Supplemental Table 1. Hyper-parameter search space.

Hyper-parameter	Search Space
Dense layer size (part 1)	choice from {4, 8, 16, 32, 64, 128}
Dense layer size (part 2)	choice from {4, 8, 16, 32, 64, 128}
Number of Dense layers (part 1)	choice from {1,2}
Number of Dense layers (part 2)	choice from {1, 2, 3}
Number of Convolutional layers set	choice from {1, 2, 3}
Number of filters	choice from {4, 8, 16, 32, 64, 128}
Kernel size	choice from {1, 3, 5, 8, 16, 32}
Pooling size	choice from {3, 5, 8, 16, 32}
Strides	choice from {1, 3, 5, 8}
Dropout rate	choice from {0, 0.05, 0.1, 0.2, 0.3}
Batch size	choice from {16, 32, 64, 128}
Beta1	selection from uniform distribution [0.9, 0.9999]
Beta2	selection from uniform distribution [0.9, 0.9999]
Learning rate	choice from {0.1, 0.01, 0.001, 0.0001}

We have eleven hyper-parameters to define before training models. Batch size, beta1, beta2, and learning rate are hyper-parameters about the model's training and optimizer (Adam: (25)). Dense layer sizes, dense and convolutional layer numbers, number of filters, kernel size, pooling size, strides, and dropout rates define the model's architecture. Possible choices or ranges are provided as search space. Then, the hyper-parameter selection part of the study defines which parameters provide higher performance. Selected hyper-parameters are given in the Bitbucket repository of the paper. More information about parameters can be obtained from the Keras package web site (https://keras.io/). We also provide definitions below:

Number of Dense layers: The number of dense layers. Part 1: Dense layers between convolutional layers and flattening. Part 2: Dense layers after flattening.
Number of Convolutional layers sets: The number of convolutional layers.
Dense layer size: The number of units (neurons) in a dense layer.
Number of filters: The number of filters in a convolutional layer.
Kernel size: Shape/size of the kernels in a convolutional layer.
Pooling size: Pooling size in a pooling layer.
Strides: The number of strides while moving kernels.
Dropout rate: Some neurons are deactivated/dropped out randomly to prevent overfitting. This hyper-parameter defines what percentage of the neurons will be dropped out.

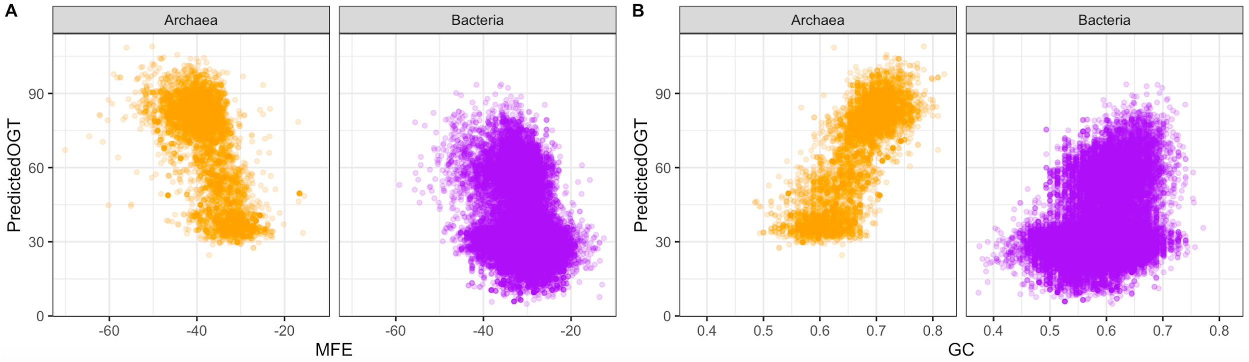
Batch size: The number of training samples utilized in one iteration.

Beta1 and Beta2: Adam optimizer specific parameters.

