Reservoir Computing with Bacteria

Paul Ahavi[1], Thi-Ngoc-An Hoang[1], Philippe Meyer[1], Sylvie Berthier[2], Federica Fiorini[2], Florence Castelli[2,3], Olivier Epaulard[2,4], Audrey Le Gouellec[2], Jean-Loup Faulon[1]

[1] University Paris Saclay, INRAE, AgroParisTech, MICALIS Institute, 78350 Jouy-en-Josas, France.
[2] University Grenoble Alpes, CNRS, CHU Grenoble, Grenoble INP, TIMC-IMAG UMR 5525, 38000 Grenoble, France
[3] University Paris Saclay, CEA, INRAE, Médicaments et Technologies pour la Santé (MTS), MetaboHUB, 91190 Gif-sur-Yvette, France
[4] Infectious Diseases Department, Grenoble-Alpes University Hospital, 38000 Grenoble, France

Summary

In contrast to the traditional bottom-up approach of engineering biological devices to perform computations within living organisms[1,2], this study explores a novel strategy by harnessing bacterial strains within a reservoir computing framework[3] to address regression and classification problems.

We use *Escherichia coli* K-12 MG1655 as a biological reservoir, cultivated in M9 minimal medium supplemented with nutrients, and measure growth rates across different media compositions. Our *E. coli*-based reservoir computer (RC) consistently outperforms multiple linear regression (MLR) and support vector machines (SVM), and achieves performance comparable to multilayer perceptron (MLP) across a range of regression and classification tasks. Furthermore, reservoirs based on genome-scale metabolic models (GEMs) of various bacterial species show performance trends that correlate with the phenotypic diversity of the organisms.

As a practical classification application, we deploy our *E. coli* RC system on plasma samples collected from early-stage COVID-19 patients to predict the outcome of the disease (mild vs. severe). Our system achieves in few hours competitive accuracies in distinguishing negative from positive and mild from severe cases. Unlike current approaches relying on omics[4] or hospital record analyses[5] to distinguish mild from severe cases, our solution is infrastructure-independent, inexpensive, and easy to use.

These results present a novel paradigm for biocomputing, capable of addressing computational challenges traditionally reserved for digital approaches. Our bacterial reservoir offers a practical advantage over other reservoir types[6,7] due to its simplicity, minimal technical requirements, cost-effectiveness, and energy efficiency. Our approach opens promising avenues for biosensing, medical diagnostics, and integration with emerging platforms such as neuromorphic hardware and engineered living materials.

Main

Performing computation with living organisms is one of the main endeavors of synthetic biology where many electronic-like circuits have been engineered in a bottom-up fashion[1,2]. Engineering and tuning such circuits is challenging and several factors such as metabolic burden[8], orthogonality issues[9] and low noise tolerance[10] strongly restrain their complexity. Several alternatives have been developed to address these shortcomings including directed evolution[9], multicellular computing[11] and the engineering of less noisy and energy-consuming devices such as analog circuits[12] or perceptron architectures[13]. However, these solutions are still limited in scope, which restricts their applicability in real-world scenarios[14].

Synthetic biology devices have largely been inspired by behaviors observed in natural systems. Indeed, information processing systems analogous to logic gates[15], switches[16], and perceptrons[17] have all been discovered in various organisms. This raises the question: instead of engineering bottom-up devices, could we develop a top-down approach where natural organisms are repurposed to perform complex computational tasks?

The study of physical systems' ability to solve computational problems falls within the field of Artificial Intelligence known as reservoir computing (RC). Initially developed as an alternative to classical artificial neural networks, particularly recurrent neural networks (RNNs)[3], RC offers a more efficient training process. There are two kinds of RC systems: conventional RC and physical RC.

In a typical conventional RC framework, input data is provided to a reservoir, the states of the reservoir nodes are read, and this information is fed into a post-readout, usually a simple linear classifier or regressor, that interprets the reservoir states to produce the final output (*cf*. Fig. 1). Unlike RNNs with trainable weights, conventional RC uses a fixed-weight RNN reservoir, making it faster to train and suitable for applications such as time series prediction, forecasting dynamic system control, and Internet of Things (IoT) applications where quick, efficient processing is crucial[18].

In physical RC, the reservoir is replaced by an actual physical system or process. The system operates by first feeding input data into the physical reservoir, which processes the inputs and transforms them into a high-dimensional, dynamic state. These transformed states are then passed to a post-readout layer, similar to conventional RC. Physical RC has many applications, including the seminal liquid state machine for pattern recognition in a bucket of water[19], chemical RC developments for classification

tasks and forecasting solutions of ODE systems[6], and biological RC systems using rat cortical neurons cultured on micropatterned substrates to solve classification tasks[7].

The aim of the current paper is to use bacteria as reservoirs and to probe to which extent bacterial models in conventional RC systems and *E. coli* (K-12 MG1655) in a physical RC system can solve machine learning regression and classification tasks of increasing difficulties. We also present a practical *E. coli* physical RC application with the prediction of COVID-19 disease severity (mild vs. severe) using plasma samples.

Bacterial reservoir computing framework

In typical machine learning regression and classification tasks, training sets are composed of features and labels, and the goal is to predict the labels from the features. Similar to the physical RC systems mentioned earlier, the bacteria act as our reservoir. We transform the problem features into nutrients provided to the strain, observe the strain's behavior by measuring its phenotype, and feed these phenotype measurements to a feedforward neural network, which outputs the problem's solution.

One challenge in this workflow is finding the most appropriate mapping between features and nutrients. There is no generic procedure in the RC literature; instead, there are various methods depending on the reservoir (e.g., using water waves to mimic sound waves for classifying voice recordings in liquid state machines[19]). In our case, since we are working with several features, one strategy could be an arbitrary one-to-one mapping of features to nutrients. However, this strategy might not yield the best performance. Another approach could be to find the optimal nutrients by training a prior input layer that generates nutrient sets. While this is not customary in RC, there is no reason why the prior layer cannot be trained[20]. To train the prior layer, one needs to backpropagate a gradient from the post-readout layer to the prior-input layer. This requires a gradient-compatible model that mimics the behavior of the strain, which in turn necessitates replacing the physical strain reservoir with a conventional reservoir that acts as a surrogate for the strain.

Many solutions have been developed over several decades to model strain behavior. Among these, we are particularly interested in models dealing with metabolism, as we provide the strain with various metabolites (nutrients) and measure phenotypes such as growth rates. While kinetic models have been proposed to model metabolism, they are still somewhat limited in scope regarding species and sizes[21]. Instead, we propose to use genome-scale metabolic models (GEMs) that have been developed for many strains[22]. To capture the dynamics of these models, we choose flux balance analysis (FBA)[23] a well-established technique in systems biology and bioengineering.

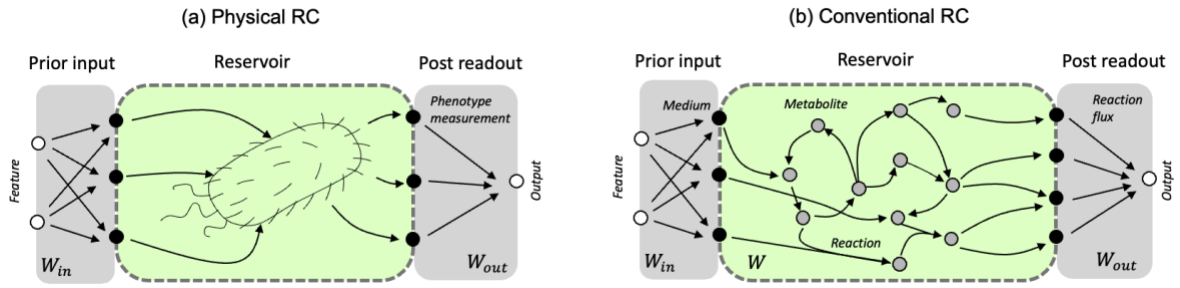


Fig. 1. Bacterial physical and conventional RC. *The Figure, inspired by Tanaka et al.*[3] *was adapted for bacterial strain and models. (a) In physical RC the reservoir is realized by a physical (here biological) system or device (here a strain). (b) A conventional RC is a recurrent neural network (here a GEM model). In both cases the post readout neural network weights (W_{out}) are learned during training and the prior input weights (W_{in}) and reservoir weights (W) are fixed. Yet, with conventional RC, the weights W_{in} can be learned through training as long as the reservoir allows gradient backpropagation.*

Throughout this paper we will be using an *E. coli* physical RC and a conventional twin based on an *E. coli* GEM model. We will also make use of conventional RC for other bacteria for which GEM reservoirs are available.

RC to improve the performances of GEM models

To evaluate the ability of a conventional reservoir to accurately surrogate a physical reservoir, we need to assess how well the conventional reservoir reproduces phenotype measurements when the strain is grown in different media. To this end, we grew *E. coli* K-12 MG1655 on M9 minimal media supplemented with 28 metabolites (including 20 amino acids, 5 nucleobases, and 3 sugars or acids; *cf.* Extended Data Fig. 1a and 'Bacterial strain and culture media' in the Methods section). All metabolites were added at fixed concentrations, allowing each composition to be represented by a binary vector. We then measured the growth rate of *E. coli* on 280 such media compositions (*cf.* 'Experimental training set' in the Methods section).

For modeling *E. coli*, we used iML1515[24], the latest *E. coli* genome-scale metabolic model (GEM). When performing flux balance analysis (FBA) using our 280 growth media, we found a poor correlation coefficient with the experimental measurements (*cf.* Extended Data Fig. 1b). It is well known that FBA struggles with reproducing quantitative measurements accurately[25]. This difficulty arises because FBA does not take input media compositions directly but instead relies on media uptake fluxes. While uptake fluxes can be measured for individual metabolites, the interactions among multiple metabolites in a medium make such measurements prohibitive.

To enhance the performance of FBA, we developed an RC system where the prior-input layer performs the mapping between media composition and media uptake fluxes, the reservoir carries out FBA calculations, and the post-readout layer, which does not need to be trained, just prints out the FBA calculated growth rates (Extended Data Fig. 1c). FBA is a linear program making use of the Simplex algorithm[26], which does not allow gradient backpropagation to the prior layer. To overcome this issue, we replaced the FBA reservoir by an Artificial Metabolic Network (AMN), a surrogate of FBA which we previously developed[27], and is gradient backpropagation compatible (*cf.* 'Building GEM reservoirs' in the Methods section, and Supplementary Figure S1). Results obtained using 5-fold cross-validation (*cf.* 'Cross validation' in Methods) indicate a substantial improvement over FBA when predicting growth rate (*cf.* Extended Data Fig. 1d where $R^2 = 0.78$ vs. 0.12 for FBA).

Conventional GEM RC to solve regression problems

Having shown that RC can substantially improve phenotype predictions when compared to measurements, we now utilize RC to solve regression problems extracted from the OpenML platform (*cf.* 'Training sets' in the Methods section and Supplementary Table S1). Our RC system architecture (Fig. 2a) consists of a prior network mapping problem features to medium composition, a reservoir composed of the trained neural layer and AMN shown in Extended Data Fig. 1c, and a post-readout layer transforming the reaction fluxes computed by the AMN into the problem readout via a feedforward neural network.

We developed two different methods for the prior layer. The first method (fixed prior) does not require training; instead, features are one-to-one mapped to medium metabolites using permutation importance (*cf.* 'Regression with RC / fixed prior' in the Methods section and Supplementary Table S2). The second method (generative prior) outputs medium composition from features, with a generative model trained to produce the best medium composition for each element of the training set (*cf.* 'Regression with RC / generative prior' in the Methods section and Supplementary Table S2).

