Designing minimal *E. coli* genomes using variational autoencoders

Anastasiia Shcherbakova[1], Daniel Buchan[2], and Chris P. Barnes[3]

[1]Research Software Engineering Group, Institute of Cancer Research, London, UK
[2]Department of Computer Science, University College London, London, UK
[3]Department of Cell and Developmental Biology, University College London, London, UK

Abstract

Designing minimal bacterial genomes remains a key challenge in synthetic biology. There is currently a lack of efficient tools for the rapid generation of streamlined bacterial genomes, limiting research in this area. Here, using a pangenome dataset for *Escherichia coli*, we explore how generation of minimal genomes can be achieved using variational autoencoders. Our results show that variational autoencoders can successfully create minimised genomes retaining the essential genes identified in the literature. This study proposes a rapid, machine learning-based approach for bacterial sequence generation, that could accelerate the genomic design process.

Keywords

engineering biology; synthetic biology; bacterial genomes; minimal genomes; machine learning; variational autoencoder

1 Introduction

Research on reduced genomes is important for understanding biological principles, revealing the minimal set of genes essential for cell survival and for industrial purposes [1]. A number of interesting outcomes have arisen from the creation of bacteria with reduced genomes including increased growth rate and biomass production [2, 3, 4, 5], increased gene product output [6, 7, 8] and enhanced cellular stability[9]. In addition to the increase in growth rate and decrease in byproduct production, Mogger-Reischer *et. al.* have also reported that cells engineered with only essential genes show faster evolution in terms of relative fitness compared to wild-type *Mycoplasma mycoides* cells [10]. This finding may suggest that genome reduction may also enhance bacteria's adaptability and evolutionary potential, further supporting its utilisation in biotechnological and research settings.

There are two primary approaches to creating new minimal genomes: The bottom-up approach involves *de novo* synthesis of a new artificial genome allowing for the complete redesign of bacterial genomes from scratch [11]. The top-down approach involves the reduction of existing wild-type genomes by deleting non-essential genetic elements. Despite the bottom-up approach being a success, the bacterial design principles remain unknown, posing challenges for completely new assemblies.

Several top-down reduced genome strains have been developed from the *Escherichia coli* strains K-12 W3110 and K-12 MG1655, each tailored for specific biotechnological applications [12]. One such strain, the Minimal Genome Factory (MGF-01), was designed to contain only the essential genes necessary for fermentative production. MGF-01 lacks 22% of the original K-12 W3110 genome while remaining comparable to the wild-type strain in many aspects [13]. Notably, MGF-01 accumulated higher L-threonine levels than the wild-type strain, supporting the hypothesis that genome reduction can lead to increased productivity [14]. Another key development is the Multiple Deletion Series (MDS), designed to focus on genetic stability and metabolic efficiency to optimise the insertion of foreign genes for industrial purposes [15]. The MDS series was derived from the MG1655 strain by eliminating insertion sequences and reducing additional genetic components to create strains such as MDS12, MDS41, MDS43, and MDS69 [12]. Of these, MDS41 and MDS42 exhibited no significant

differences in growth rates compared to the wild-type strain [16], although MDS69 showed a nearly 20% reduction in growth rate relative to MG1655 [17]. Despite its slower growth, MDS42 demonstrated a significant increase - 83% more L-threonine than the engineered MG1655, highlighting that genome reduction can enhance the production of specific compounds even when growth is affected [8]. The smallest *E. coli* genome reported so far is Δ33a which has a size of 2.83Mb[18]

Increasingly, computational modelling and machine learning techniques have been used for minimal genome design [19, 20, 21]. Recently variational autoencoders (VAEs), generative models combining deep neural networks and probabilistic modelling[22], were used for capturing genomic variation. These were used to simulate human genomes [23, 24] and for the design of minimal bacterial genomes [25]. In the proof-of-concept study by Dudek and Precup [25], the authors used high-quality bacterial genomes from the KEGG database, encoding each genome sequence into a binary representation of the presence or absence of genes. They corrupted the original genomes to train the model and then used the VAE to reconstruct the uncorrupted versions, validated through a genome viability pipeline by comparing the KEGG Orthology (KO) codes of genes with the KEGG database to identify missing essential proteins. While this approach showed promise, the study faced limitations due to the small dataset size. They also did not focus on specific species of bacteria, resulting in genomes that mix features of bacteria from across the bacterial tree of life.

In this work we address these limitations by focussing on *E. coli* and use a collection of high quality genomes comprising a large-scale pangenome [26]. We use the presence-absence gene matrix to train a VAE and explore different loss functions to capture variation such that the resultant model produces reduced genomes. Finally, we develop a greedy algorithm to move from simulated binary genomes to actual reduced genomes. We show that our generated genomes have a size comparable and smaller that the state of the art in experimental minimal *E. coli* genomes. This approach also provides a means for a a systematic experimental programme to achieve reduced and minimal *E. coli* genomes.

2 Results

2.1 Data Properties

The data set comprises a curated a collection of *E. coli* and *Shigella* genomes [26]. We used as input the presence-absence binary matrix. The matrix consists of rows representing the gene names (55,039 total), columns representing the sample names (7,511 total) and one column representing the sample phylogroup. The data was pre-processed by removing the samples with no metadata provided, reducing the samples from 7,511 to 5,953. Figure 1a shows the distribution of genome size in terms of gene number. The maximum and minimum genome sizes are approximately 3800 and 5,300 genes respectively.

In order to explore individual genes across the dataset, we examined the gene count frequency, which was calculated by summing the presence of each gene across all genomes (Figure 1b). As expected, there is a collection of genes that are present across most of the genomes, which indicates the core genome. There were also many genes that occurred with very low frequency, indicating heterogeneous accessory genomes across isolates. Figure 1c shows this dependence more explicitly and as we can see, around 15,000 genes are present in at least 50 genomes. Although, previous analyses have removed genes with low frequencies [25], we decided to keep all the genes, even the low frequency ones, for training the models.

