



HAL
open science

What Makes a Bacterial Model a Good Reservoir Computer? Predicting Performance from Separability and Similarity

Laura Alonso Bartolomé, Jean-Loup Faulon, Xavier Hinaut

► To cite this version:

Laura Alonso Bartolomé, Jean-Loup Faulon, Xavier Hinaut. What Makes a Bacterial Model a Good Reservoir Computer? Predicting Performance from Separability and Similarity. 2026. <hal-05592447>

HAL Id: hal-05592447

<https://hal.science/hal-05592447v1>

Preprint submitted on 16 Apr 2026

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY-NC 4.0 - Attribution - Non-commercial use - International License

What Makes a Bacterial Model a Good Reservoir Computer? Predicting Performance from Separability and Similarity

Laura Alonso Bartolomé^{1,2,3,4}, Jean-Loup Faulon^{1*}, and Xavier Hinaut^{2,3,4*}

¹University Paris Saclay, INRAE, AgroParisTech, MICALIS Institute, 78350 Jouy-en-Josas, France

²Centre Inria de l'Université de Bordeaux, France

³Institut des Maladies Neurodégénératives, Université de Bordeaux, CNRS, UMR 5293

⁴LaBRI, Bordeaux INP, CNRS, UMR 5800, France

*co-supervised the study; laura.alonso-bartolome@inria.fr

Abstract

Biological systems are promising substrates for computation because they naturally process environmental information through complex internal dynamics. In this study, we investigate whether bacterial metabolic models can act as physical reservoirs and whether their computational performance can be predicted from dynamical properties linked to separability and similarity. We simulated the growth dynamics of five bacterial species, one yeast species, and 29 *Escherichia coli* single-gene deletion mutants using dynamic flux balance analysis (dFBA), with glucose and xylose concentrations as inputs and growth curves as reservoir states. Computational performance was assessed on random nonlinear classification tasks using a linear readout, while reservoir properties linked to separability and similarity were characterised through kernel and generalisation ranks computed from growth-curve state matrices. Several microbial models achieved high classification accuracy, showing that bacterial metabolic dynamics can support nonlinear computation. Clear differences were observed between species, with some models converging more rapidly and others reaching higher maximum accuracy, revealing a trade-off between convergence speed and peak performance. In contrast, all *E. coli* mutants were dominated by the wild-type model, suggesting that gene deletions reduce the dynamical richness required for efficient computation. The difference between kernel and generalisation ranks was generally associated with improved accuracy, but deviations across models and sensitivity at low rank values limited its predictive power in practice. Overall, these results show that bacterial metabolic models constitute promising substrates for reservoir computing and provide a first step towards identifying microbial strains with favourable computational properties for future experimental implementations.

Submission type: **Full Paper**

Code repository will be available after acceptance. Data/Code available at: github.com/lauraalonsobart/alife2026-bacteria-reservoir-rank-difference

Introduction

Biological systems are attractive substrates for computation because they naturally process environmental information through complex internal dynamics. In bacteria, this potential has long motivated efforts in synthetic biology to engineer cells capable of implementing specific information-processing behaviours (Ahavi et al., 2026b). Landmark studies demonstrated that synthetic genetic circuits could produce dynamical functions such as oscillations and bistability (Elowitz and Leibler, 2000), (Gardner et al., 2000), and subsequent work has extended these approaches towards increasingly sophisticated forms of microbial computing (Ahavi et al., 2026b). However, such bottom-up strategies rely on genetic engineering, often requiring substantial optimisation for each new application and facing important limitations in scalability, including metabolic burden and lack of orthogonality (Ahavi et al., 2026b).

Physical reservoir computing (see Figure 1) offers an alternative approach in which computation emerges from the intrinsic dynamics of a system rather than from redesigning its internal architecture (Tanaka et al., 2019; Nakajima, 2020). In the reservoir computing framework (Lukoševičius and Jaeger, 2009), inputs perturb a fixed dynamical system, called the reservoir, and only a readout layer is trained to map the resulting states to the desired output (Tanaka et al., 2019; Nakajima, 2020). Because the reservoir itself does not need to be trained, this framework can exploit a wide variety of physical substrates with sufficiently rich nonlinear dynamics (Tanaka et al., 2019; Nakajima, 2020). More broadly, reservoir computing has also been discussed from an evolutionary perspective, suggesting that natural systems may already possess properties compatible with information processing in this sense (Seoane, 2019).

