Accelerated design of *Escherichia coli* genomes with reduced size using a whole-cell model and machine learning

Ioana M. Gherman[1], Joshua Rees-Garbutt[1,2], Wei Pang[3†], Zahraa S. Abdallah[1†], Thomas E. Gorochowski[2,4†], Claire S. Grierson[2,4†], Lucia Marucci[1,4*†]

[1]School of Engineering Mathematics and Technology, University of Bristol, Bristol BS8 1TW, United Kingdom.

[2]School of Biological Sciences, University of Bristol, Bristol BS8 1TQ, United Kingdom.

[3]School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh EH14 4AS, United Kingdom.

[4]BrisEngBio, School of Chemistry, University of Bristol, Bristol BS8 1TS, United Kingdom.

*Corresponding author. Email: lucia.marucci@bristol.ac.uk †Joint senior authors.

Whole-cell models (WCMs) are multi-scale computational models that aim to simulate the function of all genes and biological processes within a cell. While WCMs offer deeper insights into how cells behave, they also require significant computational resources to run. Here, we show how machine learning (ML) surrogates can be used to address this limitation by training them on WCM outputs to accurately predict whether cells divide successfully across several generations when a subset of genes are removed (knocked out). Our ML surrogate achieves a 95% reduction in computational time when compared to the original WCM. We then used the surrogate and a genome-design algorithm to generate an *in silico* reduced *E. coli* cell, where 40% of the genes included in the WCM were removed. This approach illustrates how the holistic understanding provided by the WCM can be leveraged for synthetic biology tasks, while mitigating its computational

burden through machine learning.

Introduction

Whole-cell models (WCMs) are comprehensive computational models that aim to capture the dynamics of all core biomolecular processes and components of a cell, as well as their interactions with the wider environment. The development of whole-cell modelling is considered a grand challenge of the 21st century and is seen as particularly useful for understanding the complex relationship between the genotype and phenotype of a cell (*1*). The derivation of WCMs has been made possible by the rapidly increasing availability of experimental data and high-throughput assays that allow researchers to gain a more holistic understanding of biological processes. There are currently three published WCMs: one for *Mycoplasma genitalium* (*2*), one for *Escherichia coli* (*3, 4*) and one for the minimal synthetic cell JCVI-syn3A (*5*). The *E. coli* WCM is represented mechanistically both at the single-cell (*3, 6*) and at the population level (*4, 7*).

Numerous biological discoveries have arisen from the development and utilization of WCMs. For example, Sanghvi *et al.* used a WCM in conjunction with experimental data to examine the growth rates of *M. genitalium* and make predictions concerning kinetic parameters of specific enzymes (*8*). By cross-validating numerous *E. coli* datasets during the construction of the WCM, Macklin *et al.* identified several inconsistencies in the literature, which when resolved, enabled the model to accurately predict *in vitro* protein half-lives (*3*). Additionally, the *E. coli* WCM has recently been applied to better understand the differences between *in vivo* and *in vitro* measurements of tRNA aminoacylation (*9*). The population-level *E. coli* WCM, which considers many independent cells within a shared environment, has been instrumental in assessing antibiotic responses and identifying sub-generationally expressed genes that significantly impact cell survival (*7*). Design-build-test-learn cycles can also be optimised using WCMs, with the *M. genitalium* WCM being used to predict minimal genomes *in silico* (*10–12*). Furthermore, the relationship between gene knock outs, cellular subsystems and phenotypes has been better understood through a combination of WCM simulations and machine learning-based classifiers (*9, 13*).

The comprehensive nature of WCMs enables a deeper understanding of life at the cellular level, but it also makes these models inherently complex. Predominantly, the complexity of WCMs

stems from their aim to simulate the integrated function of every gene and molecule in a cell. For example, the *M. genitalium* WCM captures the functionality of 401 genes and 28 cellular processes that were each modeled independently on a short time-scale and then integrated over longer time-scales (*2*). The more recently published *E. coli* single-cell WCM initially included the functionality of 1214 genes (43% of the organism's well-annotated genes) and 10 cellular processes (e.g. transcription, translation, RNA/protein degradation, complexation, metabolism) (*3*). This model is under continuous development and when this study began the number of genes modelled had risen to 1219. To build the *E. coli* single-cell WCM, Macklin *et al.* identified and curated over 19,119 parameters from the literature and integrated them in a computational model based on over 10,000 interdependent mathematical equations representing different biological systems (*3*). The output of this model consists of approximately 219,000 time series representing the state of every molecule modelled by the WCM over time, the flux of every metabolic reaction, but also other metrics summarising the state of the cell (e.g. the proportion of RNA per gene degraded at each time step, total macro-molecular masses, the probability of each gene being expressed at every time step, the number of ribosomes bound to each gene). An example of a subset of the output of the *E. coli* WCM can be seen in the top panel in Figure 1 where we illustrate how the number of cells simulated grows exponentially as cells divide. Notably, in Figure 1 we only illustrate 3 time series for each cell, but the WCM actually outputs approximately 219,000 time series for each cell.

Despite the richness of information enabled by the detailed temporal tracking of cellular processes and components, this detail also imparts a substantial computational burden. As WCMs are starting to model bigger and more complex organism, their computational complexity can increase as well. To address this, when scaling up from *M.genitalium* to *E. coli*, Macklin *et al.* implemented several computational improvements. For example, the simulation run-time was decreased by two orders of magnitude, from approximately 10 hours for *M.genitalium* WCM to 15 minutes for *E. coli* WCM. This was achieved by writing inner loops in Cython and C, warm-starting the linear solver of the metabolic model and improving the file I/O (*3*). This boost in runtime together with the richness of *E. coli* data in the literature, allowed Macklin *et al.* to model cell division and growth such that several generations of cells can be simulated (*3*). This suggests that despite simulation of the *E. coli* WCM being much faster than for the *M. genitalium* model, the increase in the number of modelled genes and the ability of the model to simulate several cell divisions (i.e. generations)

enlarges the possible combinations that have to be tested when using these models to predict phenotypes from genotypes. The complexity of WCMs in general stems from the massive number of time series they generate, tracking the tens of thousands of variables (i.e. molecule counts) that are present. Nevertheless, in many applications, researchers are only interested in a specific subset of these outputs (e.g., the overall cellular mass). ML surrogates represent a solution to address this computational burden by exclusively making predictions for the outputs of interest after being trained on data generated by a WCM.

ML surrogates, or emulators as they are sometimes known, are black-box models trained to approximate the behaviour of complex and computationally expensive mathematical models (*14*). The major advantage of their use is the reduced computational resources required, thus allowing the simulation time to fall from hours or days to seconds. ML surrogates are trained using data simulated by mechanistic models, taking as input their initial conditions and/or parameters, and are designed to predict some or all of the original model's outputs. The choice of ML algorithm depends on the nature of the problem, the amount and type of available data, and the desired accuracy. ML surrogates have been used to emulate several biomedical mechanistic models (*15–17*) and some molecular biology systems (*18, 19*), allowing for efficient exploration of a mechanistic model's parameter space (*20, 21*), as well as supporting model optimisation (*22*) and sensitivity analysis (*19*). ML surrogates have emulated mechanistic models based on Partial Differential Equations (PDEs) (*15–17*). Ordinary Differential Equations (ODE)-based models have only been used to show how a surrogate can be trained, but no significant improvement in computational time was observed for the system analysed to date (*18, 19*).

