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

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

[1] School of Engineering Mathematics and Technology, University of Bristol, BS8 1TW, UK
[2] School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh, EH14 4AS, UK
[3] School of Biological Sciences, University of Bristol, 24 Tyndall Avenue, Bristol BS8 1TQ, UK
[4] BrisEngBio, School of Chemistry, University of Bristol, Cantock's Close, Bristol BS8 1TS, UK
[*] These authors should be considered joint senior authors

Abstract

Whole-cell models (WCMs) are multi-scale computational models that aim to accurately 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, making their development and use challenging. To address this limitation it is possible to use simpler machine learning (ML) surrogates that can learn to approximate specific behaviours of larger and more complex models, while requiring only a fraction of the computational resources. Here, we show how a ML surrogate can be trained on WCM outputs to accurately predict whether cells divide successfully when a subset of genes are removed (knocked out). Our ML surrogate performs simulations up to 15 times faster than 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. These results demonstrate the value of adopting WCMs and ML surrogates for enabling genome-wide engineering of living cells, offering promising new routes for biologists to design cellular phenotypes, and understand how they can emerge from engineered genotypes.

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 optimized using WCMs, with the *M. genitalium* WCM being used to predict minimal genomes *in silico* [10, 11, 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 [13, 9].

The comprehensive nature of WCMs enables detailed spatial-temporal tracking of cellular processes and components. However, this detail also imparts a substantial computational burden. Simulating a single generation of the *M. genitalium* WCM can take between 5 and 12 hours using a single CPU and 8 GB of RAM [10]. In comparison, simulating a single cell division of *E. coli* using the single-cell model only takes approximately 15 minutes on a similar computer [3]. Despite simulation of the *E. coli* WCM being much faster than for the *M. genitalium* model, the increase in the number of modelled genes enlarges

the possible combinations that have to be tested when using these models to predict phenotypes from genotypes over several cell divisions (i.e. generations). 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, 16, 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 optimization [22] and sensitivity analysis [19]. ML surrogates have emulated mechanistic models based on Partial Differential Equations (PDEs) [15, 16, 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, 24, 25] and *in silico* [10, 5, 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* WCM (used here) captures the functionality of 1,219 genes and can be run for as many generations as we choose. 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 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.

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, aims to predict whether a cell will divide when a subset of its genes are removed. 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 features/states from the WCM to use to train the ML surrogate. Preliminary experiments where we simulated cell growth for four generations showed that in some cases the cells divided but also exhibited significant growth defects, especially in protein mass (Figures S.1 and S.2). This suggested that more than four generations should be simulated. The analysis presented in Supplementary Section S.1 showed that 96% of the cells that failed to divide in eight generations, actually did so in the first six generations (Figure S.3) 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. Due to the high dimensionality of the WCM (over 10,000 equations and 19,000 parameters), we began by looking only at 22 different macromolecular masses as variables that could capture a summary of the cell's state. We compared the accuracy of ML surrogates using the values of these variables (i.e. features) either at the beginning of the simulations ($t = 0$, generation 1, Table S.1) or at the end of generation 1, just before the first cell divides (Table S.3). We saw an improvement of approximately 16% in the F1-score of the training set and 18% for the test set, 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 Supplementary Section S.1. An overview of the simulations run to train the ML surrogates is presented in Table S.2.

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 S.4 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