We tested these architectures on 10 different regression problems of increasing difficulty and compared their performance with classical regressors : multi-linear regression (MLR), multilayer perceptron (MLP), and XGBoost (XGB) (Figs. 2b, 2c). Hyperparameters for the classical regressors can be found in Supplementary Table S3)

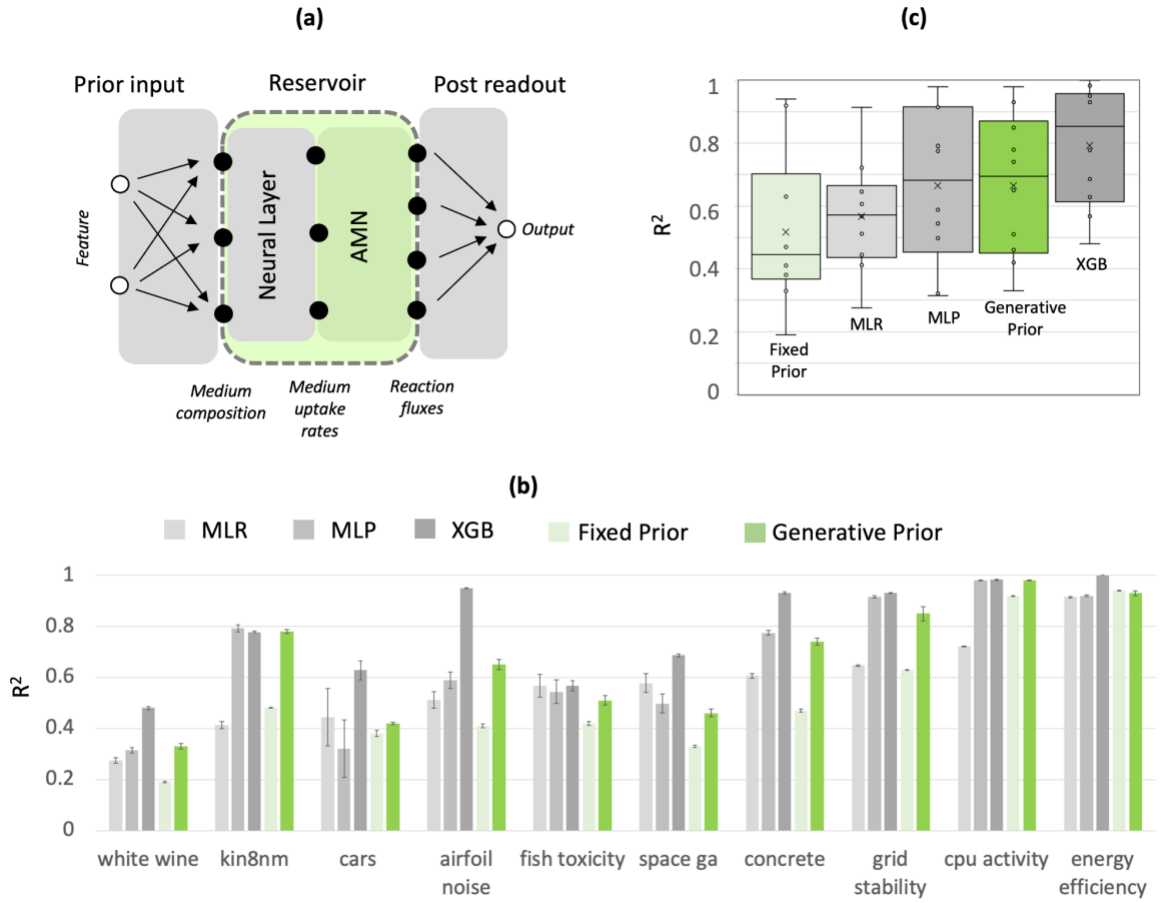


Fig. 2. Regression problem performances with conventional *E. coli* RC. *(a) Schematic of the RC system architecture used for regression problems. The prior input layer maps problem features to medium composition, the reservoir (composed of a trained neural layer coupled to an AMN) computes all reaction fluxes, and the post-readout layer converts reaction fluxes into problem readouts via a feedforward neural network. (b) Performance comparison of different regression models (MLR, MLP, XGB, and conventional RC with fixed and generative priors) across 10 problems from the OpenML platform. (c) Box plot of R^2 values for the 5 different models and the 10 regression problems of panel (b). All R^2 values were acquired on cross-validation sets (5-fold), standard deviation was calculated on 3 replicates. Data are in Supplementary file 'Reservoir_Regression'.*

We observe from Fig. 2 that the architecture with fixed prior is less performant than all other methods, and that the generative prior outperforms MLR and has performances similar to MLP. This indicates that a one-to-one mapping between features and medium metabolites is not necessarily the best strategy to find the most appropriate medium composition. On average, we find an improvement of 0.15 in the regression coefficient between the generative and fixed priors, indicating that some learning is still performed in the fixed prior architecture and consequently by the GEM reservoir (since nothing is learned through training in the fixed prior). Additionally, we observe that for some problems, the fixed prior architecture outperforms MLR.

To further assess the ability of bacterial strain models to solve regression problems, we tested 10 different GEM models (*cf*. Supplementary Table S4) with the same 10 regression problems. As experimental phenotype measurements are not available for all these strains, our GEM reservoirs were trained on FBA simulated data (*cf*. 'Building GEM reservoirs' in the Methods section) with the 28 media metabolites we used previously. In agreement with our previous observations, results shown in Extended Data Fig. 2 indicates that the fixed prior architecture is less performant overall, and the generative prior architecture outperforms MLR and even MLP for many strains. Interestingly, we also observe that different strains yield different results, and these differential performances are independent of the chosen prior layer.

To probe what makes a given GEM model better (or worse) suited to learning than another, we computed the number of different phenotypes the GEMs could produce with media composed of the 28 supplemented metabolites we already used (*cf*. 'Building GEM reservoirs' in the Methods section). Extended Data Fig. 2 (panels b and e) indicate a trend: the higher the number of phenotypes, the higher the regression coefficient. The ranking of the strain thus obtained is conserved between the fixed and generative prior layers. This behavior can naively be understood when considering that a strain producing n=2 different phenotypes (e.g., like growth, no-growth) should yield poor performance when subjected to regression problems requiring far more than 2 outputs. This trend, using numbers of phenotypes, is even more pronounced when computing the Shannon entropy for the phenotypes (*cf.* 'Phenotype number and entropy calculation' in the Methods section). Indeed, Extended Data Fig. 2 (panels c and f) indicate Pearson correlation coefficients between the log of the entropy and the median R^2 values of 0.79 and 0.86 for the fixed and generative priors, respectively.

A physical *E. coli* RC to solve regression problems

As shown in Fig. 3a, with a physical *E. coli* RC the reservoir is an actual strain. There are two main differences between such a physical RC and a conventional one: (1) only growth rates are considered among the phenotypes, and (2) the growth rates used to predict problem readouts are measured rather than being predicted by a model. Following these constraints, our physical *E. coli* RC is built following the workflow given below.

We first trained a conventional *E. coli* RC having a reservoir that mimics the behavior of the strain. The reservoir of this conventional twin is similar to the one presented in Extended Data Fig. 1 with phenotypes reduced to only growth rates. This reservoir was trained on the set of 280 media compositions and growth rates presented in Extended Data Fig. 1 and Supplementary file

'Reservoir_Regression.' We next collected the media compositions predicted by the conventional twin to best solve our 10 regression problems, and grew *E. coli* K-12 MG1655 with the provided media and measured growth rates. Lastly we fed black the growth rates to the post layer of the twin to predict the problem readout.

Our conventional twin makes use of two different prior architectures : generative and selective. The generative prior, seen previously, is producing media compositions some of which might already be in our 280 media compositions set while others are novel. We note that growth rates need to be measured for these novel media compositions when feeding the generative prior output to the physical RC. The selective prior is forced to choose media compositions that are in the 280 media compositions set. That selection is performed using a Gumbel-softmax[28] activation function as described in 'Regression with RC / Selective prior' in the Methods section. With this prior all growth rates have already been measured and it is not necessary to carry out additional growth measurements.

We note that since our physical RC and conventional twin are considering only growth rate as phenotype is it not necessary to train a post-readout layer, in fact after normalizing the growth rate and the problem readout the problem readout *is* the growth rate.

Results on the 10 regression problems obtained with the twin conventional RC and with generative or selective priors are presented in Fig. 3b and Extended Data Fig. 3. As seen before, the twin conventional RC outperforms MLR and yields performances similar to MLP.

To further assess the capacity of our physical *E. coli* RC to solve regression problems we selected 3 problems for which the conventional twin predicted low and high R^2 values. Additional growth rate measurements were carried out when using the generative prior but not when using the selective prior. Results presented for three regression problems in Fig. 3 panels c, d, e show that the physical reservoir leads to a performance drop compared to results obtained with the conventional twin when using the generative prior but not when using the selective prior. The drop with the generative prior can easily be explained by the fact the reservoir of the conventional twin is not a perfect twin as it predicts growth rate with a regression coefficient of 0.78 (*cf*. Extended Data Fig. 1 and Supplementary Figure S2). As we have seen before with conventional RC, our physical RC generally outperforms MLR and when using the selective prior has similar or better performances than MLP for some regression problems (*cf.* Extended Data Fig. 3 for selective prior).

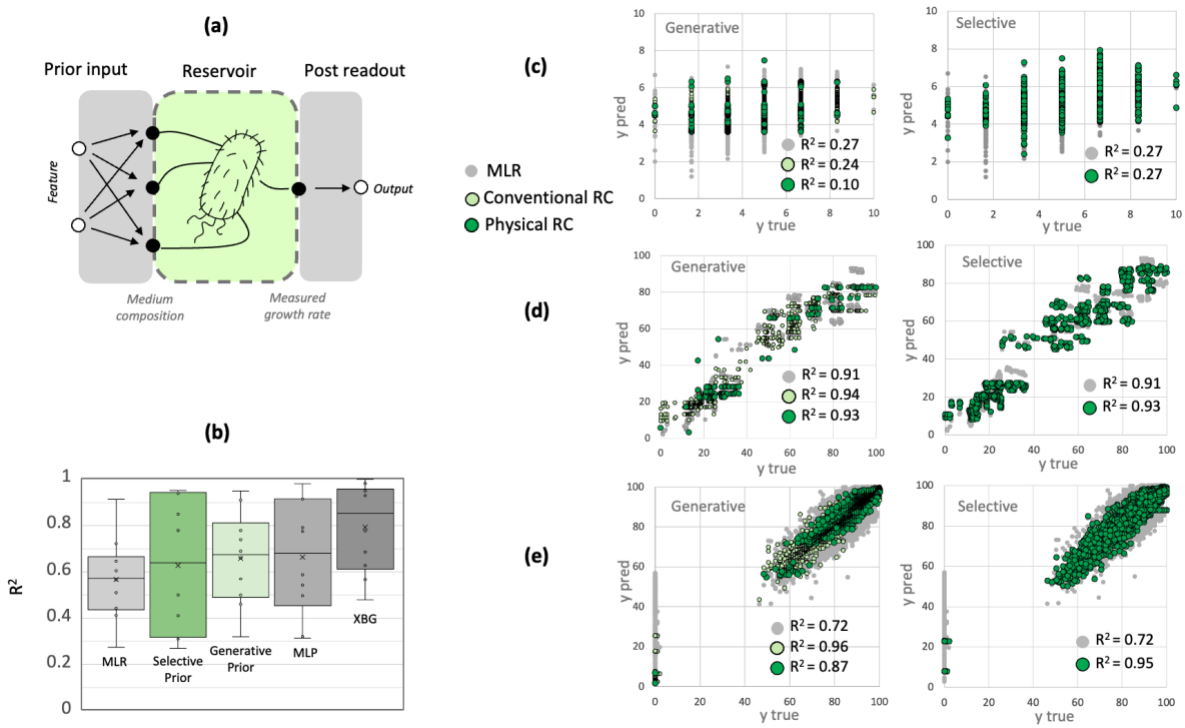


Fig. 3. **Regression problem performances with physical *E. coli* RC**. *(a) The architecture of the physical RC system. (b) Box plot of R^2 values obtained for 10 regression problems of Supplementary Table S1 with MLR, MLP , XGB and conventional twin models with the selective and generative priors. (c) Results obtained for the 'white wine' regression problem with MLR and conventional and physical RC making use of generative and selective priors. All physical RC results are derived from measured growth rates. With the generative prior 4898 y values were predicted with conventional RC, 1639 of which corresponding to an experimentally measured growth rate (many y values corresponding to the same growth rate). With the selective prior all y values were derived from experimentally measured growth rates. (d) Results obtained for the 'energy efficiency' regression problem. With the generative prior 768 y values were predicted with conventional RC, 162 of which corresponding to an experimentally measured growth rate. With the selective prior all y values were derived from experimentally measured growth rates. (e) Results obtained for the 'cpu activity' regression problem. With the generative prior 8192 y values were predicted with conventional RC, 3057 of which corresponding to an experimentally measured growth rate. With the selective prior all y values were derived from experimentally measured growth rates. Results presented in all panels were acquired on cross-validation sets (5-fold). Data are in Supplementary file 'Reservoir_Regression'.*

Conventional and physical *E. coli* RC to solve classification problems

Turning our attention to classification problems, we chose to benchmark our *E. coli* reservoirs against both linear and non-linear classification tasks, similar to those used in a chemical reservoir computing system[6]. As shown in Fig. 4a, we focused on eight classification tasks, generating a two-dimensional input space from 500 randomly sampled x_1, x_2 values in the [0,1] interval. Labels (y = 0 or 1) were assigned based on the patterns (*cf.* "Classification Problems" in the Methods section for further details). As with our regression tasks, the x_1, x_2 values were transformed into medium metabolite compositions using a prior layer (*cf.* "Regression and classification with RC" in the Methods section).

