ProtiGeno: a prokaryotic short gene finder using protein language models

Tony Tu [1]   Gautham Krishna [2]   Amirali Aghazadeh [1]

Abstract

Prokaryotic gene prediction plays an important role in understanding the biology of organisms and their function with applications in medicine and biotechnology. Although the current gene finders are highly sensitive in finding long genes, their sensitivity decreases noticeably in finding shorter genes (<180 nts). The culprit is insufficient annotated gene data to identify distinguishing features in short open reading frames (ORFs). We develop a deep learning-based method called ProtiGeno, specifically targeting short prokaryotic genes using a protein language model trained on millions of evolved proteins. In systematic large-scale experiments on 4,288 prokaryotic genomes, we demonstrate that ProtiGeno predicts short coding and noncoding genes with higher accuracy and recall than the current state-of-the-art gene finders. We discuss the predictive features of ProtiGeno and possible limitations by visualizing the three-dimensional structure of the predicted short genes. Data, codes, and models are available at https://github.com/tonytu16/protigeno.

1. Introduction

Advances in low-cost and high-throughput sequencing technologies have led to a deluge of genomic data from prokaryotic organisms. Automatic gene annotation tools are crucial for describing the biological functions of genes and ultimately understanding these organisms. Most of the current prokaryotic gene finders belong to one of the families of Glimmer (Delcher et al., 1999), GeneMarkS (Besemer et al., 2001), and Prodigal (Hyatt et al., 2010), all of which are based on Markov models (Rabiner & Juang, 1986) equipped with various biologically-inspired heuristics (e.g., GC content and the Shine–Dalgarno sequence). While these methods are highly sensitive for finding long protein-coding genes, their sensitivity (recall rate) drops drastically for finding short coding genes (<180 nts) (see Figure 5). One reason behind this performance drop is the lack of sufficiently rich genomic features that could be gleaned from the Markov models (Zhang et al., 2017). Departing from the conventional Markov-based methods, *ab inito* machine learning methods have been recently developed for gene finding. Balrog (Sommer & Salzberg, 2021), for example, trains a temporal convolutional deep neural network on a diverse collection of high-throughput prokaryotic organisms. These supervised models capture a richer class of genomic signatures for gene prediction than Markov models. They also alleviate the need for retraining for each target genome, allowing for a universal gene prediction model. However, similar to Markov-based models, these machine learning models are primarily developed for annotating long genes. As a result, their sensitivity for short genes remains low due to a lack of annotated short gene sequences in the Ref-Seq database (Nielsen & Krogh, 2005; Skovgaard et al., 2001). This lack of information from short genes renders the machine learning methods ill-equipped to identify genes belonging to many underrepresented species, resulting in missing gene types (Warren et al., 2010). It stands to reason that we need to borrow statistical power from independent, unsupervised biological data resources to compensate for the scarcity of short genes and complex distinguishing features within them to develop highly sensitive models for gene annotation.

Parallel to the recent advances in gene prediction, Transformer-based deep learning models (Rao et al., 2020) trained on the amino acid sequences of the proteins evolved throughout millions of years of evolution and stored in protein databases such as UniProt (Consortium, 2015) have revolutionized protein structure prediction (Jumper et al., 2021; Sapoval et al., 2022). In particular, masked protein language models have demonstrated success in fast and accurate protein structure prediction without explicitly training to learn the rules of protein folding (Lin et al., 2022). This power of protein language models in predicting the secondary and tertiary structure and biological functions of many proteins motivates the main intellectual motivation behind our approach to gene prediction. Herein, we hypothesize that it is feasible to build a sensitive and universal short gene finder by training a classifier on the output em-

---

[1] School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta GA, USA [2] School of Biological Sciences, Georgia Institute of Technology, Atlanta GA, USA. Correspondence to: Amirali Aghazadeh <amiralia@gatech.edu>.
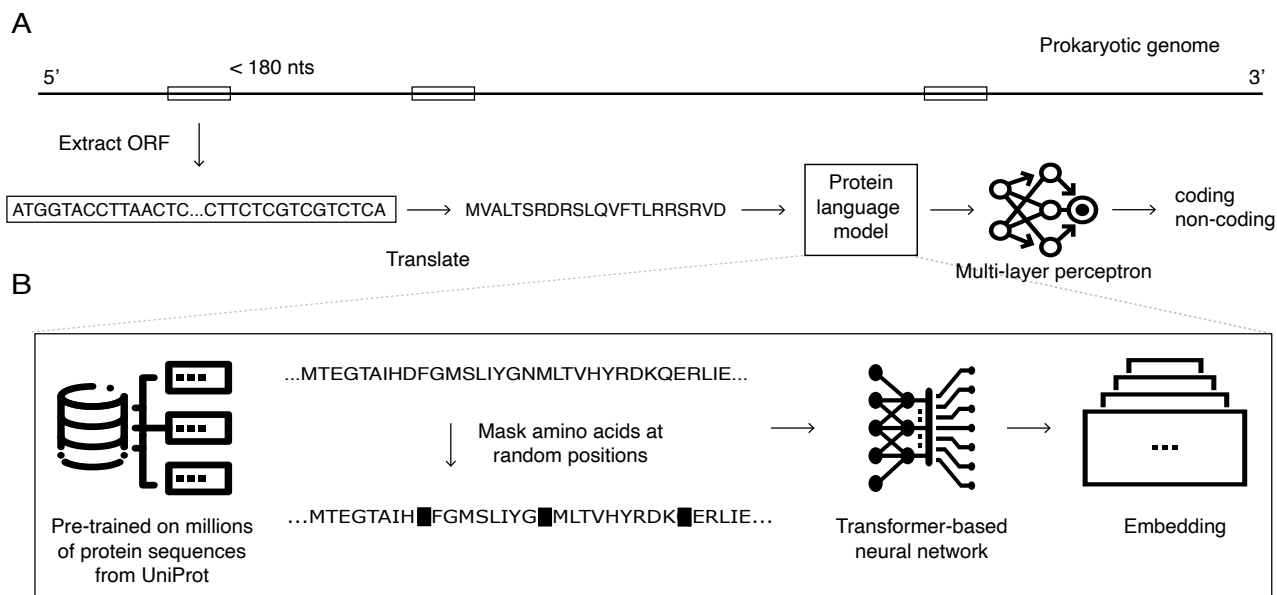
Figure 1. A) Schematic of our ProtiGeno method. ProtiGeno extracts ORFs from a prokaryotic genome and translates them to amino acid sequences. It then finds a 1280-dimensional embedding vector using a protein language model. ProtiGeno passes the embedding vector through a multi-layer perceptron and obtains a binary output indicating whether the input sequence is protein coding or noncoding. B) Overview of protein language model. We use a protein language model called ESM-1b from Meta AI (Rives et al., 2021). ESM-1b is trained on the UR50/S dataset (Suzek et al., 2015) from the UniProt database using a variant of the masked language modeling technique on protein sequences. The training process involves randomly masking a certain percentage of amino acids for each input sequence and asking the model to predict the missing amino acids. This enables the model to learn representations that capture meaningful biological properties of the amino acid sequence, such as their secondary and tertiary structures.