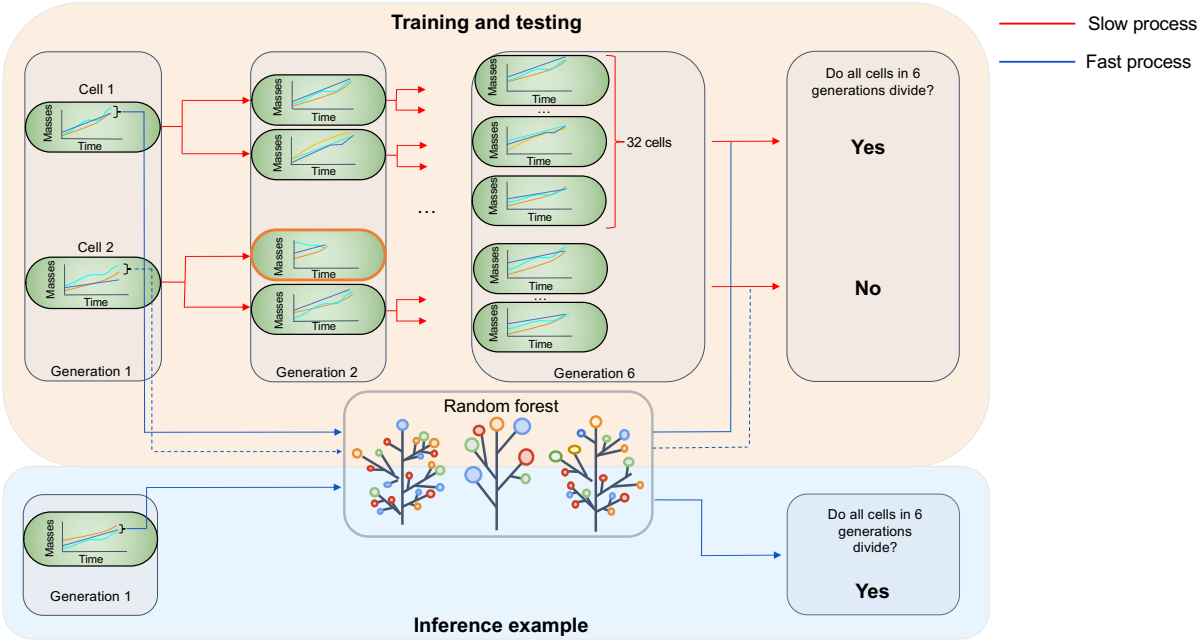

Figure 1: A random forest model is trained to emulate the behaviour of the *E. coli* WCM. To obtain the training data for the ML surrogate, the WCM is run for six generations. Simulating the WCM for six generations is computationally expensive and time consuming (red arrows). Therefore, we use the macromolecular 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).

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 4). The 12 key macromolecular masses 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 simplicity, explainability and performance for classification tasks in diverse contexts [28]. 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 S.3). 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 2.A). 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.

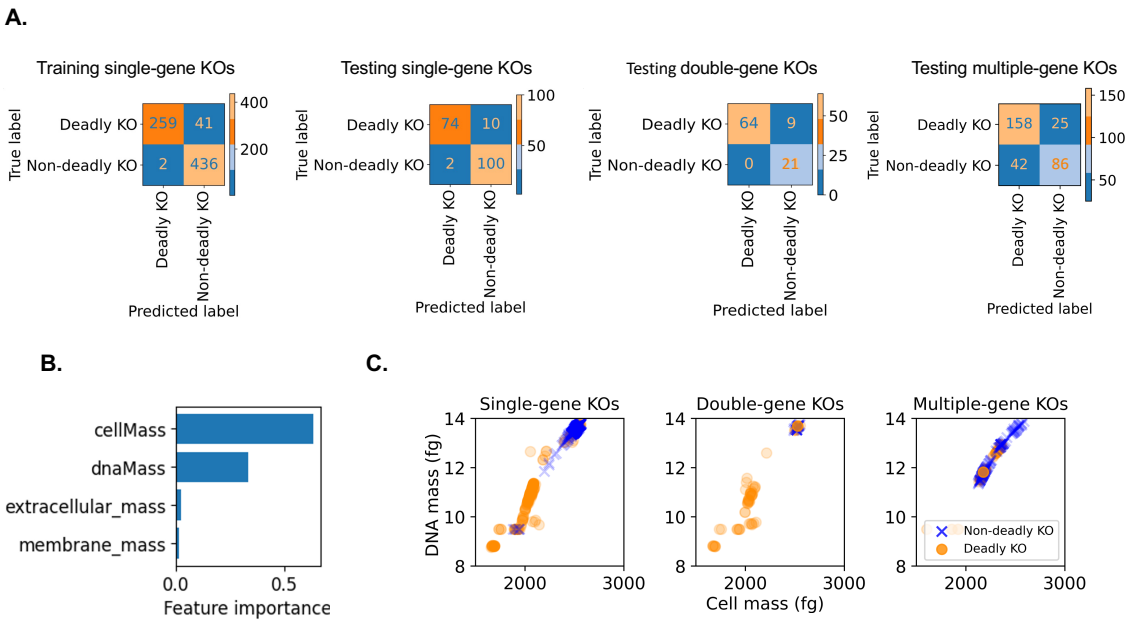

Figure 2: Results of the initial ML surrogate trained on single gene knock out (KO) data. (A) The confusion matrices for the training of the ML surrogate on single gene knock outs, and testing on single, double and multiple gene knock outs. The true label is based on the output of the WCM and the predicted 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.

Our random forest classifier was trained, validated and tested repeatedly on these selected 12 features and the results showed that only four of them 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 that the most important feature is the cell mass, followed by the DNA mass, the extracellular mass and the membrane mass. More details about the impact of the features on the classifier and the misclassified data can be found in Supplementary Section S.2.

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 S.3). 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 S.3), 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 5). 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 S.3, 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 Supplementary Section S.3.