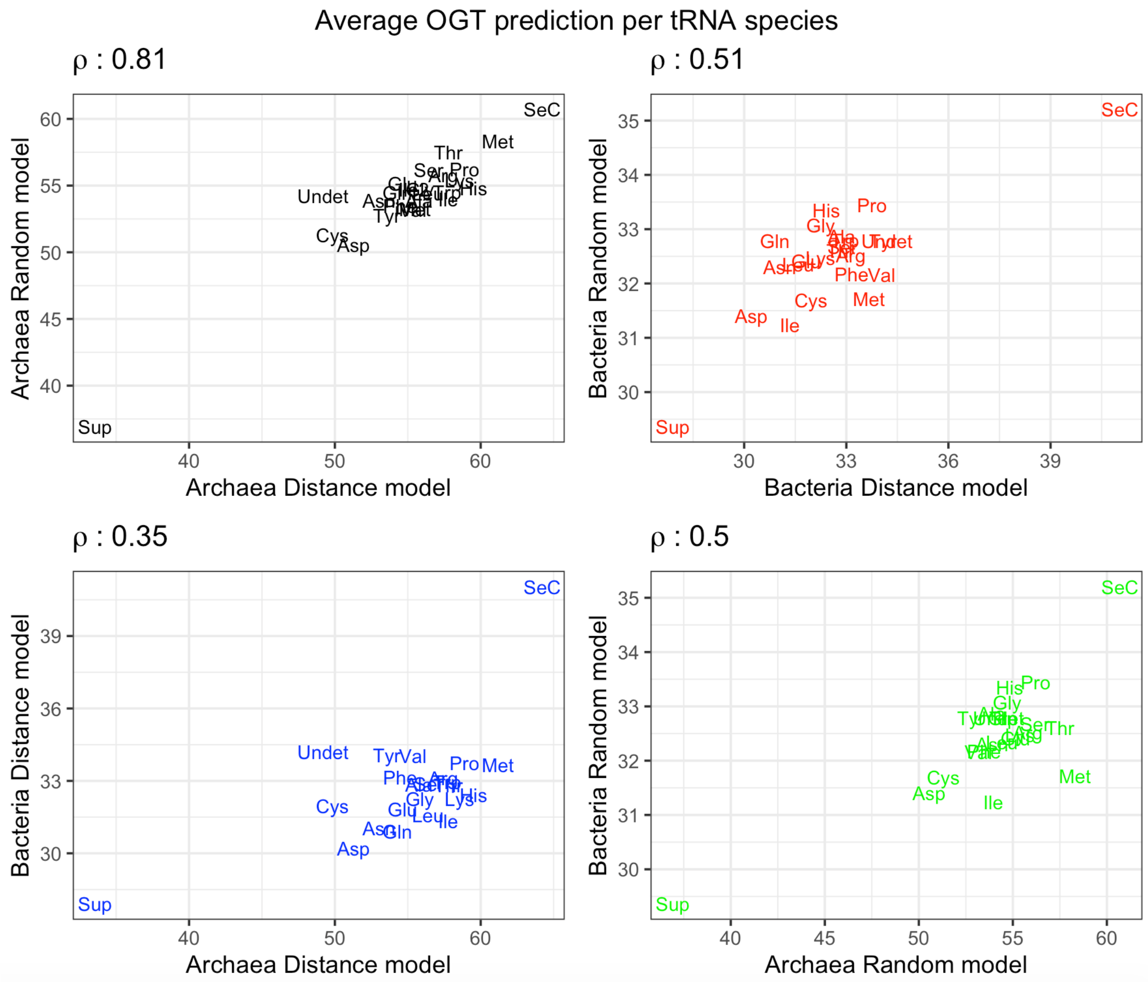
Learning rate: Convergence speed (step size) of optimizer in each iteration.

Supplemental Table 2. Selected Hyper-parameters.