bedding vector of a protein language model once input with the translated sequence of a short open reading frame. Having this goal in mind, we developed ProtiGeno, a **Prot**ein language model-based method for short **Gen**e predicti**on** (Figure 1). ProtiGeno can be applied, without any further retraining, to find short genes in any prokaryotic species. We systematically compared ProtiGeno with baseline gene finders on short coding and noncoding regions from 4,288 prokaryotic genomes. ProtiGeno consistently outperforms the baseline gene finders in terms of accuracy, recall, and F1 score with a lower precision rate (Table 1). Our main contributions are:

- We develop a gene parsing pipeline that extracts protein sequences for short coding and noncoding regions.

- We collect a database of ORFs from 4,288 genomes, consisting of 145,232 short coding regions and 3,465,408 noncoding regions extracted using our pipeline.

- We develop ProtiGeno, a deep learning-based short gene finder using protein language models. ProtiGeno opens up new opportunities for discovering previously uncharacterized short genes and can expand our knowledge of prokaryotic organisms.

2. Data collection and model architecture

Short prokaryotic gene data. We collected 4,288 prokaryotic genomes from the NCBI GenBank database and extracted all short ORFs of size between 87 and 177 nucleotides (nts) annotated from protein homology or RefSeq (Pruitt et al., 2007). This data comprised 145,232 short protein-coding regions and 3,465,408 short noncoding regions (see Appendix I for more details regarding data parsing and filtering). The histogram of the number of ORFs across the genomes (Figure 6) shows that, on average, each prokaryotic genome has 34 coding and 808 noncoding regions, ranging from genomes having as low as two coding and no noncoding regions to genomes having as high as 436 coding and 6,227 noncoding regions. The data is highly skewed towards noncoding regions (1:24 ratio). In order to train a balanced-class classifier in ProtiGeno, we sampled the same number of noncoding regions as the number of coding regions. Also, to control for the ORF length as a potential spurious predictive feature, we sampled the noncoding regions from the same length distribution as the coding regions per genome (see Appendix B). We also removed all duplicate coding and noncoding regions to avoid bias and overfitting, resulting in 128,578 unique short coding and 143,325 noncoding regions.

Method	Accuracy	Precision	Recall	F1
ProtiGeno	**0.933** ±0.009	0.930 ±0.019	**0.928** ±0.012	**0.929** ±0.009
GeneMarkS (PGP)	0.910 ±0.006	**1.0** ±0.000	0.809 ±0.012	0.894 ±0.007
GeneMarkS (GP)	0.923 ±0.005	**1.0** ±0.000	0.837 ±0.011	0.911 ±0.007
Prodigal (PGP)	0.877 ±0.006	**1.0** ±0.000	0.741 ±0.012	0.851 ±0.008
Prodigal (GP)	0.892 ±0.005	**1.0** ±0.000	0.772 ±0.011	0.872 ±0.007
FragGeneScanRS (PGP)	0.790 ±0.006	**1.0** ±0.000	0.555 ±0.011	0.714 ±0.009
FragGeneScanRS (GP)	0.813 ±0.005	**1.0** ±0.000	0.604 ±0.010	0.753 ±0.008
Random	0.499 ±0.003	0.472 ±0.002	0.501 ±0.004	0.486 ±0.002

Table 1. Comparison of ProtiGeno with baseline gene finders. We randomly split the genomes into ten folds. We evaluate the baselines based on two measures. PGP stands for Precise Gene Prediction and counts true positives when the 3' and 5' ends match. GP stands for Gene Prediction and counts true positives when the 3' end matches with a possible misplacement of the 5' end. We report the mean and standard error of the mean (SEM) of the test performance across ten folds. Due to high computational cost, we did not include Balrog results as each genome takes approximately 15 minutes to process, which amounts to 45 days on all 4,288 genomes

Protein language model and neural network classifer. We utilize ESM-1b (Rives et al., 2021), a masked protein language model, to embed the translated open reading reading frames. ESM-1b is a 33-layer self-supervised transformer model with 650M parameters trained on the UniRef50 dataset (Suzek et al., 2015). We used ESM-1b to embed short genes into embedding vectors of size 1280. We then trained a 7-layer fully connected deep neural network to capture the non-linearities in the protein embeddings (see Appendix E for ablation studies and details on the network).

3. Results

Gene prediction performance across all prokaryotic genomes. To systematically test the performance of ProtiGeno on all the 4,288 genomes, we do a 10-fold cross-validation, where we use 90% of the genomes for training and the other 10% for evaluation. This way of splitting the data avoids data leakage across the genomes. In Table 1, we report the mean and standard error of the mean (SEM) of test accuracy, precision, recall, and F1 score across folds for ProtiGeno, GeneMarkS (Besemer et al., 2001), Prodigal (Hyatt et al., 2010) and FragGeneScanRS (Van der Jeugt et al., 2022) (see Appendix D for detailed results). All baseline methods are run with default settings and no hyperparameter tuning is involved. The performance metrics were averaged over all the ORFs in the genomes of the test fold. For baselines, we report results using two measures. PGP stands for Precise Gene Prediction and counts true positives when the 3' and 5' ends match. GP stands for Gene Prediction and counts true positives when the 3' end matches with a possible misplacement of the 5' end. Table 1 shows that ProtiGeno performs consistently better than the baselines in terms of accuracy, recall, and the F1 score. The higher recall rate comes at the cost of slightly lower precision. In addition to the 10-fold cross validation, we additionally report the per-genome evaluation in appendix J, where we report the percentage of genomes on which ProtiGeno outperforms

the baseline methods. ProtiGeno outperforms all baseline methods on the majority of the genomes at accuracy, recall, and F-1 score, showing a similar trend as table 1, where ProtiGeno achieves a higher recall rate at the cost of a lower precision. We analyze the false positives of ProtiGeno in more detail in the next subsections.

