit in **a.** For the total number of species, the two values correspond to the number of species for a specific number b of inputs (left) and for all n possible inputs (right). **d,** Fluorescence kinetics data (dotted lines) and simulations (solid lines) of the circuit behaviour with nine representative input patterns (shown in the plots). **e,** Fluorescence level of each pair of outputs at 24 h or longer after the inputs were added, collected from 99 experiments with 11 example patterns per digit. Each coloured point corresponds to an example pattern from the labelled class of digit; each grey point corresponds to an out-of-class example pattern. Weights and inputs are listed in Supplementary Table 2. The initial concentrations of all species are listed in Extended Data Fig. 10 (details in Supplementary Table 3).

Indeed if we look at the projection in panel Fig.,4e, it becomes obvious that the images are linearly separable. For clarity, we have drawn possible lines to separate each target digit from the other images. (Note that this one possible classification using just two weighted sums as coordinates, other linear classifications with thicker decision margin are likely possible in higher dimension). This is a visual demonstration, but we have verified computationally that the digits are indeed linearly separable):



Lastly, it should be noted the classifier of Cherry and Qian cannot handle analog inputs, as acknowledged at the end of their paper[4]:

> *Using a variable-gain amplifier, winner-take-all DNA circuits could be adapted to process analogue inputs, which would enable a wider range of signal-classification tasks, including applications in detecting complex disease profiles that consist of mRNA and microRNA signals.*

In other words, the work of Qian et al. is a milestone that classified *linearly* separable sets of *digital* inputs (images). In our work, we classified **nonlinearly** separable sets of **analog** inputs. In addition, we now perform nonlinear classification on natural RNA - like Seelig's paper - to motivate the biological appeal of our work (new set of experiments on the rectangular function).

We now better summarize these remarks in the introduction:

Cherry and Qian reported in a tour de force a classifier with the potential to recognize up to 20% of the handwritten digits of the MNIST database[8]. Together, these molecular classifiers showcased the benefits of neuromorphic networks over Boolean circuits: massive parallelism, handling of analog inputs, and tolerance to corrupted patterns. However, these nonenzymatic classifiers had limited decision margins, *i.e.* they could not discriminate between two similar inputs belonging to different classes. They also suffered from leaks that made the composition of layers delicate. Overall, fully molecular classification was only demonstrated on datasets that could be linearly separated by a margin wide enough to accommodate the thick decision margin of the DNA classifiers.

2) **The deep network that the authors showed only contained two layers with three neurons** and **the realized function is logic NOR,** which is not attractive enough as the function is very easy to realize with DNA-based logic gates. The output layer was achieved by polymerase induced strand displacement, w**hich is very different from the hidden layer.** Lacking weighted summation and autocatalytic activation, this layer performed similar to a reporting device rather than a neuron.

Summary: We thank the reviewer for the insightful comment. In response to this comment, we have performed a new set of experiments: the output layer is now achieved with exactly the same chemistry as the hidden layer, i.e., fully adjustable molecular neurons.

The deep network that the authors showed only contained two layers with three neurons

We thank the reviewer for this comment and agree that the term "deep" could be ambivalent for non-specialist readers. Since this point was raised several times in the review, we made a detailed and common answer which is given in the responses to reviewer 2 (comment 2), which we encourage the reviewer to read. For convenience, a short summary is below.

Our networks are deep based on the mathematical definition of depth, for instance the one given by Lecun, Bengio and Hilton in Nature[11] in 2015 ("A deep-learning architecture is a multilayer stack of simple modules"). But the practical question is how deep do the networks need to be? It depends on the applications. If we look for instance at storage of multimedia content in DNA[12], then 10-20 layers are required for advanced image recognition. We are still behind by a factor of 10, but we are hopeful that the gap could be filled - at least partially - within ten years thanks to advances in DNA synthesis, sequencing and microfluidics. If we focus on molecular diagnosis, our networks are already deep enough. We now give examples in the discussion of two-layer networks with 2-5 hidden neurons that successfully classify cancer. We have expanded the discussion to reflect this point and contextualize the meaning of depth:

With two layers, our networks are deep enough for molecular diagnosis. While gene expression datasets[66] have far too few samples (~100-1,000 patients per cohort) and too many dimensions (~10^4-10^5 RNA transcripts per patient) to meaningfully train very deep networks[52], small networks can be successfully trained with dimensionality reduction. Two-layer *in silico* perceptrons with two to five hidden nodes reliably predicted metastasis[67] or diagnosed breast cancer[68]. In the near future, a fully molecular neuromorphic system could integrate many clues[9] to diagnose patients at the point-of-care, and may even autonomously trigger a drug response[69]

Molecular networks could also empower DNA storage[70], as they would offer fully molecular querying and processing of DNA databases. For recognition of images or patterns in DNA databases, future networks would need between 10 and 20 layers[1], a tenfold gap with the networks reported here. But the gap could be closed – or at least partially filled – within the next decade. Our molecular architecture is scalable and modular (as demonstrated by the majority voting), writing and reading DNA is getting exponentially cheaper, and droplet microfluidic is enabling the handling of millions of molecular systems in parallel[46,47,71].

the realized function is logic NOR

Before delving into our reply, we would like to clarify the function computed by the network in Fig. 5. of our paper (previously Fig. 4) The last layer does compute a logical NOR of the previous layer (a function that is easily realized with DNA-based logic gates, as rightly pointed by the reviewer). **But the network *as a whole* computes a function that is vastly more complex than a NOR**: it computes the partition of the concentration plane into three non-linearly separable regions - and it does so in a fully molecular way. We chose this analog computation because it examplifies multilayer perceptrons. It is an analog computation operating on continuous inputs (concentration of DNA), and cannot easily be realized with DNA-based logic gates, which are digital computations operating on binary inputs.

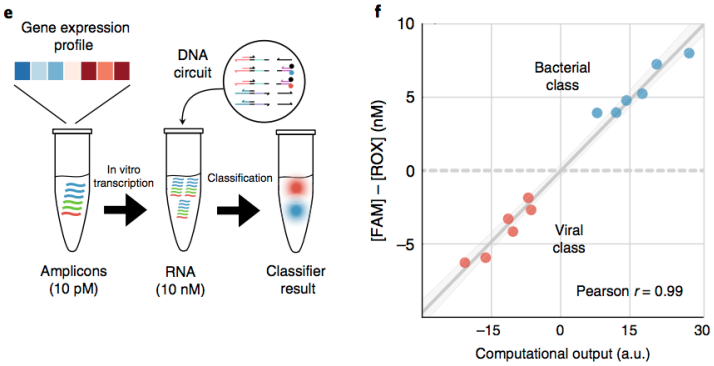
which is very different from the hidden layer.

We agree with the reviewer that the chemistry of the output layer was different from the hidden layer. We did it because we wanted to demonstrate the connection of neural and logical computations. But we took this comment seriously and we performed a new set of experiments where all the nodes have the same chemistry (see below). Before we describe this new experiment, we would like to stress that previous works (Seelig[9] and Qian[4]) also had hybrid architectures. Both papers tweaked the chemistry of their last layer to achieve their proof-of-concept.

In the case of Seelig (Fig. 5 of Lopez et al.[9]), they kept the summation layer, removed the last layer that performed the comparison, and finished this computation in-silico:

> *Rather than performing the subtraction at the molecular level as we have done in the previous example, we chose to use two distinct fluorophores to read out the positive and negative output strands individually, which allowed us to more quantitatively characterize the performance of individual classifier components.*

In other words, their molecular computation is not complete: the first layer is molecular and computes two sums of concentrations, but the second layer is electronic, and compares these two sums inside a computer.



[Lopez et al., Nature chemistry 2018] Fig. 5 e-f *"**e**, Gene expression patterns for each sample were replicated by mixing gene amplicons containing T7 RNA polymerase promoter sequences in the ratios expected from the microarray data. Subsequently, the samples were in vitro transcribed, resulting in production of RNA molecules with approximately 1,000× amplification. Upon addition of the molecular classifier, fluorescence signals were recorded across both channels and a classification value was recorded. **f**, All samples were classified correctly by the molecular classifier: a positive normalized signal was obtained for bacterial class samples and a negative for*

viral class samples. The normalized fluorescence signal matches the estimated computational SVM output, reflecting the correct implementation of the weights in a sample containing multiple RNA transcripts."

In the case of Cherry *et al.*[4], they acknowledged that they could not scale up the WTA computation to 9 digits ("*Using the same method, it would be difficult to construct networks that remember more patterns*"). So they changed their strategy and decided to set for a less ambitious goal: recognize groups of 3 digits instead of a single digit (as discussed above)

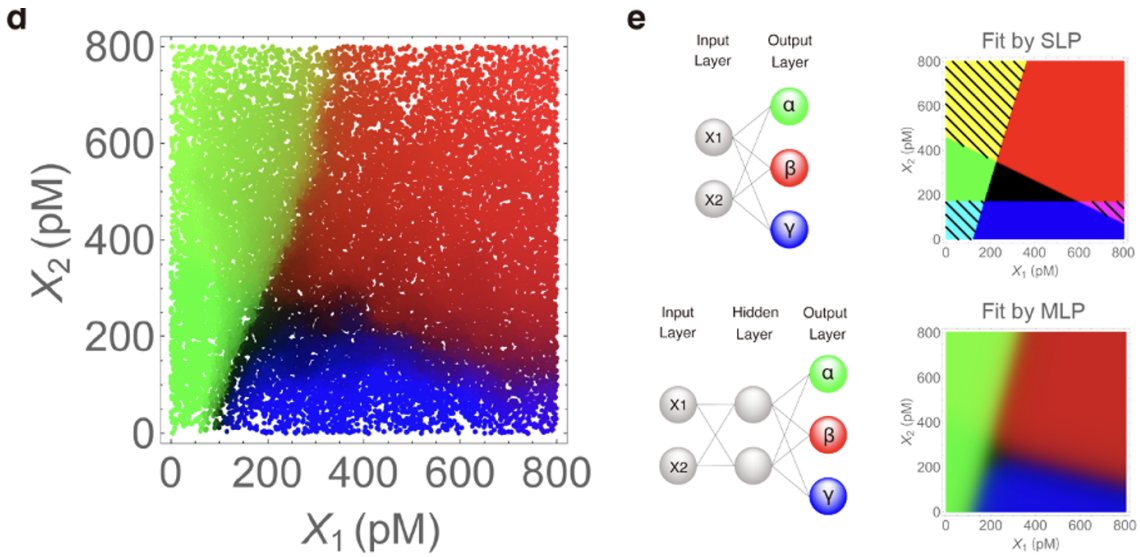
In both cases, the authors tweaked their last layer because they faced technical hurdles. In our case, we changed the chemistry of the last layer not because of fundamental technical hurdles, but because we wanted to demonstrate a hybrid computation that connected logic circuits and neuromorphic networks. This line of work on hybrid computation was mentioned by Cherry *et al*. in their paper[4] as a future development that would be desirable:

> *allowing the outputs of winner-take-all circuits to be connected to downstream logic circuits, could enable more sophisticated pattern*

So, we aimed to demonstrate a hybrid network combining neuromorphic networks and logic circuits by having a NOR in the last layer. Rather than limiting our network, this expands the range of possible usage for our enzymatic framework. We now better describe this in the text:

"The network is hybrid and comprises 2 computational layers (Fig. 5 b): a hidden neural layer deciding membership of the α and β region with linear classifiers, and a logical layer deciding membership of the γ region with a NOR gate"

We note that contrary to Seelig and Qian, our computation of space partitioning is complete and end-to-end molecular: our network takes as input a pair of chemical species (X1,X2) and returns a chemical species among three that represents the region the input belongs to. In the case of Seelig, the computation is completed in a computer, and in the case of Qian, the network returns a pair of species that still need to be decoded to reveal the digit class. Lastly, unlike Seelig and Qian, our computation is nonlinear because it cannot be satisfyingly performed by a single-layer perceptron(see extract of Fig. 5 below).



(Extract of Figure 5 of the revised manuscript) **d,** Merged fluorescence plots **e**, Fit of d by a single-layer perceptron (top), and a two-layer perceptron (bottom). The hatched filling indicates erroneous areas where two classes are outputted

Back to actual changes in the manuscript, we took the comments from this reviewer very seriously because the readership of Nature is broad, and other readers would be left wondering if we can implement other multilayer perceptrons than the one shown in Figure 5. In response, we now implement not one nonlinear function, but a whole family of rectangular functions with tunable windows (Fig. 4, see below). We wired three chemical neurons into a two-layer perceptron with one input, two nodes in its hidden layer, and one output neuron. The network computes a canonical non-linear function on one input: a rectangular function, and the input is now a microRNA. With microfluidic scanning, we map this family of functions (changing the input and the drain of one of the nodes). We find that the drain behaves exactly like a bias in a perceptron, and that it controls linearly the width of the window in the rectangular function. Incidentally, we took this opportunity to tune our network. It now detects RNA below 10 pM, a ten-fold improvement over our own linear classifier, and a 1000-fold improvement over the linear classifier of Seelig. We believe this new set of experiments emphasizes the programmability and equivalence of our networks with multilayer perceptrons, as well as their biological relevance thanks to their handling of microRNA.

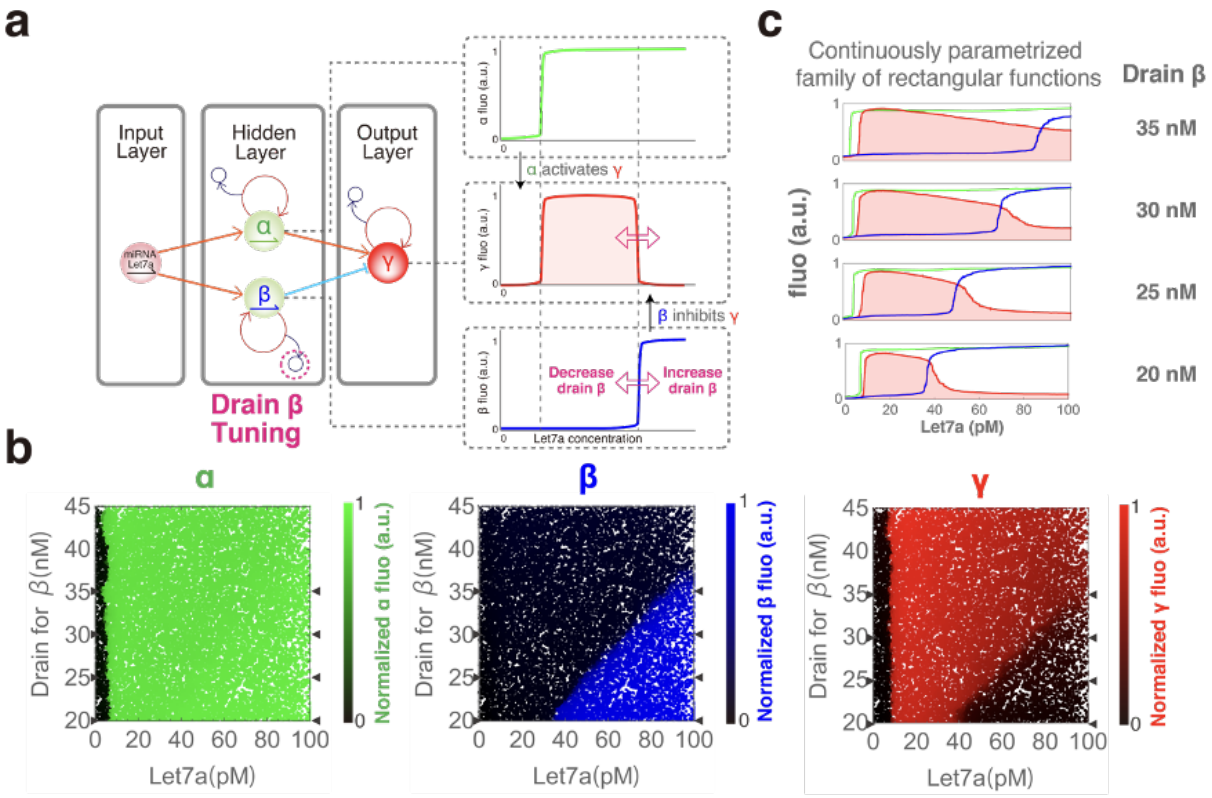


Figure 4: **Synthesis of a parametric family of rectangular functions with a multilayer perceptron.** **a**, A microRNA input activates two neurons (α and β) in the hidden layer. The output neuron γ is activated by α and inhibited by β. As a result of these opposing actions, γ is only active when α is active and β inactive – producing a rectangular function on the input. **b**, Droplets microfluidic scanning of the concentration of input and a parameter of the network (the concentration of drain for β) . The steady state fluorescence of α, β and γ (after 14h) are shown against the concentrations of let7a and β drain. **c**, Smoothed horizontal slices in the γ plot (gray arrows) reveal the tunable rectangular function.

The followings are a list of relatively minor issues that the authors may want to address:

3) The fitting algorithm for separatrix should be more clearly described. And was the fitted line deviated from the desired separatrix? Any calibrations needed?