In this study, we use a random forest-based ML surrogate to simulate *E. coli* cells with reduced genomes (i.e. genomes containing fewer genes). Experimental (*23–25*) and *in silico* (*5, 10, 26, 27*) studies have aimed to design reduced genomes for different organisms, removing as many genes as possible, while ensuring the cell remains viable (i.e. is able to divide and replicate). WCMs have already been used to design minimal genomes *in silico* for *M. genitalium* (*10*), which was possible due to the small size of the genome (401 genes modelled) and the fact that the cells were simulated for only one generation (*10*). In contrast, the *E. coli* single-cell WCM (used here) captures the functionality of 1,219 genes and can be run for as many generations as we choose (*3*). Given this additional complexity, we estimated that using a similar approach to the one used by Rees-Garbutt

et al. for *M. genitalium* would not be possible for *E. coli* without reducing the simulation time. To further illustrate this, consider that we want to simulate 10 genomes using the *M. genitalium* WCM. If there are 10 nodes available on a computing cluster, this will require approximately 10 hours because all 10 genomes can be run in parallel. If we want to simulate 10 genomes using the *E. coli* WCM for six generations (this study will show why this is an optimal number of generations) and we assume that all cells in six generations divide, this means that we will simulate 630 cells in total (63 cells for each genome because at every division a cell produces two daughter cells). If we have 10 cores available simulating the 630 cells would take 15.75 hours (each cell simulation depends on the simulation of the mother cell, therefore the generations need to be run sequentially and the number of cells grows exponentially with the generations). Given that the *E. coli* WCM contains three times more genes compared to the *M. genitalium* WCM and the combinatorial aspect of gene deletions when searching for a minimal genome, simulating tens of thousands of genomes for six generations using the *E. coli* WCM is not feasible without using a surrogate model. To the best of our knowledge, the present study is the first to show how an ML surrogate can emulate a large and complex biological model, that integrates different mathematical modelling techniques and contains over 10,000 equations. Furthermore, this represents the first example of how the *E. coli* WCM can be used for synthetic biology applications, specifically how it is possible to leverage the power of such models through the richness of information that they provide, while making them more accessible by addressing the computational challenges that they can pose.

Results

Data preparation for the ML surrogate

The ML surrogate presented in this study, trained on data produced by the *E. coli* single-cell WCM (*3*), aims to predict whether a reduced genome is producing viable cells that successfully divide for several generations. Before we could train the ML surrogate, it was necessary to perform some data pre-processing on the WCM outputs. Firstly, we needed to decide how long the simulations should be run for (e.g., the number of generations) and what outputs/states from the WCM simulations to use to train the ML surrogate. Preliminary experiments where we simulated cell growth for

four generations showed that in some cases all the cells divided but also exhibited significant growth defects, especially in protein mass (Figures S1 and S2). This suggested that more than four generations should be simulated in order to accurately estimate the viability of a genome. The analysis presented in the Supplementary Text showed that 96% of the cells that failed to divide in eight generations, actually did so in the first six generations (Figure S3) and therefore this number of generations was used to generate the ML surrogate training data.

Additional decisions had to be made regarding the features to be used. Given the high dimensionality of the WCM, which includes over 10,000 equations, 19,000 parameters, and generates more than 219,000 time series, we focused our analysis on the 22 different macromolecular masses produced by the WCM. These masses, such as the total mass of all protein molecules, RNA molecules, DNA molecules, or membrane mass (see the description of Table S1 for the full list), offer a concise summary of the cell's state. Collectively, they provide insights into growth and division, metabolic activity, genetic stability, and environmental interactions. We compared the accuracy of different ML surrogates using the values of these variables (i.e. features) either at the beginning of the simulations ($t = 0$, generation 1) or at the end of generation 1 simulation, just before the first cell divides. The results show that the three algorithms tested (Random Forest, XGBoost and K-Nearest Neighbours) perform similarly, with Random Forest showing slightly higher accuracy irrespective of the features used (Table S1). We saw an improvement of approximately 17% in the F1-score of the training and test sets when using values at the last time step of generation 1 as features. Therefore, for the rest of this study, we chose to use as features the state of the cell at the last time step before the first division. A summary of the training procedure that will be used in this study can be seen in Figure 1 and the data preparation techniques are summarised in the Supplementary Text. An overview of the simulations ran to train the ML surrogates is presented in Table S2.

Upon further investigation, we found that many of the macromolecular masses used as features to train the ML model were highly correlated, suggesting that not all 22 features were necessary. Figure S4 shows the correlation matrix between all the 22 features. These values were used to select only those features with a Pearson correlation < 0.95, allowing us to reduce the number of features used to train the ML model to 12 (cell mass, growth, DNA mass, tRNA mass, extracellular mass, protein mass, projection mass, pilus mass, mRNA mass, small molecule mass, instantaneous growth rate, membrane mass). This reduction did not affect the accuracy of the ML surrogate.

Training the ML surrogate

We ran 924 single gene knock out simulations corresponding to the knock out of 462 genes, each run twice, of which 738 were used to train the ML surrogate and 186 to test it (Supplementary Table S4). The 12 key macromolecular masses (identified following the correlation analysis) at the last time point of the first generation were then used by a random forest classifier to predict whether all cells in 6 generations divide (Figure 1).

A random forest classifier was chosen due to its superior performance (Table S1) but also due to its simplicity, explainability and robustness (*28*). One of the primary strengths of random forest classifiers stems from its high accuracy, achieved by aggregating the predictions of multiple decision trees, which reduces the risk of overfitting compared to single decision trees. This ensemble approach allows random forest to handle complex, non-linear relationships between features and the target variable effectively. It is also highly robust, capable of managing outliers, missing values, and high-dimensional datasets without requiring extensive preprocessing like feature scaling. Additionally, random forest provides insights into feature importance, which can be valuable for understanding data and guiding feature selection.

To assess the random forest model proposed here we used k-fold cross-validation on the training set to tune the model's parameters and avoid overfitting. We also tested the model's performance on a hold-out test set (containing data not used when training). We obtain an F1-score of 0.92 on the training set, 0.9 when averaging the cross-validation results, and 0.93 on the hold-out test set (Table S3). The corresponding confusion matrices show that when the ML model predicts that not all cells in 6 generations will divide (i.e. the gene knock out impacted viability), it is correct for 259 out of 261 cases (99%) of the data points in the training set, and 74 out of 76 (98%) of the data points in the test set (Figure 2A). In the context of minimal genome design, this suggests that guided by the ML surrogate, it is possible to limit the removal of essential genes, that upon deletion from the genome cause failure in the division of at least one cell in six generations.

Our random forest classifier was trained, validated and tested repeatedly on these selected 12 features and the results showed that only four of them (cell mass, DNA masss, extracellular mass and membrane mass) were necessary for making an accurate prediction. One of the advantages of using a random forest classifier is that it also ranks the importance of the features used for a prediction.

Figure 2B shows the ranking of the features used by the ML model. More details about the impact of the features on the classifier and the misclassified data can be found in the Supplementary Text.