Gene prediction performance on representative genomes. We tested ProtiGeno on a real-world gene prediction task. We selected five organisms identified as model organisms (MOs) in Dimonaco et al. (2022) due to their diversity in genome size, GC content, and their quality and comprehensiveness of genome assembly and annotation. These organisms include (1) *Bacillus subtilis subsp. subtilis str. 168* (2) *Caulobacter vibrioides NA1000* (3) *Escherichia coli str. K-12 substr. MG1655* (4) *Mycoplasmoides genitalium G37*, and (5) *Staphylococcus aureus subsp. aureus NCTC 8325*. We report the precision and recall for ProtiGeno against four baselines, including GeneMarkS (Besemer et al., 2001), Prodigal (Hyatt et al., 2010), FragGeneScanRS (Van der Jeugt et al., 2022), and Balrog (Sommer & Salzberg, 2021) in Table 2. ProtiGeno achieves a higher recall rate than the baselines with a significant margin at a cost of a drop in precision (higher overal F1 score).

Explaining ProtiGeno. To obtain further details into ProtiGeno predictions, we performed a structural analysis using ESMFold (see Figure 2). We compared the distribution of alpha helices and beta sheets between coding and noncoding regions, hypothesizing that the distribution of one is proportional to the probability of being a gene. We sampled four organisms, two performing well in ProtiGeno (i.e., *B. choerinum, and M. bovirhinis*) and two performing poorly in ProtiGeno (i.e., *F. frigiditurris, and T. geofontis OPF15*). We analyzed 70 coding and noncoding regions. Each data point in Figure 2C represents the fraction of alpha helix residue and Figure 2D represents the fraction of beta sheet residue. Figures show the distribution of beta sheets is significantly different between the coding and noncoding

Genomes	ProtiGeno		GeneMarkS		Prodigal		FGS		Balrog	
	Prec.	Recall	Prec.	Recall	Prec.	Recall	Prec.	Recall	Prec.	Recall
B. Subtilis	0.841	**0.736**	**1.0**	0.601	**1.0**	0.615	**1.0**	0.490	**1.0**	0.543
C. Vibrioides	0.866	**0.587**	**1.0**	0.430	**1.0**	0.446	**1.0**	0.273	**1.0**	0.149
E. Coli	0.772	**0.661**	**1.0**	0.426	**1.0**	0.566	**1.0**	0.295	**1.0**	0.598
M. Genitalium	0.889	**1.0**	**1.0**	0.375	**1.0**	0.625	**1.0**	0.500	**1.0**	0.625
S. Aureus	0.909	**0.310**	**1.0**	0.243	**1.0**	0.243	**1.0**	0.235	**1.0**	0.119

Table 2. Evaluation results of 5 representative organisms on ProtiGeno against four baseline gene annotators using Precise Gene Prediction (PGP): ProtiGeno consistently achieves better recall but lower precision across all five genomes, illustrating a tradeoff between ProtiGeno and the baseline gene annotators: optimizing for higher recall comes at the cost of a lower precision. ProtiGeno outputs more positive predictions, which reduces the possibility of missing a short coding region but also results in a higher false positive rate, thereby lowering precision. However, this could also potentially help us identify uncharacterized new genes.
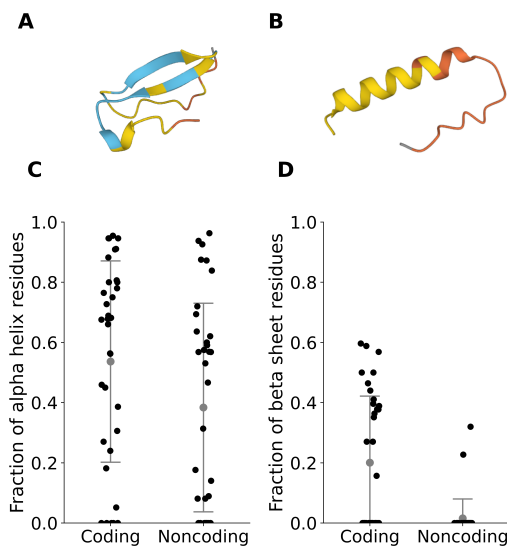

Figure 2. Structural analysis of 70 frames predicted by ProtiGeno using ESMFold. A) Plot A shows the 3D structure of a short protein coding region from sample organism *F. frigiditurris*, predicted by ESMFold. B) Plot B shows the 3D stucture of a short protein noncoding region from sample organism *F. frigiditurris* predicted by ESMFold. C) We compare the fraction of residues predicted as alpha helices for coding and noncoding regions across four organisms, two performing well in ProtiGeno (i.e., *B. choerinum, and M. bovirhinis*) and two performing poorly in ProtiGeno (i.e., *F. frigiditurris, and T. geofontis OPF15*). Two-sided t-test shows no significant difference between the means ($p > 0.06$). D) We compare the fraction of residues predicted as beta sheets. Two-sided t-test shows significant differences between the means ($p < 10^{-5}$). Our analysis demonstrate that fraction of residues in beta sheets is one of the explainable predictive feature in ProtiGeno.

regions ($p < 10^{-5}$). However, the difference is not significant for alpha helices ($p > 0.06$). The analysis demonstrates that the fraction of residues in beta sheets is an explainable predictive feature in ProtiGeno. Simultaneously, alpha helix can be a potential explainable cause of the false positives.

4. Discussion

Tables 1 and 2 show that ProtiGeno has a lower precision rate compared to the baseline methods, indicating higher false positives. We identify two factors that could potentially cause this phenomenon: (1) alpha helices confusing coding and noncoding regions and (2) the discrepancies between AlphaFold2 and ESMFold secondary structure predictions. Figure 2 demonstrates the possibility of ProtiGeno using secondary structures as a distinguishing feature. Although the fraction of residues predicted as beta sheets by ESM-Fold are significantly different between coding regions and noncoding regions and can serve as a distinguishing feature, the same claim does not hold true for residues predicted as alpha helices by ESMFold (see Figure 2). Therefore, alpha helices can potentially cause false positives in ProtiGeno, resulting in a lower precision rate. More analysis is needed to fully understand the structural features in ProtiGeno.

