Published in final edited form as:

Cell Host Microbe. 2021 January 13; 29(1): 121–131.e4. doi:10.1016/j.chom.2020.11.002.

Automated Prediction and Annotation of Small Open Reading Frames in Microbial Genomes

Matthew G. Durrant[1,2], **Ami S. Bhatt**[1,2,3]

[1)]Department of Genetics, Stanford University, Stanford, California, 94305, USA

[2)]Department of Medicine (Hematology, Blood and Marrow Transplantation), Stanford University, Stanford, California, 94305, USA

[3)]Lead Contact

Summary

Small open reading frames (smORFs) and their encoded microproteins play central roles in microbes. However, there is a vast unexplored space of smORFs within human-associated microbes. A recent bioinformatic analysis used evolutionary conservation signals to enhance prediction of small protein families. To facilitate annotation of specific smORFs, we introduce *SmORFinder*. This tool combines profile hidden Markov models of each smORF family and deep learning models that better generalize to smORF families not seen in the training set, resulting in predictions enriched for Ribo-Seq translation signals. Feature importance analysis reveals that the deep learning models learn to identify Shine-Dalgarno sequences, deprioritize the wobble position in each codon, and group codon synonyms found in the codon table. A core genome analysis of 26 bacterial species identifies smORFs of unknown function. We pre-compute smORF annotations for thousands of RefSeq isolate genomes and HMP metagenomes, and provide these data through a public web portal.

Graphical Abstract

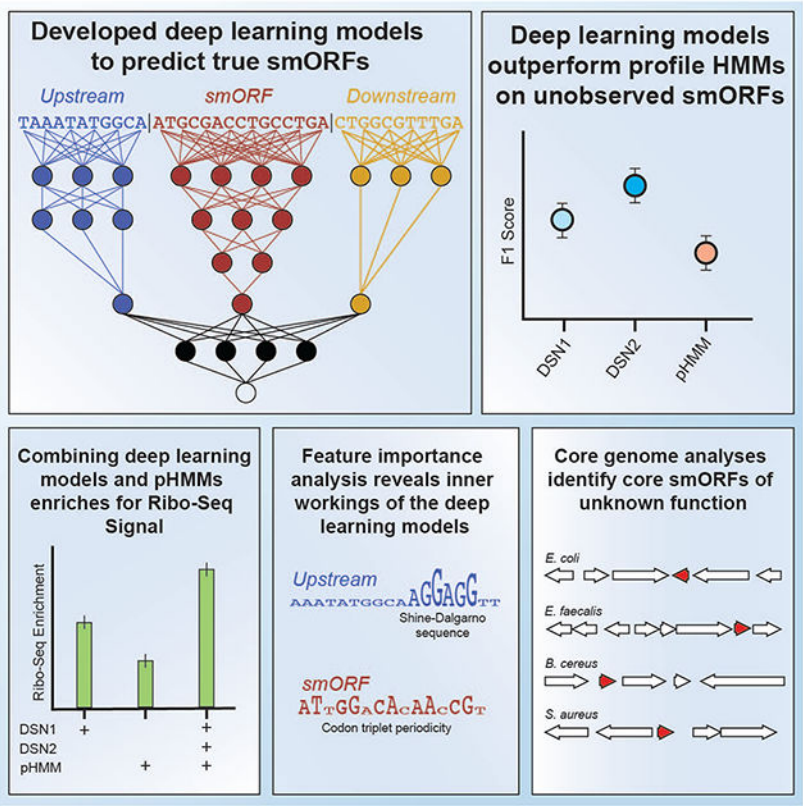
Correspondence: asbhatt@stanford.edu.

Author Contributions

M.G.D. conceived the study and analyses, designed the software, performed formal analyses, visualized the data, wrote the manuscript, and coordinated the project. A.S.B. helped with conceptualization, writing, and editing the manuscript, and funding acquisition.

Declaration of Interests

The authors declare no competing interests.



eTOC Blurb

Small proteins are often overlooked using available research tools. Durrant and Bhatt use deep learning models to improve the detection of small proteins commonly found in the human microbiome. This annotation tool is freely available, along with a re-analysis of thousands of publicly available genomes.

Introduction

Small open reading frames (smORFs, alternatively sORFs; ≤50 amino acids in length) and the microproteins (also referred to as small proteins) they encode play important roles in microbes, including housekeeping, phage defense, and cell signaling functions. Microproteins have been identified across multiple domains of life, and given their potential role in mediating cell-cell communication, have been a topic of growing interest in various fields of biology and translational medicine (Hanada et al., 2013; Storz, Wolf and Ramamurthi, 2014; Makarewich et al., 2018; Leslie, 2019). Despite their importance, the smORFs that encode these proteins are difficult to identify, and as a result they are often overlooked (Su et al., 2013; Storz, Wolf and Ramamurthi, 2014; Duval and Cossart, 2017). Techniques such as ribosome profiling (Ribo-Seq) and proteomic approaches have had some success and provide evidence of transcription and translation of candidate smORFs (Aspden et al., 2014; Miravet-Verde et al., 2019; Weaver et al., 2019; Lohmann et al., 2020). However, these approaches are limited by an experimental bottleneck, usually requiring the

isolation and culture of an organism of interest, and they can only detect what is being actively translated.

Microbial smORFs can be difficult to accurately detect using computational annotation of sequenced genomes due to their small size (Hyatt et al., 2010). In the past, many microproteins were discovered largely by serendipity, either overlapping noncoding RNAs or in the intergenic spaces between large ORFs (Jørgensen et al., 2013; Pinel-Marie, Brielle and Felden, 2014). Others have used machine learning techniques to identify smORFs in a limited number of bacterial species (Friedman et al., 2017). A more systematic way to identify and annotate smORFs within microbial genomes would be of great value.

Recently, a bioinformatic analysis used evolutionary conservation signals to enhance smORF prediction and identified thousands of microprotein families in human-associated metagenomes at a large scale (Sberro et al., 2019). Unfortunately, many of these smORFs remain unannotated in microbial reference genomes and standard genome annotation tools do not accurately predict them. While the microprotein families identified by Sberro et al. provide a larger set of candidate proteins, no method exists to automatically annotate these smORFs in existing genomic sequences from bacterial isolates. Some recent studies have shown that logistic regression and support vector machines (SVMs) hold promise as methods to identify microproteins (Zhu and Gribskov, 2019; Li and Chao, 2020). In one case, the source code is not yet available and it has not yet been published in a peer-reviewed journal, making it difficult to evaluate and understand the underlying approach. In the other case, the model's precision on bacterial microproteins from outside of their training set was not assessed. Being able to annotate these microproteins in microbial genomes is an important step toward understanding their diverse functions. To that end, a computational tool that can streamline their annotation and can be applied to any sequenced genome or metagenome would be an important step toward understanding the biological functions of these microproteins.

Using the >4,500 smORF families identified by Sberro et al. (2019), we sought to build an annotation tool that combines profile Hidden Markov models (pHMMs) and deep learning models to annotate smORFs in genome and metagenome assemblies. Deep learning has rapidly increased in popularity in the field of genomics, and has been applied to the task of ORF prediction generally (Al-Ajlan and El Allali, 2019). Deep learning models obviate the need for "feature engineering", the practice of summarizing raw features into metrics and statistics that are believed to be more predictive, and can learn important higher-order features automatically by analyzing raw sequence data (Zou et al., 2019). However, deep learning often does require a careful model architecture and hyperparameter selection process to achieve optimal performance (Li et al., 2017), which can be computationally expensive (See Table S1, "Glossary of Terms").

Here, we develop *SmORFinder*, a tool that combines profile HMMs and deep learning classifiers to identify smORFs in microbial genomes. First, we train deep learning models that analyze the predictions of the Prodigal ORF annotation tool (Hyatt et al., 2010) to determine if the predictions are true smORFs, using the Sberro et al. (2019) data as a training set. We demonstrate that the deep neural networks have higher performance (F1

score) than profile HMMs when it comes to the classification of smORF families that did not exist in the training set. Then we apply this tool to Ribo-Seq and MetaRibo-Seq (Fremin, Sberro and Bhatt, 2020) datasets, demonstrating that its predictions are enriched for actively translated genes, and that combining predictions from different models improves performance. Next, we find evidence that our deep learning models have learned to identify Shine-Dalgarno sequences, to deprioritize the wobble position in each codon, and to group codons in a way that strongly corresponds with the codon table. Finally, we re-annotate all bacterial genomes in the RefSeq database, making the standalone tool and annotations freely available to the research community.

Results

Deep learning models detect unobserved smORF families with greater recall and F1 score than profile HMM models

Profile Hidden Markov Models (pHMMs) are widely used in bioinformatics to annotate proteins that are believed to belong to a certain family, or to contain specific domains (Eddy, 1998). Annotation tools such as Prokka (Seemann, 2014) use pHMMs built to recognize specific protein domains, which can then be used to annotate predicted microbial ORFs. We sought to compare deep learning models to pHMMs for predicting smORFs. To maximize the potential for comprehensive annotation of smORFs, including those that are highly divergent from those included in the training set, we optimized for deep learning models that performed well on smORF families (smORF clusters identified by Sberro et al., 2019) that were intentionally excluded from the training set (unobserved smORFs; Fig. S1A).

We used a deep learning model architecture that took three different nucleotide sequences as inputs - the smORF itself, 100 bp immediately upstream of the smORF, and 100 bp immediately downstream of the smORF. Using our training set of predicted true positive smORFs (positives) and predicted true negative smORFs (negatives), we developed two deep learning models using the hyperband algorithm to tune hyperparameters. We refer to these deep learning model architectures generally as DeepSmORFNets (DSN). The first model (DSN1) was tuned to have the lowest validation loss on a validation set of observed smORF families (“Validation - Observed”), and the second model (DSN2) was tuned to have the highest F1 score on a validation set of unobserved smORF families (“Validation - Unobserved”) (Fig. S1B and S1C). The models differ in interesting ways: of note, DSN2 uses bidirectional LSTM layers while DSN1 does not, and DSN1 has more convolutional layers and fewer total parameters (Fig. S2E).

We then compared the performance of these models against pHMMs. We randomly split the training and validation sets 64 times, where each random split resulted in a unique training set, a validation set of observed smORF families (“Validation - Observed”), and a validation set of unobserved smORF families (Validation - Unobserved). We trained the DSN models on each of these randomly split training sets, and we chose the model with the lowest loss in the “Validation - Observed” set. Likewise, we trained pHMMs on the 64 randomized training sets, and compared their performance to the deep learning models on the validation sets (Fig. 1).