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

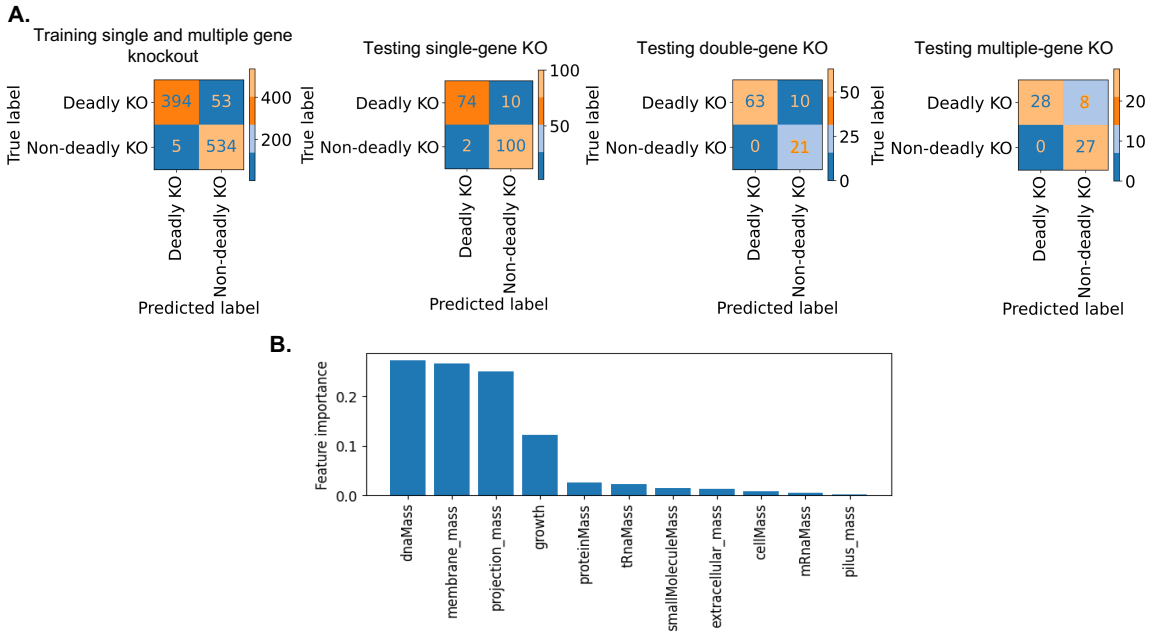

Figure 3: Results of the updated ML surrogate trained on single and multiple gene knock outs data. (A) The confusion matrices for the training and testing of the ML surrogate on single, double and multiple gene knock outs. The true label is based on the output of the WCM and the predicted label is represented 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.

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 6). 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 Methods 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 Supplementary Section S.3. 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

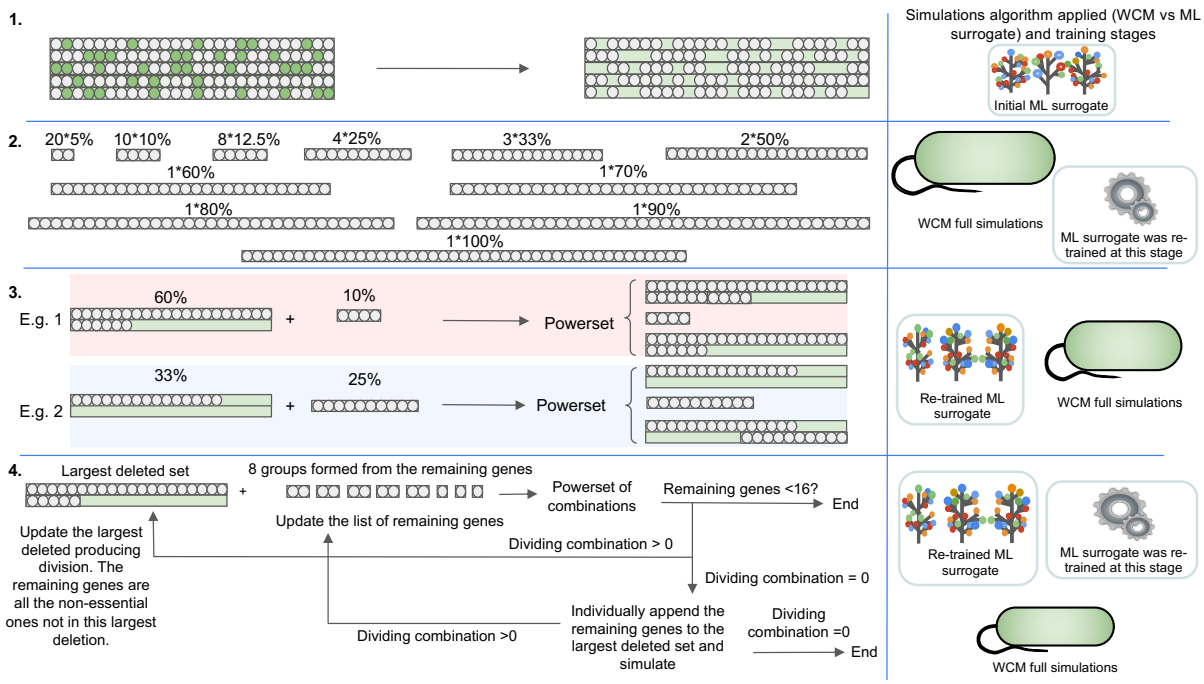

Figure 4: Summary of the framework used to generate the minimal genome. This is an adaptation of the Minesweeper framework [10], using the *E. coli* WCM [3] and the ML surrogate. The process started by using the ML surrogate trained on a subset of single gene knock out data to predict the essentiality 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.

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 (Supplementary Section S.3). 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 7). 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 6 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. In total, all simulations described above took approximately 4 months to complete.

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 the 50 repetitions of EMine-737 to those of 50 repetitions of wildtype cells (Figure 5). We found that the average cell mass of the EMine-737 cells is 1286.96 fg compared to 1669.52 fg for wildtype cells. The reduced genome cells also grow more slowly (0.13 average growth rate of EMine-737 compared to 0.24 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 are often higher than those of wildtype cells. However,

when looking at the statistics across the time series, the average protein mass over six generations is 215.64 fg for EMine-737 and 230 fg for wildtype cells, while the standard deviation of the protein mass over six generations is 89.14 fg for EMine-737 and 54.18 fg for wildtype cells.