We also checked the abundance of essential genes reported by Goodall *et. al.* [27] over all genomes. This work reports that there are 358 essential genes in the *E. coli* genome. In the Horesh *et. al.* dataset, some genes were split into multiple gene fragments. We aggregated these fragments into a single representation for each gene and the dataset contained only 328 essential genes, with 30 not present in the data set. The essential gene abundance distribution is given in Figure 1d. Most genomes contain a high number of essential genes; however, interestingly, a much smaller portion does not have all the essential genes, with the minimal number being 284. This possibly reflects both the ambiguous concept of essential genes, which depends on the environment [27], and also the fact that many of these strains are members of a complex mammalian gut microbiome.

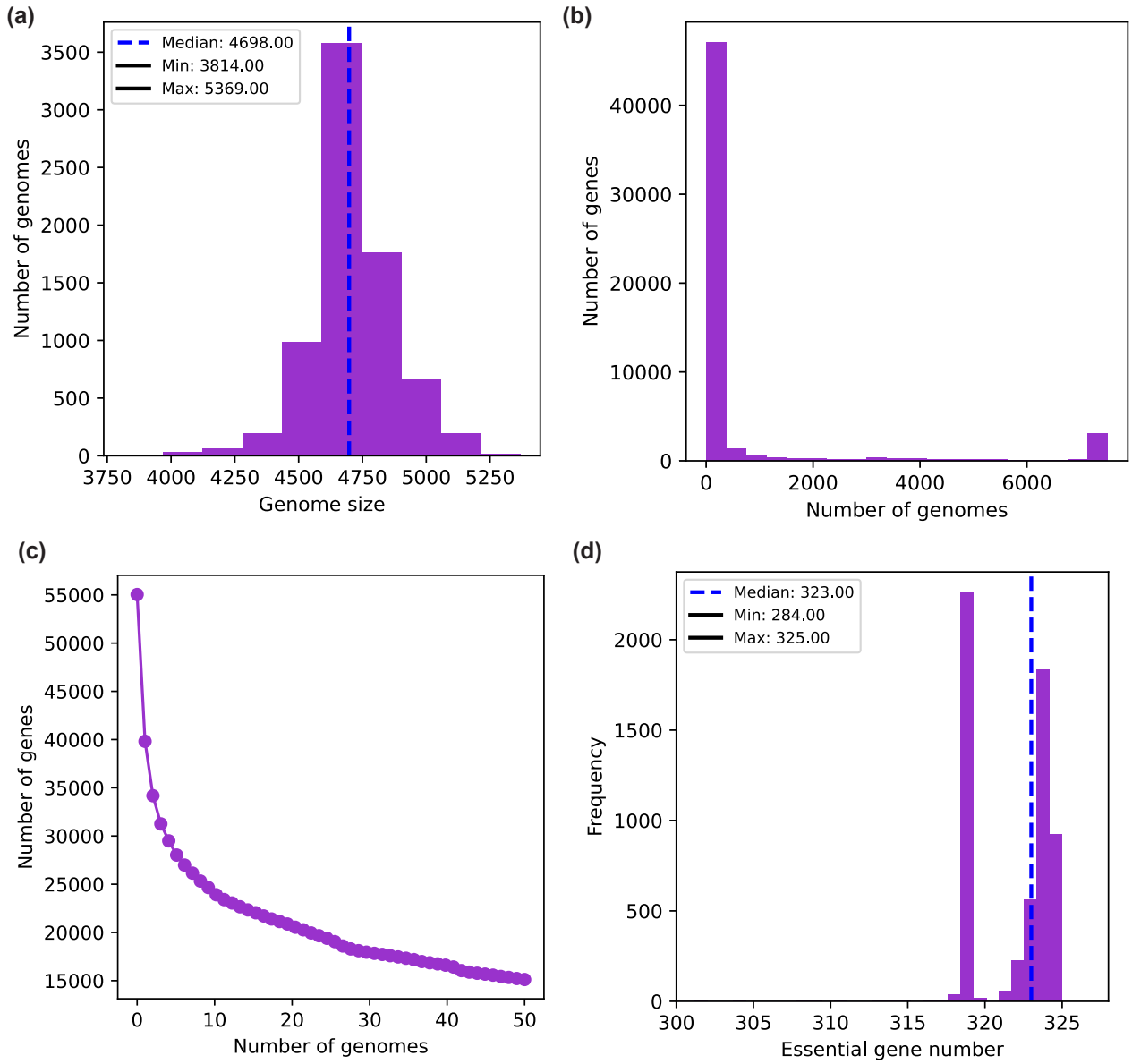

Figure 1: Genome properties in the data. (a) Distribution of genome size. (b, c) Gene frequency (d) Number of essential genes.

To understand more closely the diversity of the dataset we performed PCA. Figure 2a shows the first two principal components of the data, coloured by phylogroup. PC1 greatly separating phylogroup E and the rest of the phylogroups. PC2 separates the B1 group from the E group. As described by Horesh *et. al.* [26], all phylogroup E lineages contain *Enterohemorrhagic Escherichia coli (EHEC)* isolates. The dataset contains around 50 distinct lineages and Figure 2b shows their genetic relationship. The tree structure could indicate imbalance, as indicated by certain branches being significantly longer than others. This could suggest unequal rates of evolution or diversification among the lineages, where some lineages have undergone rapid changes while others remain more conserved.

To further investigate the data imbalance, we generated a phylogenetic tree from a sample of 1,000 genomes (Figure 2c). Ideally, for training a generative model we would have a well balanced tree, providing uniform genomic variation. The high Colless index (23,804) and Sackin index (29,872) reveal unequal lineage diversification. The correlation between the observed imbalance and asymmetry in both the lineage tree from Horesh *et. al.* and our phylogenetic tree reinforces some level of unequal lineage diversification and relatively low genetic divergence among the *E. coli* genomes.