We find that DSN1, DSN2, and the pHMMs all perform well on the “Validation - Observed” set (Fig. 1A), with recall, precision, and F1 Scores that exceed 0.975 for both DSN1 and DSN2 at P(smORF) > 0.5, and 0.99 for pHMMs at E-value < 1e-6 (see Fig. S2A for Precision and Recall metrics; Table S1 & S2). Performance on the “Validation - Unobserved” set shows an interesting difference between deep learning models and pHMMs. At a cutoff of P(smORF) > 0.5, DSN2 has better average recall than pHMMs at a cutoff of E-value < 1 (Paired t-test; $P < 1 \times 10^{-16}$), better precision (Paired t-test; $P < 1 \times 10^{-16}$), and a better F1 score (Paired t-test; $P < 1 \times 10^{-16}$). At the same cutoff, DSN1 has slightly worse recall than pHMMs at a lenient cutoff of E-value < 1 (Paired t-test; P = 0.00684), but a slightly better F1 Score (Paired t-test; $P = 1.9 \times 10^{-10}$). These results suggest that the deep learning models are better at generalizing to the unobserved smORF families overall, while the precision of pHMMs continues to be superior at a significance cutoff of E-value < 1e-6. This suggests the models may complement each other when used together to identify smORFs.

We then compared DSN1 and DSN2 models to simpler neural networks without hyperparameter optimization (Fig. S2B). A simple neural network with only one convolutional layer and all three input sequences performs well on the “Validation - Unobserved” set (F1 score = 0.756±0.029), although not as well as DSN2 (F1 score = 0.776±0.021; Paired t-test; $P = 4.4 \times 10^{-14}$). A simple neural network analyzing only the smORF nucleotide sequence performed poorly on the “Validation - Unobserved” set (F1 score = 0.676±0.02). We also compared the final DSN models to simple k-nearest neighbor algorithms trained on nucleotide and protein k-mer composition, and we show substantial improvements by DSN models on both the “Validation - Observed” and “Validation - Unobserved” sets (Fig. S2C). Finally, we analyzed how the false positive rate varied across curated negatives (naturally occurring smORF sequences with no evidence of codon conservation) and shuffled negatives (smORF sequences randomized using tetramer shuffling) (Fig. S2D). We find that both the DSN1 and DSN2 models perform well at correctly identifying curated negatives for both the training set and the “Validation - Observed” set. It was more difficult for DSN1, DSN2, and pHMM models to correctly identify curated negatives that were in the “Validation - Unobserved” dataset, with the false positive rate reaching as high as 0.189±0.034 when using a lenient significance cutoff of P(smORF) > 0.5. In general, the DSN models and the pHMMs performed relatively well on the shuffled negatives with all models reaching an average false positive rate less than 0.06 at the most lenient significance cutoffs.

Despite the increased precision of pHMMs, there are several advantages to the deep learning models. Both deep learning models have fewer learnable parameters than the pHMMs (Fig. S2E), with the pHMMs having 4x more parameters than DSN1. As implemented in python’s keras package and the command line tool hmmsearch, DSN1 runs about as fast as the pHMMs, while DSN2 is slower (Fig. S2F) (Eddy, 1998; Chollet and Others, 2015). Notably, the deep learning models require no sequence clustering or alignment, just the raw smORF and flanking nucleotide sequences. To construct pHMMs for each smORF family, they must be clustered and aligned, with a different pHMM being built for each family. This alignment-free approach to building the model can be considered an advantage of the deep

learning models. However, the pHMMs are generative models and require no negative examples for training, while the deep learning models do require these negative examples.

After hyperparameter tuning of the final DSN models and evaluating their performance on the 64 randomized validation sets, we trained the models one final time on the final training set, which included at least one representative of each smORF family. We then evaluated all three models (pHMMs, DSN1, and DSN2) on a test set that was held out entirely from the hyperparameter tuning process. We find that all three models perform well on the validation and test sets, with the pHMMs performing the best overall, but all three models having recall, precision and F1 scores that exceed 0.98 on the test set (Fig. 1B).

We find that while there is considerable overlap in the positive predictions made by DSN1, DSN2, and the pHMM models across datasets (Fig. S3), some of their predictions are complementary. Thus, we built an ensemble model that could combine predictions of each model to optimize overall performance. We tested ensemble model combinations at lenient (pHMM E-value < 1, DSN1 P(smORF) > 0.5, and DSN2 P(smORF) > 0.5) and strict (pHMM E-value < 1e-6, DSN1 P(smORF) > 0.9999, and DSN2 P(smORF) > 0.9999) significance cutoffs (Fig. 1C). We find that by combining the union of all predictions that meet a strict significant cutoff for one model (“Union (Strict)”) with the intersection of all lenient predictions (“Intersection (Lenient)”) we can maintain both a high F1 score in the training and “Validation - Observed” sets (0.998 and 0.994, respectively), and a relatively high F1 score in the “Validation - Observed” set (0.757). This final ensemble model functions as a balance between the high performance of the pHMMs in the Training and “Validation - Observed” sets with the high performance of the DSN models in the “Validation - Unobserved” set.

Predicted smORFs are enriched for Ribo-Seq signal

We next decided to gauge the quality of the smORF predictions using Ribo-Seq signal as a proxy. Ribo-Seq, a method for ribosome profiling, can identify mRNA sequences that are directly bound by a ribosome, indicating active translation (Ingolia et al., 2009). We reasoned that a more accurate set of smORF predictions would be more likely to be translated, and thus more likely to be enriched for a Ribo-Seq signal. We used previously generated and sequenced Ribo-Seq libraries for one *Bacteroides thetaiotaomicron* isolate (previously published (Sberro et al., 2019)), and four metagenomic human microbiome samples using MetaRibo-Seq, a technique for metagenomic ribosome profiling (Fremin, Sberro and Bhatt, 2020).

We packaged all three models (pHMMs, DSN1, and DSN2) together into a single command line tool that we refer to as *SmORFinder*. We kept all predicted smORFs that met a pHMM cutoff of E-value < 1.0, a DSN1 cutoff of P(smORF) > 0.5, or a DSN2 cutoff of P(smORF) > 0.5. We found 15 smORF families were false positives, and they are automatically excluded from consideration by *SmORFinder*. These false positive families corresponded to the N-terminus of Peptide chain release factor RF2 (*prfB*) in many different species. This gene contains a naturally occurring programmed frameshift that is corrected upon translation (Curran, 1993), and the Prodigal tool fails to account for this, leading to a spurious smORF annotation.

In *B. thetaiotaomicron*, we find that 86.1% of non-smORFs (ORFs coding for proteins greater than 50 aa in length) have a Ribo-Seq signal (Reads Per Kilobase Million (RPKM) ≥ 0.5; Fig. 2A) compared to 23.6% of smORFs. We find that genes predicted by at least one model to be true smORFs are more likely to be enriched for Ribo-Seq signal than "Rejected smORF" predictions (smORFs that did not meet minimum significance cutoffs for any of the three models). The set of predicted smORFs that met strict significance cutoffs for DSN1, DSN2 and pHMMs was found to be enriched for Ribo-Seq signal over the "Rejected smORFs" (Fisher's exact test; P=0.0250). This set was identical to the set identified by DSN1 alone with a high significance cutoff of P(smORF) > 0.9999. The small number of predicted smORFs in this bacterium reduces the power to detect Ribo-Seq enrichment.

We repeated this analysis using published MetaRibo-Seq data generated from stool samples of four human subjects (Fig. 2B) (Fremin, Sberro and Bhatt, 2020). We find in general that predicted smORFs are much more enriched for MetaRibo-Seq signal than "Rejected smORFs". In these samples, the MetaRibo-Seq enrichment of predicted smORFs even exceeds the non-smORF enrichment for high-confidence sets that meet high significance thresholds for one or more of the models. As we require higher significance thresholds for the three models (pHMM, DSN1, DSN2), and concordance across multiple models, the MetaRibo-Seq enrichment increases across the four samples. This suggests that there is value in combining the predictions of the three sets to generate a confident set of smORF predictions.

Feature importance analysis reveals inner workings of deep learning models

Deep learning models have been criticized in the past due to their lack of interpretability, often described as a "black box". Recent advances in deep learning interpretation have overcome this challenge, enabling us to gain insight into the features of the input that play a role in the final model prediction, a technique called feature importance analysis (Shrikumar, Greenside and Kundaje, 2017). We applied feature importance analysis to our deep learning models using the Deep Learning Important FeaTures (DeepLIFT) method as implemented in the SHapley Additive exPlanations (SHAP) python package (Lundberg and Lee, 2017). Briefly, this method calculates the importance of individual input features relative to a set of randomized references by backpropagating the contributions of all neurons to every feature of the input. In the case of our smORF nucleotide sequences, this results in importance scores (also called contribution scores) assigned to each nucleotide in the sequence. For example, if a deep learning model was built to identify ChIP-seq binding sites for a given transcription factor, such as CTCF, feature importance analysis using the DeepLIFT method would identify the CTCF binding motif in individual examples, producing experimentally actionable information.

We apply this technique to both DSN1 and DSN2 to see if we can gain insight into how the models identify true smORFs. First, we analyze the average DeepLIFT importance scores of all upstream and downstream smORF-flanking nucleotide sequences found in the training set (Fig. 3A). In the upstream sequence, we see a distinct peak at −12 bp in both the DSN1 and DSN2 importance scores. This is in the range of where we typically find the Shine-Dalgarno sequence (a well-described and conserved ribosomal binding site), and upon

inspection of individual examples we see that both models did in fact identify the AGGAGG Shine-Dalgarno motif as an informative discriminating feature (Fig. S4A and S4B). In the downstream sequence, it would appear that DSN1 places greater importance on the more proximal nucleotide sequences, while DSN2 seems to have identified two particularly important positions at +13 and +4 positions downstream from the stop codon of the smORF. At positions +1 through +10, the nucleotides with the highest average importance scores for DSN2 are all adenine, with the exception of position +3 which is a thymidine. At positions +10 through +20, the nucleotides with the highest average importance scores for DSN2 are all adenine. This suggests that DSN2 has determined that an A-rich downstream sequence may be predictive of a true smORF. By contrast, DSN1 places greater importance on cytosines in the downstream sequence, although it assigns much less importance to the downstream sequence overall. The prioritization of A-rich downstream regions may indicate the rho-independent (intrinsic) transcription termination mechanism, which includes a chain of uracils in the mRNA transcript (d'Aubenton Carafa, Brody and Thermes, 1990; Peters, Vangeloff and Landick, 2011).