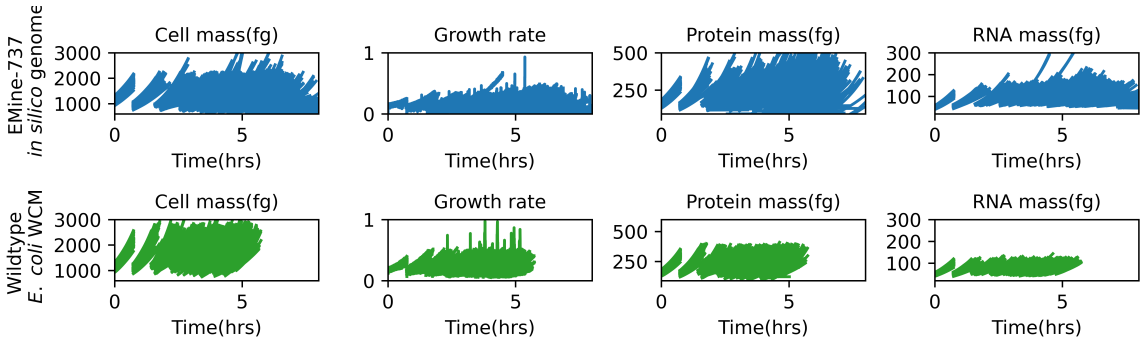

Figure 5: Data from fifty *in silico* repetitions of EMine-737 (first row) and wildtype *E. coli* WCM (second row), run for 6 generations, with each cell division producing two daughters.

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, 50repetitions).

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 8.

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 [29, 30], 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 [29, 31, 32]. 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 [29].

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 6 shows that the two amino acids are still being produced in EMine-737, although less robustly compared to the wildtype cells.

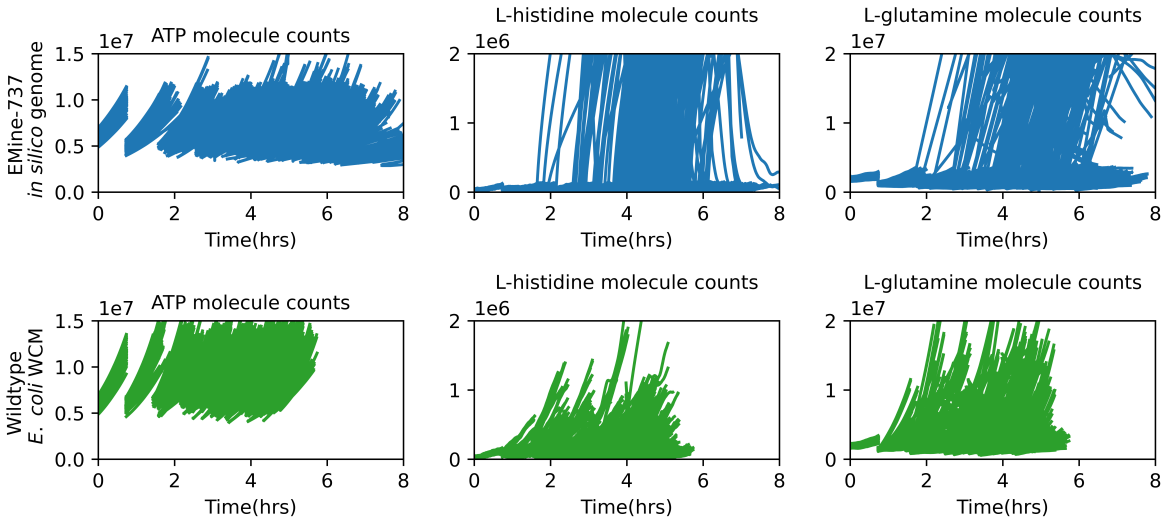

Figure 6: A comparison between the molecule counts of ATP, L-histidine and L-glutamine between wildtype cells and EMine-737 cells. Each genome was simulated 50 times with each cell division producing two daughter cells.

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 [33], but these do not seem to significantly affect the ability of the cell to grow or to produce ATP (Figures 5 and 6). *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 [29]. 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 [29].

The dimethyl sulfoxide metabolic process is also removed from EMine-737, which represents a terminal electron acceptor in anaerobic conditions [34], 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 [35]. 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 [29]. 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) [36, 37, 38], but only three studies tested the reduced MG1655 genomes in minimal media [39, 40, 41]. 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 [39]. Then Posfai *et al.* removed additional regions from the genome to obtain the MDS43 strain

(15.3% of the genome was removed) [40]. Later, Park *et al.* removed another 13 genomic regions to obtain the MS56 strain with 23% of the original genome deleted [41]. Since MS56 contained the gene deletions from MDS12 and MDS43, we only compare EMine-737 to MS56.