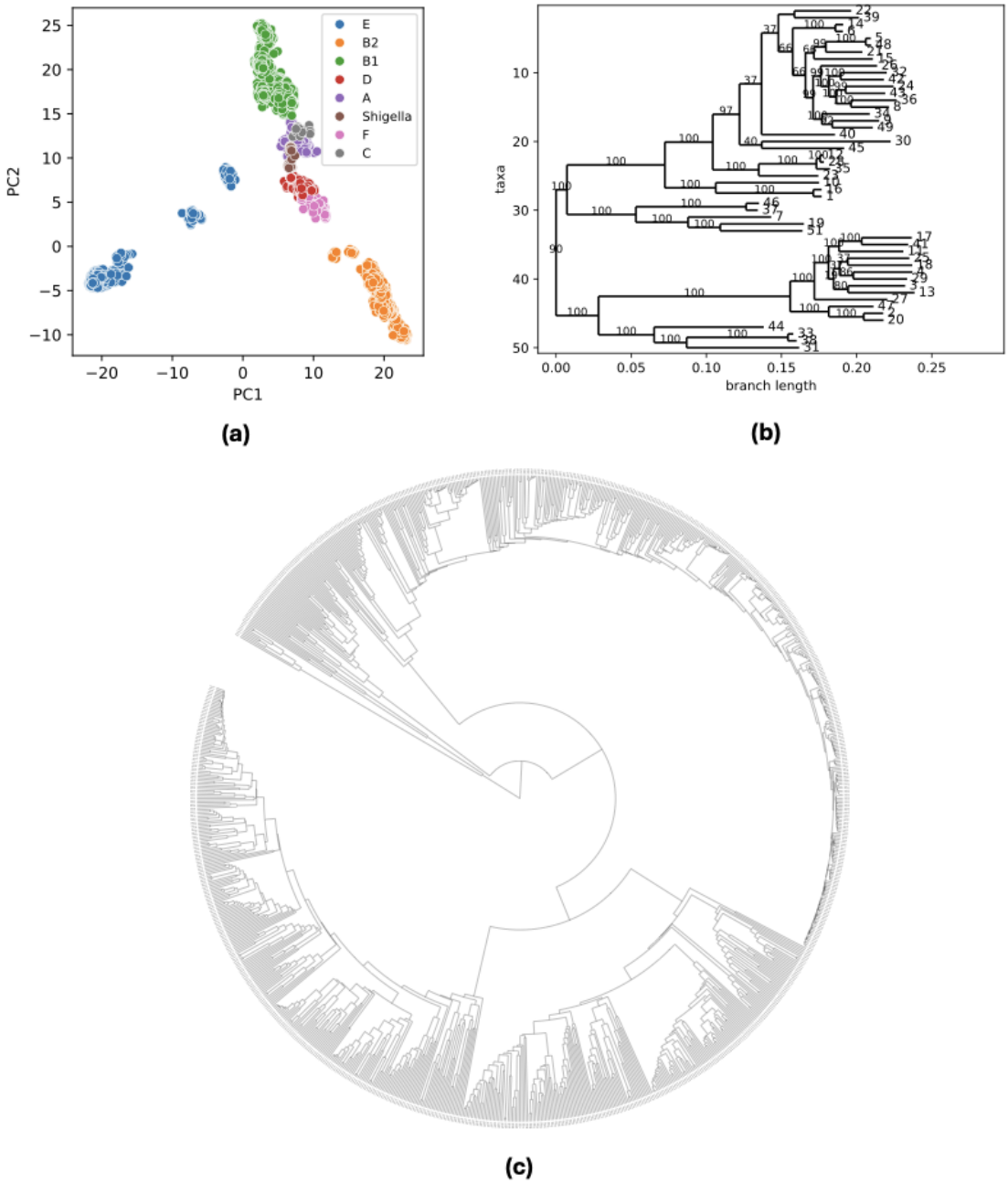

Figure 2: Genetic relatedness in the data. (a) PCA of the dataset coloured by the phylogroups. (b) Genetic tree of the dataset lineages. The branches show the genetic distance between each lineage and their common genetic origins. (c) Genetic distance tree of 1,000 dataset samples.

2.2 Exploring loss functions to capture minimal genomes

We developed a number of VAE model variations depending on the loss function (see Methods). Version v0 used the standard VAE loss function with terms for reconstruction loss and KL divergence. Version v1 includes a total gene number term and v2 explored cyclical annealing for the beta coefficient which modified KL divergence influence. The results are shown in Figure 3, where each column represents the results for models v0, v1 and v2 respectively.

Comparing the base model v0 with v1, we can notice more clustered points on the latent space by phylogroup and reduction in sparsity. Genome size distribution considerably changed in size as intended, with the median reducing from around 4500 to 3950 genes in a genome. The number of essential genes stayed relatively similar with the addition of the total genes term, with a slight reduction in minimum and maximum values. F1 score distribution remained the same with the loss function change. This shows that the model reconstruction ability was not compromised despite adding a total gene number term.

Comparing v1 to v2, there is no considerable difference in the genome size, essential gene number or F1 score distribution. However, a significant alteration occurred in the model's latent space. The clustered latent space in v1 became more sparse and there is a clear vertical division of phylogroups B2, B1 and E. An increase in sparsity and a decrease in the mixing of the points from different phylogroups allowed for the production of more meaningful latent space, prompting us to use the method for the final model.

2.3 Designing minimal *E. coli* genomes

We next focused on creating a model which can minimise bacterial genomes by modifying the influence of the total gene number term. We specifically focused on sampling the genomes from the minimal genomes region of the model's latent space. We experimented with both static weights and gamma coefficient scheduling to achieve these results. While the weights remain fixed throughout training, gamma acts as a dynamic scaling factor that progressively decreases over time. We found that using static weights above 2.5 caused the model to produce all-zero outputs, likely because the corresponding loss term became too dominant. However, when we used gamma scheduling, transitioning from 2.0 to 0.1 instead of from 1.0 to 0.1, we achieved successful training. Even though a larger static weight would exert a similar influence on the total gene number, the dynamic gamma allowed the model to train effectively. By varying gamma over time, the loss function adapts, gradually shifting focus from the total gene abundance to other aspects of the training process. The best combination we found is denoted v3, but the results for other models are shown in Tables 1 and 2.