In this context, bacterial populations are promising candidates for physical reservoir computing. Rather than modifying their regulatory, signaling or metabolic networks, one can directly exploit their natural dynamics as computational resources, as illustrated in Figure 1. A recent study by Ahavi et al. (2026a) demonstrated this idea experimentally by using nutrient inputs and growth dynamics of *Escherichia coli*

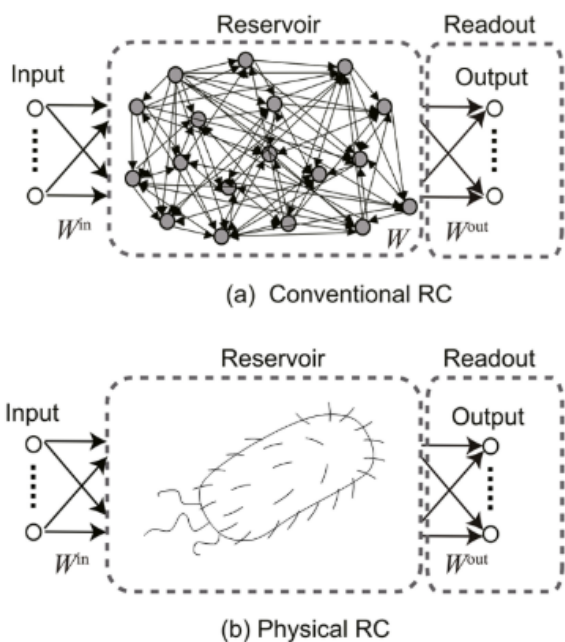


Figure 1: Conventional and physical reservoir computing frameworks. In conventional RC (a) the reservoir is a fixed weight system. In physical RC (b) the reservoir is a physical system with complex dynamics, for instance a bacterial population. Adapted from Tanaka et al. (2019).

to perform classification and regression tasks. The same study also showed a practical sensing application, in which bacterial responses enabled discrimination between plasma samples from COVID-19 patients that later developed moderate or severe forms of the disease (Ahavi et al., 2026a).

These results raise a broader question: beyond *E. coli*, what makes a microbial system suitable for reservoir computing? More specifically, can the computational performance of a bacterial strain be predicted from the properties of its internal dynamics? Related questions have previously been investigated for neural circuit models by Legenstein and Maass (Legenstein and Maass, 2007), who showed that computational performance depends on the separability/similarity trade-off: a balance between being able to distinguish meaningful differences without being disturbed by noise. Particularly effective regimes are supposed to occur near the “edge of chaos” (Legenstein and Maass, 2007).

In this paper, we apply an analogous analysis to bacterial models to determine whether these dynamical properties can also predict computational capacity in bacterial reservoirs. Our aim is to characterise bacterial strains using metrics related to kernel rank (separability) and generalisation rank (similarity), and to assess whether these metrics are informative predictors of reservoir performance. In the long term, this could help identify optimal bacterial strains and support the development of new biological experimental setups.

Methods

The proposed implementation of bacterial reservoirs is inspired from Ahavi et al. (2026a). As shown in Figure 2, nutrients are given as input and the growth dynamics of the bacterial population are interpreted as the reservoir state. In the present study, however, only computational models of the different bacterial species are tested. We used *dynamic flux balance analysis* (dFBA) to predict growth curves, which act as reservoir states. We use such growth curves states to measure computational performance and to create the state matrices for the separability and similarity measures. To produce the simplest possible experiment that could get significantly different responses from the bacteria we take only two different nutrients as inputs: glucose and xylose. There were found to have the highest effect on growth rate among the 28 metabolites tested in Ahavi et al. (2026a) and are known to produce what is known as diauxic growth. *Diauxic growth* is a growth in two phases that is caused by the bacteria having different preference for the sugars¹. This shift when the bacteria change from consuming one sugar to the other can produce more complex dynamics and improve the amount of information we can get from the growth curve.

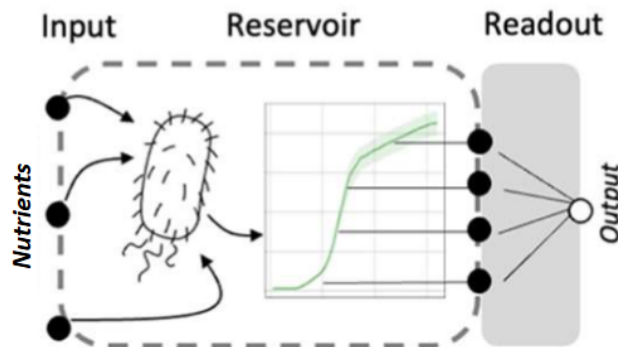


Figure 2: Bacterial reservoir setup. Nutrients (e.g., sugars) are given directly as inputs to the bacterial population (model). Different growth curves were generated using dynamic flux balance analysis (dFBA) depending on these nutrients. At the end of the simulation, for each model, 200 time point values of the growth curve are fed to the readout to predict the output.