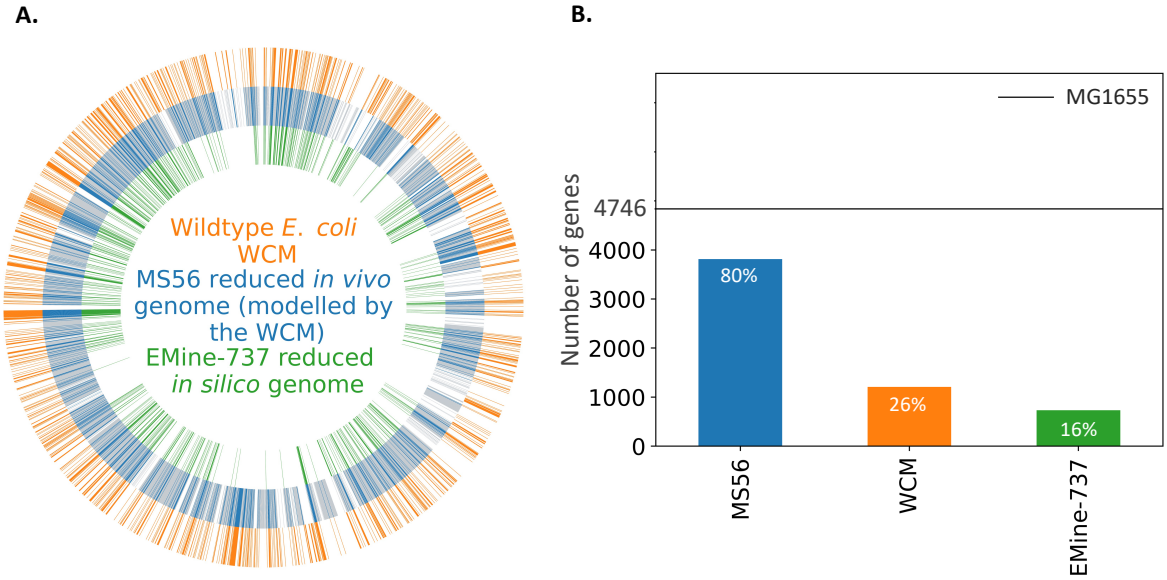

Figure 7: (A) Genes included in the *E. coli* WCM (in orange), the genes included in MS56 that are also modelled in the WCM [41] (in blue), the genes present in MS56, but not modelled by the WCM (in grey) and the genes included in EMine-737 (in green). (B) Number of genes included in each of the genomes. The percentages are calculated relative to the size of the MG1655 genome.

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 7A 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 7B 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 26% 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 [39, 40, 41]. A list of the genes used for this analysis is shown is Supplementary Table 9.

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 greatly accelerate 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. Here, 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 in six generations was reduced from 63 using only the *E. coli* WCM to one using the ML surrogate (assuming all daughter cells in six generations divide).

The performance of our surrogate, 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 10), 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. 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), therefore we do not expect that when the genome proposed here is tested experimentally, the cell will remain viable. However, we believe that this data could guide future experimental avenues for obtaining reduced *E. coli* genomes.

The increase in protein and RNA mass of the reduced *in silico* genome is another avenue worth investigating in future experimental and computational studies (Figure 5). Similarly, to what we observe here, other studies have also shown that genome reduction in *S. lividans* increased the secretion of proteins [42]. One of the main reasons for building reduced genomes is to have organisms that are highly efficient at utilizing the resources available [37], for example, more efficient synthesis of proteins and RNAs could yield heightened gene expression levels, with more proteins produced per given unit of cellular resources.

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.

Methods

The *E. coli* whole-cell model simulations

In this study, we are using the single-cell version of the *E. coli* whole-cell model, which can be thought of as a large system of Ordinary Differential Equations (ODEs) where the state variables of the ODEs correspond to different cellular states, and the differential equations correspond to cellular processes [3]. Simulating a cell is similar to numerical integration of the ODE system. The resources stored in cellular states must be shared with all cellular processes to ensure mass conservation [3]. In addition, the model has two main stochastic components: initializing RNA/protein synthesis and degrading RNA/protein. This is implemented by using a multinomial and Poisson distribution, respectively. The model is implemented in Python, using Cython and is available at https://github.com/CovertLab/wcEcoli. A version of the model available in March 2022 was used to run the simulations from in this study.

The model simulates the log phase growth of each cell and it is possible to do so for several generations. In this study, we used the default M9 minimal media supplemented with 0.4% glucose. For each cell simulation, the model outputs 219,207 time series after approximately 15 minutes of running time on a CPU. Given the large number of experiments needed for this study, the simulations were run in parallel on the University of Bristol's supercomputers BlueCrystal phase 4 and BluePebble, on a cluster in the cloud platform called Digital Labs, and on Oracle Cloud. Furthermore, the *E. coli* WCM uses Fireworks, a workflow management tool, to automate the execution of different jobs.

For the scope of this project, simulations were run for six generations, unless during this process a cell was not dividing in which case the simulations were sometimes stopped after four generations. Unless otherwise specified in the results section or in the supplementary material, the simulations were run for one repetition.

ML surrogate

A machine learning surrogate model is generally used to emulate the behaviour of a mechanistic model, with the added advantage that once trained, it can be used to make predictions several orders of magnitude faster than the original mechanistic model. The process for training such a machine learning model involves running the mechanistic model several times in order to generate the training and test sets. After that, the ML model is trained using this data, and once the accuracy is satisfactory, it can be used to replace the original mechanistic model in future simulations [14]. Some common techniques

used to create surrogate models include regression models [19, 43], Gaussian processes [21], support vector machines [43], decision tree-based models [16, 17] and neural networks [17, 15].