We utilized the conventional reservoir from Fig. 2a with a fixed prior, alongside the *E. coli* physical reservoir from Fig. 3a with a selective prior. In all cases, the post-readout layer consisted of a simple sigmoid function that took as input all reaction fluxes for the fixed prior and growth rates for the selective prior.

From the results shown in Fig. 4, we observe that the conventional reservoir computing (RC) system with fixed priors performs better on classification tasks compared to regression tasks. With this RC system, the prior layer remains untrained, and learning occurs exclusively in the post-readout layer. For all non-linear tasks the fixed-prior RC system outperformed classical SVM and outperform MLP classifiers for the SINE and DOTS tasks. We recall that strain-RC performance correlates with the diversity of strain phenotypes (*cf.* Extended Data Fig. 2). Since classification tasks require only two classes and phenotypes, this may explain the observed improvement in the performance of the fixed-prior conventional reservoir on classification tasks compared to regression tasks.

Our physical RC system with a selective prior, where all reservoir outputs are experimentally validated, outperformed SVM in all tasks and exhibited similar or superior performance to MLP for the AND, OR, LINEAR, SINE, and DOTS tasks. These findings indicate that our physical *E. coli* reservoir with a selective prior is capable of handling non-linear classification tasks. Similar trends have already been observed in chemical reservoirs[6] although in our case, we rely solely on growth rate measurements instead of monitoring the concentrations of multiple species.

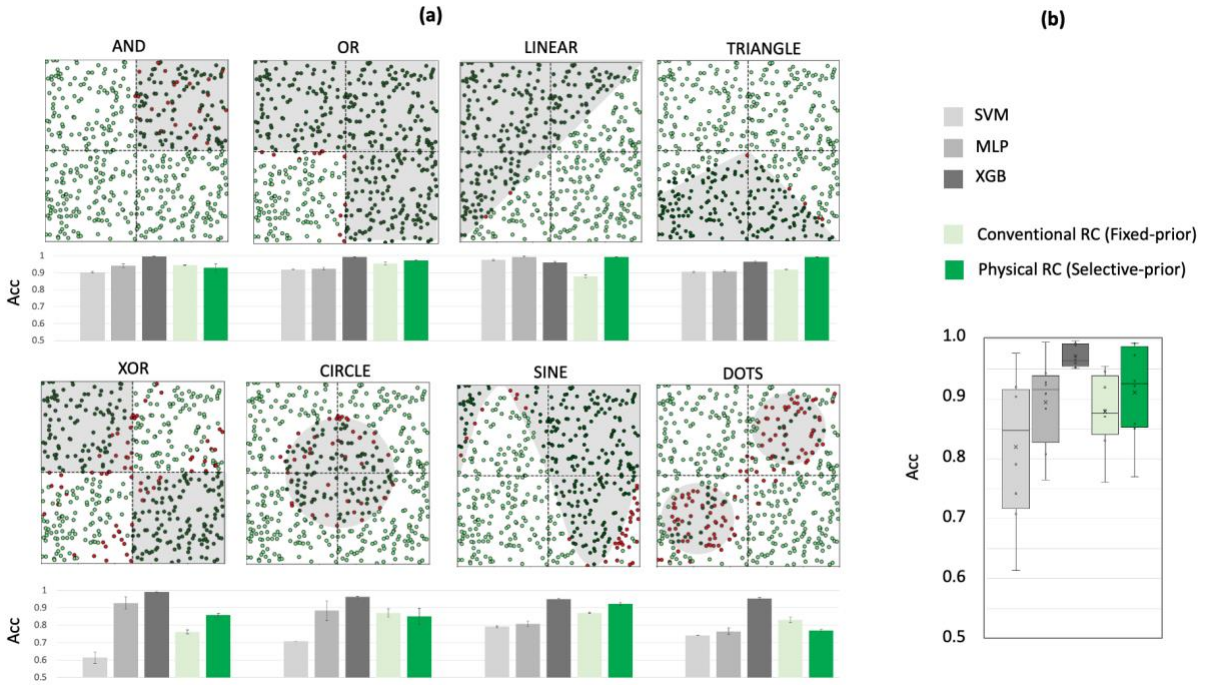


Fig. 4. **Classification problem performances with conventional and physical *E. coli* RC.** ***(a)*** *Performances for 4 linear (AND, OR, LINEAR, TRIANGLE) and 4 nonlinear (XOR, CIRCLE, SINE, DOTS) classification tasks with SVM, MLP and XGB classical classifiers along with E. coli conventional RC system with fixed prior and E. coli physical RC system with selective prior**.** Red dots are false positives and false negatives for the physical RC predictions.* ***(b)*** *Color codes and box plots of accuracies obtained for the 8 classification tasks of panel (a). Data are in Supplementary file 'Reservoir_Classification'. Results using Mathews Regression coefficients (MCCs) and 26-fold cross validation as in Baltussen et al[6] are given in Supplementary Figure S3.*

Physical *E. coli* RC to classify COVID-19 plasma samples

Classifying COVID-19 patients to predict whether they will develop a mild or severe form of the illness has been a significant research focus since the onset of the pandemic[4]. Most methods performing classification are making use of proteomic and metabolomic profiling conducted on biological samples (plasma[29], sera[4], urine[30]) collected from patients upon their admission to the hospital. After having been trained on these measurements, ML techniques are used to predict the disease outcome (e.g., mild or severe according to whether ICU admission was required[29,31]). In line with this approach, we utilized a cohort of COVID-19 patients from the CHUGA University Hospital (*cf*. 'Covid cohort acquisition' in Methods and Supplementary file 'Covid_Data_MS_ML') and performed untargeted metabolomics analyses (via high resolution mass spectrometry) on plasma samples collected on 81 COVID-19 patients (50 mild, 31 severe, *cf*. 'Metabolomics Analyses' in the Methods section and Supplementary file 'Covid_Data_MS_ML'.

Extended Data Fig. 4a demonstrates that disease outcomes can be predicted using this metabolomics data with cross-validation accuracies greater than 0.9 using feature selection (cf. Methods and Supplementary Figure S4). As our goal is to develop a physical reservoir for classifying the samples, we also successfully made predictions by extracting signals from the metabolomics data corresponding to metabolites expected to cross the *E. coli* cell envelope and contributing to its growth medium (*cf*. 'ML analyses on metabolomics data' in Methods for the selection of medium metabolites and Extended Data Fig. 4a and Supplementary Figure S4 where the accuracy for medium metabolites is found above 0.8 using feature selection).

Having demonstrated that medium metabolite MS signals could be used to distinguish mild from severe samples, we tested if a conventional *E. coli* GEM reservoir could also distinguish mild from severe based on strain phenotypes. The architecture of that reservoir is shown in Extended Data Fig. 4b. Without measured growth rates, we trained the AMN of Extended Data Fig. 4b with data produced by a GEM (iML1515 model) coupled with FBA as our conventional reservoir. The data was generated using the classical objective of maximizing growth rate when provided with the MS signals of metabolites predicted to cross the *E. coli* cell envelope. When using the conventional reservoir of Extended Data Fig. 4b we were unable to classify mild versus severe outcomes with sufficient confidence when the post-readout was based solely on growth rate. However, when the post-readout was based on GEM/FBA-computed flux distributions for all reactions, we reached an accuracy above 0.9 using feature selection for several classifiers (*cf*. Extended Data Fig. 4c and Supplementary Figure S5). This high accuracy must be interpreted carefully, since GEM/FBA-computed fluxes show limited agreement with experimental measurements (*cf*. Extended Data Fig. 1). At best, this accuracy supports further development of a physical reservoir.

Seeking to construct a physical *E. coli* reservoir growing on plasma samples, we collected samples from patients of the CHUGA COVID-19 cohort before they developed either mild or severe forms of the disease *(cf*. 'COVID-19 patient plasma samples' in Methods). To this cohort, we added plasma samples from COVID-negative patients, obtained from the French Blood Establishment (Etablissement français du sang, EFS). Since *E. coli* cannot grow in the presence of antibiotics, any samples containing antibiotics were excluded. The final dataset consisted of 27 negative samples, 37 mild cases, and 36 severe cases (*cf*. Supplementary file 'Reservoir_Covid').

Plasma samples had to be pretreated in order to optimize bacterial growth on plasma and therefore classification potential across sample categories. In brief, samples were deproteinated and

supplemented with magnesium and a phosphate buffer. All experimental procedures are detailed in the Methods section under 'Growth experiments on patient plasma samples and in Supplementary Methods 'Optimization of *E. coli* growth in plasma samples'.

By growing *E. coli* on negative, mild, and severe plasma samples, we first experimentally confirmed our conventional reservoir prediction: such a reservoir is unable to solve the mild vs. severe classification task when relying solely on growth rate as input to the post-readout classifier. We also observed we could not classify mild vs. severe when taking lag time and maximum optical density taken at 600 nm (OD_{MAX}) as input features in addition to growth rates (*cf.* Supplementary Table S5). To further increase the number of features available to the classifier, we could not use flux distributions as in a conventional reservoir. Instead, we utilized optical density measurements taken at all time points along the growth curves (*cf.* Fig. 5).

Using this approach and 20-fold cross-validation, the *E. coli* physical reservoir achieved a classification accuracy of 0.84 ± 0.006 for distinguishing mild from severe samples. Furthermore, the reservoir demonstrated high accuracy (0.94 ± 0.005) in distinguishing negative samples from positive ones. These results were obtained by selecting sets of time points that maximized classification accuracies (*cf.* 'Feature selection for classification' in Methods and Supplementary Figure S6). Results for other classifiers (SVM and MLP) and other folds (5 and LOO) are presented in Supplementary Table S6.

For the mild vs. severe classification task, the selected time points are all within the first 6 hours. In contrast, for the negative vs. positive classification, the selected time points extend up to 8 hours. Generally, *E. coli* biomass at 8 hours is lower in positive samples compared to negative ones, and we observe that growth is generally delayed in the first 4 hours when *E. coli* is grown on severe samples compared to mild ones.