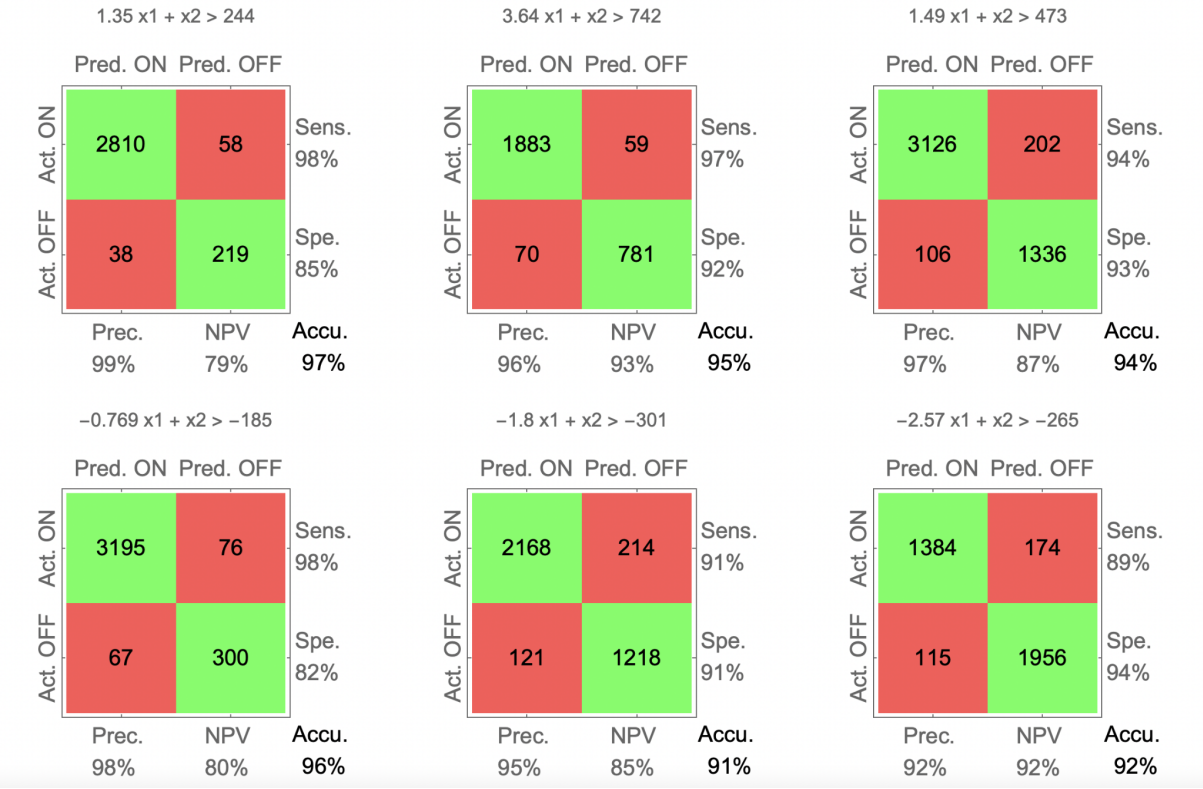
We thank the reviewer for this excellent comment as we realized that we had not clearly described the fitting procedure. Briefly, we first find points around the boundary by selecting points around a given fluorescence. This required some calibration and we settled for a fluorescence level of 0.3, which gives consistent results across all data sets. Then we linearly fitted these sets of points to extract the equation of the boundary.

We now added the following language in the method section:

"We fitted the separatrix as follows. First, we bin the droplets along one axis, say X1, by groups of ~100 droplets. For each bin, we then determine the position of the boundary along the X2 axis by running a moving median on the fluorescence of the droplets, and finding the first X2 for which the median hits a given threshold. (We chose a common threshold of 0.3, rather 0.5, because it fitted better the datasets for the negative classifier – which had a slightly lower

fluorescence at their boundaries). This procedure yielded a set of points on the boundary, which we linearly fit to extract the equation of the separatrix."

We note that the quality of the fit is confirmed *a posteriori* by the confusion matrices that we now compute for each classifier (as suggested by reviewer, comment 5). Each confusion matrix gives a measure from the deviation from the desired separatrix, by plotting the number of true/false positives and negatives. From this, we extracted 5 statistical measures: specificity, sensitivity, precision, accuracy, and negative predictive value, which are in the range of 80%-100%



4) The linearity and decision margin for negative-weighted classifiers are worse than the positive ones. The authors should include a more detailed discussion about this.

We thank the reviewer for this important observation. First, we note that we can only observe this deviation from perfect linearity because of the quality and quantity of our data: the switch between ON and OFF is clearly cut, and the large number of data points makes it easy to see where the experimental separatrix deviates from ideality. But it is true that the linearity of the negative-weighted classifiers is worse than the positive one. This is not entirely surprising because negative weighting is built on a doubly catalytic strategy, and also because the total amount of drain varies with X1 in the negative-weighted classifier (unlike the positive-one). We think that the deviation from linearity depends on the working temperature and concentrations.

We added the following detailed discussion in the SI

"While the boundary of the negative-classifier is sharp, its shape is slightly less linear than the positive-weighted classifier. This is likely explained by the doubly catalytic activity of the input X1 when it is negatively weighted. Each X1 input catalyzes the production of several drains, and each drain catalyzes the inactivation of several triggers. If the concentration of X1 or drains are too large (compared to their corresponding Michaelis-Menten constant Km), then the Michaelis-Menten equation for the production of drain by X1, or the equation for the inactivation of the trigger by the drain, cease to be linear, and the boundary is expected to deviate from linearity. Additionally, the total amount of drain varies in the negative-weighted classifier, while it is constant in the positive-weighted classifier. This could account for the downward drift in fluorescence observed in the ON region of the negative-weighted classifier (Fig. 2). Increasing X1 produces more drains in solution, which decrease the steady state of trigger through two mechanisms. First the drain recruits more polymerases for inactivating α, which mechanically reduces the polymerase that is free to activate α (through recruitment by the autocatalytic template). Secondly, the drain acts as a kinetic sink for α: increasing the drain increases the removal rate of α, and thus decreases its steady-state level. In principle, linearity could be improved by working in a more linear regime of concentrations by lowering the concentration of inputs or converter templates."

5) It is unfair to compare the Hill coefficients with Lopez's work as they used RNA as input, not "DNA-only networks". Maybe a comparison with Qian's work (Nature 559, 370, 2018) is more reasonable?

We thank the reviewer for this important note, which encouraged us to do new experiments and demonstrate classification on RNA inputs.

To summarize

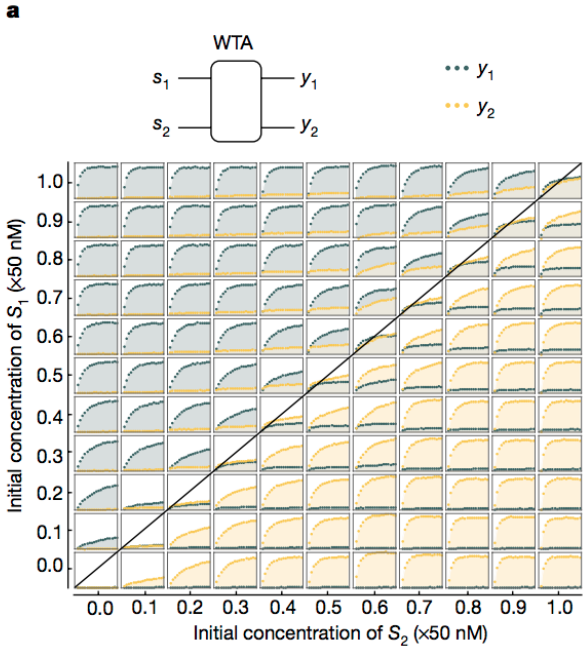
- **We replaced occurrence of "DNA-only networks" by "nonenzymatic networks"**
- **It is difficult to compare with Qian's paper because they did not characterize a linear classifier, and they use highly optimized and artificial sequences, unlike us and Seelig who used natural sequences**
- **We performed new experiment where we do nonlinear classification on RNA inputs**

The work of Lopez has been an inspiration and a yardstick for us, because it is the state-of-the-art for linear classifiers in DNA computing. The goal of Qian was to classify images encoded in DNA, but it was not to demonstrate linear classification, nor was it to work in conditions relevant for molecular diagnosis. This contrasts with Seelig's paper who demonstrated linear classification with tunable weights on natural RNA in the range of 2-20 nM.

We agree that the expression "DNA-only network" is confusing, because it implies that the inputs are also restricted to DNA. We have now replaced occurrences of "DNA-only networks" in the paper by "nonenzymatic networks"

We and Seelig implemented a complete linear classifier, with a weighted summation layer and a comparison layer. In the paper of Qian, there are no such examples of linear classification *per se*. The closest thing is Figure 2a (Cherry et al.[4]) where they compare species s1 and s2.

But it is not truly a linear classifier because there is no summation layer (just a comparison layer). The species s1 and s2 are directly injected together at their final concentrations. It turns out that the comparator is highly sensitive to kinetics, and if s1 and s2 were produced by a summation layer, their production kinetics would need to be finely tuned to avoid errors. So we do not believe this example counts like a general example of linear classification.



[Cherry, Nature 2018] Fig 2-a . *Two-species winner-take-all behaviour. The standard concentration is 50 nM (1×). The circuit is composed of two weighted- sum strands (S1 and S2), an annihilator molecule ([Anh1,2] = 75 nM (1.5×)), two restoration gates ([RG1] = [RG2] = 50 nM (1×)), twofuel strands ([YF1] = [YF2] = 100 nM (2×)) and two reporters ([Rep1] = [Rep2] = 100 nM (2×)). Initial concentrations of S1 and S2 are shown as fractions of the standard concentration.*

Lastly and most importantly, in Qian's comparator, the sequences of the inputs are not those of naturally occurring nucleic acids, but artificial sequences with highly constrained designs: three letters alphabet, no runs of more than four consecutive As or Ts or more than three consecutive Cs, G/C content between 30% and 70%, clamp in the branch migration domain, and all pairs of sequence have at least 30% different nucleotides (see Method section of their paper[4]). By contrast, we and Seelig work with sequences of naturally occurring nucleic acids mir-21 and mir-31 in our case, GAPDH and hTERT for Seelig.

This being said, we fully accept that we worked with DNA inputs rather than RNA inputs. But our networks can equally work with RNA inputs, because the fundamental enzymatic operation on inputs (elongation of the input by the polymerase) also works with RNA: the only requirement for an input to be extended by the polymerase is to possess a 3' OH end - which both DNA and RNA do.

In response to this important comment by the reviewer, and in order to broaden the appeal of the paper for the life science community, we used this revision to perform experiments with RNA inputs (Figure 4). This set of experiments - which was encouraged by comments from this reviewer and others - is focused on a more difficult problem than linear classification. As indicated above, we synthesized a continuous family of parametric networks for computing rectangular functions (an important set of functions in signal processing), and we used RNA

inputs in response to this comment. Overall the results are excellent: we detect a microRNA below 10 pM (a 10-fold improvement over the first version of the manuscript) with a sharp response. So we hope that this experiment restores fairness with Seelig's work, in that we now show computations with RNA input.

6) In the majority voting part, the author included a comparison of their neuromorphic network to DNA logic circuits. Actually, the advantages of neural networks over logic circuits have been discussed previously (Nature 559, 370, 2018). As the novelty of this manuscript lies in the use of enzymatic reactions, discussions on realizing the multi-bit majority voting with enzymatic networks as compared to the DNA-only networks may provide better insights.

We thank the reviewer for this excellent suggestion.

To summarize:

- **We have added a discussion on the scalability of the multi-bit majority voting with enzymatic networks as compared to the DNA-only networks.**
- **We find that the networks of Qian and Seelig could not realistically perform majority voting on 3 bits or more due to their limited decision margin**

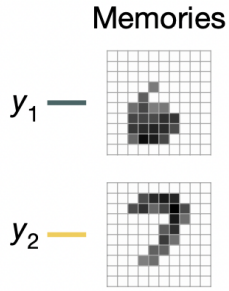
First, we would like to note that Qian and Cherry compared the size of neural networks and logic circuits for a benchmark (pattern recognition) that naturally favors neural networks. After all, neural networks are known to excel at detecting patterns in images, and it is not entirely surprising that a DNA neural network would outperform a DNA logic circuit on this task. Now we consider a different benchmark, one that naturally favors logic circuits: the computation of a Boolean function (majority voting). In that case, the dominance of neural networks over logic circuits is not obvious. If we allow NOT gates, then the majority function can be computed with a number of logic gates that scales linearly with the input size - similarly to neural networks. It is only if we forbid NOT gates (*i.e.* we restrict to the class of monotone circuits) that the size of the logic circuits blows up (at least) quadratically with the input size. But this case is highly relevant for DNA computing because in practice it has proved difficult to make NOT gates in DNA circuits. The community has borrowed a workaround from electrical engineering (dual-rail logic), which pushes the NOT gates back onto the inputs, where they can be computed by the operator. This dual-rail design was used in the landmark Science paper of Qian *et al.* in 2011[6], which is why we spent some time explaining why monotone circuits are relevant for DNA computing. But dual-rail design may not be entirely relevant when the inputs are fed by a biological system, because this design essentially asks the operator to compute the NOT itself. So that is why discussed the scaling in the context of monotone circuit

This being said, the question of the reviewer is highly relevant: how does the scalability of DNA-only networks compare with enzymatic networks for majority voting? Again we address this question using the work of Qian and Seelig as a benchmark. Although it is a thought experiment (because they have not actually done majority voting), we can estimate the scalability of a hypothetical majority voting based on nonenzymatic DNA networks. First, the size of both networks (enzymatic and DNA-only) would scale linearly because they use the

same design at a high level (summation of concentrations and nonlinear thresholding). But the sharpness of the linear classifier constrains the maximum number of votes that can be distinguished: the more there are votes, the sharper the decision margin must be. To make a majority vote with a classifier, the relative decision margin (that is the relative difference between the closest concentrations that can be discriminated) must be below 2/n or 2/(n-1), where n is the number of input in the majority vote (details of computations have been added in the SI section on majority voting). For instance, a relative decision margin of 50% is needed for a majority vote on 4 inputs. Based on our decision margin (~20%), we estimate that we could do majority voting on 10-11 bits, which is confirmed experimentally.

We estimated the relative decision margins for the classifier of Seelig (~66%) and Qian (~85%), which is significantly worse than our classifier (~20%) (details of the estimates have been included in the SI in the section on Majority Voting). Based on this, it is unlikely that these nonenzymatic networks could tally the votes of more than 2 or 3 bits.

For Seelig, this limit of 2 or 3 inputs probably explains why they performed the nonlinear computations in-silico when they dealt with 7 genes (Fig. 5 of Lopez et al.[9]). For Qian, it shows that high-dimensional DNA classification (like classifying images of handwritten digits from MNIST) can be easier than a low-dimensional DNA classification, like taking a majority over 4 bits. It might be counterintuitive at first, but MNIST is a peculiar dataset in some sense. As 80% of the pixels in the images are OFF, one can find groups of pixels that are a unique signature for a digit. For instance, if we look at the average patterns for 6 and 7 in Figure 3 of Qian:



On average, the 6 digits tend to cluster on the bottom left part of the image, while the 7 digits tend to cluster on the top right part of the image. Which means one could distinguish a 6 from a 7 just by looking at a few pixels in the bottom left or top right. This is made possible because images live in a 100-dimensional space, and there are many ways to choose a hyperplane that separates a cluster of 6 from a cluster of 7 with decent accuracy. Actually, one of the first papers on MNIST reported a surprising classification accuracy of 88% with just a linear classifier[10]. The seminal paper of Cherry and Qian is a first step toward achieving this accuracy with molecules, as their strategy has the potential to recognize up to 20% of MNIST with DNA.

Although it is counter-intuitive, majority voting on n=10 bits demands better decision margins than recognizing 20% of MNIST. When we do majority voting on *n* bits, we want to linearly separate the cluster with >n/2 inputs ON from the cluster with ≤ n/2 inputs ON. The separating hyperplane is essentially fixed by definition (the normal being the unit vector (1,1,...1)), and

we only get to tune the distance of this hyperplane to each cluster - with a small relative margin of 2/n for manoeuvre. In the case of MNIST, we can choose more freely the best fitting hyperplane, whereas for majority voting, the hyperplane and classification margin are fixed, and we have no other choice than to improve the raw performance of the classifier.

In response to this comment, we have added estimates and discussion of scalability in the SI (section on majority voting)

It is also instructive to compare the theoretical *scalability* of enzymatic networks and nonenzymatic networks for computing the majority function. Let us define *c* as the concentration of an input strand that is ON. The nonenzymatic networks of Cherry *et al.*[13] and Lopez *et al.*[14] both compare weighted sums of concentrations; they could *in principle* compute the strict majority of *n* inputs by setting the weights of one sum to 1, and fixing the other sum to an appropriate threshold between (n/2-1)*c and (n/2+1)*c. In theory, the number of strands required would scale similarly (*i.e.* linearly) for both types of networks (enzymatic or nonenzymatic) because they share the same design at a high-level (summation of concentrations and comparison to a fixed threshold). But in practice, the sharpness of their decision margin constrains the maximum number of votes they can reliably tally. For strict majority voting on *n* inputs, the linear classifier must discriminate between a total concentration of inputs of n/2*c and (n/2+1)*c if *n* is even, or (n/2-1/2)*c and (n/2+1/2)*c if *n* is odd, a relative difference of 2/n or 2/(n-1) respectively. For instance, voting with 4 or 5 inputs requires the decision margin to be better than 50%. Our enzymatic network discriminate inputs whose total concentrations change by as little as 20% (Fig. 2), and so it could handle majority voting for 10 to 11 bits – in line with experimental performances.

As for the linear classifier of Lopez *et al.*, given two concentrations of species X (GAPDH) and Y (HTERT) (expressed in absolute concentration), the equation of its boundary is Y=2X (Fig. 3d in[14]). For instance, setting X = 5.7 nM (5th column from the left) for majority voting, this linear classifier can discriminate between Y = 8.6 nM (4th row from the bottom) and Y = 14.3 nM (6th row from the bottom), a relative difference of ~66%. This decision margin would limit majority voting to 2 or 3 bits at most.

Regarding the comparator of Cherry and Qian, given two sums of concentrations S_1 and S_2 (expressed in normalized concentrations), it can distinguish between $S_1 > S_2+0.15$ and $S_1 < S_2-0.15$ (Fig. 3d in[13]). Setting $S_2 = 0.5$ for majority voting, this comparator would discriminate between $S_1 = 0.35$ and $S_1 = 0.65$, a relative difference of 85%. This decision margin would also limit majority voting to 2 or 3 bits at most.