In addition, our comparisons of the predicted secondary structures of 84 false positive samples from 11 genomes using AlphaFold2 and ESMFold demonstrate that the fraction of predicted secondary structures is correlated with $R^2 = 0.74$ (Appendix 7). However, we still observe discrepancies between the predicted 3D structures from AlphaFold2 and ESMFold, which could potentially lead to ProtiGeno misclassifying noncoding regions as coding regions. Nonetheless, while AlphaFold2 is generally the more accurate protein structure predictor (Bertoline et al., 2023), the family of ESMFold methods (e.g., ESM-1b) might be preferred for large-scale gene finding tasks due to its massive computational advantages (60 times faster), offering opportunities to identify uncharacterized short genes and novel functions in a more efficient manner.

5. Acknowledgements

References

Bertoline, L. M., Lima, A. N., Krieger, J. E., and Teixeira, S. K. Before and after alphafold2: An overview of protein structure prediction. *Frontiers in Bioinformatics*, 3, 2023.

Besemer, J., Lomsadze, A., and Borodovsky, M. Gene-MarkS: a self-training method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions. *Nucleic Acids Research*, 29 (12):2607–2618, 2001.

Consortium, U. Uniprot: a hub for protein information. *Nucleic Acids Research*, 43(D1):D204–D212, 2015.

Dallago, C., Mou, J., Johnston, K. E., Wittmann, B. J., Bhattacharya, N., Goldman, S., Madani, A., and Yang, K. K. Flip: Benchmark tasks in fitness landscape inference for proteins. *bioRxiv*, pp. 2021–11, 2021.

Delcher, A. L., Harmon, D., Kasif, S., White, O., and Salzberg, S. L. Improved microbial gene identification with GLIMMER. *Nucleic Acids Research*, 27(23):4636–4641, 1999.

Dimonaco, N. J., Aubrey, W., Kenobi, K., Clare, A., and Creevey, C. J. No one tool to rule them all: prokaryotic gene prediction tool annotations are highly dependent on the organism of study. *Bioinformatics*, 38(5):1198–1207, 2022.

Hyatt, D., Chen, G.-L., LoCascio, P. F., Land, M. L., Larimer, F. W., and Hauser, L. J. Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC Bioinformatics*, 11(1):1–11, 2010.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv*, 2022.

Nielsen, P. and Krogh, A. Large-scale prokaryotic gene prediction and comparison to genome annotation. *Bioinformatics*, 21(24):4322–4329, 2005.

Pruitt, K. D., Tatusova, T., and Maglott, D. R. Ncbi reference sequences (refseq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Research*, 35(suppl_1):D61–D65, 2007.

Rabiner, L. and Juang, B. An introduction to hidden markov models. *IEEE ASSP Magazine*, 3(1):4–16, 1986.

Rao, R., Meier, J., Sercu, T., Ovchinnikov, S., and Rives, A. Transformer protein language models are unsupervised structure learners. *Biorxiv*, pp. 2020–12, 2020.

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021.

Sapoval, N., Aghazadeh, A., Nute, M. G., Antunes, D. A., Balaji, A., Baraniuk, R., Barberan, C., Dannenfelser, R., Dun, C., Edrisi, M., et al. Current progress and open challenges for applying deep learning across the biosciences. *Nature Communications*, 13(1):1728, 2022.

Skovgaard, M., Jensen, L. J., Brunak, S., Ussery, D., and Krogh, A. On the total number of genes and their length distribution in complete microbial genomes. *TRENDS in Genetics*, 17(8):425–428, 2001.

Sommer, M. J. and Salzberg, S. L. Balrog: A universal protein model for prokaryotic gene prediction. *PLoS Computational Biology*, 17(2):e1008727, 2021.

Suzek, B. E., Wang, Y., Huang, H., McGarvey, P. B., Wu, C. H., and Consortium, U. Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, 2015.

Van der Jeugt, F., Dawyndt, P., and Mesuere, B. Fraggenescanrs: faster gene prediction for short reads. *BMC Bioinformatics*, 23(1):198, 2022.

Warren, A. S., Archuleta, J., Feng, W.-c., and Setubal, J. C. Missing genes in the annotation of prokaryotic genomes. *BMC Bioinformatics*, 11(1):1–12, 2010.

Zhang, S.-W., Jin, X.-Y., and Zhang, T. Gene prediction in metagenomic fragments with deep learning. *BioMed Research International*, 2017, 2017.

A. Coding region vs. all noncoding region lengths distribution before sampling & filtering

We present the length distribution of short coding regions and short noncoding regions before sampling and filtering. As we can see in Figure 3, the length distribution for short coding regions are skewed to the left, most of which are concentrated at 177 nts, while the length distribution for short noncoding regions are skewed to the right, most of which are concentrated at 77 nts.
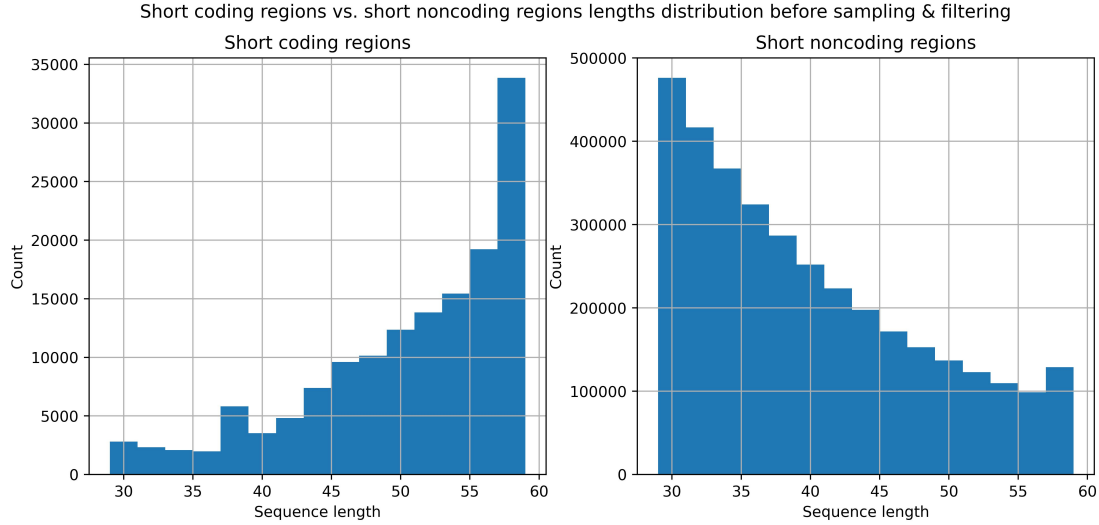

Figure 3. We plot the length distribution for the coding regions (in number of amino acids) as well as all noncoding regions before sampling. All short coding regions and noncoding regions are within 87 and 177 nts. The length distribution for short coding regions are skewed to the left and the length distribution for noncoding regions are skewed to the right.