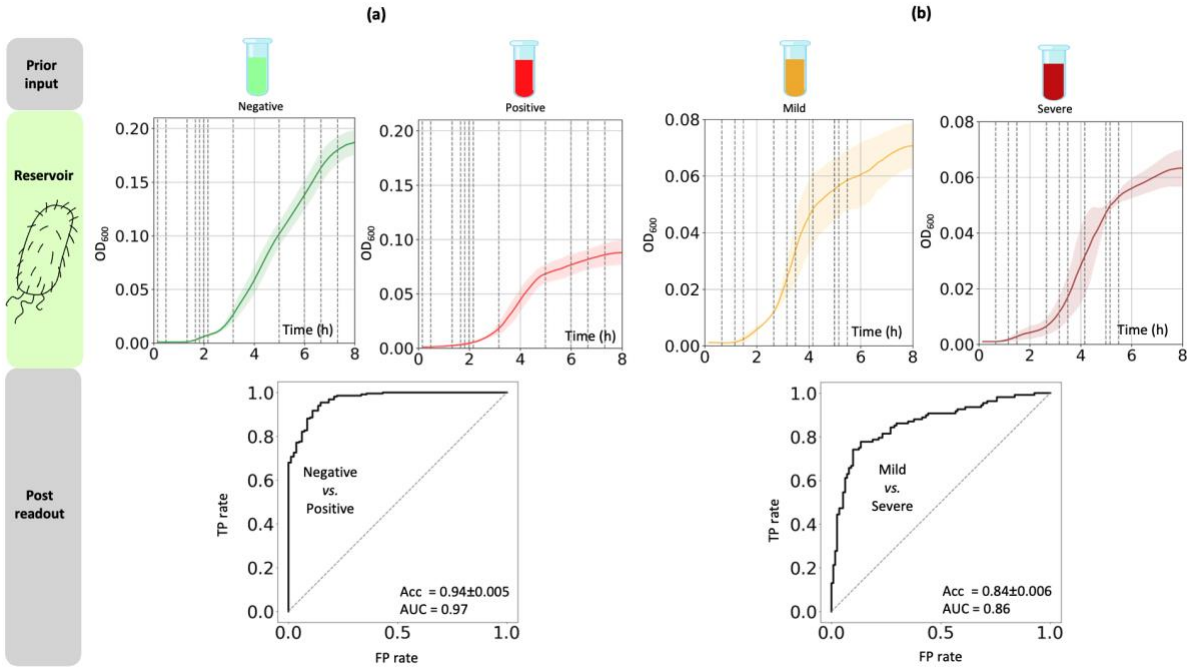


Fig. 5. Classifying COVID-19 plasma samples with *E. coli* physical reservoir. *The prior input layer is a sample used to grow E. coli K-12 MG1655. The samples are either COVID negative, mild, or severe, and are supplemented with the salts listed in 'Growth experiments on patients' plasma samples' in Methods. The reservoir is MG1655 on which growth curves (2 technical replicates) are acquired using a plate reader recording OD_{600} at 48 time points taken from 0 to 8h. The readout layer is a XGB classifier that takes as input OD_{600} values for selected time points and prints out an accuracy for 20-fold cross validation repeated 3 times. The time points indicated by dashed vertical lines are the one selected among all time points in order to maximize accuracy (cf. 'Feature selection for classification' in Methods). (a) Results obtained when E. coli grown on a negative sample (N1 in Supplementary Figure S7) and a COVID-19 positive sample (M10 in Supplementary Figure S8). (b) Results obtained when E. coli is grown on a mild sample (M9 in Supplementary Figure S8) and a severe sample (S8 in Supplementary Figure S9). Growth curves for all samples are plotted in Supplementary Figures S7-S9 and OD_{600} values can be found in the Supplementary file 'Reservoir_Covid'.*

To further asses group differences in OD trajectories, we employed Generalized Additive Mixed Models (GAMMs)[32] in Supplementary Method 'Statistical Modeling of *E. coli* growth curve'. For the positive vs. negative classification, GAMMs found statistically significant differences between groups in both the average OD across time points (p-value = 5.37×10^{-5}) as well as in the temporal trajectories of OD over time (p-value < 2×10^{-16}). For mild vs. severe cases, the overall mean OD did not differ significantly between the groups (p-value = 0.654). Notably, the mild mean OD was higher than severe during the first 4 hours and slightly lower than severe after 4 hours and GAMMs revealed the presence of significant group-specific smooth terms (p-value < 2×10^{-16}) indicating that the temporal evolution of OD differed substantially between mild and severe cases (*cf.* Supplementary Table S7). These findings imply that while the average OD levels may be similar, the dynamics of how OD changes over time are distinct between the mild and severe groups.

Viral replication, organ dysfunction and immune cell expansion following a COVID-19 infection are responsible for wide metabolomic alterations in serum and plasma. These variations might explain the growth pattern differences observed in the different cases. Decreased levels of lipids and amino acids derivatives[4,33] (mostly from the arginine metabolism) have been observed in COVID-19 patients compared to uninfected individuals[4]. Given that *E. coli* can catabolize some classes of lipids, these observations might explain the reduced OD values during the exponential and stationary phase when *E. coli* is cultured in plasma from COVID-19 patients. It should also be noted that increased levels of other carbon sources such as glucose, glucuronate or pyruvate have been reported in the plasma of COVID-19 patients[4,34]. However, it appears that the increased levels of these carbon sources doesn't compensate for the effects of lipid depletion.

Regarding the mild vs. severe classification task, the more subtle differences in growth pattern may reflect more diverse alterations in the plasma metabolome at early stages of the infection. Decreased levels of several lysophosphatidylcholine (LCP) lipids and increased levels of kynurenine have been identified as negative prognosis biomarkers across a majority of the studies[4]. Altered levels of amino acids derivatives, nucleobases derivatives and polyamines have also been used as prognosis markers, though with less consistency across studies[29,35,31]. In a broader context, Maeda *et al*.[36] demonstrated that amino acid deamination is associated with poor prognosis. Therefore, increased plasmatic levels of amino acid catabolites are expected for patients who will develop a severe form of the disease. Taken together, these variations may represent a plausible mechanistic basis for the growth differences observed in the mild vs. severe classification task.

Discussion

This study demonstrates the potential of leveraging conventional reservoirs constructed from bacterial strain models, alongside with physical *E. coli* reservoirs, to address both regression and classification tasks.

Our conventional bacterial RC systems outperform traditional regression approaches, including multi-linear regression (MLR) and multi-layer perceptron (MLP), particularly when a generative prior is used to optimize media composition. In classification tasks, even untrained fixed reservoir priors achieved unexpectedly strong performance, surpassing standard SVM classifier in all cases and MLP in several tasks. The differences between regression and classification performance may stem from the fact that

binary classification requires only two separable phenotypic responses, while regression tasks demand a richer output space.

More broadly, the hypothesis that increased phenotypic diversity enhances task-solving performance (cf. Extended Data Fig. 2) is conceptually supported by prior research in reservoir computing. Dambre *et al*.[37] theoretically demonstrated that the computational capacity of abstract dynamical systems is limited by the number of independent internal states. This concept was later validated using dynamical observables in quantum spin network reservoirs[38] while Dale *et al*.[39] correlated response diversity with reservoir quality in simulated neural systems.

Our physical RC systems, utilizing an *E. coli* strain with selective priors, further demonstrates the robustness of bacterial RC systems in handling complex, regression and classification tasks. With this prior, RC system consistently outperform MLR and SVM across all tasks and achieved comparable or better performance than MLP in many regression tasks (*cf.* Extended Data Fig. 3) and in half classification tasks (*cf.* Fig. 4).

Compared to the Formose chemical reservoir[6] and biological neuronal reservoirs (mBNNs)[7], the *E. coli* physical reservoir offers a practical balance of classification performance, simplicity, and cost-effectiveness. The setup requires only standard microbiology equipment: an incubator and a plate reader with minimal consumables. In contrast, the chemical reservoir depends on mass spectrometry and flow control systems, while mBNNs require optogenetics, microfluidics, and live-cell imaging, all of which demand substantial infrastructure and technical expertise. Despite this simplicity, the *E. coli* reservoir delivers competitive accuracies across a range of regression and classification tasks, relying solely on growth rates as inputs to the readout.

The main key advantage of the *E. coli* physical reservoir compared to other published chemical and biological reservoirs lies in its applicability to clinically relevant settings. To test this potential, we applied the system to classify COVID-19 patient plasma samples by disease severity. Standard ML pipelines applied to untargeted metabolomics data yielded classification accuracies around 0.9. Using OD_{600} growth curves, our *E. coli* reservoir distinguished between mild and severe COVID-19 cases with a cross-validated accuracy of 0.84 with 20-fold cross-validation (0.86 with LOO), and achieved even higher accuracy (0.94) when classifying between COVID-19 positive and negative samples.

Aside from using omics analyses only few studies have been carried out to predict COVID-19 severity and these generally include human health records as features[5]. Compared to these methods, our physical *E. coli* reservoir approach offers several advantages. Omics-based pipelines are generally expensive, time-consuming, and require specialized infrastructure and expertise. Our reservoir system

achieves competitive accuracy without including patient data, operates with minimal cost and instrumentation, and produces results within hours using only a plate reader and bacterial cultures. These properties make it a highly accessible, scalable, and infrastructure-independent alternative for severity prediction, particularly valuable in decentralized or low-resource healthcare settings.

Overall, our results highlight the promise of using bacterial strains in physical reservoir computing (RC) systems as an alternative to engineering in synthetic biology approaches. Rather than engineering genetic circuits from scratch (a strategy often limited by metabolic burden, component orthogonality, and input scalability), our bacterial reservoir leverages *in vivo* metabolic complexity to solve regression and classification tasks. Future enhancements should focus on optimizing feature-to-nutrient mappings, and leveraging full growth curves as input to readout, but this will require the development of conventional reservoirs capable of modeling time-series data and supporting gradient backpropagation to optimize the input-prior. Other future work could include extending to new bacterial strains and metabolic models, and integrating this platform with emerging technologies such as neuromorphic architectures, and engineered living materials.

All together this work supports the idea that living systems possess intrinsic computational capacity, echoing the concept of cellular computing supremacy proposed by Goñi-Moreno[40]. Our findings position bacterial RC as a biomimetic and infrastructure-light alternative capable of performing meaningful and useful information processing without the need for complex genetic re-engineering.

Methods

Abbreviations

AMN: Artificial Metabolic Network, **ANN**: Artificial Neural Network, ***E. coli*** : *Escherichia coli*, **FBA**: Flux Balance Analysis, **GEM**: Genome-scale metabolic model, **MLP**: Multi-Layer Perceptron, **MLR**: Multi-Linear Regression, **RC**: Reservoir Computer, **R^2**: Regression coefficient. **SVM**: Support Vector Machine. **XGB**: XGBoost (parallel tree boosting).

Genome Scale Metabolic Models (GEMs)

A set of 10 GEMs were downloaded from the BiGG database[22]. Following a method outlined in Faure *et al*.[27] metabolic reactions having no specific directions were duplicated one forward one backward. Further information on these models are provided in the Supplementary Table S4.