It could seem counterintuitive that a problem of classification in low dimension (majority voting on 4 inputs or more) would be more difficult for a DNA classifier than a classification in high-dimension on (recognizing 100-bits images from MNIST). This exemplifies the peculiarity of classification in high-dimension. For instance, it is much easier for the human brain to recognize an object in a high-resolution image than in a low-dimension image – simply because there is more information available to discriminate between objects. This is reinforced by the peculiarity of MNIST: not only do its images live in high-dimensional space, but ~80% of their pixels are OFF. So, it is relatively easy to find a group of ON pixels that uniquely identify

a given digit - a signature that can be used to separate - even roughly - this digit from the other digits with a hyperplane. Actually, as much as 88% of MNIST can be linearly separated (i.e. its digits can be separated by a single-layer perceptron[15]). As long as one does not aim to recognize a large portion of MNIST, majority voting on *n* bits remains more demanding for a DNA linear classifier than MNIST classification. This is because the definition of majority voting essentially fixes the position and direction of the separating hyperplane (its normal is the n-dimensional vector full of 1), and there is little room for maneuver but to get the relative decision margin of the DNA classifier below 2/n or 2/(n-1) (which is the relative distance between the sets of OFF and ON vectors)

Referee #2 (Remarks to the Author):

The authors present a linear classifier, a classifier using decision tree and a majority logic with ten inputs using DNA and enzymes. Their functionality is validated by experimental results at temperatures ranging from 40-49 degrees Celsius. Comments on the paper are listed below.

We are immensely grateful to the reviewers for their time on the manuscript and their expert reading. We used their comments to revise and vastly improve the manuscript, and we hope that they will be satisfied with the revised version.

To summarize at a high level:

- We have clarified, disambiguated and expanded various points, stressing why our paper is novel and original. We added references to better explain background research on molecular neuromorphic computation and possible future developments.
- We discuss what it means for a network to be deep. We added discussion about two possible applications of our networks (DNA storage and molecular diagnosis of cancer) to show the depths that would be needed for practical computations
- We revamped figures, making them clearer, more visually appealing and more streamlined. We now make hybrid raw/smooth plots that retain their experimental feeling while being easier to read and saving space.
- We performed new analysis to back our claims - confirming the excellent performance of our networks over the state-of-the-art, and demonstrating that our network cannot be satisfyingly captured by a single-layer perceptron.
- We vastly extended the experimental demonstration to broaden the appeal of the paper, implementing a fully tunable multilayer perceptron that computed a boxcar function. We also extended majority voting and implemented a new functionality: majority right

1) In the Abstract and P. 2, the authors state “molecular classifications equivalent to a multilayer neural network (i.e. nonlinear partitioning of a concentration space) have been elusive.” Recent papers have demonstrated feasibility of perceptrons and artificial neural networks using DNA including activation functions such as sigmoid, ReLU and softmax. See for example,

https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9028230

http://www.nature.com/articles/s41598-018-26709-6

We thank the referee for pointing to these important papers. We had kept our overview on DNA computing short, for sake of brevity and space. We focused on the two most recent and significant experimental results in DNA computing: the paper of Lopez *et al*. in Nature Chemistry[9], and the paper of Cherry *et al.* in Nature[4], both in 2018. But we agree that we should have added other and more general results, including feasibility studies, to provide

more breadth and give a better sense of what is feasible with DNA, and what has been achieved so far.

In response, we have revised the introduction. We follow the narrative of (Salehi *et al,* Sci. Rep., 2018) on chemical neuromorphic computation[13], and added about a dozen of references ranging from two foundational papers from Hjelmfelt *et al.* in the 90s[14,15], to the more recent papers from the Parhi group cited by the reviewer[13,16]. We believe that these papers provide a more comprehensive background to the reader of neuromorphic molecular computations.

On the molecular side, various groups have explored the feasibility of neuromorphic computation with chemical reactions over the past three decades [3–7,18,22–28]

2) The claim that the paper has demonstrated "deep neuromorphic" in the title and "equivalent to a multilayer perceptron in ~25,000 cell-sized droplets" are partially true. **The networks may not be considered deep. Equivalence with multi-layer perceptron may be overstated.** Multilayer functions **can synthesize many functions** all of whom may not be realizable by the proposed experimental framework.

We thank the reviewer for this comment that we broke down in three replies on depth, equivalence with multi-layer perceptron, and synthesis of functions.

"**The networks may not be considered deep."**

This is a common (and long) reply made to the reviewers about the question of depth.

Summary:

Our networks are deep based on the historical and mathematical definition of depth (two layers or more). But the practical question is how deep do the networks need to be ? It depends on the applications. If we look at DNA storage, then 10-20 layers are required for advanced image recognition. We are still behind by a factor of 10, but we are hopeful that the gap could be filled - at least partially - within ten years thanks to advances in DNA synthesis, sequencing and microfluidics. If we focus on molecular diagnosis, our networks are already deep enough. We now give examples of two-layer networks with 2-5 hidden nodes that successfully classify cancer. We have added text to reflect this point and contextualize the meaning of depth.

In the collective imagination, "deep networks" have come to evoke massive networks with hundreds or thousands of layers. But this colloquial meaning has strayed from its mathematical and historical definition (perceptrons with two or more layers). It is also a far cry from the typical size of networks handled by practitioners of deep learning (typically 10 to 50 layers), as they have come to realize that deeper networks do not necessarily equate better networks. We chose the term "deep" to broaden the appeal of the paper for non-specialists, and also because our networks could be considered deep based on the mathematical and historical definition of depth, and the size of electronic networks used for molecular diagnosis. This being said, we are open to revising our terminology and replacing occurrences of "deep

networks" with "multilayer neural networks". But we would like first to put our work and terminology in context.

In their reference book on Deep learning published in 2015, Bengio, Courville and Goodfellow defined deep feedforward networks as perceptrons with several layers[17]:

> *Deep feedforward networks, also called feedforward neural networks, or Multilayer perceptrons (Chapter 6)*

Similarly, in their influential review of Deep Learning published in Nature in 2015, LeCun, Bengio and Hilton defined deep networks as architectures with multiple layers[11]:

> *A deep-learning architecture is a multilayer stack of simple modules, all (or most) of which are subject to learning, and many of which compute non-linear input–output mappings.*

This chasm between one layer and two or more layers follows from the historical development of artificial neural networks. Papert and Minsky famously proved in their 1969 book *Perceptrons* that a single-layer perceptron was limited: it can only compute linearly separable functions[18]. This dampened the enthusiasm of many who had hoped - like Rosenblatt, one of the inventors of perceptrons - that single-layer perceptrons could solve advanced tasks like image recognition. These negative results threw confusion in the field, contributed to a freeze of research fundings (AI winter) and raised an open question: if a single-layer perceptron is no good, then what can a two-layer perceptron compute ? Cybenko settled the question in a famous paper[19], showing that a perceptron with only two layers could approximate any continuous function. These negative results for single-layer perceptrons and positive results for two-layer perceptrons solidified the chasm between one and two layers: shallow networks (single-layer perceptrons) are limited to linearly separable functions, while deep networks (two or more layers) approximate any reasonable function. This is why perceptrons are considered "deep" as soon as they contain two layers.

Of course, the question now becomes: how deep does a DNA network need to be to do interesting computations? Then it depends on the field of applications of DNA networks. DNA storage is coming of age, and it has been speculated that DNA could store the whole multimedia production of humanity[12]. Neuromorphic DNA networks may provide a fully molecular way to query those databases, for example counting how many images of cats, dogs or cars a DNA database contains. If the current practice in computer vision is any guide, then a DNA network would need between 10 to 20 layers to perform advanced image recognition on a DNA database[20–22]. But this is encouraging given that molecular computing is roughly where electronic computing was 70 years ago: we know how to program basic chemical primitives (the equivalent of transistors), and we are now connecting them into larger networks (which would be the equivalent of integrated circuits). It is not a far stretch to imagine that we could close the gap with *in-silico* networks -partially at least - in the next decade and build DNA networks with 10 to 20 layers. Our molecular architecture is scalable and modular, writing and reading DNA is getting exponentially cheaper, and droplet microfluidic is enabling the handling of millions of molecular systems in parallel[23].

The gap between *in-moleculo* and *in-silico* networks is almost closed if we focus on the field molecular diagnosis - the most obvious application for our molecular networks. Thanks to the coming of age of techniques like transcriptomics or liquid biopsy, researchers now measure the expression level of hundreds to thousands of RNA transcripts for a single patient. They are increasingly using machine learning to fit this data and predict the type, stage and prognosis of a disease based on the gene expression profile of a patient[24]. But due to the peculiar nature of these datasets, researchers have tended to fit them with small networks that only have a few layers.

This is because biomedical datasets are sparse: collection of clinical data is expensive, lengthy and fraught with ethical problems. While standard datasets in image processing (MNIST, CIFAR-10 or ImageNet..) contain 10^4 to 10^6 examples, a clinical study only enrols a typical cohort of 10 to 1000 patients, of whom a substantial number will likely drop before the end of the study. Since very deep networks tend to be hungry for data, it is not necessarily wise to train impractically deep networks on such small datasets.

In addition, biomedical datasets have another peculiarity that makes them ill-suited for very deep learning: their dimensionality far exceeds their size. A transcriptomic study easily collects the expression level of tens of thousands of genes (the dimension of the dataset), but only for a limited number of patients (the sample size), which is in the range of 100-1000s for typical studies. The world of high-dimensionality/low-sample size statistics is replete with counter-intuitive behaviours[25], and does not easily lend itself to classical tools of statistical learning. When very deep networks are trained on such datasets, they show overfitting and high variance, because there is not enough information present in the dataset to constrain their numerous weights[26].

For these reasons, biomedical researchers have tended to stick - not without success - to the shallowest of the deep networks, and two-layer perceptrons have enjoyed a lasting popularity in the field[24,27]. To handle the curse of high-dimensionality and low-sample size, researchers first reduce dimensionality by homing in on a few genes that account for most of the observed phenotype, then they train small networks on this reduced set - adding or removing genes to improve accuracy. In oncology, Lancashire et al. trained a two-layer perceptron with 5 nodes in its hidden layer that predicted metastasis with 98% accuracy from the expression of 9 genes[28]. McDermott et al. classified miRNA signatures in blood for breast cancer with a 2-layer perceptron[29]. They achieved a sensitivity of 77% and a specificity of 74% with a network that only had three input nodes, two hidden nodes, and one output node. Overall, these results show that perceptrons with only two-layers and few nodes classify diseases with high accuracy - which suggests that our molecular networks could perform this kind of task in the near future.

To summarize our long answer: with two-layers, our networks are deep based on the historical and mathematical definition of depth, but they are still one order of magnitude behind the practical depth of networks used in computer vision - a gap which could be closed in the next ten years. If we restrict to the field of molecular diagnosis, then the size and depth of our networks are on par with the two-layers in-silico networks that have been successfully used to classify diseases.

This being said, the length of this reply shows that the word “deep” needs to be disambiguated in the paper. In the introduction we now emphasize that multilayer perceptrons and deep feedforward networks have been interchangeably used:

“Multilayer perceptrons - interchangeably called *deep feedforward networks* on historical and mathematical grounds[1,2]”

In the discussion, we now added two paragraphs to contextualize the depth of our networks. We acknowledge that there is still a factor of 10 to perform advanced image recognition, but we highlight avenues where progress has been fast (DNA writing and reading, microfluidic, enzymatic networks). Lastly, we cite literature in machine learning and biomedicine to show that a two-layer network with 2-5 node is deep enough to classify diseases:

“With two layers, our networks are deep enough for molecular diagnosis. While gene expression datasets[65] have far too few samples (~100-1,000 patients per cohort) and too many dimensions (~10^4-10^5 RNA transcripts per patient) to meaningfully train very deep networks[51], small networks can be successfully trained with dimensionality reduction. Two-layer *in silico* perceptrons with two to five hidden nodes reliably predicted metastasis[66] or diagnosed breast cancer[67]. In the near future, a fully molecular neuromorphic system could integrate many clues[9] to diagnose patients at the point-of-care, and may even autonomously trigger a drug response[68]

Molecular networks could also empower DNA storage[69], as they would offer fully molecular querying and processing of DNA databases. For recognition of images or patterns in DNA databases, future networks would need between 10 and 20 layers[1], a tenfold gap with the networks reported here. But the gap could be closed – or at least partially filled – within the next decade. Our molecular architecture is scalable and modular (as demonstrated by the majority voting), writing and reading DNA is getting exponentially cheaper, and droplet microfluidic is enabling the handling of millions of molecular systems in parallel[46,47,70].”

We are also open to revising our terminology. In the title of the paper, we use the term “deep network” rather than “multilayer perceptron” to broaden the appeal of the paper for the readership of Nature - who might not be aware that those two terms refer essentially to the same thing. But we accept that the word “deep” is ambiguous, and that readers may expect to find hundreds of layers in our paper (an unrealistic expectation since they would not commonly find this number of layers even in deep learning papers). For this reason, we are open to replacing occurrences of “deep network” by “multilayer perceptron” or “multilayer neural network”, which are equivalent. We are agnostic on this point, and we would be happy to hear from the reviewers to know if the broad - but potentially ambiguous - denomination of “deep” should be used, rather than the technical - but intimidating - term “multilayer perceptron”.

“Equivalence with multi-layer perceptron may be overstated. “

We thank the reviewer for this comment. First, we would like to clarify *why* we hardwired a NOR in the last layer. We did not do it because of fundamental technical hurdles, but because we wanted to demonstrate a hybrid computation that connected a logic circuit and a

neuromorphic network. This line of work on hybrid circuits was mentioned by Qian *et al.* in their Nature paper as a future development that would be desirable[4]:

> *allowing the outputs of winner-take-all circuits to be connected to downstream logic circuits, could enable more sophisticated pattern*

So, we aimed to demonstrate a hybrid network combining neuromorphic network and logic circuits by having a NOR in the last layer. Rather than limiting our network, this expands the range of possible usage for our enzymatic framework. We now better mention this motivation:

"The network is hybrid and comprises 2 computational layers (Fig. 5 b): a hidden neural layer deciding membership of the α and β region with linear classifiers, and a logical layer deciding membership of the γ region with a NOR gate"

Secondly, we would like to emphasize that this computation is equivalent to a multilayer perceptron (the multilayer perceptron with its last node hardwired to compute a NOR). Thirdly, in response to a comment by this reviewer, we fitted a single-layer perceptron and a two-layer perceptron to the data produced by the space partitioning experiment. Only the two-layer perceptron correctly captures our computations, which emphasizes the equivalence with multilayer perceptrons (see below, comment 6 of this reviewer).

"Multilayer functions can synthesize **many functions** all of whom may not be realizable by the proposed experimental framework."

But ultimately, we agree with the reviewer that the last node was fixed to compute a NOR gate, and that it could limit the expressivity of the network, that is its capacity to compute other multilayer perceptrons. (But this is not unlike the papers of Qian[4] and Seelig[9], who tweaked their last layer in their proof-of-principle to overcome technical hurdles: Seelig computed the last layer *in silico*, and Qian et al. gave up on their strategy to recognize a single digit, and went to recognize groups of three digits instead).

We took this comment seriously, and in response, we now synthetize not one nonlinear function, but a whole family of rectangular functions with tuneable windows (Fig.4 and below). We do this with a complete multilayer perceptron that has three fully tuneable chemical neurons. With microfluidic scanning, we map in one go this family of functions with respect to one parameter (the drain of one of the nodes). We find that the drain behaves exactly like a bias in a perceptron, and that it controls linearly the window's width of the rectangular function. We believe this new set of experiments emphasizes the programmability and equivalence of our networks with multilayer perceptrons. This new set of experiments is now presented in a new section in the paper.

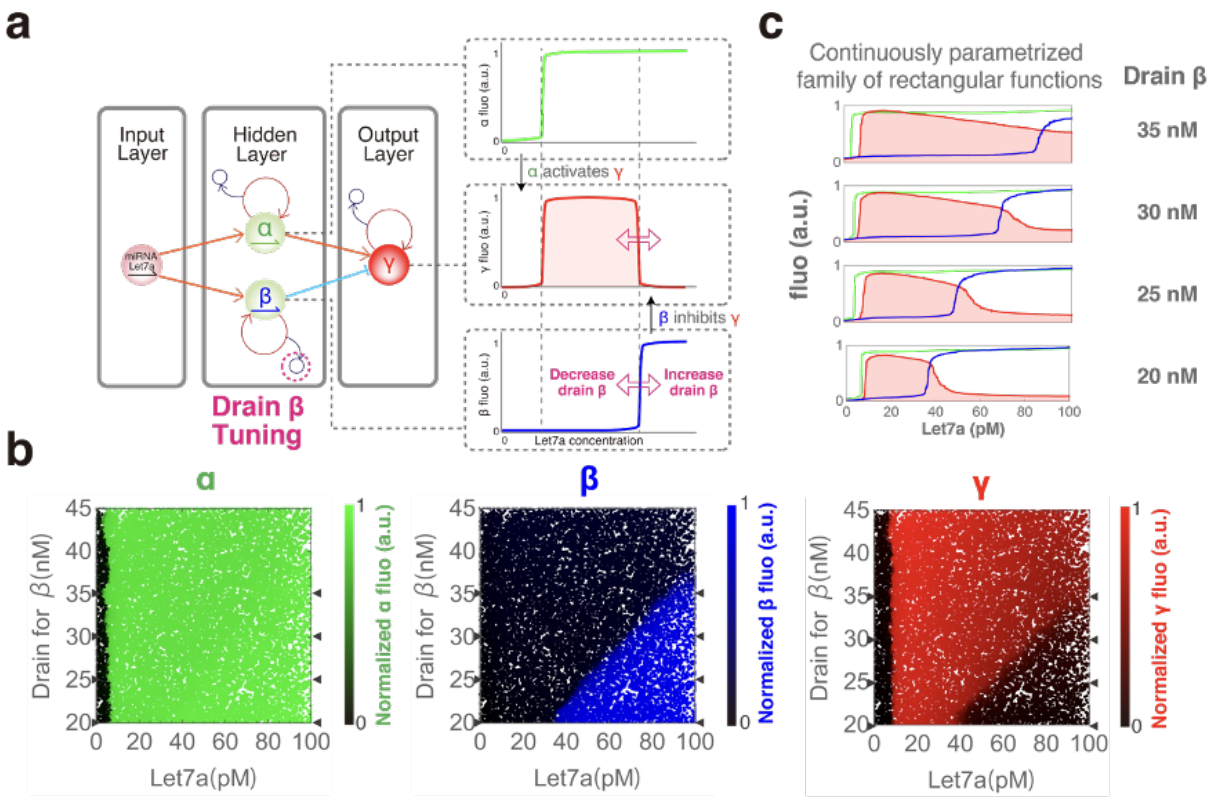


Figure 4: **Synthesis of a parametric family of rectangular functions with a multilayer perceptron. a**, A microRNA input activates two neurons (α and β) in the hidden layer. The output neuron γ is activated by α and inhibited by β. As a result of these opposing actions, γ is only active when α is active and β inactive – producing a rectangular function on the input. **b**, Droplets microfluidic scanning of the concentration of input and a parameter of the network (the concentration of drain for β) . The steady state fluorescence of α, β and γ (after 14h) are shown against the concentrations of let7a and β drain. **c**, Smoothed horizontal slices in the γ plot (gray arrows) reveal the tunable rectangular function.