This initial random forest surrogate was trained on single gene knock outs and tested on both single and double gene knock outs. The F1-score when testing the surrogate on double gene knock outs was 0.93 and there were no false positives (Figure 2A, Table S3). However, testing this initial ML surrogate on multiple gene knock outs (necessary for generating a reduced genome) resulted in an F1-score of 0.82 (Figure 2A, Table S3), a drop in performance when compared to the single and double gene knock out results. Figure 2C shows a scatter plot of the main two features used by the classifiers (at the end of generation one, just before cell division). We observed that when deleting more genes, the cell and DNA masses become more similar among the deadly and non-deadly knock outs. Specifically, both the DNA and cell mass of the deadly knock outs are higher compared to when we knock out single genes. This suggests that in order to classify multiple gene knock outs, the surrogate has to be retrained using more features and data representing multiple gene knock outs.

To address this, the surrogate was retrained by extending the initial training set of 738 single gene knock outs to additionally include 248 multiple gene knock outs, each run once for 6 generations (a list of the genes knocked out can be found in Supplementary Table S5). The F1-score when tested on single and double gene knock outs remained similar, but when tested on the hold-out set of multiple gene knock outs (that were not used for training) it was 0.88, representing a 0.06 increase in F1-score (Table S3, Figure 3A) despite dealing with a much more challenging task compared to classifying single gene knock outs. When retraining using the multiple gene knock outs, we used 11 of the uncorrelated macromolecular masses obtained following the correlation analysis described previously (the instantaneous growth rate was removed due to its similarity to the growth output perceived by the random forest). Figure 3B shows that the DNA mass, the membrane mass and the projection mass are essential for making accurate predictions. More information on the data used to retrain the ML surrogate can be found in (*29*).

Genome reduction

To demonstrate the value of the trained surrogate model, we showed that it can be used to design a reduced *E. coli* genome that still functions (i.e. divides) *in silico*. To do this, we used Minesweeper, an algorithm inspired by laboratory fragment-cassette-fraction genome engineering (*25*) that employs a divide and conquer approach, which was used previously to find minimal genomes using the *M. genitalium* WCM (*10*). Minesweeper requires simulating single gene knock outs and thousands of multiple gene knock outs. To do this, we used our trained surrogate model.

Figure 4 shows the framework used to obtain the reduced genome using Minesweeper (*10*). This algorithm was used to identify genes that can be removed from the genome, without losing the cell's capability to divide successfully in six generations. Stage one of Minesweeper predicted the essentiality of every single gene in the WCM. This was done using the initial random forest surrogate trained only on single gene knock outs for all modelled genes (1219), as the F1-score was sufficiently high for single gene knock outs. The algorithm predicted that 1047 genes were non-essential and 172 genes were essential (Supplementary Table S6). Only the 1047 genes predicted as non-essential were considered for future deletions.

In stage two, we created 56 groups containing different percentages of genes to be removed from the genome (this is a modification compared to the previous version of Minesweeper, see (*29*) for details). The simulations corresponding to the removal of these groups were run using the WCM (simulating six generations). Out of the 56 deletion groups, the removal of 20 of them produced dividing cells in six generations. These 20 groups correspond to two 12.5% groups, four 10% groups and fourteen 5% groups.

At this point the ML surrogate was retrained using combinations of these 20 segments as described in (*29*). Next, stage three took the three largest segments that the WCM suggested can be deleted in stage two (two 12.5% groups and one 10% group) and created powersets: all possible combinations for each of these largest segments and the remaining division-producing segments from stage two that do not overlap with the largest segment. The retrained ML surrogate was used at this stage to predict whether all cells in six generations divided when these combinations were deleted. To make sure that the largest deletion that divided was carried to the next stage of Minesweeper, we tested the top 20 largest deletions using the full six generations simulations of

the WCM.

Stage four is cyclical. It started by taking the largest segment from stage three (for each of the three segments tested), which when removed produced a dividing cell, and it generated eight groups from the remaining genes (not included in the segment). After that, it created a powerset of combinations between these groups of genes and the current largest deleted segment. If some deletants formed by the removal of these combinations divided, it updated its record of the largest deletion segment and the remaining genes accordingly. If none of the combinations divided, it appended individual genes to the previous largest deletion segment. If some of these divided, it updated the list of remaining genes. If none divided, the algorithm stops. We ran this cycle three times starting from the largest deletions of each of the three groups from stage three. The retrained ML surrogate was used to make predictions for these three iterations, after which additional data was added to the training set and the surrogate was trained once again (see (*29*)). These three cycles produced three reduced cells (i.e. ignoring the single gene additions to the largest deleted set from stage four in Figure 4). The size of these reduced cells varies between 803 genes at the beginning of stage four simulations and 737 genes at the end.

The final and smallest reduced genome, EMine-737, contained 737 genes out of the 1219 modelled by the WCM (Supplementary Table S7). This represents a 40% reduction of the WCM's genome in minimal M9 media supplemented with 0.4% glucose. To obtain this result, we simulated 1525 genomes running the WCM for six generations and 39,086 genomes running the WCM for one generation and predicting division using the ML surrogates. Assuming that we are running simulations using one CPU on a standard desktop computer and that all cells in six generations would have divided, running all simulations (1525 for six generations and 39,086 for one generation) would have taken approximately two and a half years. Without the ML surrogate (running all simulations for six generations) would have taken approximately 36 years, resulting in a reduction in computational time of 94%. The simulations used in this study were run in parallel, on a supercomputer, where possible.

Reduced *in silico* genome (EMine-737) analysis

Reduced genome growth analysis

The reduced genome was simulated 50 times for six generations (each dividing cell producing two daughters) using the WCM to assess its stability. The results showed that for 96% of repetitions, all cells in six generations divide. We compared the growth and masses of one EMine-737 genome to those of one wildtype genome, both allowed to divide for six generations and producing two daughter cells at each division (Figure 5). We found that the median cell mass of the EMine-737 cells is 1309.4 fg compared to 1571.3 fg for wildtype cells. The reduced genome cells also grow more slowly (0.14 median growth rate of EMine-737 compared to 0.22 for wildtype cells) and therefore take longer to divide compared to their wildtype counterpart, which may be caused by the fact that the cells simulated by the WCM need to reach a certain mass before they divide. Figure 5 suggests that the RNA and protein mass of the reduced genome cells are often higher than those of wildtype cells (with a median protein mass of 232.9 fg for EMine-737 cells and 214.5 fg for the wildtype cells and a RNA mass of 104.6 fg for EMine-737 and 73.9 fg for wildtype cells).

To further assess the stability of the EMine-737 genome, we also ran 10 generation simulations for 50 repetitions (seeds) using the WCM, where each dividing cell produced only one daughter (to ensure reasonable computational demands). The results show that for 90% of these 50 repetitions, all cells in 10 generations divided, compared to 98% for wildtype *E. coli* simulations (10 generations, 50 repetitions).

Gene ontology analysis

The full *E. coli* WCM has 1219 genes, corresponding to 142 complete gene ontology terms (i.e. where all genes corresponding to those terms are included in the WCM). Eighteen GO categories no longer had any categorised gene in the EMine-737 genome and 72 categories had only some of their original genes included. The eighteen categories with all categorised genes removed from the genome correspond to ten biological processes (representing different catabolic and metabolic processes, as well as the import of different substances across the plasma membrane), four cellular components (represented by genes involved in different complexes) and four molecular functions (representing genes involved in activities such as oxidoreductase, dehydrogenase and hydratase).

All these gene ontology terms and their corresponding genes can be found in Supplementary Table S8.

