Deep Learning for Predicting 16S rRNA Gene Copy Number

Jiazheng Miao[1], Tianlai Chen[1], Mustafa Misir[1, *], and Yajuan Lin[1, 2, *]

[1]*Division of Applied and Natural Sciences, Duke Kunshan University, Suzhou, China*

[2]*Deparment of Life Sciences, Texas A&M University-Corpus Christi, Corpus Christi, USA*

*Correspondence:

Mustafa Misir

mustafa.misir@dukekunshan.edu.cn

Yajuan Lin

yajuan.lin@tamucc.edu

22 **ABSTRACT**

23 **Background**

24 Culture-independent 16S rRNA gene metabarcoding is a commonly used method in
25 microbiome profiling. However, this approach can only reflect the proportion of sequencing
26 reads, rather than the actual cell fraction. To achieve more quantitative cell fraction estimates,
27 we need to resolve the 16S gene copy numbers (GCN) for different community members.
28 Currently, there are several bioinformatic tools available to estimate 16S GCN, either based on
29 taxonomy assignment or phylogeny.

30 **Method**

31 Here we develop a novel algorithm, Stacked Ensemble Model (SEM), that estimates 16S
32 GCN directly from the 16S rRNA gene sequence strings, without resolving taxonomy or
33 phylogeny. For accessibility, we developed a public, end-to-end, web-based tool based on the
34 SEM model, named Artificial Neural Network Approximator for 16S rRNA Gene Copy
35 Number (ANNA16).

36 **Results**

37 Based on 27,579 16S rRNA gene sequence data (rrnDB database), we show that ANNA16
38 outperforms the most commonly used 16S GCN prediction algorithms. The prediction error
39 range in the 5-fold cross validation of SEM is completely lower than all other algorithms for
40 the 16S full-length sequence and partially lower at 16S subregions. The final test and a mock
41 community test indicate ANNA16 is more accurate than all currently available tools (i.e.,

42 rrnDB, CopyRighter, PICRUSt2, & PAPRICA). SHAP value analysis indicates ANNA16
43 mainly learns information from rare insertions.

44 **Conclusion**

45 ANNA16 represents a deep learning based 16S GCN prediction tool. Compared to the
46 traditional GCN prediction tools, ANNA16 has a simple structure, faster inference speed
47 without precomputing, and higher accuracy. With increased 16S GCN data in the database,
48 future studies could improve the prediction errors for rare, high-GCN taxa due to current under
49 sampling.

50

51 **1 INTRODUCTION**

52 Culture-independent 16S rRNA metabarcoding is a molecular biology technique that
53 applies DNA sequencing technology to the prokaryotic 16S rRNA gene to identify and quantify
54 the microbial communities present in a sample [1]. This technique has been established as the
55 gold standard for inferring organismal phylogeny and microbial ecology for over 20 years [2].
56 However, a major limitation of 16S rRNA metabarcoding is that next-generation sequencing
57 estimates the proportion of 16S rRNA gene read counts, rather than the true cell counts. Since
58 it is typical for prokaryotes to have more than one copy of the 16S rRNA gene per cell (1 - 21
59 copies/genome) [3], the proportion of 16S rRNA gene read counts (also known as relative 16S
60 abundance) does not accurately reflect the true composition of a microbial community [4, 5].
61 To address this issue and accurately represent the profile of the microbiota, the proportion of
62 16S rRNA gene read counts needs to be weighted by the inverse of 16S gene copy number

63 (GCN). This step yields a corrected microbial composition that represents the proportion of
64 cell counts of each lineage in the community [4].

65 To estimate 16S GCN, researchers typically rely on experimental methods such as whole
66 genome sequencing [3] or competitive PCR [6]. However, these methods can be expensive
67 and/or culture-dependent, and thus experimental GCN estimates are only available for a limited
68 number of species. To overcome this limitation, several bioinformatics tools have been recently
69 developed to predict the GCN of unmeasured species from measured species. Although there
70 are exceptions [7], the basic assumption is that 16S GCN is correlated with phylogenetic
71 distance across species [4, 8], i.e., 16S GCN variation is relatively small among closely related
72 species and increases with phylogenetic distance. This allows 16S GCN to be estimated either
73 from taxonomy or phylogeny [4, 8, 9]. For instance, rrnDB [3] is a well-cited tool that corrects
74 16S GCN based on taxonomy. It calculates the mean 16S GCN of a taxon from the means of
75 its sub-taxa. Another example is PICRUSt2 [10], one correction tool that constructs a
76 phylogenetic tree of the 16S sequence and estimates the GCN of the unmeasured species from
77 its close measured relatives.

78 Deep learning, as a popular machine learning strategy, has been recently applied to learn
79 complex patterns from biological sequencing data [11, 12]. In this study, we explore deep
80 learning as a novel approach for predicting 16S GCN. Deep learning is a branch of machine
81 learning which uses many layers of Artificial Neural Networks (ANN) to learn complex
82 representations of data [13]. The concept of ANN is inspired by the way a human brain learns
83 and processes information through synaptic interactions between neurons [14]. An ANN is a

84 computational model that is constituted by multiple interconnected processing elements—
85 artificial neurons [14]. Neurons are utilized in groups as the layers of a network. Each artificial
86 neuron receives and processes input with a non-linear function, known as the activation
87 function [15]. Collectively, one ANN model learns to solve a specific task by capturing non-
88 linear trends and relationships from the presented data. Beyond these capabilities, ANNs are
89 even competent in disclosing relationships or mechanisms that are still unknown to scientists
90 [16]. Deep learning gets its name by having more than one (hidden) layer connecting the input
91 and output units. Having deeper networks refer to representing complex input-output
92 relationships while accommodating a large amount of data. In that sense, high-dimensional
93 biological data is a popular target for deep learning. For example, various deep learning
94 architectures, like the Convolutional Neural Network, and Recurrent Neural Network, have
95 been applied to microbiome research, including metagenomic-based taxonomy classification
96 [17], 16S rRNA classification [18], host phenotype prediction [19], disease prediction [20], and
97 microbial community predication [21]. Furthermore, deep learning algorithms have been
98 developed to conduct data mining for antibiotic resistance genes [22], antimicrobial peptides
99 [23], and microbial source [24].

100 With the great potential of extracting biological information from DNA sequences
101 demonstrated by the previous studies, this study explores the possibility of estimating GCN
102 directly from the 16S DNA sequence through deep learning. To the best of our knowledge, no
103 deep learning-based method has been utilized for 16S GCN prediction yet. A Stacked Ensemble
104 Model (SEM) is first trained on the 16S full-length sequences and compares the performance

105 to the taxonomy-based and phylogeny-based algorithms. The sequences and GCN data are
106 retrieved from the ribosomal RNA operon copy number database (rrnDB), a publicly available,
107 curated resource for copy number information for bacteria and archaea [3]. Because microbial
108 studies tend to target one to three adjacent hypervariable regions in 16S rRNA [25], this study
109 also trains deep learning models on different combinations of the adjacent hypervariable
110 regions in 16S rRNA and compares the performance with conventional algorithms. Finally, a
111 deep learning-based bioinformatic tool for 16S GCN correction is developed. The tool is
112 compared with existing correction tools to assess the performance. Shapley Additive
113 exPlanations (SHAP), a game theory-based model explanation method [26], is applied to
114 ANNA16 to reveal how this deep learning model understands the 16S rRNA gene sequences.
115

116 **2 MATERIALS AND METHODS**

117 **2.1. Data Source and Preprocessing**

118 16S rRNA gene sequences and the GCN data used to train deep learning models in this
119 study were retrieved from rrnDB, version 5.7 [3]. The dataset includes a total of 20,277 16S
120 gene sequences with unique accession numbers, containing both (+) strands and (-) strands.

121 Firstly, the orientation for all sequences in this dataset were identified using BLASTN
122 algorithm (BLAST+, version 2.11.0) [27]. The first 50 bp of the *E. coli* 16S rRNA gene
123 sequence (+) strand (accession number: GCF_002953035.1), a conserved region on 16S rRNA
124 [28], were used as a template. Sequences with High-Scoring Segment Pairs (HSPs) that had
125 sstart (subject start) > send (subject end) and evalue < 0.01 were identified as (+) strands, and

126 those with sstart < send and evalue < 0.01 were identified as (-) strands. 389 sequences that did
127 not meet these criteria were labeled as unrecognizable and excluded from the dataset. The
128 reverse complement of these (-) strand sequences were then added to the (+) strand sub-dataset.
129 Next, in order to have uniform starting and ending positions, all 16S rRNA gene sequences
130 in the database were further trimmed by HyperEx [29], using the universal full-length 16S
131 primer 27F and primer 1492R (Table S1) [25, 30]. After that, 19,520 sequences remained in
132 this full-length dataset, involving 1,331 genera, with an average sequence length of 1,498.72
133 nucleotides and a standard deviation of 21.41 nucleotides. 16S GCN ranges from 1 to 21
134 copies/genome (Mean = 5.26, SD = 2.74). In addition, seven commonly targeted subregions
135 by metabarcoding, i.e., V1-V2, V1-V3, V3-V4, V4, V4-V5, V6-V8, and V7-V9 [25], were also
136 extracted by HyperEx (primers listed in Table S1) to generate subregion datasets. The full-
137 length and subregional datasets were used as the training datasets.
138 Additionally, during the course of this study, a new version rrnDB 5.8 were release on June
139 23rd, 2022, including 8,059 newly added sequences and their GCN. The added data involves
140 676 genera, and 219 of the genera that are not present in the version 5.7 dataset. The average
141 sequence length is 1,499.79 nucleotides (SD = 21.67), and the range of 16S GCN is 1 - 20
142 copies/genome (Mean = 5.72, SD = 2.85). The newly published data were also processed by
143 BLASTN and HyperEx and produced an independent final test dataset.
144 By conducting a five-fold cross-validation at each region, the initial GCN datasets were
145 first split into training and test sets to compare the performance of SEM and other conventional
146 algorithms. Data in the train datasets were used to pre-train a SEM model, and the model was

147 tested at each region using the test datasets. Existing GCN tools were also applied to the test
148 datasets and their performance were compared with that of SEM. Once an optimal GCN model
149 was selected, the train datasets and test datasets were combined to train the final model. Finally,
150 the independent final test dataset was used to evaluate the robustness and accuracy of the final
151 model.
152

153 **2.2. Method Comparison: Conventional GCN Algorithms and Existing Tools**

154 There are two predominant CGN algorithms: taxonomy-based and phylogeny-based
155 approaches.

156 Taxonomic aggregation (TA) is a commonly used taxonomy-based GCN algorithm [3, 31].
157 The algorithm calculates the mean 16S GCN of a taxon from the GCN means of its sub-taxa.
158 To implement this algorithm, 16S sequences were first classified using the *assignTaxonomy*
159 function in the DADA2 package in R (version 1.21.0) [32]. Two well-established 16S database,
160 RDP (version 18) [33] and Greengenes (version 13.8) [34] were used as the reference for the
161 *assignTaxonomy* function to determine whether the choice of classification database would
162 impact GCN prediction performance. The algorithm was implemented in Python (version 3.8)
163 with an object-oriented programming approach. The algorithm took the taxonomic and GCN
164 data and treated every taxon as a Node object. Every Node is linked with its downstream Nodes
165 (taxa), so eventually the Nodes formed a hierarchical tree where the *Prokaryote domain* is the
166 root, and the species are the tips. Finally, the algorithm pre-computed the GCN of every Node
167 by taking the mean of its sub-Nodes. When a query is given, the algorithm will search the

168 hierarchical tree for the lowest taxon of the query lineage. If a query has unassigned taxon, it
169 assigns the value of the query's parent node to the query node as an estimation.