Typically, a machine learning surrogate model is trained on the parameters and/or initial conditions of the mechanistic model. However, due to the high number of parameters and variables of the WCM, this was not possible in this study due to the curse of dimensionality. Therefore, the ML surrogate is trained on the final state (just before division) of the first cell simulated for each repetition, and it predicts whether all the cells in the future five generations will divide. The advantage of this approach is that once it is trained, the mechanistic WCM has to be run for one generation, rather than six, and the data from this first generation is used to predict the fitness of all the cells in the future five generations.

In this paper we assess the accuracy of the ML surrogate using the F1-score metric, which combines the precision and recall scores of the models. We chose this metric due to its benefits when dealing with unbalanced data. A higher F1-score represents a better model.

Random forest

Random Forest is a popular supervised machine learning algorithm used for classification and regression problems [44]. It trains a large number of individual decision trees that operate as an ensemble. Each of these trees is trained on a random subset of the training data and features. The final predicted class is the one that was output by the majority of the decision trees. This ensemble-based approach reduces overfitting and aids interpretability since it becomes possible to understand which features contributed to the prediction the most [45], by averaging the gini impurity of the trees. Furthermore, compared to more complex deep learning models, random forest can be trained very quickly and it often has comparable accuracy. The ML surrogate model based on Random Forest was built in Python using the scikit learn package.

Minesweeper

Minesweeper is a four-stage algorithm inspired by laboratory fragment-cassette-fraction genome engineering [25], which is similar to the divide-and-conquer paradigm. It was originally published in [10] to predict minimal genomes using the *M. genitalium* WCM, and it is freely available on https://github.com/GriersonMarucciLab/Minesweeper. Minesweeper starts by running all the single gene knock outs. In the original model [10], during the second stage, all the predicted non-essential genes are split into 26 segments of sizes 100%, 90%, 80%, 70%, 60%, 50%, 33%, 25% and 12.5%. Since the *E.*

coli WCM includes three times more genes, the 12.5% deletion segments were too big. To address this, we added 30 additional segments of 10% and 5%, respectively. During the third stage of the algorithm, the largest three deletions that produced dividing cells in six generations during stage two, are combined with smaller, dividing, non-overlapping segments. Powersets of unique combinations of these segments are generated and simulated. The largest segment that when deleted produced dividing cells from each of the three groups moves forward to stage four, where it is combined with the remaining non-essential genes that are split into eight groups. The powersets of these segments are simulated and if none of the reduced cells divide, the remaining genes are appended individually to the largest deletion segment. The genes that are individually added and do not produce a dividing cell are not considered for further combinations. This stage is cyclical and it stops when there are fewer than 16 genes (in [10] this was eight) to be added or if the deletion of none of the combinations obtained after adding individual genes produced dividing cells. The threshold for stopping stage four was changed from eight to 16 genes to reduce the number of repetitions of this cycle, again due to the larger *E. coli* genome compared to *M. genitalium*. Also, as noted in the conclusion, the algorithm was stopped before this cycle ended. The main reason for this was the high number of simulations that were already run, time constraints and the fact that a significantly reduced genome was already obtained.

Gene ontology analysis

Gene Ontology (GO) is a major bioinformatics initiative that aims to unify the representation of gene and gene product attributes across all species. It provides an ontology of defined terms that represent gene product properties. The ontology covers three main domains: the Cellular Component domain, the Molecular function domain and the Biological Process domain. In this study, we used the databases and annotations from the Gene Ontology Consortium to perform our analysis [46, 47]. The multiple false discovery rate used in this study is 0.05. The.obo file used to generate these results corresponds to the version available on 11th of June 2023.

During our analysis, we noticed that genes included in some ontology categories differ slightly between EcoCyc [29] and the Gene Ontology Consortium database. One such example is the GO:0015871, which on EcoCyc includes the *betT* gene and in the database we use in this study it does not. We believe this emphasises the need for more curated data sources in synthetic biology.

Data and code availability

The data used to train the ML surrogates and to generate the figures presented in the manuscript are available from our group's Research Data Repository at the University of Bristol (link will be added with the final submission). The full simulation outputs can be obtained by contacting the authors. The code used in this study is available at 10.5281/zenodo.11179831, GitHub.

Acknowledgements

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 Gwanggyu Sun. 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.

References

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

[2] Jonathan R Karr, Jayodita C Sanghvi, Derek N Macklin, Miriam V Gutschow, Jared M Jacobs, Benjamin Bolival Jr, Nacyra Assad-Garcia, John I Glass, and Markus W Covert. A whole-cell computational model predicts phenotype from genotype. *Cell*, 150(2):389–401, 2012.

[3] Derek N Macklin, Travis A Ahn-Horst, Heejo Choi, Nicholas A Ruggero, Javier Carrera, John C Mason, Gwanggyu Sun, Eran Agmon, Mialy M DeFelice, Inbal Maayan, et al. Simultaneous cross-evaluation of heterogeneous e. coli datasets via mechanistic simulation. *Science*, 369(6502):eaav3751, 2020.