The cells simulated in this study are grown in M9 media with glucose under aerobic conditions. Therefore, they will primarily rely on glucose metabolism pathways such as glycolysis and the tricarboxylic acid (TCA) cycle for energy production and carbon metabolism, suggesting that genes involved in the breakdown and metabolism of other sugars might not be essential in this environmental context. To exemplify this, the gene ontology analysis of the genes removed from EMine-737 showed the genes involved in the glycolate catabolic process, the glyoxylate catabolic process and the glyoxylate metabolic process are removed from EMine-737. The glycolate degradation pathway that feeds into glycolysis is essential for *E. coli* to grow on glycolate or glyoxylate substrate (*30, 31*), but not when glucose is the carbon source. Another two processes removed from EMine-737 that involve the breakdown or synthesis of sugars are the D-tagatose 6-phosphate catabolic process and the dTDP-rhamnose biosynthesis (*30, 32, 33*). These two processes are not directly involved in the metabolism of glucose, so their absence is unlikely to have an immediate impact on *E. coli*'s ability to utilize glucose as a carbon source for energy production. Similarly, the L-ascorbic acid degradation process was also affected in EMine-737 by the removal of the *sgbU, ulaE* and *ulaF* genes, which in aerobic conditions may affect the cell's capability to use L-ascorbic acid as a carbon source, however since the media is supplemented with glucose, this is unlikely to be a problem (*30*).

Another set of gene ontology terms removed from EMine-737 include the transport of amino acids such as L-histidine and L-glutamine across the plasma membrane. Despite this process being impaired in EMine-737, *E. coli* is likely able to synthesise these amino acids internally. In fact, all the genes involved in the biosynthesis pathway of L-histidine (*hisABCDFGHI*) and L-glutamine (*glnA*) are still present in the EMine-737 genome. Furthermore, Figure S5 shows that the two amino acids are still being produced in EMine-737, although in smaller amounts compared to the wildtype cells (with a median L-histidine molecule counts of 52,483 for EMine-737 and 75,704 for wildtype, and median L-glutamine molecule counts of 1,483,958 for EMine-737 and 2,096,906 for wildtype).

Several ontology classes corresponding to the respiration and oxidase complexes (phenylacetyl-CoA 1,2-epoxidase complex, cytochrome o ubiquinol oxidase complex, respiratory chain complex I), as well as oxidoreduction and oxidoreductase activities were removed in EMine-737 as well. The

removal of the epoxidase complex is likely to affect the cell's ability to metabolize certain aromatic compounds efficiently (*34*), but these do not seem to significantly affect the ability of the cell to grow or to produce ATP (Figures 5 and S5). *E. coli* can use alternative terminal oxydase pathways to cytochrome o ubiquinol oxidase complex and respiratory chain complex I, in aerobic conditions, for energy production, such as the cytochrome *bd*-I and cytochrome *bd*-II complexes (*30*). The genes involved in the oxidoreduction transmembrane activity (one of the molecular functions removed from EMine-737) are the same as those involved in the cytochrome o ubiquinol oxidase complex and they are classified in the literature as non-essential (*30*).

The dimethyl sulfoxide metabolic process is also removed from EMine-737, which represents a terminal electron acceptor in anaerobic conditions (*35*), but without significant activity in aerobic conditions. Similarly, the plasma membrane fumarate reductase enzyme complex removed in EMine-737 is catalysing the reduction of fumarate to succinate in anaerobic conditions, but its absence in aerobic conditions should not affect the cell significantly (*36*). The other two molecular function ontology terms removed from EMine-737 involve a combination of the genes *fadBIJ* and *paaFHZ*. The deletion of these genes might cause the cell to exhibit a defect in the utilization of phenylacetate (in the case of *paaFHZ* deletions) and fatty acids (in the case of *fadBIJ* deletions) as sources of carbon (*30*). However, this is probably addressed by the presence of glucose in the growth media.

Reduced genome literature comparison

The construction of the *E. coli* WCM involved the integration of data from three distinct laboratory strains: K-12 MG1655 , B/r and BW25113 (*3*). This approach was necessary due to the insufficient amount of data available from a singular strain for the development of a comprehensive WCM. This multi-strain data integration strategy might potentially constrain the capacity for direct comparison with other reduced *E. coli* genomes that are documented in the scientific literature.

A number of reviews have compared existing laboratory reductions of the *E. coli* genome (primarily in K-12 MG1655 and K-12 W3110 strains) (*37–39*), but only three studies tested the reduced MG1655 genomes in minimal media (*40–42*). First, the MDS12 strain built by Kolisnychenko *et al.* in 2002 deleted 12 genomic segments from MG1655 (8.1% deletion) without impacting growth rates in minimal media (*40*). Then Posfai *et al.* removed additional regions from the genome to

obtain the MDS43 strain (15.3% of the genome was removed) (*41*). Later, Park *et al.* removed another 13 genomic regions to obtain the MS56 strain with 23% of the original genome deleted (*42*). Since MS56 contained the gene deletions from MDS12 and MDS43, we only compare EMine-737 to MS56.

The *E. coli* WCM includes only 16.5% of the genes removed in MS56 (151 genes). From these, 98 were also removed from the *in silico* reduced genome EMine-737. There are 383 genes removed from EMine-737 that are not removed from MS56. Figure 6A shows a comparison between the genes of EMine-737, the genes included in the *E. coli* WCM and the genes included in MS56, showing that the genomic regions deleted to obtain EMine-737 are different to the ones deleted in MS56. Figure 6B shows the percentage of genes included in each of the three genomes with respect to the total number of genes of *E. coli* MG1655 strain. Given that the version of the *E. coli* WCM used in this study includes the functionality of a fraction of the total number of genes, it is unlikely that the EMine-737 will be viable *in vivo*. However, a subset of the genomic regions removed to obtain EMine-737 could represent a new starting point for deriving minimal genomes *in vivo*; similar to the way MDS12 represented a starting point for deriving MDS43 and MS56 (*40–42*). A list of the MS56 genes, the WCM genes and the genes deleted in EMine-737 are presented in Supplementary Table S9.

The 383 genes that have been removed only in EMine-737 and not in MS56, together with their corresponding gene ontology terms are presented in Supplementary Table S10. Among the gene ontology terms that have been discussed in the previous section and that are fully represented by a subset of the 383 genes are: L-histidine import across plasma membrane, dimethyl sulfoxide metabolic process, respiratory chain complex I, plasma membrane fumarate reductase complex, cytochrome o ubiquinol oxidase complex, oxireductase activity acting on NADPH, quinone or similar compounds and oxidoreduction-driven active transmembrane transporter activity.

Discussion

In this work, we showed that by using an ML-based surrogate to emulate the predictions of an *E. coli* WCM, it is possible to enable the use of such models for synthetic biology tasks, such as the search for *in silico* reduced genomes. Specifically, we showed that a search that on a typical

desktop PC would take over 36 years, can be reduced to approximately 2.5 years. With access to a supercomputer that allows running up to 40 simulations in parallel, the equivalent simulation time using only the WCM is 666.2 days and using the WCM together with the surrogate is 35 days, showing that by using the surrogate it is possible to achieve a 95% decrease in computational time (see (*29*) for more details). These results emphasise that even with the help of supercomputers, WCMs are prohibitively expensive and difficult to use in scenarios that require simulations of many genomes without the use of surrogate models. Here, for designing reduced genomes, we used our ML surrogate model to predict a binary variable (cell division), but we are confident that this approach can be expanded to predict continuous variables such as metabolic yield, or to forecast time series corresponding to some variables of interest. To predict continuous variables, it would be necessary to train regression-based surrogate models, while for time series forecasting, deep learning approaches such as Long Short Term Memory (LSTM) networks could be used. These methods could be useful when using the *E. coli* WCM for metabolic engineering, particularly when many simulations need to be run.