To further illuminate the role of the upstream and downstream regions in the DSN1 and DSN2 models, we perform a feature ablation experiment (Chuang and Keiser, 2018) where we only train the model architectures using the upstream sequence input branch, the downstream sequence input branch, and the ORF sequence input branch independently (Fig. S4C). We find that the upstream and downstream regions of the DSN2 model perform quite well independently of the ORF sequence, and substantially better than the DSN1 model, corresponding to the higher number of parameters allotted to these flanking regions in the DSN2 model. We find that the high F1 scores of the DSN1 and DSN2 models in the "Validation - Unobserved" set are only achieved when all three sequence regions are combined in the full model that includes the upstream, downstream, and smORF sequences.

Next, we analyze the average importance scores across the first 21 and the last 21 base pairs within each smORF (Fig. 3B). There appears to be some difference across both scores for the two models, but what is most striking is the obvious periodicity in the signal. This is not surprising considering the periodic nature of codons found in functional ORFs. When we average the importance scores across all codons, we see that both DSN1 and DSN2 place greatest importance on the second codon position, and the least importance on the third codon position or "wobble" position (Fig. 3C). While it is not clear why greater importance would be placed on the second codon position compared to the first codon position, the fact that the wobble position has less overall importance is intriguing considering its often redundant role in the codon. This implies that the model has learned to deprioritize the identity of the wobble position when making its predictions.

We next look at average importance scores of each unique codon across all true smORFs (Fig. 3D). We find that codons are highly correlated in their average importance across the two models (Spearman $r = 0.906$; $P < 2.2 \times 10^{-16}$). When the importance score of each amino acid is averaged across all codon synonyms, glutamate, aspartate, valine, and alanine have the four highest average importance scores for both DSN1 and DSN2. The four amino acids with the lowest average importance scores across the two models are arginine, serine, tryptophan, and cysteine. In the case of DSN2, the average importance scores of these four

amino acids are actually negative, indicating that on average these amino acids actually prompt the model toward making negative predictions, implying that in certain contexts true smORFs may typically lack these amino acids.

To further shed light on the inner workings of the DSN model, we analyze the average feature importance score for each codon in the training set to determine if there was a significant correlation with Shannon positional entropy of microprotein family multiple sequence alignments. We find that for the training set on the whole there is a slight negative correlation (Pearson $r = -0.0145$; $P < 1e\text{-}16$), indicating that regions with higher positional entropy tend to have lower feature importance scores (Fig. S4D). However, as we increase the minimum number of unique microprotein sequences per family above 40, we begin to see a positive correlation between feature importance and positional entropy. This suggests that while there is some relationship between positional entropy and feature importance, this may vary between microprotein families of different sizes, and it cannot fully explain the behavior of the DSN models.

Finally, we investigated if the deep learning models learned to assign similar importance to codon synonyms. This implies that some representation of the codon table was learned during training. We developed a Codon Synonym Similarity Score (CSS score), which is the average standard deviation of importance values among codon synonyms (Fig. 3E). We first calculated the CSS score for the DSN1 and DSN2 models, and then we permuted the codon synonyms across the importance scores to generate a null distribution of CSS scores. We find that for both models, the true CSS score is very low in the range of permuted CSS scores, indicating that codon synonyms share similar importance scores, and that some representation of the codon table was learned by the model.

Core genome analysis identifies core smORFs of unknown function

Seeing the value in pre-computing smORF annotations for RefSeq genomes for the scientific community, we used *SmORFinder* to analyze 191,138 RefSeq genomes, in addition to the HMP metagenomic samples that were used as part of the initial smORF family identification process. This included genomes across 63 bacterial phyla, with 104,658 genomes belonging to members of the Proteobacteria phylum, and 19,681, 12,338, and 11,511 genomes belonging to the *Escherichia coli*, *Staphylococcus aureus*, and *Salmonella enterica* species, respectively. These data, along with other useful tools for smORF analysis, are available through our web portal that can be accessed through our github repository at https://github.com/bhattlab/SmORFinder.

We carried out a core-genome analysis of 26 of the most common species' genomes found in RefSeq (Table S3). We find 692 putative smORFs to be part of the core genome across the 26 species (Fig. 4A). These include all smORFs annotated by Prodigal with lowered minimum size cutoffs, and prior to filtering according to DSN or pHMM significance cutoffs. The total number of such core smORFs varies widely across species, with 106 identified in *Bacillus cereus*, and 4 identified in *Helicobacter pylori*. However, the total number of core smORFs is difficult to meaningfully compare across species, as they vary in their overall diversity. Across all species, 70.7% of these smORFs contain no recognized

Pfam domains (El-Gebali et al., 2019), 9.54% contain a ribosomal protein domain, and 19.77% contain some other known domain.

When we then use our DSN predictions as calculated by the *SmORFinder* tool to filter this list of core smORFs, we reduce the total number from 692 to 213 (Fig. 4B). By default, this tool uses an ensemble model that combines the predictions of the three models (See Fig. 1C). This enriches for smORFs that contain a predicted Pfam domain, with 31.0% being ribosomal proteins, 30.49% containing some other Pfam domain (including domains of membrane bound YbgT-like proteins, entericidins, and Multidrug efflux pump-associated protein AcrZ among others), and 38.5% containing no Pfam domain. This list can be further reduced by relying only on pHMM models of known smORF families with a strict significance cutoff (E-value < 1e-6), resulting in 167 such core smORFs (Fig. 4C). Using only this significance cutoff as a filter, the total number of core smORFs drops dramatically for some species, such as *B. cereus* whose total number of core smORFs drops from 23 to 8. We find that overall, using the smORFs identified by the *SmORFinder* tool as opposed to the strict pHMM predictions alone increases the number of core smORFs with a domain of unknown function from 18 to 20, the number of core smORFs with some other domain of known function from 37 to 45 (including Staphylococcus haemolytic domains and Stage V sporulation domains), and the number of core smORFs with no domain from 53 to 90.

We find four smORF families with no recognized Pfam domain that appear in more than two different species' core genomes (Fig. 4D). The smORF family smorfam02479 is homologous to YshB, a predicted transmembrane protein recently shown to play a role in intracellular replication in *Salmonella* virulence (Bomjan, Zhang and Zhou, 2019). We find that members of this smORF family exist in the core genomes of *S. enterica* as well as other Enterobacteriaceae such as *E. coli*, *K. pneumoniae*, and *S. sonnei*. The smORF family smorfam02447 shown in Fig. 4D is a gene encoding a 40 aa protein found between genes encoding the P-loop guanosine triphosphatase YjiA and zinc uptake system protein ZnuA in the *Enterococcus faecalis* genome. Members of this smORF family were found in the core genomes of *S. aureus, S. agalactiae, S. pyogenes*, and *E. faecalis*, and its function has not been characterized. The smORF family smorfam04045 protein shown in Fig. 4D is a 49 aa protein found between genes encoding a largely uncharacterized protein and a predicted lipase in the *B. cereus* genome. Members of this smORF family were found in the core genomes of *B. cereus, S. suis, and S. pyogenes*, and the representative member of this family is 91.8% identical to a *B. manliponensis* gene described as an "Alcohol dehydrogenase" in UniProt, although most other homologs are described as uncharacterized. The smORF family smorfam00860 shown in Fig. 4D encodes a 44 aa protein found between the genes encoding an uncharacterized protein and putative HMP/thiamine permease protein YkoE in the *S. aureus* genome. Members of this smORF family were found in core genomes of *S. aureus* and *S. epidermidis*, and its function has not been characterized.

Discussion

Recent advances in bioinformatic annotation approaches and *de novo* annotation of genes using Ribo-Seq have enabled the discovery of thousands of smORFs. The microproteins that they encode have emerged as macromolecules of interest in organisms ranging from

microbes to plants to mammals. Unfortunately, to date, no method exists for the accurate annotation of microbial genomes for these smORFs, and most existing microbial genomes are lacking comprehensive annotation for ORFs less than 150 nucleotides in length. In this study, we present and evaluate the performance of a smORF annotation pipeline based on the 4,500 smORFs identified by Sberro et al. (2019). We demonstrate that deep learning models can distinguish between true smORFs and spurious smORFs about as well as pHMMs trained on observed smORF families, and they perform better than pHMMs on unobserved smORF families.

We find that both the deep learning models and the pHMMs dramatically increase the Ribo-Seq and MetaRibo-seq enrichment signal of the annotated smORF set. This suggests that selecting smORFs based on the predictions of these models greatly enriches for actively translated and thus likely functional smORFs. Including the three different models (DSN1, DSN2, and the pHMMs) in the *SmORFinder* annotation tool enables a user to select a range of options for filtering a set of candidate smORFs. For example, rather than relying on a strict significance cutoff for one or multiple models, we find that using lenient significance cutoffs that must be met by all three models is another good strategy for narrowing down a list of candidate smORFs. We recommend using the default ensemble model to achieve a balance between recall and precision when annotating smORFs, but for applications where precision is more important than recall we recommend using even more stringent cutoffs or relying exclusively on the pHMMs with a stringent cutoff.

Recent advances in feature importance analysis allow us to peer into the “black box” of deep learning. This is a fascinating look at how these powerful predictive algorithms learn to identify true smORF families, and we can see that they automatically learn features that scientists characterized long ago by experimental means (Shine-Dalgarno sequences, codon periodicity, codon synonyms, etc.). It also acts as an interesting opportunity to find generalizable features that may have previously gone unnoticed. For example, DSN2 appears to assign greater importance to 3’ downstream sequences that are A-rich. This could indicate that the model has learned to recognize rho-independent (intrinsic) transcription termination sequences, which are known to contain a chain of uracils in the mRNA transcript (d’Aubenton Carafa, Brody and Thermes, 1990; Peters, Vangeloff and Landick, 2011). Intrinsic terminator sequences are not taken into consideration by ORF annotation algorithms such as Prodigal (Hyatt et al., 2010).

