Predicting gene essentiality in prokaryotes using deep learning

Ian Lo and Carolin Schulte
MIT 20.C51
Spring 2022

Abstract

Essential genes are commonly defined as those genes that are absolutely required for the survival of an organism, at least under certain growth conditions. Due to the importance of essential genes in evolutionary and synthetic biology studies as well as drug design, high-throughput experimental efforts have been developed to identify essential genes in bacteria. Using a large set of data obtained mainly from transposon insertion-sequencing experiments, this study developed a feed-forward neural network to predict gene essentiality in bacteria based on readily obtainable features of the gene sequence. The model achieved an overall test AUC score of 0.79. The analyses further highlighted the importance of accounting for gene orthology when defining training and test sets for model development. Our results therefore present an important step towards developing accurate models predicting gene essentiality, which have the potential to substantially accelerate research efforts in several areas.

1 Introduction

Identifying essential genes in bacteria is of major importance in many fields of research: with regard to basic research questions, elucidating gene essentiality has the potential to expand our understanding of fundamental concepts underlying the functioning of a cell, improving genome annotations, and determining context-specific metabolic pathway usage [1]. More applied research endeavors have focused on leveraging knowledge of gene essentiality for creating synthetic cells with a minimal number of genes [2] or designing drugs targeted against various pathogens [1].

Determining gene essentiality on the genome-scale was initially a very labor-intensive process whereby single gene deletion mutants had to be created experimentally [3]. Recent advances in experimental methods, such as transposon-insertion sequencing approaches, have greatly enhanced our ability to identify essential genes in a high-throughput manner. However, these methods still require substantial experimental effort and are limited to genetically tractable organisms that can be cultured in laboratory conditions [4]. Due to these drawbacks and the rapidly increasing availability of genome sequences, computational methods are now commonly used for predicting gene essentiality. While homology- and metabolic modeling-based approaches have been successfully applied to this end, they either rely on the existence of conserved orthologs or are limited to genes involved in metabolic processes [5], which substantially limits their scope. In contrast, machine learning models trained on data for model organisms have the potential to identify universal features of essential genes and accurately predict essential genes in unstudied species. Several studies have explored machine learning methods such as Logistic Regression, Decision Tree or Random Forest [6], however Deep Learning methods have so far not been extensively examined [7]. In addition, early studies in particular only used small data sets for a few well-investigated bacterial species, limiting the scope of the models [8, 9].

In this study, we develop a feed-forward neural network to predict the genes essential for survival based on data for 64 bacterial strains and trained solely on features derived from the gene sequence. The DeeplyEssential model developed by Hasan and Lonardi recently introduced the application of a deep neural network for this task [7]. This work aims to improve against this baseline approach by training on a larger sample size and expanding on the features that are used as inputs. We anticipate that an accurate Deep Learning model will aid researchers in efficiently elucidating bacterial survival mechanisms and serve as a secondary validation of experimental results.

2 Materials and methods

2.1 Data

The data for this study were obtained from the Online Gene Essentiality (OGEE) Database [10] which is a comprehensive database of published gene essentiality data. The data is collected both from large-scale gene essentiality studies, such as transposon insertion-sequencing experiments, and through text mining of publications to add data from small-scale studies. While results from transposon insertion-sequencing experiments are generally highly accurate, it is important to note that they are still imperfect, and biases depending on gene length or nucleotide composition are possible [4]. In addition, gene essentiality can not always be unambiguously defined and may be context-specific.

Only prokaryotic gene essentiality data were considered in this analysis. The bacterial strains were matched with the GenBank accessions in NCBI that correspond to the genome assembly with the gene identifiers used to report the results of the experimental data. Genes were then matched with the corresponding gene and amino acid sequence as well as the strand they are encoded on based on the GenBank file. The analysis was limited to protein-coding sequences, and pseudogenes or genes coding for non-translated RNAs were therefore excluded.

