nature microbiology

Article

https://doi.org/10.1038/s41564-023-01376-0

BacterAI maps microbial metabolism without prior knowledge

Received: 20 October 2021

Accepted: 29 March 2023

Published online: 04 May 2023

Check for updates

Adam C. Dama[1], Kevin S. Kim[1], Danielle M. Leyva[1], Annamarie P. Lunkes[1], Noah S. Schmid[1,2], Kenan Jijakli[1] & Paul A. Jensen[1,2,3,4,5]

Training artificial intelligence (AI) systems to perform autonomous experiments would vastly increase the throughput of microbiology; however, few microbes have large enough datasets for training such a system. In the present study, we introduce BacterAI, an automated science platform that maps microbial metabolism but requires no prior knowledge. BacterAI learns by converting scientific questions into simple games that it plays with laboratory robots. The agent then distils its findings into logical rules that can be interpreted by human scientists. We use BacterAI to learn the amino acid requirements for two oral streptococci: *Streptococcus gordonii* and *Streptococcus sanguinis*. We then show how transfer learning can accelerate BacterAI when investigating new environments or larger media with up to 39 ingredients. Scientific gameplay and BacterAI enable the unbiased, autonomous study of organisms for which no training data exist.

The microbiome revolution has identified thousands of species of bacteria that deserve scientific investigation[1,2]. Scientists cannot keep pace with the expanding tree of life and most species of bacteria remain unstudied. AI and automation could accelerate scientific discovery by replacing humans with algorithms that mine the scientific literature and design new experiments[3,4]. The least-studied bacteria would benefit the most from automated research, but, ironically, the lack of data makes it difficult to deploy autonomous agents to study these species.

Recently, AI has surpassed human performance in games long thought to be too complex for machines[5–7]. These stunning advances are powered by deep reinforcement learning (RL), a branch of AI in which agents solve games by trial and error. RL agents can learn tabula rasa—from a blank slate—without prior knowledge of strategies or even the rules of the game[5,7]. Simply rewarding agents for winning can lead to optimal strategies and beat human world champions[8].

Converting biological research questions into games allows the study of microbes using RL techniques. We developed an RL agent that solves combinatorially large research questions by 'playing' science with automated experiments. We asked our RL agent—called BacterAI—to learn what combinations of amino acids support growth of the oral bacterium *Streptococcus gordonii*. This seemingly simple question belies a complex answer. There are $2^{20} = 1{,}048{,}576$ possible subsets of the 20 proteinogenic amino acids and BacterAI cannot rely on a brute force search of every combination. Instead, it must select the most informative experiments and train a computational model to predict the results for untested combinations.

Learning from a blank slate avoids biasing results with prior knowledge. Using BacterAI, we learned that another oral microbe, *Streptococcus sanguinis*, has different amino acid auxotrophies from *S. gordonii* even though the two species are closely related and live in the same environment. Although BacterAI can operate with no prior knowledge, we show that reusing models from previous runs can accelerate BacterAI. This form of transfer learning allows us to quickly scale BacterAI to larger problems and learn how metabolic logic differs between species and environments.

Results

BacterAI uses laboratory automation to play scientific games

RL agents learn by trial and error, so BacterAI needs the freedom to design and execute its own experiments. When learning computer games, two RL agents can play against each other to gather experience[9]. For automated biology, BacterAI needs to plan and execute physical

[1]Department of Bioengineering, University of Illinois at Urbana-Champaign, Urbana, IL, USA. [2]Department of Biomedical Engineering, University of Michigan, Ann Arbor, MI, USA. [3]Department of Microbiology, University of Illinois at Urbana-Champaign, Urbana, IL, USA. [4]Carl R. Woese Institute for Genomic Biology, University of Illinois at Urbana-Champaign, Urbana, IL, USA. [5]Department of Chemical Engineering, University of Michigan, Ann Arbor, MI, USA. e-mail: pjens@umich.edu

https://doi.org/10.1038/s41564-023-01376-0

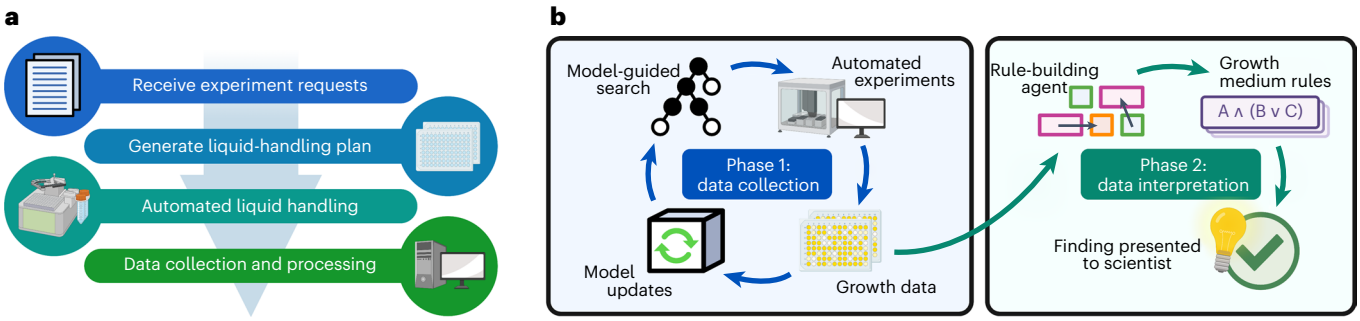


Fig. 1 | BacterAI uses an automated experiment pipeline and an AI agent to study bacterial metabolism. a, A pipeline for converting experiments into liquid-handling instructions for laboratory robots and human technicians who supervise the automated liquid handling. Growth data are collected, processed and returned to the researcher who requested them. **b**, BacterAI using an internal neural network and rollout simulations during Phase 1 (data collection) to find experiments at the predicted growth front. These experiments are sent to the automated experiment pipeline in **a** for execution. BacterAI retrains its neural network on the new experimental data. The next set of experiments is planned using the updated neural network and the cycle repeats. When sufficient data have been collected, a second agent can begin phase 2 (data interpretation). The experimental data are used to build a logical rule that can be interpreted by human scientists.

experiments. We built a laboratory automation pipeline to perform growth assays requested by BacterAI (Fig. 1a). BacterAI selects 336 experiments each day, but replicates and controls expand this number to 1,152 assays with up to 50 liquid-handling operations per assay. A customized scheduler accepts BacterAI's experimental designs and produces instructions for 11 laboratory instruments and a human technician who prepares reagents and loads the machines. All data processing and quality control are automated without human intervention.

We split our biological game into two phases: data collection (phase 1) and interpretation (phase 2) (Fig. 1b). In phase 1, BacterAI tries to find media that span the 'growth front'—the boundary that separates a growth medium and a no-growth medium that share all but one ingredient. BacterAI trains an internal neural network to predict the fitness of the bacterium in all 2^{20} media. (Fitness is measured relative to growth with all 20 amino acids.) Guided by this model, BacterAI searches for untested grow/no-grow pairs and requests batches of 336 experiments per day. The search for media uses a rollout algorithm[10] and follows two strategies. The first strategy, exploiting, looks for media with the fewest ingredients that lie directly on either side of the growth front. The second search strategy, exploring, begins with a medium containing all 20 amino acids and randomly removes ingredients until the agent reaches the growth front. Both exploiting and exploring end their search at the model's predicted growth front, but exploration experiments venture into uncertain regions of the model. Strategies for exploiting and exploring are always at odds with each other and exploitation versus exploration is a fundamental tradeoff in RL[11,12].

Each morning, a technician uploads the final growth measurements from the previous day's experiments to BacterAI. After retraining its neural network, BacterAI searches the model for untested media along the growth front. A new experimental design is returned within an hour so that the robots can assemble the experiments in the afternoon for overnight incubation. The learning cycle continues until the neural network can consistently predict the outcomes of new experiments before they are performed.

BacterAI learns the amino acid auxotrophies of an oral microbe

We tested BacterAI with the bacterium *S. gordonii*, a commensal species found in the oral microbiome of most humans. BacterAI began its gameplay with no prior information about *S. gordonii*. It did not know that the 20 inputs were amino acids or even that the measured output was growth. Everything about the relationship between inputs and outputs was learned through trial and error. On day 1, the agent's neural network was untrained and the agent requested 336 randomly selected experiments. The results of these experiments showed no correlation with the agent's predictions (Fig. 2). After day 1, the experiments requested by BacterAI formed a bimodal distribution (Fig. 2a). The experiments planned when the agent exploited had fewer amino acids, while the exploring experiments had more amino acids. The number of amino acids shifted as the agent learned the location of the growth front. When BacterAI overpredicted growth, retraining the model on the new data moved the predicted growth front towards experiments with more amino acids. Under-predicting growth encouraged the agent to remove more ingredients during the next round. These changes in strategy were not preprogrammed into BacterAI, but were the result of updating the model used during the rollout search.