Our core genome analysis of 26 different bacterial species identified many smORFs that appear to be highly conserved, including smORFs that were identified using permissive Prodigal annotation and clustering before any *SmORFinder* models were applied. Using *SmORFinder* predictions to filter these core smORFs showed a significant reduction in the total number of smORFs for some species. For example, 106 core smORFs were found in *B. cereus* genomes prior to *SmORFinder* filters, and reduced to only 8 core smORFS after applying strict filters. This could indicate that there are many smORFs that were not found in the initial set of core smORFs but are found in the *B. cereus* genome, or that a large number of the core smORFs found in the *B. cereus* genome are false positives. Further experiments and efforts to supplement our set of core smORFs will likely shed light on this question.

While efficient and powerful, the approach that we take in this study has several limitations. First, the *SmORFinder* annotation tool is primarily limited by the Prodigal calling algorithm. The original set of >4,500 families identified by Sberro et al. relied on a downstream analysis of smORFs that were identified by Prodigal with a lowered minimum size threshold. This resulted in a set of candidate microproteins that are still biased toward the larger end of this size distribution. *SmORFinder* is also limited to predictions made by Prodigal, and can be thought of as an additional filter step on top of Prodigal predictions. Second, we are also limited by the accuracy of the predictions made by Sberro et al. in their original study. In the course of completing this analysis, we identified 15 smORFs that are false positives; while this is a relatively small number of overall false positives, it is likely that there are other such false positives in the overall set. Third, the true generalizability of the models introduced in this study is also questionable. That is, it does not appear that the model can reliably identify true smORFs that are completely unrelated to smORFs in the original training set. This means that some number of true smORFs that are not represented in the training set will be overlooked by our tool. Due to the origin of the 4,500 smORF families used in the training set, *SmORFinder* is particularly well-suited for the analysis of human microbiomes, but it may not as readily generalize to species limited to other environments. Finally, we rely on Ribo-Seq signals as a test of whether our model enriches for true microproteins, and we acknowledge that Ribo-Seq may not be able to accurately identify true smORFs in all circumstances.

These limitations notwithstanding, with the growing interest in microproteins, *SmORFinder* should be valuable to the research community as it will allow researchers to filter down lists of candidate smORFs to a more accurate list of smORF predictions. We have precomputed the smORFs of thousands of bacterial RefSeq genomes and HMP metagenomes and made them available for download through a web portal. The annotation tool can easily be installed as a python package and is ready for use. This will enable the study of smORFs, opening up many avenues for biological research. For example, the reannotation of these bacterial genomes could help gain insights into previously conducted experiments, such as transposon-mutagenesis experiments, affording researchers a wealth of functional data. Data are freely available for the research community through our github repository and web portal (https://github.com/bhattlab/SmORFinder). It is possible that a suite of tools, including but not limited to *SmORFinder,* will be developed and applied for the comprehensive, sensitive and specific detection of smORFs across prokaryotes. As such, we anticipate that *SmORFinder* may be augmented by other models as they are published and thoroughly validated.

STAR★Methods

Resource Availability

Lead Contact—Further information regarding the data and code presented in this study is available through the Lead Contact, Ami S. Bhatt (asbhatt@stanford.edu).

Materials Availability—This study did not generate new unique reagents.

Data and Code Availability—The code for *SmORFinder* is available at github.com/bhattlab/SmORFinder. The web portal can be accessed through the github repository. A link to the web portal is available on the github. All data used to train the models presented in this study are available in the github repository github.com/bhattlab/SupplementaryInformation/tree/master/SmORFinder.

Method Details

Curating positive and negative training examples—A critical first step in any approach to develop a homology/pattern-based annotation algorithm is development of a positive and negative training set. In this case, positive and negative examples of smORFs were needed to train the neural network. Positive examples were derived from the 4,539 microprotein families that were originally reported by Sberro et al. (2019). A maximum of 64 examples per protein family were kept, and these 64 were randomly chosen. 100 base pairs of upstream and downstream sequences were used as model inputs, along with the ORF sequence itself. In the event that the upstream and downstream sequences were shorter than 100 base pairs, whatever sequence was available was used. The strategy used to identify negative examples was similar to the one used to identify the positive examples, but with criteria reversed. Prior to applying filters, Sberro et al. identified approximately 444,000 microprotein family clusters (clustered using CD-HIT with the parameters -n 2 -p 1 -c 0.5 -d 200 -M 50000 -l 5 -s 0.95 –aL 0.95 –g 1). To identify negative examples (smORF families that are most likely spurious ORFs), the following filters were applied: First, smORF clusters were excluded if they were predicted to contain a known protein domain according to an Reversed Position Specific (RPS) BLAST search of the Conserved Domains Database (CDD) database (A predicted domain was considered significant if the e-value was less than or equal to 0.01, and the microprotein sequence aligned to at least 80% of the length of the position-specific scoring matrix (PSSM)) (Lu et al., 2020). Next, families with less than 4 unique members were excluded, as this is too few examples to be properly analyzed by RNAcode. Next, RNAcode was run on each family with the parameter --num-samples 200, and default parameters otherwise. RNAcode can identify conserved coding sequences with samples as small as 4 unique examples, but the average pairwise identity of these examples must be below 90% (Washietl et al., 2011). For families with 4 to 7 members, the RNAcode results were considered only if this average pairwise identity threshold was met for the family. With families that contained >8 members, we required at least one pair to fall below the 90% identity threshold. Any families that were predicted by RNAcode to contain coding sequence (CDS) regions were excluded. Next, protein families were aligned to each other using the DIAMOND search algorithm (Buchfink, Xie and Huson, 2014). If any remaining negative example was a significant (e-value < 1e-3) match for any positive example, it was excluded. If any negative example aligned to another negative example that failed the RNAcode conserved CDS detection, it was also excluded. This resulted in 4,705 high-confidence negative microprotein examples. To further supplement this dataset of negative examples, more negative examples were synthesized by shuffling upstream, downstream, and ORF nucleotide sequences. Both positive and negative sequence examples were shuffled using a tetramer shuffling algorithm implemented in the tool by fasta-shuffle-letters in the MEME Suite (Bailey et al., 2009), which was built on the uShuffle algorithm (Jiang et al., 2008). Start and stop codons were preserved, and shuffled ORF sequences were only kept if

they could be fully translated using the same translation table as the original sequence. This was to ensure that the deep learning model would learn to discriminate true positives from random sequences. In summary, the two types of negative data we included were 1) Randomly shuffled tetramers of both positive and negative examples (called “shuffled negatives” and 2) naturally occurring smORFs that have no conservation signal attributable to orfs, and do not contain known protein domains (called “curated negatives”). The final ratio of positive to negative examples was 0.417.

Splitting dataset into training, validation, and test sets—We used a stratified sampling approach to randomly split the full dataset into training, validation, and test sets (Fig. S1A). First, we made sure that each dataset had at least one member of each smORF family, which were randomly distributed across the three datasets. After this requirement was met, the remaining examples were randomly allocated to the three different datasets, with approximately 80% being allocated to the training set, 10% to the validation set, and 10% to the test set. The final training set included 367,184 examples (112,427 positive, 254,757 negative), the final validation set included 47,248 examples (13,192 positive, 34,056 negative), and the final test set included 46,932 examples (12,933 positive, 33,999 negative). The training and validation sets were combined and permuted such that certain protein families in the validation set were excluded from the training set (unobserved smORF families). These permuted datasets were used to estimate the performance of the model on the unobserved smORF families.

Deep learning model architecture and hyperparameter tuning—Hyperparameter tuning was used to identify model architectures that performed best on the validation dataset. The basic model included three inputs, a one-hot encoded vector with dimensions 153x4 to represent the smORF sequence itself, with zeroes padded on the right for smORFs shorter than 153 bp, 100 bp upstream of each smORF encoded as a 100x4 vector, and 100 bp downstream of each smORF encoded as a 100x4 vector. These are then fed into one-dimensional convolutional layers, which are followed by a dropout layer and a pooling layer. All three input branches are flattened and concatenated as a single vector, which is processed by a final dense layer, a dropout regularization, and a final dense layer with a sigmoid activation function that calculates the probability that the input smORF is a true smORF. The many hyperparameters in this model were tuned using the hyperband algorithm (Li et al., 2017) as implemented by the keras-tuner python package (O’Malley, 2020). This algorithm randomly sampled the hyperparameter space, including number of convolutional layers per input branch (1, 2, or 3), the number of filters per layer (32, 64, 128, 256, 512, or 1024), the size of each filter (6, 12, 18, or 24), the dropout rate of the convolutional layers (0.1, 0.3, or 0.5), the dropout rate of the final dense layer (0.1, 0.3, or 0.5), the number of neurons in the final dense layer (16, 32, 64, 128, 256, or 512), the learning rate (1e-5, 1e-4, or 1e-3), the padding method (“valid” or “same”), and the pooling method (max pooling or average pooling). Adam optimization with a learning rate of 1e-4 was used to train the model (Kingma and Ba, 2014). After the first convolutional layer, the number of convolution filters is divided by 2, and the filter size is reduced by one-third of the original filter size. For example, if a model had three layers and 1024 filters of length 18, the second layer would have 512 filters of length 12, and the third layer would have 256 filters of length 6. Our aim

was to identify models that minimized the loss across the “Validation - Observed” dataset, and maximized the F1 score of the “Validation - Unobserved” dataset. Hyperband was run with a maximum number of epochs of 200 and a downsampling factor of 3 over 4 complete iterations, resulting in 512 different hyperparameter combinations. This was repeated for both the “Validation - Observed” and “Validation - Unobserved” datasets. Finally, the same process was repeated with an additional LSTM layer added at the end of each input branch, which added hyperparameters including the number of LSTM neurons per input branch (16, 32, 64, 128, or 256) and the LSTM dropout rate (0.1, 0.3, or 0.5). DSN1 and DSN2 were chosen as the models with the best loss in calculated over the “Validation - Observed” dataset, and the best F1 score calculated over the “Validation - Unobserved” dataset, respectively. See Fig. S1 for a final description of each model’s hyperparameters.

Building profile Hidden Markov Models—Using the positive examples (true smORFs) of each protein family found in the training dataset, profile HMM models were constructed. Microprotein families were aligned using MUSCLE (Edgar, 2004). We then used the command line tool hmmbuild to construct the pHMM for each family (HMMER, no date). All pHMMs were combined into a single file, and the e-value of the pHMM with the lowest pHMM is used to assign a given sequence to a smORF to a family.