Growth curve simulation: dFBA models

To simulate the behavior of the bacteria and generate the growth curves from the nutrient concentration data we implemented a dFBA pipeline using COBRAPy (Ebrahim

¹Various diauxic growth shifts depending on the type of sugar can be seen at https://en.wikipedia.org/wiki/Diauxic_growth#/media/File:Monod's_Diauxic_growth.gif

et al., 2013). FBA uses *genome scale metabolic models* (GEMs) of organisms to simulate cell metabolisms and mathematically predict the fluxes of metabolites (including the increase in biomass) in steady state conditions under different constraints. Dynamic FBA involves simulating the evolution of the bacterial population over time by tracking the concentration of some molecules and iteratively running FBA simulations while modifying the constraints related to these molecules. In this study, we track the concentration of glucose and xylose.

The growth curve simulation works as follows:

- Uptake fluxes – how fast a metabolite is being transported into the cells – are calculated from the current concentration of glucose and xylose, and set in the model as lower bounds for the corresponding uptake reactions.
- FBA simulation returns growth rate and fluxes for glucose and xylose.
- These values are used to calculate the biomass growth in that time interval and the consumption of glucose and xylose. This allows to calculate the concentrations to be used in the next iteration.

The uptake fluxes for glucose and xylose were defined by the following kinetic equations:

$$V_g = V_{g,max} \frac{G}{K_g + G}$$

$$V_x = V_{x,max} \frac{X}{K_x + X} \frac{1}{1 + \frac{G}{K_{ig}}}$$

The transport of both sugars into the cells is assumed to follow Michaelis-Menten kinetics with an extra glucose dependent inhibitory term on xylose. The value for the related parameter K_{ig} was selected to ensure no xylose consumption in the presence of glucose, as the later is known to produce strong catabolic inhibition of other sugars (Kaplan et al., 2024). That is, when glucose is available, cells consume it first and only move onto consuming other sugars once glucose runs out. Additionally, the maximum oxygen uptake rate was limited to $5 \text{ mmol}/g_{DW} \cdot h$ to reflect the low oxygen conditions cells would likely be subjected to in a plate reader which is where this kind of experiments are usually carried out. The parameters selected for this simulation and their sources can be found on Table 1. Simulations were carried out for an experiment duration of 20 hours with a timestep dt of 0.1 hours, resulting in 200 timepoints.

Measuring computational performance

To act as a measure of computational performance the reservoir models were fed inputs consisting of different concentrations of two nutrients: glucose and xylose. Twenty combinations of concentrations were drawn from a uniform distribution of values between 0 and 45 mmol for both glucose

Parameter	Value	Source
$V_{g,max}$	10.0	Mahadevan et al. (2002)
K_g	0.015	Mahadevan et al. (2002)
$V_{x,max}$	9.0	Henson and Hanly (2014)
K_x	0.01	Henson and Hanly (2014)
K_{ig}	0.01	Kaplan et al. (2024)

Table 1: Nutrient uptake parameters for dFBA simulations.

and xylose. The bacterial models were fed noisy versions of these templates. Then, given the growth curves used as reservoir states, a linear readout was trained to perform 100 random classification tasks (i.e., 100 out of the 2^{20} possible binary classifications of the templates). Statistically, all 100 of these tasks are expected to be nonlinear in two dimensions, as random binary labeling of a 20-point dataset in a high-dimensional feature space almost surely results in a nonseparable decision boundary like in the example in Figure 3). To implement the readout we used the ridge classifier (*RidgeClassifierCV*) from scikit-learn (Pedregosa et al., 2011), which optimize regularization parameter with cross-validation (alphas ranging from 10^{-3} to 1).

In Figure 4, we see a representation of the noisy input versions of the templates. The added noise was modelled as a small Gaussian perturbation of the concentration levels (mean 0, std 0.2), representing experimental error that may occur during the preparation of media with defined nutrient concentrations (IEC et al., 1995). Mean accuracy for 100 such tasks was reported for each model from a set of 400 training samples and 100 test samples.

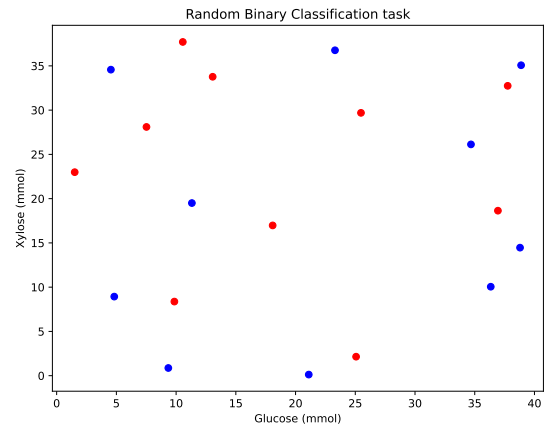


Figure 3: Example of a random binary classification task with 20 points.