B. Noncoding region sampling algorithm

To avoid bias against length and overfitting in the gene classifier, we designed a sampling algorithm which samples noncoding regions from the same length distribution as the coding regions per genome.

Algorithm 1 Noncoding Region Sampling Algorithm

1: **procedure** SAMPLENONCODINGSEQS(genome, CodingSeqs, NoncodingSeqs)
2:     SampledNoncodingSeqs ← []
3:     CodingSeqFrequencyDict ← Create frequency Dict for coding region lengths
4:     NoncodingSeqFrequencyDict ← Create frequency Dict for noncoding regions lengths
5:     carry ← 0
6:     **for** key in CodingSeqFrequencyDict keys sorted in reverse order **do**
7:         NumCodingSeqs ← CodingSeqFrequencyDict[key]   ▷ Coding Frequency Dict value for this key
8:         NumNoncodingSeqs ← NoncodingSeqFrequencyDict[key]   ▷ Noncoding Frequency Dict value for this key
9:         **if** NumCodingSeqs + carry ≤ NumNoncodingSeqs **then**
10:             Noncoding samples ← Sample NumCodingSeqs + carry samples from NoncodingSeqs
11:             carry ← 0 ▷ If we have more noncoding samples available, we randomly sample NumCodingSeqs + carry
    many samples from NoncodingSeqs with length equal to key and assign carry to 0
12:         **if** NumCodingSeqs + carry > NumNoncodingSeqs **then**
13:             Noncoding samples ← Sample all NumNoncodingSeqs number of noncoding regions
14:             carry ← NumCodingSeqs + carry - NumNoncodingSeqs   ▷ If NumCodingSeqs + carry is greater than the
    number of noncoding regions available, we simply take all NumNoncodingSeqs number of NoncodingSeqs with length
    equal to key and assign carry to the difference between the two numbers
15:         SampledNoncodingSeqs ← SampledNoncodingSeqs + Noncoding samples
16:         **return** CodingSeqs, SampledNoncodingSeqs

C. Coding region vs. noncoding region lengths distribution after sampling & filtering

After applying the sampling algorithm from Appendix B, we also filtered out duplicate sequences to avoid bias in the dataset. As a result, we obtained a total of 128,578 unique coding regions and 143,325 unique noncoding regions, which we use for training with approximately the same length distribution, which we use to train the gene classifier.
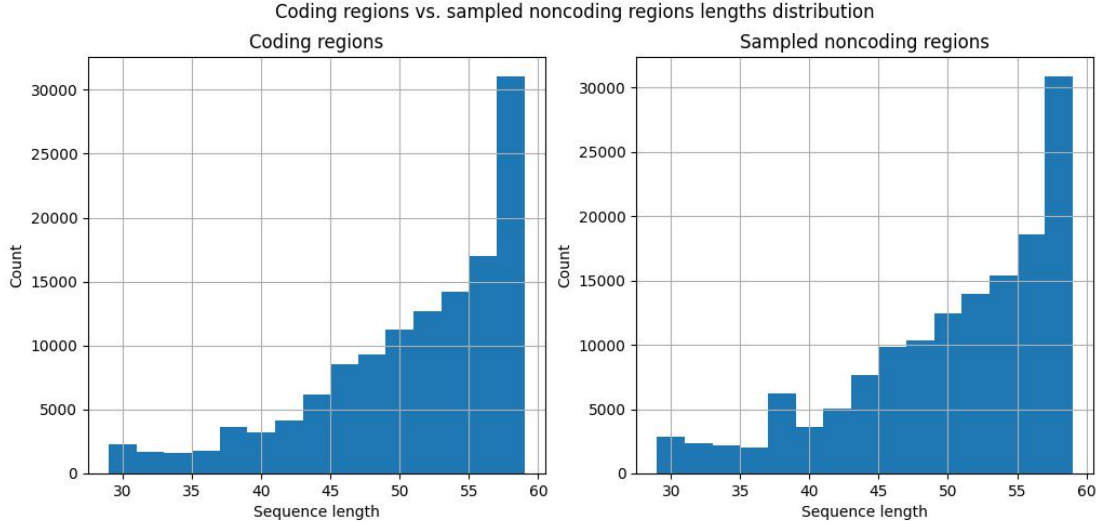

Figure 4. We plot the length distribution for the coding regions (in number of amino acids) as well as the sampled noncoding regions. Our sampling algorithm samples approximately the same number of noncoding regions as there are coding regions while ensuring similar length distribution to avoid bias against length in model training.

D. ProtiGeno 10-fold cross validation full results (aggregated)

As illustrated in Table 1, we conducted a 10-fold cross validation where we train on 90% of the genomes and evaluate on the rest of the 10% of the genomes. Here, we present the full 10-fold cross validation results. The (aggregated) results are calculated by concatenating all predictions and groundtruths in one fold. The mean value accuracy, precision, recall and F1 score take the average across all 10 folds.

Fold	Train seqs	Test seqs	Accuracy	Precision	Recall	F1
1	245544	26359	0.936	0.950	0.912	0.931
2	243767	28136	0.929	0.931	0.917	0.924
3	241748	30155	0.928	0.927	0.922	0.924
4	244110	27793	0.939	0.949	0.921	0.935
5	245386	26517	0.934	0.932	0.930	0.931
6	244359	27544	0.940	0.921	0.954	0.937
7	246605	25298	0.935	0.920	0.944	0.932
8	244532	27371	0.935	0.942	0.919	0.931
9	245299	26604	0.943	0.946	0.933	0.940
10	245777	26126	0.909	0.882	0.931	0.906
Mean (Aggregate)	-	-	0.933 $\pm$0.009	0.930 $\pm$0.019	0.928 $\pm$0.012	0.929 $\pm$0.009

Table 3. ProtiGeno 10-fold cross validation full result (aggregated). We report the aggregated mean and standard error of mean (SEM) for accuracy, precision, recall and F1 score across 10-folds. Results are aggregated, where accuracy, precision, recall and F1 scores are calculated by concatenating predictions and groundtruths across all genomes in a fold.