Determining how deep learning models and profile HMMs generalize to unobserved smORF families—A permutation approach was used to determine how well the deep learning models and pHMMs generalize to Unobserved smORF families. As depicted in Fig. S1A, the training set and validation set were combined, and the two validation sets were created - one that contained smORF families that were observed at least once in the training set (Validation - Observed), and one that contained smORF families that were not observed in the training set (Validation - Unobserved). The smORF families that were excluded from the training set were randomly chosen, and this process was repeated 64 times to create 64 train-validation splits. The deep learning models (DSN1 and DSN2) were both trained on all 64 training sets for up to 2000 epochs to minimize the training loss. Early stopping was used to choose the model that had no improvement in the calculated “Validation - Observed” loss after 100 epochs. This final trained model was then evaluated on the training set and validation sets to estimate the model’s precision, recall, and F1 score. The distribution of these performance metrics across the 64 permuted datasets was used to determine the error of each estimate as shown in Fig. 1A. The pHMM models were also trained independently on the same 64 permuted datasets to get comparable performance estimates.

Finalizing deep learning model—We trained the final deep learning models and pHMMs on the initial, unmodified training set (Fig. S1A). We selected the deep learning models with the lowest loss in the validation set, with a maximum of 2000 training epochs and early stopping after 100 epochs of no improvement in the validation loss. We then evaluated all models on the validation set and the test set, which was held out from the beginning and was not included in the model architecture selection process. These final models are included in the *SmORFinder* annotation tool in its current implementation.

Validating SmORFinder with Ribo-Seq datasets—Ribo-Seq datasets were used to determine whether the *SmORFinder* annotation tool enriches for actively translated smORFs. We used previously published Ribo-seq datasets that are available through the NCBI SRA portal under the projects PRJNA540869 (Ribo-Seq of *B. thetaiotaomicron* isolate) and PRJNA510123 (MetaRibo-Seq of human stool samples from 4 individuals) (Sberro et al., 2019; Fremin, Sberro and Bhatt, 2020). The *B. thetaiotaomicron* reference genome was annotated using Prodigal configured to identify smORFs. Assembled metagenomes of the 4 MetaRibo-Seq samples were also annotated and used as a reference for each respective sample. Ribo-Seq reads were aligned to reference genomes using bowtie2 (Langmead and Salzberg, 2013). Ribo-Seq coverage of each predicted ORF was calculated using bedtools (Quinlan and Hall, 2010). Any ORF that had a calculated RPKM ≥0.5 was considered to have a Ribo-Seq signal. The *SmORFinder* annotation tool was used to identify predicted smORFs. Any smORF that met at least one of the significance cutoffs (pHMM e-value < 1; DSN1 > 0.5; DSN2 > 0.5) was considered a potential smORF. All smORFs that did not meet any of these cutoffs were considered “Rejected smORFs”. Different subsets of smORFs were identified based on their statistical significance and agreement across the three different models. These subsets were compared to the “Rejected smORFs” subset in terms of Ribo-Seq signal, and Fisher’s exact test was used to determine if the subset significantly differed.

Feature importance analysis—A feature importance analysis of both the DSN1 and DSN2 models was performed to interpret, in part, how the deep learning models were learning to identify true smORFs. This was done using the DeepLIFT algorithm (Shrikumar, Greenside and Kundaje, 2017) as implemented in the SHAP python package (Lundberg and Lee, 2017). This technique measures the importance of individual features, nucleotides in this case, in determining the model’s prediction relative to some references. Dinucleotide shuffling of upstream and downstream nucleotide sequences were used as a reference. The start and stop codons of the ORF sequences were preserved in the references, and the intermediate sequence was dinucleotide shuffled until a non-interrupted ORF was generated. Twenty shuffled references were used for each example. Averages across all examples in the training dataset are shown in Fig. 3.

Codon Synonym Similarity Score—A codon synonym similarity score (CSS score) was calculated to determine how similar the DeepLIFT importance scores were for codons that code for the same amino acid. This was calculated as:

$$CSS\ score = \frac{1}{k}\sum_{i=1}^{k}\hat{\sigma}_i$$

Where $\hat{\sigma}_i$ is the standard deviation of the average feature importance scores for codon synonym group i. This was calculated for $k = 18$ amino acids, excluding methionine and tryptophan which only have one codon for each. To determine a null distribution of this score, codon synonym labels were randomly permuted across the feature importance scores, and the CSS score was recomputed. This was repeated 10,000 times to calculate a permuted

null distribution. The original CSS score was compared to this permuted null distribution to determine its statistical significance.

Annotating smORFs in RefSeq genomes—All RefSeq bacterial genomes were downloaded on April 29th, 2020. This included all genomes matching the NCBI Entrez search query '"Bacteria"[Organism] AND (latest[filter] AND (all[filter] NOT anomalous[filter] AND all[filter] NOT partial[filter]))'. In total, 191,138 genomes were downloaded. These were annotated using the *SmORFinder* annotation tool, and data were compiled into a database that can be accessed through the github repository https://github.com/bhattlab/SmORFinder.

Core-genome analysis—A core-genome analysis was carried out on 26 bacterial species with a high number of available isolates (Table S3). This included *Acinetobacter baumannii, Bacillus cereus, Burkholderia pseudomallei, Campylobacter jejuni, Clostridioides difficile, Enterococcus faecalis, Enterococcus faecium, Escherichia coli, Helicobacter pylori, Klebsiella pneumoniae, Listeria monocytogenes, Mycobacterium tuberculosis, Mycobacterium abscessus, Neisseria meningitidis, Pseudomonas aeruginosa, Pseudomonas viridiflava, Salmonella enterica, Shigella sonnei, Staphylococcus aureus, Staphylococcus epidermidis, Streptococcus agalactiae, Streptococcus pneumoniae, Streptococcus pyogenes, Streptococcus suis, Vibrio cholerae,* and *Vibrio parahaemolyticus*. Mash distances for all isolates of each species, and only isolates that had an average mash distance < 0.05 (corresponding roughly to 95% average nucleotide identity (ANI)) to all other isolates of the species were kept (Ondov et al., 2016). All isolate genomes were annotated using Prodigal (Hyatt et al., 2010) to identify all open reading frames, with the minimum gene size filter lowered to 15 nucleotides. All identified protein sequences were then clustered at 80% identity using CD-HIT (Huang et al., 2010), where each cluster represented a unique "gene". All genes that were found to exist in greater than 97% of all isolates for each species were considered to be part of that species' "core genome."

Comparison to k-nearest neighbors algorithm—The k-nearest neighbor algorithm was used as a comparison with the trained DSN models. The k-mer composition of the smORF nucleotide sequences, the smORF+US+DS sequences, and the microprotein sequence was calculated in python. For nucleotide sequences, the 1-mer, 2-mer, 3-mer, and 4-mer compositions were calculated. For protein sequences, the 1-mer and 2-mer compositions were calculated. This was repeated for all 64 randomized validation sets. The distance between each example in the randomized "Validation - Observed" and "Validation - Unobserved" sets and all of the examples in the training set was calculated using numpy. The mode of the labels of the top k-nearest neighbors was used as the final prediction, with k = 1, 3, 5, 7, 8, and 11 being used.

Correlation between feature importance and positional entropy—The correlation between feature importance of the smORF sequence as calculated by DeepLIFT and Shannon positional entropy of each amino acid in the microprotein family multiple sequence alignment was calculated. The average feature importance of each codon was used to directly compare with microprotein positional entropy. Only the final training set examples

were used for the analysis, with the initial methionines being excluded. The Pearson correlation coefficient was calculated and the significance of the correlation was calculated using the cor.test function in R. An iterative filter was applied to the examples by only including families that had a minimum number of unique examples in the training set, and the correlation test was repeated.

Quantification and Statistical Analysis

All details of statistical analyses and software used in this study can be found in the method details, which we summarize here briefly. Statistical analyses were all conducted in the R programming language. A paired t-test was used to compare performance metrics of different models across the 64 randomized training and validation sets. For Ribo-Seq and MetaRibo-seq enrichment tests, a Fisher's exact test was used to determine if specific subsets of smORFs were enriched or depleted or Ribo-Seq signal relative to the "Rejected smORFs" category.

Additional Resources

A web server that includes (a) pre-computed smORF annotations for RefSeq genomes and HMP metagenomes and (b) a tool to enable uploading and annotation of genomes of interest is linked to at the bottom of the smORFinder github repository webpage: https://github.com/bhattlab/SmORFinder.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgements

We thank Hila Sberro for assistance with compiling information about the smORFs identified in her original study. We thank Brayon J. Fremin for his help with previously published Ribo-Seq and MetaRibo-Seq datasets and for providing feedback on the manuscript. We thank Soumaya Zlitni, Dylan Maghini, Aaron Behr, and Chris Severyn for providing feedback on the manuscript. This work was supported by NIH R01AI148623 and NIH R01AI143757 to A.S.B., the National Science Foundation Graduate Research Fellowship to M.G.D., and in part by NIH P30 CA124435 which supports the Stanford Cancer Institute Shared Resource Genetics Bioinformatics Service Center.

References

Al-Ajlan A and El Allali A (2019) 'CNN-MGP: Convolutional Neural Networks for Metagenomics Gene Prediction', Interdisciplinary sciences, computational life sciences, 11(4), pp. 628–635.

Aspden JL et al. (2014) 'Extensive translation of small Open Reading Frames revealed by Poly-Ribo-Seq', eLife, 3, p. e03528. [PubMed: 25144939]

Bailey TL et al. (2009) 'MEME SUITE: tools for motif discovery and searching', Nucleic acids research, 37(Web Server issue), pp. W202–8. [PubMed: 19458158]

Bomjan R, Zhang M and Zhou D (2019) 'YshB Promotes Intracellular Replication and Is Required for Salmonella Virulence', Journal of bacteriology, 201(17). doi: 10.1128/JB.00314-19.

Buchfink B, Xie C and Huson DH (2014) 'Fast and sensitive protein alignment using DIAMOND', Nature methods, 12(1), pp. 59–60. [PubMed: 25402007]

d'Aubenton Carafa Y, Brody E and Thermes C (1990) 'Prediction of rho-independent Escherichia coli transcription terminators. A statistical analysis of their RNA stem-loop structures', Journal of molecular biology, 216(4), pp. 835–858. [PubMed: 1702475]

Chollet F and Others (2015) Keras. Available at: https://keras.io.