The nucleotide and amino acid sequence of each gene was retrieved since it contains information connected to its essentiality, which enables the use of machine learning-based approaches to predict which genes are critical for survival. The DNA sequences and their corresponding amino acid sequences were used to generate the features described in the following section. Information about whether the gene is encoded on the leading or lagging strand was included as essential genes tend to be more conserved in the leading strand [11]. In addition, since it has been found that essential genes are preferentially located in the front position of operons [12], the operon position of each gene was determined using Operon-mapper [13], a publicly available web-based tool that predicts operons for a given bacterial genomic nucleotide sequence based on the intergenic distance of neighboring genes and the functional relationships of their protein-coding products.

Based on a review of published models for predicting gene essentiality, the features calculated for each gene were gene length, codon frequency, amino acid frequency, strand, GC content, codon adaptation index (CAI), and operon position. The codon adaptation index was calculated using the `cai_for_gene` function from the `BioPython` package. Three categories were considered for the operon position: (i) gene not part of an operon, (ii) gene in the front position of an operon, and (iii) gene in any other position of an operon. These features were concatenated into a single vector to be passed as the input layer to the neural network. The reason for using a neural network model based solely on sequence-based features was that it enhances the general applicability of the model because other gene information such as structural or topological attributes are more difficult to obtain, especially for non-model organisms [7].

2.2 Machine learning approaches

Logistic regression and Random Forest were implemented as baseline methods to compare the performance of the neural network to. Models were fitted using the `sklearn` package in Python. Hyperparameters that were tuned included the regularization parameter and type of regularization for logistic regression and the number of trees, minimum number of samples required for splitting a node and the maximum tree depth for Random Forest. A random search with 5-fold cross validation using the ROC-AUC score (Area Under the Receiver Operating Characteristic Curve) as an evaluation metric was performed for hyperparameter optimization.

A feed-forward neural network was chosen as the machine learning method most suitable for performing supervised binary classification of gene essentiality based on the format of the inputs. Neural networks were implemented in PyTorch [14]. Binary cross entropy was used as a loss function, ReLU was used as an activation function, and dropout layers were implemented to prevent overfitting. Hyperparameter optimization for the neural network was performed using the `Optuna` framework [15] to find the model architecture, optimizer, and learning rate that maximized the ROC-AUC score

on a validation data set (Table S2).
The data was split into 80/20 training and test set for logistic regression and Random Forest, and a 70/10/20 split was used to obtain training, validation and test set for training neural networks (Table S1). This process was repeated using five different random seeds to assess reliability of the estimated model performance. To define distinct training and test sets that do not share genes derived from a common ancestor, groups of orthologous genes were defined using OrthoFinder [16]. Training and test data were then obtained by pseudo-random sampling, where all genes that are members of the same orthogroup had to be contained in the same set. Genes that could not be assigned to an orthogroup were randomly assigned to training or test set. As essential genes comprised only 24% of the data set, we also hypothesised the oversampling essential genes would improve model performance. To this end, naive oversampling was implemented using the `imblearn` package in Python.

In addition to the main analysis involving the entire data set with oversampling of essential genes in the training set and stratification by orthology for generating train and test sets, separate analyses were conducted where no oversampling was performed or gene orthology was not considered in the training and test split. A separate analysis was performed where the model was trained on the entire data set apart from one bacterial species which served as the test set. This was done to assess model performance in a realistic use case scenario.

The ROC-AUC score was used as the main metric to evaluate model performance since it characterizes the ability of a model to distinguish between two classes. Due to the low prevalence of essential genes in the data set, accuracy was only considered as a secondary performance metric and not used for hyperparameter tuning. For the feed-forward neural network, the final training after hyperparameter optimization was implemented using early stopping if the validation loss increased more than 20 times since it was observed that a high number of epochs often led to overfitting to the training data. Additional metrics reported in this study include recall defined as the number of true positives divided by the total number of positive samples as well as precision, which is the the number of true positives divided by the total number of positive class predictions.