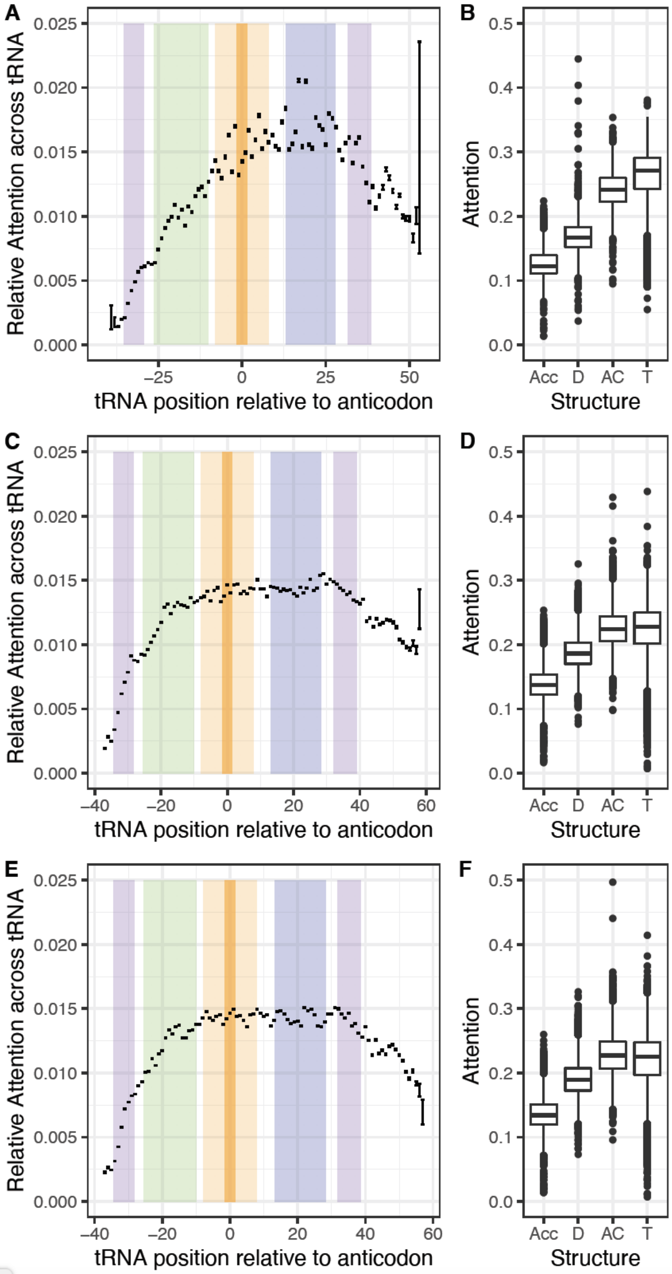
	Regression			Classification		
Hyper-parameter	**Archaea**	**Bacteria**	**Archaea & Bacteria**	**Archaea**	**Bacteria**	**Archaea & Bacteria**
Dense layer size (part 1)	8	128	8	64	64	16
Dense layer size (part 2)	8	128	8	32	16	4
Number of Dense layers (part 1)	2	2	2	1	2	1
Number of Dense layers (part 2)	3	3	3	1	2	2
Number of Convolutional layers set	3	3	3	1	2	1
Number of filters	32	64	32	16	16	8
Kernel size	32	16	32	8	16	16
Pooling size	32	3	32	8	3	3
Strides	8	8	8	1	1	3
Dropout rate	0	0	0	0	0.1	0
Batch size	64	16	64	128	128	16
Beta1	0.964	0.966	0.964	0.974	0.909	0.968
Beta2	0.946	0.986	0.946	0.968	0.912	0.964
Learning rate	0.001	0.001	0.001	0.0001	0.0001	0.0001



Supplemental Figure 1: Components related to secondary structure and folding are strongly correlated with archaea OGT predictions and more weakly correlated with bacteria OGT predictions. Correlation between model predictions of OGT and minimum free energy of folding (A) or GC content (B). Plots in both cases show OGT predictions for each individual tRNA sequence. Predictions from all tRNA in a species were aggregated to predict overall species OGT.



Supplemental Figure 2: Predicted OGT varies for different tRNA species, but the order of the predicted OGT values is largely consistent between models. Spearman rank correlation coefficients are included at the top left of each plot. Selenocysteine and suppressor tRNAs are consistently assigned a higher and lower OGT respectively than canonical tRNAs. Proline, methionine, and histidine tend to have higher OGT predictions than most tRNAs, while cysteine and aspartic acid tend to have lower OGT predictions than other tRNAs.



Supplemental Figure 3: Relative mean and standard error for CNN attention, calculated for each nucleotide across the tRNA (A, C, E) and summed within tRNA structures (B, D, F). Results are consistent across all models (A-B: model trained on archaea with a random split; C-D: model trained on bacteria with a random split; E-F: model trained on bacteria with a phylogenetic split). Colors represent structures: purple=acceptor stem, green=D arm, orange=anticodon arm, blue=T arm. The dark orange band around the x=0 position indicate the anticodon. Nucleotides that are not highlighted fall outside the average positions of each structure.