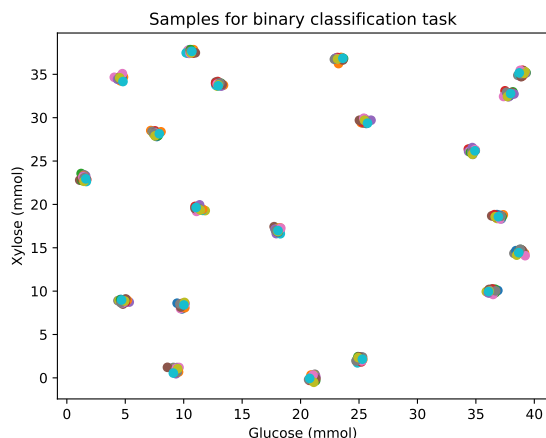


Figure 4: Input samples used to measure computational performance. The samples are aggregated in small clusters comprising 20 noisy inputs around their corresponding templates. This level of noise mimics experimental preparation error.

Kernel and generalisation ranks

We assessed the computational properties of each bacterial model using two rank-based metrics derived from the framework introduced by Legenstein and Maass (2007): *kernel rank*², which reflects separability, and *generalisation rank*³, which reflects similarity. More specifically, kernel rank reflects the ability of the system to produce linearly separable states, whereas generalisation rank reflects the ability to produce similar states in response to noisy variations of the same input. Accordingly, lower generalisation rank values indicate greater similarity among the corresponding states, whereas higher values indicate lower similarity.

For each input condition, the bacterial model generated a growth-curve time series. For a given input ensemble, these response trajectories were assembled into a state matrix, with one row per input, and columns corresponding to the sampled time points of the growth curve. The rank of this matrix was then used to quantify the diversity of the reservoir responses.

To estimate separability, 100 templates were randomly drawn from the previously mentioned uniform concentration distributions and applied to the bacterial models. The kernel rank was computed as the rank of the resulting state matrix.

²Legenstein and Maass (Legenstein and Maass, 2007) use the term *kernel quality*. We instead use *kernel rank* to emphasise that the measure is defined as the rank of a state matrix and to simplify later discussion of rank differences.

³Legenstein and Maass (Legenstein and Maass, 2007) use the term *generalization capability*. We instead use *generalisation rank* to avoid confusion with generalisation in the machine learning sense of predictive performance on unseen data.

A high kernel rank indicates that distinct inputs are mapped to distinct, linearly independent response trajectories, which corresponds to strong separability. To estimate similarity, 5 templates were selected and 20 noisy variants were generated from each template, yielding 100 input samples in total. The generalisation rank was computed from the corresponding state matrix. In this case, lower rank values indicate that similar inputs produce similar response trajectories, reflecting stronger generalisation⁴. These two measures were used jointly to characterise the balance between discriminating meaningful differences in the input space and maintaining robustness to small perturbations.

Metabolic models

The genome scale metabolic models (GEMs) used in this study include five different bacterial species (*Escherichia coli*, *Bacillus subtilis*, *Staphylococcus aureus*, *Salmonella* pan-reactome and *Yersinia pestis*) and one yeast (*Saccharomyces cerevisiae*). Even though *S. cerevisiae* is not a bacterial species we decided to add it to the study because it is also a unicellular organism that is very commonly used in microbiology labs and it is interesting to see if there are notable differences in eukaryotes. The specific strains and model IDs can be found in table 2 in the appendix. It is important to note that the same uptake reactions were used for all models, even though the parameters were extracted from scientific publications concerning *E. coli*. This is because glucose and xylose transport parameters are not available for all species, and different uptake kinetics might interfere when drawing comparisons.

Additionally, 29 variants of the *E. coli* GEM that reflect mutations relevant to the experimental conditions were also tested. These mutants were identified using the single gene deletion method from COBRAPy, which enabled the computation of growth rates for all possible single gene deletions. To focus on biologically meaningful phenotypes, only deletions that resulted in growth rates between 0.05 and 0.95 times that of the wild-type model were retained. Deletions with no significant effect on growth or those that were excessively deleterious (leading to near-zero growth) were excluded.

Observing evolution over time

The default time for the simulations was 20 hours so that there was enough time to capture the full growth dynamics of most models even if some had slower growth. With a timestep of 0.1 hours, this results in 200 time points which constitute the state of the reservoir for each input. This state can then be used to solve the tasks and measure the accuracy of the model, or compute the kernel and generalisation

⁴To understand the link to the VC-dimension, one can refer to Legenstein and Maass (2007) and this interactive tutorial from Lauer (2014): <https://mlweb.loria.fr/book/en/VCdimhyperplane.html>

ranks. However, while we evaluated the accuracy achieved by the models over a 20-hour experiment, it is also important to determine whether comparable accuracy can be obtained with a shorter experiment. To investigate this, we recalculated all metrics using progressively increasing numbers of time points, up to the original 200, allowing us to track how model accuracy and the ranks evolved over time.