3) In Fig. 1, it appears that the input is regenerated. Is the input not consumed?

We thank the reviewer for drawing our attention to this point as we had not mentioned it explicitly in the caption. As correctly noted, the input is regenerated during the enzymatic process. It binds to a template, which triggers its elongation by a polymerase. A nickase comes in and cuts the elongated domain, releasing the intact input, which can either be elongated again, or unbind from the template and start the whole process again with another template. By contrast the drain and reporting template do not regenerate their substrate. The drain irreversibly elongates the input, which makes it inactive. The reporting template uses the elongation of the input to produce a fluorescent signal.

In response to this important comment by the reviewer, we added the following language to the caption of Figure 1 and labels X1 in the figure to explicitly show where the input is regenerated.

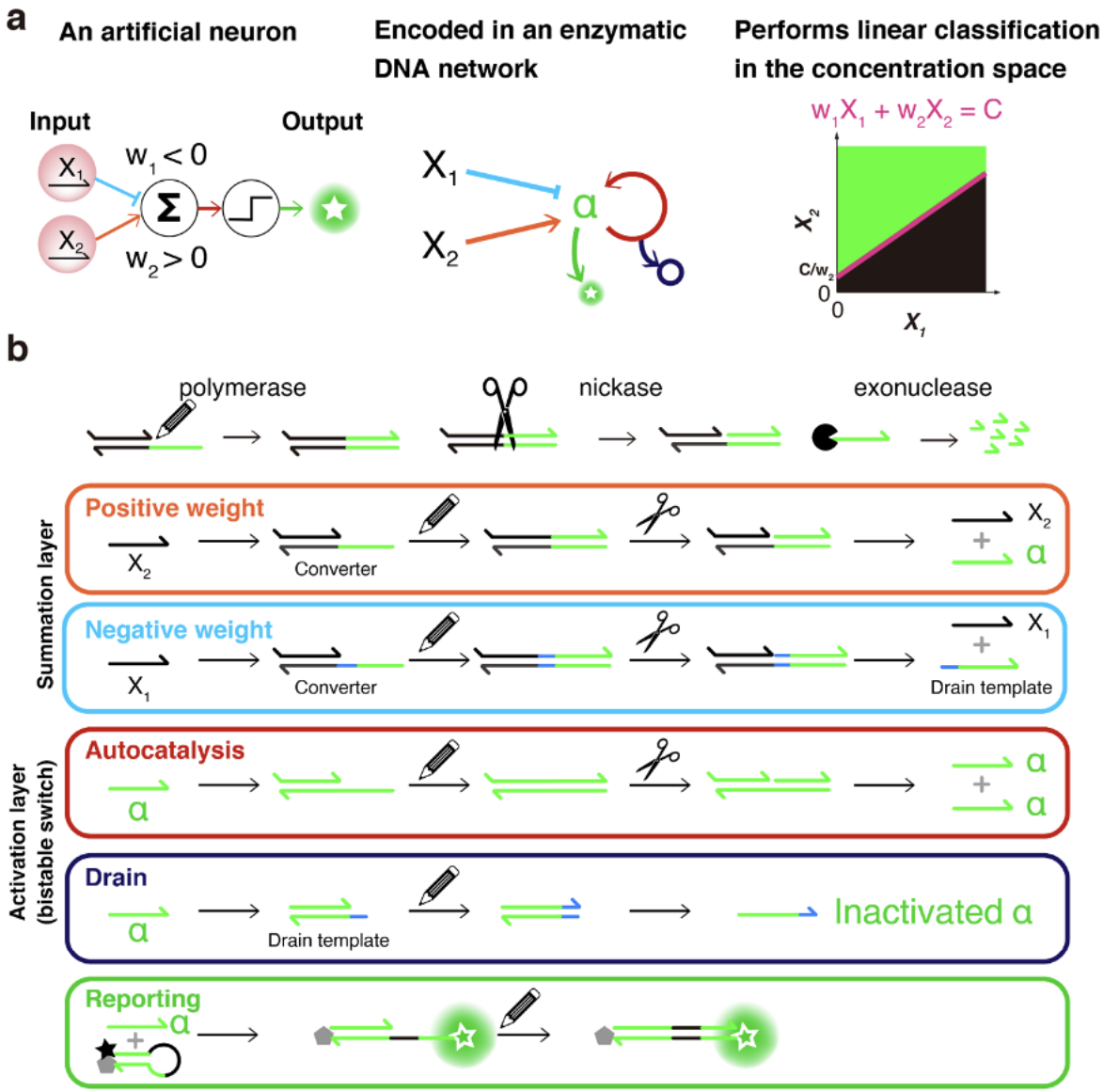


Fig. 1 ***Architecture of a DNA-encoded enzymatic neuron. a,*** *The molecular neuron is a DNA-encoded chemical reaction network that takes two DNA strands X_1 and X_2 as inputs, and produces a signal strand α only if the weighted sum of their concentrations exceeds a threshold.* ***b,*** *It is actuated by 3 DNA-processing enzymes that continuously produce (polymerase and nickase) or degrade (exonuclease) DNA strands. Positive-weight converters produce output signal strands (a, green) when primed by their input strand (black). Negative-weight converters produce drain templates that suppress the replication of the signal strand. The concentrations are adjusted so that the converters work in the linear regime. α is replicated by an autocatalytic template, inactivated by a drain template and degraded by the exonuclease - giving rise to bistability and thresholding. A fluorescent reporter tracks the level of its target strand. Converters and autocatalysis templates regenerate their substrate strand, while the drain and reporting templates do not.*

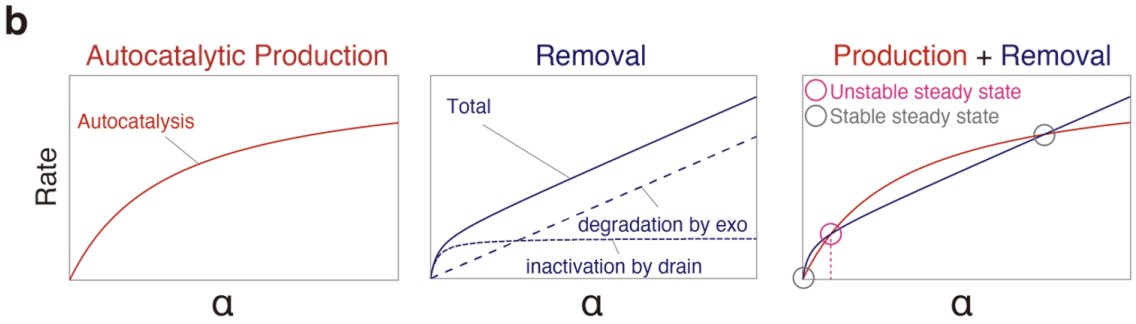
4) For Extended Data Fig. 4b, what condition can guarantee the crossover between autocatalytic production and removal?

We thank the reviewer for this comment. The crossover between autocatalytic production and removal is controlled by the unstable steady state indicated in red in the panel. If the concentration of α exceeds that threshold, then the drain is saturated, and production

overcomes removal, resulting in a net production. If α is below the threshold, then the inverse is true and alpha is removed.

The conditions for the existence of a crossover are controlled by the experimental conditions, most importantly, the concentrations of drain, polymerase, exonuclease, and of course temperature. The initial slope of the removal by the drain is controlled by the concentration of drain, the slope of the removal by the exonuclease is controlled by the concentration of this enzyme, and the shape of the production curve is controlled by the concentration of autocatalytic templates, polymerase and nickase. Those concentrations must be balanced so that the production curve intercept the degradation curve in three points, producing three steady states, of which one is unstable.

We have now added the following language in the caption of Extended Data Figure 4 to clarify this point:

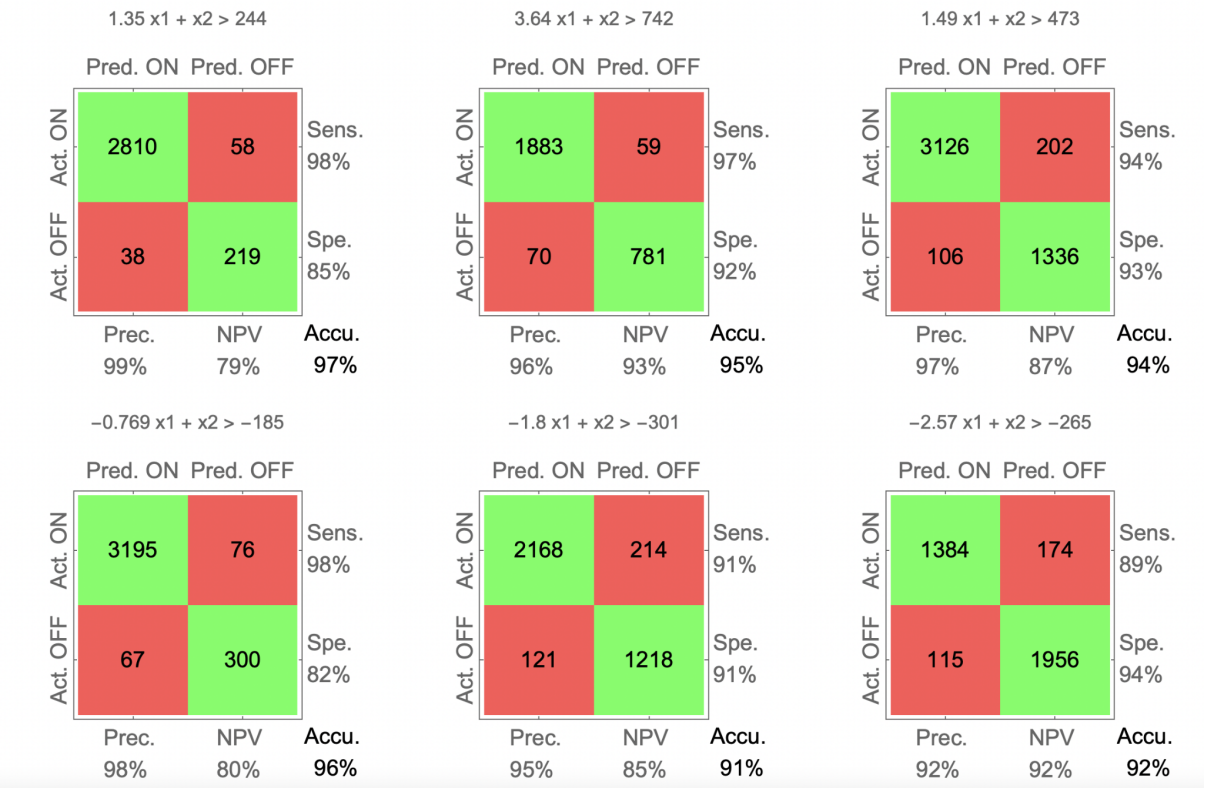


"the middle point is an unstable steady state, a threshold which controls the crossover between autocatalytic production and removal. If α is over the threshold, the drain is saturated, production overcomes the removal by the exonuclease and the drain, and α is amplified up to the ON state. Otherwise, α is removed down to the OFF state. The existence of an unstable steady state is controlled by the shape of production and removal curves, which must intersect at three points. The shape of the removal curve is controlled by the concentrations of drain templates, exonuclease and polymerase (for the inactivation step in the drain). The shape of the production curve is controlled by the concentration of autocatalytic templates, polymerase and nickase"

5) The scatter plots showing decision boundaries are good. It would be useful for readers to see the classification accuracy including sensitivity and specificity in tabular form and/or a confusion matrix form.

We thank the reviewer for this excellent suggestion because our high-throughput approach allows robust statistics to be computed. We now present confusion matrices in the panel **i** of the revised Figure 2 (see below for reference) for each of the 6 linear classifiers, from which we compute 5 statistical measures (namely sensitivity, specificity, precision, negative

predictive value and accuracy). Overall, this confirms the performance of our classifiers, with the statistical measures being in the range of 80% to 100%.



6) The data generated by the experiment could be fit into a multilayer perceptron to see the difference between the multilayer perceptron and the nonlinear enzymatic network.

We thank the reviewer for this excellent suggestion. We have fitted a single-layer perceptron and a multilayer perceptron to our data. Unsurprisingly, the single-layer perceptron does not fit our data correctly - failing on nonlinearly separable regions-while the multilayer perceptron quasi-perfectly fits. This is unlikely to be overfitting since the MLP only has two nodes in its hidden layer. We believe that this analysis underscores the equivalence of our computation with a multilayer perceptron.

We added the fit to the revised figure 4 (now Fig. 5e). We took this occasion to revamp the figure, and we moved the data on the temperature tuning to the SI. We added text to comment on this fit. We also added languages in the Method section to explain the fitting.

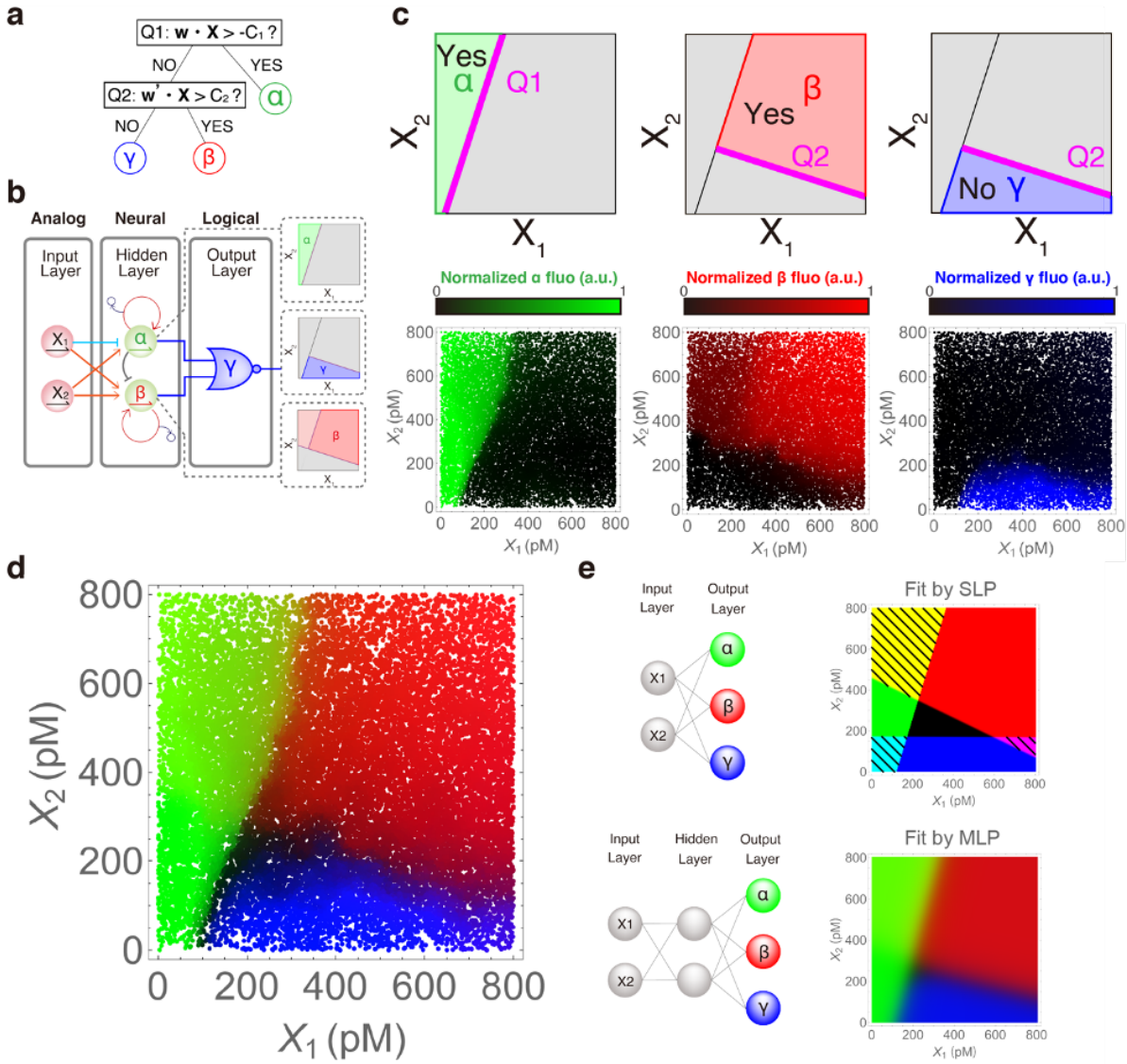
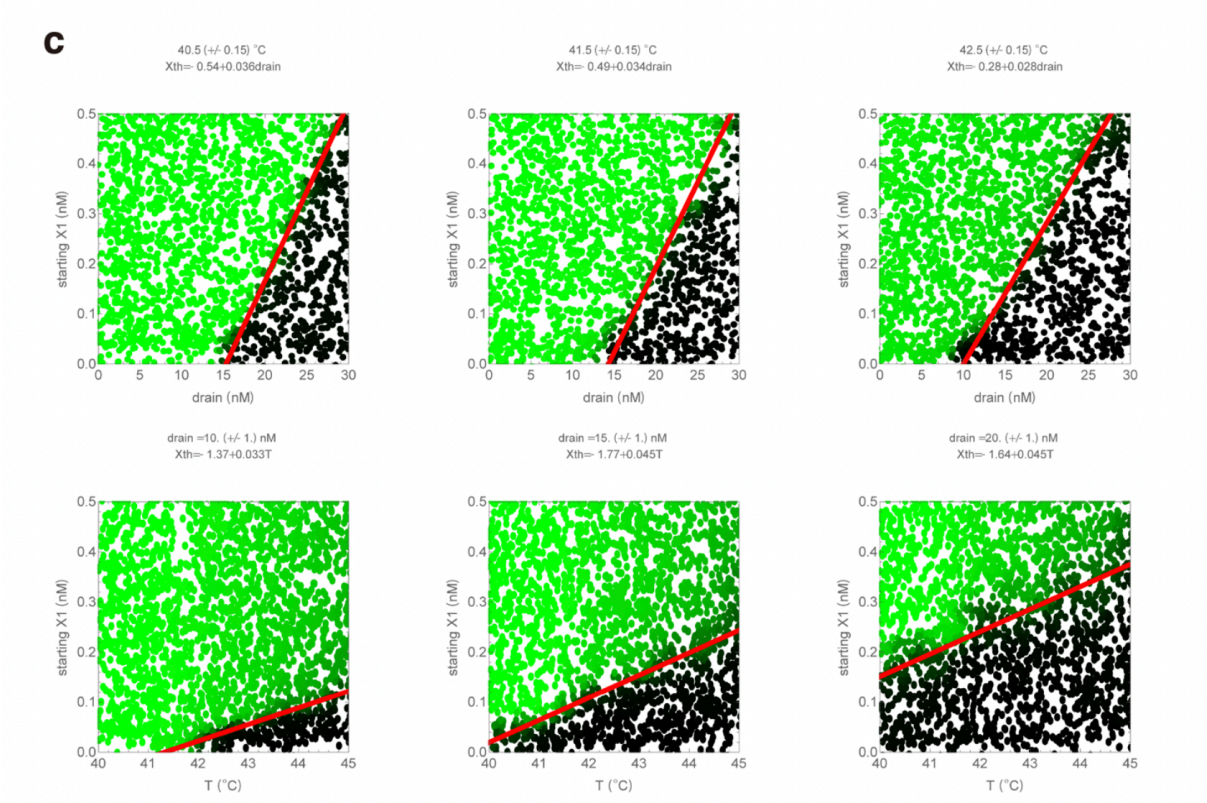