E. Gene classifier model architecture ablation study

We conducted an ablation study comparing the effect of model architecture on the performance of gene classifier. We trained gene classifiers with 3 model architectures: (1) 4-layer fully connected net, (2) 7-layer fully-connected net, and (3) 9-layer fully-connected layer. As shown in the table below, the 7-layer fully-connected net achieves the highest accuracy, recall, and F1 score. The 4-layer FC-Net has dimensions [1280, 32, 64, 1]. The 7-layer FC-Net has dimensions [1280, 32, 64, 128, 64, 32, 1]. The 9-layer FC-Net has dimensions [1280, 128, 256, 512, 256, 128, 64, 32, 1]. All models use xavier weight initialization and ReLU activation and are trained for 300 epochs using a learning rate of 0.01 and binary cross entropy loss.

Network Architecture	Accuracy	Precision	Recall	F1
4-layer fully-connected net	0.934	**0.952**	0.904	0.927
7-layer fully-connected net	**0.936**	0.950	**0.912**	**0.931**
9-layer fully-connected net	0.925	0.929	0.911	0.920

Table 4. ProtiGeno 10-fold cross validation result (aggregated): We conduct an ablation study to compare performance across 3 different neural network architectures. We evaluate these 4 models on the first fold and report the accuracy, precision, recall and F1 score.

F. Recall vs. Coding region sequence length on GeneMarkS

We plot the GeneMarkS PGP recall rates on coding regions of different lengths. The recall rate drops as the coding region length increases. These recall rates are different from those presented in Table 1, where we filtered out the incomplete genes, pseudogenes and duplicate genes.
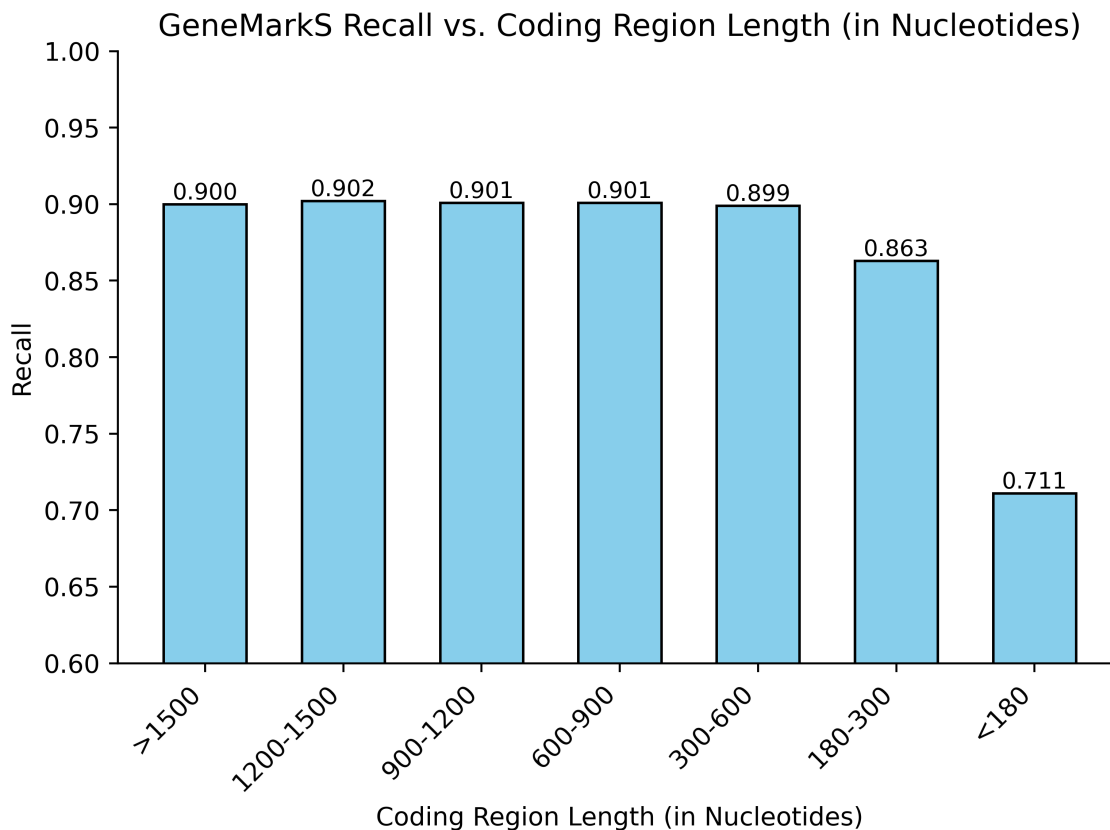

Figure 5. GeneMarkS PGP recall rates on coding regions of different lengths. There is a downward trend in recall as the coding region length decreases, demonstrating the need to improve gene prediction performance on shorter genes. These recall values are different from those in Table 1, where we filtered out incomplete genes, duplicate genes, and duplicate genes for training ProtiGeno.

G. Number of sequences vs. genome before sampling & filtering

We plot the number of coding regions and noncoding regions (sorted in descending order) per genome. As shown in the figure, the number of short noncoding regions exceeds the number of short coding regions by a large margin. On average, each prokaryotic genome has 34 coding and 808 noncoding regions, ranging from genomes having as low as 2 coding and 0 noncoding regions to genomes having as high as 436 coding and 6,227 noncoding regions. The data is highly skewed towards noncoding regions (1:24 ratio). To avoid class imbalance and thereby creating bias in our gene classifier, we apply a sampling algorithm described in Appendix B to downsample noncoding regions.

Number of Short Sequences vs. Genomes (sorted)
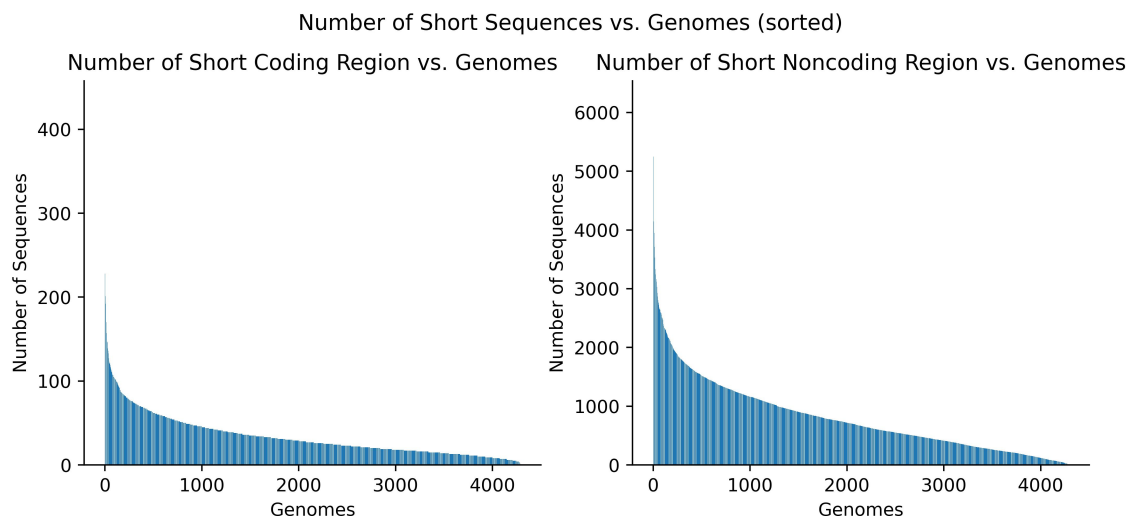
Figure 6. Number of short coding and noncoding regions vs. genome (sorted in descending order). We plot the Number of short (between 87 and 177 NT) coding and noncoding regions for every genome, sorted in descending order. As we see that the number of short noncoding regions exceeds the number of short coding regions by a large margin, marking a significant class imbalance in our dataset.