Typically, ML surrogates are trained on initial conditions and/or the parameters of mechanistic models. In this study, this was not possible due to the high dimensionality and complexity of the WCM. We would have needed more than 10,000 training examples to avoid having more features than training data. Instead, the surrogate presented here was trained using feature variables that summarise the state of the cells (i.e. macromolecular masses). Moreover, rather than using the initial state of these variables, we chose to use the state at the end of the first generation of each genome simulation, which increased the F1-score of the classifier by 16-18%, as shown in the data preparation section. This suggests that there is not enough information regarding the cellular dynamics at the beginning of each simulation to make an accurate prediction. Even with this approach, the number of cells that had to be simulated for each individual genome was reduced from 63 when running only the *E. coli* WCM for six generation to one using the ML surrogate (assuming all cells in six generations divide).

The performance of our surrogate, with a F1-score of 0.93 on single gene knock outs test set, suggests that the cell mass, DNA mass, extracellular mass and membrane mass at the end of generation one hold enough information to predict the future viability of the daughter cells for another 5 generations. Regarding the false negative predictions made by the ML surrogate,

they are likely caused by the fact that the decrease in cell and DNA mass is not obvious enough at the end of generation one, resulting in the ML surrogate misclassifying the knock out. Gene ontology analysis performed on the misclassified genes (Supplementary Table S11), did not reveal any obvious pattern regarding a particular type of genes that are more likely to be misclassified. False negatives are not a problem in the context of genome reductions, as misclassified knock outs will still be identified in later stages of reduction. In contrast, false positives are more problematic, as they decrease the potential pool of genes to be deleted when searching for a minimal genome. The ML surrogates presented in this study have a small false positive rate of 1-2%. The performance of the ML surrogate could potentially be further improved by building a time series classifier that takes as input the whole time series representing the macromolecular masses of first generation cells and predicts whether all cells in the future generations divide.

The genome obtained by using a combination of six generations of WCM simulations and the ML surrogates was reduced by 40%, from 1219 genes to 737. A direct comparison to the literature *in vivo* results is not possible because the WCM only includes the functionality of a fraction of *E. coli's* MG1655 genome. Despite this limitation, the 383 genes deleted only in EMine-737 genome (i.e. not removed as part of MS56's genome reduction) should be analysed further and tested experimentally (Supplementary Table S10). Mathematical models are not perfect and we are aware of the current limitations of the *E. coli* WCM (e.g. not all genes are modelled yet, the WCM was built using data from several strains, it was not tuned to perform genome design tasks).. However, we believe that the reduced genome data generated in this study could guide future experimental avenues for obtaining smaller *E. coli* genomes.

In this study the Minesweeper algorithm was stopped before converging due to the high number of simulations already run (over 39,000) and resulting time constraints, emphasising the importance of using a surrogate. However, it is possible that if continued, this approach can lead to a further reduced genome. Additionally, as more genes are added to the *E. coli* WCM, the number of algorithmic iterations needed to search for reduced genomes will increase. As WCMs become more powerful and better at matching cellular phenotypes, their computational complexity needs to be addressed for genome design tasks, making the use of ML surrogates timely and advantageous.

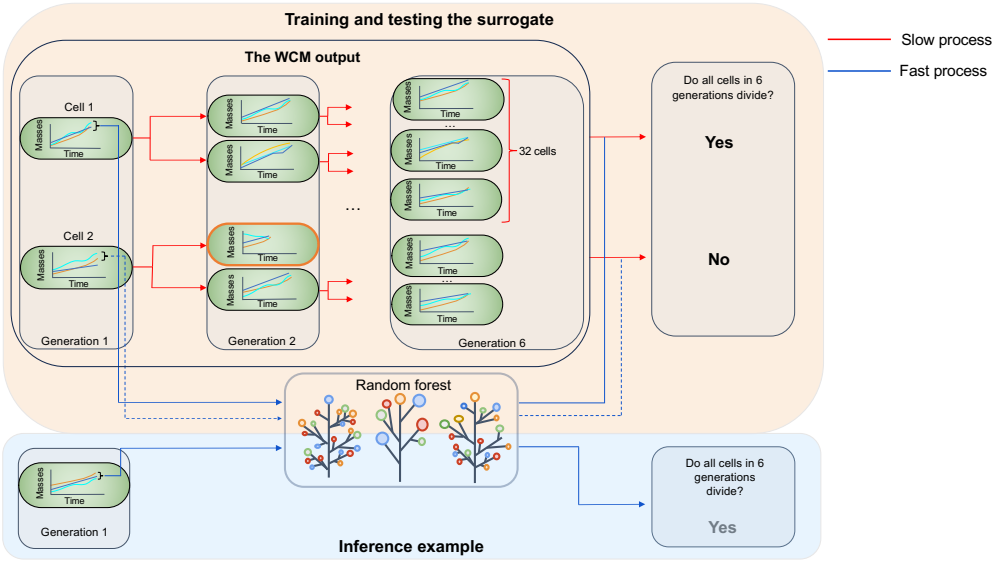

Figure 1: **An example of a subset of the outputs of the *E. coli* WCM and the framework used to train a machine learning classifier that emulates the output of the WCM.** Simulations start from two cells (Cell 1 and Cell 2) that are simulated for six generations. A random forest model is trained to emulate the behaviour of the *E. coli* WCM (from generations 2-6). Data from the WCM output is used to train and test the machine learning surrogate. Simulating the WCM for six generations is computationally expensive and time consuming (red arrows). Therefore, we use the value of the macromolecular masses at the end of generation one to train a random forest model that predicts whether all the cells in six generations will divide (e.g. Cell 1 and all its replicates) or not (e.g. for Cell 2 the orange replicate in generation 2 did not divide). Once the model is trained, for future use, only one generation of the WCM will have to be run (instead of six), and the ML surrogate can be used to predict whether all the cells in six generations divide. This process is significantly faster (blue arrows).

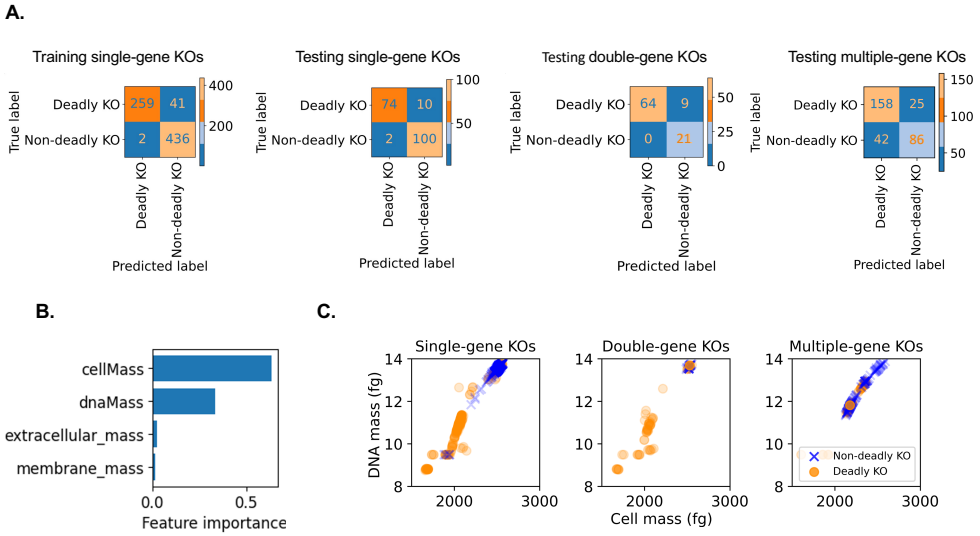

Figure 2: **Results of the initial ML surrogate trained on single gene knock out (KO) data**. (**A**) The confusion matrices for the training of the ML surrogate on single gene knock outs, and testing on single, double and multiple gene knock outs. The true label is based on the output of the WCM and the predicted label is represented by the output of the ML surrogate. (**B**) The feature importance as outputted by the random forest algorithm trained on single gene knock outs. (**C**) Scatter plots of the two main features used for learning show different data sets containing single, double and multiple gene knock outs.