Fig. 5 A hybrid network computing recursive space partitioning. **a,** A space partitioning tree takes a point **X**=(X_1, X_2) in the concentration plane, and at each of its nodes, tests if **X** is a member of the corresponding half-plane. Computation finishes when a leaf is reached. The tree can be traversed in 3 ways, partitioning the plane into three convex regions. **b,** Architecture of a hybrid network computing the partitioning tree (Extended Data Fig. 8).. The inputs are two strands encoding a position (X_1,X_2) in the concentration plane. The hidden layer is neural and decides membership of the α and β region with two linear classifiers that are indirectly coupled by competitive inhibition (membership is readout by fluorescent reporters). The output layer is logical and decides membership of the γ region with a NOR gate (which turns its fluorescence OFF if X is a member of either α or β region). **c,** Fluorescence levels of α, β and γ, measured in ~25,000 droplets after 16 hours. **d,** Merged fluorescence plots **e,** Fit of d by a single-layer perceptron (top), and a two-layer perceptron (bottom). The hatched filling indicates erroneous areas where two classes are outputted.

7) There is sufficient variation when the temperature is varied. Some form of robustness or error analysis with respect to experimental parameters such as input concentration and temperature etc. may be useful.

We thank the reviewer for this highly relevant comment. Since our chemical neuron is analog, its computation continuously depends on its parameters. We use this control to tune the neuron, changing for instance the bias by tuning the concentration of drain, or the weights by tuning the concentration of converters. But it also makes the neuron sensitive to experimental errors, for instance pipetting errors or errors in setting the temperature.

In response to this comment, we analysed the data from Extended Data Figure 4, in which we studied a single-input neuron, and which is the ideal experiment to study sensitivity and robustness to experimental errors. In this experiment, we varied the drain, the starting concentration of X1, and the temperature of a neuron, and measured its level at steady state (fluoresence of alpha). We now take slices of this 3D dataset, showing how the bias varies with drain at fixed temperature, and how it varies with temperature at fixed drain. We linearly fit the bias and extract its sensitivity to drain and temperature;



Extract of Extended Data Fig.4 ***c,*** *Microfluidic mapping of the dependence of the bias on X1 to the drain. We prepared droplets with varying concentrations of drain for α and input X_1. We incubated the droplets in a temperature gradient, and imaged their content after 6 hours. The top plots show the fluorescence of droplets in the space (drain, X_1), the colour indicating the level of α. The bottom plots show the fluorescence of droplets in the space (temperature, X_1). The red lines are a linear fit of the boundary, with the equation indicated above each plot. The concentrations are in nM and the temperature in Celsius.*

For typical errors on concentrations (set by pipettes) and temperature (set by the Peltier elements and its sensors), we find that the error on the bias is 10% at most. We added a discussion in the main text referring to this new analysis.

"The chemical neuron is analog and its computation depends continuously on its parameters. On one hand, this can be used to program the parameters of the neuron, for instance its bias by tuning the drain. On the other hand, it makes the neuron sensitive to experimental errors. However, we find that for a typical operation, the deviations are likely to be minimal. More precisely, we analyzed the sensitivity of the bias of a single-input neuron to the drain and temperature. A pipetting error of 3% on the drain (typical for a calibrated pipette) translates into a 10% error on the bias, and an error of 0.1°C on temperature (typical reproducibility for a thermocycler) translates into a 1.6% error on the bias (both measured for drain= 20 nM and T=41.5 °C, Extended Data Figure 4)."

****

Referee #3 (Remarks to the Author):

A majority voting algorithm and a 3-layer perceptron were implemented using the so-called PEN TOOLBOX. They have succeeded in visualizing the characteristics of the obtained circuit by making a large number of droplets using a microfluidic device and comprehensively investigating the whole concentration space of the input chemical substances.

Although some studies on neural networks by DNA computing have been conducted so far, it has been difficult to construct a multi-layered circuit since the concentration can only be a positive value. By adopting PEN, the logic can be greatly simplified, and it is highly evaluated that the three-layer perceptron could be implemented by chemical reaction.

Comprehensive characterization using the droplet system has made it possible to completely visualize the characteristics of the constructed system, which is a result supported by extremely reliable experimental technology.

We are immensely grateful to the reviewers for their time on the manuscript and their expert reading. We used their comments to revise and vastly improve the manuscript, and we hope that they will be satisfied with the revised version.

To summarize at a high level:

- We have clarified, disambiguated and expanded various points, stressing why our paper is novel and original. We added references to better explain background research on molecular neuromorphic computation and possible future developments.
- We discuss what it means for a network to be deep. We added discussion about two possible applications of our networks (DNA storage and molecular diagnosis of cancer) to show the depths that would be needed for practical computations
- We revamped figures, making them clearer, more visually appealing and more streamlined. We now make hybrid raw/smooth plots that retain their experimental feeling while being easier to read and saving space.
- We performed new analysis to back our claims - confirming the excellent performance of our networks over the state-of-the-art, and demonstrating that our network cannot be satisfyingly captured by a single-layer perceptron.

- We vastly extended the experimental demonstration to broaden the appeal of the paper, implementing a fully tunable multilayer perceptron that computed a boxcar function. We also extended majority voting and implemented a new functionality: majority right

Some suggestions:

1) The expression “deep neuromorphic network” in the title is generally considered to refer to a large scale multi-layered neural network. I think it is an exaggeration to call a perceptron with 2 inputs, 3 layers, and 1 output a “deep” neural network.

We thank the reviewer for this comment and agree that the term “deep” could be ambivalent for non-specialist readers. Since this point was raised several times in the review, we made a detailed and common answer which is given in the responses to reviewer 2 (comment 2), which we encourage the reviewer to read. For convenience, a short summary is below.

Summary:
Our networks are deep based on the historical and mathematical definition of depth (two layers or more). But the practical question is how deep do the networks need to be ? It depends on the applications. If we look at DNA storage, then 10-20 layers are required for advanced image recognition. We are still behind by a factor of 10, but we are hopeful that the gap could be filled - at least partially - within ten years thanks to advances in DNA synthesis, sequencing and microfluidics. If we focus on molecular diagnosis, our networks are already deep enough. We now give examples of two-layer network 2-5 hidden nodes that successfully classify cancer. We have added text to reflect this point and contextualize the meaning of depth.

2) The expression “demonstrating computations equivalent to multilayer perceptron in ~ 25000 cell-sized droplets” at the end of the abstract is misleading. This expression is misunderstood as if many droplets form a network. In reality, each droplet contains a complete network.

We thank the reviewer for this note, and we agree that our expression was ambivalent. We have revised this part of the abstract. This part now reads

“Finally, we connect neural and logical computations into a hybrid circuit that recursively partitions a concentration plane according to a decision tree in cell-sized compartments “

3) For me, the significance of majority voting with veto is not clear. Is the correct answer rate 100% when the veto right is given to other than X3?

To summarize

- **We clarified the significance of veto**
- **We gave veto to inputs other than X3**
- **We added a symmetric functionality: majority right**
- **We revamped the figure on Majority voting**

We thank the reviewer for this comment. Majority voting is a central function of Boolean computing, and it is used widely to aggregate information from various sources in ensemble learning, decision making, data communication and storage, or molecular diagnosis. However the sources that vote may have different statistical errors and be prone to systematic bias. In the case of majority voting performed *in vitro*, molecular artifacts may also bias the results. We have added text in SI (section on majority voting) to discuss the possible use of veto right to detect contaminations and prevent false-positives:

"Similarly, giving a veto right to some DNA sequences could mitigate the uncertainty and errors inherent to molecular computations. For instance, in a laboratory setting where DNA amplification techniques like PCR or LAMP are routinely used, the repeated amplifications of nucleic acids can cause massive and permanent contamination of the environment with specific DNA sequences. This was highlighted by reports of researchers developing SARS-Cov-2 nucleic acid tests, and later testing positive during surveillance screening –which are likely to be false-positives that were caused by environmental contamination by amplified DNA fragments of SARS-Cov-2. Obviously, majority voting would also be susceptible to this kind of false-positive if its panel contains genes that are commonly amplified in laboratories around the world (e.g. BRCA1 or BRCA2). However, a contaminated environment could be detected by adding DNA watermarks to PCR tests, which are DNA sequences unique to a laboratory that would be amplified during PCR (for instance by adding watermarks to the 5' tails of the PCR primers). The presence of such watermarks in the environment would be a telltale sign that contamination has occurred and that other DNA sequences are likely to contaminate the environment, which greatly increases the likelihood of false-positives. In turn, giving a veto right in majority voting to these watermark sequences would be a safeguard that prevents false-positives."

In response to this comment, we also performed new experiments. We gave a veto right to two other strands (X7 et X8): the accuracy is in the range of 96-100%, the only errors being near the decision boundary.

We also introduced a new functionality - majority right - that is the symmetric function of veto. It gives an input the power to turn the result ON, independent of the other inputs. For instance, the presence of mutations in genes like BRCA1 is highly predictive of cancer. By giving a majority right to a mutation in this gene, we could improve the robustness of the classifier. We gave this right to 2 inputs and obtained an accuracy in the range of 96%-100%.

The new figure of Majority Voting is as follow

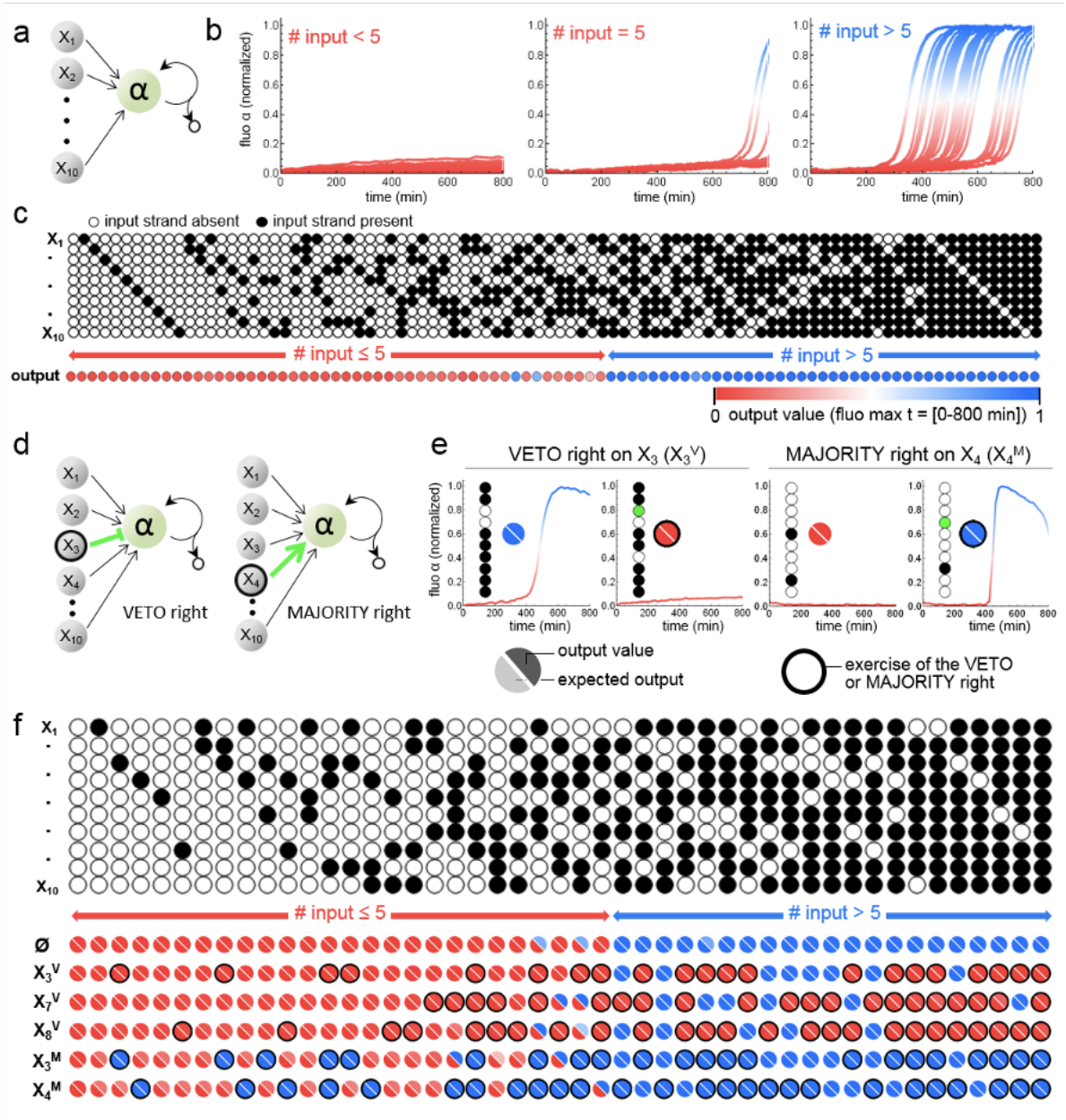


Fig. 3 **Majority functions on 10 bits with a high-dimensional linear classifier**. **a,** Ten input strands X_i (each input being encoded by a strand at concentration 0 or c) are connected by equal-weight converters to a single neuron. The threshold is adjusted so that more than 5 inputs must be present to trigger the replication of α. **b,** Bulk time traces of majority voting for 92 inputs patterns. The color of traces reflects the instantaneous level of α. **c,** Summary of results. The output color corresponds to the maximum fluorescence intensity over 800 minutes of incubation. **d,** Majority vote with veto or majority right. A strand is discretionarily granted a veto right by replacing its positive-weighted converter by a negative-weighted converter (veto right). Similarly, a strand is granted a majority right by increasing the concentration of its positive-weighted converter (green arrows). **e**, Examples of majority voting with or without veto right (left) or majority right (right). **f,** Summary of majority voting on a batch of patterns, and with different rules (democratic, veto right to X_3, X_7 or X_8, majority right to X_3 or X_4).

Lastly, in response to a comment by another reviewer, we have also added a discussion about the theoretical scalability of majority voting for enzymatic networks and DNA-only networks (SI discussion)

4) In the Materials section, please explain why you used a special exonuclease and what makes it different from a commercial one.

We thank the reviewer for this comment as we did not detail this point. The exonuclease works as a "garbage collector' - continuously degrading short DNA strands from their 5' and leaving only monomers. This clean removal of short DNA prevents them from accumulating and poisoning the system. Historically, we developed our system to work in the temperature range of 40-50 °C. This operating temperature was needed to ensure the quick melting of input and output strands from their template. We then built the system around this range, looking for enzymes that operate in this temperature range. As commercial RecJ exonucleases are only stable up to ~37°C, we expressed a thermostable version of RecJ (called ttRecJ) that extends the stability range to 50 °C.

However, it must be noted that the temperature range is not set in stone. Collaborators in Paris needed to operate cells with our enzymatic toolbox. So, they redesigned the templates and lowered the melting temperature of oligonucleotide, which enabled them to operate the enzymatic networks at 37 °C and provide spatial-patterning for cell culture[30]. In principle we could operate our molecular neural networks at 37 °C and below, provided we redesign the strands in the system.