Chuang KV and Keiser MJ (2018) 'Adversarial Controls for Scientific Machine Learning', ACS chemical biology, 13(10), pp. 2819–2821. [PubMed: 30336670]

Curran JF (1993) 'Analysis of effects of tRNA:message stability on frameshift frequency at the Escherichia coli RF2 programmed frameshift site', Nucleic acids research, 21(8), pp. 1837–1843. [PubMed: 8493101]

Duval M and Cossart P (2017) 'Small bacterial and phagic proteins: an updated view on a rapidly moving field', Current opinion in microbiology, 39, pp. 81–88. [PubMed: 29111488]

Eddy SR (1998) 'Profile hidden Markov models', Bioinformatics , 14(9), pp. 755–763. [PubMed: 9918945]

Edgar RC (2004) 'MUSCLE: multiple sequence alignment with high accuracy and high throughput', Nucleic acids research, 32(5), pp. 1792–1797. [PubMed: 15034147]

El-Gebali S et al. (2019) 'The Pfam protein families database in 2019', Nucleic acids research, 47(D1), pp. D427–D432. [PubMed: 30357350]

Fremin BJ, Sberro H and Bhatt AS (2020) 'MetaRibo-Seq measures translation in microbiomes', Nature communications, 11(1), p. 3268.

Friedman RC et al. (2017) 'Common and phylogenetically widespread coding for peptides by bacterial small RNAs', BMC genomics, 18(1), p. 553. [PubMed: 28732463]

Fu L et al. (2012) 'CD-HIT: accelerated for clustering the next-generation sequencing data', Bioinformatics , 28(23), pp. 3150–3152. [PubMed: 23060610]

Hanada K et al. (2013) 'Small open reading frames associated with morphogenesis are hidden in plant genomes', Proceedings of the National Academy of Sciences of the United States of America, 110(6), pp. 2395–2400. [PubMed: 23341627]

HMMER (no date) Available at: http://hmmer.org (Accessed: 25 June 2020).

Huang Y et al. (2010) 'CD-HIT Suite: a web server for clustering and comparing biological sequences', Bioinformatics , 26(5), pp. 680–682. [PubMed: 20053844]

Hyatt D et al. (2010) 'Prodigal: prokaryotic gene recognition and translation initiation site identification', BMC bioinformatics, 11, p. 119. [PubMed: 20211023]

Ingolia NT et al. (2009) 'Genome-wide analysis in vivo of translation with nucleotide resolution using ribosome profiling', Science, 324(5924), pp. 218–223. [PubMed: 19213877]

Jiang M et al. (2008) 'uShuffle: a useful tool for shuffling biological sequences while preserving the k-let counts', BMC bioinformatics, 9, p. 192. [PubMed: 18405375]

Jørgensen MG et al. (2013) 'Dual function of the McaS small RNA in controlling biofilm formation', Genes & development, 27(10), pp. 1132–1145. [PubMed: 23666921]

Kingma DP and Ba J (2014) 'Adam: A Method for Stochastic Optimization', arXiv [cs.LG], Available at: http://arxiv.org/abs/1412.6980.

Langmead B and Salzberg SL (2013) 'Langmead. 2013. Bowtie2', Nature methods, 9, pp. 357–359.

Leslie M (2019) New universe of miniproteins is upending cell biology and genetics, Science. Available at: https://www.sciencemag.org/news/2019/10/new-universe-miniproteins-upending-cell-biology-and-genetics (Accessed: 6 July 2020).

Li L et al. (2017) 'Hyperband: A novel bandit-based approach to hyperparameter optimization', The Journal of Machine. Available at: https://dl.acm.org/doi/abs/10.5555/3122009.3242042.

Li L and Chao Y (2020) 'sPepFinder expedites genome-wide identification of small proteins in bacteria', bioRxiv. doi: 10.1101/2020.05.05.079178.

Lohmann P et al. (2020) 'Function is what counts: how microbial community complexity affects species, proteome and pathway coverage in metaproteomics', Expert review of proteomics, 17(2), pp. 163–173. [PubMed: 32174200]

Lundberg SM and Lee S-I (2017) 'A Unified Approach to Interpreting Model Predictions', in Guyon I et al. (eds) Advances in Neural Information Processing Systems 30 Curran Associates, Inc., pp. 4765–4774.

Lu S et al. (2020) 'CDD/SPARCLE: the conserved domain database in 2020', Nucleic acids research, 48(D1), pp. D265–D268. [PubMed: 31777944]

Makarewich CA et al. (2018) 'MOXI Is a Mitochondrial Micropeptide That Enhances Fatty Acid β-Oxidation', Cell reports, 23(13), pp. 3701–3709. [PubMed: 29949755]

Miravet-Verde S et al. (2019) ‘Unraveling the hidden universe of small proteins in bacterial genomes’, Molecular systems biology, 15(2), p. e8290. [PubMed: 30796087]

O’Malley T (2020) ‘Hyperparameter tuning with Keras Tuner’.

Ondov BD et al. (2016) ‘Mash: fast genome and metagenome distance estimation using MinHash’, Genome biology, 17(1), p. 132. [PubMed: 27323842]

Peters JM, Vangeloff AD and Landick R (2011) ‘Bacterial transcription terminators: the RNA 3’-end chronicles’, Journal of molecular biology, 412(5), pp. 793–813. [PubMed: 21439297]

Pinel-Marie M-L, Brielle R and Felden B (2014) ‘Dual toxic-peptide-coding Staphylococcus aureus RNA under antisense regulation targets host cells and bacterial rivals unequally’, Cell reports, 7(2), pp. 424–435. [PubMed: 24703849]

Quinlan AR and Hall IM (2010) ‘BEDTools: a flexible suite of utilities for comparing genomic features’, Bioinformatics , 26(6), pp. 841–842. [PubMed: 20110278]

Sberro H et al. (2019) ‘Large-Scale Analyses of Human Microbiomes Reveal Thousands of Small, Novel Genes’, Cell, 178(5), pp. 1245–1259.e14. [PubMed: 31402174]

Seemann T (2014) ‘Prokka: rapid prokaryotic genome annotation’, Bioinformatics , 30(14), pp. 2068–2069. [PubMed: 24642063]

Shrikumar A, Greenside P and Kundaje A (2017) ‘Learning Important Features Through Propagating Activation Differences’, in Proceedings of the 34th International Conference on Machine Learning - Volume 70 Sydney, NSW, Australia: JMLR.org (ICML’17), pp. 3145–3153.

Storz G, Wolf YI and Ramamurthi KS (2014) ‘Small proteins can no longer be ignored’, Annual review of biochemistry, 83, pp. 753–777.

Su M et al. (2013) ‘Small proteins: untapped area of potential biological importance’, Frontiers in genetics, 4, p. 286. [PubMed: 24379829]

Washietl S et al. (2011) ‘RNAcode: robust discrimination of coding and noncoding regions in comparative sequence data’, RNA , 17(4), pp. 578–594. [PubMed: 21357752]

Weaver J et al. (2019) ‘Identifying Small Proteins by Ribosome Profiling with Stalled Initiation Complexes’, mBio, 10(2). doi: 10.1128/mBio.02819-18.

Zhu M and Gribskov M (2019) ‘MiPepid: MicroPeptide identification tool using machine learning’, BMC bioinformatics, 20(1), p. 559. [PubMed: 31703551]

Zou J et al. (2019) ‘A primer on deep learning in genomics’, Nature genetics, 51(1), pp. 12–18. [PubMed: 30478442]

Highlights

- Deep learning approaches to smORF identification improve performance
- Deep learning models learn biologically meaningful features of smORF sequences
- *SmORFinder* annotation tool identifies several core smORFs of unknown function

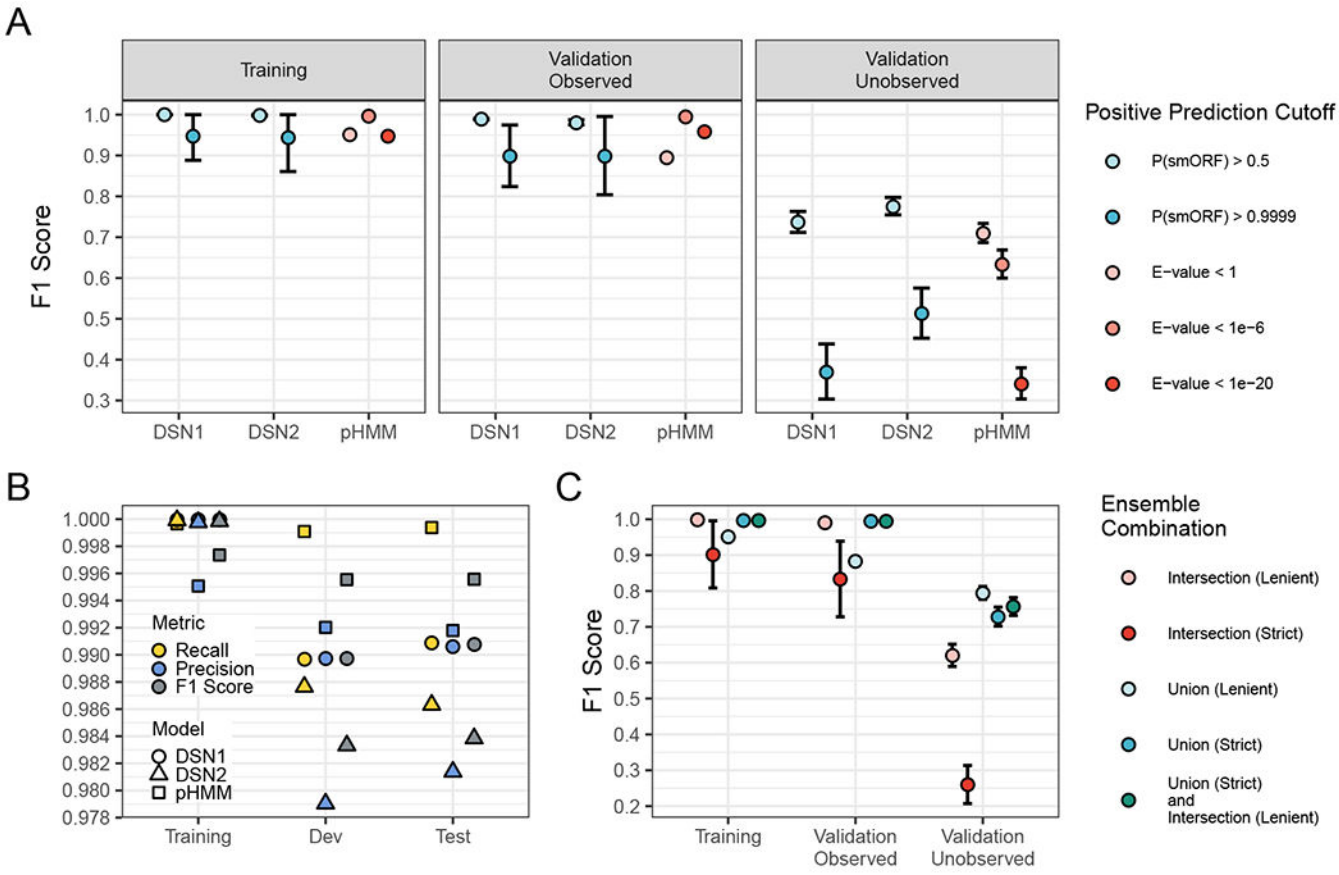


Figure 1: Deep learning models detect unobserved smORF families with greater recall and F1 score than profile HMM models