3 Results

3.1 Summary statistics of the data

The final data set contained 92,624 genes, of which 22,275 (24.05%) have been experimentally determined to be essential (Table S3). The comparatively high fraction of essential genes indicates that some data sets in OGEE may preferentially or exclusively report essential genes. The data set contained a diverse range of bacterial species, with the majority (81.16%) being Gram-negative. The largest group of Gram-negative bacteria among both essential and non-essential genes was Gammaproteobacteria, which is unsurprising considering that this group contains several model organisms such as *Escherichia coli*, *Salmonella enterica* or *Pseudomonas aeruginosa*. The absence of non-essential genes for Deltaproteobacteria and Bacteroidetes confirms that some experimental data sets in OGEE only included essential genes. In addition, it can be seen that among essential genes, a lower fraction of genes belonged to Firmicutes and Gammaproteobacteria compared to non-essential genes, but a lower fraction of genes belonged to Betaproteobacteria. No difference regarding the strand positioning of essential and non-essential genes was observed. While essential genes were less likely to occur outside of operons compared to the non-essential genes, a higher fraction was predicted to be in a position other than the front of an operon. This was unexpected considering that essential genes have been suggested to preferentially occur in the front position of operons [12].

3.2 Baseline methods

Modelling results in this and the following section are reported for three different models that were separately trained. The reference model was trained using a training data set that was augmented by naive oversampling of essential genes, and gene orthology was taken into account when dividing the data set into training, validation and test sets. It has been shown that neglecting gene orthology for training and test splits can cause data leakage and thus artificially inflate the estimated performance on the test set [7]. The other two models were trained either without oversampling or without factoring in orthology.

Logistic regression and Random Forest models were implemented to establish baselines for classification performance. As can be seen in Fig. 1, Random Forest generally performed better than logistic regression and achieved a maximum ROC-AUC score of 0.806 on the test set when the training set contained an equal number of essential and non-essential genes after oversampling. Random Forest performed much worse when no oversampling was used, even though this did not affect the performance of logistic regression. Whether gene orthology was considered when generating training and test sets did not appear to affect the performance of either method.

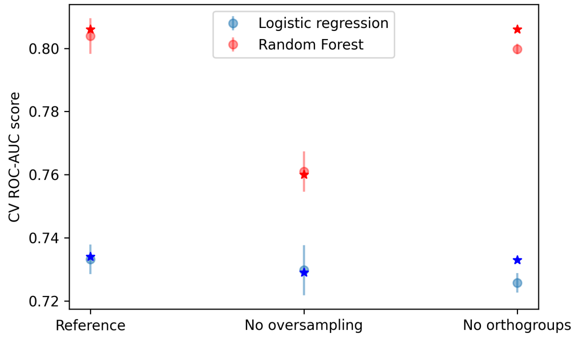

Figure 1: Performance of logistic regression and Random Forest. Shown is the mean and standard deviation of of the ROC-AUC score obtained in five-fold cross-validation after hyperparameter optimization using random search. Star symbols indicate the test set ROC-AUC score.

Despite its worse predictive performance, logistic regression offers the advantage of interpretability, where model coefficients can be examined to identify the effect of individual predictors. For all three data sets, the codon frequency for ATC (Ile), GAA (Glu) and GTA (Val) as well as the frequency of the amino acid leucine and non-front operon position were found among the 10 covariates that most strongly contributed to predicting essentiality for a gene. ATC and GAA have been previously reported to occur at higher frequencies in essential genes compared to non-essential genes [7], and the result regarding operon position was expected considering the summary statistics in Table S3. These findings indicate that frequency of branched-chain amino acids in particular may be an important feature distinguishing essential from non-essential genes.

3.3 Feed-forward neural network

A feed-forward neural network for predicting gene essentiality was implemented next for all three data sets. To this end, the hyperparameters shown in Table S2 were optimized using Optuna and the ROC-AUC score as a performance metric for parameter tuning. The network was trained to minimize the loss function `BCEWithLogitsLoss` from Pytorch, which combines a Sigmoid layer and binary cross-entropy loss into one single class. This version is considered more numerically stable than using a plain Sigmoid followed by a binary cross-entropy loss as, by combining the operations into one layer, it takes advantage of the log-sum-exp trick for numerical stability [17].