170 Phylogenetic algorithms were based on a pre-constructed tree of 16S rRNA gene
171 sequences and the GCN of each ancestral node was estimated from its daughter species through
172 weighing each labeled daughter node with phylogenetic distance to the parental node [35]. For
173 method comparison, this study employed five phylogeny-based algorithms, including
174 empirical probabilities (EP), subtree averaging (SA), phylogenetic independent contrasts (PIC)
175 [36], Sankoff's maximum-parsimony (MPR) [37], and weighted-squared-change parsimony
176 (WSCP) [38]. To implement these algorithms, 16S rRNA gene sequences in the dataset were
177 first aligned by MAFFT v7.490 [39] and then a 16S phylogenetic tree was constructed by
178 FastTree 2.1.10 [40]. The five phylogeny-based GCN algorithms were then executed using the
179 castor package in R [35], with modifications made from the R script in [9]. In particular, MPR
180 was implemented by function *hsp_max_parsimony*, WSCP by function
181 *hsp_squared_change_parsimony*, SA by function *hsp_subtree_averaging*, EP by function
182 *hsp_empirical_probabilities*, and PIC by *hsp_independent_contrasts*, all configured to default
183 parameters.

184 Apart from conventional GCN algorithms, the existing GCN tools include rrnDB [3],
185 CopyRighter [4], PAPRICA [41], and PICRUSt2 [10] were also implemented for method
186 comparison. rrnDB is a taxonomy-based tool using taxonomic aggregation, while the latter
187 three are phylogeny-based tools. CopyRighter uses the PIC algorithm to pre-compute the copy
188 number of all taxa in the Greengenes database [34] and makes predictions by searching the

189 lineages of the queries in its pre-computed GCN table [4]. PAPRICA and PICRUSt2 first utilize
190 EPA-ng [42] to place the queries on their built-in phylogenetic tree and then use subtree-
191 averaging (PAPRICA) or maximum parsimony (PICRUSt2) to predict the GCN of the queries
192 [10, 41].

193 Because rrnDB and CopyRighter output GCN corrected microbial community
194 composition rather than 16S GCN values directly, this study extracted their precomputed tables
195 (rrnDB version 5.7) to generate GCN values, searching the precomputed tables for assigned
196 lineages. The taxonomy of full-length 16S sequences and all subregion sequences were
197 generated by the *assignTaxonomy* function in the DADA2 package through RDP (version 18)
198 and Greengenes (version 13.8, formatted by DADA2) [34] database. The assigned RDP
199 lineages were searched in the rrnDB precomputed table, and Greengenes lineages were
200 searched in CopyRighter precomputed table. As for PAPRICA and PICRUSt2, the DNA
201 sequences in the test dataset were directly input to the tools. For PAPRICA, *paprica-run.sh*
202 was called without additional parameters. For PTCRUSt2, *place_seqs.py* and *hsp.py* were
203 called without additional parameters.

204

205 **2.3. Development of ANNA16**

206 2.3.1. DNA sequence K-merization

207 The 16S rRNA gene sequences were first preprocessed with K-merization. An example of
208 a set with three 6-mers is displayed in Figure 1a. The K-mers were converted into the number

of counts in the set via the *CountVectorizer* function in the scikit-learn module (version 0.23.2) [43]. Subsequently, the K-mer counts and GCN data were used to train the SEM models.

After testing K from 3 to 7 on a prototype 3-layer Multi-layer Perceptron (MLP) model, the model performance was found to improve with increasing K. However, a larger K value required significantly more memory. Thus, the optimal K value was determined to be 6, striking a balance between good performance and manageable memory requirements.

2.3.2. Model Construction

Based on 6-mers, the Stacked Ensemble Model (SEM) was constructed for 16S GCN prediction using the Tensorflow package (version 2.8.0) in Python (version 3.8). The initial dataset was split for training and testing by 3:1. The SEM model training was conducted on a Nvidia RTX 3090 GPU. This SEM method involves training multiple base models on the available data, after which a meta-model is used to aggregate the outputs of all base-models and make the final prediction. Typically, the stacked model performs better than any single base model [44, 45].

The first base model of SEM is a Multi-layer Perceptron (MLP). MLP is a type of artificial neural network that consists of layers of artificial neurons, and the input is processed layer by layer [14]. The calculation of the output for each neuron is as follows [14]:

$$\nu_k = \sum_{i=0}^{n} \omega_{k,i} x_i$$

$$\gamma_k = \varphi(\nu_k + \nu_{k_0})$$

229 where v_k is the net output of neuron k, with x_1, x_2, . . ., x_n being the input from the last
230 layer, and $\omega_{k,1}$, $\omega_{k,2}$, . . ., $\omega_{k,n}$ being the weights on the input. γ_k is the output of the neuron,
231 where v_{k_0} represents the bias, and $\varphi(.)$ refers to the activation function. And the MLP part
232 is trained with Rooted Mean Squared Error (RMSE) loss:

233 $$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=0}^{n}(Y_{true} - Y_{pred})^2}{n}}$$

234 where n is the size of the input dataset, Y_{true} is the true value of copy number, Y_{pred} is the
235 value of the predicted copy number.

236 In addition to MLP, the K-mer counts data would undergo a dimension reduction by the
237 Principal Component Analysis (PCA) algorithm, implemented by the *PCA* function in scikit-
238 learn with *n_components* set to 100, and then be input to the several other machine learning
239 models, including Support Vector Machine (SVM) [46], Random Forest [47], K-nearest
240 Neighbors [48], and Gradient Boosting [49]. Linear Regression, Ridge Regression [50], and
241 Lasso Regression [51] were tested as meta-models which accept the output of MLP, PCA and
242 machine learning models and make the final prediction. All machine learning algorithms were
243 implemented by the scikit-learn module (version 0.23.2). After the initial testing, SVM was
244 selected among the base-models and Ridge Regression among the meta-models (Figure 2c).

245

246 2.3.3. Parameter Optimization

247 The parameter of SEM was optimized through the *SMAC4BB* function in the SMAC3
248 package (version 1.4.0) [52] for 500 iterations. For the MLP component, hyperparameters
249 including the number of dense layers, hidden layer size, activation functions, and learning rate

250 were tuned. The number of dense layers of the model was tested in the range of [1, 7], hidden
251 layer size from the range of [1, 1024], activation functions are selected among Rectified Linear
252 Unit (RELU), Exponential Linear Unit (ELU), Gaussian Error Linear Unit (GELU), Scaled
253 Exponential Linear Unit (SELU), and Linear, and the learning rate was tested in the range of
254 [0.00005, 0.1]. The mathematic formulae of the activation functions are demonstrated in the
255 Supplementary Methods.

256 The final MLP architecture (Figure 2d) contains a dense block of 6 hidden layers, with
257 neuron numbers of 489, 926, 645, 929, 582, and 82 respectively, and the activation functions
258 are GELU, ReLU, ReLU, ELU, GELU, and Linear respectively. For SVM, the *kernel* value
259 was chosen from "rbf", "sigmoid", and "linear", *gamma* was tested between "scale" and "auto",
260 and *C* was tested as a float in the range of [0.3, 0.9] or as an integer in the range of [1, 100].
261 For Ridge, the value of *alpha* was tested as a float in the range of [0.3, 0.9] or as an integer in
262 the range of [1, 100]. After tuning, the optimized parameters are *kernel = 'rbf'*, *gamma =*
263 *'auto'*, *C = 11*, and *alpha = 49* (a complete list of parameters is shown in Table S2).

264 The resulted final model is named the Artificial Neural Network Approximator for 16S
265 rRNA Gene Copy Number (ANNA16).

266

267 2.3.4. Other Deep Learning Models Tested

268 In addition to SEM, this study also developed and evaluated a Transformer model and a
269 Residual Multi-layer Perceptron model for 16S GCN prediction. Details about these two

270 models are described in the Supplementary Methods and Figure S1 & S2. However, neither
271 model achieved perform comparable to the SEM and they were not selected.
272

273 **2.4. Mock Community Test**

274 To assess model performance, an *in silico* test study was designed based on two
275 commercially available mock communities [53], one even community (ATCC, MSA-2003)
276 and one staggered community (ATCC, MSA-1001) (Figure 5, Table S3). Both mock
277 communities consist of ten common bacterial strains, but of different ratios. This study
278 retrieved genome sequences of the ten strains from NCBI, and copy number data from rrnDB
279 (version 5.7). An arbitrary total cell number (1×10^5 cells) were assigned to each community
280 and the cell number for each strain was calculated based on the known/targeted fraction.

281 To evaluate the accuracy of community profile prediction by different tools, Bray-Curtis
282 Dissimilarity [54] was calculated between the true cell fraction and the estimated cell fraction
283 using the *vegdist* function in the vegan (version 2.5-7) package.
284

285 **2.5. Model Explanation**

286 In this study, SHAP a unified measure of feature importance that quantifies individual
287 feature contributions to a model prediction, was utilized to explain DNA loci's contributions
288 [26]. Based on collational game theory [55] and the classical Shapley method [56], SHAP
289 views each feature as a player. The final contribution of a feature is the sum of all the marginal
290 contributions across various subsets. Positive and negative SHAP values indicate positive and

291 negative contribution to the model's prediction, respectively. In this study, each K-mer is
292 considered a feature, and its SHAP value is calculated using the *KernelExplainer* function in
293 shap module (version 0.41.0) in Python, treating each single prediction as an independent game.
294 This study randomly sampled 10% of the whole dataset for SHAP value analysis (N =
295 2758). As a result, a list of 2,758 SHAP values was assigned to every K-mer to represent the
296 K-mer's overall contribution. These SHAP values were mapped back to the 16S rRNA gene
297 sequence according to the position of the corresponding K-mers to investigate if there is a
298 positional pattern in contribution. What is more, to study the relationship between contribution
299 and nucleotide conservation in a position, the rate of nucleic substitution, insertion, and
300 deletion of each position was calculated using aligned 16S rRNA gene sequences, with *E. coli*
301 (accession number: GCF_002953035.1) as the reference, to represent how conserved a position.
302 The upper boundary of the plot of SHAP values versus the insertion rate was fitted with the elu
303 function to explore how ANNA16 weighs one insertion. To represent the upper boundary, the
304 insertion rate was first clustered with the *KMeans* function in the scikit-learn package into 50
305 ranks. For each rank, the mean insertion rate and max SHAP values were calculated. Adam
306 optimizer in the tensorflow package were used to fit the elu function with a learning rate of
307 0.01 and 2000 iterations.
308

309 **3 RESULTS**

310 This study evaluated the performance of SEM in comparison with other algorithms for
311 full-length 16S GCN correction. Figure 2 and Table 1 summarize the results of 5-fold cross-

312 validation. Among all algorithms, taxonomic aggregation methods (TA) perform the worst. TA
313 based on RDP or Greengenes reference database exhibit a mean RMSE in copies/genome 1.18
314 (SD = 0.0356) and 2.52 (SD = 0.0388), respectively. Due to the much higher mean RMSE of
315 TA with Greengenes, only results from TA with RDP are shown in Figure 2. All phylogeny-
316 based algorithms perform better than taxonomic aggregation, with WSCP the best phylogeny-
317 based algorithm. SEM, the deep learning model, outperforms all other algorithms with the
318 lowest mean RMSE 0.685 copies/genome, with SD = 0.0379. Its error range on the 16S full-
319 length is significantly lower than all other algorithms ($p < 0.01$).