The results for the best model are presented in Figure 4. The sampled genomes from the final model's minimal genomes region show a considerable reduction in genome size, with a median distribution of 2399 genes per genome (Figure 4a). This gene reduction was mirrored in the decrease in the essential gene numbers, with a median distribution of 293 out of 327 essential genes in the dataset per genome (Figure 4b). We also applied a minimisation algorithm on 100 of the sampled genomes and measured the distribution of actual sizes visualised in Figure 4c. The size variation resulted to be narrow, with 2.42 Mb minimum and 2.50 Mb maximum genome sizes, with a median of 2.42 Mb, which is overall smaller than the state of the art minimal *E. coli* genome of size 2.8 Mb [18]. Figure 4d shows the tradeoff between genome size and the number of essential genes.

Figure 4e illustrates the genetic distance tree constructed from 100 sampled *E. coli* genomes from the minimal genomes region of the model's latent space. In this tree, the length of each edge represents the genetic distance between genomes, specifically the number of genes not shared (differences in gene content). The tree shows shallow evolutionary divergence, which may be a result of the artificial selection of the genomes from one region of the latent space. The Collness and Sacking indices of 674 and 1118 respectively indicate a potential lower imbalance in the simulated data than in the original data.

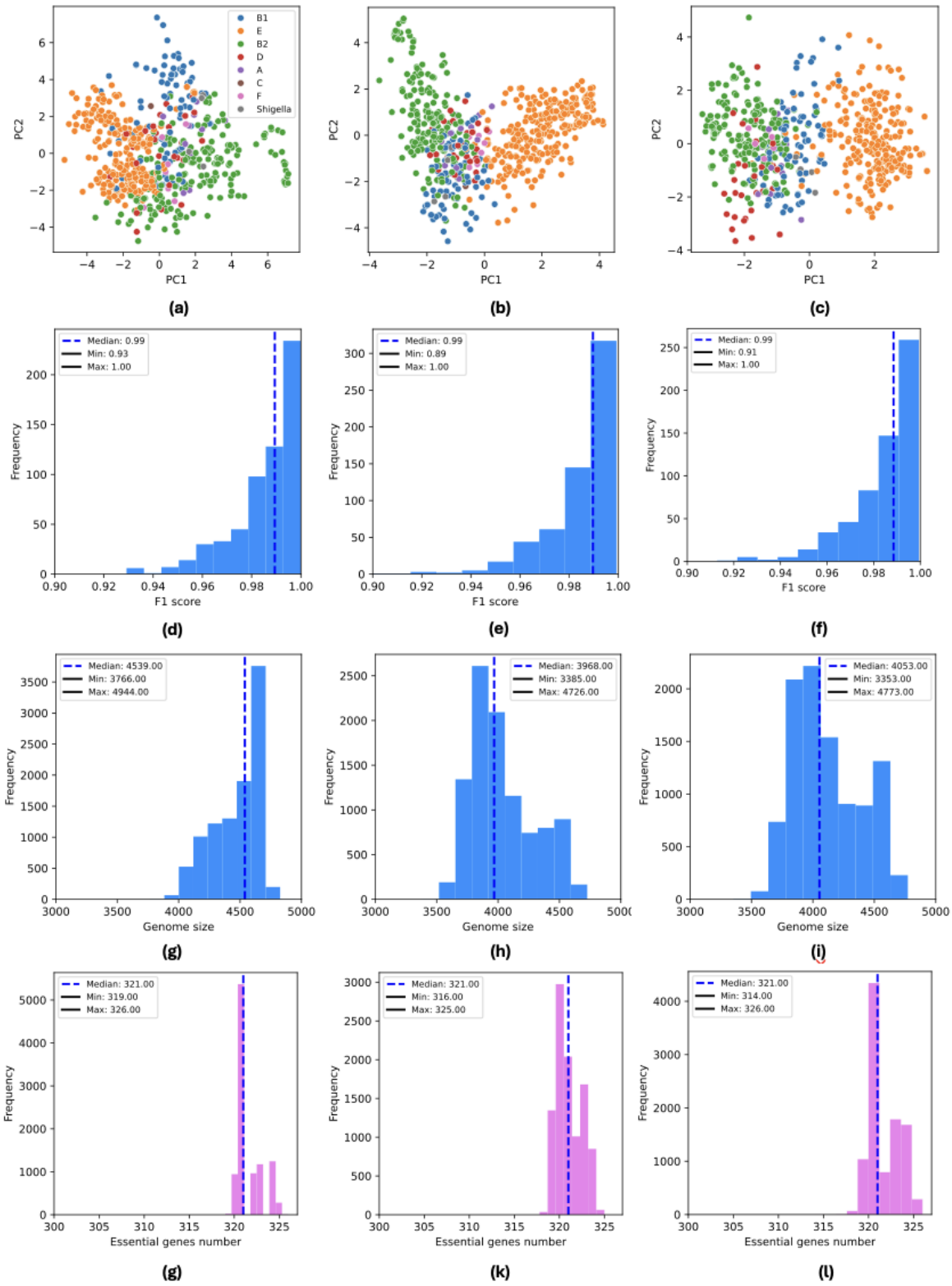

Figure 3: Latent space and distributions of F1 score, genome size and essential gene number for model v0 (a, d, g, j), v1 (b, e, h, k) and v2 (c, f, i, l).

	Genome size		
	Median	**Min**	**Max**
Original Dataset	4698	3814	5369
w =1	3327	3240	3723
w =1.5	2996	2923	3083
w =2	3351	3320	3450
w =2.5	2923	2838	3022

Table 1: Results for VAE model v3 for different weights with a gamma schedule from 1.0 to 0.2.

	Genome size		
	Median	**Min**	**Max**
Original Dataset	4698	3814	5369
w =1	2229	2187	2272
w =2	2339	2271	2517

Table 2: Results for VAE model v3 for different weights with a gamma schedule from 2.0 to 0.2.