Results

In Figure 5, we can see the evolution of mean accuracies for the classification tasks of the different microbial models. All species start at a baseline level around 0.6 of accuracy: this matches the accuracy obtained when using only the input without the bacteria model: i.e., the two sugar concentration values are directly fed to the linear readout. As the experiment time increases and growth dynamics are unveiled, the linear readout is able to produce better accuracies for these non-linear classification tasks with some species reaching almost perfect accuracy. This demonstrates the ability of the bacterial models to perform non-linear computations.

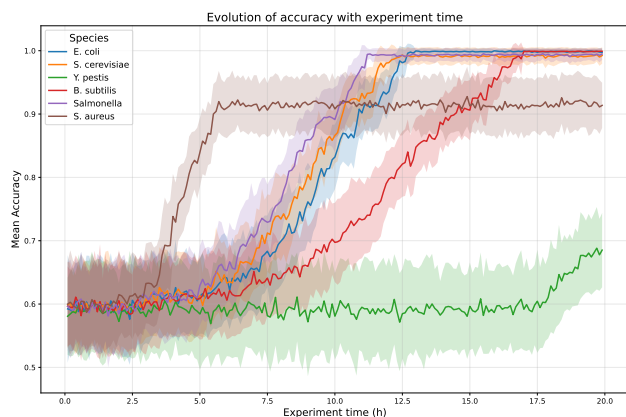


Figure 5: Accuracy over time for the different microbial species and one yeast species. Shaded areas represent the standard deviation around each curve. The x-axis spans 20 h, with one time point every 0.1 h.

Visualizing the evolution over time allows us to see differences between the four models that reach near perfect accuracy in the 20 hour experiment. While *Salmonella* and *B. subtilis* perform similarly in longer experiments, *Salmonella* is able to reach this level around five hours earlier, which is preferable if minimal time is the main objective. We also see that even though the *S. aureus* model has the fastest growing accuracy it reaches a plateau at approximately 0.92 and is unable to improve further. Lastly, *Y. pestis* is the most underperforming of the evaluated species, only starting to improve accuracy from the baseline level at around 17.5 hours.

Out of the 29 *E. coli* mutants that were evaluated, we only get a handful of different behaviours (Figure 6). Firstly be-

cause some of these genes encoded proteins that are part of bigger enzymatic complexes like NADH dehydrogenase, ATP synthase and cytochrome oxidase so the elimination of either of them results in the same functional consequence which is the elimination of those reactions. In the figure, mutants that affect the same reaction are represented by the same color. Secondly, because some others encode enzymes that are involved in the same pathway, like Glyceraldehyde 3-phosphate Dehydrogenase (GAPD) and Phosphoglycerate Kinase (PGK) being consecutive steps of the more common Embden–Meyerhof–Parnas glycolysis pathway. The absence of either of these enzymes diverts the flux of sugar degradation to the Entner-Doudoroff pathway producing similar results.

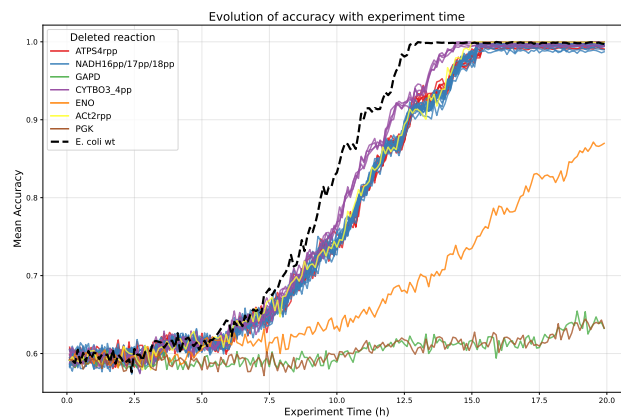


Figure 6: Evolution of accuracy with experiment time for the *E. coli* mutants. The color of a mutant corresponds to the reaction that is affected by the gene deletion. The legend includes the BiGG ID of the affected reactions which comprise several mutants in the case of big enzymatic complexes. The data for the wild type strain is represented by the black dashed line. Standard deviations are not shown for visual clarity, a version with standard deviations can be found in the appendix (Figure 10).

None of the mutants produce better results than the wild type strain but some are clearly more damaging to the models ability to produce dynamics that allow accurate classification of the inputs. The worse performing mutants (PGK, GAPD and, to a lesser extent, ENO) are those that affect the glycolysis pathway which integrates the input sugars into the broader metabolic network. Mutants that affect enzymes involved in other pathways like the currency metabolites ATP and NADH can slow down growth but seem to not so critically affect the richness of the internal dynamics.