We edited the material section to explain why we use a special exonuclease:

"This is a thermophile variant of RecJ that works in the temperature range (~40-50 °C) for which the templates were designed, enabling fast melting and release of the output strands from the templates. (Note that that this enzymatic framework can be redesigned to work at 37°C, as demonstrated by recent experiments with biological cells)"

References

1. Soloveichik, D., Seelig, G. & Winfree, E. DNA as a universal substrate for chemical kinetics. *PNAS* **107**, 5393–5398 (2010).
2. Srinivas, N., Parkin, J., Seelig, G., Winfree, E. & Soloveichik, D. Enzyme-free nucleic acid dynamical systems. *Science* **358**, (2017).
3. Vasic, M., Chalk, C., Khurshid, S. & Soloveichik, D. Deep Molecular Programming: A Natural Implementation of Binary-Weight ReLU Neural Networks. in *International Conference on Machine Learning* 9701–9711 (PMLR, 2020).
4. Cherry, K. M. & Qian, L. Scaling up molecular pattern recognition with DNA-based winner-take-all neural networks. *Nature* **559**, 370–376 (2018).

5. Qian, L., Winfree, E. & Bruck, J. Neural network computation with DNA strand displacement cascades. *Nature* **475**, 368–372 (2011).
6. Qian, L. & Winfree, E. Scaling Up Digital Circuit Computation with DNA Strand Displacement Cascades. *Science* **332**, 1196–1201 (2011).
7. Zhang, D. Y. Cooperative Hybridization of Oligonucleotides. *J. Am. Chem. Soc.* **133**, 1077–1086 (2011).
8. Genot, A. J., Fujii, T. & Rondelez, Y. Scaling down DNA circuits with competitive neural networks. *Journal of The Royal Society Interface* **10**, 20130212 (2013).
9. Lopez, R., Wang, R. & Seelig, G. A molecular multi-gene classifier for disease diagnostics. *Nature chemistry* **10**, 746–754 (2018).
10. Lecun, Y., Bottou, L., Bengio, Y. & Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **86**, 2278–2324 (1998).
11. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
12. Hao, Y., Li, Q., Fan, C. & Wang, F. Data Storage Based on DNA. *Small Structures* **2**, 2000046 (2021).
13. Salehi, S. A., Liu, X., Riedel, M. D. & Parhi, K. K. Computing Mathematical Functions using DNA via Fractional Coding. *Scientific Reports* **8**, 8312 (2018).
14. Hjelmfelt, A., Weinberger, E. D. & Ross, J. Chemical implementation of neural networks and Turing machines. *PNAS* **88**, 10983–10987 (1991).
15. Hjelmfelt, A., Weinberger, E. D. & Ross, J. Chemical implementation of finite-state machines. *PNAS* **89**, 383–387 (1992).
16. Liu, X. & Parhi, K. K. Molecular and DNA Artificial Neural Networks via Fractional Coding. *IEEE Transactions on Biomedical Circuits and Systems* **14**, 490–503 (2020).
17. Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning*. (MIT Press, 2016).
18. Minsky, M. & Papert, S. A. *Perceptrons: An Introduction to Computational Geometry*. (MIT Press, 2017).
19. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control Signal Systems* **2**, 303–314 (1989).

20. Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* **25**, 1097–1105 (2012).
21. He, K., Zhang, X., Ren, S. & Sun, J. Deep Residual Learning for Image Recognition. in 770–778 (2016).
22. Simonyan, K. & Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [cs]* (2015).
23. Shang, L., Cheng, Y. & Zhao, Y. Emerging Droplet Microfluidics. *Chem. Rev.* **117**, 7964–8040 (2017).
24. Khan, J. *et al.* Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine* **7**, 673–679 (2001).
25. Hall, P., Marron, J. S. & Neeman, A. Geometric Representation of High Dimension, Low Sample Size Data. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **67**, 427–444 (2005).
26. Liu, B., Wei, Y., Zhang, Y. & Yang, Q. Deep neural networks for high dimension, low sample size data. in *Proceedings of the 26th International Joint Conference on Artificial Intelligence* 2287–2293 (AAAI Press, 2017).
27. Ahmed, F. E. Artificial neural networks for diagnosis and survival prediction in colon cancer. *Mol Cancer* **4**, 29 (2005).
28. Lancashire, L. J. *et al.* A validated gene expression profile for detecting clinical outcome in breast cancer using artificial neural networks. *Breast Cancer Res Treat* **120**, 83–93 (2010).
29. McDermott, A. M. *et al.* Identification and Validation of Oncologic miRNA Biomarkers for Luminal A-like Breast Cancer. *PLOS ONE* **9**, e87032 (2014).
30. Van Der Hofstadt, M., Galas, J.-C. & Estevez-Torres, A. Spatiotemporal Patterning of Living Cells with Extracellular DNA Programs. *ACS Nano* **15**, 1741–1752 (2021).

Reviewer Reports on the First Revision:

Referees' comments:

Referee #1 (Remarks to the Author):

I have read the other referee's comments. I remain to feel that this manuscript represents a major advance in the important field of molecular computing systems. The proposed DNA-enzymic perceptron represents a novel strategy to develop molecular neural networks, showing advantages including low leakage, high sensitivity, and dynamic molecular generation and degradation. Nonlinear classification was first experimentally demonstrated. The droplet-based computing and recording workflow also provide new possibilities to evaluate logic or neural computing functions with high parallelism.
Having said these, I agree with the Referee 2 in several aspects that prevent the paper from being accepted in its current version, after reading the comments and more thoroughly the manuscript. In particular, I have reservations on the functional components of perceptron. I suggest the authors to consider the following issues to further improve their work.
a) In figure 1a the authors showed illustrations of the conventional neural network model and the corresponding enzymatic network model. However, they used a mixed mode of the two models, which might be somewhat misleading.
b) As pointed out by Referee 2, weighted sum and nonlinear activation are two basic functions for a perceptron. Reorganization of Figure 1 is thus necessary. In addition, they might want to provide new experiments to describe weighted sum and nonlinear activation functions separately, which would more solidly demonstrate the proposed perceptron.
c) It is suggested to indicate the exact values for w1 and w2 in figure 2a and b.
d) The authors only compared the nonlinearity with that in DNA-only networks. I'm wondering whether the accuracy is better or worse? For linear classifications in figure 2e, please also display the theoretical lines.
e) For figure 2f, if the authors wanted to emphasize that temperature is a programmable parameter, it would be more convincing to provide details on how to set a proper temperature.
f) The authors have not provided full architecture for rectangular function. Whether the output layer is a perceptron is important to evaluate the successful implementation of a multi-layered neural network, given that the authors have changed the last figure as a hybrid network.
g) In Figure 5a, although the authors recorded fluorescence intensity of a and b, only gamma is the output for the network. It would be clearer to use two colors for figure 5d, since functionally it represents the result of binary classification.
i) I suggest the authors to add an applicational demonstration using a multi-layered network, considering that deep feedforward network is a typical machine learning model. It would much more impressive to show the classification of a practical dataset.

Referee #2 (Remarks to the Author):

The revised manuscript has more clarity and the presentation has improved. The reviewer appreciates the authors' efforts with the linear equations and the confidence matrices. My comments on the revised manuscript are listed below.

1. The claim of deep neuromorphic networks is still misleading. What has been implemented is a shallow decision tree (also called classification and regression tree, CART). These correspond to linear decision boundaries.

2. The power of neural networks comes from the nonlinear activation functions. A simple perceptron first computes a weighted sum and then a nonlinear activation function of that

weighted sum. These activation functions could be sigmoid, tangent hyperbolic function, or rectified linear unit (ReLU). No perceptron is implemented in this paper. No multi-layer perceptron (MLP) is implemented in this paper as nonlinear activation functions are not realized. Outputs of hidden layers neurons are not weighted and passed to the output layer.

3. The goal of the paper is a (faster) inference engine implemented via enzymes. From Fig. 2(c), first and third plots, it appears the weights are functions of the threshold (15 nM vs. 18 nM) and the temperature. Given a (linear) equation, it appears that there are infinite ways to program the equations into the enzyme reaction. Is this true? If it is, then how does one select the experimental parameters? What are the tradeoffs? I would assume the authors could develop a theoretical framework that explains the experimental results. This would complete the analysis-synthesis loop where experimental parameters (temperature, concentrations, thresholds) can be obtained from the linear boundary equations.

4. What are the limitations of the current experimental setup and the results that can be achieved currently? Given the authors' claim of advances over prior work, what are the bottlenecks today in this setup and what if achievable would be considered an important advance in future, either with respect to number of inputs, weights, depth of layers etc?

Here are minor comments:

1. As mentioned in the response, inputs X1 and X2 are regenerated during the enzymatic process. If so, do the initial concentrations of the two inputs still matter?

2. In Fig. 1, there are multiple steps/actions even in one layer. What are the time schedules for each step? If there are more layers, will the errors increase?

Referee #3 (Remarks to the Author):

As Reviewer 2 pointed out, a ‘multilayer’ perceptron is one that uses a nonlinear function for the activation function. Therefore, it is not appropriate to use the name for the network in this paper, which uses a threshold function for the activation function, as a multilayer perceptron. I would like to suggest the authors to use a more appropriate name.

The enzymatic network proposed in this paper allows for a simpler and modular design compared to conventional reaction systems that use non-enzymatic DNA logic gates. In addition, the proposed droplet-based visualization of the entire input space allows us to exhaustively verify the behavior of the system. With these methods in place, the technical barriers to extending their network to a multilayer perceptron have been removed. Namely, this paper has a high value as a novel implementation method of a scalable chemical reaction system. It will be a milestone towards the realization of practical chemical artificial intelligence.

From a theorist’s point of view, it may seem that one of the infinite combinations of possible reaction systems has been chosen, but from an experimentalist's point of view, the requirement of Reviewer 2 to derive a general design theory is clearly excessive. It is rare for a reaction system based on chemical reactions to behave as the designer intended due to various factors that cannot be fully modeled, and it is of great value to have a guaranteed system that reacts as intended within a certain input space.

As pointed out by Reviewer 2, the discussion on the limitations of the proposed method is certainly of interest to the reader and should be included in the Discussion.

nature portfolio

Table of Contents

Referee #1

I have read the other referee's comments. I remain to feel that this manuscript represents a major advance in the important field of molecular computing systems. The proposed DNA-enzymic perceptron represents a novel strategy to develop molecular neural networks, showing advantages including low leakage, high sensitivity, and dynamic molecular generation and degradation. Nonlinear classification was first experimentally demonstrated. The droplet-based computing and recording workflow also provide new possibilities to evaluate logic or neural computing functions with high parallelism.

Having said these, I agree with the Referee 2 in several aspects that prevent the paper from being accepted in its current version, after reading the comments and more thoroughly the manuscript. In particular, I have reservations on the functional components of perceptron. I suggest the authors to consider the following issues to further improve their work.

We warmly thank the referee for their careful reading and appraisal of the manuscript. To summarize, we have made the following changes, keeping in mind the need to keep the manuscript short and concise:

- We have added a theoretical framework
- We have performed new experiments
- We have performed a new analysis
- We have clarified the text, and added a section in SI to more clearly state what our networks can and cannot do
- We have shortened the text and moved technical discussion to the SI
- We have changed the title
- We have cut down the number of references to 50 in the paper
- We have completely redrawn figure1, adding the new experiments
- We have moved non-experimental extended data figures to the SI

We also clarified how we tune the weights. We use decoy templates, which were mentioned in Extended Figure 5 (now moved Figure S2) and the Supplementary Information, but which were not clearly presented in the main paper due to space constraint. We have now improved this, presenting these decoy templates in Figure 1.

a) In figure 1a the authors showed illustrations of the conventional neural network model and the corresponding enzymatic network model. However, they used a mixed mode of the two models, which might be somewhat misleading.

b) As pointed out by Referee 2, weighted sum and nonlinear activation are two basic functions for a perceptron. Reorganization of Figure 1 is thus necessary. In addition, they might want to provide new experiments to describe weighted sum and nonlinear activation functions separately, which would more solidly demonstrate the proposed perceptron.

We thank the reviewer for those two comments. We have completely reorganized Figure 1. We now start from the general abstraction (a multilayer network) and we progressively decompose it into an artificial neuron, then into its enzymatic implementation. We provide new experiments to demonstrate separately weight tuning, weighted summation, and the application of the step function.

The figure is as below:

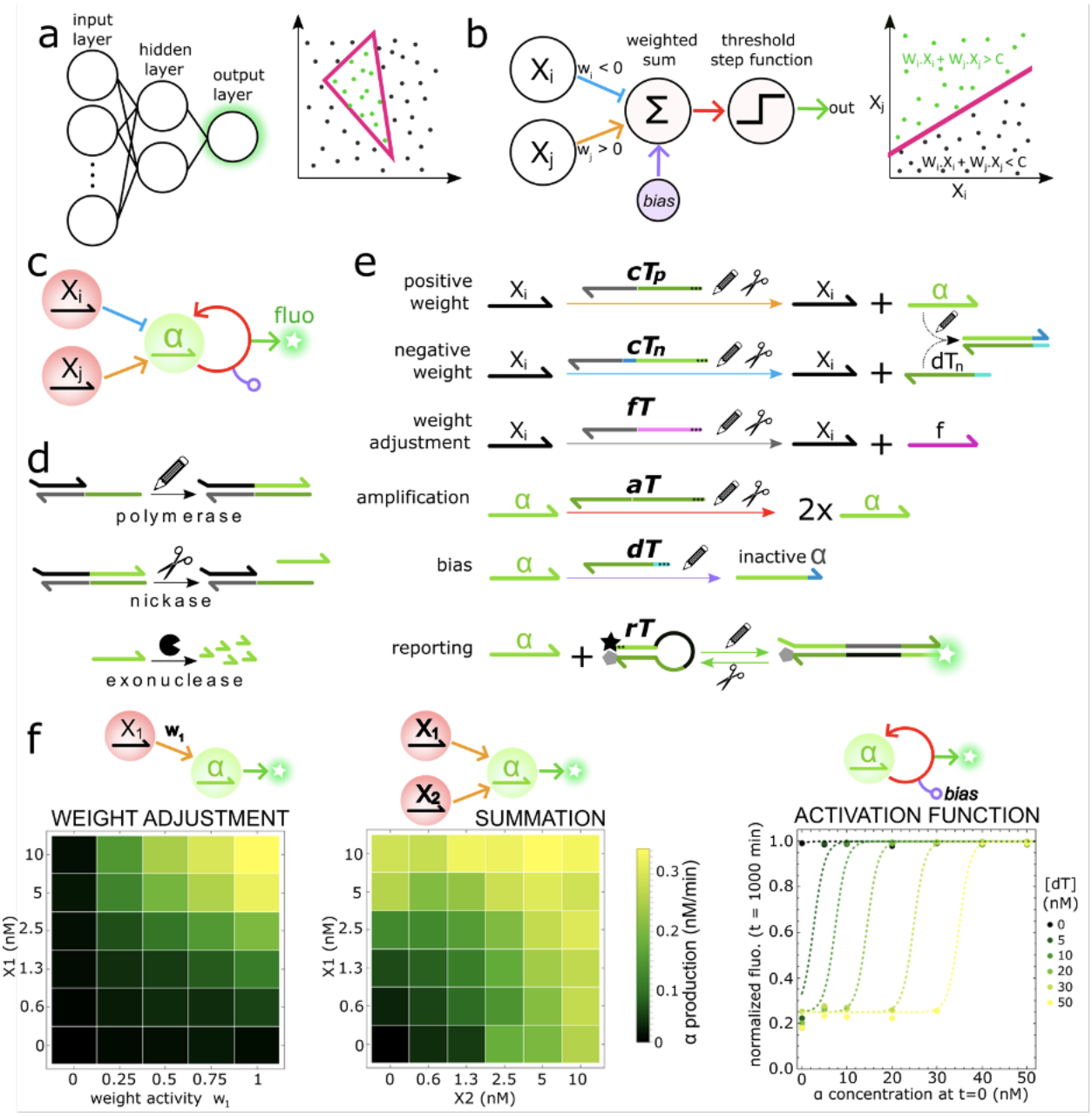


Fig. 1 **Architecture of DNA-encoded enzymatic neural networks. a,** Multilayer neural networks can classify nonlinearly separable regions. **b**, Our individual neuron computes a weighted sum on its inputs and generates an output if the sum exceeds a threshold (linear classification). **c**, Chemical architecture of the neuron. The autocatalytic amplification of the output strand α (red arrow) is triggered when the weighted activation (blue and orange) by input strands X_i and X_j overcomes the thresholding mechanism (purple). **d**, The chemical neuron is powered by three enzymes producing (polymerase), cutting (nickase) and degrading (exonuclease) DNA. **e**, Building blocks of the enzymatic neural networks. Positive and negative weights are computed by converter templates cT_p and cT_n. They produce species α or dT_n whose steady state concentration is proportional to the input X_i. Weights can be independently tuned with fake templates (fT) that compete with cT for the inputs. The activation function -a step function - is encoded in a bistable switch composed of an amplification template (aT, which replicates the species α) and a drain template (dT, which deactivates α and controls the bias). α concentration is monitored using a reporter template (rT).

f, Experimental validation of the basic components: weight adjustment on a single input (left), weighted summation on two inputs (w_1 = 0.5, w_2 = 1) (middle) and application of the step function on α (right). Full traces are available in Extended Data Figure 1.

c) It is suggested to indicate the exact values for w1 and w2 in figure 2a and b.
We have added the fitted weights to those panels. (Of note, fitting bulk data is more difficult than with droplets, because they are many less data points. This makes fitting a bit delicate in the case of the negative-weighted classifier, whose boundary is slightly curved near the origin)

d) The authors only compared the nonlinearity with that in DNA-only networks. I'm wondering whether the accuracy is better or worse?