As can be seen in Table 1 and Fig. 2A, all models achieved an ROC-AUC score on the test set between 0.79 and 0.82. Four additional random trials validated that fluctuation of the scores due to randomization of training and test splits was minimal, with standard deviation across all five trials below 0.017 for each model type (Table S4). These scores are similar to test set performance of published models with comparable scope [7, 8, 18, 19], but also do not present a substantial improvement over the predictive performance of the Random Forest model. As suggested in [7], it can be seen that disregarding gene orthology during the creation of train and test set led to slightly improved test set performance, which may be explained by data leakage caused by the orthology of conserved genes. Oversampling of essential genes in the training data did not appear to impact model performance and precision and recall were very similar to the reference model. In contrast, the model trained without accounting for gene orthology had lower precision but higher recall for essential genes indicating that a higher fraction of truly essential genes was correctly classified but there was also a higher fraction of false positive predictions. This is likely due to the data leakage enabling the model to identify more essential genes based on orthology. The increase in false positive predictions may be due to context-specific gene essentiality in the experimental data as well as differences in the essentiality of orthologous genes in different species.

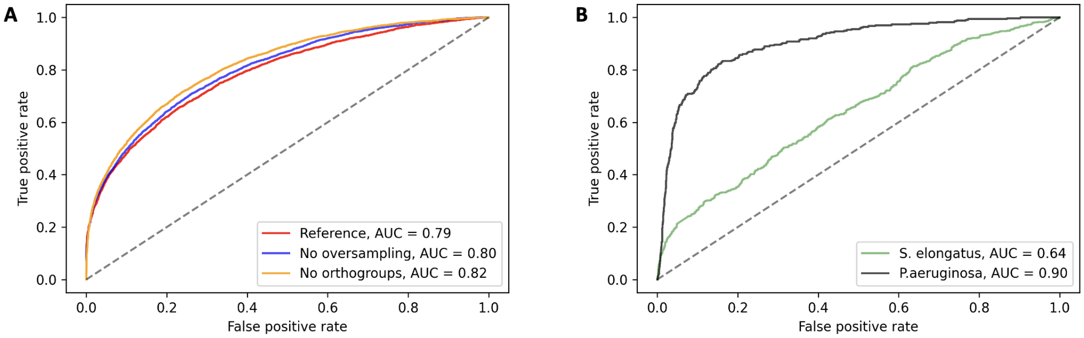

Figure 2: Performance metrics for the feed-forward neural networks evaluated on the test set after hyperparameter tuning for (**A**) models trained and tested on samples of the entire data set and (**B**) models trained on all species except one that was held out as a test set. The leave-one-species-out approach was run separately using either *S. elongatus* or *P. aeruginosa* as the test species.

Table 1: Summary of neural network performance on the test set.

	ROC-AUC score	**Precision**	**Recall**
Reference model	0.7881		
Essential genes		0.5767	0.4954
Non-essential genes		0.8573	0.8929
No oversampling	0.8035		
Essential genes		0.6016	0.4856
Non-essential genes		0.8566	0.9053
No orthogroups	0.8200		
Essential genes		0.4698	0.7371
Non-essential genes		0.8980	0.7356

To investigate which genes the reference model did or did not perform well for, the functions of all genes in the test data were examined. The analysis was limited to the 15 most commonly occurring functional annotations in each group (Table S5 and S6). These annotations were obtained during the operon mapping process and are based on best hits in a search against a subset of the Uniprot database [13]. Correctly classified essential genes mainly included those coding for ribosomal proteins, genes involved in cell division and genes encoding enzymes involved in highly conserved and widely used metabolic pathways, e.g., subunits of the pyruvate/2-oxoglutarate dehydrogenase complex and glucosamine 6-phosphate synthetase. In contrast, functional groups of essential genes that were wrongly classified as non-essential were much less well-defined and comprised a large fraction of uncharacterized conserved proteins. It can also be seen that several transporter and membrane proteins were falsely classified as non-essential. This may partly be due to the heterogeneity of transport proteins, which would make it difficult for the model to learn patterns in the corresponding genes. In addition, this finding could be due to the nature of the data set, where gene essentiality was determined in different growth media, meaning that reported essentiality of transport proteins would be context-specific and not expected to generalise across genes in different species.