(A) Each point is the average value after running the training procedure 64 times with randomly selected families excluded from the training set and assigned to the "Validation - Unobserved Families" set. Showing F1 Score (the weighted average of precision and recall) for the three sets with different significant cutoffs. Data are represented as mean ± SEM. (B) The F1 Score, Recall, and Precision of the final DSN1, DSN2, and pHMM models. A positive prediction cutoff of P(ORF) > 0.5 was used for DSN, and a cutoff of E-value < 1e-6 was used for pHMM. (C) The average F1 score of various ensemble model combinations across sets. "Intersection (Lenient)" indicates that all positive predictions met the lenient significance cutoffs (pHMM E-value < 1, DSN1 P(smORF) > 0.5, and DSN2 P(smORF) > 0.5), "Intersection (Strict)" indicates that all positive predictions met the strict significance cutoffs (pHMM E-value < 1e-6, DSN1 P(smORF) > 0.9999, and DSN2 P(smORF) > 0.9999), "Union (Lenient)" indicates that at least one of the three models met the lenient significance cutoffs, and "Union (Strict)" indicates that at least one of the three models met the strict significance cutoffs. See also Figure S1, Figure S2 and Table S2.

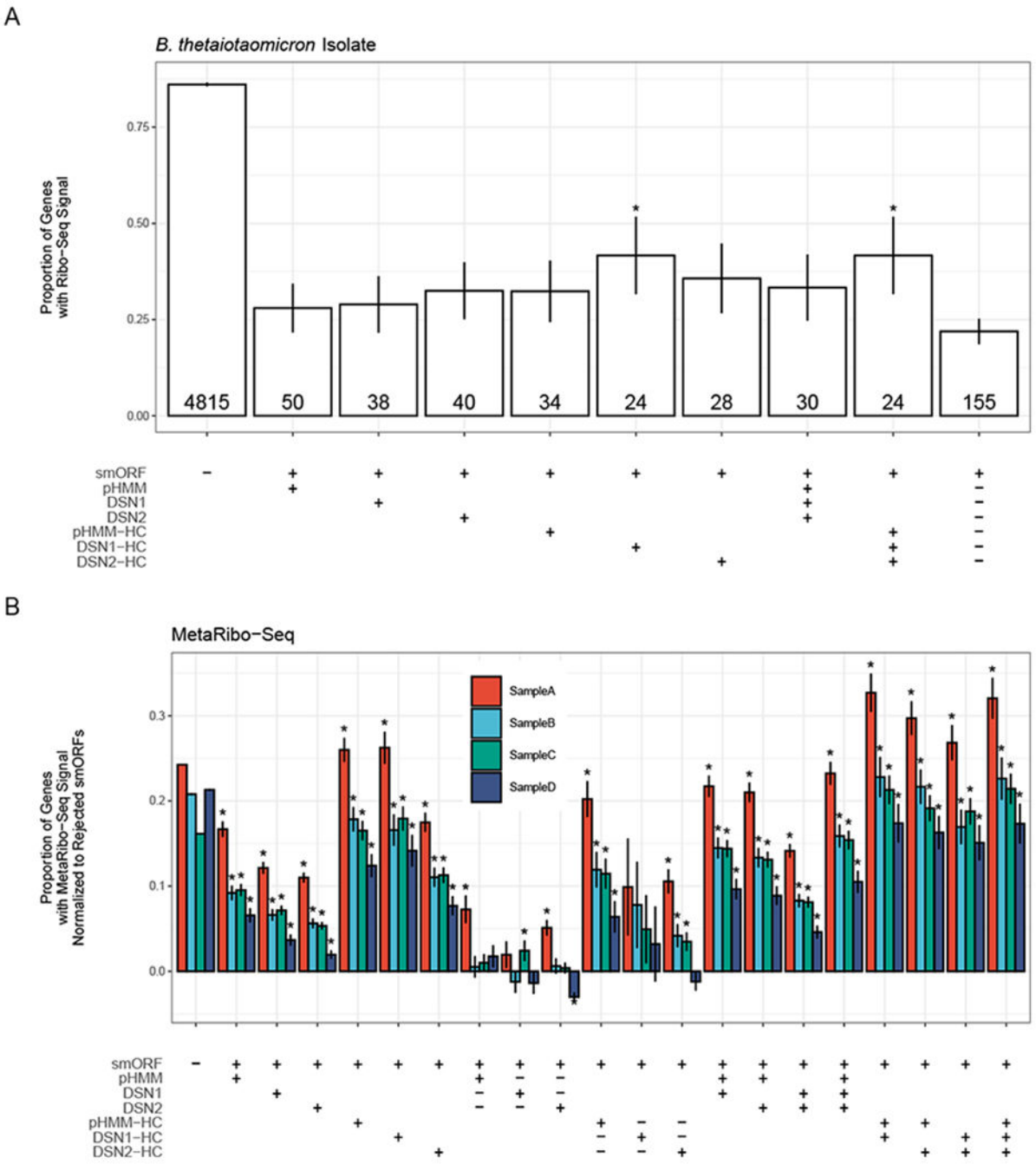


Figure 2: Predicted smORFs are enriched for Ribo-Seq signal

(A) The proportion of genes with Ribo-seq signal (RPKM ≥ 0.5) in different gene sets in a *Bacteroides thetaiotaomicron* isolate. Table along the x-axis denotes the genes included in each set. The label "smORF" indicates if set includes smORFs (+) or only non-smORFs (−), "pHMM" indicates that the set includes pHMM-predicted smORFs at E-value < 1.0 (+), "DSN1" indicates the set includes DSN 1-predicted smORFs at P(ORF) > 0.5 (+), "DSN2" indicates the set includes DSN2-predicted smORFs at P(ORF) > 0.5 (+), "pHMM-HC" indicates the set includes pHMM-predicted smORFs at E-value < 1e-6 (+), "DSN1-HC"

indicates the set includes DSN1-predicted smorfs at P(ORF) > 0.9999, and “DSN2-HC” indicates the set includes DSN2-predicted smorfs at P(ORF) > 0.9999. If multiple “+” symbols are found in a column, then that means all genes in the set meet each cutoff. The symbol “−” indicates that all genes meeting the specified cutoff were excluded from the set. The final column indicates all smORFs that were not predicted by any model to be a true smORF at any significance cutoff. Error bars indicate the standard error of each proportion. These smORFs are referred to as “Rejected smORFs”. The number of total genes in each gene set is given at the bottom of each bar. Asterisks indicate that the proportion is significantly higher ($P < 0.05$) in the specified set than in the “Rejected smORFs” set. (B) The proportion of genes with MetaRibo-seq signal (RPKM ≥ 0.5), normalized to rejected smORF MetaRibo-Seq signal, in different gene sets in four different MetaRibo-seq samples. Normalization was performed by subtracting the proportion of rejected smORFs with a MetaRibo-Seq signal from the proportion of genes in each set with MetaRibo-Seq signal. The x-axis table is the same as the one shown in (A) with additional gene sets added. For example, one additional column is the 8th column from the left, designated by “smORF = +”, “pHMM = +”, “DSN1 = −” and “DSN2 = −” indicates the set of smORFs with predicted by the pHMM model to be a true smORF, but predicted by both DSN1 and DSN2 to be a false smORF. Asterisks indicate that the proportion is significantly higher ($P < 0.05$) in the specified set than in the “Rejected smORFs” set. See also Figure S3.