Experimental training set for regression and classification tasks

Bacterial strain and culture media. *E. coli* K-12 MG1655 (accession number: NC_000913.3) was the strain used to construct this training set. All the media were based on a M9 minimal medium composed of (i) M9 salts at pH = 7.2 : 47.76 mM Na_2HPO_4 (Merck/Sigma-Aldrich, CAS number : 7558-79-4), 22.04 mM KH_2PO_4 (Merck/Sigma-Aldrich, CAS number : 7778-77-0), 8.56 mM NaCl (Merck/Sigma-Aldrich, CAS number : 7647-14-5), 18.7 mM NH_4Cl (Merck/Sigma-Aldrich, CAS number : 12125-02-09), (ii) magnesium sulfate : 4.08 mM $MgSO_4$ (Merck/Sigma-Aldrich, CAS number : 7488-88-9), (iii) calcium chloride dihydrate : 0.1 mM $CaCl_2{\cdot}2H_20$ (Merck/Sigma-Aldrich, CAS number : 10035-04-08), and (iv) trace elements : 0.4 mM $Na_2EDTA{\cdot}2H_2O$ (Merck/Sigma-Aldrich, CAS number : 6381-92-6), 0.03 mM $FeCl_3{\cdot}6H_2O$ (Merck/Sigma-Aldrich, CAS number : 7705-08-0), 6.16 µM $ZnCl_2$ (Merck/Sigma-Aldrich, CAS number : 7646-85-7), 0.76 µM $CuCl_2{\cdot}2H_2O$ (Merck/Sigma-Aldrich, CAS number : 7447-39-4), 0.42 µM $CoCl_2{\cdot}6H_2O$ (Merck/Sigma-Aldrich, CAS number : 7791-13-1), 1.62 µM H_3BO_3 (Merck/Sigma-Aldrich, CAS number : 10043-35-3), 0.08 µM $MnCl_2{\cdot}4H_2O$ (Merck/Sigma-Aldrich, CAS number : 13446-34-9), 1.65 µM $Na_2MoO_4{\cdot}2H_2O$ (Merck/Sigma-Aldrich, CAS number : 7631-95-0).

This minimal medium was supplemented with different combinations of nutrients selected from 3 sugars or acids, 20 proteinogenic amino acids and 5 nucleobases. The 3 sugars or acids used as carbon source and their final concentrations are : 44.4 mM D-(+)-glucose, 53.29 mM D-(+)-xylose (Merck/Sigma-Aldrich, CAS number : 58-86-6), 49.37 mM sodium succinate dibasic (Merck/Sigma-

Aldrich, CAS number : 150-90-3). Final amino acids and nucleic bases concentrations used are based on the EZ rich defined medium composition[41] : 0.8 mM L-alanine (Merck/Sigma-Aldrich, CAS number : 56-41-7), 5.2 mM L-arginine (Merck/Sigma-Aldrich, CAS number : 74-79-3), 0.4 mM L-asparagine (Merck/Sigma-Aldrich, CAS number : 70-47-3), 0.4 mM L-aspartic acid (Merck/Sigma-Aldrich, CAS number : 56-84-8), 0.1 mM L-cysteine (Merck/Sigma-Aldrich, CAS number : 52-90-4), 0.55 mM L-glutamic acid (Merck/Sigma-Aldrich, CAS number : 56-86-0), 0.615 mM L-glutamine (Merck/Sigma-Aldrich, CAS number : 56-85-9), 0.8 mM glycine (Merck/Sigma-Aldrich, CAS number : 56-40-6), 0.2 mM L-histidine (Merck/Sigma-Aldrich, CAS number : 71-00-1), 0.4 mM L-isoleucine (Merck/Sigma-Aldrich, CAS number : 73-32-5), 0.8 mM L-leucine (Merck/Sigma-Aldrich, CAS number : 61-90-5), 0.5 mM L-lysine (Merck/Sigma-Aldrich, CAS number : 56-87-1), 0.2 mM L-methionine (Merck/Sigma-Aldrich, CAS number : 63-68-3), 0.4 mM L-phenylalanine (Merck/Sigma-Aldrich, CAS number : 63-91-2),0.4 mM L-proline (Merck/Sigma-Aldrich, CAS number : 147-85-3),10 mM L-serine (Merck/Millipore, CAS number : 56-45-1), 0.4 mM L-threonine (Merck/Sigma-Aldrich, CAS number : 72-19-5), 0.1 mM L-tryptophan (Merck/Sigma-Aldrich, CAS number : 73-22-3), 0.2 mM L-tyrosine (Merck/Sigma-Aldrich, CAS number : 60-18-4), 0.6 mM L-valine (Merck/Sigma-Aldrich, CAS number : 72-18-4), 0.2 mM adenine (Merck/Sigma-Aldrich, CAS number : 73-24-5), 0.2 mM cytosine (Merck/Sigma-Aldrich, CAS number : 71-30-7), 0.2 mM uracil (Merck/Sigma-Aldrich, CAS number : 66-22-8), 0.2 mM guanine (Merck/Sigma-Aldrich, CAS number : 73-40-5), 0.2 mM thymidine (Merck/Sigma-Aldrich, CAS number : 50-89-5).

The objective was to develop a set of media capable of supporting a broad spectrum of maximal growth rates for *E. coli* K-12 MG1655. To this effect, different combinations of amino acids, nucleobases, and other carbon sources were designed to supplement a M9 minimal medium. Amino acids involved in the same biosynthesis pathway were grouped together as well as purines and pyrimidines (regarding nucleobases). 9 groups of amino acids were formed : Gly/Ser/Thr, Asn/Asp/Ala, Glu/Gln, Arg/Pro, Val/Leu/Ile, Lys, Cys/Met, His, Phe/Tyr/Trp (His and Lys were often associated together for the sake of simplicity, Glu and Gln were separated from Asn/Asp/Ala to ensure balanced groups). Each group was ranked according to their metabolic cost for *E. coli*, as defined in Zampieri *et al.*[42]. It has been demonstrated that amino acids with a low metabolic cost are consumed at a higher rate than the other ones[42]. The same study suggested that these low-cost amino acids could be used as carbon and energy sources[42]. Therefore, the hypothesis underlying the ranking was that amino acids with lower metabolic costs would have a greater contribution to the growth rate than the other ones. By varying the number of groups of amino acids supplemented in the medium as well as their rank, it was possible to generate a first set of media capable of supporting a wide range of maximal growth rates for *E. coli*. This range was further widened by designing new media supplemented with

amino acids but also nucleobases and different energy-rich or energy-poor carbon sources (D-(+)-glucose, D-(+)-xylose, succinate).

Bacterial growth monitoring. An *E. coli* K-12 MG1655 glycerol stock was streaked onto a M9 agar plate supplemented with 44.4 mM (0.8%) D-(+)-glucose (the composition is described in « Bacterial strain and culture media ») which was then incubated overnight at 37°C. A colony was used to inoculate a preculture of 10 mL of M9 liquid medium supplemented with 44.4 mM D-(+)-glucose in a 100 mL Erlenmeyer flask. The flask was incubated at 37°C under agitation at 180 rpm (Orbital floor shaker ZWY-B3222, Labwit) until the optical density at 600 nm of the culture reached 0.5 (measured on a UVisco V-1100D spectrophotometer, LC-Instru). 198 µL of each medium to test was prepared in each well of a clear flat-bottom 96-well plate (except the edge wells that were filled with sterile milliQ water to avoid the edge effect) and inoculated with the *E. coli* preculture so that the starting optical density at 600nm was estimated at 0.005. Each medium was tested in triplicate. The plate was then incubated in a plate reader (BioTek HTX Synergy, Agilent Technologies) at 37°C with continuous orbital shaking at 807cpm (1mm). Bacterial growth was assessed by monitoring the optical density at 600nm every 20 minutes for a period ranging from 16 to 24 hours.

Growth rate determination. A natural logarithm function is applied to the optical density measurements at 600 nm. First, the three timepoints between which the growth rate is maximal were determined using linear regression. The growth rate determination window was then extended in order to englobe all timepoints of the exponential phase. To do so, a fitting score was calculated between the linear model and the experimental data. The growth rate determination window was expanded in both directions (before and after the two initial timepoints) as long as the fitting score remained below a chosen threshold. The maximal growth rate was finally calculated using the timepoints within the final window. Each medium was tested in triplicate. The mean and standard deviation of each triplicate was computed. The training set file contained the media (in a binary form capturing the presence or absence of a specific medium metabolite) the average growth rate and the corresponding standard deviation. In addition to growth-rate we also computed lag time and maximal density (OD_{MAX}). We define the lag time as the last time point for which the slope did not exceed 25% of the growth rate, and OD_{MAX} as the average of the top 5 highest OD values.

FBA simulated training set

A set of 10,000 media were generated for each of the 10 GEM models listed in Supplementary Table S4. Each medium comprised two parts: a minimal medium and a variable medium. The minimal media were specific to each model, and defined as the set of metabolites necessary for the model to produce

a growth. The variable media were generated at random among the same 28 metabolites listed in the previous section. As with the experimental training set each variable medium was represented by a 28 bits vector. FBA was run for each medium and each model, all reaction rates (including the growth rate) were collected and stored in the training set files.

Building GEM reservoirs

Two kinds of GEM reservoirs were constructed. The first made use of *E. coli* iML1515 GEM. Following the procedure outlined in Faure *et al.*[27] an AMN was first trained on an FBA simulated training set comprising 10,000 instances (*cf.* training performances in Supplementary Figure S1). Among the solutions proposed in Faure *et al*. we chose the AMN-QP architecture which was inspired by Physics Informed Neural Networks (PINNs). As in PINNs, the loss function was composed of 3 terms capturing the fit to the true values (FBA calculated growth rates), the boundary condition constraints (lower and upper bounds for media uptake fluxes) and the stoichiometric constraint (SV = 0, where S is the GEM stoichiometric matrix and V the vector of fluxes from all GEM reactions). Additional information on AMN-QP architecture can be found in the Supplementary file 'Reservoir_Supplementary/AMN-QP architecture and training results'. The trained AMN was then used as the reservoir in the RC architecture depicted in Extended Data Fig. 1 and trained on the experimental training set composed of 280 different medium compositions and measured growth rates. In this RC architecture only the prior layer was trained to find the mapping between media compositions and media uptake rates. The iML1515 GEM RC model was saved and later used as a reservoir to solve regression and classification problems.

The second kind of GEM reservoir was constructed for all strains listed in Supplementary Table S4. These GEM reservoirs comprise an input layer representing a medium composition which is directly fed to an AMN. One note that the conversion from medium compositions to medium uptake rates is carried out by the neural layer of the AMN computing initial flux values (V^0 in Extended Data Fig. 1 see also Faure *et al*.[27]). As with iML1515 GEM we used the AMN-QP architecture with the loss function described above. All GEM RC models were saved and later used as reservoirs to solve regression problems.

Regression problems

The 10 regression problems downloaded from OpenML are listed in Supplementary Table S1. To compare conventional and physical reservoirs regression results, we used the MLR, MLP and XGB regressors from SciKitLearn[43] and acquired regression coefficients (R^2) on 5-fold cross validation sets.

We iterated 5 times to compute standard deviation. The hyperparameters were the default ones for MLR and XGB, for MLP we used one hidden layer of size twice the number of features, an Adam solver[44] with early stopping and adaptive learning rate, and a maximum number of iteration of 1000 (*cf*. Supplementary Table S3 for further details).

Classification problems

To generate data for the classification tasks shown in Fig. 4, we draw pairs from a uniform [0,1] distribution to get 500 samples with independence x_1 and x_2 features. Similar to Baltussen *et al.*[6], 8 different conditions (AND, OR, LINEAR, TRIANGLE, XOR, CIRCLE, SINE, DOTS) are used to calculate classes labeled 0 or 1. To produce Fig. 4, we used the SVM, MLP and XGB classifiers from SciKitLearn[43] to acquire accuracies on 5-fold cross validation sets (*cf*. Supplementary Table S3 for further details).

Regression and classification with RC

Below are various architectures for prior-input and post-readout layers for all conventional RC systems present in the study. In all cases the reservoir has fixed weights (W) and biases (b) and is transforming the output of the prior layer (X_{prior}) into an input for the post layer (X_{post}): $X_{post} = W X_{prior} + b$. The parameters W and b and those calculated after training the reservoir (*cf*. 'Building GEM reservoir' subsection).