Among non-essential genes that were incorrectly classified as essential, it is interesting to note that several of the functional groups correspond to those that are also prevalent among essential genes, such as ribosomal proteins or subunits of the pyruvate/2-oxoglutarate dehydrogenase complex. It is therefore possible that some bacterial species can either use alternative metabolic pathways or have multiple copies of those genes, some of which are non-functional or non-essential (paralogs) and the model is unable to distinguish those genes from truly essential genes based on the input features that were used. Similar to the reasoning for genes encoding transporters described above, another explanation would be the imperfect determinations using transposon insertion-sequencing, where the experimental results wrongly label essential genes as non-essential.

3.4 Leave one species out test

For practical application, it may be of interest to assess how well the model performs on genes from a species that was not part of the training data. To this end, the model was trained (and optimized) twice on the entire data set while excluding either (i) *Synechococcus elongatus* PCC 7942 = FACHB-805 or (ii) *Pseudomonas aeruginosa* PAO1 to be used as the test set. Training and validation sets were still stratified by gene orthology. The species that were left out were specifically chosen to evaluate model predictive performance for (i) a strain that is phylogenetically distant from the other strains in the data set (*S. elongatus* is the only cyanobacterium in the data set) and (ii) a strain that is closely related to many strains in the training data (the data set contains several samples for pseudomonads and Gammaproteobacteria in general). For *S. elongatus*, the ROC-AUC score was low (0.64) whereas very good predictive performance (ROC-AUC score = 0.90) was obtained for *P. aeruginosa*. This result highlights that while overall predictive performance of the model is good, substantial improvements such as incorporating alternative features will be necessary for application to non-model species or in general species dissimilar to those used to train the neural network. Nevertheless, accurate gene essentiality prediction for bacterial strains that are similar to those in the training data may be useful for certain research endeavors.

4 Discussion

In this study, a feed-forward neural network was trained using a large data set of gene essentiality data for diverse bacterial species. Using simple features derived from the gene sequence alone, the model achieved good predictive performance with an AUC score of 0.79, which is similar to published models. However, the model was found to perform better for bacterial strains similar to those making up the majority of the training data. Our analyses further highlighted the importance of accounting for gene orthology when splitting data into training and test sets to avoid data leakage.

While the feed-forward neural network achieved good predictive performance, it did not perform better than the Random Forest model, indicating that a larger, more diverse data set or the use of more complex features and architectures may be required to improve model performance. In particular, since our results indicated that genes with similar functions but different essentiality were not well distinguished by the model, additional features or network architectures such as recurrent neural networks, which can account for sequence context, could be helpful to implement. The features used in this study can be easily obtained for any organism with a sequenced genome, making our approach attractive for predicting gene essentiality in non-model organisms with limited availability of experimental data. However, due to increasing capabilities of computational tools that accurately predict features such as protein-protein interactions, including those features as additional predictors could be useful as has been shown in previous studies [20].

A general limitation of the data used in this study is the ambiguity surrounding the definition of gene essentiality, which is expected to have introduced noise into the labels that were used for training and evaluating our models. This noise is partly due to varying conditions in which gene essentiality experiments were performed, but also due to the inherent imperfect accuracy of high-throughput gene essentiality experiments, where very short genes or genes lacking TA sites may for example not allow for the integration of transposons and be wrongly labeled as essential. For operons, downstream effects of transposon insertion can hamper accurate determination of which genes in an operon are actually essential [21], which could explain why the distribution of operon positions in the data set was different from what was expected [12].

Overall, the model developed in this study presents a promising deep learning framework to develop highly accurate models for predicting gene essentiality. Due to the increasing availability of transposon insertion-sequencing data, additional models could be developed to specifically predict gene essentiality in pathogens under simulated *in vivo* conditions, which would be beneficial for the identification of novel drug targets.

Author contribution statement

Both authors contributed to all parts of this work. C.S. performed initial data wrangling and identification of operon position. I.L. implemented orthogroup assignment and stratification. Both authors contributed equally to conceptualizing the analyses, writing code, interpreting the results and writing the report.

References

[1] Giulia Rancati et al. "Emerging and evolving concepts in gene essentiality". en. In: *Nature Reviews Genetics* 19.1 (Jan. 2018). Number: 1 Publisher: Nature Publishing Group, pp. 34–49. ISSN: 1471-0064. DOI: 10.1038/nrg.2017.74. URL: https://www.nature.com/articles/nrg.2017.74 (visited on 03/13/2022).