BacterAI is not programmed to distribute its experiments in any particular way. The agent's search for informative experiments did not reflect the distribution of possible experiments for a given number of amino acids. For example, 12.7% of the experiments requested by BacterAI contain 17 amino acids, but media of this size represent only 0.11% of the 2^{20} possible combinations. The final distribution of experiments selected by BacterAI depends on the auxotrophies of the bacterium under investigation and would probably change when studying different strains or species.

BacterAI needed only 9 days and 3,024 experiments before the growth/no-growth prediction accuracy of its neural network exceeded 90% (Fig. 2b; data for all 13 rounds appear in Extended Data Fig. 1). Having acquired sufficient experimental data, the agent switched to phase 2: communicating its findings to human scientists. BacterAI's internal neural network is a black-box model used only to plan experiments and is not meant to be directly interpreted by humans. Instead, a second 'interpretation' agent builds logical rules that define the combinations of amino acids that support growth (Fig. 1b, green). Building logical rules is a combinatorial optimization problem and our interpretation agent uses a genetic algorithm to find rules that match the experimental data. The agent enforces parsimony by regularization—penalizing overly complex rules that add logical clauses with only a small increase in accuracy.

BacterAI noticed that its logical rule had stabilized when the new data acquired on day 13 did not change the rule produced on day 12. The agent terminated the game and reported its final rule for *S. gordonii*:

Arg and Leu and Phe and Ser and Tyr and Val and (Gln or Glu).

The rule's cross-validation accuracy was 88.6% on the experimental data used for training, with a true positive/growth rate (TPR) of 0.906

https://doi.org/10.1038/s41564-023-01376-0

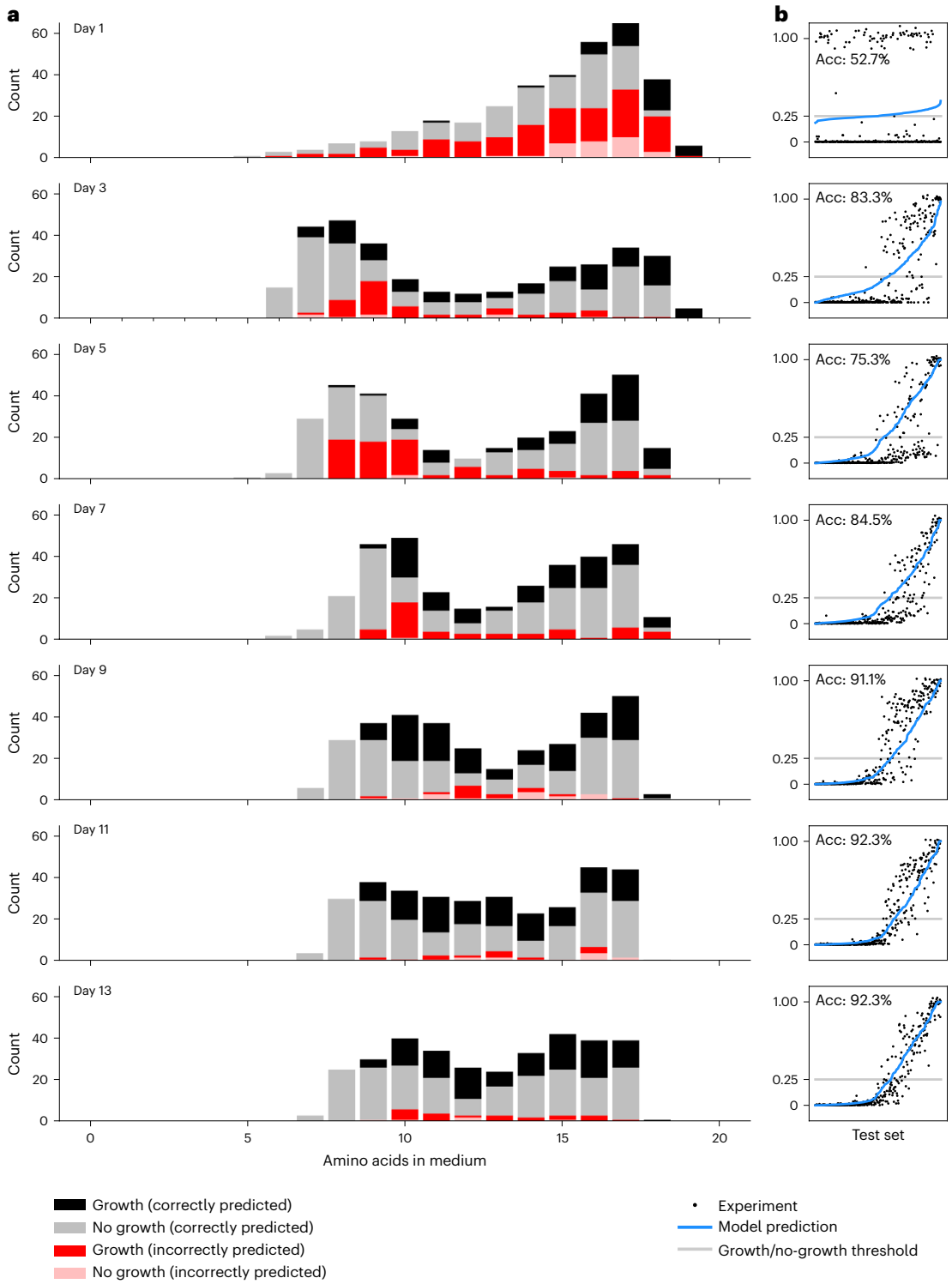


Fig. 2 | BacterAI selects experiments to train an internal neural network that predicts the fitness of *S. gordonii*. **a**, Experiments selected at the predicted growth front where the removal of a single ingredient switches from growth to no growth. On day 1, the selections are random, causing many incorrect grow/no-grow predictions. The agent adjusts the depth of the experiments (the number of amino acids removed) as it learns the location of the growth front. Data from the experiments are used to train a neural network that predicts the growth of all combinations of amino acids. The day 1 network is untrained and the model is retrained each day on the entire set of collected data. The predictions in the figure are made before the neural network is trained on the data from the corresponding day. **b**, The accuracy (Acc) for the model is the number of correct growth/no-growth predictions using a growth threshold of 0.25. Data for all 13 days and the plots of the training accuracy for the neural network are available in Extended Data Fig. 1.

and a true negative/no growth rate (TNR) of 0.869. As expected, the accuracy of the logical rule was lower than the accuracy of the neural network used by BacterAI ($p < 5 \times 10^{-8}$, normal proportion test, one-sided). The logical rule should be reserved for interpretation, and the more accurate neural network can be used to predict growth in untested environments.