This is a great suggestion and we initially thought about doing it. But in general it is delicate to compare two classifiers just based on their confusion matrices, because the matrices are sensitive to the composition of the inputs panel. Actually, we had already computed for ourselves the confusion matrix of (Lopez et al., Nature Chemistry, 2018). We found that the matrix was much more sensitive to measurement errors than the fitting of the nonlinearities, and at the end we had decided not to include it in the paper.

For reference, we reproduce below this confusion matrix of Lopez *et al.* We tried two methods to compute it. First, we measured the fluorescence shift in Figure 3.d in ImageJ (which is the method that we used to extract the shape of the nonlinearities). This manual measurement of distance typically incurs a ~10% error on the absolute fluorescence shift, and an error of ~20-30% on the normalized fluorescence. While this error is unlikely to affect the fitting of nonlinearities much, it easily changes the OFF/ON classification of points whose end fluorescence is borderline (actually with this method we could not reproduce the classification of Fig. 3e). So we tried a slightly more robust method for computing the confusion matrix. We directly looked at the colour of each point in fig.3e, and manually assigned a ON/OFF state to each point based on its colour: a ON state if it is blueish, or an OFF state if it is reddish. The results are shown below:

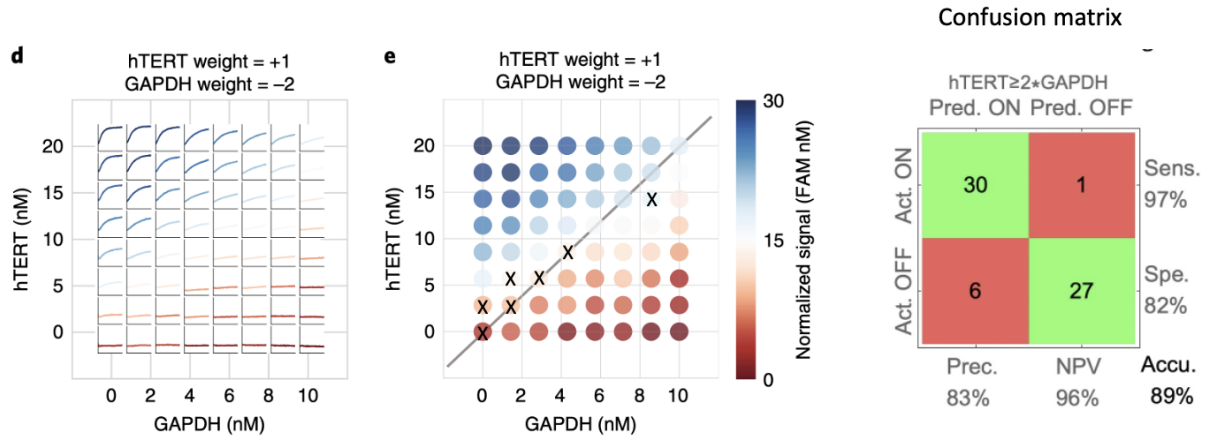


Figure: Confusion matrix of (Lopez et al., Nature Chemistry, 2018). Panels **d** and **e** are extracted from Figure 3. In panel **e**, we marked erroneous computations with a cross. The resulting confusion matrix is show in the right

At face value, we can claim that all metrics (except Negative Predictive Value) are better for our classifier by ~5-10 percentage points. However, one must be careful with this interpretation. First, this confusion matrix is highly sensitive to measurement errors. Many inputs located near the border are close to the threshold. A small error on the readout can completely change their ON or OFF attribution, and sway the metrics by a few percentage points. Thirdly, the confusion matrix loses all information about the strength of nonlinearities, since we are actually doing the nonlinear computation ourselves by attributing a OFF or ON state to a point. (If you take a classifier which perfectly sums its inputs, but does not apply any activation function, it would still score 100% on all metrics). Therefore, we decided that the confusion matrix was not a fair way to accurately compare our classifier with that of Seelig, and we did not include it in the paper.

Parenthetically, similar issues arise when comparing diagnostics tests made by different manufacturers (a diagnostic is also a molecular classification). Manufacturers often report the specificity and sensitivity of their test (thus a percentage), but those numbers highly depend on the composition of the patient panel, which makes it difficult to compare tests between them. A more accurate metric to compare tests is the limit of detection, which is much more robustly defined, and routinely used by the FDA as a benchmark.

Regarding the classifier of Cherry and Qian, their aim was not to implement a linear classifier, and so it is even more delicate to benchmark their classifier. Again, the closest thing to a linear classifier in their paper is the Figure 2a of (Cherry and Qian, Nature, 2018), but no weighted summation is performed (only thresholding), so we believe it would not be a fair comparison.

For linear classifications in figure 2e, please also display the theoretical lines.
Unless we misunderstood this comment, the fitted lines are shown in red in figure 2e.

e) For figure 2f, if the authors wanted to emphasize that temperature is a programmable parameter, it would be more convincing to provide details on how to set a proper temperature.

We thank the reviewer for this constructive comment, which is related to a question by another reviewer. In response we added a discussion on temperature in the SI. Put briefly, predicting the optimal temperature from scratch is clearly difficult, and we advocate an empirical approach to pinpoint the optimal thermal regime. This direct mapping of temperature was one of the motivations for developing our temperature platform. We have added the following text to the SI:

> Temperature will also be an important parameter to tune. Its effect on computation is difficult to model because temperature influences all chemical reactions, and we expect each network to have a slightly different optimal temperature. In general, enzymatic activity increases with temperature - which should speed up the computation - but in a slightly different way for each enzyme in our toolbox (polymerase, nickase, exonuclease). And our networks are made of many antagonist chemical reactions (some reactions promote the production of DNA, while others promote its removal), so the overall balance is difficult to predict for each network. Additionally, enzymes degrade at high temperatures, further complicating the picture.
>
> The complex effect of temperature motivated the development of the thermal gradient platform, which allowed us to quickly find the optimal temperature in a range of~10-15 °C. At the moment, we use spatial readout to find the temperature of a droplet (i.e. we measure its position in the gradient with respect to the temperature sensors). But this would not be compatible with end-point sequencing, which obviously loses information about the position of droplets in an array. The temperature of a droplet could be recorded at the time of incubation with various means, for instance with a DNA thermometer[39–41] or auxiliary enzymatic network that “remembers” the incubation temperature[42], or using top-down methods that write information directly in droplets during incubation (for instance with UV light and photocontrol[43,44]).

f) The authors have not provided full architecture for rectangular function. Whether the output layer is a perceptron is important to evaluate the successful implementation of a multi-layered neural network, given that the authors have changed the last figure as a hybrid network.

This is true, and we have now added the full architecture, which is now available as Figure S3.

g) In Figure 5a, although the authors recorded fluorescence intensity of a and b, only gamma is the output for the network. It would be clearer to use two colors for figure 5d, since functionally it represents the result of binary classification.

We thank the reviewer. The results of each binary classification are already shown in fig. 5c. In fig. 5d, we merge the results to show the full classification. We believe it would be slightly more confusing if we used only gamma for fig. 5d, as this plot is meant to show the final results.

i) I suggest the authors to add an applicational demonstration using a multi-layered network, considering that deep feedforward network is a typical machine learning model. It would be much more impressive to show the classification of a practical dataset.

We thank the reviewer for this comment. In the future, it will be important to apply our enzymatic networks to practical datasets, yet we feel that it would be out of the scope of this paper. Real datasets (like MNIST, CIFAR or ImageNet) typically contain at least ~50,000 examples. It would take significant time (much longer than the allotted time for revision) to design, test and debug enzymatic networks for those datasets, and at the end we would only be able to test a miniscule fraction of this dataset.

Cherry and Qian did test their DNA network on MNIST - a laudable tour de force - but it was on a miniscule fraction, 0.1% of the dataset (~100 examples out of ~60,000). And to do so, they had to binarize and downsample the images, turning them from 6272 bits pattern (28x28 pixels in 8 bits) into 100-bit patterns (10x10 pixel in 1 bit). This drastic downscaling worked because MNIST is a peculiar dataset whose digits can be recognized by looking at a few pixels, but it is unclear if it will generalize to more realistic and practical datasets, like CIFAR or even ImageNet. And this tour de force still required an acoustic liquid handler, a machine which costs ~$250,000 and is out of reach of most labs.

We believe that it would be a better use of our resources to develop a general method to handle and test large datasets, using technologies that could be used by most labs in the field of DNA computing. As we now discuss now in SI, we propose to leverage the expected exponential drop in the cost of DNA synthesis, combined with the massive parallelism of microfluidics to screen a large number of inputs and network topologies. This will obviously take a few more years to develop, but would yield a powerful and general technique applicable to a wide variety of problems in DNA computing.

Referee #2

The revised manuscript has more clarity and the presentation has improved. The reviewer appreciates the authors' efforts with the linear equations and the confidence matrices. My comments on the revised manuscript are listed below.

We warmly thank the reviewer for their suggestions which vastly improved the manuscript. In response to the reviewers, we have made the following the changes, keeping in mind the need to keep the manuscript short and concise.

- We have added a theoretical framework
- We have performed new experiments
- We have performed a new analysis

- We have clarified the text, and added a section in SI to more clearly state what our networks can and cannot do
- We have shortened the text and moved technical discussion to the SI
- We have changed the title
- We have cut down the number of references to 50 in the paper
- We have completely redrawn figure1, adding the new experiments
- We have moved non-experimental extended data figures to the SI

We also clarified how we tune the weights. We use decoy templates, which were mentioned in Extended Figure 5 (now moved Figure S2) and the Supplementary Information, but which were not clearly presented in the main paper due to space constraint. We have now improved this, presenting these decoy templates in Figure 1.

1. The claim of deep neuromorphic networks is still misleading.

We thank the reviewer for this comment. We used the historical definition of “deep” because it is mathematically well-defined (two layers or more). But we agree that the colloquial meaning of “deep” could be ambiguous for readers. For the avoidance of doubt, we have replaced “deep” by “multilayer” where it clarifies meaning.

We disagree that the claim of “neuromorphic” is misleading. One may argue about whether our networks are perceptrons in the modern sense, but one may difficultly argue that they are shaped after neuronal architectures, which is literally the meaning of “neuromorphic”. But we agree that the term “neuromorphic” is charged because it has come to mostly refer to a special type of computation in electrical engineering (such as spiking networks). So we propose to simply refer to our networks as “enzymatic neural networks”. We have changed the title and text accordingly.

What has been implemented is a shallow decision tree (also called classification and regression tree, CART).

Decision trees and multilayer neural networks are not mutually exclusive: we compute a decision tree using a multilayer neural network in Figure 5. We chose decision trees to broaden the appeal of our paper to the wide readership of Nature because they are widely used in clinical settings to make decisions.

These correspond to linear decision boundaries.

Each neuron in our network individually computes a linear decision boundary. But when those neurons are connected in a network, they collectively compute nonlinear decision boundaries, i.e. regions of space that cannot be linearly separated. In response to a previous comment by this reviewer, we had previously fitted a single-layer perceptron and a multi-layer perceptron to the data of Figure 5. Only the latter fits well the data, which shows that the decision boundaries computed by the whole network are not linearly separable (otherwise they would be captured by the single-layer perceptron). The new experiment we had provided in Figure 4 also shows boundaries in 1D that are clearly nonlinear.

2. The power of neural networks comes from the nonlinear activation functions. A simple perceptron first computes a weighted sum and then a nonlinear activation function of that weighted sum.

We thank the reviewer for this comment. Our enzymatic neuron computes a weighted sum of its inputs, and then applies a step function on that weighted sum. This is exactly the perceptron of Rosenblatt in his 1958 paper "The perceptron: A probabilistic model for information storage and organization in the brain". We agree that this is a particular case of modern perceptrons, which are understood to apply a variety of nonlinear activation functions on their weighted sums. We clarify this point in the revised manuscript (see below).

These activation functions could be sigmoid, tangent hyperbolic function, or rectified linear unit

Step functions are adapted for the binary classifications that we perform in the paper, which are aimed at molecular diagnostics. But we fully agree with the reviewer, and we do not want to belittle the formidable challenge of computing arbitrary activation functions. As pointed out by (Salehi et al., Scientific Report, 2018), this question has only recently received attention

> *No prior publication has considered molecular ANNs using a sigmoid activation function.*

We now explicitly acknowledge that we implement one activation function among many. Implementing other types of activation functions will be important in the future. To that end, we note that Systems biologists have long known sigmoidal responses to be feasible with enzymatic networks, using cooperativity or push-pull networks. We also point to a recent paper by the Weiss group that describes how to implement ReLU with chemical production, degradation and sequestration. Those 3 chemical primitives are feasible in our enzymatic framework, and so in principle it should be possible to implement ReLU in the future (see summary of changes below).

No perceptron is implemented in this paper.

We would not go as far as saying we can implement a perceptron with **any** nonlinear activation function, but we would not say we implemented **no** perceptron either. In Figure 2 and 3, we implemented a chemical system that takes a weighted sum of its inputs, and triggers a response if this weighted sum exceeds a threshold. This is again exactly the perceptron defined by Rosenblatt in his 1958 paper.

No multi-layer perceptron (MLP) is implemented in this paper

This reviewer previously commented that**:**

> The data generated by the experiment could be fit into a multilayer perceptron to see the difference between the multilayer perceptron and the nonlinear enzymatic network.

In response to this, we had fitted a single-layer perceptron and a multilayer perceptron (Figure. 5 e). The single-layer perceptron poorly captures the nonlinearly separable regions, while the multilayer perceptron provides an excellent fit. These fits - suggested by this reviewer - show that,

if anything, our networks are much closer to being multilayer perceptrons than being single-layer perceptrons.

as nonlinear activation functions are not realized.
Following this comment and another one by reviewer 1, we have completely redrawn and reorganized Figure 1. We now carefully decompose weighted sum and activation, and we show individual experiments for each operation. When the weighted summation is performed without the nonlinear module, it is not sharp, exactly as expected. This shows that the full neuron does apply a nonlinear activation function.

****
Action: In response, we have taken several actions:

>We have revised the paragraph introducing our enzymatic neuron. We explicitly state that it is modelled after the perceptron defined by Rosenblatt in 1958 (which was already cited, but not explicitly mentioned in the previous version of the manuscript). We also limit our claim by stating that our enzymatic neuron corresponds to one special class of modern perceptrons, namely the perceptrons with step function as the activation function. We have also reworked the paragraph to make it clearer. It now reads:

> Our neuromorphic networks are built around a generic enzymatic neuron (Fig. 1) that emulates the perceptron proposed by Rosenblatt in 1958[27]. The neuron takes DNA or RNA strands as input. The state of the neuron is encoded by the concentration of a short DNA strand α (the signal sent by the neuron). The neuron computes a weighted sum of its inputs thanks to converter templates. They act like programmable-gain amplifiers in analog electronics[28], the gain being tuned by the composition of the templates (Fig. 1f). The neuron then takes an ON state (yielding a high concentration of α) if the weighted sum of inputs exceeds a concentration threshold, and remains OFF otherwise (low concentration of α). In modern terms, this mimics a perceptron with a step function as the nonlinear activation function. The inputs here are two DNA analogues of the miRNAs miR-21 and miR-31, which are involved in cancer[29]. The network is modular and can easily be rewired to accept different inputs or produce new outputs.

>We have removed the paragraph in the introduction that described general results in machine learning (Papertand Minsky theorem on XOR, Cybenko theorem on two-layer network, various references to modern deep learning). This shortens the manuscript (which was too long), makes it more focused, and avoids instilling doubts about whether our neurons are perceptrons in the modern sense (i.e. they can compute arbitrary activation functions). We hope that it will help to delineate our claim and avoid misunderstanding.

>We have reworked the discussion to explicitly acknowledge the limitations of our networks. We acknowledge that we compute one possible activation function among many, and we point to (Salehi et al., 2018, Sci Rep.) for the general question of computing arbitrary activation functions.

We also note that sigmoidal and ReLU functions are known to be feasible in Systems biology with simple reaction schemes.

> Our enzymatic neural networks bring tangible benefits over nonenzymatic ones, namely speed of operation, compactness of network, composition of computations, sharpness of decision margins, sensitivity of detection, correction of errors, and weighing of analog variables with programmable-gain enzymatic amplification (Supplementary Information Section 4). Yet developments will be needed to match the modern form of perceptrons, which apply a wider variety of activation functions than a step function, such as the sigmoid function, for making soft decisions and predicting probabilities[4], or the Rectified Linear Unit (ReLU) to ease training. Sigmoidal responses have long been known in systems biology to be feasible with enzymatic networks (e.g. with cooperativity or push/pull motifs[38]), and simple chemical schemes to compute ReLU were recently proposed[39,40].

>Doing so, we removed from the discussion the paragraph on power consumption and evolution of networks due to space constraints (but we keep the section in SI on power consumptions)

>We have completely redrawn figure 1 and added next experiments following comments from reviewer 1. We now clearly decompose our neuron into weighted summation and activation, and added experiments to show those individual operations.

Regarding the last part of comment 2 by this reviewer:

Outputs of hidden layers neurons are not weighted and passed to the output layer.
Reviewer 1 made a similar comment in the previous round of review:

> "Lacking weighted summation and autocatalytic activation, this layer performed similar to a reporting device rather than a neuron."

In response to this, we had performed a whole new set of experiments, where we computed a rectangular function using 3 neurons *with identical chemistry*. This was shown in Figure 4 in the revised manuscript. The last neuron uses exactly the same chemistry as the hidden neuron, and so it weighs its incoming signals in the same way that the hidden neurons weigh the input of the network. This being said, we acknowledge that it was not obvious from the simplified schematic of Figure 4, and reviewer 1 asked us to depict the full architecture in the Extended Data, which we now do in the revised version of the manuscript.

>We have now added computations (section S2 in the SI) showing how weighing of analog concentrations naturally emerges in our network from the production and degradation of DNA, and the use of decoy templates. We also added a discussion (section S4 in SI) explaining how this weighing is a notable difference with nonenzymatic networks, where amplification with a fixed gain is more delicate to do because of the lack of a simple mechanism to remove DNA strands.