[2] Daniel G. Gibson et al. "Creation of a Bacterial Cell Controlled by a Chemically Synthesized Genome". In: *Science* 329.5987 (July 2010). Publisher: American Association for the Advancement of Science, pp. 52–56. DOI: 10.1126/science.1190719. URL: https://www.science.org/doi/full/10.1126/science.1190719 (visited on 03/13/2022).

[3] Tomoya Baba et al. "Construction of Escherichia coli K-12 in-frame, single-gene knockout mutants: the Keio collection". In: *Molecular Systems Biology* 2 (Feb. 2006), p. 2006.0008. ISSN: 1744-4292. DOI: 10.1038/msb4100050. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1681482/ (visited on 03/13/2022).

[4] Amy K. Cain et al. "A decade of advances in transposon-insertion sequencing". en. In: *Nature Reviews Genetics* 21.9 (Sept. 2020). Number: 9 Publisher: Nature Publishing Group, pp. 526–540. ISSN: 1471-0064. DOI: 10.1038/s41576-020-0244-x. URL: https://www.nature.com/articles/s41576-020-0244-x (visited on 03/13/2022).

[5] Olufemi Aromolaran et al. "Machine learning approach to gene essentiality prediction: a review". In: *Briefings in Bioinformatics* 22.5 (Sept. 2021), bbab128. ISSN: 1477-4054. DOI: 10.1093/bib/bbab128. URL: https://doi.org/10.1093/bib/bbab128 (visited on 03/13/2022).

[6] Chuan Dong et al. "Comprehensive review of the identification of essential genes using computational methods: focusing on feature implementation and assessment". In: *Briefings in Bioinformatics* 21.1 (Jan. 2020), pp. 171–181. ISSN: 1477-4054. DOI: 10.1093/bib/bby116. URL: https://doi.org/10.1093/bib/bby116 (visited on 03/13/2022).

[7] Md Abid Hasan and Stefano Lonardi. "DeeplyEssential: a deep neural network for predicting essential genes in microbes". In: *BMC Bioinformatics* 21.14 (Sept. 2020), p. 367. ISSN: 1471-2105. DOI: 10.1186/s12859-020-03688-y. URL: https://doi.org/10.1186/s12859-020-03688-y (visited on 03/13/2022).

[8] Kai Song, Tuopong Tong, and Fang Wu. "Predicting essential genes in prokaryotic genomes using a linear method: ZUPLS". In: *Integrative Biology* 6.4 (Apr. 2014), pp. 460–469. ISSN: 1757-9708. DOI: 10.1039/c3ib40241j. URL: https://doi.org/10.1039/c3ib40241j (visited on 05/08/2022).

[9] Kitiporn Plaimas, Roland Eils, and Rainer König. "Identifying essential genes in bacterial metabolic networks with machine learning methods". In: *BMC Systems Biology* 4.1 (May 2010), p. 56. ISSN: 1752-0509. DOI: 10.1186/1752-0509-4-56. URL: https://doi.org/10.1186/1752-0509-4-56 (visited on 05/08/2022).

[10] Sanathoi Gurumayum et al. "OGEE v3: Online GEne Essentiality database with increased coverage of organisms and human cell lines". In: *Nucleic Acids Research* 49.D1 (Jan. 2021), pp. D998–D1003. ISSN: 0305-1048. DOI: 10.1093/nar/gkaa884. URL: https://doi.org/10.1093/nar/gkaa884 (visited on 04/07/2022).

[11] Eduardo P. C. Rocha and Antoine Danchin. "Gene essentiality determines chromosome organisation in bacteria". In: *Nucleic Acids Research* 31.22 (Nov. 2003), pp. 6570–6577. ISSN: 0305-1048. DOI: 10.1093/nar/gkg859. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC275555/ (visited on 05/08/2022).

[12] Tao Liu, Hao Luo, and Feng Gao. "Position preference of essential genes in prokaryotic operons". en. In: *PLOS ONE* 16.4 (Apr. 2021). Publisher: Public Library of Science, e0250380. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0250380. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0250380 (visited on 05/07/2022).