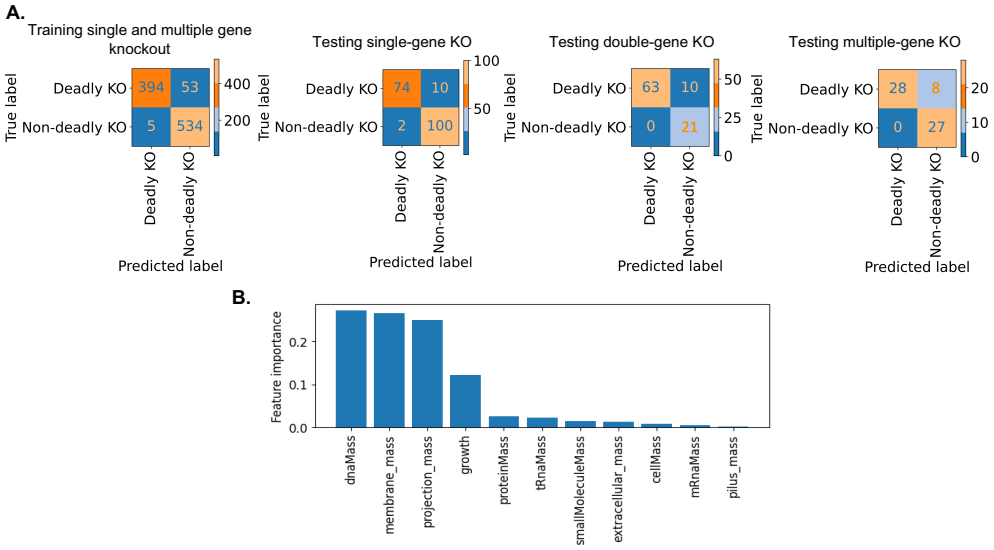

Figure 3: **Results of the updated ML surrogate trained on single and multiple gene knock outs data**. (**A**) The confusion matrices for the training and testing of the ML surrogate on single, double and multiple gene knock outs. The true label is based on the output of the WCM and the predicted label is represented by the output of the ML surrogate. (**B**) The feature importance as outputted by the random forest algorithm trained on single and multiple gene knock outs.

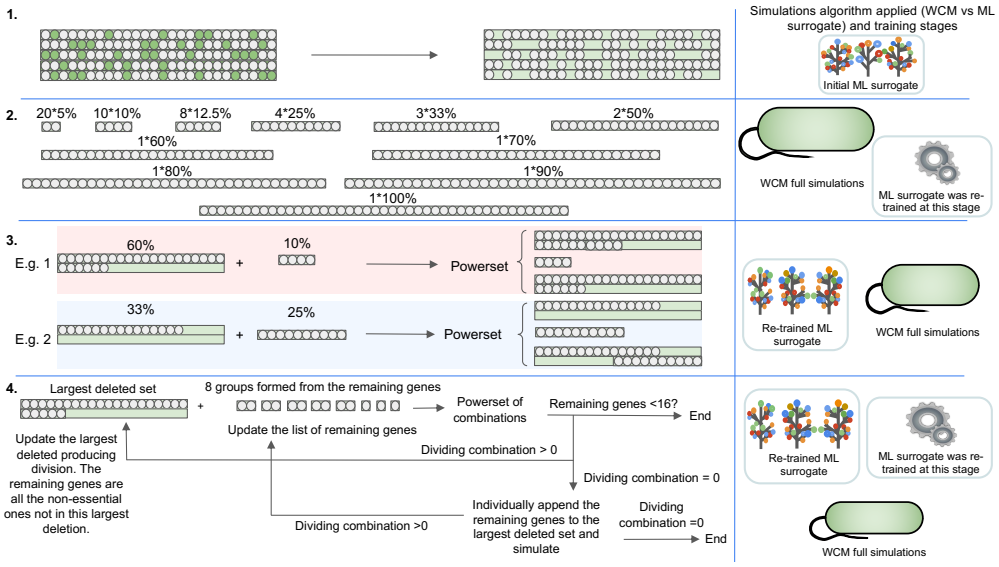

Figure 4: **Summary of the framework used to generate the minimal genome**. This is an adaptation of the Minesweeper framework (*10*), using the *E. coli* WCM (*3*) and the ML surrogate. The process started by using the ML surrogate trained on a subset of single gene knock out data to predict the essentiality of every single gene knock out (not included in the training set). Only the genes characterised as non-essential go to stage two (grey circles). In stage two, 56 groups representing different percentages of the non-essential genes were deleted individually from the genome, and the corresponding simulations were run for 6 generations using the WCM. Permutations of these groups were used to re-train the ML surrogate on multiple gene knock outs. In stage three, we show examples where we took the two largest deletion segments that when removed produced cells that divided (red and blue), appended all combinations of the division-producing non-overlapping segments from stage two, created a powerset of combinations from these segments and ran the corresponding simulations where they were removed (not all combinations shown). The fourth stage is cyclical and predictions were made using the retrained ML surrogate, randomly testing the dividing genomes using the six generations WCM simulations and retraining once.

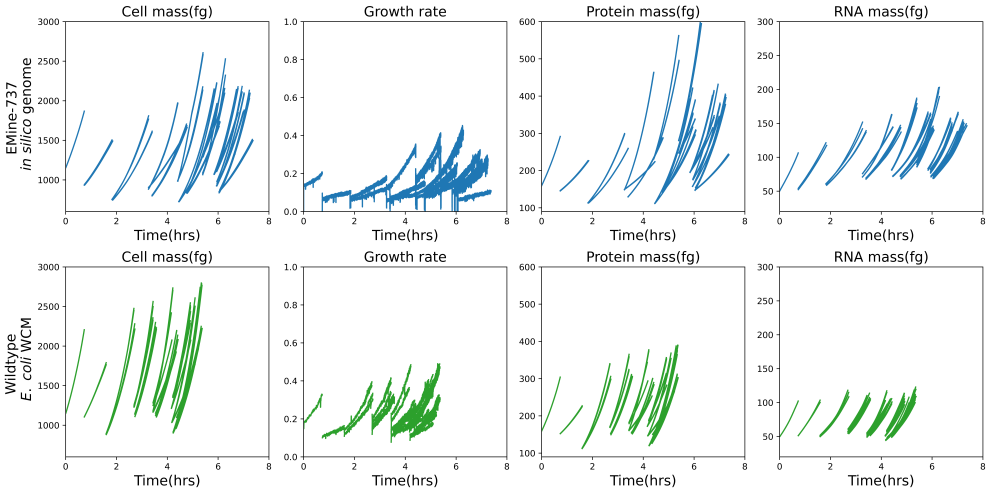

Figure 5: **An example of the WCM output for one repetition of an EMine-737 genome and one repetition of a wildtype genome.** The EMine-737 (first row) and wildtype (second row) genomes are allowed to divide for 6 generations, with each cell division producing two daughters. Each line in the graphs represents the cell mass, growth rate, protein mass or RNA mass of a cell.