[4] Eran Agmon, Ryan K Spangler, Christopher J Skalnik, William Poole, Shayn M Peirce, Jerry H Morrison, and Markus W Covert. Vivarium: an interface and engine for integrative multiscale modeling in computational biology. *Bioinformatics*, 38(7):1972–1979, 2022.

[5] Zane R Thornburg, David M Bianchi, Troy A Brier, Benjamin R Gilbert, Tyler M Earnest, Marcelo CR Melo, Nataliya Safronova, James P Sáenz, András T Cook, Kim S Wise, et al. Fundamental behaviors emerge from simulations of a living minimal cell. *Cell*, 185(2):345–360, 2022.

[6] Travis A Ahn-Horst, Luis Santiago Mille, Gwanggyu Sun, Jerry H Morrison, and Markus 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] Christopher J Skalnik, Sean Y Cheah, Mica Y Yang, Mattheus B Wolff, Ryan K Spangler, Lee Talman, Jerry H Morrison, Shayn M Peirce, Eran Agmon, and Markus W Covert. Whole-cell modeling of e. coli colonies enables quantification of single-cell heterogeneity in antibiotic responses. *PLOS Computational Biology*, 19(6):e1011232, 2023.

[8] Jayodita C Sanghvi, Sergi Regot, Silvia Carrasco, Jonathan R Karr, Miriam V Gutschow, Benjamin Bolival Jr, and Markus W Covert. Accelerated discovery via a whole-cell model. *Nature methods*, 10(12):1192–1195, 2013.

[9] Heejo Choi and Markus 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*, 05 2023. gkad435.

[10] Joshua Rees-Garbutt, Oliver Chalkley, Sophie Landon, Oliver Purcell, Lucia Marucci, and Claire Grierson. Designing minimal genomes using whole-cell models. *Nature communications*, 11(1):836, 2020.

[11] Joshua Rees-Garbutt, Oliver Chalkley, Claire Grierson, and Lucia Marucci. Minimal genome design algorithms using whole-cell models. *Computational Methods in Synthetic Biology*, pages 183–198, 2021.

[12] Lucia Marucci, Matteo Barberis, Jonathan Karr, Oliver Ray, Paul R Race, Miguel de Souza Andrade, Claire Grierson, Stefan Andreas Hoffmann, Sophie Landon, Elibio Rech, 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] Sophie Landon, Oliver Chalkley, Gus Breese, Claire Grierson, and Lucia Marucci. Understanding metabolic flux behaviour in whole-cell model output. *Frontiers in molecular biosciences*, page 1114, 2021.

[14] Ioana M Gherman, Lucia Marucci, Zahraa Abdallah, Claire Grierson, Thomas Gorochowski, and Wei Pang. Bridging the gap between mechanistic biological models and machine learning surrogates. 2022.

[15] Liang Liang, Minliang Liu, Caitlin Martin, and Wei 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] Yaghoub Dabiri, Alex Van der Velden, Kevin L Sack, Jenny S Choy, Ghassan S Kassab, and Julius M Guccione. Prediction of left ventricular mechanics using machine learning. *Frontiers in physics*, 7:117, 2019.

[17] Li Cai, Lei Ren, Yongheng Wang, Wenxian Xie, Guangyu Zhu, and Hao Gao. Surrogate models based on machine learning methods for parameter estimation of left ventricular myocardium. *Royal Society open science*, 8(1):201121, 2021.

[18] Shangying Wang, Kai Fan, Nan Luo, Yangxiaolu Cao, Feilun Wu, Carolyn Zhang, Katherine A Heller, and Lingchong You. Massive computational acceleration by using neural networks to emulate mechanism-based biological models. *Nature communications*, 10(1):1–9, 2019.

[19] Marissa Renardy, Tau-Mu Yi, Dongbin Xiu, and Ching-Shan Chou. Parameter uncertainty quantification using surrogate models applied to a spatial model of yeast mating polarization. *PLoS computational biology*, 14(5):e1006181, 2018.

[20] Paolo Di Achille, Ahmed Harouni, Svyatoslav Khamzin, Olga Solovyova, John J Rice, and Viatcheslav Gurev. Gaussian process regressions for inverse problems and parameter searches in models of ventricular mechanics. *Frontiers in physiology*, 9:1002, 2018.

[21] Vinny Davies, Umberto Noè, Alan Lazarus, Hao Gao, Benn Macdonald, Colin Berry, Xiaoyu Luo, and Dirk Husmeier. 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] Nestor V Queipo, Raphael T Haftka, Wei Shyy, Tushar Goel, Rajkumar Vaidyanathan, and P Kevin Tucker. Surrogate-based analysis and optimization. *Progress in aerospace sciences*, 41(1):1–28, 2005.

[23] Yumi Iwadate, Hirofumi Honda, Haruhiko Sato, Masayuki Hashimoto, and Jun-ichi Kato. Oxidative stress sensitivity of engineered escherichia coli cells with a reduced genome. *FEMS microbiology letters*, 322(1):25–33, 2011.

[24] Yasutaka Hirokawa, Hiroaki Kawano, Kimie Tanaka-Masuda, Noriko Nakamura, Akira Nakagawa, Mikito Ito, Hideo Mori, Taku Oshima, and Naotake Ogasawara. Genetic manipulations restored the growth fitness of reduced-genome escherichia coli. *Journal of bioscience and bioengineering*, 116(1):52–58, 2013.