https://doi.org/10.1038/s41564-023-01376-0

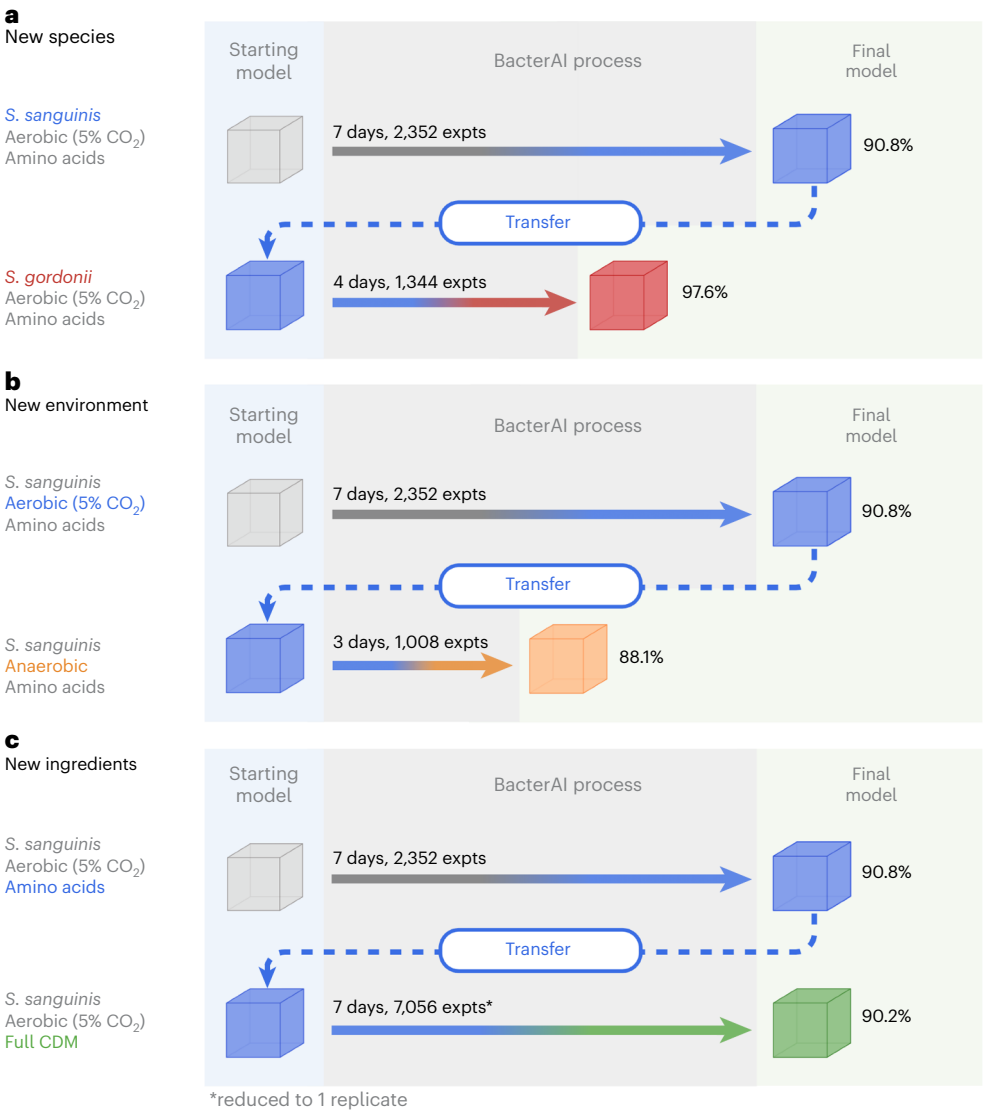


Fig. 3 | Transfer learning accelerates BacterAI. a, Agents reusing the internal neural network from *S. sanguinis* learning amino acid growth requirements for *S. gordonii* in only 4 days. expts, experiments. **b**, Agents using models from aerobic growth experiments to learn anaerobic growth requirements in 3 days. **c**, Reusing amino acid growth data to allow BacterAI to learn the growth requirements of *S. sanguinis* in media with 39 ingredients, a problem that is 2^{19} = 524,288× larger.

Learning a growth rule required a balanced training set with sufficient grow and no grow results. BacterAI achieved balance by selecting experiments along the growth front, which for *S. gordonii* is a difficult task. The agent's final neural network predicted that only 1.24% of all amino acid combinations support growth of *S. gordonii*; however, *S. gordonii* grew in 33% of the media selected by BacterAI. The agent's deliberate sampling found far more growth conditions than would be expected in a random sample of media. Randomly selected media are likely to be far from the growth front with fitnesses that cluster around zero or one, while fitness values of media selected by BacterAI are more evenly distributed (Extended Data Fig. 2). Without BacterAI, it would be difficult to randomly select training data that include enough intermediate conditions to train a predictive model. Balanced data are also required for validating the growth rules from BacterAI. We randomly selected 1,120 experiments that were not previously requested by BacterAI. The *S. gordonii* growth rule was correct for 98.5% of the experiments (TPR = 0.706, TNR = 0.989), but the large number of true negatives (1,091) and the high TNR inflate the rule's accuracy. To acquire a more balanced testing set, we used BacterAI's neural network to select 1,000 experiments with 500 experiments that were predicted to grow and 500 that were predicted to not grow. The rule's accuracy on these data was 83.3% (TPR = 0.678, TNR = 0.988), which was closer to the cross-validation accuracy on the training data (88.6%).

BacterAI finds differences in metabolic logic between similar microbes

We asked BacterAI to play another game and learn the amino acid requirements of a different oral bacterium, *S. sanguinis*. Starting from a blank slate, BacterAI found the growth rule:

Gly and (Arg or Cys) and (Cys or Leu)

and (Gln or Glu) and (Leu or Val)

with 89.1% test accuracy after 11 rounds of autonomous experiments (Extended Data Fig. 3). Again, the logical rule was more interpretable but less accurate than the neural network used to select experiments (92% > 89.1%, $p < 4 \times 10^{-9}$, normal proportion test, one-sided). BacterAI discovered that *S. gordonii* and *S. sanguinis* have different amino acid auxotrophies despite being closely related[13] and living exclusively in the human oral cavity. Human scientists may expect *S. gordonii* and

https://doi.org/10.1038/s41564-023-01376-0

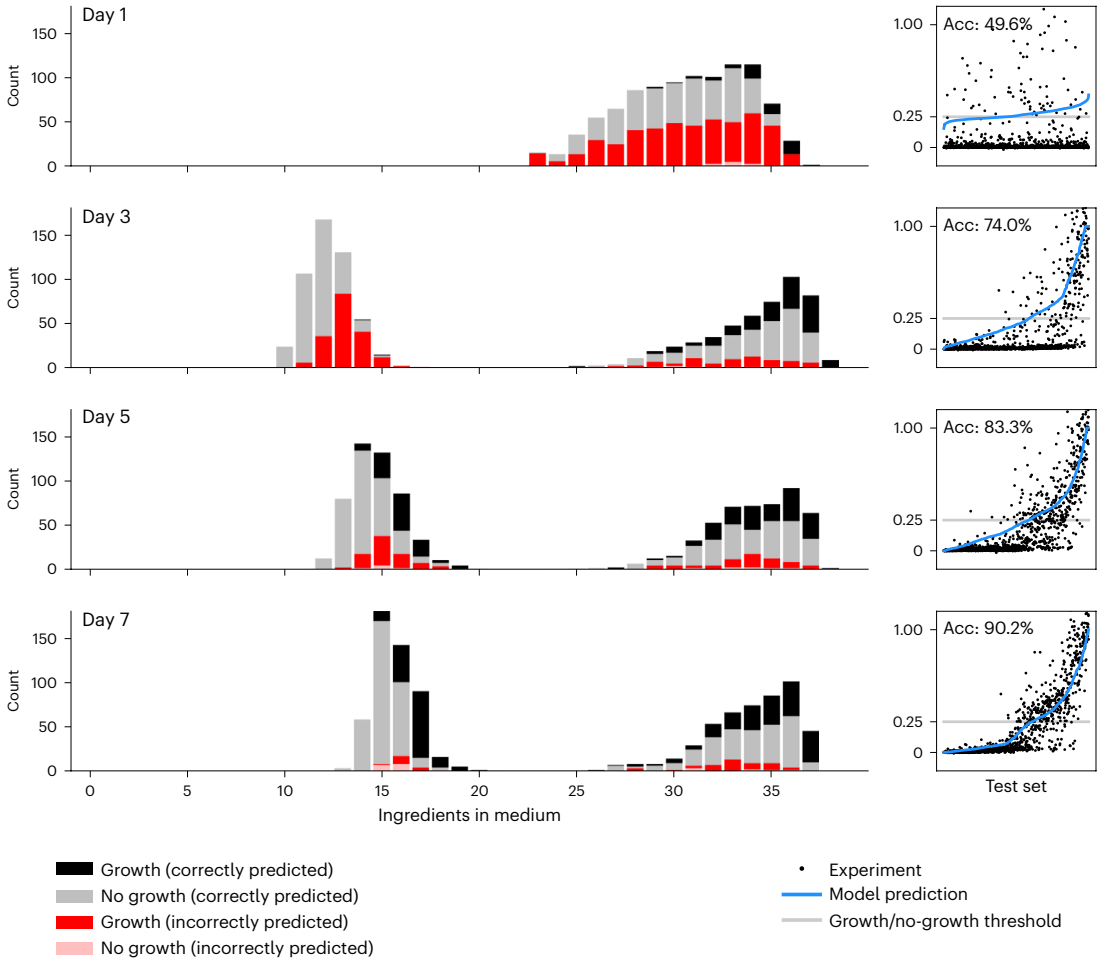


Fig. 4 | BacterAI quickly learns to predict growth for *S. sanguinis* in CDM with 39 ingredients. Training was started by transferring the amino acid data and used 1,008 unreplicated experiments each day for 7 days.

S. sanguinis to have similar amino acid requirements, but BacterAI presumes nothing about the two species. Each game played by BacterAI is independent and the similarities between *S. gordonii* and *S. sanguinis* are irrelevant to the agent.

Transfer learning accelerates BacterAI

BacterAI originally needed 12 days to learn the amino acid requirements of *S. gordonii* without prior knowledge. Rather than start from a blank slate, we modified BacterAI to begin the *S. gordonii* experiments with the model trained on *S. sanguinis* data. We continued training the *S. sanguinis* model with new data until the model accurately predicted the growth requirements for *S. gordonii* (Fig. 3a). Using this form of transfer learning, BacterAI learned *S. gordonii*'s auxotrophies in only 4 days (Extended Data Fig. 4). Transfer learning avoids re-learning common auxotrophies (for example, the dependence on either glutamate or glutamine), focusing future experiments on the differences between the species.