H. Fraction of secondary structure in ESMFold vs AlphaFold2.

We plot the fraction of secondary structure in ESMFold vs AlphaFold2 .This plot is a correlation of false positives for a sample size of 11 genomes which are *F. frigiditurris, T. geofontis OPF15, S. roseirectus, N. salina, M. sediminilitoris, B. wiedmannii, C. freundii, A. isosaccharinicus, N. moolapensis, A. marina, P. aryabhattai* containing 84 sequences in total. The scatter plot represents the fraction of secondary structures predicted by the models ESMFold and AlphaFold2. The regression line on the plot shows the goodness of fit test for the data. The data points are scattered among all regions of the plot and the $R^2 = 0.74$. The Welch's t-test performed for the ESMFold and AlphaFold2 groups (p>0.6) indicates that the null hypothesis fails to be rejected. Therefore no significant difference is observed between the predictions from ESMFold and AlphaFold2 for the false positive data.
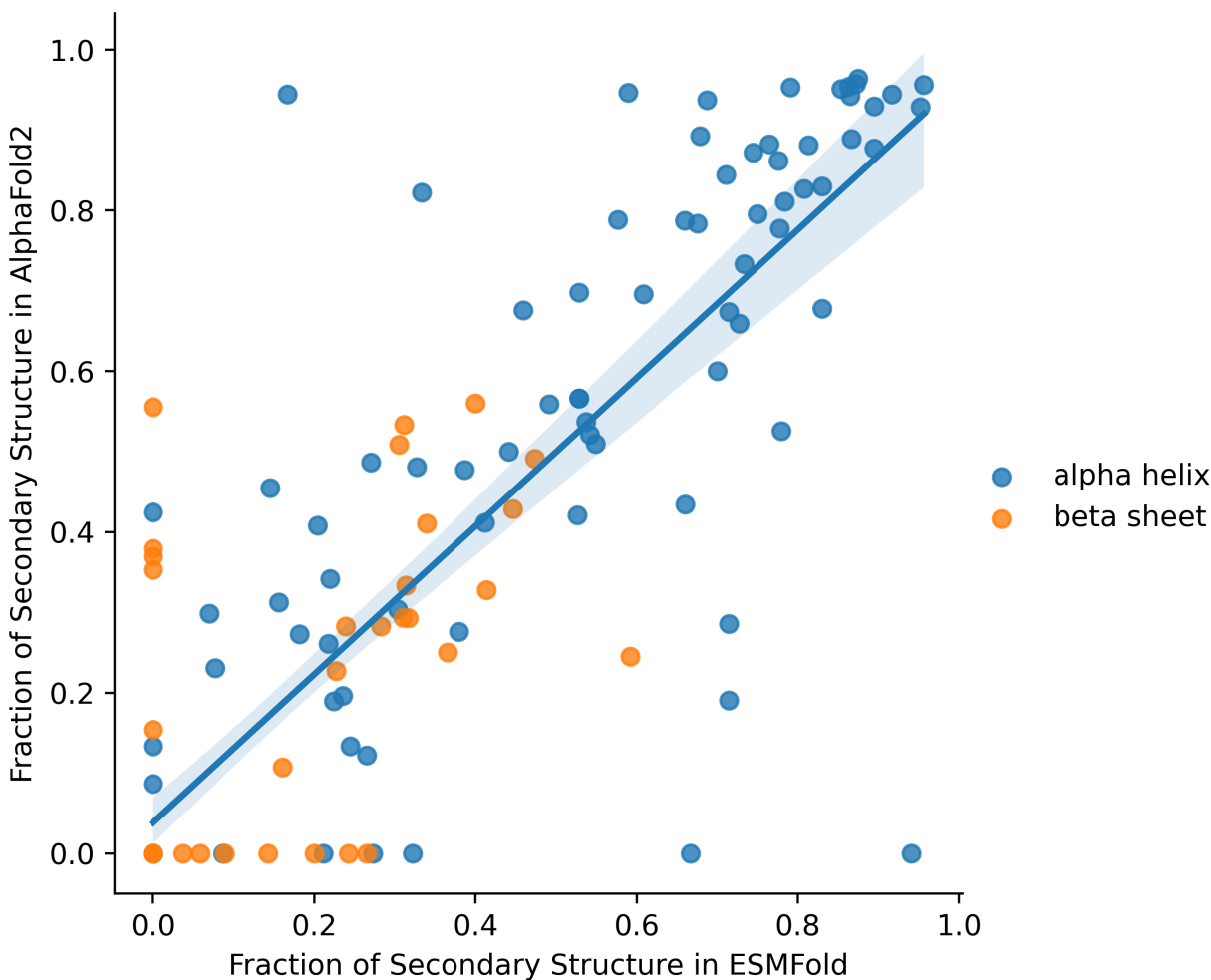

Figure 7. Correlation between secondary structure predictions by ESMFold and Aplhafold2 for 84 false positive samples from 11 genomes. The scatter plot shows the distribution of secondary structures generated by both methods. A Welch's t-test was performed to find the similarity between the predictions from the two models (p>0.6). The analysis fails to reject the null hypothesis, showing no significant difference between the fractions of alpha helices and beta sheets generated by ESMFold and AlphaFold2 for the false positive samples.