Separability and similarity

We then analysed the temporal evolution of the separability and similarity properties. As shown in Figure 6, the kernel

rank, the generalisation rank, and their difference—which we term the rank difference—change over time. Consistent with expectations, the kernel rank rises more rapidly than the generalisation rank, while accuracy increases as the difference between the two becomes larger.

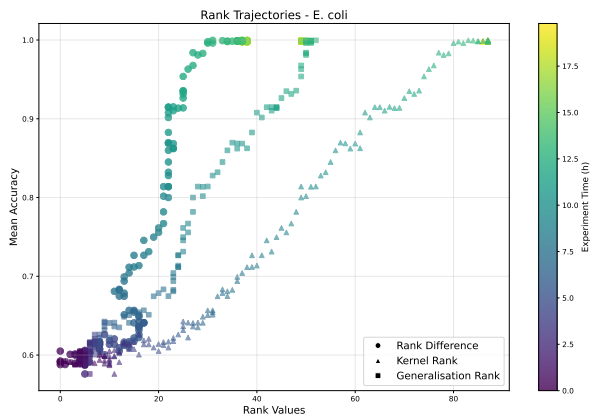


Figure 7: Trajectories of kernel and generalisation ranks individually next to their difference. Color gradient indicates time.

Figure 8 shows the trajectory of the rank difference for the different microbial species. Two species stand out. *S. aureus* reaches relatively high accuracies at low rank differences, with a sharp increase in accuracy occurring as the rank difference rises only slightly from 12 to 14. By contrast, *Y. pestis* appears unable to generate large rank differences, although its slight improvement in accuracy at the end of the experiment is still achieved at very low rank difference values. The other models follow broadly similar trajectories, even if some take longer than others to reach higher accuracies. *B. subtilis* deviates slightly, requiring a somewhat larger rank difference to attain accuracy levels comparable to those of the other models.

Lastly, Figure 9 shows the trajectories of the *E. coli* mutants. In this case we see a clear aggregation of most models on a similar trajectory. The only exception is the enolase mutant which underperforms at equal rank differences when compared to the other models. Conversely, while GAPD (brown) and PGK (green) never reached higher rank differences or accuracies they do not deviate from the general trajectory.

Over time, the rank difference exhibits an S-shaped pattern, already apparent in the previous figure and even clearer here. This temporary reversal in the trend arises from the diauxic shifts of two templates, and their corresponding 40 samples, used for the generalisation rank calculation, occurring in close succession. Because this corresponds to a dynamically informative regime, the generalisation rank temporarily rises more rapidly, reducing the rank difference. A

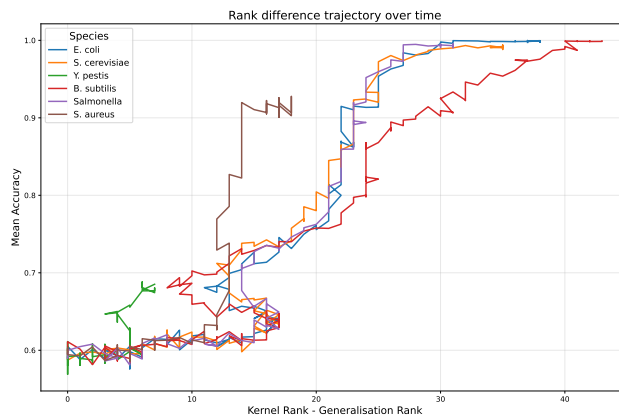


Figure 8: Relationship between rank difference and mean accuracy for classification tasks in the different microbial models.

similar effect occurs at later time points, but it is only faintly visible in the ENO mutant. This suggests that the measure is especially sensitive at low rank values and may therefore be less reliable for predicting accuracy in that regime.

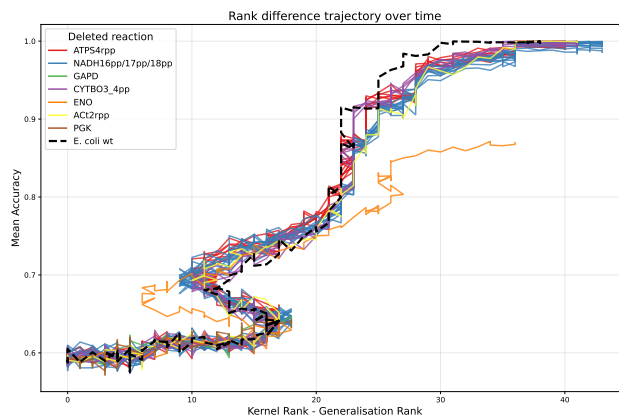


Figure 9: Relationship between rank difference and performance in the *E. coli* mutants. The data for the wild type strain is represented by the black dashed line.