We also used transfer learning to discover how *S. sanguinis*'s auxotrophies change when the bacterium switches from aerobic to anaerobic conditions (Fig. 3b). As expected, the aerobic model was not a good predictor of anaerobic growth (Extended Data Fig. 5). Transfer learning required only three rounds of anaerobic experiments to improve the model from 61% to >88% accuracy (Fig. 3b). The final anaerobic growth rule for *S. sanguinis* is:

Gly and Cys and (Gln or Glu) and Leu and (Ile or Val).

In addition to amino acids, oral streptococci require carbohydrates, vitamins, minerals and cofactors for growth. Our final challenge for BacterAI was to learn the growth rule for *S. sanguinis* in a complete, 39 ingredient, chemically defined medium (CDM)[14]. The resulting $2^{39} \approx 5.5 \times 10^{11}$ possible media define a search space that is a 524,288× larger than the 2^{20} combinations of only amino acids. We made two changes to BacterAI to help it manage the added complexity. First, we once again used transfer learning by training an initial model with the amino acid data and random experiments for the 19 other ingredients. Second, we switched from three replicates of 336 experiments to 1,008 unreplicated experiments per round. We noticed little variation between the biological replicates (Extended Data Fig. 6) and hypothesized that tripling the number of conditions tested per round would more than offset the noise introduced by removing the replicates.

Using transfer learning and unreplicated experiments, BacterAI needed only 7 rounds (7,056 total experiments) to achieve >90% accuracy at predicting growth in media with up to 39 ingredients (Figs. 3c and 4). Up to 27 ingredients can be removed while preserving growth, and the overall growth rule is:

glucose and adenine and thiamine and $MgSO_4$

and riboflavin and pantothenate

and (NADP or niacinamide) and (sodium acetate or vitamin B_{12})

and Gly and (Arg or Cys) and (Cys or Leu)

and (Gln or Glu) and (Leu or Val).

Discussion

BacterAI's combination of gamification and blank slate learning offers a generalizable approach to automated biology. We used BacterAI to learn growth rules for amino acids and other nutrients, but the same algorithms could learn any input/output relationship. The agent's search for experiments is driven only by a numerical reward signal and the agent is free to select any combination of inputs that it believes will maximize the reward. By basing its decisions only on data it collects, BacterAI avoids human biases and preconceptions when planning experiments. However, this freedom demands an automated platform that can handle experiments with combinatorial complexity and tens of thousands of liquid handling operations each day. BacterAI also relies on bacteria that can be grown in vitro in defined media. Transcriptome profiles of oral bacteria grown in vitro resemble in vivo data[15], but growth assays in the lab cannot reproduce every feature of the oral microenvironment. The BacterAI approach could also be applied to communities of bacteria provided the growth of individual strains can be measured.

BacterAI is a form of active learning where prior results inform future data collection. BacterAI is in some ways similar to Bayesian optimization with a neural network surrogate; however, BacterAI does not search for conditions that optimize a single objective, e.g. a minimal medium or the conditions that maximize growth rate. Instead, BacterAI focuses on model improvement to maximize knowledge acquisition over the entire problem domain. BacterAI separates data collection from knowledge distillation, but in principle rules could be learned directly from the data as they are collected. BacterAI does not depend on any particular type of mathematical model. The neural networks and genetic algorithms inside BacterAI could be replaced with Gaussian processes, logical induction systems, or other machine learning techniques.

Blank slate learning avoids the need for any prior knowledge of the organism. Previous automated biology systems developed and tested hypotheses gleaned from the scientific literature[3,16]. These projects necessarily studied model organisms with extensive prior knowledge for training the agent's models. BacterAI's ability to learn solely from its own data enables the study of the unknown corners of microbiology.

On the other hand, our experiments with transfer learning demonstrate the advantage of starting from relevant prior knowledge. Retraining an *S. sanguinis* model to predict *S. gordonii* growth is more efficient than learning an *S. gordonii* model from scratch even though the two species have different auxotrophies. The differences we observed between *S. sanguinis* and *S. gordonii* suggest that even closely related species in the same environment can have different metabolic capabilities. Applying BacterAI to more species will expand our functional knowledge of oral microbiome beyond what can be extracted from metagenomic databases alone.

Constraint-based metabolic models are another potential source of prior knowledge[17,18]. These models predict growth based on a bacterium's annotated genome and could help select informative experiments before BacterAI has sufficient data to train its own models. Conversely, constraint-based models would benefit from the experimental data collected by BacterAI to fill knowledge gaps about metabolic regulation that is lacking from pathway databases. All of these tools—models, AI agents, and robotics—will need to come together to uncover the logic of all the understudied bacteria in our microbiome.

Methods

Strains, reagents and conditions

S. gordonii CH1 and *S. sanguinis* SK36 were grown in Todd Hewlitt Broth with 0.2% yeast extract or in a CDM without cystine[14]. A CDM base containing all ingredients except the 20 amino acids was prepared in a 2x stock and sterilized by autoclave. Stocks of individual ingredients were prepared at their solubility limit and filter sterilized. All reagents were purchased from Millipore-Sigma.

Frozen bacteria were revived in 5 ml of THY and grown overnight in 5% CO_2 at 37 °C. Overnight cultures were diluted 1:40, restarted in CDM and grown until exponential phase (optical density (OD) ~0.20). Cells were pelleted, washed three times and resuspended in sterile water before inoculating 384 well plates with 2 µl of culture per well. Plates were incubated overnight at 37 °C in 5% CO_2 or in a Coy anaerobic chamber with a 5% H_2, 10% CO_2 and 85% N_2 atmosphere.

Laboratory automation

Experiments requested by BacterAI were converted into liquid-handling instructions through a combinatorial optimization algorithm and the PlatePlan library[19]. A customized web interface created instructions for the human technicians and downloadable protocol files for the liquid-handling robots. Technicians used the website to upload raw data from the plate readers.

Each day's experiments were randomized across 384-well plates with 24 blanks and 24 positive controls containing full CDM and all 20 amino acids. For amino acid experiments, a 2x base medium containing all CDM ingredients except the amino acids was added to the wells by a MultiFlow FX dispenser connected to a BioStack 2WR plate stacker. The plates were moved to a Formulatrix Tempest non-contact liquid dispenser with integrated plate stackers to dispense solutions of each ingredient. Sterile water was added to equalize the volumes of all wells and each well (except the blanks) was inoculated with 2 µl of exponential bacterial culture resuspended in sterile water.

OD was read before and after overnight incubation with a BioTek Epoch 2 plate reader connected to a BioTek BioStack 2WR plate stacker. The plates were not agitated during incubation as is common when growing streptococci. Each plate underwent pulse centrifugation for 6–8 s to remove bubbles before the initial read. Plates were vortexed for 10 s at 2,000 r.p.m. and then shaken by the Epoch 2 reader for 10 s at 1,096 c.p.m. (1 mm) immediately before the final reading to resuspend any settled cells.

The OD changes (final minus initial) for three biological replicates of each experiment were normalized to the mean of the plate controls and the median was selected. These data were returned to BacterAI. Quality control consisted of a visual inspection of the positive controls and blanks by the human technician to check for contamination or inoculation errors, in addition to a dilution assay performed daily with the Tempest to check for pipetting accuracy and consistency.

Selecting experiments with BacterAI

Experiment planning is described by a Markov decision process (MDP)[20]. Each state represents a combination of ingredients in the media, with a total of 2^{20} = 1,048,576 states for amino acid experiments and 2^{39} = 549,755,813,888 states for experiments with the complete CDM. All trajectories through the MDP begin from the complete medium. The agent transitions between states by removing an ingredient from the medium. A neural network trained on the experimental data predicts the fitness—the growth at each state relative to the growth of the bacterium in the complete medium. An agent is not allowed to remove any ingredient that would cause the predicted fitness to drop below 25%—the fitness threshold chosen to indicate growth/no growth. The MDP terminates when the agent cannot remove any ingredients from its current medium. The agent receives a reward of +1 every time it removes an ingredient, so the total reward for a trajectory through the MDP is equal to the number of ingredients removed.

The agent selects ingredients to remove using a rollout algorithm[10]. For each ingredient in the current medium, the agent simulates ahead by removing ingredients until the neural network predicts no growth. The agent repeats this simulation 500× for each ingredient in the current medium, averaging the returns to estimate the total expected reward associated with removing each ingredient (R_i).