Fixed prior. In this instance the prior-input layer is not trained, instead, features are one-to-one mapped to medium metabolite compositions. We recall we have 28 metabolites in our media each being either turned on or off (*cf*. 'Experimental training set' subsection). In the case of classification tasks (*cf*. Fig. 4), the two x_1 and x_2 input features are transformed into binary vectors of length 14 each, the binary vectors are initialized with zero values and the bit turned on is the one corresponding to the bin of the feature value (modulo 14). In the case of regressions tasks (*cf*. list in Supplementary Table S1) the mapping between features and medium metabolites is performed following permutation importance[67]. Precisely, individual features and medium metabolites are first ranked independently depending on their importance toward respectively the problem readout and the growth rate. This is achieved using the ScikitLearn[43] 'permutation_importance' function with a Random Forest Regressor model (with 100 estimators). Ranking is performed by sorting by decreasing importance. Next, the ith ranked feature is mapped to the ith ranked medium metabolites. To obtain an equal number of features and medium metabolites, when the number of features is lower than the number of medium metabolites, features are randomly selected and duplicated, conversely, when the number of features is greater than the number of medium metabolites, features with less importance

are ignored. Since the medium composition fed to the reservoir is a binary vector, the feature values need to be binarized; this is achieved by normalizing the feature values and applying a threshold to transform features into 0 (< threshold) and 1 (≥ threshold) values.

Generative prior. The purpose of the generative prior is to produce media compositions fed to the GEM reservoir. Since media compositions are binary vectors it is necessary to transform the features into 0 or 1 values. To that end the features are first processed by a hidden layer with linear activation functions, values are then fed to an output layer with sharp sigmoid activation functions. The sharp sigmoid is a standard sigmoid function where the operand is multiplied by a large constant value.

Selective prior. Instead of generating media compositions, the selective prior selects compositions from a predefined set. This set comprises media compositions previously used to grow *E. coli* and measure growth rates (*cf.* subsection 'Experimental training set'). To ensure each feature vector in the training set is associated with one of the experimental dataset media, we first process the features through several hidden layers to generate a vector sized to match the number of media. Applying a softmax activation function to this vector yields the probability of selecting each medium from the experimental dataset. However, we need to transform this vector so that the highest probability is set to 1 and all others to 0. To achieve this one-hot encoding, we use Gumbel-softmax[28], a differentiable approximation to argmax when the temperature approaches 0. During training, we start with a high temperature and gradually reduce it at each epoch. We found that using a logarithmic temperature decay provides better results than a linear decay.

Post-readout. A post-readout is trained when the GEM reservoir outputs all reaction rates, these rates are fed to a feedforward neural network which outputs the problem readout (y). When the GEM reservoir outputs only growth rates, we normalize both the growth rate and the problem readout and the problem readout equals the growth rate. In call cases the activation function of the post-readout is ReLU for regression tasks and sigmoid for classification tasks.

The hyperparameters corresponding to the above architectures are summarized in Supplementary Table S2. Additionally, for all runs carried out in this study and making use of the AMN reservoir , the number of epochs was set to 1000 for the selective prior architecture and 500 for all others, dropout was set to 0.25 and a learning rate of 10^{-4} was used using the Adam optimizer. Cross-validation was systematically used for all runs and all results reported in the current study (such as R^2 and Accuracy) are computed on 5-fold cross validation sets (*cf.* 'Cross validation' In Methods). The batch sizes were set to 10 when the training set comprised less than 1000 elements, and 100 otherwise. All runs were carried out in triplicates to compute standard deviations.

Cross validation

We used the same cross validation method for all regressors and classifiers on all training sets. The training set is randomly divided into X folds. One-fold (validation set) is selected to make prediction while training is performed on the remaining X-1 folds. The process is repeated selecting each fold as a validation set one after another. All regressions coefficients and accuracies reported in the manuscript and Supplementary Figures and Tables were computed using predicted values from the X aggregated validation sets. One notes that when the validation set contains only one element (X equals the size of the training set) our cross-validation method is Leave One Out (LOO).

Phenotype number and entropy calculations

To compute the number of phenotypes for the 10 strains considered in this study we used the 10,000 instances training sets described in the 'Training set' subsection and used to produce Extended Data Fig. 2. The number of different phenotypes is the number of different flux distributions of all reactions of the models. Several media can lead to the same flux distribution especially if one does not distinguish two distributions when they are separated by a distance lower than a predefined threshold. We chose an arbitrary threshold of 0.01 mol h^{-1} $cell^{-1}$ which is the limit of what can be measured experimentally using a spectrophotometer with growth rate. We note that the other thresholds we tested (0.1, 0.05, 0.005 and 0.001) produced trends similar to those in Extended Data Fig. 2. To obtain the number of phenotypes, we used the *fcluster* hierarchical clustering method from SciKitLearn[43] with a distance matrix between log-transformed flux distributions, and using complete linkage to ensure that all distributions within a cluster remained within the threshold of 0.01 mol h^{-1} $cell^{-1}$. The log transformation was applied to reduce the impact of highly variable flux magnitudes and better capture relative differences between distributions.

We also computed the phenotype Shannon entropy (H) for each strain using the formula: $H = \sum_i p_i \, log_2(p_i)$ where p_i is the probability of appearance of cluster *i*, in other words the number of flux distributions in cluster *i* divided by training set size (10,000)

COVID-19 patient plasma samples

All samples were extracted from BIOMARCOVID, a single-center, retrospective cohort study of patients hospitalized at the Grenoble-Alpes University Hospital (CHUGA) for COVID-19. This non-interventional, monocentric, retrospective study involving data and samples from human participants has been carried out in CHUGA according to French current regulation (*cf*. Supplementary Methods

'BIOMARCOVID cohort'). Subjects were all informed and did not oppose, written consent for participation was not required for this study in accordance with the national legislation and the institutional requirements. Additional information regarding the BIOMARCOVID cohort can be found in Supplementary Methods 'BIOMARCOVID cohort', anonymized patient data can be found in Supplementary files 'Covid_Data_MS_ML.xls' and 'Reservoir_Covid.xls'.

Metabolomics analyses on plasma samples

Metabolomics analysis was carried out on 81 patient samples (50 mild, 31 severe) from the BIOMARCOVID cohort. Mass spectrometry (MS) data were acquired using two liquid chromatography (LC) columns C18 and HILIC.

For C18, LC-MS analyses were performed using a UHPLC Vanquish Flex Binary system coupled with an Orbitrap Q-Exactive Plus mass spectrometer (Thermo Fisher Scientific). Chromatographic separation employed an Accucore RP-MS column with a gradient elution: phase A (water with 0.1% formic acid) and phase B (acetonitrile with 0.1% formic acid).

For HILIC, LC-MS analyses were performed using a U3000 liquid chromatography system coupled to a Q-Exactive mass spectrometer from Thermo Fisher Scientific (Courtaboeuf, France) fitted with an electrospray source operated in the positive and negative ion modes. HPLC chromatographic separations were performed on a Sequant ZICpHILIC 5 µm, 2.1 x 150 mm (HILIC) at 15°C (Merck, Darmstadt, Germany) equipped with an on-line prefilter (Thermo Fisher Scientific, Courtaboeuf, France).

Additional information is provided in Supplementary Method 'LC-MS analyses of COVID-19 plasma samples'. Metabolomics data is given in the Supplementary file 'Covid_Data_MS_ML'.

ML analyses on metabolomics data

Metabolomics data for 81 plasma samples were used as input to various classifiers. Data was normalized separately for the C18 and HILIC datasets by dividing all raw signal values by the average values found for the mild samples (*cf*. 'Covid_Data_MS_ML' Supplementary file). The two normalized datasets were merged and fed to the ML classifiers. The merged dataset comprised 624 MS signals (423 from C18 and 201 from HILIC). Additionally, we reduced the merged dataset selecting only MS signals corresponding to metabolites crossing the *E. coli* membrane. These medium metabolites were those found in the exchange reaction of iML1515 model. The mapping between the MS

metabolites/ions and iML1515 exchange metabolites was carried out using their InChI code which was found in the HMDB database from their names. The reduced medium dataset comprised 56 metabolites.

As classifiers, we used the SVM, MLP and XGB classifiers from the SciKitLearn[43] library. Default parameters were used in all cases and for MLP, the neural network comprised one hidden layer with a size twice the input size. MLP was trained for 10000 epochs with early stopping and the Adam optimizer. For all classifiers 20-fold cross validation was performed. Standard deviations were acquired iterating the cross-validation process 5 times.

Feature selection for classification

When performing classification in Fig. 5 and Extended Data Fig. 4, accuracies were calculated after feature selection. To find the final set of features (i.e. the set of selected time points in Fig. 5 or the set of selected MS signals in Extended Data Fig. 4) we removed features one at a time, at each iteration the deleted feature was the one leading to the highest accuracy after deletion. As this process is rather slow when the number of features in the training set exceeds 100, the features are first ranked using SciKitLearn[43] RandomForestClassifier/features_importance and only the top 100 most important ones are retained. The accuracies following selected features for COVID-19 mild vs. severe classification using MS data data are shown in Supplementary Figure S4. The accuracies following selected features for COVID-19 mild vs. severe classification with conventional reservoir data are shown in Supplementary Figure S5. The accuracies following selected features for COVID-19 negative vs. positive and mild vs. severe classification with *E. coli* physical reservoir predictions data are shown in Supplementary Figure S6.

Growth experiments on patients' plasma samples

Plasma deproteinization. The deproteinization process enables to inactivate the supplement system by removing its proteins from the plasma samples (*cf.* 'Optimisation of *E. coli* growth in plasma samples' in 'Supplementary Methods'). Samples were centrifuged (1h, 4000g) in Amicon Ultra-4 Centrifugal Filters (3 kDa MWCO, 4 mL of sample volume), resuspended and re-centrifuged (1h, 4000g) in the same centrifugal filters. Filtrates were transferred and aliquoted into 1mL sterile tubes (100µL of deproteinized plasma per tube). The plasma selected for this series of experiments came from patients that were following antibiotic treatment before the blood sampling. The absence of antibiotics in the plasma samples was verified during the growth *monitoring of the physical E. coli reservoir on plasma samples* (presence of antibiotics should result in the absence of bacterial growth

or an elongated lag phase). This procedure confirmed that no patients' plasma used in this study contained antibiotics.

Growth monitoring of the physical E. coli reservoir on plasma samples. E. coli K-12 MG1655 was cultured in all deproteinated patients' plasma samples (37 mild, 36 severe and 27 negative) in order to assess its ability to classify : (i) plasma samples from early symptomatic mild vs. severe patients and (ii) COVID-19 infected patients vs. negative blood donors.

A glycerol stock of *E. coli* was streaked onto a M9 agar plate (see « Bacterial strain and culture media » in the « Experimental training set for regression and classification tasks » section) supplemented with 3.89 mM (0.07%) D-(+)-glucose (to match a physiological glycemia) and 0.5% yeast extract which was then incubated overnight at 37°C. A colony was used to inoculate a preculture of 3 mL of M9 liquid medium supplemented with 3.89 mM (0.07%) D-(+)-glucose and 0.5% yeast extract in a 14 mL preculture tube. The tube was incubated at 37°C under agitation at 180 rpm (Multitron, Infors HT) until exponential phase with an optical density at 600 nm measured between 0.5 and 1 (UVisco V-1100D spectrophotometer, LC-Instru).

The plasma samples used in this study were collected in tubes containing ethylenediaminetetraacetic acid (EDTA), a commonly used anticoagulant to prevent blood clotting. EDTA chelates divalent cations such as $MgSO_4$ which is required for *E. coli*'s growth. Moreover, the bicarbonate buffer system naturally present in the plasma is ineffective here. In light of these constraints, optimal quantities of $MgSO_4$ and a phosphate buffer were added to the samples before inoculation in order to ensure optimal *E. coli* growth.

In each well of a clear flat-bottom 96-well plate (except the edge wells that were filled with sterile milliQ water to avoid the edge effect), 1.02 µmol of $MgSO_4$, 4.78 µmol of Na_2HPO_4 and 2.2 µmol of $KHPO_4$ were added to 50 µL of deproteinized plasma. Sterile milliQ water was added to reach 100µL. Each well was inoculated with the *E. coli* preculture so that the starting optical density at 600 nm was estimated at 0.005. The plate was then incubated in a plate reader (BioTek HTX Synergy, Agilent Technologies) at 37°C with continuous orbital shaking at 807cpm (1mm). Bacterial growth was assessed by monitoring the optical density at 600 nm every 10 minutes for a period ranging from 8 to 24 hours. A technical duplicate was conducted for each sample by repeating the full procedure.