[13] Blanca Taboada et al. "Operon-mapper: a web server for precise operon identification in bacterial and archaeal genomes". In: *Bioinformatics* 34.23 (Dec. 2018), pp. 4118–4120. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bty496. URL: https://doi.org/10.1093/bioinformatics/bty496 (visited on 05/07/2022).

[14] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *arXiv:1912.01703 [cs, stat]* (Dec. 2019). arXiv: 1912.01703. URL: http://arxiv.org/abs/1912.01703 (visited on 05/07/2022).

[15] *Optuna: A hyperparameter optimization framework — Optuna 2.10.0 documentation*. URL: https://optuna.readthedocs.io/en/stable/index.html (visited on 05/08/2022).

[16] David M. Emms and Steven Kelly. "OrthoFinder: phylogenetic orthology inference for comparative genomics". In: *Genome Biology* 20.1 (Nov. 2019), p. 238. ISSN: 1474-760X. DOI: 10.1186/s13059-019-1832-y. URL: https://doi.org/10.1186/s13059-019-1832-y (visited on 03/13/2022).

[17] *BCEWithLogitsLoss — PyTorch 1.11.0 documentation*. URL: https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html (visited on 05/08/2022).

[18] Xiao Liu et al. "Selection of key sequence-based features for prediction of essential genes in 31 diverse bacterial species". en. In: *PLOS ONE* 12.3 (Mar. 2017). Publisher: Public Library of Science, e0174638. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0174638. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0174638 (visited on 04/03/2022).

[19] Jingyuan Deng et al. "Investigating the predictability of essential genes across distantly related organisms using an integrative approach". In: *Nucleic Acids Research* 39.3 (Feb. 2011), pp. 795–807. ISSN: 0305-1048. DOI: 10.1093/nar/gkq784. URL: https://doi.org/10.1093/nar/gkq784 (visited on 04/03/2022).

[20] Karthik Azhagesan, Balaraman Ravindran, and Karthik Raman. "Network-based features enable prediction of essential genes across diverse organisms". en. In: *PLOS ONE* 13.12 (Dec. 2018). Publisher: Public Library of Science, e0208722. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0208722. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0208722 (visited on 05/08/2022).

[21] Clyde A. Hutchison et al. "Polar Effects of Transposon Insertion into a Minimal Bacterial Genome". In: *Journal of Bacteriology* 201.19 (2019). Publisher: American Society for Microbiology, e00185–19. DOI: 10.1128/JB.00185-19. URL: https://journals.asm.org/doi/10.1128/JB.00185-19 (visited on 05/08/2022).

A Appendix

Table S1: Summary of training, validation and test sets.

Subset	Training	Validation	Test
Essentiality			
Essential genes	16,000	1,965	4,310
Non-essential genes	49,164	6,556	14,629
Gram			
Gram-negative	53,071	6,830	15,277
Gram-positive	12,093	1,691	3,662
Orthology			
Orthogroups	6,307	902	1,803

Oversampling was performed to balance essentiality in the training set.

Table S2: Hyperparameters used in the neural network.

Parameters	Range	Selected
Number of layers	[2 - 8]	3
Number of nodes	[16, 32, 64, 128, 512, 1024, 2048]	2048, 2048, 2048
Learning rate	[0.00001 - 0.01]	0.00087
Dropout rate	[0.0 - 0.5]	0.3, 0.1, 0.4
Epoch	-	100
Optimizer	[Adam, RMSprop, SGD]	Adam

Parameters optimized for model AUC using Optuna over 50 trials.

Table S3: Summary statistics of the data set used in this study.

	Essential	**Non-Essential**
N	22,275 (24.05)	70,349 (75.95)
Taxonomy		
Gram-negative		
Alphaproteobacteria	3,386 (15.20)	13,822 (19.65)
Betaproteobacteria	3,294 (14.79)	4,334 (6.16)
Gammaproteobacteria	8,011 (35.96)	33,452 (47.55)
Deltaproteobacteria	159 (0.71)	0 (0.0)
Epsilonproteobacteria	981 (4.40)	2,333 (3.32)
Bacteroidetes	2,626 (11.79)	0 (0.0)
Tenericutes	339 (1.52)	88 (0.13)
Cyanobacteria	689 (3.09)	1,664 (2.37)
Gram-positive		
Actinobacteria	693 (3.11)	3,132 (4.45)
Firmicutes	2,097 (9.41)	11,524 (16.38)
Gene features		
Gene length	921 (585-1,305)	822 (510-1,209)
Leading strand	10,985 (49.32)	35,020 (49.78)
Operon position		
Not in an operon	5,569 (25.00)	24,962 (35.48)
Front position	4,411 (19.80)	15,652 (22.25)
Other position	12,295 (55.20)	29,735 (42.27)

Counts and percentages are reported for all variables apart from gene length, where median and interquartile range are shown.