The ingredient selected for removal is drawn from a softmax distribution over all the expected rewards(R_i):

$$P(i) = \frac{e^{kR_i}}{\sum_j e^{kR_j}}$$

where the hyperparameter k determines the bias towards removing the ingredient with the largest expected reward. When $k = 0$, the probability of selecting an ingredient is uniform and independent of the expected reward. As $k \to \infty$, the agent deterministically selects the ingredient with the highest expected reward.

The hyperparameter k controls the exploring versus exploiting behaviour of the agents. When $k = 0$, the agent explores and takes a random walk through the MDP until it reaches the growth–no-growth boundary. Increasing k shifts the agent to exploit the information in the neural network and search for experiments with the fewest ingredients. To select the experiments in each batch, the agent begins with high exploitation behaviour ($k = 100$), adding experiments on each side of the growth–no-growth boundary (the first medium encountered that is predicted to not grow and its predecessor, the last medium predicted to grow). When exploiting, the agent frequently arrives at the same experiments—including experiments that were tested in previous batches—which are not repeated. The hyperparameter k decreases depending on the repeat score r and the fraction of ingredients ϕ remaining at the current state:

$$k = a(1 + \phi)\exp\{br^3\}.$$

The repeat score parameters were set at $a = 50$ and $b = 1/27{,}000$ based on simulations against data extrapolated from previous growth experiments for a different bacterium, *Streptococcus mutans*. The repeat score begins at 0 and is increased or decreased by 1 (with a floor of 0) whenever the agent arrives at a repeated experiment or an unseen experiment, respectively. Decreasing the repeat score decreases exploration in subsequent trajectories. To ensure that every round includes adequate exploration, half of each batch of experiments is selected when $k = 0$.

Neural network training

A set of 25 bootstrap aggregated (bagged) artificial neural networks[21] was trained to predict the fitness of the bacterium given a medium. Each individual model is identical in structure, consisting of one input layer taking in a binary representation of the growth medium, two hidden layers and one output layer. ReLU activation functions[22] were used after all but the final layer. The networks were trained independently on all previous experimental data using a mean squared error loss function, Adam optimizer[23] with a learning rate of 0.001 and a batch size of 360 across 50 epochs. Hyperparameter tuning software Optuna v.3.0 was used to select these values[24]. Each of the 25 models was trained on a random sampling with replacement matching the total number of data points in the full dataset. During evaluation, the results of the bagged models were averaged to create a single fitness prediction.

Building logical rules

Logical rules were constructed using a genetic algorithm[25]. A rule in conjunctive normal form with q clauses and n atoms per clause was represented by a vector with qn entries. Each entry contained an integer 0, ..., 20 representing no amino acid (0) or 1 of the 20 amino acids. In each round, an elite population of 20% of the top performing candidates was carried over from the previous round. Pair selections for recombination were drawn using a softmax distribution of the fitnesses calculated from a balanced accuracy score. Each candidate pair underwent a random single point crossover and were then mutated. The number of mutations was distributed with $P(0) = 0.25$ and $P(j) = 0.0375$ for $j = 1, \ldots, 20$. The mutations were sampled with replacement and a probability of 0.75 for a mutation to no amino acid and a probability of 0.0125 for all 20 amino acids. The increased mutation rate towards no amino acid helped regularize the rules. To start, 1,000 candidates with 4 clauses of 4 atoms per clause were selected for 250 generations. After finding the best scoring rule, the number of clauses was increased by one and the optimization was repeated until the best rule contained an empty clause.

The final rules were polished with a customized branch-and-bound solver that searched for improvements to the best rule from the genetic algorithm. Branch-and-bound was terminated after 4 h of searching or if the optimality gap dropped to <5%. The accuracy of the final rule was assessed with a holdout set of 25% of the data collected by BacterAI.

Implementation

BacterAI is implemented in the Python programming language v.3.10 using the PyTorch library v.1.13 for neural networks[26]. The branch-and-bound solver was implemented in the Julia programming language v.1.8. Simulations were performed on a workstation with an AMD Ryzen Threadripper 1950X processor (16-core, 3.4 GHz), 64-GB RAM and a Nvidia Titan V graphics card.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

All data are available at http://github.com/jensenlab/BacterAI and the authors' website: http://jensenlab.net/tools.

Code availability

All code is available at http://github.com/jensenlab/BacterAI and the authors' website: http://jensenlab.net/tools.

References

1. Parks, D. H. et al. A standardized bacterial taxonomy based on genome phylogeny substantially revises the tree of life. *Nat. Biotechnol.* **36**, 996–1004 (2018).
2. Quast, C. et al. The SILVA ribosomal RNA gene database project: improved data processing and web-based tools. *Nucleic Acids Res.* **41**, D590–D596 (2013).
3. Coutant, A. et al. Closed-loop cycles of experiment design, execution, and learning accelerate systems biology model development in yeast. *Proc. Natl Acad. Sci. USA* **116**, 18142–18147 (2019).
4. King, R. D. et al. The automation of science. *Science* **324**, 85–89 (2009).
5. Schrittwieser, J. et al. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature* **588**, 604–609 (2020).
6. Silver, D. et al. Mastering the game of Go with deep neural networks and tree search. *Nature* **529**, 484–489 (2016).
7. Silver, D. et al. A general reinforcement learning algorithm that masters chess, dhogi, and Go through self-play. *Science* **362**, 1140–1144 (2018).
8. Silver, D., Singh, S., Precup, D. & Sutton, R. S. Reward is enough. *Artif. Intell.* **299**, 103535 (2021).
9. Tesauro, G. TD-Gammon, a self-teaching backgammon program, achieves master-level play. *Neural Comput.* **6**, 215–219 (1994).
10. Tesauro, G. & Galperin, G. On-line policy improvement using Monte-Carlo Search. In *Advances in Neural Information Processing Systems* (eds Mozer, M. C. et al) (MIT Press, 1996); https://proceedings.neurips.cc/paper_files/paper/1996/file/996009f2374006606f4c0b0fda878af1-Paper.pdf
11. Fel'dbaum, A. A. Theory of dual control. *Autom. Remote Control* **21**, 1240–1249 (1960).

12. Witten, I. H. The apparent conflict between estimation and control—a survey of the two-armed bandit problem. *J. Frankl. Inst.* **301**, 161–189 (1976).
13. Patel, S. & Gupta, R. S. Robust demarcation of fourteen different species groups within the genus *Streptococcus* based on genome-based phylogenies and molecular signatures. *Infect. Genet. Evol.* **66**, 130–151 (2018).
14. van de Rijn, I. & Kessler, R. E. Growth characteristics of group A streptococci in a new chemically defined medium. *Infect. Immun.* **27**, 444–448 (1980).
15. Lewin, G. R., Stocke, K. S., Lamont, R. J. & Whiteley, M. A quantitative framework reveals traditional laboratory growth is a highly accurate model of human oral infection. *Proc. Natl Acad. Sci USA*. https://doi.org/10.1073/pnas.2116637119 (2022).
16. King, R. D. et al. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature* **427**, 247–252 (2004).
17. Magnúsdóttir, S. et al. Generation of genome-scale metabolic reconstructions for 773 members of the human gut microbiota. *Nat Biotechnol*. https://doi.org/10.1038/nbt.3703 (2017).
18. Jijakli, K. & Jensen, P. A. Metabolic modeling of *Streptococcus mutans* reveals complex nutrient requirements of an oral pathogen. *mSystems*. https://doi.org/10.1128/mSystems.00529-19 (2019).
19. Dama, A. C. & Jensen, P. A. PlatePlan. GitHub https://github.com/jensenlab/PlatePlan (2020).
20. Bellman, R. A Markovian decision process. *J. Math. Mech.* **6**, 679–684 (1957).
21. Breiman, L. Bagging predictors. *Mach. Learn.* **24**, 123–140 (1996).
22. Glorot, X., Bordes, A. & Bengio, Y. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* Vol. 15 (eds Gordon, G. et al.) 315–323 (PMLR, 2011); https://proceedings.mlr.press/v15/glorot11a.html
23. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. In *3rd International Conference on Learning Representations* (eds Bengio, Y. & LeCun, Y.) Preprint at *arXiv* https://doi.org/10.48550/arXiv.1412.6980 (2015).
24. Akiba, T., Sano, S., Yanase, T., Ohta, T. & Koyama, M. Optuna: a next-generation hyperparameter optimization framework. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* https://doi.org/10.1145/3292500.3330701 (Association for Computing Machinery, 2019).
25. Holland, J. H. *Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence* (MIT Press, 1992).
26. Paszke, A. et al. PyTorch: an imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32* (eds Wallach, H. et al.) 8024–8035 (Curran Associates, Inc., 2019); http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Acknowledgements

This research was supported by the National Institutes of Health (grant nos. EB027396 and GM138210 to P.J.). The Titan V used for this research was donated by the NVIDIA Corporation. We thank K. Janes for his comments on the manuscript. Figures 1 and 3 were created with BioRender.com.