3 Discussion

We elucidated the ability of the VAE to generate new minimised bacterial *E. coli* genomes by sampling latent vectors from a multivariate normal distribution, yielding genomes of the average size of 2.3 Mb, which is smaller than the current state-of-the-art minimised E. coli genomes of 2.8 Mb [18]. Such a significant reduction highlights the potential of this approach to create more efficient and streamlined genomes for industrial bioprocessing, where smaller genomes can lead to reduced metabolic burden and increased production yields. Our work extends the efforts of Dudek and Precup [25], who also explored machine learning driven bacterial genome generation using VAEs. While they focused on generating genomes across various bacterial species and faced challenges with small and unbalanced datasets, we concentrated solely on *E. coli* to address specific industrial and research applications. To achieve that, we used a large high quality dataset [26]

The outcome of the minimal genome generation experiments also shows a path to new genome reduction algorithms. A key challenge in genome minimisation is determining the organism's viability after gene deletions. In the future, by building a phylogenetic tree of the generated genomes, we can in principle estimate the genome at an internal nodes of the tree. There is therefore potential to traverse the tree in a greedy fashion, selecting the internal node with the largest number of descendants, removing those genes and then verifying the viability of the genome experimentally. If the genome is viable, we continue to the next internal node below; if that genome is not viable we move to the adjacent node with the next largest number of descendants. Future work will focus on validating this hypothesis and developing computational and experimental tools to implement this strategy.

There are a number of limitations. We used a set of essential genes to gauge whether the genomes were realistic, which has issues with not being well defined [27]. Whole-cell models could in the future also be used for validation purposes [19, 20, 21]. Ultimately, the generated genomes will need to be experimentally verified, which is currently a time consuming and costly process [1]. New experimental approaches for engineering the *E. coli* genome have been developed [28], and our modelling approach could help guide future attempts. We used a large data set of *E. coli* genomes and unsurprisingly found evidence of tree imbalance and limited genetic diversity. In the future, data could be combined from multiple closely related species inorder to increase genetic diversity. Another limitation of our approach was the limited hyperparameter tuning. Although we explored hidden dimensions, latent

dimensions, and the learning rate, in the future, a thorough exploration using Bayesian optimisation should be performed.

In conclusion, this study has further demonstrated the potential of using generative models to capture the variation in bacterial genomes sequences. The work has also practically demonstrated the ability of VAEs to generate minimised genomes with currently classified essential genes.

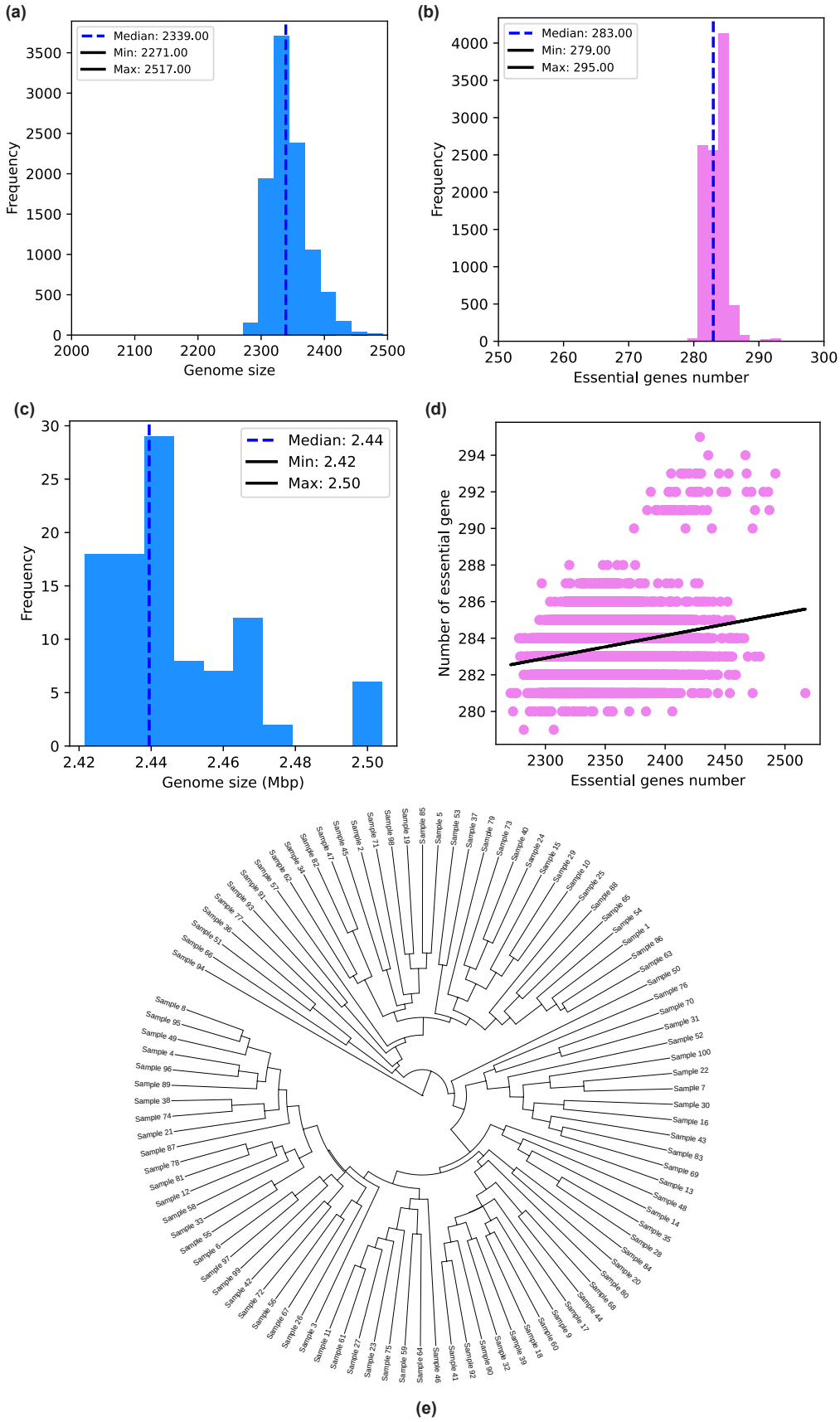

Figure 4: Minimal genome results for the final model sampled from the latent space. (a) Distribution of genome size. (b) Essential genes distribution. (c) Genome size distribution in Mb of 100 sampled genomes which have been processed through the reduction of the MG1655 *E. coli* genome. (d) Reduced genome size vs number of essential genes. (e) A phylogenetic tree created from the 100 sampled genomes using iTOL software.