Additional information regarding the 'Optimization of *E. coli* growth in plasma samples' are provided in Supplementary Methods and Supplementary Figure S10.

References

1. Purnick, P. E. M. & Weiss, R. The second wave of synthetic biology: from modules to systems. *Nat Rev Mol Cell Biol* **10**, 410–422 (2009).
2. Xiang, Y., Dalchau, N. & Wang, B. Scaling up genetic circuit design for cellular computing: advances and prospects. *Nat Comput* **17**, 833–853 (2018).
3. Tanaka, G. *et al.* Recent advances in physical reservoir computing: A review. *Neural Networks* **115**, 100–123 (2019).
4. Shen, B. *et al.* Proteomic and Metabolomic Characterization of COVID-19 Patient Sera. *Cell* **182**, 59-72.e15 (2020).
5. Ozawa, T. *et al.* Predicting coronavirus disease 2019 severity using explainable artificial intelligence techniques. *Sci Rep* **15**, 9459 (2025).
6. Baltussen, M. G., De Jong, T. J., Duez, Q., Robinson, W. E. & Huck, W. T. S. Chemical reservoir computation in a self-organizing reaction network. *Nature* **631**, 549–555 (2024).
7. Sumi, T. *et al.* Biological neurons act as generalization filters in reservoir computing. *Proc. Natl. Acad. Sci. U.S.A.* **120**, e2217008120 (2023).
8. Borkowski, O., Ceroni, F., Stan, G.-B. & Ellis, T. Overloaded and stressed: whole-cell considerations for bacterial synthetic biology. *Current Opinion in Microbiology* **33**, 123–130 (2016).
9. Meyer, A. J., Segall-Shapiro, T. H., Glassey, E., Zhang, J. & Voigt, C. A. Escherichia coli "Marionette" strains with 12 highly optimized small-molecule sensors. *Nat Chem Biol* **15**, 196–204 (2019).
10. Ho, J. M. L., Miller, C. A., Parks, S. E., Mattia, J. R. & Bennett, M. R. A suppressor tRNA-mediated feedforward loop eliminates leaky gene expression in bacteria. *Nucleic Acids Research* **49**, e25–e25 (2021).
11. Regot, S. *et al.* Distributed biological computation with multicellular engineered networks. *Nature* **469**, 207–211 (2011).
12. Daniel, R., Rubens, J. R., Sarpeshkar, R. & Lu, T. K. Synthetic analog computation in living cells. *Nature* **497**, 619–23 (2013).
13. Rizik, L., Danial, L., Habib, M., Weiss, R. & Daniel, R. Synthetic neuromorphic computing in living cells. *Nat Commun* **13**, 5602 (2022).
14. Moraskie, M. *et al.* Microbial whole-cell biosensors: Current applications, challenges, and future perspectives. *Biosensors and Bioelectronics* **191**, 113359 (2021).

15. Silva-Rocha, R. & De Lorenzo, V. Mining logic gates in prokaryotic transcriptional regulation networks. *FEBS Letters* **582**, 1237–1244 (2008).
16. Oppenheim, A. B., Kobiler, O., Stavans, J., Court, D. L. & Adhya, S. Switches in Bacteriophage Lambda Development. *Annu. Rev. Genet.* **39**, 409–429 (2005).
17. Scheres, B. & van der Putten, W. H. The plant perceptron connects environment to development. *Nature* **543**, 337–345 (2017).
18. Chen, H. *et al.* Emerging memristors and applications in reservoir computing. *Front. Phys.* **19**, 13401 (2024).
19. Fernando, C. & Sojakka, S. Pattern Recognition in a Bucket. in *Advances in Artificial Life* (eds. Banzhaf, W., Ziegler, J., Christaller, T., Dittrich, P. & Kim, J. T.) vol. 2801 588–597 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2003).
20. Carcano, L., Plebani, E., Pau, D. P. & Piastra, M. Pre-trainable Reservoir Computing with Recursive Neural Gas. Preprint at https://doi.org/10.48550/arXiv.1807.09510 (2018).
21. Srinivasan, S., Cluett, W. R. & Mahadevan, R. Constructing kinetic models of metabolism at genome-scales: A review. *Biotechnology Journal* **10**, 1345–1359 (2015).
22. King, Z. A. *et al.* BiGG Models: A platform for integrating, standardizing and sharing genome-scale models. *Nucleic Acids Research* **44**, D515–D522 (2016).
23. Orth, J. D., Thiele, I. & Palsson, B. Ø. What is flux balance analysis? *Nat Biotechnol* **28**, 245–248 (2010).
24. Monk, J. M. *et al.* iML1515, a knowledgebase that computes Escherichia coli traits. *Nat Biotechnol* **35**, 904–908 (2017).
25. O'Brien, E. J., Monk, J. M. & Palsson, B. O. Using Genome-scale Models to Predict Biological Capabilities. *Cell* **161**, 971–987 (2015).
26. Karloff, H. The Simplex Algorithm. in *Linear Programming* (ed. Karloff, H.) 23–47 (Birkhäuser, Boston, MA, 1991). doi:10.1007/978-0-8176-4844-2_2.
27. Faure, L., Mollet, B., Liebermeister, W. & Faulon, J.-L. A neural-mechanistic hybrid approach improving the predictive power of genome-scale metabolic models. *Nat Commun* **14**, 4669 (2023).
28. Jang, E., Gu, S. & Poole, B. Categorical Reparameterization with Gumbel-Softmax. Preprint at https://doi.org/10.48550/arXiv.1611.01144 (2017).
29. Danlos, F.-X. *et al.* Metabolomic analyses of COVID-19 patients unravel stage-dependent and prognostic biomarkers. *Cell Death Dis* **12**, 258 (2021).
30. Bi, X. *et al.* Proteomic and metabolomic profiling of urine uncovers immune responses in patients with COVID-19. *Cell Reports* **38**, 110271 (2022).

31. Roberts, I. *et al.* Untargeted metabolomics of COVID-19 patient serum reveals potential prognostic markers of both severity and outcome. *Metabolomics* **18**, 6 (2022).
32. Wood, S. N. *Generalized Additive Models: An Introduction with R*. (Chapman and Hall/CRC, 2017). doi:10.1201/9781315370279.
33. Fraser, D. D. *et al.* Metabolomics Profiling of Critically Ill Coronavirus Disease 2019 Patients: Identification of Diagnostic and Prognostic Biomarkers. *Critical Care Explorations* **2**, e0272 (2020).
34. López-Hernández, Y. *et al.* Targeted metabolomics identifies high performing diagnostic and prognostic biomarkers for COVID-19. *Sci Rep* **11**, 14732 (2021).
35. Correia, B. S. B. *et al.* ^{1}H qNMR-Based Metabolomics Discrimination of Covid-19 Severity. *J. Proteome Res.* **21**, 1640–1653 (2022).
36. Maeda, R. *et al.* Amino acid catabolite markers for early prognostication of pneumonia in patients with COVID-19. *Nat Commun* **14**, 8469 (2023).
37. Dambre, J., Verstraeten, D., Schrauwen, B. & Massar, S. Information Processing Capacity of Dynamical Systems. *Sci Rep* **2**, 514 (2012).
38. Martínez-Peña, R., Nokkala, J., Giorgi, G. L., Zambrini, R. & Soriano, M. C. Information Processing Capacity of Spin-Based Quantum Reservoir Computing Systems. *Cogn Comput* **15**, 1440–1451 (2023).
39. Dale, M. *et al.* The Role of Structure and Complexity on Reservoir Computing Quality. in *Unconventional Computation and Natural Computation* (eds. McQuillan, I. & Seki, S.) vol. 11493 52–64 (Springer International Publishing, Cham, 2019).
40. Goñi-Moreno, Á. Biocomputation: Moving Beyond Turing with Living Cellular Computers. *Commun. ACM* **67**, 70–77 (2024).
41. Neidhardt, F. C., Bloch, P. L. & Smith, D. F. Culture Medium for Enterobacteria. *J Bacteriol* **119**, 736–747 (1974).
42. Zampieri, M., Hörl, M., Hotz, F., Müller, N. F. & Sauer, U. Regulatory mechanisms underlying coordination of amino acid and glucose catabolism in Escherichia coli. *Nat Commun* **10**, 3354 (2019).
43. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. (2012) doi:10.48550/ARXIV.1201.0490.
44. Kingma, D. P. & Ba, J. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]* (2017).
45. Ebrahim, A., Lerman, J. A., Palsson, B. O. & Hyduke, D. R. COBRApy: COnstraints-Based Reconstruction and Analysis for Python. *BMC Syst Biol* **7**, 74 (2013).

Data and Code Availability

Will be available upon journal publication

Acknowledgements

Authors would like to acknowledge funding provided by the ANR funding agency grant numbers ANR-22-PEBB-0008 (PEPR B-BEST France 2030 program) and ANR-21-CE45-0021-01 (AMN project) and the UE HORIZON BIOS program (grant number 101070281). J.L.F is getting support from all programs, A.L.G. is supported by the ANR-21-CE45-0021-01, the ANR-24-CE44-1190 project, the Foundation University Grenoble Alpes and Foundation Air Liquid (BIOMARCOVID Project), P.A. by the AMN project, A.H. by the BIOS project, P.M. by the PEPR B-BEST project. We thank Alexandra Martin (INRAE, University of Paris Saclay) for her help in acquiring *E. coli* growth curves for different media, and we thank Claire Roman (Campus Innovation Paris, Air Liquide & IRIMAS, University Haute Alsace) for her help and suggestions regarding the statistical analyses.

Authors Contribution

P.A. performed all microbiology work and generated all training sets used in the study growing *E. coli* K-12 MG1655 for different media and on COVID-19 samples. T.N.A.H. performed the statistical analysis, created and maintained the GitHub implementation and tested all codes. P.M. wrote the clustering code and performed phenotype clustering. O.E. and A.L.G wrote the clinical trial. O.E. and S.B. collected the plasma samples. F.F., F.C., and A.L.G performed metabolomics experiments. J.L.F designed and supervised the study, acquired the fundings, and wrote the reservoir and machine learning codes. All authors contributed to writing manuscript sections, reading and editing the whole manuscript. All authors read and approved the manuscript.

The authors declare no competing interests.