320 Furthermore, algorithm performance was evaluated and compared for commonly
321 amplified 16S subregions (Figure 2). Compared to the performance on full length, in general
322 all algorithms produce larger RMSE on subregions. SEM displays relatively consistent
323 performance on subregions, ranging from 0.731 (V1-V3) to 0.774 (V4) copies/genome.
324 Notably, except for V3-V4, V4, and V4-V5 of WSCP and V4 of SA, the error range of SEM
325 on subregions is significantly lower than all other algorithms in all subregions ($p < 0.05$). These
326 findings indicate SEM outperforms EP, MPR, PIC, and TA with RDP or Greengenes, while
327 partially outperforms SA, and WSCP.

328 The prediction results produced by SEM from the first cycle of cross-validation are
329 presented in Figure 3 and Figure S3. The figure shows a strong linear relationship between the
330 predicted values and the true values, with density plot showing similar data distribution patterns.
331 However, it is worth noting that SEM tends to underestimate the GCN when the true value is
332 greater than 14. This issue is further discussed in Section 4.

333 Due to SEM's superior performance, a GCN prediction tool, ANNA16, is developed based
334 on SEM and compared with other available tools in the final test, the newly published data in
335 rrnDB version 5.8 (Figure 4 and Table 2). Because PICRRUSt2 has a much higher RMSE (i.e.,
336 larger than 3.4 copies/genome for all regions), it is excluded from Figure 4 and the following
337 case study. ANNA16 achieves the lowest RMSE among all the tools examined, with the
338 smallest RMSE (0.683) at the full-length and the highest RMSE (0.780) at V4. In contrast, the
339 RMSE of CopyRighter ranges from 1.822 (V4-V5) to 2.181 (V7-V9) copies/genome, rrnDB
340 from 1.079 (V4-V5) to 1.207 (V4) copies/genome, PAPRICA from 0.773 (V3-V4) to 1.321
341 (V4-V5), and PICRUSt2 from 3.459 (V4-V5) to 4.731 (V6-V8). ANNA16 outperformed all
342 other tools, demonstrating superior performance at all subregions and full-length, which
343 indicated its superior ability to predict GCN in newly discovered species.

344 To demonstrate the performance of ANNA16 intuitively, a case study of *in silico* mock
345 communities was conducted. Figure 5, Table S4, and Table S5 show uncorrected copy fraction,
346 true cell fraction, and cell fraction (microbial profiles) estimated by ANNA16, rrnDB (version
347 5.7), CopyRighter, and PAPRICA. The difference in uncorrected copy fraction and true cell
348 fraction demonstrates that GCN correction is necessary for obtaining an accurate microbiome
349 profile. The Bray-Curtis Dissimilarity analysis shows that in general the dissimilarity values
350 for the even community are considerably higher than those for the staggered community. The
351 cell fraction estimated by ANNA16 has the smallest dissimilarity value for both even and
352 staggered communities. For the even community, PAPRICA and CopyRighter exhibit very
353 close dissimilarity values and are the second most accurate tools. For the staggered community,

354 PAPRICA is the second most accurate tool, followed by rrnDB, but CopyRighter is the least
355 accurate one. Overall, ANNA16 outperforms all the other tools in terms of similarity patterns
356 with the true cell fraction.

357 To illustrate how ANNA16 makes its predictions, the shap module in Python is used to
358 calculate the SHAP value of each K-mer, which represents the contribution of each K-mer to
359 the prediction. Figure S4 displays the top 20 K-mers ranked by mean absolute SHAP value,
360 while Figure 6 illustrates mean absolute SHAP values of each nucleic position across the full
361 length 16S rRNA gene sequences. The pattern does not fully coincide with the hypervariable
362 regions, which implies the distribution of high SHAP values does not strictly align with the
363 traditionally identified hypervariable regions. To confirm this implication, the relationship
364 between SHAP value and the rate of mutations (substitution, insertion, and deletion) is plotted
365 in Figure 7. Figure 7a and 7b shows that no non-linear pattern could be observed between
366 SHAP values and substitution or deletion, and the Pearson correlation coefficients indicate the
367 linear association is also weak (cor=0.144 or cor=0.248). However, intriguingly, the insertion
368 rate exhibits a non-linear relationship with SHAP values, with high SHAP values reaching its
369 peak when the insertion rate is extremely low. These results imply ANNA16 mainly relies on
370 rare insertions to predict GCN, instead of substitutions.

371

372 **4 DISCUSSION**

373 This study introduces ANNA16, a novel method for predicting 16S GCN based on 16S
374 DNA sequence using a deep learning model. This novel approach outperforms currently

375 available taxonomy- and phylogeny-based algorithms in accuracy, excelling in predicting GCN
376 not only for the 16S full-length sequences but also in all commonly amplified subregions.
377

378 **4.1. Strengths of ANNA16**

379 Compared with existing GCN correction tools, the ANNA16 method owns their strengths
380 and meanwhile avoids the drawbacks. Existing GCN correction tools can be classified into two
381 categories based on their inference strategies: pre-computation or real-time phylogenetic
382 placement. On the one hand, pre-computational tools (i.e., rrnDB & CopyRighter) compute the
383 GCN value for every taxon in the database beforehand, and the tools retrieve the GCN values
384 from the database when a query is given. This strategy has the advantage of short inference
385 time, but is also heavily relay on the reference database, which can lead to errors if used with
386 databases having differences in nomenclature. For example, the SILVA database [57] merges
387 *Allorhizobium*, *Neorhizobium*, *Pararhizobium*, and *Rhizobium* as one genus, named
388 *Allorhizobium-Neorhizobium-Pararhizobium-Rhizobium*, but in RDP these are independent
389 genera.

390 On the other, real-time phylogenetic placement tools (i.e., PAPRICA and PICRUSt2) place
391 a query sequence into their built-in phylogenetic trees to predict the GCN value at runtime.
392 This strategy leads to better prediction performance than pre-computational tools, as shown in
393 Figure 2 and 4, but requires much longer inference time because phylogenetic placement can
394 be computationally extensive. However, the performance of ANNA16 can even be better than
395 real-time phylogenetic placement tools, as shown in Figure 2 and 4. Compared with these tools,

396 ANNA16 exhibits even better performance than real-time phylogenetic placement tools, avoids
397 database-dependence by predicting GCN values directly from 16S rRNA gene sequences, and
398 maintains a relatively short inference time because the prediction can be accelerated by GPUs.
399

400 **4.2. Explaining ANNA16**

401 After proving the excellence of ANNA16, an intriguing question this study contemplates
402 is how a deep learning model interprets the intricate biological/evolutionary information
403 contained within the 16S DNA sequence. Interestingly, Figure 6 and Figure 7 suggest
404 ANNA16's approach is highly different from existing methods.

405 First, the peaks of high SHAP values across the 16S full-length sequence are not aligned
406 with the nine hypervariable regions. As shown in Figure 6, although there are peaks of high
407 SHAP values at the V1, V2, V4, and V5 regions, yet the conserved regions before V1 and
408 between V2, V3, V4, V5, and V6 also contain some loci that are significantly contributory.
409 Those conserved regions even seem more contributory than the remaining variable regions (i.e.,
410 V3 and V6-V9). These results indicate the concept of variability is not equivalent to
411 informativeness for ANNA16.

412 Second, SHAP values shows no association with nucleic substitutions. Previous studies
413 mainly use nucleic substitutions to infer evolutionary information. For example, the fraction of
414 nucleic substitutions between two sequences are commonly used as a representation for
415 evolutionary distance [58], or the substitution rate was used as the metric for studying the
416 positional variability in 16S rRNA gene [59]. However, as shown in Figure 7a, the rooted

417 SHAP values remain at a constant low level (about 0.05) despite the variation in the substitution
418 rate. This means implies that substitutions may not be so important for ANNA16, as every
419 substitution makes the similarly low degree of contribution regardless of the prevalence.

420 Given the specialty of ANNA16, SHAP values was plotted against the deletion and
421 insertion rate (Figure 7b & 7c) to further explore how ANNA16 perceives the information in
422 DNA sequences. The deletion rate demonstrates a similar pattern with SHAP values as the
423 substitution rate, where the rooted SHAP values also remain near 0.05. However, an intriguing
424 L-shaped trend emerges between the insertion rate and the rooted SHAP values. High SHAP
425 values aggregate at the positions with the lowest end of the insertion rate, whereas in other
426 areas, SHAP values also maintain the constant level of 0.05. This pattern suggests that
427 insertions of extremely low frequency are the most informative for ANNA16. Each rare
428 insertion may only exist in a limited number of taxonomic lineages, so they could serve as
429 identifiers of the corresponding lineages for ANNA16. What is more interesting, the upper
430 boundary of Figure 7c appears to be highly similar with the elu activation function used in
431 ANNA16. After using the elu function to fit the upper boundary, 91.17% variability in the
432 SHAP values could be explained (Figure 7d). In general, these results imply that ANNA16
433 might understand one DNA sequence by screening for rare insertions.

434 One potential explanation for these trends in Figure 6 and Figure 7 is that some neurons in
435 the MLP part of ANNA16 are differentiated for detecting the rare insertions, and they are
436 assigned with high weights by other neurons. Similar to biological neurons, artificial neurons
437 also receive input signals and fire an output to other neurons if the stimulus reaches a certain

threshold. Hypothetically, some differentiated neurons might only fire when ANNA16 encounters one insertion whose rarity reaches the threshold (roughly 10^{-2} per site according to Figure 7d). This might result in the elu-shape of the upper boundary in Figure 7c, which places high SHAP values at positions with rare insertions and reduces the SHAP values of the remaining positions to a relatively low level.