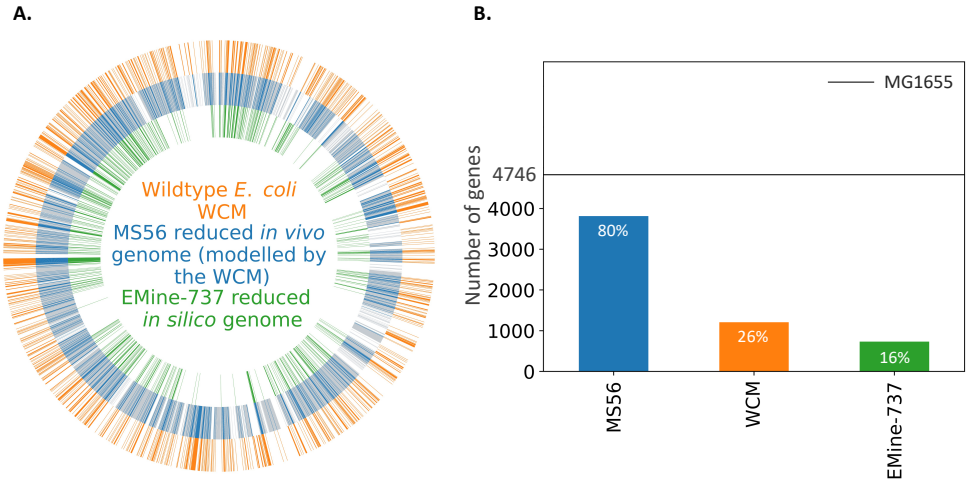

Figure 6: **A comparison between the genome of EMine-737, the *E. coli* WCM wildtype genome and the MS56 genome.** (**A**) Genes included in the *E. coli* WCM (the outter circle in orange), the genes included in MS56 that are also modelled in the WCM (*42*) (the middle circle in blue), the genes present in MS56, but not modelled by the WCM (the middle circle in grey) and the genes included in EMine-737 (the inner circle in green). (**B**) Number of genes included in each of the genomes. The percentages are calculated relative to the size of the MG1655 genome.

References

1. M. Tomita, Whole-cell simulation: a grand challenge of the 21st century. *TRENDS in Biotechnology* **19** (6), 205–210 (2001).

2. J. R. Karr, *et al.*, A whole-cell computational model predicts phenotype from genotype. *Cell* **150** (2), 389–401 (2012).

3. D. N. Macklin, *et al.*, Simultaneous cross-evaluation of heterogeneous E. coli datasets via mechanistic simulation. *Science* **369** (6502), eaav3751 (2020).

4. E. Agmon, *et al.*, Vivarium: an interface and engine for integrative multiscale modeling in computational biology. *Bioinformatics* **38** (7), 1972–1979 (2022).

5. Z. R. Thornburg, *et al.*, Fundamental behaviors emerge from simulations of a living minimal cell. *Cell* **185** (2), 345–360 (2022).

6. T. A. Ahn-Horst, L. S. Mille, G. Sun, J. H. Morrison, M. W. Covert, An expanded whole-cell model of E. coli links cellular physiology with mechanisms of growth rate control. *NPJ Systems Biology and Applications* **8** (1), 30 (2022).

7. C. J. Skalnik, *et al.*, Whole-cell modeling of E. coli colonies enables quantification of single-cell heterogeneity in antibiotic responses. *PLOS Computational Biology* **19** (6), e1011232 (2023).

8. J. C. Sanghvi, *et al.*, Accelerated discovery via a whole-cell model. *Nature methods* **10** (12), 1192–1195 (2013).

9. H. Choi, M. W. Covert, Whole-cell modeling of E. coli confirms that in vitro tRNA aminoacylation measurements are insufficient to support cell growth and predicts a positive feedback mechanism regulating arginine biosynthesis. *Nucleic Acids Research* (2023), gkad435, doi: 10.1093/nar/gkad435, `https://doi.org/10.1093/nar/gkad435`.

10. J. Rees-Garbutt, *et al.*, Designing minimal genomes using whole-cell models. *Nature communications* **11** (1), 836 (2020).

11. J. Rees-Garbutt, O. Chalkley, C. Grierson, L. Marucci, Minimal Genome Design Algorithms Using Whole-Cell Models. *Computational Methods in Synthetic Biology* pp. 183–198 (2021).

12. L. Marucci, *et al.*, Computer-aided whole-cell design: taking a holistic approach by integrating synthetic with systems biology. *Frontiers in Bioengineering and Biotechnology* **8**, 942 (2020).

13. S. Landon, O. Chalkley, G. Breese, C. Grierson, L. Marucci, Understanding metabolic flux behaviour in whole-cell model output. *Frontiers in molecular biosciences* p. 1114 (2021).

14. I. M. Gherman, *et al.*, Bridging the gap between mechanistic biological models and machine learning surrogates. *PLoS Computational Biology* **19** (4), e1010988 (2023).

15. L. Liang, M. Liu, C. Martin, W. Sun, A machine learning approach as a surrogate of finite element analysis–based inverse method to estimate the zero-pressure geometry of human thoracic aorta. *International journal for numerical methods in biomedical engineering* **34** (8), e3103 (2018).

16. Y. Dabiri, *et al.*, Prediction of left ventricular mechanics using machine learning. *Frontiers in physics* **7**, 117 (2019).

17. L. Cai, *et al.*, Surrogate models based on machine learning methods for parameter estimation of left ventricular myocardium. *Royal Society open science* **8** (1), 201121 (2021).

18. S. Wang, *et al.*, Massive computational acceleration by using neural networks to emulate mechanism-based biological models. *Nature communications* **10** (1), 1–9 (2019).

19. M. Renardy, T.-M. Yi, D. Xiu, C.-S. Chou, Parameter uncertainty quantification using surrogate models applied to a spatial model of yeast mating polarization. *PLoS computational biology* **14** (5), e1006181 (2018).

20. P. Di Achille, *et al.*, Gaussian process regressions for inverse problems and parameter searches in models of ventricular mechanics. *Frontiers in physiology* **9**, 1002 (2018).

21. V. Davies, *et al.*, Fast parameter inference in a biomechanical model of the left ventricle by using statistical emulation. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **68** (5), 1555–1576 (2019).

22. N. V. Queipo, *et al.*, Surrogate-based analysis and optimization. *Progress in aerospace sciences* **41** (1), 1–28 (2005).

23. Y. Iwadate, H. Honda, H. Sato, M. Hashimoto, J.-i. Kato, Oxidative stress sensitivity of engineered Escherichia coli cells with a reduced genome. *FEMS microbiology letters* **322** (1), 25–33 (2011).