4 Methods

4.1 Genomic data

4.1.1 Tree building

We constructed a phylogenetic tree using the Unweighted Pair Group Method with Arithmetic Mean (UPGMA) based on a Jaccard distance matrix derived from the presence-absence data of 1,000 dataset genomes. The Jaccard distance was computed to quantify the pairwise dissimilarity between the samples, which is well-suited for presence-absence data, capturing how different the genomes are regarding shared and unique genetic elements. The UPGMA algorithm was applied using the average linkage method, clustering the samples hierarchically based on their computed genetic distances. The resulting tree was visualised using the Interactive Tree of Life (iTOL) online tool [29]. We used several statistics to evaluate the trees using standrad packages [30, 31].

4.2 VAE model

4.2.1 Basic VAE model (v0)

All the VAE models were implemented using the PyTorch framework. The encoder consists of three fully connected layers with a ReLU activation function and additional batch normalisation. The first layer contains the input dimension of the total number of features (genes number) and a hidden dimension of 1,024 for the basic model and 512 for the rest. The output of the encoder is the mean and log variance for the latent space. The VAE also employs a reparametrisation trick that allows sampling to be differentiable; instead of sampling from the distribution described by mean and log squared standard deviation, the sampling is sampled from a normal distribution and then scaled to our distribution through:

$$z = \mu + \sigma \cdot \epsilon \tag{1}$$

Where μ - mean, σ - standard deviation and ϵ - sample from the standard normal distribution. The decoder consists of four fully connected layers. The first 3 layers contain the ReLU activation function and batch normalisation. The last layer contains the Sigmoid activation function, which gives outputs between 0 and 1. The first layer includes the same input and output dimensions as the encoder. To make the training process more efficient, the model uses Xavier initialisation for the weights of the linear layers and initialised biases to zero. The model diagram can be seen in figure 5.

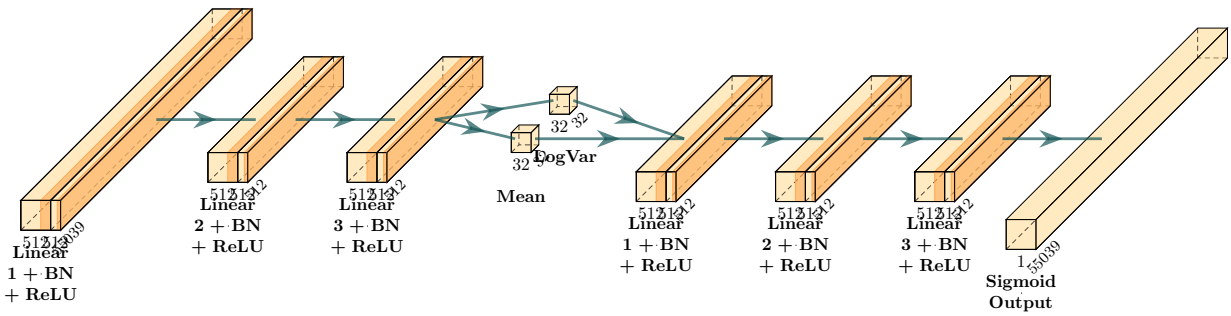

Figure 5: Implemented VAE model diagram.

The loss function of the base VAE consisted of the sum of reconstruction loss and KL divergence loss. The KL term is annealed linearly through a beta term from 0.1 to 1.0 (3.2).

$$\text{loss} = \text{reconstruction loss} + \beta \cdot \text{KL divergence} \tag{2}$$

Gradient clipping was used for all VAE model training to prevent the problem of exploding gradients, which occurs when gradients become large during backpropagation. This problem is common in deep architecture with exponentially growing gradients. The gradients were clipped to a maximum

value of 1.0, which ensured stable model training. When grid search was used, three parameters were optimised:

- Hidden Dimension: tested with values of 256, 512, and 1,024.
- Latent Dimension: tested with values of 32, 64, and 128.
- Learning Rate: tested with values of 0.01 and 0.001.

Finally, all models used the early stopping regularisation technique. This technique prevents the model from overfitting by monitoring its performance on a validation set and halting the training process if the performance does not improve to preserve its generalisation capability. The epoch number was set at 10,000 for all models.

4.2.2 VAE models with modified loss function (v1)

The first modification of the loss function included total gene number, which was linearly annealed with a gamma term from 1.0 to 0.1:

$$\text{loss} = \text{reconstruction loss} + \beta \cdot \text{KL divergence} + \gamma \cdot \text{total gene number} \tag{3}$$

Where gamma is linearly annealed from 1.0 to 0.1. L1 regularisation was used to prevent overfitting. It was added as a penalty term to the loss function. This method allows sparsity in the model parameters and drives the less important weights to zero to reduce noise, reducing the overfitting of the training data and improving generalisation. The lambda coefficient to control the regularisation was chosen to be 0.01.

4.2.3 VAE model with cyclic annealing (v2)

Previously, the KL divergence term followed a linear annealing schedule. Linear annealing is not always the optimal schedule because it can apply too much regularisation too early. Early regularisation hinders models' ability to explore the latent space and leads to poor reconstruction [32]. Therefore, the KL divergence term was annealed to cyclic annealing for more optimal learning of the latent space. Such an annealing schedule introduces high and low regularisation periods, letting the model explore different regions of the latent space. In this case, the annealing follows the cosine structure, where the weight of the KL divergence term changes periodically following the cosine function, ramping up and down. The cyclic annealing schedule aims to enhance the learning of the latent space, ensuring a balanced trade-off between reconstruction accuracy and latent space regularisation. The loss function for this model looked the following:

$$\text{loss} = \text{reconstruction loss} + \beta \cdot \text{KL divergence} + \gamma \cdot \text{total gene number} \tag{4}$$

L1 regularisation was also used.