These implications may explain why ANNA16 or MLP outperform conventional algorithms. In the aspect of information extraction, traditional algorithms have two drawbacks. First, these algorithms typically weigh every position equally by converting the variance in the sequence-to-sequence similarity, which can underrepresent informative positions. Second, especially for phylogeny-based algorithms, some information could be wasted while inferring the phylogeny as most tree-building methods would ignore alignment gaps [58, 60-63]. Previous studies have proved that indels are likely to experience few homoplasy and may produce more congruent phylogenetic trees if combined with substitutions [58, 60-63]. However, ANNA16 adopts a more nuanced approach, which not only considers the substitutions but also pays more attention to insertions. This allows ANNA16 to more effectively harness the information embedded in the 16S sequences, confirming the necessity of including indels in phylogenetics. Future studies could also consider turning from classic theory-based methods to machine learning, a data-driven and alignment-free approach, for extracting the evolutionary information from the sequences. The effectiveness of ANNA16 also highlights the significant potential of deep learning in genetics. Not only can deep learning models like ANNA16 accurately predict traits based on sequencing data, but by analyzing the

459 contribution of each DNA feature, researchers can also establish connections between specific
460 genes or loci and the trait of interest. For example, by combining deep learning with the SHAP
461 methodology, previous studies have evaluated the contribution of genes for predicting tissue
462 types [64], or the pathway-level contribution for the phenotype of diffuse large B-cell
463 lymphoma [65].
464

465 **4.3. Limitations & Future Studies**

466 Although ANNA16 has shown great performance in 16S GCN prediction, there are several
467 limitations. One limitation is its tendency to underestimate the GCN of species with high GCN.
468 As shown in Figure 3, ANNA16's predictions become increasingly underestimated when the
469 true GCN is greater than or equal to 14. This limitation could be attributed to the imbalanced
470 distribution of GCN in the currently available dataset (Figure S5). Sequences with GCN ≥ 12
471 account for only a small proportion of the whole dataset, and sequences with GCN > 15 are
472 extremely rare. The model could be improved with more new sequence data with high GCN
473 species, as new sequencing techniques are rapidly generating more genomic data. Similar
474 implications are also discussed by [4, 9]. Alternatively, data imbalance methods, such as class
475 re-balancing, information augmentation, and module improvement [66] could be applied to
476 ANNA16 and other future models to address the issue.

477 Furthermore, transfer learning [67, 68] could also be explored as a potential solution. In
478 the framework of transfer learning, a deep learning model is first pre-trained on one or more
479 source tasks to capture general features or representation, and then is fine-tuned to apply the

480 captured knowledge to target tasks [69]. To utilize this strategy, future studies could first collect
481 a large amount of unlabeled 16S rRNA sequences and pre-train a model with an autoregression
482 or autoencoder task, similar to Large Language Models like BERT [70] or GPT [71], and then
483 fine-tune the model on the child task, 16S GCN prediction, which may yield more accurate
484 results.
485

486 **5 CONCLUSIONS**

487 In conclusion, this study developed ANNA16 as a deep learning-based new bioinformatic
488 tool for predicting the gene copy number of 16S rRNA. Compared to existing taxonomy-based
489 and phylogeny-based algorithms, ANNA16 showed significantly lower error ranges at 16S
490 full-length and all 16S subregions in the cross-validation. Furthermore, in the final test and
491 mock community case study, ANNA16 outperformed existing tools such as rrnDB,
492 CopyRighter, PICRUSt2, and PAPRICA. ANNA16's simple structure also offers several
493 advantages, including faster inference speed, independence of pre-computation, and less
494 demand for computing resources. The SHAP analysis implies ANNA16 could utilize the
495 information contained in the 16S rRNA gene sequences more effectively by not only
496 considering substitutions, but also paying more attention to rare insertions. Future studies could
497 focus on strategies such as pre-training the model on unlabeled data and addressing data
498 imbalance to further decrease the error of copy number correction.
499
500

501 **LIST OF ABBREVIATIONS**

502 ANNA16: Artificial Neural Network Approximator for 16S rRNA Gene Copy Number

503 ELU: Exponential Linear Unit

504 EP: Empirical Probabilities

505 GCN: Gene Copy Number

506 GELU: Gaussian Error Linear Unit

507 MLP: Multi-layer Perceptron

508 MPR: Sankoff's Maximum Parsimony

509 PIC: Phylogenetic Independent Contrasts

510 RELU: Rectified Linear Unit

511 SA: Subtree Averaging

512 SELU: Scaled Exponential Linear Unit

513 SEM: Stacked Ensemble Model

514 SHAP: Shapley Additive exPlanation

515 TA: Taxonomic Aggregation

516 WSCP: Weighted Squared Change Parsimony

517

518 **DECLARATIONS**

519 ***Ethics approval and consent to participate***

520 Not applicable.

521 ***Consent for publication***

522 Not applicable.

523 ***Availability of data and materials***

524 16S rRNA sequences and the GCN data analyzed during the current study are available in the
525 rrnDB repository, https://rrndb.umms.med.umich.edu/. The codes and model files of ANNA16
526 are available at https://github.com/Merlinaphist/ANNA16. ANNA16 is deployed on Google
527 Colab,
528 https://colab.research.google.com/drive/1XwpTMCHSfTmzpHyKrmiD8aC8C_1nndUV#scro
529 llTo=M0jLLsHC9W4M.

530 ***Competing interests***

531 This study receives funding from the Oracle Cloud Starter Award by the Oracle Corporation.

532 ***Funding***

533 This work was supported by the Summer Research Scholars (SRS) program at Duke Kunshan
534 University (DKU), the startup funds of Y.L. and M.M. granted by DKU, and the Oracle Cloud
535 Starter Award by the Oracle Corporation.

536 ***Authors' contributions***

537 J.M. conceived the study. J.M. and T.C. analyzed the data and implemented the models. Y.L.
538 and M.M. provided mentorship for the study. All authors contributed to the manuscript writing.

539 ***Acknowledgments***

540 The authors would like to thank Dr. Jiang Long for his kind comments on algorithm design.

541

542

543 **REFERENCES**

544 1. Woese CR, Fox GE: **Phylogenetic structure of the prokaryotic domain: the primary**
545 **kingdoms**. *Proc Natl Acad Sci U S A* 1977, **74**(11):5088-5090.

546 2. Woo PC, Leung PK, Leung KW, Yuen KY: **Identification by 16S ribosomal RNA**
547 **gene sequencing of an Enterobacteriaceae species from a bone marrow transplant**
548 **recipient**. *Mol Pathol* 2000, **53**(4):211-215.

549 3. Stoddard SF, Smith BJ, Hein R, Roller BR, Schmidt TM: **rrnDB: improved tools for**
550 **interpreting rRNA gene abundance in bacteria and archaea and a new foundation**
551 **for future development**. *Nucleic Acids Res* 2015, **43**(Database issue):D593-598.

552 4. Angly FE, Dennis PG, Skarshewski A, Vanwonterghem I, Hugenholtz P, Tyson GW:
553 **CopyRighter: a rapid tool for improving the accuracy of microbial community**
554 **profiles through lineage-specific gene copy number correction**. *Microbiome* 2014,
555 **2**(1):11.

556 5. Vandeputte D, Kathagen G, D'hoe K, Vieira-Silva S, Valles-Colomer M, Sabino J,
557 Wang J, Tito RY, De Commer L, Darzi Y *et al*: **Quantitative microbiome profiling**
558 **links gut community variation to microbial load**. *Nature* 2017, **551**(7681):507-511.

559 6. Fogel GB, Collins CR, Li J, Brunk CF: **Mini-Review Article: Prokaryotic Genome**
560 **Size and SSU rDNA Copy Number: Estimation of Microbial Relative Abundance**
561 **from a Mixed Population**. *Microbial Ecology* 1999, **38**(2):93-113.

562 7. Lavrinienko A, Jernfors T, Koskimäki JJ, Pirttilä AM, Watts PC: **Does Intraspecific**
563 **Variation in rDNA Copy Number Affect Analysis of Microbial Communities?**
564 *Trends in Microbiology* 2021, **29**(1):19-27.
565 8. Kembel SW, Wu M, Eisen JA, Green JL: **Incorporating 16S Gene Copy Number**
566 **Information Improves Estimates of Microbial Diversity and Abundance**. *PLOS*
567 *Computational Biology* 2012, **8**(10):e1002743.
568 9. Louca S, Doebeli M, Parfrey LW: **Correcting for 16S rRNA gene copy numbers in**
569 **microbiome surveys remains an unsolved problem**. *Microbiome* 2018, **6**(1):41.
570 10. Douglas GM, Maffei VJ, Zaneveld JR, Yurgel SN, Brown JR, Taylor CM, Huttenhower
571 C, Langille MGI: **PICRUSt2 for prediction of metagenome functions**. *Nature*
572 *Biotechnology* 2020, **38**(6):685-688.
573 11. Jurtz VI, Johansen AR, Nielsen M, Almagro Armenteros JJ, Nielsen H, Sønderby CK,
574 Winther O, Sønderby SK: **An introduction to deep learning on biological sequence**
575 **data: examples and solutions**. *Bioinformatics* 2017, **33**(22):3685-3690.
576 12. Park Y, Kellis M: **Deep learning for regulatory genomics**. *Nature Biotechnology* 2015,
577 **33**(8):825-826.
578 13. Dechter R: **Learning while searching in constraint-satisfaction-problems**. In:
579 *Proceedings of the Fifth AAAI National Conference on Artificial Intelligence;*
580 *Philadelphia, Pennsylvania*. AAAI Press 1986: 178–183.

581 14. Lancashire LJ, Lemetre C, Ball GR: **An introduction to artificial neural networks in**
582 **bioinformatics—application to complex microarray and mass spectrometry**
583 **datasets in cancer studies**. *Briefings in Bioinformatics* 2009, **10**(3):315-329.
584 15. Schmidt B, Hildebrandt A: **Deep learning in next-generation sequencing**. *Drug*
585 *Discov Today* 2021, **26**(1):173-180.
586 16. Stepniewska-Dziubinska MM, Zielenkiewicz P, Siedlecki P: **Development and**
587 **evaluation of a deep learning model for protein–ligand binding affinity prediction**.
588 *Bioinformatics* 2018, **34**(21):3666-3674.
589 17. Fioravanti D, Giarratano Y, Maggio V, Agostinelli C, Chierici M, Jurman G, Furlanello
590 C: **Phylogenetic convolutional neural networks in metagenomics**. *BMC*
591 *Bioinformatics* 2018, **19**(2):49.
592 18. Zhao Z, Woloszynek S, Agbavor F, Mell JC, Sokhansanj BA, Rosen GL: **Learning,**
593 **visualizing and exploring 16S rRNA structure using an attention-based deep**
594 **neural network**. In: *PLoS computational biology.* vol. 17; 2021: e1009345.
595 19. Lo C, Marculescu R: **MetaNN: accurate classification of host phenotypes from**
596 **metagenomic data using neural networks**. *BMC Bioinformatics* 2019, **20**(12):314.
597 20. Wang Y, Bhattacharya T, Jiang Y, Qin X, Wang Y, Liu Y, Saykin AJ, Chen L: **A novel**
598 **deep learning method for predictive modeling of microbiome data**. *Brief Bioinform*
599 2021, **22**(3).
600 21. Zha Y, Chong H, Qiu H, Kang K, Dun Y, Chen Z, Cui X, Ning K: **Ontology-aware**
601 **deep learning enables ultrafast and interpretable source tracking among sub-**