24. Y. Hirokawa, *et al.*, Genetic manipulations restored the growth fitness of reduced-genome Escherichia coli. *Journal of bioscience and bioengineering* **116** (1), 52–58 (2013).

25. C. A. Hutchison III, *et al.*, Design and synthesis of a minimal bacterial genome. *Science* **351** (6280), aad6253 (2016).

26. L. Wang, C. D. Maranas, MinGenome: an in silico top-down approach for the synthesis of minimized genomes. *ACS synthetic biology* **7** (2), 462–473 (2018).

27. J. Solana, E. Garrote-Sánchez, R. Gil, DELEAT: gene essentiality prediction and deletion design for bacterial genome reduction. *BMC bioinformatics* **22**, 1–17 (2021).

28. J. Ali, R. Khan, N. Ahmad, I. Maqsood, Random forests and decision trees. *International Journal of Computer Science Issues (IJCSI)* **9** (5), 272 (2012).

29. Materials and methods are available as supplementary material.

30. I. M. Keseler, *et al.*, The EcoCyc database in 2021. *Frontiers in microbiology* **12**, 711077 (2021).

31. R. W. Hansen, J. A. Hayashi, Glycolate metabolism in Escherichia coli. *Journal of Bacteriology* **83** (3), 679–687 (1962).

32. A. Brinkkötter, A. Shakeri-Garakani, J. W. Lengeler, Two class II D-tagatose-bisphosphate aldolases from enteric bacteria. *Archives of microbiology* **177**, 410–419 (2002).

33. I. A. Rodionova, *et al.*, Comparative genomics and functional analysis of rhamnose catabolic pathways and regulons in bacteria. *Frontiers in microbiology* **4**, 74064 (2013).

34. C. Fernández, A. Ferrández, B. Minambres, E. Diaz, J. L. Garcia, Genetic characterization of the phenylacetyl-coenzyme A oxygenase from the aerobic phenylacetic acid degradation pathway of Escherichia coli. *Applied and environmental microbiology* **72** (11), 7422–7426 (2006).

35. D. Sambasivarao, J. H. Weiner, Dimethyl sulfoxide reductase of Escherichia coli: an investigation of function and assembly by use of in vivo complementation. *Journal of bacteriology* **173** (19), 5935–5943 (1991).

36. G. Unden, J. Bongaerts, Alternative respiratory pathways of Escherichia coli: energetics and transcriptional regulation in response to electron acceptors. *Biochimica et Biophysica Acta (BBA)-Bioenergetics* **1320** (3), 217–234 (1997).

37. M. Kurokawa, B.-W. Ying, Experimental challenges for reduced genomes: The cell model Escherichia coli. *Microorganisms* **8** (1), 3 (2019).

38. D. Choe, S. Cho, S. C. Kim, B.-K. Cho, Minimal genome: Worthwhile or worthless efforts toward being smaller? *Biotechnology journal* **11** (2), 199–211 (2016).

39. S. Landon, J. Rees-Garbutt, L. Marucci, C. Grierson, Genome-driven cell engineering review: in vivo and in silico metabolic and genome engineering. *Essays in biochemistry* **63** (2), 267–284 (2019).

40. V. Kolisnychenko, *et al.*, Engineering a reduced Escherichia coli genome. *Genome research* **12** (4), 640–647 (2002).

41. G. Posfai, *et al.*, Emergent properties of reduced-genome Escherichia coli. *science* **312** (5776), 1044–1046 (2006).

42. M. K. Park, *et al.*, Enhancing recombinant protein production with an Escherichia coli host strain lacking insertion sequences. *Applied microbiology and biotechnology* **98**, 6701–6713 (2014).

43. W. A. Pruett, R. L. Hester, The creation of surrogate models for fast estimation of complex model outcomes. *PloS one* **11** (6), e0156574 (2016).

44. T. K. Ho, Random decision forests, in *Proceedings of 3rd international conference on document analysis and recognition* (IEEE), vol. 1 (1995), pp. 278–282.

45. G. Biau, E. Scornet, A random forest guided tour. *Test* **25**, 197–227 (2016).

46. C. Wade, K. Glynn, *Hands-On Gradient Boosting with XGBoost and scikit-learn: Perform accessible machine learning and extreme gradient boosting with Python* (Packt Publishing Ltd) (2020).

47. L. E. Peterson, K-nearest neighbor. *Scholarpedia* **4** (2), 1883 (2009).

48. S. A. Aleksander, *et al.*, The Gene Ontology knowledgebase in 2023. *Genetics* **224** (1), iyad031 (2023).

49. M. Ashburner, *et al.*, Gene ontology: tool for the unification of biology. *Nature genetics* **25** (1), 25–29 (2000).

50. M. Verleysen, D. François, The curse of dimensionality in data mining and time series prediction, in *International work-conference on artificial neural networks* (Springer) (2005), pp. 758–770.

51. S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions. *Advances in neural information processing systems* **30** (2017).

Acknowledgments

We would like to thank the Advanced Computing Research Centre (ACRC) at the University of Bristol for access to the BlueCrystal and BluePeble supercomputers. Special thanks to the HPC and RDSF teams of the ACRC, particularly Dianaimh Greene and Callum Wright for their help with BlueCrystal and BluePebble. Furthermore, we would like to thank Oracle Cloud for allowing us to run some of the simulations used in this study on their clusters. Special thanks to Dr. Christopher Woods for helping us to set everything up and to Richard Pitts from Oracle. The authors also gratefully acknowledge the support of the EPSRC Digital Core Capability funding (EP/X034828/1) for the University of Bristol Digital Labs platform which provided high performance computing

resources, as well as the support of all the staff involved. We would also like to thank the members of the Covert lab at Stanford University who have answered all our questions regarding the installation and use of the *E. coli* WCM, especially Dr. Travis Ahn-Horst and Dr. Gwanggyu Sun.

Funding: This work was funded through the UK's Biotechnology and Biological Sciences Research Council (BB/W012235/1 to L. M. and J. R-G, BB/W013770/1 to I. M. G.) and the UK's Engineering and Physical Sciences Research Council (EP/S01876X/1 to L. M.). T. E. G. was supported by a Turing Fellowship from The Alan Turing Institute under EPSRC grant EP/N510129/1, and a Royal Society University Research Fellowship grant URF/R/221008. L. M., C. S. G. and T. E. G. were supported by BrisEngBio, a UKRI-funded Engineering Biology Research Centre grant BB/W013959/1. The funders had no role in study design, data collection and analysis, and decision to publish or preparation of the manuscript.

Author contributions: I. M. G. developed the research concept (with assistance from L. M., W. P., Z. S. A., C. S. G. and T. E. G), built and trained the ML models, adapted the genome reduction algorithm, conducted the numerical simulations, analysed the results, provided interpretation of the biological relevance and wrote the manuscript (with assistance from J. R-G., L. M., C. S. G. and T. E. G). All authors reviewed the manuscript and approved the final version.

Competing interests: There are no competing interests to declare.

Data and materials availability: The data used in this paper is stored using the storage system of the University of Bristol at https://data.bris.ac.uk/data/dataset/2wnzwzenfn4tt2iva9xxmb7ayh. The code used to generate the results presented in this study can be found at https://zenodo.org/records/13692990.

Supplementary materials

Materials and Methods

Supplementary Text

Figs. S1 to S7

Tables S1 to S3

References *(43-51)*

Supplementary Table S5-S12