I. Experimental methods

Data collection and curation. We collected 4,288 prokaryotic genomes from the National Center for Biotechnology Information (NCBI) GenBank database by setting the filters to (1) Prokaryotes, (2) Bacteria, (3) Archaea, (4) RefSeq, (5) Representative, (6) Reference and (7) Complete. After obtaining the genomes and annotations, we selected short coding

regions with length between 87 nts and 177 nts annotated from protein homology or RefSeq (Pruitt et al., 2007), and excluded incomplete genes and pseudogenes. We also parse noncoding short ORFs in the intergenic regions from the same genomes, select noncoding regions between 87 nts and 177 nts, and create hypothetical translations. Due to the relative abundance of intergenic regions and noncoding regions compared to the number of short genes in genomes, we sample the same number of noncoding regions as there are short genes for each genome to avoid class imbalance, which could potentially cause our classifier to over-classify the majority group (noncoding regions) due to its increased prior probability. Furthermore, to prevent our classifier model from learning input sequence length as a feature, for each genome, we applied a sampling algorithm (see Appendix B) to sample the same number of noncoding regions as there are coding regions while ensuring approximately the same length distribution. In case of annotated gene duplication (two genes with the same sequence), only one protein is selected to avoid bias in the dataset. After applying our sampling algorithm and filtering out duplicates genes, we obtained 128,578 unique coding regions and 143,325 unique noncoding regions for training.

J. Protigeno per-genome evaluation and comparison to baseline methods

To gain more insights on how Protigeno performs on individual genomes, we provide a comprehensive table where each genome's performance was evaluated using the model trained on the remaining 90 percent of the genomes. This table collectively summarized the model performance for all genomes and is provided here in the github repository. Note that this table consists of 4,280 genomes instead of 4,288 because after filtering out duplicate sequences, 8 genomes do not have either coding regions or non-coding regions, which we have excluded from the evaluation. In table J, we report the percentage of genomes ProtiGeno outperforms against each baseline method in terms of accuracy, precision, recall and F-1 score. Similar to table 1, ProtiGeno outperforms the baseline methods in majority of genomes in terms of accuracy, recall and F-1 score.

Baseline Method	% of outperformed genomes (Accuracy)	% (Precision)	% (Recall)	% (F1)
GeneMarkS (PGP)	80.2	37.7	91.4	80.5
GeneMarkS (GP)	73.9	37.7	88.5	74.1
Prodigal (PGP)	91.6	37.9	97.4	92.0
Prodigal (GP)	87.9	37.8	96.1	88.2
FragGeneScanRS (PGP)	93.5	37.9	99.5	98.9
FragGeneScanRS (GP)	91.5	37.9	99.3	98.2

Table 5. Genome-wise comparison of ProtiGeno with baseline gene finders. We randomly split the genomes into ten folds to avoid data leakage across genes in the same genome. We report the percentage of genomes which ProtiGeno outperforms against each baseline method in terms of accuracy, precision, recall and F-1 score. Similar to the aggregated results reported in Table 1, ProtiGeno outperforms the baseline methods in majority of genomes at accuracy, recall and F-1 score.

K. Computational Resources and Runtime

Our pipeline uses sequences embeddings generated with code provided by Dallago et al. (2021). All sequences are processed using NVIDIA RTX A6000 GPU. We randomly sample 30 genomes from all 4,288 genomes we used in this paper and report their overall runtime. Detailed runtime information for the coding regions can be found here, and noncoding regions can be found here. Out of the 30 sampled genomes, each genome has 38 coding regions and 836 noncoding regions on average, and takes 31.4 seconds to process on average. The multi-layer perceptron models are trained on 2.3 GHz 8-Core Intel Core i9 CPU, each model used in the 10-fold cross validation in Table 1 tables about 1 hour to train.

L. Baseline Implementaion and Parameters

In this paper, we compared ProtiGeno with four baseline methods: GeneMarkS2 (Besemer et al., 2001), Prodigal (Hyatt et al., 2010), Balrog (Sommer & Salzberg, 2021), and FragGeneScanRS (Van der Jeugt et al., 2022). All of these baseline methods were run on whole prokaryotic genomes. The baseline methods were all run with no specific hyperparameter tuning. All of them were ran with their default input options. For GeneMarkS2, we mentioned bacteria as the genome type, gcode was set to 11 since that is the default code for bacteria, and the output file format was specified as gff3. In Prodigal's case, the input file's protein translation was specified as prot_trans, and the output file format was gff. Balrog was run with its default parameters, as mentioned in their colab notebook; no changes were made beyond uploading the whole genome as

input. Similarly, FragGeneScanRS was run with default setting without any hyperparameter tuning.

M. ProtiGeno evaluation results on original dataset

In addition to the results we obtained in Table 1, we also trained and evaluated ProtiGeno on ALL available coding regions and non-coding regions embeddings without any sampling, considering that in real-world scenarios, noncoding ORFs are much more abundant. We split genomes into ten folds with the same randomness as Table 1 and report the mean and standard error of the mean (SEM) in Table J. As we see that, compared to Table 1, ProtiGeno still outperforms all baseline models on recall, indicating the method is less likely to miss real coding regions compared to the existing baseline methods, although at the cost of a lower precision. Notice that the precision is much lower than those reported in Table 1 because with the majority of the training and testing data being noncoding regions, ProtiGeno creates more false positives compared to the sampled, balanced data.

Method	Accuracy	Precision	Recall	F1
ProtiGeno	0.935 ±0.028	0.376 ±0.083	**0.928** ±0.012	0.529 ±0.090
GeneMarkS (PGP)	0.993 ±0.001	**1.0** ±0.000	0.809 ±0.012	0.894 ±0.007
GeneMarkS (GP)	**0.994** ±0.001	0.998 ±0.000	0.837 ±0.011	**0.910** ±0.007
Prodigal (PGP)	0.990 ±0.001	**1.0** ±0.000	0.741 ±0.012	0.851 ±0.008
Prodigal (GP)	0.992 ±0.001	0.998 ±0.000	0.772 ±0.011	0.871 ±0.007
FragGeneScanRS (PGP)	0.984 ±0.001	**1.0** ±0.000	0.556 ±0.011	0.714 ±0.009
FragGeneScanRS (GP)	0.985 ±0.001	0.996 ±0.001	0.604 ±0.010	0.752 ±0.008
Random	0.499 ±0.003	0.472 ±0.002	0.501 ±0.004	0.486 ±0.002

Table 6. Comparison of ProtiGeno with baseline gene finders evaluated on ALL available coding regions and non-coding regions, considering that in real-world scenarios, noncoding ORFs are much more abundant. We split all genomes into ten folds in with the same random seed as table 1 to avoid data leakage across genes in the same genome. We report the mean and standard error of the mean (SEM) of the test performance across ten folds.