602 **million microbial community samples from hundreds of niches**. *Genome Medicine*
603 2022, **14**(1):43.
604 22. Arango-Argoty G, Garner E, Pruden A, Heath LS, Vikesland P, Zhang L: **DeepARG:**
605 **a deep learning approach for predicting antibiotic resistance genes from**
606 **metagenomic data**. *Microbiome* 2018, **6**(1):23.
607 23. Ma Y, Guo Z, Xia B, Zhang Y, Liu X, Yu Y, Tang N, Tong X, Wang M, Ye X *et al*:
608 **Identification of antimicrobial peptides from the human gut microbiome using**
609 **deep learning**. *Nature Biotechnology* 2022, **40**(6):921-931.
610 24. Chong H, Zha Y, Yu Q, Cheng M, Xiong G, Wang N, Huang X, Huang S, Sun C, Wu S
611 *et al*: **EXPERT: transfer learning-enabled context-aware microbial community**
612 **classification**. *Briefings in Bioinformatics* 2022:bbac396.
613 25. Abellan-Schneyder I, Matchado MS, Reitmeier S, Sommer A, Sewald Z, Baumbach J,
614 List M, Neuhaus K: **Primer, Pipelines, Parameters: Issues in 16S rRNA Gene**
615 **Sequencing**. *mSphere* 2021, **6**(1).
616 26. Lundberg SM, Lee S-I: **A Unified Approach to Interpreting Model Predictions**. In:
617 *Advances in Neural Information Processing Systems: 2017*: Edited by Guyon I,
618 Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R. Curran
619 Associates, Inc. 2017.
620 27. Camacho C, Coulouris G, Avagyan V, Ma N, Papadopoulos J, Bealer K, Madden TL:
621 **BLAST+: architecture and applications**. *BMC Bioinformatics* 2009, **10**(1):421.

622 28. Yarza P, Yilmaz P, Pruesse E, Glöckner FO, Ludwig W, Schleifer K-H, Whitman WB,
623 Euzéby J, Amann R, Rosselló-Móra R: **Uniting the classification of cultured and**
624 **uncultured bacteria and archaea using 16S rRNA gene sequences**. *Nature Reviews*
625 *Microbiology* 2014, **12**(9):635-645.
626 29. Ebou A, Koua D, Zeze A: **HyperEx: A Tool to Extract Hypervariable Regions from**
627 **16S rRNA Sequencing Data**. *bioRxiv* 2021:2021.2009.2003.455391.
628 30. Parada AE, Needham DM, Fuhrman JA: **Every base matters: assessing small subunit**
629 **rRNA primers for marine microbiomes with mock communities, time series and**
630 **global field samples**. *Environmental Microbiology* 2016, **18**(5):1403-1414.
631 31. Sugihara G, Bersier LF, Schoenly K: **Effects of taxonomic and trophic aggregation**
632 **on food web properties**. *Oecologia* 1997, **112**(2):272-284.
633 32. Callahan BJ, McMurdie PJ, Rosen MJ, Han AW, Johnson AJ, Holmes SP: **DADA2:**
634 **High-resolution sample inference from Illumina amplicon data**. *Nat Methods* 2016,
635 **13**(7):581-583.
636 33. Cole JR, Wang Q, Fish JA, Chai B, McGarrell DM, Sun Y, Brown CT, Porras-Alfaro
637 A, Kuske CR, Tiedje JM: **Ribosomal Database Project: data and tools for high**
638 **throughput rRNA analysis**. *Nucleic Acids Research* 2013, **42**(D1):D633-D642.
639 34. DeSantis TZ, Hugenholtz P, Larsen N, Rojas M, Brodie EL, Keller K, Huber T, Dalevi
640 D, Hu P, Andersen GL: **Greengenes, a chimera-checked 16S rRNA gene database**
641 **and workbench compatible with ARB**. *Appl Environ Microbiol* 2006, **72**(7):5069-
642 5072.

643 35. Louca S, Doebeli M: **Efficient comparative phylogenetics on large trees**.
644 *Bioinformatics* 2017, **34**(6):1053-1055.
645 36. Felsenstein J: **Phylogenies and the Comparative Method**. *The American Naturalist*
646 1985, **125**(1):1-15.
647 37. Sankoff D: **Minimal Mutation Trees of Sequences**. *SIAM Journal on Applied*
648 *Mathematics* 1975, **28**(1):35-42.
649 38. Maddison WP: **Squared-Change Parsimony Reconstructions of Ancestral States**
650 **for Continuous-Valued Characters on a Phylogenetic Tree**. *Systematic Zoology*
651 1991, **40**(3):304-314.
652 39. Katoh K, Misawa K, Kuma Ki, Miyata T: **MAFFT: a novel method for rapid multiple**
653 **sequence alignment based on fast Fourier transform**. *Nucleic Acids Research* 2002,
654 **30**(14):3059-3066.
655 40. Price MN, Dehal PS, Arkin AP: **FastTree 2 – Approximately Maximum-Likelihood**
656 **Trees for Large Alignments**. *PLOS ONE* 2010, **5**(3):e9490.
657 41. Bowman JS, Ducklow HW: **Microbial Communities Can Be Described by**
658 **Metabolic Structure: A General Framework and Application to a Seasonally**
659 **Variable, Depth-Stratified Microbial Community from the Coastal West Antarctic**
660 **Peninsula**. *PLOS ONE* 2015, **10**(8):e0135868.
661 42. Barbera P, Kozlov AM, Czech L, Morel B, Darriba D, Flouri T, Stamatakis A: **EPA-ng:**
662 **Massively Parallel Evolutionary Placement of Genetic Sequences**. *Systematic*
663 *Biology* 2019, **68**(2):365-369.

43. Buitinck L, Louppe G, Blondel M, Pedregosa F, Mueller A, Grisel O, Niculae V, Prettenhofer P, Gramfort A, Grobler J *et al*: **API design for machine learning software: experiences from the scikit-learn project**. *CoRR* 2013, **abs/1309.0238**.

44. Wolpert DH: **Stacked generalization**. *Neural Networks* 1992, **5**(2):241-259.

45. Breiman L: **Stacked regressions**. *Machine Learning* 1996, **24**(1):49-64.

46. Cortes C, Vapnik V: **Support-vector networks**. *Machine Learning* 1995, **20**(3):273-297.

47. Ho TK: **Random Decision Forests**. In: *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1: 1995*. M: IEEE Computer Society 1995: 278-282.

48. Cover TM, Hart PE: **Nearest neighbor pattern classification**. *IEEE Trans Inf Theory* 1967, **13**:21-27.

49. Friedman JH: **Greedy function approximation: a gradient boosting machine**. *Annals of statistics* 2001:1189-1232.

50. Hoerl AE, Kennard RW: **Ridge Regression: Biased Estimation for Nonorthogonal Problems**. *Technometrics* 1970, **12**(1):55-67.

51. Tibshirani R: **Regression Shrinkage and Selection via the Lasso**. *Journal of the Royal Statistical Society Series B (Methodological)* 1996, **58**(1):267-288.

52. Lindauer M, Eggensperger K, Feurer M, Biedenkapp A, Deng D, Benjamins C, Ruhkopf T, Sass R, Hutter F: **SMAC3: A Versatile Bayesian Optimization Package for Hyperparameter Optimization**. In: *ArXiv: 210909831: 2021*. 2021.

685 53. Moriya S: **Simple mapping-based quantification of a mock microbial community**
686 **using total RNA-seq data**. *PLoS One* 2021, **16**(7):e0254556.
687 54. Bray JR, Curtis JT: **An Ordination of the Upland Forest Communities of Southern**
688 **Wisconsin**. *Ecological Monographs* 1957, **27**(4):325-349.
689 55. Shapley LS: **A Value for N-Person Games**. Santa Monica, CA: RAND Corporation;
690 1952.
691 56. Štrumbelj E, Kononenko I: **Explaining prediction models and individual predictions**
692 **with feature contributions**. *Knowledge and Information Systems* 2014, **41**(3):647-665.
693 57. Quast C, Pruesse E, Yilmaz P, Gerken J, Schweer T, Yarza P, Peplies J, Glöckner FO:
694 **The SILVA ribosomal RNA gene database project: improved data processing and**
695 **web-based tools**. *Nucleic Acids Research* 2012, **41**(D1):D590-D596.
696 58. Birth N, Dencker T, Morgenstern B: **Insertions and deletions as phylogenetic signal**
697 **in an alignment-free context**. *PLoS Comput Biol* 2022, **18**(8):e1010303.
698 59. Van de Peer Y, Chapelle S, De Wachter R: **A quantitative map of nucleotide**
699 **substitution rates in bacterial rRNA**. *Nucleic Acids Res* 1996, **24**(17):3381-3391.
700 60. Lloyd DG, Calder VL: **Multi-residue gaps, a class of molecular characters with**
701 **exceptional reliability for phylogenetic analyses**. *Journal of Evolutionary Biology*
702 1991, **4**(1):9-21.
703 61. Simmons MP, Ochoterena H: **Gaps as Characters in Sequence-Based Phylogenetic**
704 **Analyses**. *Systematic Biology* 2000, **49**(2):369-381.

705 62. Houde P, Braun EL, Narula N, Minjares U, Mirarab S: **Phylogenetic Signal of Indels**
706 **and the Neoavian Radiation**. *Diversity* 2019, **11**(7).
707 63. Yuri T, Kimball RT, Harshman J, Bowie RC, Braun MJ, Chojnowski JL, Han KL,
708 Hackett SJ, Huddleston CJ, Moore WS *et al*: **Parsimony and model-based analyses**
709 **of indels in avian nuclear genes reveal congruent and incongruent phylogenetic**
710 **signals**. *Biology (Basel)* 2013, **2**(1):419-444.
711 64. Yap M, Johnston RL, Foley H, MacDonald S, Kondrashova O, Tran KA, Nones K,
712 Koufariotis LT, Bean C, Pearson JV *et al*: **Verifying explainability of a deep learning**
713 **tissue classifier trained on RNA-seq data**. *Scientific Reports* 2021, **11**(1):2641.
714 65. Hayakawa J, Seki T, Kawazoe Y, Ohe K: **Pathway importance by graph**
715 **convolutional network and Shapley additive explanations in gene expression**
716 **phenotype of diffuse large B-cell lymphoma**. *PLoS One* 2022, **17**(6):e0269570.
717 66. Zhang Y, Kang B, Hooi B, Yan S, Feng J: **Deep Long-Tailed Learning: A Survey**.
718 *CoRR* 2021, **abs/2110.04596**.
719 67. Pan SJ, Yang Q: **A Survey on Transfer Learning**. *IEEE Transactions on Knowledge*
720 *and Data Engineering* 2010, **22**(10):1345-1359.
721 68. Thrun S, Pratt L: **Learning to Learn: Introduction and Overview**. In: *Learning to*
722 *Learn.* Edited by Thrun S, Pratt L. Boston, MA: Springer US; 1998: 3-17.
723 69. Han X, Zhang Z, Ding N, Gu Y, Liu X, Huo Y, Qiu J, Yao Y, Zhang A, Zhang L *et al*:
724 **Pre-trained models: Past, present and future**. *AI Open* 2021, **2**:225-250.