Author contributions

A.D. and P.J. conceived the study, implemented BacterAI, designed experiments, analysed data and wrote the manuscript. A.D., K.K., D.L., A.L., N.S. and K.J. optimized laboratory workflows, executed experiments and performed quality control on the data.

Competing interests

The authors declare no competing interests.

Additional information

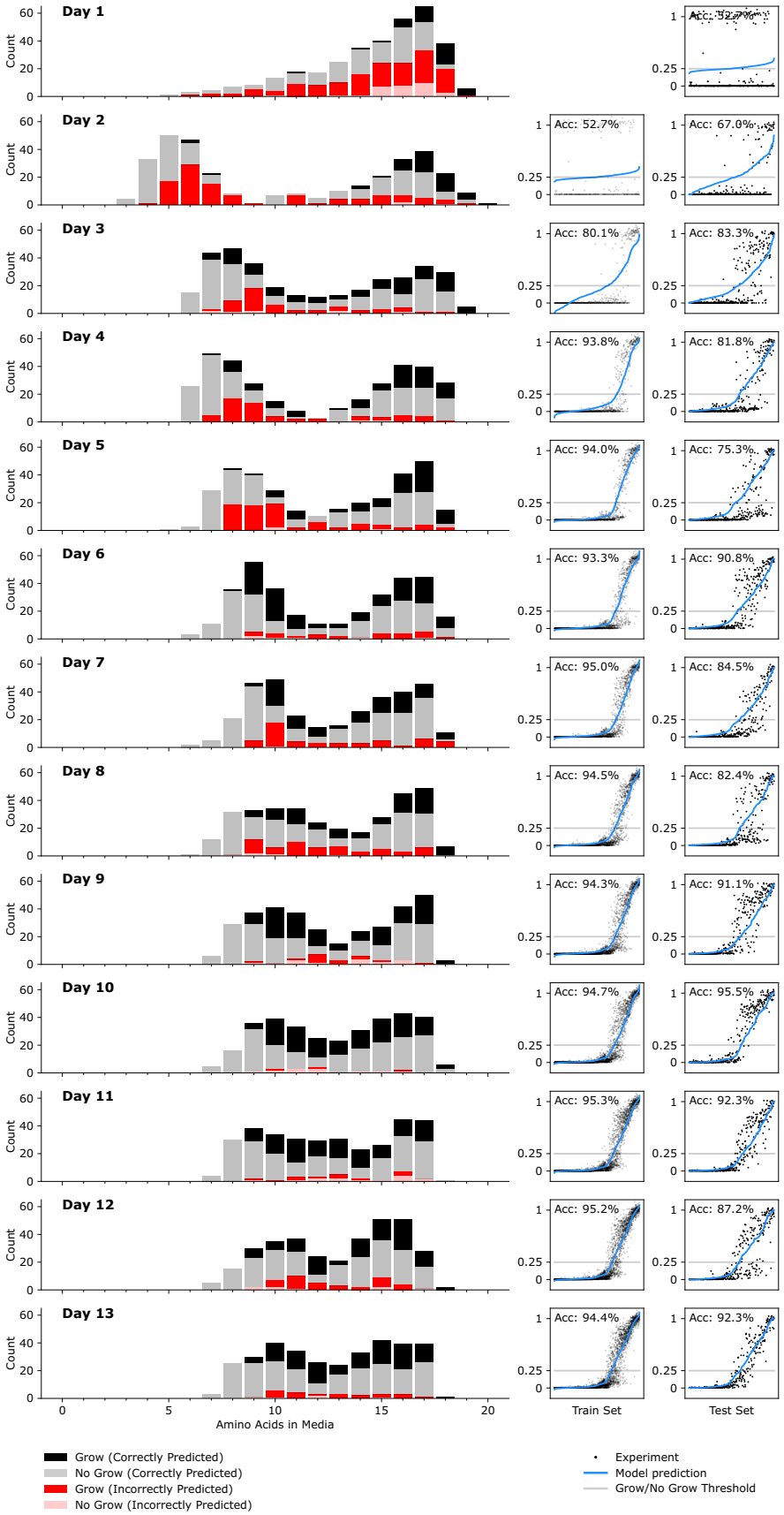
Extended data is available for this paper at https://doi.org/10.1038/s41564-023-01376-0.

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41564-023-01376-0.

Correspondence and requests for materials should be addressed to Paul A. Jensen.

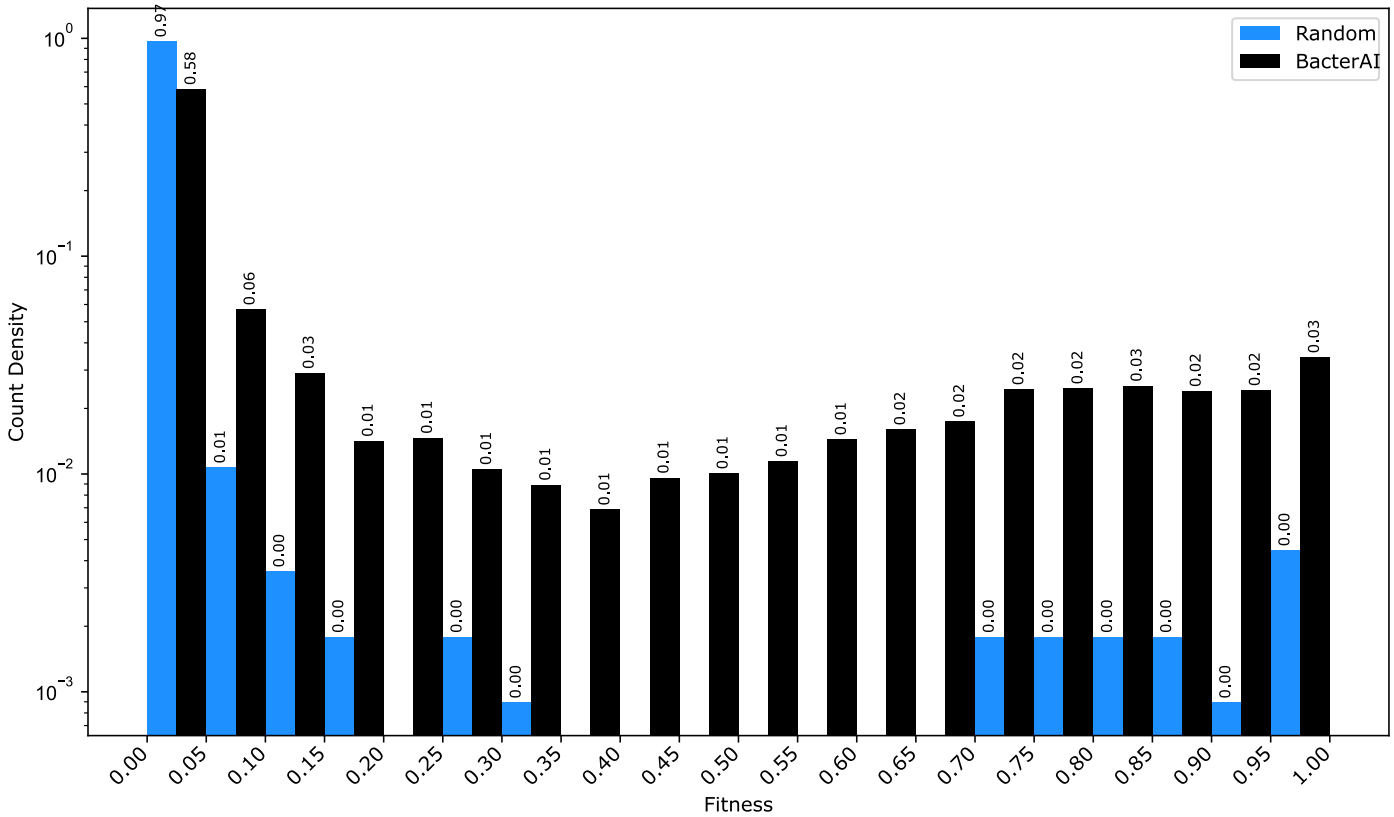
Peer review information *Nature Microbiology* thanks Nathan Price and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