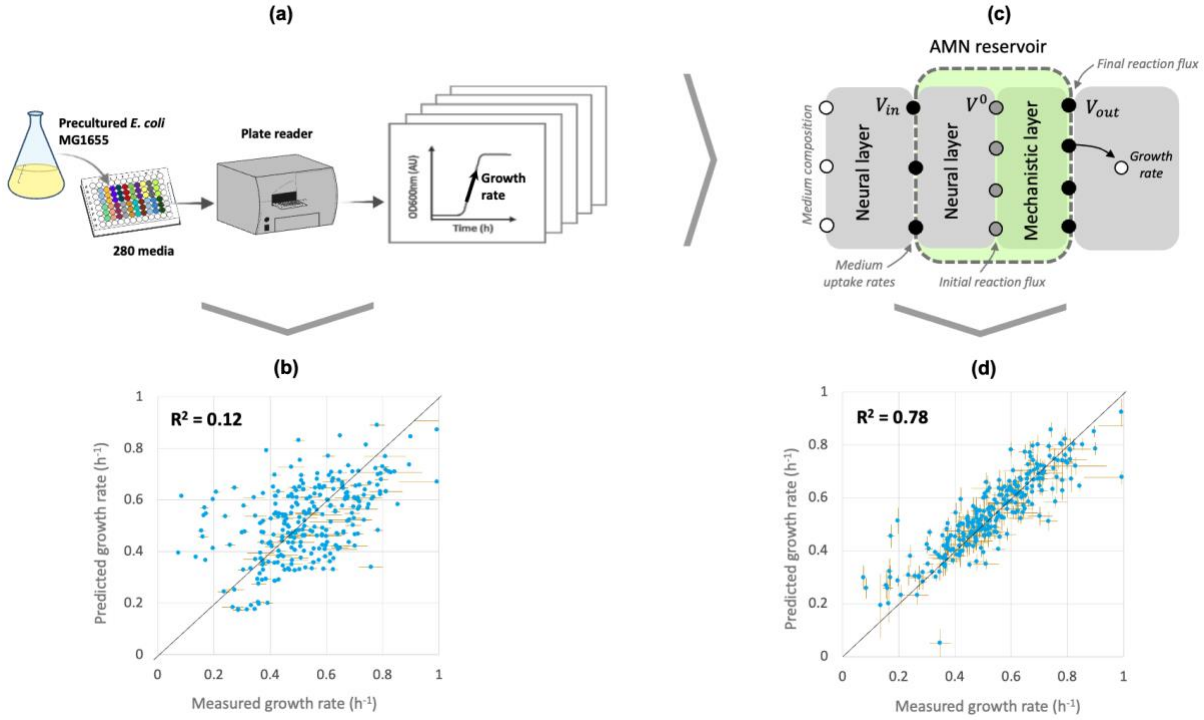
Correspondence and requests for materials should be addressed to Jean-Loup.Faulon@inrae.fr.

Supplementary Information

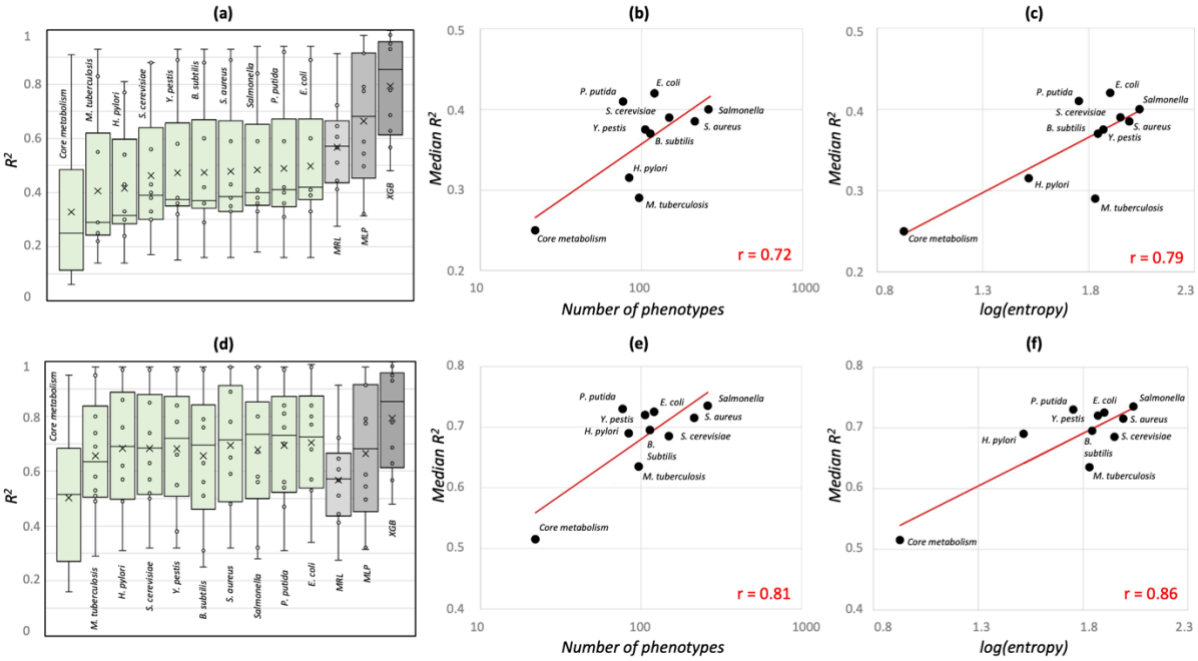
Reservoir_Supplementary.pdf, for Supplementary Methods, Tables and Figures.

Reservoir_GEM.xls, Reservoir_Regression.xls, Reservoir_Classification.xls, Covid_Data_MS_ML.xls and Reservoir_Covid.xls for the data used to run the code and to generate all Figures and Tables.

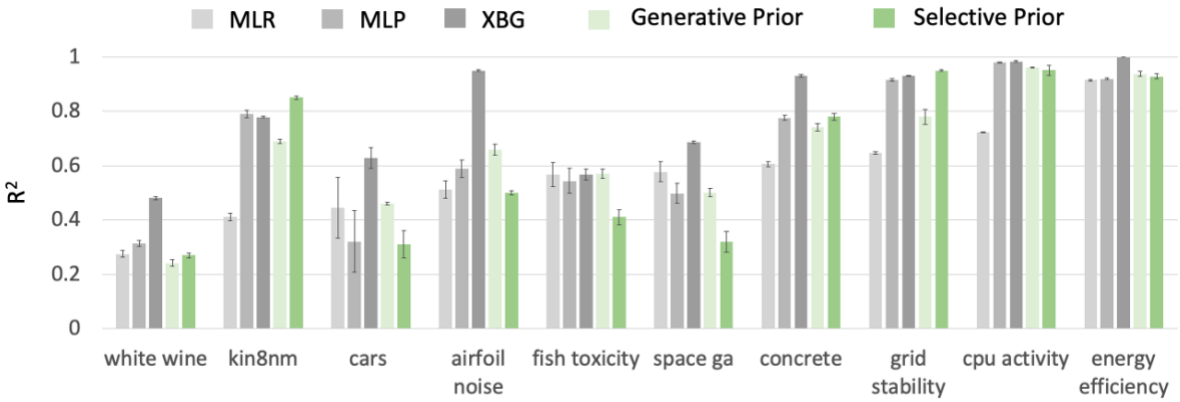
Extended Data (Figures)



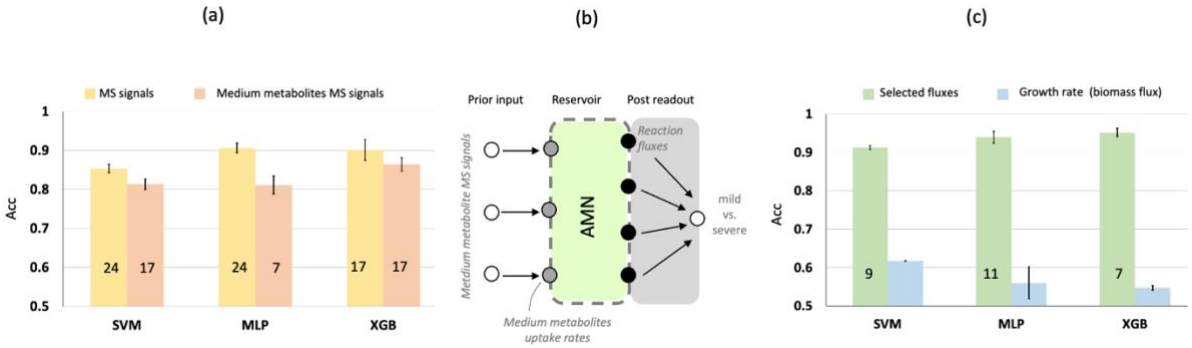
Extended Data Fig. 1. GEM-FBA vs. RC growth rate predictions for *E. coli* grown on 280 different media compositions. *(a) Experimental setup where E. coli K-12 MG1655 was pre-cultured and grown on M9 minimal media with 28 added metabolites. Growth rates were measured using a plate reader. (b) Correlation between predicted and measured growth rates using the GEM-FBA approach with the CobraPy library[45], showing a poor correlation coefficient ($R^2 = 0.12$). Prior to running FBA, the binary media compositions were scaled by a constant to produce media uptake rates. The constant was searched greedily in order to maximize the regression coefficient between measurements and predictions, the optimal constant value was found to be 6.5. (c) Schematic of the developed RC system with an input layer mapping media composition to uptake fluxes, a reservoir performing FBA calculations, and a post-readout layer printing growth rates. The reservoir, making used of a Neural Layer and an ANM (Artificial Metabolic Network) is pre-trained as described in Supplementary Method 'AMN-QP architecture and training results'. (d) Correlation between predicted and measured growth rates using the RC system with AMN, showing a substantial improvement ($R^2 = 0.78$). As in panel (b), predicted growth rates were calculated using CobraPy but in this case fed with the medium uptake rates (V_{in}) and computed during training. All points were acquired on aggregated cross-validation sets (5-fold); error bars (both measured and predicted) represent standard deviations over 3 repeats. Data are in the Supplementary file 'Reservoir_GEM'.*



Extended Data Fig. 2. Regression problem performances for several bacterial conventional RC systems. *Regression problem performances for several bacterial conventional RC systems. RC was used for 10 GEMs listed in Supplementary Table S4 and for the 10 regression problems listed in Supplementary Table S1. (a) Box plots for regression coefficients (R^2) obtained for each bacterium using a fixed prior layer. (b) Median regression coefficient vs. number of phenotypes when using a fixed prior. The Pearson correlation coefficient (r) between median R^2 values and the linear fit (y = 0.04 x + 0.08) is 0.72. (c) Median regression coefficient (R^2) vs. log(entropy) when using a fixed prior. The Pearson correlation coefficient between R^2 values and the linear fit (y = 0.09 x + 0.12) is 0.79. (d) Box plots for regression coefficients obtained for each bacterium using a generative prior layer. (e) Media regression coefficient (R^2) vs. number of phenotypes when using generative prior. The Pearson correlation coefficient (r) between R^2 values and the linear fit (y = 0.19 x + 0.31) is 0.81. (f) Median regression coefficient vs. log(entropy) when using a generative prior. The Pearson correlation coefficient between R^2 values and the linear fit (y = 0.40 x + 0.38) is 0.86. Entropy calculation is given in Methods. All R^2 values were acquired on 5-fold cross-validation sets for 3 repeats. Data are in the Supplementary file 'Reservoir_Regression'.*



Extended Data Fig. 3. Regression problem performances with a conventional twin of the physical *E. coli* RC. *The regression problems are those listed in Supplementary Table S1 and the prior methods are described in 'Regression and classification with RC' in the Methods section. R^2 values were acquired on aggregated cross-validation sets (5-fold) standard deviations were computed from three repeats. Data are in the Supplementary file 'Reservoir_Regression'.*



Extended Data Fig. 4. Classifying COVID-19 plasma samples with metabolomics data and with *E. coli* conventional reservoir. *(a) Classifying (mild vs. severe) COVID-19 with metabolomics data. All: Accuracies using all 624 metabolite signals detected by MS for 81 patients. Medium: Accuracies using 56 MS signals for metabolites expected to cross the E. coli cell wall. Accuracies are calculated for 20-fold cross validation and using feature selection (cf. 'Feature selection for classification' in Methods and Supplementary Fig. S4). The numbers indicated on the bar charts are the numbers of features selected. The parameters used for the classifiers (SVM, XGB and MLP) are given in 'ML analyses on metabolomics data' in the Methods section. (b) The conventional RC takes as input medium metabolite MS signals, these are fed to a reservoir initially trained on FBA simulations with the GEM iML1515. The reservoir outputs the flux distribution over all reactions (3682 reactions) of the GEM, the distribution is fed to a post-readout performing a classification. This conventional RC is used to predict outcome (mild vs. severe) taking as input metabolite MS signals. (c) Classification is performed using all reaction fluxes (or just the reaction biomass flux) produced by the AMN reservoir of panel b when fed with the metabolite MS signals of the 81 COVID-19 patients. The numbers indicated on the bar charts are the numbers of features selected (cf. 'Feature selection for classification' in Methods and Supplementary Fig. S5). All data are in the Supplementary file 'Covid_Data_ML'.*