725 70. Devlin J, Chang M-W, Lee K, Toutanova K: **Bert: Pre-training of deep bidirectional**
726 **transformers for language understanding**. *arXiv preprint arXiv:1810 04805* 2018.
727 71. Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, Neelakantan A, Shyam
728 P, Sastry G, Askell A *et al*: **Language Models are Few-Shot Learners**. *CoRR* 2020,
729 **abs/2005.14165**.
730

731 **TABLES**

732 **Table 1. Performance of all algorithms at 16S full-length and subregions (Mean RMSE ± SD, *p*-value).**

	Full Length	V1-V2	V1-V3	V3-V4	V4	V4-V5	V6-V8	V7-V9
SEM	0.685 ± 0.0379	0.751 ± 0.0285	0.727 ± 0.0301	0.730 ± 0.0450	0.774 ± 0.0453	0.746 ± 0.0364	0.733 ± 0.0368	0.737 ± 0.0423
EP	0.845 ± 0.0443 $p < 0.001$	0.928 ± 0.0331 $p < 0.001$	0.889 ± 0.0476 $p < 0.001$	0.902 ± 0.0447 $p < 0.001$	0.916 ± 0.0569 $p < 0.01$	0.910 ± 0.0340 $p < 0.001$	0.909 ± 0.0563 $p < 0.001$	0.931 ± 0.0354 $p < 0.001$
MPR	0.787 ± 0.0375 $p < 0.01$	0.867 ± 0.0370 $p < 0.001$	0.836 ± 0.0305 $p < 0.001$	0.858 ± 0.0477 $p < 0.01$	0.879 ± 0.0475 $p < 0.01$	0.858 ± 0.0413 $p < 0.001$	0.852 ± 0.0400 $p < 0.001$	0.846 ± 0.0370 $p < 0.01$
PIC	0.789 ± 0.0243 $p < 0.001$	0.810 ± 0.0281 $p < 0.01$	0.805 ± 0.0307 $p < 0.01$	0.789 ± 0.0503 $p < 0.05$	0.824 ± 0.0377 $p < 0.05$	0.785 ± 0.0277 $p < 0.05$	0.781 ± 0.0404 $p < 0.05$	0.802 ± 0.0393 $p < 0.05$
SA	0.774 ± 0.0329 $p < 0.01$	0.829 ± 0.0321 $p < 0.01$	0.805 ± 0.0430 $p < 0.01$	0.811 ± 0.0379 $p < 0.01$	0.822 ± 0.0388 $p = 0.054$	0.810 ± 0.0283 $p < 0.01$	0.807 ± 0.0391 $p < 0.01$	0.830 ± 0.0334 $p < 0.01$
WSCP	0.775 ± 0.0273 $p < 0.05$	0.793 ± 0.0242 $p < 0.05$	0.782 ± 0.0330 $p < 0.05$	0.779 ± 0.0483 $p = 0.067$	0.823 ± 0.0405 $p = 0.054$	0.785 ± 0.0354 $p = 0.062$	0.779 ± 0.0377 $p < 0.05$	0.790 ± 0.0375 $p < 0.05$
TA(RDP)	1.18 ± 0.0356 $p < 0.001$	1.58 ± 0.0538 $p < 0.001$	1.41 ± 0.0517 $p < 0.001$	1.47 ± 0.0370 $p < 0.001$	1.93 ± 0.0954 $p < 0.001$	1.48 ± 0.0622 $p < 0.001$	1.44 ± 0.0329 $p < 0.001$	1.73 ± 0.129 $p < 0.001$
TA(Greengenes)	2.52 ± 0.0388 $p < 0.001$	2.08 ± 0.122 $p < 0.001$	2.63 ± 0.147 $p < 0.001$	1.85 ± 0.0939 $p < 0.001$	2.12 ± 0.0590 $p < 0.001$	2.00 ± 0.0961 $p < 0.001$	2.19 ± 0.232 $p < 0.001$	2.54 ± 0.434 $p < 0.001$

733 EP: empirical probabilities; MPR: Sankoff's maximum-parsimony; PIC: phylogenetic independent contrast; SA: Subtree Averaging; WSCP:
734 weighted-squared-change parsimony; TA(RDP): taxonomic aggregation based on RDP database; TA(Greengenes): taxonomic aggregation based
735 on Greengenes database; SEM: Stacked Ensemble Model.

736

737

Table 2. Performance of 16S GCN Correction Tools in the Final Test (RMSE)

	Full Length	V1-V2	V1-V3	V3-V4	V4	V4-V5	V6-V8	V7-V9
CopyRighter	1.994	1.934	1.989	1.825	1.848	1.822	1.952	2.181
rrnDB5.7	1.083	1.094	1.082	1.101	1.207	1.079	1.109	1.110
ANNA16	0.683	0.711	0.710	0.716	0.780	0.715	0.720	0.713
PICRUSt2	4.198	4.353	4.175	4.393	3.807	3.459	4.731	4.677
PAPRICA	0.817	0.857	1.319	0.773	0.896	1.321	0.987	0.778

FIGURES

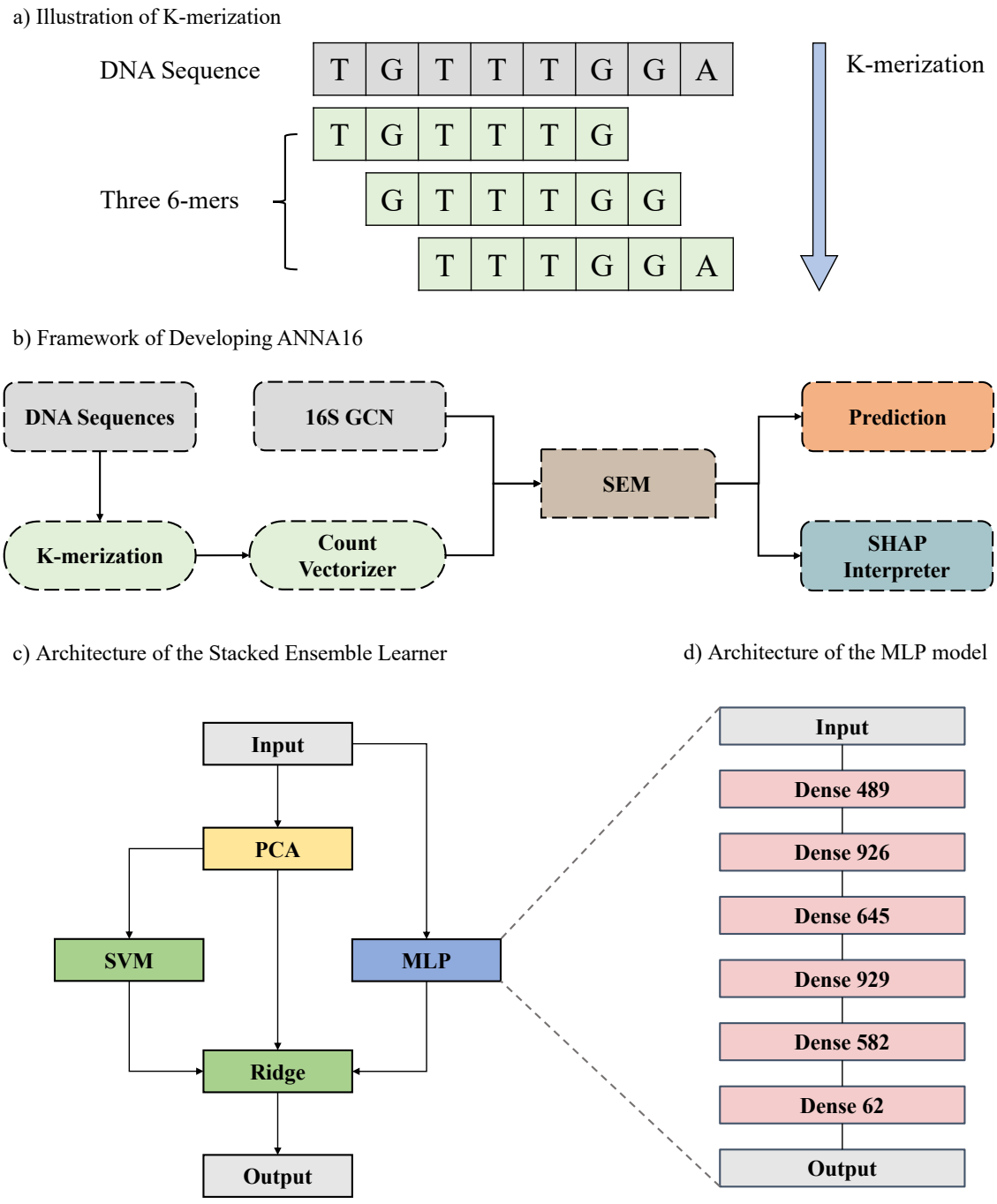

Figure 1. Illustration of the framework and architecture of the deep learning models. a) DNA sequence K-merization (K = 6). b) Framework for designing ANNA16. c) Architecture of the Stacked Ensemble Model (SEM). d) Architecture of the MLP model module inside the SEM.