More generally, increases in rank difference appear to correlate with higher accuracy. However, while many models follow a similar trajectory, deviations in some cases complicate the use of this measure for model comparison. Its sensitivity to the specific samples used for rank computation also limits its usefulness at low rank values.

Discussion and future work

As some recent work has shown Ahavi et al. (2026a), bacteria have the potential to be used as physical reservoirs. The

main aim of this study was to explore the computational capability of different species and mutated strains and to characterise properties commonly measured in reservoir computing systems. Additionally, we wanted to test whether these separability and similarity properties were good predictors of a model's performance.

First, we found that many of these microbial models are capable of supporting nonlinear computations and achieve strong performance on classification tasks with highly nonlinear structure. The results showed that some species performed better in terms of how quickly they reached higher accuracies, whereas others achieved higher maximum accuracies (see Figure 5).

Secondly, the existence of these different behaviours allows selecting a model depending on whether faster convergence or higher maximum accuracy is considered more valuable. This indicates a trade-off between convergence speed and peak performance, allowing model selection based on which objective is considered more important. In multi-objective optimization terms, the Pareto front corresponds to the set of non-dominated models, that is, those for which improving convergence speed would come at the expense of maximum accuracy, and vice versa. From the fastest to the max accuracy, the three strains on the Pareto front are : *S. aureus*, *Salmonella* and *E. coli*. In contrast, as shown in Figure 6, all *E. coli* mutants appear to be dominated by the wild-type model. This is consistent with the idea that the ability to generate a wide range of responses to different inputs is an important property of living systems. It is therefore not surprising that mutants carrying gene deletions perform worse than the wild type. In other words, a greater information-processing capacity may be advantageous in changing environments (Seoane, 2019). This may explain why the wild species display a clearer Pareto front, as is commonly observed in evolution when different traits are optimised at the expense of one another.

Lastly, while an increase in the difference between the kernel and generalisation ranks was found to be closely related to an increase in accuracy, deviations in some models and sensitivity to specific samples in low rank values diminish the potential of this metric as a predictor of accuracy in practice.

This opens up future work directions in areas like network analysis where we could investigate if any measurable properties of the metabolic networks like degree distribution or centrality are good predictors of the strains' computational capabilities. Additionally, other measures typical of reservoir systems like Information Processing Capacity (IPC) were not explored in the present study because they can only be applied to setups that handle sequential inputs. Future work will also involve developing a bacterial reservoir computing implementation that can process sequential data.

Appendix

Below we include a table with the models used in the study and reproductions of the figures with standard deviations for the cases where they were not included for visual clarity in the main text.

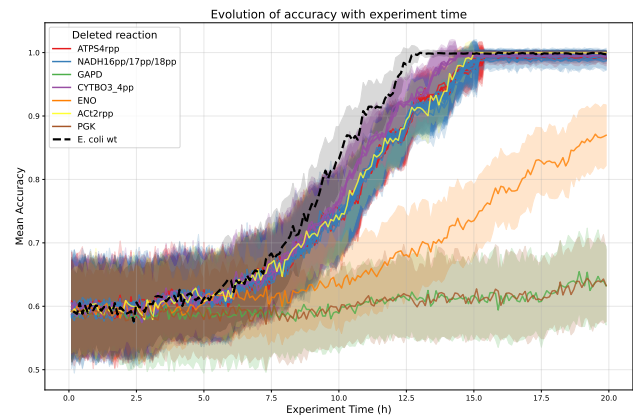


Figure 10: Evolution of accuracy with experiment time in the *E. coli* mutants. The data for the wild type strain is represented by the black dashed line. The shaded areas represent standard deviation.

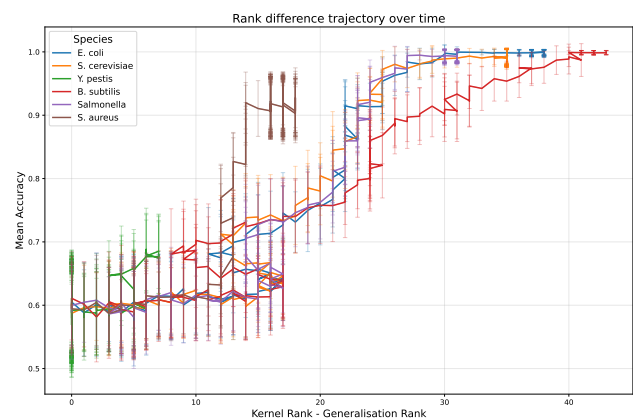


Figure 11: Relationship between rank difference and performance in the different species. Error bars represent standard deviation.