4.2.4 VAE model with modified loss function, cyclic annealing and additional weights (v3)

The final model includes all the modifications from VAE v2. However, it also includes an additional weight ω for the total gene number term to see how the increase in the influence of that term reduced the genome size distribution. We used 4 weights, 1, 1.5, 2 and 2.5 to test the loss function. The final loss function equation looked the following:

$$\text{loss} = \text{reconstruction loss} + \beta \cdot \text{KL divergence} + \omega \cdot \gamma \cdot \text{total gene number} \tag{5}$$

We also used the same loss function but with a different gamma that is linearly annealed from 2.0 to 0.1. L1 regularisation was also used.

4.3 Sampling

To generate new genomic sequences, we sampled 10,000 latent vectors from a multivariate normal distribution whose dimensionality matched the model's latent space. We assumed a standard normal distribution with a mean vector of zero and an identity covariance matrix. Each sampled latent vector was then passed through the decoder of the corresponding model to generate new genomic sequences. Such an approach allowed us to assess the diversity of the genomes that the model can produce.

We focused on the latent vector corresponding to the smallest genome identifier among the initial 10,000 samples to identify regions in the latent space associated with smaller genomes. Using the selected latent vector as the central point, we conducted additional sampling of new latent vectors from a multivariate normal distribution around this point. The distribution of the smaller genomes was only measured for relevant models. The covariance matrix was scaled by a factor of 0.1, narrowing down the sampling around the central point and focusing on smaller genomes. The newly sampled latent vectors were decoded to produce additional genomic sequences. We calculated the lengths of the generated genomes by summing the total number of genes in each sample. The lengths of the genomes were calculated as genome size distribution for each model.

4.4 Results validation and exploration

The model training was assessed by measuring training and validation losses over the trained epochs. F1 and accuracy scores were also calculated to assess the model's performance.n The latent space of the VAE models was used to validate and explore the results, especially in the basic VAE model. Such analysis allowed us to verify the model's ability to explore latent space and see valuable patterns that the VAE encoded in the data of interest.

4.5 Genome Minimisation Algorithm

An algorithm was developed to generate new minimised genomes that loads a wild-type sequence of the K12 MG1655 *E. coli* genome and a list of genes that must be present in each final genome. The algorithm then extracts non-essential genes to be removed from the wild-type sequence by comparing whether the gene is present in the list of genes that must be present in the sequence of interest. When non-essential gene features are extracted, their corresponding nucleotide positions are marked for removal. However, all the essential genes are still kept in the sequences. The algorithm then generates the minimised genomes by removing the positions marked for removal and removing genetic segments while keeping the other features intact. The algorithm saves the generated sequence in a Fasta file.

References

[1] Xin Xu, Felix Meier, Benjamin A Blount, Isak S Pretorius, Tom Ellis, Ian T Paulsen, and Thomas C Williams. Trimming the genomic fat: minimising and re-functionalising genomes using synthetic biology. *Nature Communications*, 14(1):1984, 2023.

[2] Fang Zhang, Kaiyue Huo, Xingyi Song, Yufen Quan, Shufang Wang, Zhiling Zhang, Weixia Gao, and Chao Yang. Engineering of a genome-reduced strain bacillus amyloliquefaciens for enhancing surfactin production. *Microbial Cell Factories*, 19(1):223, 2020.

[3] Rocío Aguilar Suárez, Michael Kohlstedt, Ayşegül Öktem, Jolanda Neef, Yuzheng Wu, Kaiya Ikeda, Ken-Ichi Yoshida, Josef Altenbuchner, Christoph Wittmann, and Jan Maarten van Dijl. Metabolic profile of the genome-reduced bacillus subtilis strain iig-bs-27-39: An attractive chassis for recombinant protein production. *ACS Synthetic Biology*, 13(7):2199–2214, 07 2024.

[4] Viktor Vernyik, Ildikó Karcagi, Edit Tímár, István Nagy, Ádám Györkei, Balázs Papp, Zsuzsanna Györfy, and György Pósfai. Exploring the fitness benefits of genome reduction in escherichia coli by a selection-driven approach. *Scientific Reports*, 10(1):7345, 2020.

[5] Duolong Zhu, Yuxin Fu, Fulu Liu, Haijin Xu, Per Erik Joakim Saris, and Mingqiang Qiao. Enhanced heterologous protein productivity by genome reduction in lactococcus lactis nz9000. *Microbial Cell Factories*, 16(1):1, 2017.

[6] Shamik S Sharma, John W Campbell, David Frisch, Frederick R Blattner, and Sarah W Harcum. Expression of two recombinant chloramphenicol acetyltransferase variants in highly reduced genome escherichia coli strains. *Biotechnology and Bioengineering*, 98(5):1056–1070, 2007.

[7] Taku Morimoto, Ryo Kadoya, Kei Endo, Masaya Tohata, Kohei Sawada, Shou Liu, Tetsuya Ozawa, Toshihiro Kodama, Hiroshi Kakeshita, Yasuhiro Kageyama, Kazuyuki Manabe, Shigehiko Kanaya, Katsuhiro Ara, Kunio Ozaki, and Noriyuki Ogasawara. Enhanced recombinant protein productivity by genome reduction in bacillus subtilis. *DNA research*, 15(2):73–81, 2008.

[8] Jun Hyoung Lee, Bong Hyun Sung, Mi Sun Kim, Frederick R Blattner, Byoung Hoon Yoon, Jung Hoe Kim, and Sun Chang Kim. Metabolic engineering of a reduced-genome strain of escherichia coli for l-threonine production. *Microbial Cell Factories*, 8:1–12, 2009.