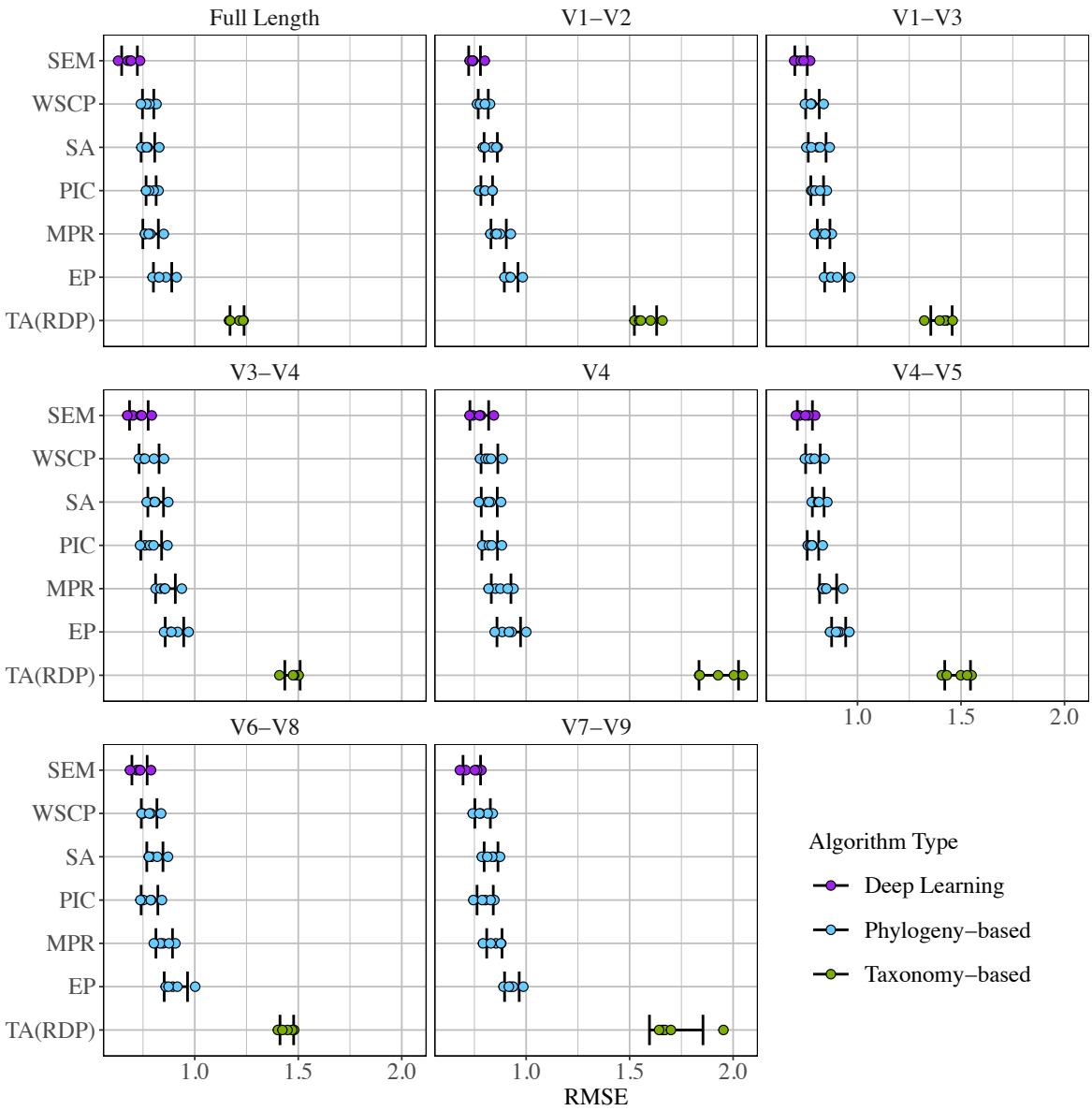

Figure 2. Algorithm performance comparisons at 16S full-length and commonly sequenced subregions. EP: empirical probabilities; MPR: Sankoff's maximum-parsimony; PIC: phylogenetic independent contrast; SA: Subtree Averaging; WSCP: weighted-squared-change parsimony; TA(RDP): taxonomic aggregation based on RDP database; SEM: stacked ensemble model.

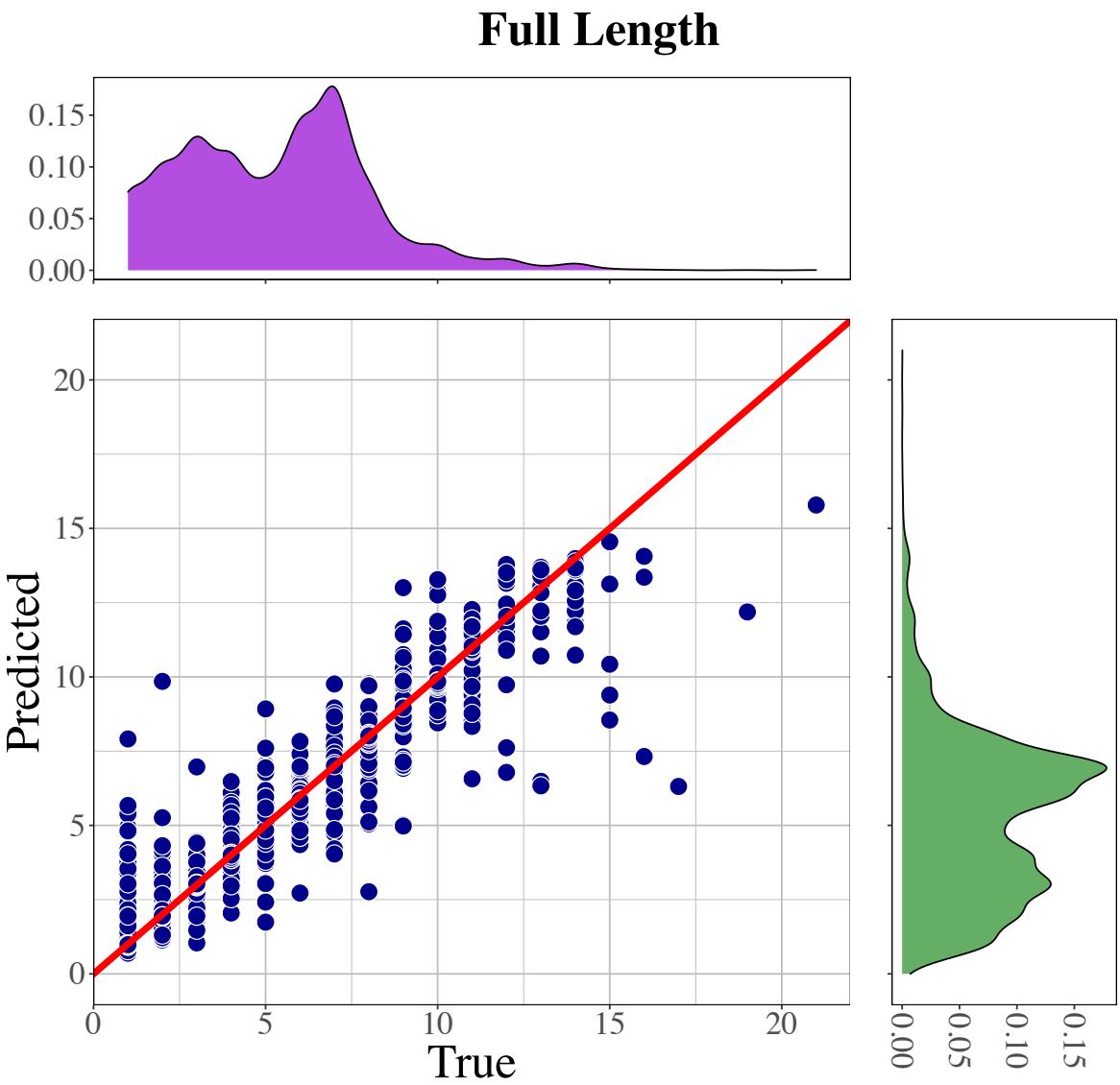

Figure 3. Prediction by SEM on 16S full-length in the first cycle of 5-fold cross-validation. The red line is a diagonal line illustrating to which extent the predicted values are derived from the true values. The density plot filled in violet shows the distribution of the true values, and the green density plot shows the distribution of the predicted values.

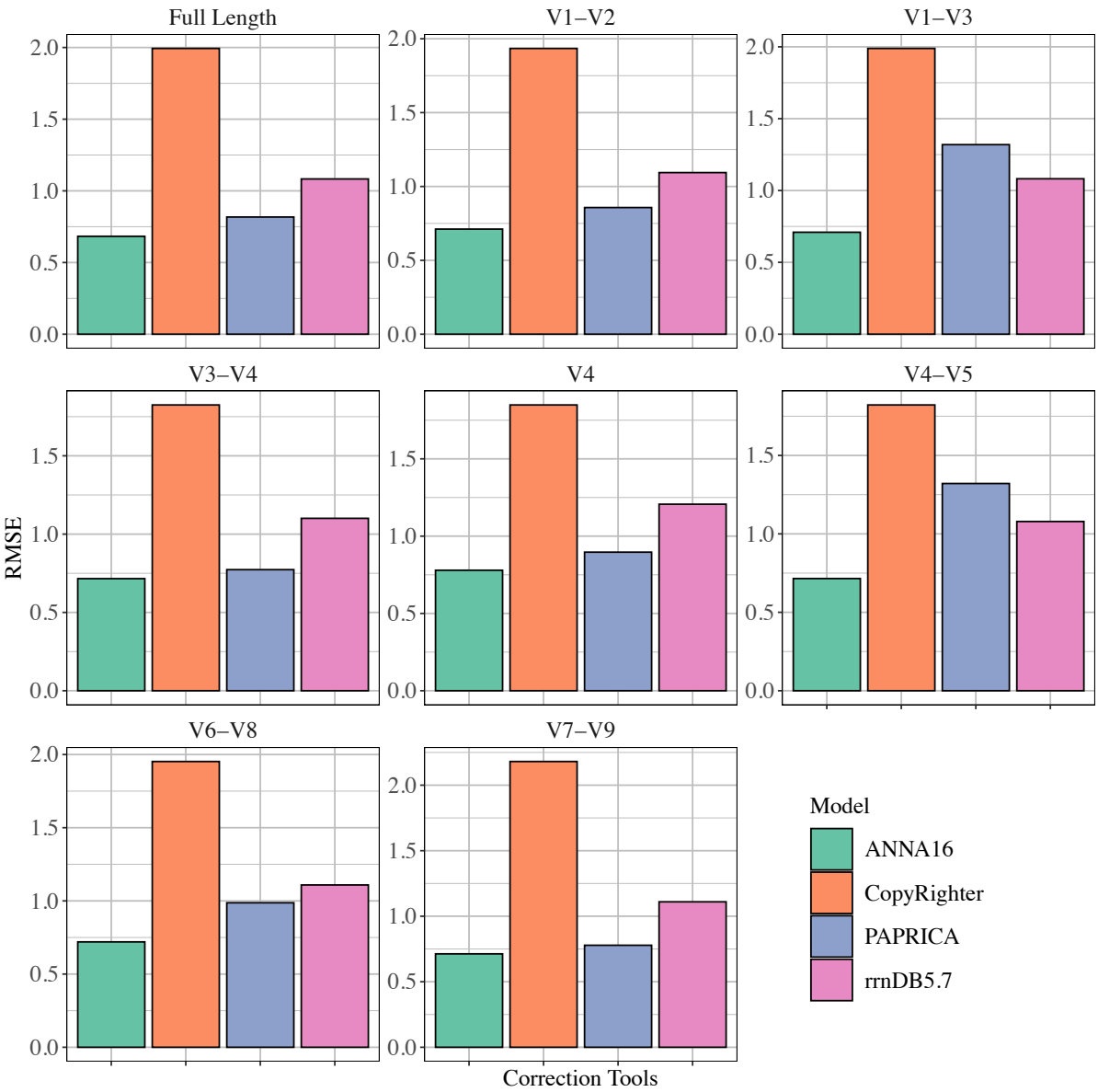

Figure 4. Comparison of performance on the final test dataset among CopyRighter, rrnDB5.7, ANNA16, and PAPRICA.