[25] Clyde A Hutchison III, Ray-Yuan Chuang, Vladimir N Noskov, Nacyra Assad-Garcia, Thomas J Deerinck, Mark H Ellisman, John Gill, Krishna Kannan, Bogumil J Karas, Li Ma, et al. Design and synthesis of a minimal bacterial genome. *Science*, 351(6280):aad6253, 2016.

[26] Lin Wang and Costas D Maranas. Mingenome: an in silico top-down approach for the synthesis of minimized genomes. *ACS synthetic biology*, 7(2):462–473, 2018.

[27] Jimena Solana, Emilio Garrote-Sánchez, and Rosario Gil. Deleat: gene essentiality prediction and deletion design for bacterial genome reduction. *BMC bioinformatics*, 22:1–17, 2021.

[28] Jehad Ali, Rehanullah Khan, Nasir Ahmad, and Imran Maqsood. Random forests and decision trees. *International Journal of Computer Science Issues (IJCSI)*, 9(5):272, 2012.

[29] Ingrid M Keseler, Socorro Gama-Castro, Amanda Mackie, Richard Billington, César Bonavides-Martínez, Ron Caspi, Anamika Kothari, Markus Krummenacker, Peter E Midford, Luis Muñiz-Rascado, et al. The ecocyc database in 2021. *Frontiers in microbiology*, 12:711077, 2021.

[30] Robert W Hansen and James A Hayashi. Glycolate metabolism in escherichia coli. *Journal of Bacteriology*, 83(3):679–687, 1962.

[31] Andreas Brinkkötter, Ansiah Shakeri-Garakani, and Joseph W Lengeler. Two class ii d-tagatose-bisphosphate aldolases from enteric bacteria. *Archives of microbiology*, 177:410–419, 2002.

[32] Irina A Rodionova, Xiaoqing Li, Vera Thiel, Sergey Stolyar, James K Fredrickson, Donald A Bryant, Andrei L Osterman, Aaron A Best, and Dmitry A Rodionov. Comparative genomics and functional analysis of rhamnose catabolic pathways and regulons in bacteria. *Frontiers in microbiology*, 4:74064, 2013.

[33] Cristina Fernández, Abel Ferrández, Baltasar Minambres, Eduardo Díaz, and Jose 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.

[34] D Sambasivarao and JOEL 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.

[35] Gottfried Unden and Johannes 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.

[36] Masaomi Kurokawa and Bei-Wen Ying. Experimental challenges for reduced genomes: The cell model escherichia coli. *Microorganisms*, 8(1):3, 2019.

[37] Donghui Choe, Suhyung Cho, Sun Chang Kim, and Byung-Kwan Cho. Minimal genome: Worthwhile or worthless efforts toward being smaller? *Biotechnology journal*, 11(2):199–211, 2016.

[38] Sophie Landon, Joshua Rees-Garbutt, Lucia Marucci, and Claire Grierson. Genome-driven cell engineering review: in vivo and in silico metabolic and genome engineering. *Essays in biochemistry*, 63(2):267–284, 2019.

[39] Vitaliy Kolisnychenko, Guy Plunkett, Christopher D Herring, Tamás Fehér, János Pósfai, Frederick R Blattner, and György Pósfai. Engineering a reduced escherichia coli genome. *Genome research*, 12(4):640–647, 2002.

[40] Gyorgy Posfai, Guy Plunkett III, Tamás Fehér, David Frisch, Gunther M Keil, Kinga Umenhoffer, Vitaliy Kolisnychenko, Buffy Stahl, Shamik S Sharma, Monika De Arruda, et al. Emergent properties of reduced-genome escherichia coli. *science*, 312(5776):1044–1046, 2006.

[41] Myung Keun Park, Sang Hee Lee, Kyung Seok Yang, Suk-Chae Jung, Jun Hyoung Lee, and Sun Chang Kim. Enhancing recombinant protein production with an escherichia coli host strain lacking insertion sequences. *Applied microbiology and biotechnology*, 98:6701–6713, 2014.

[42] Mohamed Belal Hamed, Tobias Busche, Kenneth Simoens, Sebastien Carpentier, Jan Kormanec, Lieve Van Mellaert, Jozef Anné, Joern Kalinowski, Kristel Bernaerts, Spyridoula Karamanou, et al. Enhanced protein secretion in reduced genome strains of streptomyces lividans. *Microbial Cell Factories*, 23(1):13, 2024.

[43] W Andrew Pruett and Robert L Hester. The creation of surrogate models for fast estimation of complex model outcomes. *PloS one*, 11(6):e0156574, 2016.

[44] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

[45] Gérard Biau and Erwan Scornet. A random forest guided tour. *Test*, 25:197–227, 2016.

[46] Suzi A Aleksander, James Balhoff, Seth Carbon, J Michael Cherry, Harold J Drabkin, Dustin Ebert, Marc Feuermann, Pascale Gaudet, Nomi L Harris, et al. The gene ontology knowledgebase in 2023. *Genetics*, 224(1):iyad031, 2023.

[47] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.