3. The goal of the paper is a (faster) inference engine implemented via enzymes. From Fig. 2(c), first and third plots, it appears the weights are functions of the threshold (15 nM vs. 18 nM) and the temperature. Given a (linear) equation, it appears that there are infinite ways to program the equations into the enzyme reaction. Is this true?

This is absolutely true, and in principle there is more than one way to chemically implement a given linear equation.

If it is, then how does one select the experimental parameters? What are the tradeoffs?

This is an excellent point, and in response to comment by this reviewer and reviewer 1, we added discussion in the SI on tradeoffs.

At a high level, tradeoffs are similar in spirit to in-silico neural networks. For a given training set, there are many combinations of weights and biases that give a satisfactory answer. In our case, the selection of parameters will be guided by experimental consideration. We favour small weights (i.e DNA templates with small concentrations), because this reduces the load on the enzymatic machinery and the opportunities for unwanted interactions between strands. When the networks get larger, we may even want to force many weights to be null (sparsification) to reduce synthesis costs. This can be achieved during in-silico training with L1 or L2 regularization, a simple procedure, which forces the weights to be small (L2 norm), or even null (L1 norm).

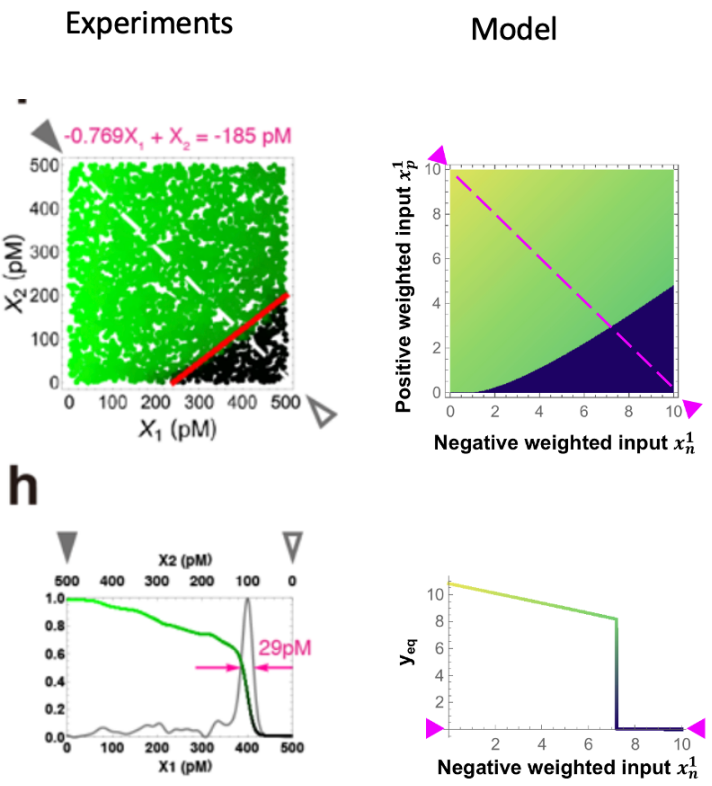
Yet there is a tradeoff between sparse networks (few connections) and dense networks (many connections). A sparse network reduces the enzymatic load and the cost of DNA synthesis. But it increases reliance on a few DNA strands, which could make it less robust experimentally (for instance, in pooled DNA synthesis, it is not uncommon for some strands to be absent from the pool). On the other hand, a dense network distributes its computation on many different strands, and it should improve its robustness to errors and failure from a single strand. But it could overload the enzymatic machinery and increase unwanted interactions between stands with similar sequences. Obviously, we plan to study this kind of tradeoff in future studies.

Temperature will also be an important experimental parameter, as was also noted by reviewer 1. The effect of temperature on computation is difficult to model because temperature influences all chemical reactions, and we expect each network to have a slightly different optimal temperature. In general, enzymatic activity increases with temperature - which should speed up the computation - but in a slightly different way for each enzyme in our toolbox (polymerase, nickase, exonuclease) . And our networks are made of many antagonist chemical reactions (some reactions promote the production of DNA, while others promote its removal), so the overall balance is difficult to predict. Additionally, enzymes degrade at high temperatures, further complicating the picture. The complex effect of temperature motivated us to develop the thermal platform, which allowed us to quickly find the optimal temperature.

I would assume the authors could develop a theoretical framework that explains the experimental results. This would complete the analysis-synthesis loop where experimental parameters (temperature, concentrations, thresholds) can be obtained from the linear boundary equations.

This is an excellent point. We have added a model to rationalize our experimental findings (SI S2). Briefly, we model the neuron with two endogenous pathways (autocatalytic production and degradation by the exonuclease) and two exogenous pathways (i.e. controlled by the inputs) that act like constant chemical source and sink. We obtain an equation governing the dynamic of the neuron, which we study qualitatively and numerically to understand how the steady state selected by the neuron depends on the concentration of inputs. Satisfyingly, this model captures the weighting of signals by converter templates, the shape of the step function, the linear border for positive-weighted classifier, as well as the slanted shape of the step function in the ON region.

For information, we show below on the left the experimental data for one negative-weighted classifier (Figure2 in the manuscript), and on the right the prediction from the modek (Figure S11 of SI). The model captures the curving of the border near the origin, which emerges when the drain removal pathway is not fully saturated (see S2 for details)



Given the power of the model, we find that previous hypotheses that we had formulated were not necessary anymore to explain our findings. So, we removed some texts (which also helped to trim down the number of words).

We remove the following sentence

> When concentrations are low, this inhibition is linearized, implementing a negative weight on the input.

because it was unclear. We replaced it with a reference to the model in SI.

We also removed the following hypothesis

> Qualitatively, this shape is explained by the doubly catalytic activity of the input X1, when it is negatively weighted. Each X1 input catalyzes the production of several drains, and each drain catalyzes the inactivation of several triggers. The total amount of drain varies in the negative-weighted classifier, while it is constant in the positive-weighted classifier.

We also removed the following hypothesis (which invoked enzymatic saturation and competition).

> If the concentration of X1 or drain is too large (compared to their corresponding Michaelis-Menten constant Km), then the Michaelis-Menten equation for the production of drain by X1, or the equation for the inactivation of the trigger by the drain, cease to be linear, and the boundary is expected to deviate from linearity.

We removed the following hypothesis which we invoked to explain the drift in fluorescence

> First the drain recruits more polymerases for inactivating α, which mechanically reduces the polymerase that is free to activate α (through recruitment by the autocatalytic template).

Now we must add a caveat: the goal of this model is to explain - not predict - what we see experimentally. As mentioned by reviewer 3, building a fully predictive model is a herculean task that is quickly defeated by experimental intricacies. That is why we do not seek to fit the parameters of the model. In our experience, such fitting is challenging and ill-defined, because there are many degrees of freedom, and they combinatorially explode as the system grows in size. This shortcoming of modelling motivated the development of our microfluidic platform, as a way to actually see -rather than guess - the behaviour of the system.

4. What are the limitations of the current experimental setup and the results that can be achieved currently**? Given the authors' claim of advances over prior work, what are the bottlenecks today in this setup and what if achievable would be considered an important advance in future, either with respect to the number of inputs, weights, depth of layers etc?**

We thank the reviewer. In the discussion and SI of the paper, we now summarize the advantage of our networks over nonenzymatic ones (e.g. S4 in SI), and we explicitly acknowledge their limitations (e.g. activation function limited to step function) and bottlenecks toward reaching larger and deeper networks. We discuss what is feasible with the current setup (molecular diagnosis),

and what could be achieved assuming bottlenecks are overcome (processing of practical image datasets).

The discussion now reads as follows:

> Our enzymatic neural networks bring tangible benefits over nonenzymatic ones, namely speed of operation, compactness of network, composition of computations, sharpness of decision margins, sensitivity of detection, correction of errors, and weighing of analog variables with programmable-gain enzymatic amplification (Supplementary Information Section 4). Yet developments will be needed to match the modern form of perceptrons, which apply a wider variety of activation functions than a step function, such as the sigmoid function, for making soft decisions and predicting probabilities[4], or the Rectified Linear Unit (ReLU) to ease training. Sigmoidal responses have long been known in systems biology to be feasible with enzymatic networks (e.g. with cooperativity or push/pull motifs[38]), and simple chemical schemes to compute ReLU were recently proposed[39,40].
>
> More generally, the scale of our networks is sufficient for molecular diagnosis (see below), but work will be needed to reach the scale of *in-silico* machine learning, where neural nets typically have dozens of layers and millions of weights, and are trained on datasets with tens of thousands of examples. In principle, enzymatic networks and droplet microfluidics can handle these scales (as evidenced by the intricate computations performed by gene regulatory networks in cells, or by consortium of single celled organisms), but the current tools for writing, handling, and reading DNA would struggle (Supplementary Information Section 6). However, these hurdles could be overcome by an exponential drop in the cost of DNA synthesis - which is expected in the coming years in response to fields that make heavy use of synthetic DNA. In the long term, enzymatic neural networks could empower the nascent field of DNA data storage[41]. Large amount of data could be stored in DNA databases, and queried, labelled or processed in a massively parallel fashion with enzymatic neural networks and droplet microfluidics.
>
> In the short term, neuromorphic computation could readily find applications in molecular diagnostic. Our enzymatic toolbox previously detected a tumor suppressor miRNA in total RNA from human colon with high specificity and sensitivity[42], and our enzymatic neural networks are similar in size to *in-silico* neural networks that reliably diagnose breast tumors[43] or prognose metastasis[44] from multiple molecular clues. This suggests that cancerous patients could be monitored at the point-of-care, using enzymatic neural networks that make diagnosis or prognosis from a panel of miRNAs present in liquid biopsies.

In addition, Section S6 in the SI discusses at lengths the bottlenecks, tradeoffs and future potential of our networks. As we see it, the main bottleneck is currently DNA synthesis. We discuss the scales that could be reached once this bottleneck is overcome, in terms of inputs, layers, weights or number of examples.

Here are minor comments:

1. As mentioned in the response, inputs X1 and X2 are regenerated during the enzymatic process. If so, do the initial concentrations of the two inputs still matter?

This is an excellent point that was not clearly explained in the previous version. As we had explained in Extended Data Figure 5 (now Figure S2), we use decoy templates (called fake templates) which compete with the real converter template for the input. Those fake templates bind to the same inputs as the real templates, but produce an inert species – thus lowering the effective activity of the input. This allows the activity to be changed by modulating the concentration of fake and real templates. We realize that it was not clearly explained in the main paper so far, and we have taken several steps to correct it.

>We now explicitly show the role fake templates in figure 1 (“weigh adjustment” reaction), and we added one experiment in Figure 1 showing their usage. We also say in the text that the weights can be tuned by changing the composition of the template. Lastly, we explicitly model the effect of fake template in our new theoretical framework (SI S2).

2. In Fig. 1, there are multiple steps/actions even in one layer. What are the time schedules for each step? If there are more layers, will the errors increase?

>We thank the reviewer for this important question, which broaches important theoretical and experimental issues. In response we have added a full section in the SI to address this question (new section “Synchronous and asynchronous computations” in S6 in SI)

Briefly, we expect errors to increase if there are more layers. This could be mitigated by adjusting the concentration of neurons, individually or in batch, using for instance our microfluidics platform. But for large and deep networks, we will likely need bottom-up mechanisms (clock, timer, delay gate...) or top-down control (e.g. chemical, optical, thermal signals..) to enforce synchronization between layers.

In the long term, we believe that we should draw inspiration again from neuronal and genetic networks, which operate with clockwork regularity at the ensemble level, in spite of being made of asynchronous and stochastic components. We discuss enzymatic spiking networks, whose neurons would be made after an oscillator rather than a toggle switch. We refer the reviewer to the SI for more details.

Referee #3

We thank the reviewer for their comments. We have made the following the changes, keeping in mind the need to keep the manuscript short and concise.

- We have added a theoretical framework
- We have performed new experiments
- We have performed a new analysis
- We have clarified the text, and added a section in SI to more clearly state what our networks can and cannot do
- We have shortened the text and moved technical discussion to the SI
- We have changed the title
- We have cut down the number of references to 50 in the paper
- We have completely redrawn figure1, adding the new experiments
- We have moved non-experimental extended data figures to the SI

As Reviewer 2 pointed out, a 'multilayer' perceptron is one that uses a nonlinear function for the activation function. Therefore, it is not appropriate to use the name for the network in this paper, which uses a threshold function for the activation function, as a multilayer perceptron. I would like to suggest the authors to use a more appropriate name.

We agree that terminology is important to avoid ambiguity and confusion. To dispel confusion, we now refer to our network as "enzymatic neural networks", which retains their neuronal inspiration, without implying that they are the same thing as modern perceptrons with arbitrary activation function. We have changed the title and the text accordingly.

The enzymatic network proposed in this paper allows for a simpler and modular design compared to conventional reaction systems that use non-enzymatic DNA logic gates. In addition, the proposed droplet-based visualization of the entire input space allows us to exhaustively verify the behavior of the system. With these methods in place, the technical barriers to extending their network to a multilayer perceptron have been removed. Namely, this paper has a high value as a novel implementation method of a scalable chemical reaction system. It will be a milestone towards the realization of practical chemical artificial intelligence.

We thank the reviewer for those warm words. We are highly grateful for their appreciation of our work, which is the culmination of almost 10 years of developing the enzymatic chemistry and droplet microfluidics.

From a theorist's point of view, it may seem that one of the infinite combinations of possible reaction systems has been chosen, but from an experimentalist's point of view, the requirement of Reviewer 2 to derive a general design theory is clearly excessive. It is rare for a reaction system based on chemical reactions to behave as the designer intended due to various factors that cannot be fully modeled, and it is of great value to have a guaranteed system that reacts as intended within a certain input space.

We thank the reviewer for their nuanced comment. We also believe that theoretical models are valuable to guide the design and understanding of molecular systems, but they quickly lose their

predictive power when confronted with reality. That is why we developed the microfluidic machinery, to actually see -rather than guess - how molecular programs behave over their input and parameter space. In the future, we aim to follow a hybrid approach, combining massive data collection with in-silicon learning to predict and design complex networks with high confidence.

This being said, in the end we included a coarse-grained model to rationalize our experimental findings. It explains features like the slightly curved border for the negative-weighted classifier. However, we do not expect it to predict the behaviour of the system from scratch, for the reasons cogently stated by this reviewer.

As pointed out by Reviewer 2, the discussion on the limitations of the proposed method is certainly of interest to the reader and should be included in the Discussion.

We thank the reviewer for this point and copy here the reply we made to reviewer 2.

We summarize in the discussion and in the SI (section S4) the benefits and limitations of our networks (e.g. activation function limited to step function). We also discuss the bottlenecks toward reaching larger and deeper networks. Lastly, we discuss what is feasible with the current setup (molecular diagnosis), and what could be achieved assuming bottlenecks are overcome.

The discussion now reads as follows:

> Our enzymatic neural networks bring tangible benefits over nonenzymatic ones, namely speed of operation, compactness of network, composition of computations, sharpness of decision margins, sensitivity of detection, correction of errors, and weighing of analog variables with programmable-gain enzymatic amplification (Supplementary Information Section 4). Yet developments will be needed to match the modern form of perceptrons, which apply a wider variety of activation functions than a step function, such as the sigmoid function, for making soft decisions and predicting probabilities[4], or the Rectified Linear Unit (ReLU) to ease training. Sigmoidal responses have long been known in systems biology to be feasible with enzymatic networks (e.g. with cooperativity or push/pull motifs[38]), and simple chemical schemes to compute ReLU were recently proposed[39,40].
>
> More generally, the scale of our networks is sufficient for molecular diagnosis (see below), but work will be needed to reach the scale of *in-silico* machine learning, where neural nets typically have dozens of layers and millions of weights, and are trained on datasets with tens of thousands of examples. In principle, enzymatic networks and droplet microfluidics can handle these scales (as evidenced by the intricate computations performed by gene regulatory networks in cells, or by consortium of single celled organisms), but the current tools for writing, handling, and reading DNA would struggle (Supplementary Information Section 6). However, these hurdles could be overcome by an exponential drop in the cost of DNA synthesis - which is expected in the coming years in response to fields that make heavy use of synthetic DNA. In the long term, enzymatic neural networks could empower the nascent field of DNA data storage[41]. Large amount of data could be stored in DNA

databases, and queried, labelled or processed in a massively parallel fashion with enzymatic neural networks and droplet microfluidics.

In the short term, neuromorphic computation could readily find applications in molecular diagnostic. Our enzymatic toolbox previously detected a tumor suppressor miRNA in total RNA from human colon with high specificity and sensitivity[42], and our enzymatic neural networks are similar in size to *in-silico* neural networks that reliably diagnose breast tumors[43] or prognose metastasis[44] from multiple molecular clues. This suggests that cancerous patients could be monitored at the point-of-care, using enzymatic neural networks that make diagnosis or prognosis from a panel of miRNAs present in liquid biopsies.

Reviewer Reports on the Second Revision:

Referees' comments:

Referee #1 (Remarks to the Author):

I have read the revised manuscript with pleasure. This work has been remarkably improved after several rounds of revisions. It seems to me that the authors have satisfactorily responded to my concerns and other reviewers'. Overall this is a great paper that demonstrates important ideas and experimental implementation to advance DNA computing.

Referee #3 (Remarks to the Author):

Throughout the paper, the results obtained are presented in an easy-to-understand manner without exaggeration, and also the readability has been greatly improved.

*The words such as "deep" and "perceptron" has been changed to more realistic expressions.

*The abstract describes what has been accomplished without excess or deficiency.

*Figure 1 has been rewritten to show the hierarchy, which makes it easier to grasp the proposed principle.

*In the discussion, the limitations and the bottlenecks to be solved by future research toward the realization of the perceptron are clearly stated. The examples of short-term applications are also appropriate.