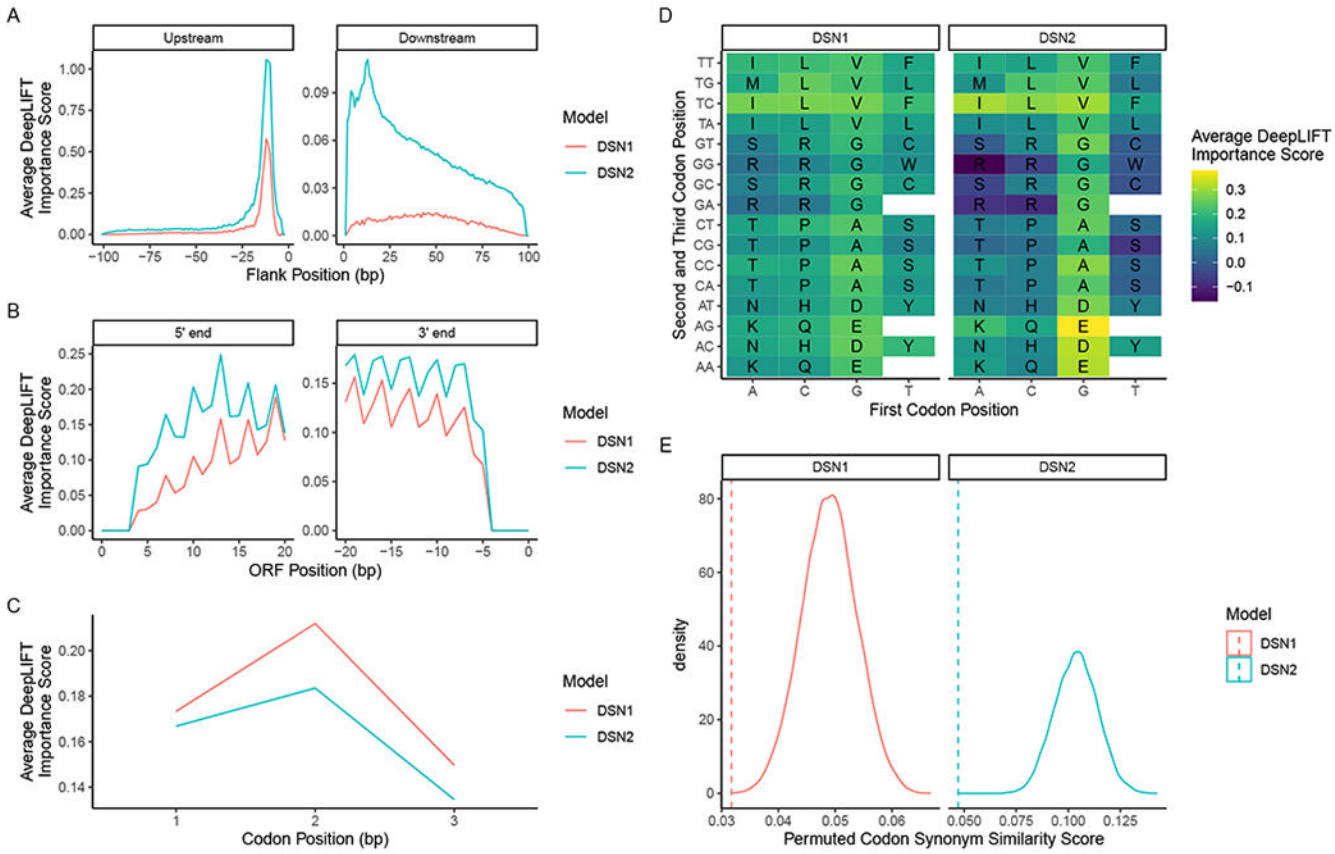


Figure 3: Feature importance analysis reveals inner workings of deep learning models

(A) Average feature importance scores across the 100 bp upstream and downstream of each true smORF example in the training set. Showing the feature importance scores for DSN1 (red) and DSN2 (blue). (B) The average feature importance scores across the first 21 and last 21 bp of all true smORF examples in the training set. (C) The average feature importance scores of the first, second, and third codon position in codons of each true smORF example in the training set. Both models assign higher feature importance to the first two codon positions than the third (wobble) position. (D) The average feature importance scores of each codon in the codon table, excluding stop codons. The nucleotide of the first codon position is on the x-axis, while the second and third positions are shown on the y-axis. (E) The true codon synonym similarity (CSS) score (dotted lines) vs. the distribution of CSS scores (solid line) observed when randomly permuting codon synonym labels across all scores. See also Figure S4.

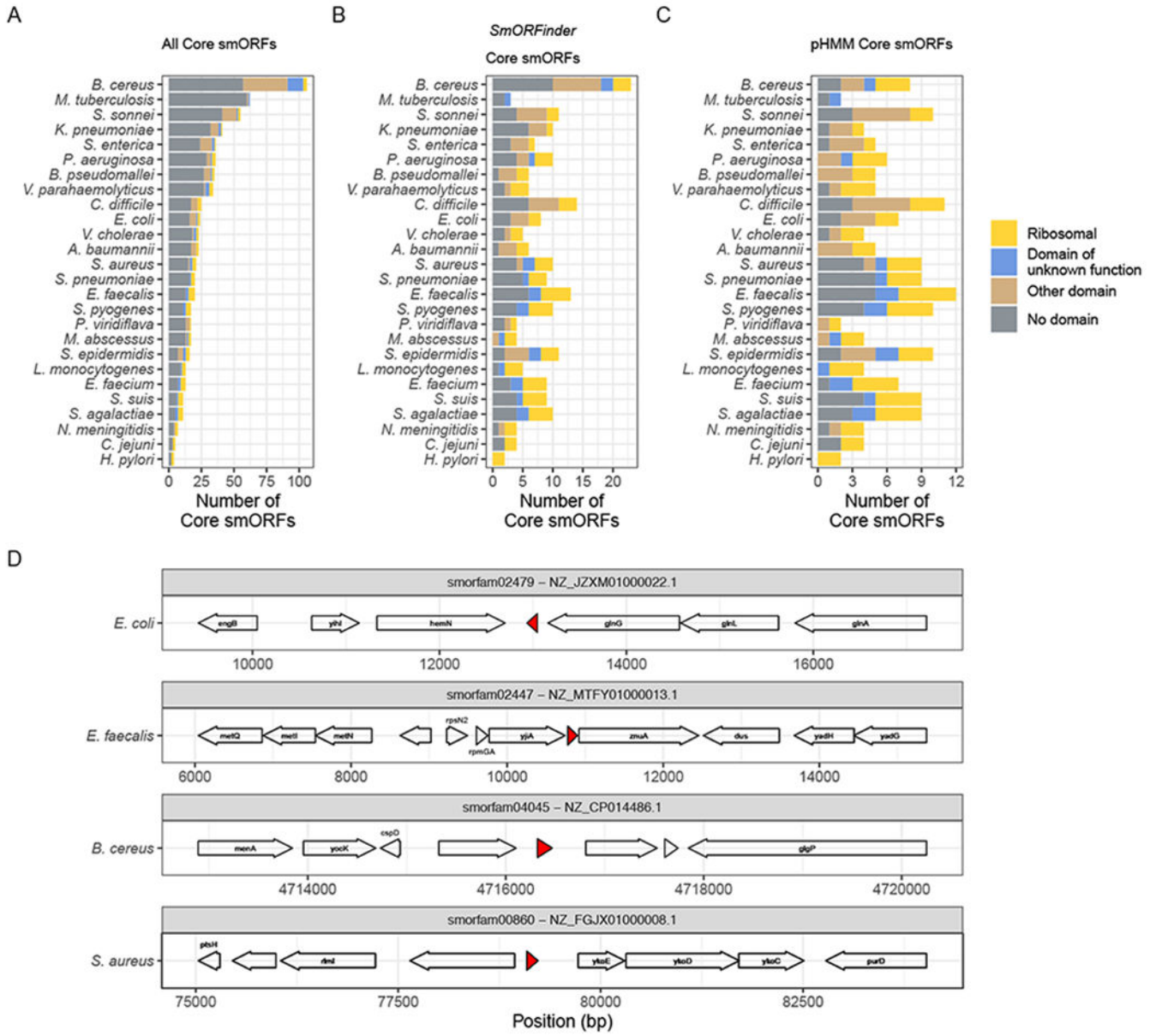


Figure 4: Core genome analysis identifies core smORFs of unknown function

(A) The total number of core smORFs found in each species' genomes. These smORFs were not filtered using DSN, all smORFs identified by the Prodigal annotation tool (with a lowered minimum size cutoff of all smORF predictions greater than 15 nucleotides) were included. (B) The total number of core smORFs identified by the *SmORFinder* annotation tool as being true smORFs. This includes all smORFs that meet strict significance cutoffs for at least one model (pHMM E-value < 1e-6, DSN1 P(smORF) > 0.9999, or DSN2 P(smORF) > 0.9999), or those that meet lenient significance cutoffs for all three models (pHMM E-value < 1, DSN1 P(smORF) > 0.5, and DSN2 P(smORF) > 0.5). (C) The total number of core smORFs per species that meet the pHMM significance cutoff of E-value < 1e-6. Colors indicate if each core smORF contains a Pfam domain (E-value < 1e-6), and which type. "Ribosomal" (yellow) implies a ribosomal protein domain, "Domain of

unknown function" (blue) implies it has a recognized domain of unknown function, "Other domain" (brown) indicates some other Pfam domain, and "No domain" (grey) indicates that it does not contain any known Pfam domain. (D) Four example core smORFs with no recognized Pfam domain that exist in the core genome of two or more species. Arrows indicate ORFs identified by *SmORFinder* or by the Prokka annotation tool. The red regions indicate the position of each core smORF. The text indicates gene names as assigned by Prokka. The absence of any gene name indicates that Prokka identified the genes as "hypothetical" proteins. The species to which each genome belongs is noted to the left of the gene diagram, the smORF family (smorfam) ID and NCBI Reference Sequence ID are given in the strip above each region. See also Table S3.

Key Resources Table

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited Data		
The datasets of smORF examples used to train the models generated in this study.	This study	github.com/bhattlab/SupplementaryInformation/tree/master/SmORFinder
Software and Algorithms		
SmORFinder Software and DBSmORF Web Portal	This study	github.com/bhattlab/SmORFinder
Keras	Chollet and Others, 2015	keras.io
Keras Tuner	O'Malley, 2020	keras-team.github.io/keras-tuner
CD-HIT	Fu et al., 2012	http://weizhongli-lab.org/cd-hit
RNAcode	Washietl et al., 2011	viennarna.github.io/RNAcode
MEME Suite	Bailey et al., 2009	meme-suite.org
uShuffle	Jiang et al., 2008	github.com/guma44/ushuffle
MUSCLE	Edgar, 2004	www.drive5.com/muscle
HMMER	HMMER, no date	hmmer.org
bowtie2	Langmead and Salzberg, 2013	bowtie-bio.sourceforge.net
bedtools	Quinlan and Hall, 2010	bedtools.readthedocs.io
DeepLIFT	Shrikumar, Greenside and Kundaje, 2017	github.com/kundajelab/deeplift
SHAP	Lundberg and Lee, 2017	github.com/slundberg/shap
Prodigal	Hyatt et al., 2010	github.com/hyattpd/Prodigal
Mash	Ondov et al., 2016	mash.readthedocs.io
DIAMOND	Buchfink, Xie and Huson, 2014	github.com/bbuchfink/diamond
Other		
B. thetaiotaomicron Ribo-Seq data	Sberro et al., 2019	BioProject PRJNA540869
MetaRibo-Seq data	Fremin, Sberro and Bhatt, 2020	BioProject PRJNA510123
Publicly available isolates for 26 species	NCBI Assembly, multiple sources	Multiple identifiers, Table S3
Conserved Domains Database	Lu et al., 2020	www.ncbi.nlm.nih.gov/cdd
Pfam database	El-Gebali et al., 2019	pfam.xfam.org