Table S4: ROC-AUC scores of randomized trials.

Random trial	Primary	2	3	4	5	**Mean**	**Std Dev**
Reference model	0.7881	0.7872	0.8006	0.8030	0.8105	0.7979	0.0100
No oversampling	0.8035	0.7993	0.7816	0.7977	0.8268	0.8018	0.0163
No orthogroups	0.8200	0.8030	0.8161	0.8145	0.8145	0.8136	0.0063

Each trial represents different randomizations of training and test splits and different optimized hyperparameters.

Table S5: Most common gene functions of essential genes.

Function	**Fraction of genes in this category(%)**
Correctly classified (N=2,135)	
Pyruvate/2-oxoglutarate dehydrogenase complex	2.03
Ribosomal protein L2	1.93
Cell division GTPase	1.87
Ribosomal protein S3	1.82
Molecular chaperone	1.72
Cell division protein FtsI	1.72
Dehydrogenases with different specificities	1.72
Ribosomal protein S7	1.67
Translation elongation factors (GTPases)	1.61
ABC-type transport system	1.51
Ribosomal protein S11	1.51
2-oxoglutarate dehydrogenase complex	1.46
Ribosomal protein S1	1.35
Ribosomal protein L13	1.25
Glucosamine 6-phosphate synthetase	1.25
Incorrectly classified (N=2,175)	
Uncharacterized protein conserved in bacteria	4.26
Dehydrogenases with different specificities	1.80
Uncharacterized conserved protein	1.75
Dephospho-CoA kinase	1.26
ABC-type transport system	1.26
Cell division protein FtsI	1.26
Predicted ATPase or kinase	1.15
Preprotein translocase subunit YidC	1.09
ABC-type multidrug transport system	1.09
Uncharacterized membrane protein	0.98
Methionine aminopeptidase	0.93
Inorganic pyrophosphatase	0.87
Acyl carrier protein	0.87
4-diphosphocytidyl-2C-methyl-D-erythritol 2-phosphate synthase	0.87
Predicted ABC-type transport system	0.87

Table S6: Most common gene functions of non-essential genes.

Function	Fraction of genes in this category (%)
Correctly classified (N=13,062)	
Uncharacterized protein conserved in bacteria	6.26
Dehydrogenases with different specificities	3.03
Uncharacterized conserved protein	2.49
NAD/FAD-utilizing enzyme	1.53
Predicted membrane protein	1.35
Transcriptional regulator	1.04
Acyl-CoA dehydrogenases	1.01
Predicted transcriptional regulators	0.91
Permeases of the drug/metabolite transporter	0.85
ABC-type multidrug transport system	0.77
Cation/multidrug efflux pump	0.76
ABC-type sugar transport system, permease	0.74
Predicted oxidoreductases	0.71
Glycosyltransferases involved in cell wall synthesis	0.70
Cation transport ATPase	0.67
Incorrectly classified (N=1,567)	
Uncharacterized protein conserved in bacteria	2.92
Dehydrogenases with different specificities	2.41
ATPases with chaperone activity	2.05
Acyl-CoA dehydrogenases	1.90
Uncharacterized conserved protein	1.31
Ribosomal protein L9	1.24
Transcriptional accessory protein	1.17
ABC-type sugar transport systems	1.10
NAD/FAD-utilizing enzyme	1.02
Transcriptional regulator	0.95
Membrane protease subunits	0.95
Pyruvate/2-oxoglutarate dehydrogenase complex	0.88
Peptide chain release factor RF-3	0.88
ABC-type multidrug transport system	0.80
ABC-type polar amino acid transport system	0.80