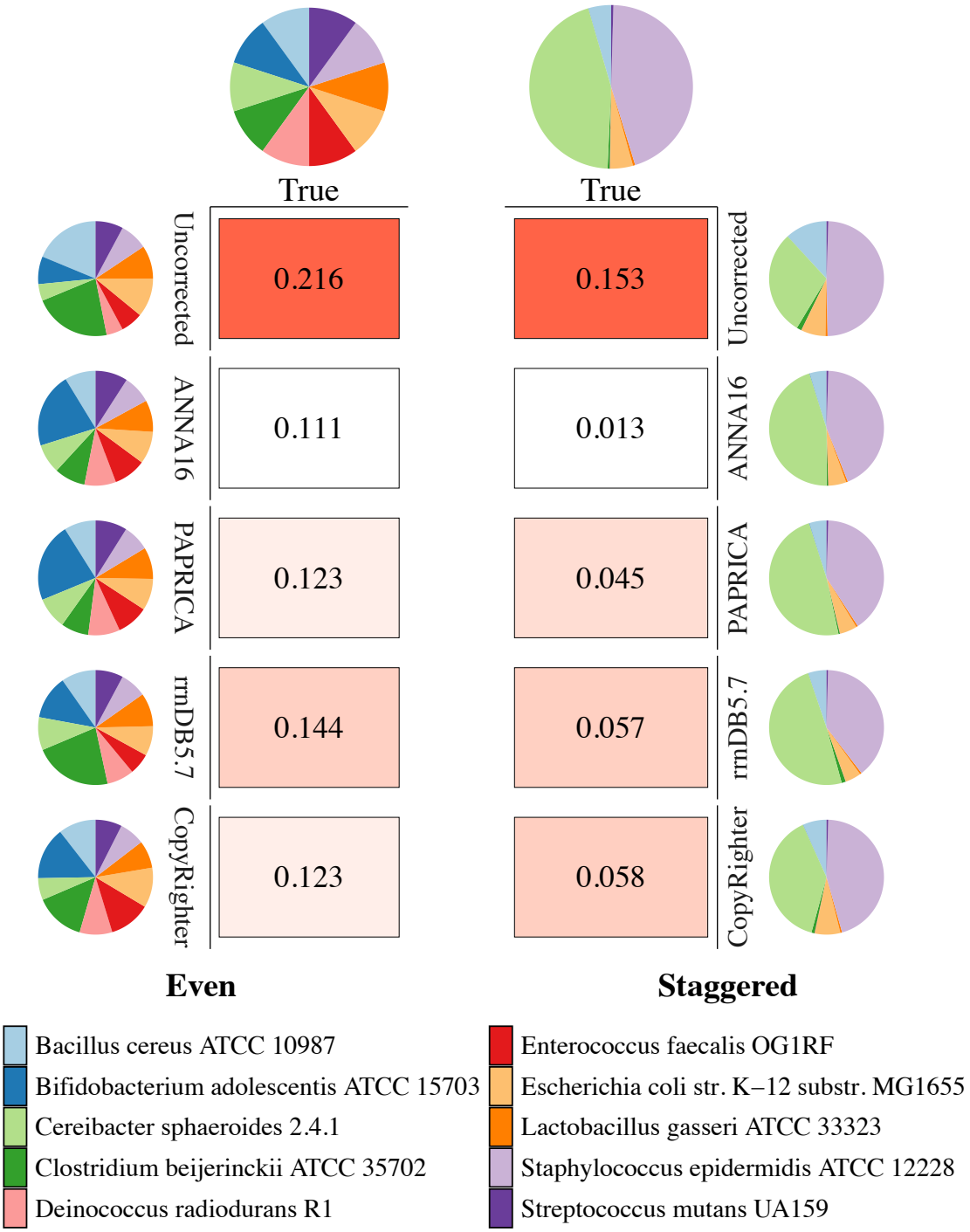

Figure 5. *In silico* test with the mock communities. Left panels: The composition of the even community estimated by uncorrected 16S gene copy fraction, ANNA16, PAPRICA, rrnDB (version 5.7), CopyRighter and their Bray-Curtis Dissimilarity with the cell fraction. The Dissimilarity values are presented as the heat map. Right panel: The composition of the

staggered community estimated by the methods and their Bray-Curtis Dissimilarity with the cell fraction. Uncorrected: Uncorrected copy fraction; True: True cell fraction; ANNA16: Cell fraction estimated by ANNA16; rrnDB5.7: Cell fraction estimated by rrnDB (version 5.7); CopyRighter: Cell fraction estimated by CopyRighter; PAPRICA: Cell fraction estimated by PAPRICA. Even Community: All bacterial strains account for the same proportion. Staggered Community: The ratio of each bacterial strains is uneven.

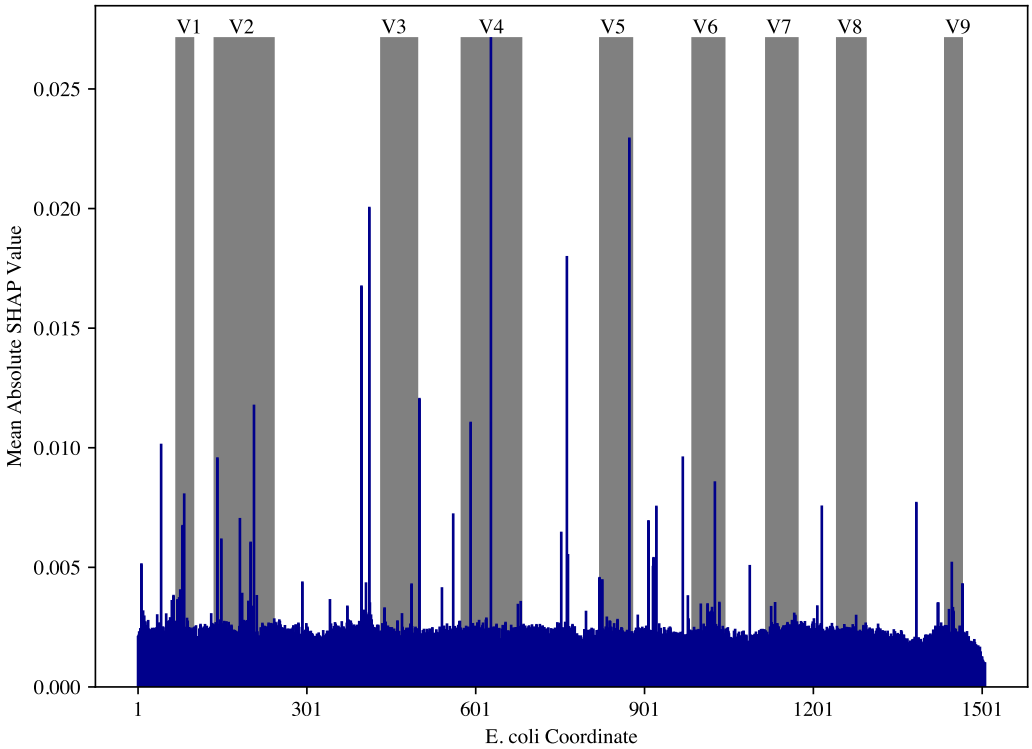

Figure 7. Mean absolute SHAP values across 16S rRNA gene full-length sequences. The mean absolute SHAP values were calculated using the aligned sequences, and then converted to *E. coli* coordinates with *E. coli* (accession number: GCF_002953035.1) as the reference.

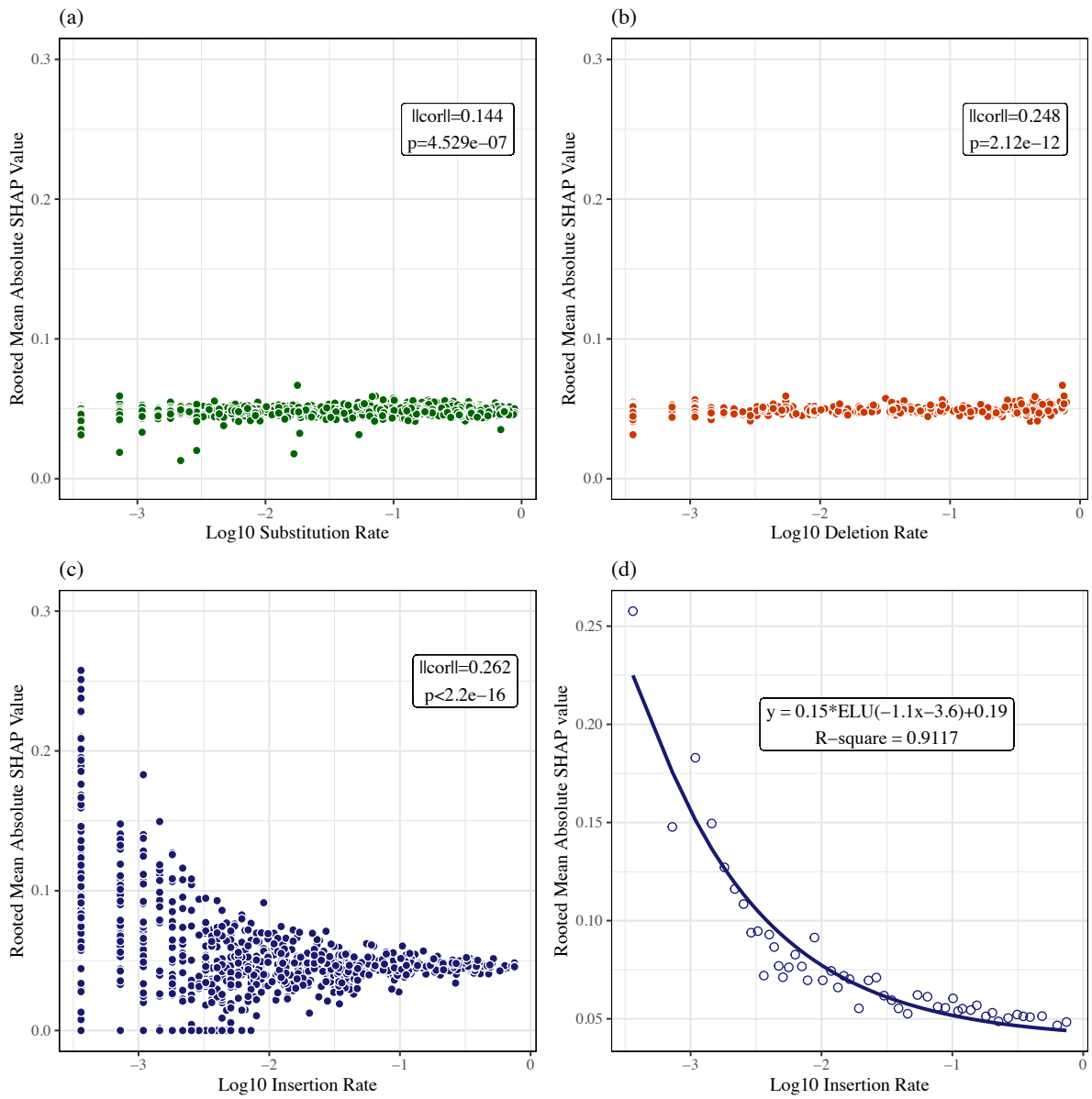

Figure 7. Relationship of SHAP values and the rate of mutations on each position of the aligned 16S rRNA gene sequences. (a) Log10 Substitution Rate vs. Rooted Mean Absolute SHAP Value. (b) Log10 Deletion Rate vs. Rooted Mean Absolute SHAP Value. (c) Log10 Insertion Rate vs. Rooted Mean Absolute SHAP Value. (d) The upper boundary of Log10 Insertion Rate vs. Rooted Mean Absolute SHAP Value, fitted by the elu function.