https://doi.org/10.1038/s41564-023-01376-0



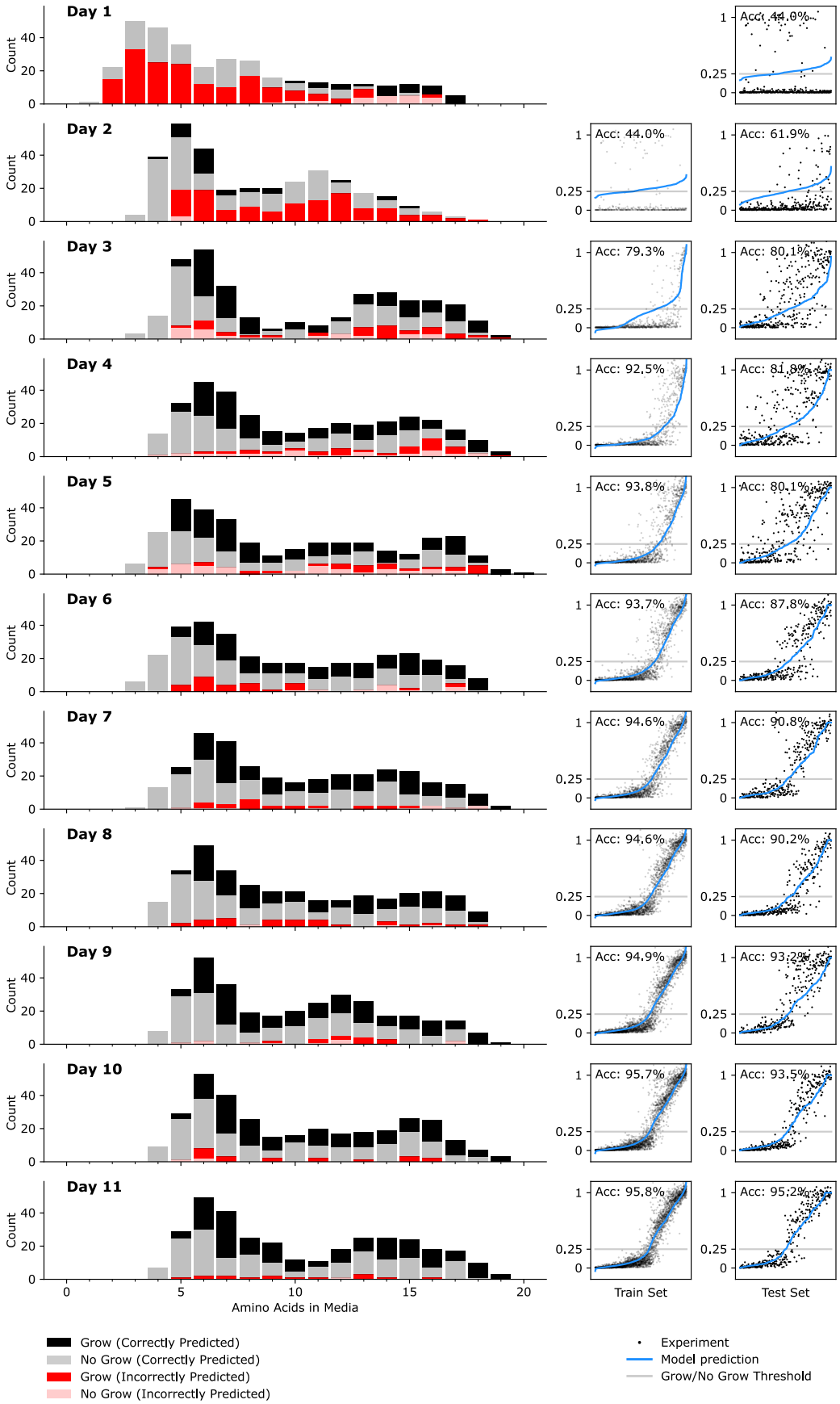
Extended Data Fig. 1 | BacterAI learns the amino acid requirements of *S. gordonii*. BacterAI learns to predict the growth of *S. gordonii* in media containing combinations of amino acids. Over 13 days, the agent trains a neural network to guide the search for new experiments along the growth front. Each day, the neural network is retrained using the data from all previous data (train set). The accuracy of the model is measured using the 336 experiments selected each day (test set).

https://doi.org/10.1038/s41564-023-01376-0



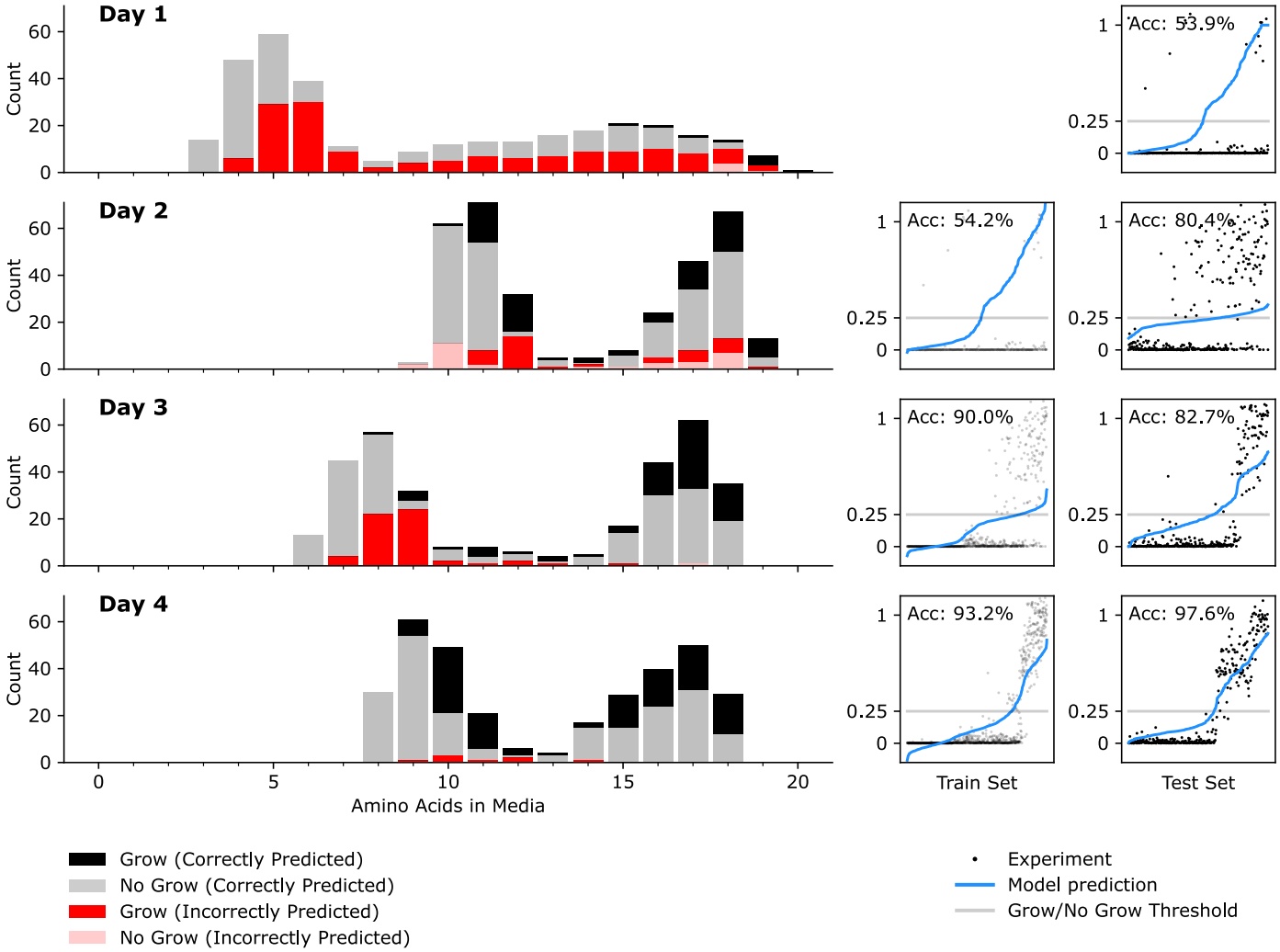
Extended Data Fig. 2 | Media selected by BacterAI are not random. The fitness of *S. gordonii* grown in randomly selected media clusters near no growth (0) and full growth (1). By contrast, experiments selected by BacterAI are more uniformly distributed.