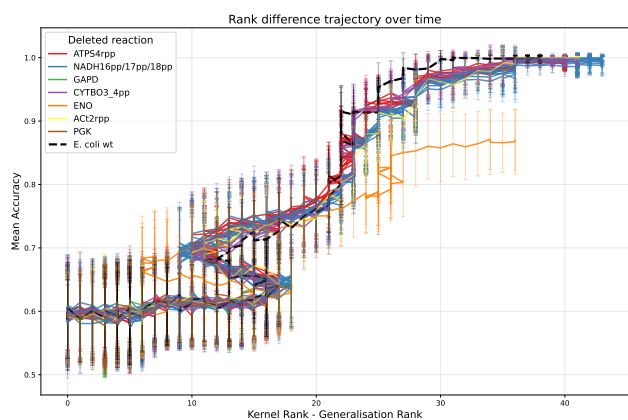


Figure 12: Relationship between rank difference and performance in the *E. coli* mutants. The data for the wild type strain is represented by the black dashed line. Error bars represent standard deviation.

BiGG ID	Short Name	Organism
iML1515	<i>E. coli</i>	<i>E. coli</i> str. K-12 substr. MG1655
iMM904	<i>S. cerevisiae</i>	<i>Saccharomyces cerevisiae</i> S288C
iPC815	<i>Y. pestis</i>	<i>Yersinia pestis</i> CO92
iYO844	<i>B. subtilis</i>	<i>Bacillus subtilis</i> subsp. <i>subtilis</i> str. 168
iYS1720	Salmonella	Salmonella pan-reactome
iYS854	<i>S. aureus</i>	<i>Staphylococcus aureus</i> subsp. <i>aureus</i> USA300.TCH1516

Table 2: Genome-scale metabolic models (GEMs) used in the study.

Acknowledgements

We would like to thank Thierry Vieville for early discussions on VC-dimension and related measures. We also acknowledge the following fundings: INRAE-Inria PhD grant, PEPR B-BEST France 2030 program (grant number ANR-22-PEBB-0008), the EU HORIZON BIOS program (grant number 101070281) and the DeepPool project (ANR-21-CE23-0009).

References

- Ahavi, P., Hoang, T.-N.-A., Meyer, P., Berthier, S., Fiorini, F., Castelli, F., Epaulard, O., Le Gouellec, A., and Faulon, J.-L. (2026a). Living bacterial reservoir computers for information processing and sensing. working paper or preprint.
- Ahavi, P., Le Gouellec, A., and Faulon, J.-L. (2026b). Microbial computing: Review and perspectives. *Biotechnology Advances*, 87:108766.
- Ebrahim, A., Lerman, J. A., Palsson, B. O., and Hyduke, D. R. (2013). Cobrapy: Constraints-based reconstruction and analysis for python. *BMC Systems Biology*, 7(1).
- Elowitz, M. B. and Leibler, S. (2000). A synthetic oscillatory network of transcriptional regulators. *Nature*, 403(6767):335–338.
- Gardner, T. S., Cantor, C. R., and Collins, J. J. (2000). Construction of a genetic toggle switch in *Escherichia coli*. *Nature*, 403(6767):339–342.
- Henson, M. A. and Hanly, T. J. (2014). Dynamic flux balance analysis for synthetic microbial communities. *IET Systems Biology*, 8(5):214–229.
- IEC, O., IFCC, R., ILAC, L., ISO, L., IUPAC, M., IUPAP, R., et al. (1995). Guide to the expression of uncertainty in measurement. *ISO, Geneva*.
- Kaplan, N. A., Islam, K. N., Kanis, F. C., Verderber, J. R., Wang, X., Jones, J. A., and Koffas, M. A. (2024). Simultaneous glucose and xylose utilization by an *Escherichia coli* catabolite repression mutant. *Applied and Environmental Microbiology*, 90(2).
- Lauer, F. (2014). VC-dimension of hyperplanes. <https://mlweb.loria.fr/book/en/VCdimhyperplane.html> (Accessed: 2026-04-10).
- Legenstein, R. and Maass, W. (2007). Edge of chaos and prediction of computational performance for neural circuit models. *Neural Networks*, 20(3):323–334.
- Lukoševičius, M. and Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. *Computer science review*, 3(3):127–149.
- Mahadevan, R., Edwards, J. S., and Doyle, F. J. (2002). Dynamic flux balance analysis of diauxic growth in *Escherichia coli*. *Biophysical Journal*, 83(3):1331–1340.
- Nakajima, K. (2020). Physical reservoir computing—an introductory perspective. *Japanese Journal of Applied Physics*, 59(6):060501.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Seoane, L. F. (2019). Evolutionary aspects of reservoir computing. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 374(1774):20180377.

Tanaka, G., Yamane, T., Héroux, J. B., Nakane, R., Kanazawa, N., Takeda, S., Numata, H., Nakano, D., and Hirose, A. (2019). Recent advances in physical reservoir computing: A review. *Neural Networks*, 115:100–123.