[9] Nicole LeBlanc and Trevor C Charles. Bacterial genome reductions: Tools, applications, and challenges. *Front Genome Ed*, 4:957289, 2022.

[10] R Z Moger-Reischer, John I Glass, Kenneth S Wise, Lijie Sun, Debora M C Bittencourt, Blake K Lehmkuhl, David R Schoolmaster, Michael Lynch, and John T Lennon. Evolution of a minimal cell. *Nature*, 620(7972):122–127, 2023.

[11] Clyde A Hutchison 3rd, Ray-Yuan Chuang, Vladimir N Noskov, Nacyra Assad-Garcia, Thomas J Deerinck, Mark H Ellisman, John Gill, Krishna Kannan, Bogumil J Karas, Li Ma, Jean-Francois Pelletier, Zhi-Qiang Qi, Ryan A Richter, Elizabeth A Strychalski, Lijie Sun, Yukari Suzuki, Borislava Tsvetanova, Kenneth S Wise, Hamilton O Smith, John I Glass, Carole Merryman, Daniel G Gibson, and J Craig Venter. Design and synthesis of a minimal bacterial genome. *Science*, 351(6280):aad6253, 2016.

[12] Masaomi Kurokawa and Bei-Wen Ying. Experimental challenges for reduced genomes: The cell model escherichia coli. *Microorganisms*, 8(1), 2020.

[13] Hiroshi Mizoguchi, Hideo Mori, and Tatsuro Fujio. Escherichia coli minimum genome factory. *Biotechnology and Applied Biochemistry*, 46(3):157–167, 2007.

[14] Hiroshi Mizoguchi, Yoshie Sawano, Jun ichi Kato, and Hideo Mori. Superpositioning of deletions promotes growth of escherichia coli with a reduced genome. *DNA Res*, 15(5):277–284, 10 2008.

[15] 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.

[16] György Pósfai, Guy Plunkett, Tamás Fehér, David Frisch, Günther M Keil, Kinga Umenhoffer, Vitaliy Kolisnychenko, Buffy Stahl, Shamik S Sharma, Monika de Arruda, Valerie Burland, Sarah W Harcum, and Frederick R Blattner. Emergent properties of reduced-genome ¡i¿escherichia coli¡/i¿. *Science*, 312(5776):1044–1046, 2006.

[17] Ildikó Karcagi, Gábor Draskovits, Kinga Umenhoffer, Gergely Fekete, Károly Kovács, Orsolya Méhi, Gabriella Balikó, Balázs Szappanos, Zsuzsanna Györfy, Tamás Fehér, Balázs Bogos, Frederick R Blattner, Csaba Pál, György Pósfai, and Balázs Papp. Indispensability of horizontally transferred genes and its impact on bacterial genome streamlining. *Molecular Biology and Evolution*, 33(5):1257–1269, 01 2016.

[18] 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, 09 2011.

[19] 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.

[20] Joshua Rees-Garbutt, Jake Rightmyer, Oliver Chalkley, Lucia Marucci, and Claire Grierson. Testing theoretical minimal genomes using whole-cell models. *ACS Synthetic Biology*, 10(7):1598–1604, 2021.

[21] Ioana M Gherman, Joshua Rees-Garbutt, Wei Pang, Zahraa S Abdallah, Thomas E Gorochowski, Claire S Grierson, and Lucia Marucci. Accelerated design of escherichia coli genomes with reduced size using a whole-cell model and machine learning. *bioRxiv*, pages 2023–10, 2023.

[22] Diederik P Kingma and Max Welling. Auto-encoding variational bayes., 2022.

[23] Burak Yelmen, Aurélien Decelle, Leila Lea Boulos, Antoine Szatkownik, Cyril Furtlehner, Guillaume Charpiat, and Flora Jay. Deep convolutional and conditional neural networks for large-scale genomic data generation. *PLOS Computational Biology*, 19(10):1–21, 10 2023.

[24] Burak Yelmen, Aurélien Decelle, Linda Ongaro, Davide Marnetto, Corentin Tallec, Francesco Montinaro, Cyril Furtlehner, Luca Pagani, and Flora Jay. Creating artificial human genomes using generative neural networks. *PLOS Genetics*, 17(2):1–22, 02 2021.

[25] N K Dudek and D Precup. Towards ai-designed genomes using a variational autoencoder. *bioRxiv*, 2024.

[26] Gal Horesh, Grace A Blackwell, Gerry Tonkin-Hill, Jukka Corander, Eva Heinz, and Nicholas R Thomson. A comprehensive and high-quality collection of escherichia coli genomes and their genes. *Microbial Genomics*, 7(2), 2021.

[27] Emily C A Goodall, Ashley Robinson, Iain G Johnston, Sara Jabbari, Keith A Turner, Adam F Cunningham, Peter A Lund, Jeffrey A Cole, and Ian R Henderson. The essential genome of ¡i¿escherichia coli¡/i¿ k-12. *mBio*, 9(1):10.1128/mbio.02096–17, 2018.

[28] Wesley E Robertson, Louise FH Funke, Daniel de la Torre, Julius Fredens, Kaihang Wang, and Jason W Chin. Creating custom synthetic genomes in escherichia coli with rexer and genesis. *Nature Protocols*, 16(5):2345–2380, 2021.

[29] Ivica Letunic and Peer Bork. Interactive tree of life (itol) v6: recent updates to the phylogenetic tree display and annotation tool. *Nucleic Acids Research*, 52(W1):W78–W82, 04 2024.

[30] Thijs Janzen and Rampal S. Etienne. Phylogenetic tree statistics: a systematic overview using the new r package 'treestats'. *Molecular Phylogenetics and Evolution*, 200:108168, 2024.

[31] Mareike Fischer, Lina Herbst, Sophie Kersting, Luise Kühn, and Kristina Wicke. Tree balance indices: a comprehensive survey. will be submitted shortly, 2021.

[32] Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin. Cyclical annealing schedule: A simple approach to mitigating kl vanishing., 2019.