https://doi.org/10.1038/s41564-023-01376-0



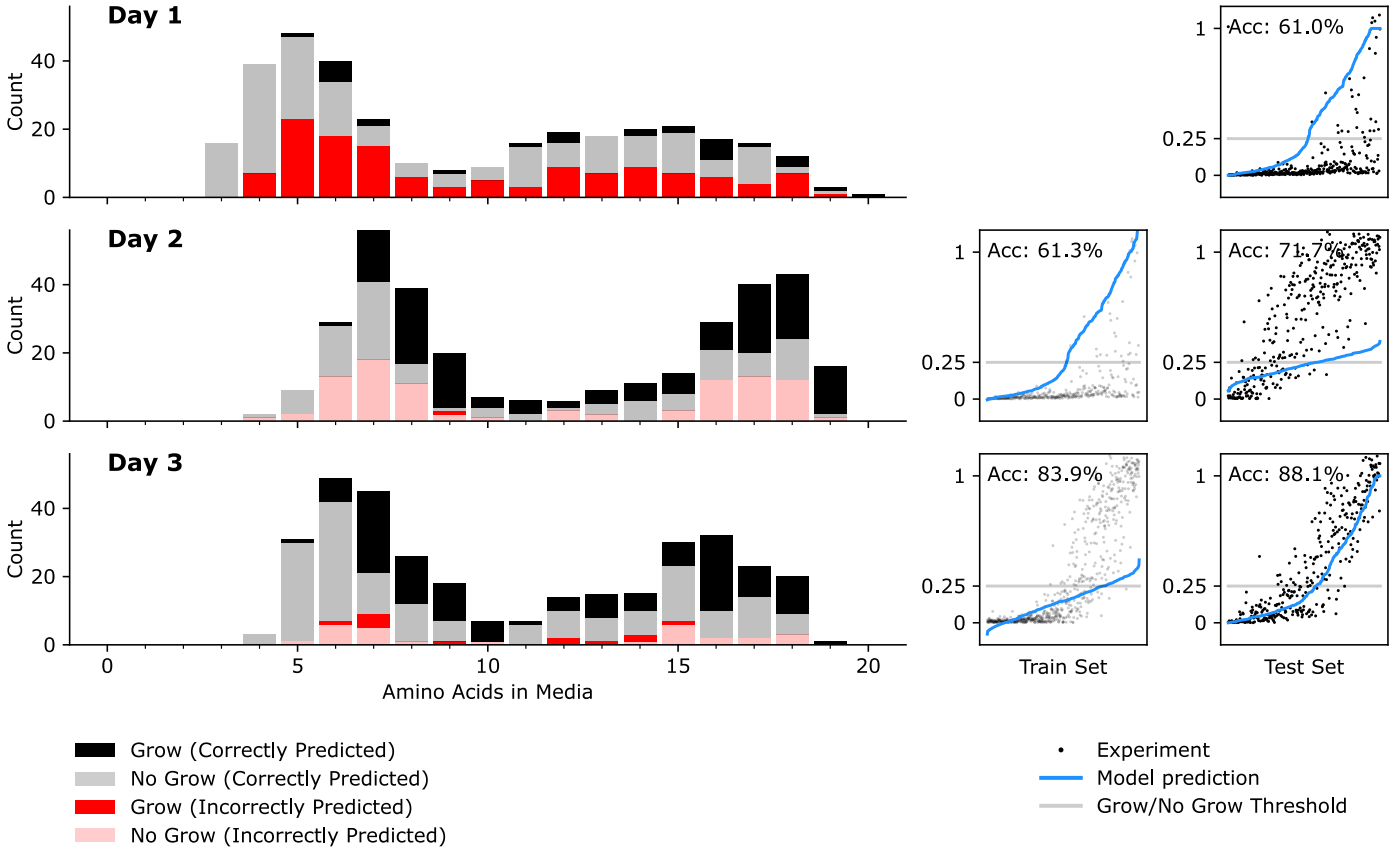
Extended Data Fig. 3 | BacterAI learns the amino acid requirements of *S. sanguinis*. BacterAI selects experiments to learn the amino acid requirements of the bacterium *Streptococcus sanguinis*. Although *S. sanguinis* is genetically similar to *S. gordonii*, the bacteria have different amino acid auxotrophies. BacterAI began its investigation of *S. sanguinis* from a blank slate and did not carry over any knowledge from the previous experiments with *S. gordonii*.

https://doi.org/10.1038/s41564-023-01376-0



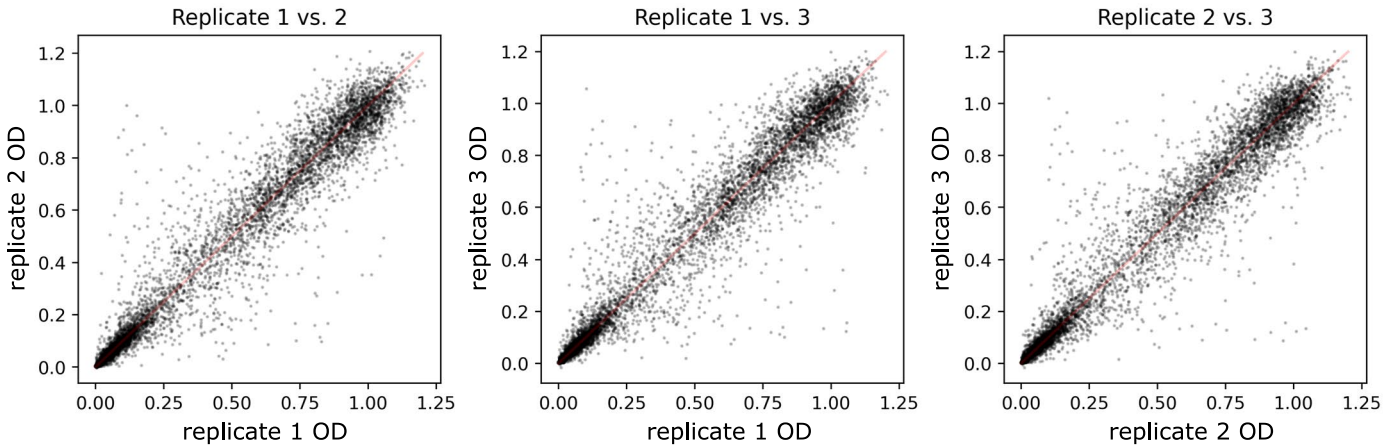
Extended Data Fig. 4 | BacterAI learns amino acid requirements of *S. sanguinis* using transfer learning. BacterAI learns a growth model for *S. gordonii* using transfer learning. A growth model for *S. sanguinis* was used to select the initial experiments and was retrained with new growth data from *S. gordonii*.

https://doi.org/10.1038/s41564-023-01376-0



Extended Data Fig. 5 | BacterAI learns amino acid requirements of *S. sanguinis* in anaerobic conditions using transfer learning. BacterAI learns an anaerobic growth model for *S. sanguinis* using transfer learning. A growth model for *S. sanguinis* in an aerobic (5% CO_2) environment was used to select the initial experiments and was retrained with new growth data from anaerobic experiments.

https://doi.org/10.1038/s41564-023-01376-0



Extended Data Fig. 6 | Replicate growth assays show little variation. Replicates from fourteen days of experiments with *S. sanguinis* show little variation in growth. Using a grow/no grow threshold of 0.25 gives 97.37% (1 vs. 2), 97.11% (1 vs. 3), and 97.39% (2 vs. 3) agreement between the replicates.

nature portfolio

Corresponding author(s): Paul Jensen

Last updated by author(s): Mar 6, 2023

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed	
☐	☒	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☐	☒	A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☐	☒	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐	A description of all covariates tested
☐	☒	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒	A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☐	☒	For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☒	☐	Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	All software is open source and available at https://github.com/jensenlab/BacterAI.
Data analysis	All software is open source and available at https://github.com/jensenlab/BacterAI.

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

All data are available at https://github.com/jensenlab/BacterAI.

Human research participants

Policy information about studies involving human research participants and Sex and Gender in Research.

Reporting on sex and gender	*Use the terms sex (biological attribute) and gender (shaped by social and cultural circumstances) carefully in order to avoid confusing both terms. Indicate if findings apply to only one sex or gender; describe whether sex and gender were considered in study design whether sex and/or gender was determined based on self-reporting or assigned and methods used. Provide in the source data disaggregated sex and gender data where this information has been collected, and consent has been obtained for sharing of individual-level data; provide overall numbers in this Reporting Summary. Please state if this information has not been collected. Report sex- and gender-based analyses where performed, justify reasons for lack of sex- and gender-based analysis.*
Population characteristics	*Describe the covariate-relevant population characteristics of the human research participants (e.g. age, genotypic information, past and current diagnosis and treatment categories). If you filled out the behavioural & social sciences study design questions and have nothing to add here, write "See above."*
Recruitment	*Describe how participants were recruited. Outline any potential self-selection bias or other biases that may be present and how these are likely to impact results.*
Ethics oversight	*Identify the organization(s) that approved the study protocol.*

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	Sample sizes were not determined a priori; instead, BacterAI collected data until its test accuracy reached 90%.
Data exclusions	Batches of data were excluded if they failed QC as determined by a quality engineer who was blinded to the experiment design. No individual data points were excluded.
Replication	The transfer learning runs successfully replicated the initial logical rules found by BacterAI.
Randomization	This study did not assign experimental units to groups for comparison; only the accuracy of models were compared.
Blinding	Blinding was not used since no group assignments were made.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☒	☐ Clinical data
☒	☐ Dual use research of concern

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging