# Deep Learning for Predicting 16S rRNA Gene Copy Number

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

[1]*Department of Applied and Natural Science, Duke Kunshan University, Suzhou, China*

Correspondence*:

Mustafa Misir

Division of Natural and Applied Sciences, Duke Kunshan University, Duke Avenue No: 8, Suzhou, 215316, China

mustafa.misir@dukekunshan.edu.cn

Yajuan Lin

Division of Natural and Applied Sciences, Duke Kunshan University, Duke Avenue No: 8, Suzhou, 215316, China

linyajuan@gmail.com

## ABSTRACT

### Background

16S rRNA is a commonly used technique for identifying prokaryotes. However, this approach can only reflect the proportion of sequencing reads, rather than the actual cell fraction. To accurately represent actual cell fraction, bioinformatic tools that estimate the 16S gene copy number (GCN) of unmeasured species are required. All currently available tools rely on taxonomy or phylogeny to predict 16S GCN.

### Method

This study develops a deep learning-based method that establishes a novel route. The proposed approach, i.e., Artificial Neural Network Approximator for 16S rRNA Gene Copy Number (ANNA16), essentially links 16S sequence string directly to GCN without the construction of taxonomy or phylogeny.

### Results

The computational analysis revealed the success of the deep learning model built on 16S GCN data retrieved from rrnDB (N = 27,579). The resulting deep learning model outperforms all the major and state-of-the-art algorithms in the literature.

### Conclusion

ANNA16 is a state-of-the-art 16S GCN prediction tool based on deep learning. Its advantages include a simple structure, faster inference speed, less computing consumption, and satisfactory performance. For the accessibility of ANNA16, it is released as a public, end-to-end, web-based tool that predicts copy numbers from 16S rRNA sequences.

43 # 1 INTRODUCTION

44 16S rRNA is the small subunit ribosomal RNA molecule found in prokaryotic ribosomes.
45 As ribosomes are self-replicating, 16S rRNA is readily isolated, and its sequence changes
46 slowly over time [1]. This characteristic allows 16S rRNA to be highly conserved within
47 species [2], while the variance increases as inter-species relatedness decreases [1]. Therefore,
48 16S rRNA is used commonly to identify prokaryotes, or to detect relatedness among distant
49 species by matching next-generation sequencing (NGS) data to reference databases [3], such
50 as RDP and Greengenes. The relatedness can be represented as genetic distance among 16S
51 rRNAs, such as mean nucleotide substitutions per site [3]. This technique, known as 16S rRNA
52 metabarcoding, has been established as the gold standard for inferring organismal phylogeny
53 and microbial ecology for over 20 years [4].

54 One of the critical limitations of 16S rRNA metabarcoding is that NGS determines the
55 proportion of 16S rRNA read counts, rather than cell counts. Since it is typical for prokaryotes
56 to have more than one copy of the 16S rRNA gene, the proportion of 16S rRNA read counts
57 (also known as relative abundance) may not accurately reflect the taxonomic composition of a
58 microbial community [5]. To address this issue and accurately represent the profile of the
59 microbiota, the proportion of 16S rRNA read counts needs to be weighted by the inverse of
60 16S rRNA gene copy numbers (GCN). This step yields a corrected microbial composition that
61 represents the proportion of cell counts of each lineage in the community [5].

62 To calculate 16S GCN, scientists typically rely on experimental methods such as whole
63 genome sequencing [6] or competitive PCR [7]. However, these methods can be expensive or

64 culture-dependent, and so far, GCN data is only available for a limited number of species. To
65 overcome this limitation, bioinformatics tools have been developed to predict the GCN of
66 unmeasured species from measured species. Fortunately, 16S GCN is correlated with
67 relatedness across species [5, 8], meaning that 16S GCN variation is relatively low within
68 species but increases as relatedness decreases. This trait allows 16S GCN to be estimated from
69 phylogeny or taxonomy [8]. Currently, all available tools rely on this kind of method [3, 5, 6].
70 For example, one correction tool could construct one taxonomy or phylogenetic tree of the 16S
71 sequence and estimate the GCN of the unmeasured species from its close measured relatives.
72 For instance, rrnDB [6] is a well-established tool that corrects 16S GCN based on taxonomy.
73 Its algorithm calculates the mean 16S GCN of a taxon from the means of its sub-taxa.

74 Deep learning, as a popular machine learning strategy, comes with its potential of learning
75 complex patterns from biological sequencing data [9, 10]. This aspect can be beneficial as a
76 novel approach for predicting 16S GCN. Deep learning is essentially an Artificial Neural
77 Network (ANN). The concept of ANN is inspired by the way a human brain learns and
78 processes information through synaptic interactions between neurons [11]. An ANN is a
79 computational model that is constituted by multiple interconnected processing elements—
80 artificial neurons [11]. Neurons are utilized in groups as the layers of a network. Each artificial
81 neuron receives and processes input with a non-linear function, known as the activation
82 function [12]. Collectively, one ANN model learns to solve a specific task through capturing
83 non-linear trends and relationships from the presented data. Beyond these capabilities, ANNs
84 are even competent in disclosing relationships or mechanisms that are still unknown to

85 scientists [13]. Deep learning gets its name by having more than one (hidden) layer connecting
86 the input and output units. Having deeper networks refer to representing complex input-output
87 relationships while accommodating a large amount of data. In that sense, high-dimensional
88 biological data is a popular target for deep learning.

89 Recently, a broad spectrum of deep learning approaches has been applied to microbiology
90 research. Various deep learning architectures like Convolutional Neural Network (CNN), and
91 Recurrent Neural Network (RNN), have been used for metagenomic-based taxonomy
92 classification [14], host phenotype prediction [15], disease prediction [16], and microbial
93 community predication [17]. Some other works focus on data mining for antibiotic resistance
94 genes [18], antimicrobial peptides [19], microbial source [20], and raw sequencing data [21].
95 All these applications show the great potential of deep Learning for microbial studies. With
96 these present successes, it should be possible to estimate GCN directly from the 16S sequence
97 through an ANN, without the construction of taxonomy or phylogeny. To the best of our
98 knowledge, there are no deep learning-based methods utilized for the 16S rRNA-based copy
99 number estimation.

100 This study first trains SEM, the developed deep learning model, on the 16S full-length
101 sequence and compares the performance to the taxonomy-based and phylogeny-based
102 algorithms. However, due to drawbacks such as the relatively high error rate or higher costs,
103 microbiologists less commonly sequence the full length of 16S rRNA at present [22]. Full-
104 length 16S could be more widely used when the cost of long-read sequencing (also known as
105 third-generation sequencing) methods is further decreased. Alternatively, microbial studies

106 tend to target the hypervariable regions in 16S rRNA. 16S rRNA contains nine hypervariable
107 regions, interspersed among conserved regions (V1-V9) [23]. Mostly, microbiologist sequence
108 only one to three adjacent hypervariable regions by NGS to study the microbial ecology [22].
109 Therefore, this study also trains deep learning models on different combinations of the adjacent
110 hypervariable regions in 16S rRNA and compares the performance with conventional
111 algorithms. Finally, a deep learning-based bioinformatic tool for 16S GCN correction is
112 developed. The tool is compared with existing correction tools to assess the performance. One
113 state-of-the-art method of model explanation is also adopted to uncover how the deep learning
114 model understands a 16S sequence.

115 Based on the Stacked Ensemble Model, the most accurate method, this study presents
116 ANNA16 (Artificial Neural Network Approximator for 16S rRNA Gene Copy Number), an
117 end-to-end, web-based tool that predicts copy numbers from 16S rRNA sequences.

118

119 # 2 MATERIALS AND METHODS

120 ## 2.1. Data Source and Preprocessing

121 16S rRNA sequences and the GCN data used to train deep learning models in this study
122 are retrieved from rrnDB, version 5.8 [6]. The dataset includes a total of 29,267 16S genes with
123 unique accession numbers. Because raw 16S sequence data contains both (+) strands and (-)
124 strands, using it directly for taxonomy-based and phylogeny-based algorithms may cause bias.
125 Therefore, this study uses the BLASTN algorithm in the BLAST+ application, version 2.11.0
126 [24], to identify the sequence orientation. The first 50 nucleotides of the *E. coli* 16S sequence

127 (+) strand (accession number: GCF_002953035.1), which belong to the conserved region on
128 16S rRNA [25], were used as a template to determine the orientation of each 16S sequence.
129 Sequences with High-Scoring Segment Pairs (HSPs) that had sstart > send and evalue < 0.01
130 are considered (+) strands, while those with sstart < send and evalue < 0.01 are considered (-)
131 strands. Sequences that did not meet these criteria are labeled as unrecognizable and dropped
132 from the dataset. The reverse complement of (-) strand sequences is generated and added to the
133 (+) strands.

134 This study also addresses the issue of varied starting positions among 16S sequences in
135 the dataset by using employs HyperEx [26] to extract full-length sequences between primer
136 27F and primer 1492R (as listed in Table S1) to simulate NGS results of PCR amplicons. This
137 ensures all 16S sequences have uniform starting and ending positions. After this step, the
138 preprocessed dataset contains 27,579 sequences, including 1,550 genera. The average sequence
139 length is 1,499.03 nucleotides, with a standard deviation of 21.49 nucleotides. Gene copy
140 number ranges from 1 copy per genome to 21 copies per genome, with a mean value of 5.39
141 and a standard deviation of 2.78. This preprocessed dataset will be referred to as such in the
142 following content. Moreover, Abellan-Schneyder et al. (2021) [22] reported that V1-V2, V1-
143 V3, V3-V4, V4, V4-V5, V6-V8, and V7-V9 are frequently targeted subregions. Thus, this study
144 also uses primers listed in Table S1 and HyperEx to extract these subregions from the
145 preprocessed dataset.

146 Because a previous study points out that the performance of existing GCN correction tools
147 is unsatisfactory on clades that are distant from training data [3], this study would like to test

148 if deep learning could perform better on these distant clades. Therefore, this study splits the
149 preprocessed dataset into the train and test datasets based on when the sequences are published.
150 Specifically, sequences that already exist in the previous rrnDB release (version 5.7) are
151 selected as the train dataset for cross-validation, while sequences newly added in version 5.8
152 are selected as the test dataset. This splitting method produces a near-random test dataset
153 containing newly published sequences. The train dataset contains 19,520 sequences, involving
154 1,331 genera, with an average sequence length of 1,498.72 nucleotides and a standard deviation
155 of 21.41 nucleotides. Gene copy number ranges from 1 copy per genome to 21 copies per
156 genome, with a mean value of 5.26 and a standard deviation of 2.74. The test dataset contains
157 8,059 sequences, involving 676 genera, with an average sequence length is 1,499.79
158 nucleotides and a standard deviation of 21.67 nucleotides. Gene copy number ranges from 1
159 copy per genome to 20 copies per genome, with a mean value of 5.72 and a standard deviation
160 of 2.85. The test dataset includes 219 genera that are not present in the train dataset.
161

162 **2.2. Conventional Algorithms and Existing Tools**

163 Sub-taxa averaging (TA) is one taxonomy-based algorithm. The algorithm calculates the
164 mean 16S GCN of a taxon from the means of its sub-taxa. To implement this algorithm, 16S
165 sequences are first input to *assignTaxonomy* function in the DADA2 package in R (version
166 1.21.0) [27], which yields two taxonomies based on RDP (version 18, formatted by DADA2)
167 [28] and Greengenes (version 13.8) [29] database, respectively. The algorithm first receives the
168 taxonomic and GCN data from the database and converts the data structure of the taxonomy to

169 an inheritance tree. In the tree, every taxon is viewed as a node on the tree, and the algorithm
170 computes the copy number of every node by taking the mean of its sub-nodes. If an unknown
171 query is given, it assigns the value of the query's parent node to the query node as an estimation.
172 This study also employs five phylogeny-based algorithms to compare with the deep
173 learning method: empirical probabilities (EP), subtree averaging (SA), phylogenetic
174 independent contrasts (PIC) [30], Sankoff's maximum-parsimony (MPR) [31], and weighted-
175 squared-change parsimony (WSCP) [32]. These algorithms first receive a pre-constructed tree
176 of 16S sequences and estimate the GCN of each ancestral node from its daughter species
177 through weighing each labeled daughter node with its phylogenetic distance to the parental
178 node [33]. To implement the algorithms, 16S sequences in the dataset are first aligned by
179 MAFFT v7.490 [34] and then converted to a phylogenetic tree by FastTree 2.1.10 [35]. The
180 algorithms are performed using the castor package in R [33], adjusted from the R script in
181 Louca, Doebeli, and Parfrey (2018) [3] (MPR is implemented by function *hsp_max_parsimony*,
182 WSCP by function *hsp_squared_change_parsimony*, SA by function *hsp_subtree_averaging*,
183 EP by function *hsp_empirical_probabilities*, and PIC by *hsp_independent_contrasts*). All
184 functions are set to default parameters.
185 After developing the deep learning-based tool, this study compares it with all available
186 tools on the test dataset, which are rrnDB [6], CopyRighter [5], PAPRICA [36], PICRUSt2
187 [37]. rrnDB is a taxonomy-based tool using sub-taxa averaging, while the latter three tools are
188 phylogeny-based. CopyRighter uses the PIC algorithm to pre-compute the copy number of all
189 taxa in the Greengenes database [29] and makes predictions by searching the lineages of the

190 queries in its pre-computed GCN table [5]. The original mission of PAPRICA and PICRUSt2
191 is to predict the metabolic profile of a microbial community based on the 16S data, but they
192 have the function of predicting the copy number of any given gene according to the phylogeny
193 [36, 37]. These two tools first utilize EPA-ng [38] to place the queries on their built-in
194 phylogenetic tree and use subtree-averaging (PAPRICA) or maximum parsimony (PICRUSt2)
195 algorithms to predict the copy number of the queries [36, 37]. Because rrnDB and CopyRighter
196 application output microbial community abundance instead of 16S GCN values directly, this
197 study retrieves their precomputed tables (rrnDB version 5.7) and predict GCN values by
198 searching the precomputed tables for assigned lineages. The taxonomy of full-length sequences
199 and all subregion sequences are generated by the *assignTaxonomy* function in the DADA2
200 package through RDP (version 18) and Greengenes (version 13.8, formatted by DADA2) [29]
201 database. The assigned RDP lineages are searched in the rrnDB precomputed table, and
202 Greengenes lineages are searched in CopyRighter precomputed table. As for PAPRICA and
203 PICRUSt2, the DNA sequences in the test dataset are directly input to the tools. For PAPRICA,
204 *paprica-run.sh* is called without additional parameters. For PTCRUSt2, *place_seqs.py* and
205 *hsp.py* are called without additional parameters.
206

207 **2.3. Development of ANNA16**

208 2.3.1. Framework

209 To build our deep predictive models, this study first implements a three-layer MLP
210 prototype model. The sequences in the train dataset are split for training and testing by 3:1. The

211 DNA sequences are preprocessed with K-merization. An example of a set with three 6-mers is
212 displayed in Figure 1a). The K-mers are converted into the number of counts in the set via the
213 *CountVectorizer* function in the scikit-learn module (version 0.23.2) [39]. The deep learning
214 model will intake the vector of occurrence counts and predicts the gene copy number.

215 After testing K from 3 to 7 on the prototype model, the model performance is found to
216 improve as K increased. However, a larger K also required significantly more memory.
217 Therefore, the optimal K value was determined to be 6, which provides good performance
218 without requiring excessive memory. 6-merization is to decompose the DNA sequence into a
219 set of sub-sequences of length 6. With the K value determined, a Stacked Ensemble Model
220 (SEM) is constructed for 16S GCN prediction. The framework is demonstrated in Figure 1b).
221 The performance of the model is evaluated by compareing it with conventional algorithms
222 using a 5-fold cross-validation on full-length and subregional DNA sequences. Shapley
223 Additive exPlanations (SHAP) [40] is introduced here for post-hoc interpretability. The entire
224 study is conducted with Python 3.8 and Tensorflow 2.8.0 on Nividia RTX 3090.

225

226 2.3.2. Model Construction

227 The deep learning model employs a strategy of ensemble learning, known as model
228 stacking. This involves training multiple base-models on the available data and then using a
229 meta-model to accept the output of all base-models and make the final prediction. The stacked
230 model would typically perform better than any single base-model [41, 42]. In this study, the
231 Stacked Ensemble Model combines one MLP model with other classic machine learning

232 models. MLP is a type of artificial neural network that consists of layers of artificial neurons,
233 and the input is processed layer by layer [11]. The calculation of the output for each neuron is
234 as follows [11]:

235 $$v_k = \sum_{i=0}^{n} \omega_{k,i} x_i$$

236 $$\gamma_k = \varphi(v_k + v_{k_0})$$

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

241 $$\text{RMSE} = \sqrt{\frac{\sum_{i=0}^{n}(Y_{true} - Y_{pred})^2}{n}}$$

242 where $n$ is the size of the input dataset, $Y_{true}$ is the true value of copy number, $Y_{pred}$ is the
243 value of the predicted copy number.

244 To build the Stacked Ensemble Model, several classic machine learning algorithms are
245 tested as base-models. These algorithms include Support Vector Machine (SVM) [43], Random
246 Forest [44], K-nearest Neighbors [45], and Gradient Boosting [46]. Linear Regression, Ridge
247 Regression [47], and Lasso Regression [48] are tested as meta-models. All machine learning
248 algorithms are implemented by the scikit-learn module (version 0.23.2). After testing, MLP
249 and SVM are selected as the base-models and Ridge Regression as the meta-model, as shown
250 in Figure 2c). For dimensionality reduction, Principal Component Analysis (PCA) is
251 implemented using the *PCA* function in scikit-learn, with *n_components* set to 100. SVM is

252 implemented by the *SVR* function in scikit-learn, and Ridge Regression by the *Ridge* function
253 in scikit-learn.

254 The parameter of SEM is optimized through the *SMAC4BB* function in the SMAC3
255 package (version 1.4.0) [49] for 500 iterations. For the MLP component, hyperparameters
256 including the number of dense layers, hidden layer size, activation functions, and learning rate
257 are tuned. The number of dense layers of the model is tested in the range of [1, 7], hidden layer
258 size from the range of [1, 1024], activation functions are selected among Rectified Linear Unit
259 (ReLU), Exponential Linear Unit (ELU), Gaussian Error Linear Unit (GELU), Scaled
260 Exponential Linear Unit (SELU), and Linear, and the learning rate is tested in the range of
261 [0.00005, 0.1]. The final MLP architecture (Figure 2d) contains a dense block of 6 hidden
262 layers, with neuron numbers of 489, 926, 645, 929, 582, and 82 respectively, and the activation
263 functions are GELU, ReLU, ReLU, ELU, GELU, and Linear respectively. For SVM, the *kernel*
264 value is chosen from "rbf", "sigmoid", and "linear", *gamma* is tested between "scale" and
265 "auto", and *C* is tested as a float in the range of [0.3, 0.9] or as an integer in the range of [1,
266 100]. For Ridge, the value of *alpha* tested as a float in the range of [0.3, 0.9] or as an integer in
267 the range of [1, 100]. After tuning, the optimized parameters are *kernel = 'rbf'*, *gamma =*
268 *'auto'*, *C = 11*, and *alpha = 49* (a complete list of parameters is shown in Table S2).
269

270 2.3.3. Dismissed Machine Learning Algorithms

271 This study also used tested a Transformer model and a Residual Multi-layer Perceptron for
272 16S GCN prediction. The Transformer model is based on the original transformer encoder

273 architecture with cosine learning rate decay [50]. It is followed by global average pooling, and
274 two dense layers. Embedding and position encoding are used as input for the Transformer
275 model. Residual Multi-layer Perceptron uses Residual Connection architectures, which involve
276 skipping connections to jump over two to three layers [51]. However, these models do not
277 perform well in the study and are therefore excluded.
278

279 **2.4. Case Study**

280 Based on the SEM method, this study presents ANNA16 (Artificial Neural Network
281 Approximator for 16S rRNA Gene Copy Number), an end-to-end tool that predicts copy
282 numbers from 16S rRNA sequences. To demonstrate ANNA16's performance, a case study is
283 designed that simulates two mock communities *in silico*. rrnDB, CopyRighter, PAPRICA, and
284 PTCRUSt2 are also applied to estimate the mock communities' microbial profile as a positive
285 control. The simulated communities are two commercially available mock communities, one
286 even community (ATCC MSA-2003) and one staggered community (ATCC MSA-1001). The
287 information of the mock communities is shown in Table S3. Such mock communities have
288 been used by Moriya (2021) to study methods of quantifying a microbial community [52]. Both
289 mock communities consist of ten common bacterial strains, but with different ratios. This study
290 retrieves genome sequences of the ten strains from NCBI, copy number data from rrnDB
291 (version 5.7), and assigns a cell count value to each strain by assuming each community has
292 1,000,000 cells. Table S3 shows the basic information about the mock communities. To extract

293 16S full-length sequences from the genomes, HyperEx is used in the same way as described in
294 the Data Source and Preprocessing.

295 To evaluate the accuracy of community profile prediction, Bray-Curtis Dissimilarity [53]
296 is calculated between the true cell fraction and the estimated cell fraction by each tool. The
297 Bray-Curtis Dissimilarity is a commonly used method in ecology and biology to measure the
298 compositional dissimilarity between two different communities. A higher Bray-Curtis
299 Dissimilarity value indicates the greater dissimilarity between the two communities [53]. Its
300 calculation is as follows:

301 $$BC = 1 - \frac{2C_{AB}}{S_A + S_B}$$

302 where $BC$ is the Bray-Curtis Dissimilarity value, $A$ and $B$ are two communities, $C_{AB}$ is the sum
303 of the lesser values for species present in both communities, $S_A$ and $S_B$ are the total number of
304 species counted at both communities. The *vegdist* function in the vegan (version 2.5-7) package
305 of R are used to calculate the Bray-Curtis Dissimilarity values, with the *method* parameter set
306 to "bray".

307

308 **2.5. Deep Learning Model Explanation**

309 To further explain ANNA16's prediction, this study uses SHAP [40], a unified
310 measurement of feature importance by quantifying the individual contribution of each feature
311 to a model prediction. This method is based on both collational game theory [54] and the
312 classical Shapley method [55] and views each feature as a player in Shapley. The marginal
313 contribution can be estimated under multiple feature subsets, and the final contribution is the

314 sum of all marginal contributions. A positive SHAP value denotes that the feature contributes
315 positively to the model's prediction over the model expectation, whereas a negative value
316 denotes that the feature contributes negatively. In this study, each K-mer is considered a feature.
317 To calculate the SHAP value for each K-mer, the preprocessed dataset and pre-trained
318 ANNA16 are input to the *KernelExplainer* function in shap module (version 0.41.0) in Python.
319 The function views each prediction of every single sequence in the dataset as an independent
320 game and calculates SHAP values for each game.

321 This study randomly samples 10% of the whole dataset for SHAP value analysis (N =
322 2758). As a result, a list of 2,758 SHAP values is assigned to every K-mer to represent the K-
323 mer's overall contribution. These SHAP values are mapped back to the 16S sequence to
324 investigate if there is a positional pattern in contribution. Each sequence is first transformed
325 into an array of K-mers, and then each K-mer is replaced with its corresponding SHAP value.
326 As a result, the contribution of each position in a sequence is represented as an array of SHAP
327 values. What is more, to study the relationship between contribution and nucleotide
328 conservation in a position, the information content of each position is calculated. The
329 information content is a measurement of the average information available in a position,
330 representing how conserved a position is if applied to analyzed nucleotides [56]. This
331 measurement has been widely used in genetics, and one well-known application is to serve as
332 the height of characters in sequence logos [57]. The information content is calculated using
333 previously aligned 16S sequences during the implementation of phylogeny-based algorithms.
334 The calculation is as follows:

335 $$H_i = -\sum f_{char,i} \times \log_2(f_{char,i}), char \in [A, T, C, G, -, N]$$

336 $$R'_i = \log_2(6) - H_i - (\frac{1}{\ln 2} \times \frac{s-1}{2n})$$

337 $$R_i = R'_i \times \sum f_{nt,i}, nt \in [A, T, C, G]$$

338 where $H_i$ is the uncertainty of position $i$, f is the frequency of a character, − represents gaps
339 generated in multiple sequence alignment, $N$ = ambiguous nucleotides, $R_i'$ is information
340 content of all characters in position $i$, $s$ is the number of characters (which is 6 in this
341 calculation), $n$ is the number of sequences in alignment (which is 2758), $R_i$ is the information
342 content of the four nucleotides, and *nt* represents nucleotides. $R_i$ is used to represent the degree
343 of conservation of the position $i$.

344

345 **3 RESULTS**

346 This study evaluates the performance of SEM in comparison with other algorithms for 16S
347 GCN correction. Figure 2 and Table 1 summarize the results of 5-fold cross-validation. The
348 results first show sub-taxa averaging based on RDP and Greengenes exhibit a mean RMSE of
349 1.18 and 2.52 copies/genome, respectively, with a standard deviation of RMSE being 0.0356,
350 and 0.0388. Due to the much higher mean RMSE of TA(Greengenes), the results are excluded
351 from Figure 2. RDP is found to be the more accurate database for this task which produces the
352 lowest error range. All phylogeny-based algorithms perform better than sub-taxa averaging (EP:
353 Mean = 0.845, SD = 0.0443; MPR: Mean = 0.787, SD = 0.0375; PIC: Mean = 0.789, SD =
354 0.0243; SA: Mean = 0.774, SD = 0.0329; WSCP: Mean = 0.775, SD = 0.0273). The error range
355 of all phylogeny-based algorithms is lower than those of taxonomy-based algorithms, with no

356 overlap. With low RMSE and small SD, WSCP turns out to be the optimal phylogeny-based
357 algorithm. SEM, the deep learning model, outperforms all other algorithms with the lowest
358 mean RMSE (0.685 copies/genome) with a standard deviation of 0.0379. Its error range on the
359 16S full-length is the lowest among all algorithms and does not overlap with the phylogeny-
360 based algorithms.

361 Next, Figure 2 and Table 1 also present the performance of the algorithms for the 16S
362 subregions. All algorithms produce an increased Mean RMSE on subregions compared with
363 the performance for 16S full-length. EP experiences the most significant increase, with the
364 mean RMSE rising to 0.9312 copies/genome on the V7-V9 region. Among phylogeny-based
365 algorithms, WSCP is the most stable one, with mean RMSE on subregions ranging from 0.7792
366 (V3-V4) to 0.8227 (V4) copies/genome. SEM also displays relatively consistent performance
367 on subregions, ranging from 0.731 (V1-V3) to 0.774 (V4) copies/genome. Notably, the error
368 range of SEM on subregions is lower than EP and MPR in all subregions, and partially lower
369 than SA, PIC, and WSCP, as demonstrated in Figure 2. These findings indicate SEM
370 completely outperforms EP and MPR and partially outperforms SA, PIC, and WSCP.

371 The first cycle of cross-validation results produced by SEM are presented in Figure 3. The
372 figure shows a strong linear relationship between the predicted values and the true values, and
373 the data point also have similar distribution patterns. However, it is worth noting that SEM
374 tends to underestimate the GCN when the true value is greater than 14. This issue will be
375 discussed in more detail in the **Discussion** section.

376 Due to its superior performance, a bioinformatics tool, ANNA16, is developed based on
377 SEM and compared with other available tools in the final test. Figure 4 and Table 2 demonstrate
378 the results of the final test. Because PICRRUSt2 has a much higher RMSE (larger than 3.4
379 copies/genome for all regions) as shown in Table 2, it is excluded from Figure 4 and the
380 following case study. ANNA16 achieves the lowest RMSE among the tools at full-length and
381 every subregion. It produces the smallest RMSE appears at the full-length, which is 0.683
382 copies/genome, and the highest RMSE at the V4 region, which is 0.780 copies/genome. In
383 contrast, the RMSE of CopyRighter ranges from 1.822 (V4-V5) to 2.181 (V7-V9)
384 copies/genome, rrnDB from 1.079 (V4-V5) to 1.207 (V4) copies/genome, PAPRICA from
385 0.773 (V3-V4) to 1.321 (V4-V5), and PICRUSt2 from 3.459 (V4-V5) to 4.731 (V6-V8).
386 ANNA16 outperformed all other tools, demonstrating superior performance at all subregions
387 and full-length.

388 Figure 5, Table S4, and Table S5 show uncorrected copy fraction, true cell fraction, and
389 cell fraction (microbial profiles) estimated by ANNA16, rrnDB (version 5.7), CopyRighter,
390 and PAPRICA. The difference in uncorrected copy fraction and true cell fraction demonstrates
391 that GCN correction is necessary for obtaining an accurate microbial profile. The Bray-Curtis
392 Dissimilarity analysis shows that the dissimilarity values of the four tools for the even
393 community are considerably higher than those for the staggered community. This coincides
394 with the compositional patterns, where the estimated staggered patterns by the four tools are
395 more similar to the patterns of true cell fractions. However, the cell fraction estimated by
396 ANNA16 has the smallest dissimilarity value for both communities. For the even community,

397 PAPRICA and CopyRighter exhibit very close dissimilarity values and are the second most
398 accurate tools. For the staggered community, PAPRICA is the second most accurate tool,
399 followed by rrnDB, but CopyRighter is the least accurate one. Overall, ANNA16 outperforms
400 the other tools in terms of similarity patterns with the true cell fraction.

401 To illustrate how ANNA16 makes its predictions, the shap module in Python is used to
402 calculate the SHAP value of each K-mer. Figure S1 displays the top 20 K-mers ranked by mean
403 absolute SHAP value, while Figure 6 shows the distribution of SHAP values across the
404 sequences. To determine the range of hypervariable regions, the sequence of conserved regions
405 of *E. coli* is first extracted according to region coordinates provided by Yarza et al. [25]. All
406 sampled sequences undergo BLASTN (version 2.11.0) against *E. coli* conserved sequences to
407 determine the range of conserved regions in each sequence. The range of hypervariable regions
408 is thus deduced from conserved regions. To reduce complexity, only 500 randomly chosen
409 sequences are labeled in Figure 6. The results indicate that ANNA16 does not simply extract
410 information from hypervariable regions; conserved regions also provide information. To
411 further illustrate the relationship between SHAP value and the information content (degree of
412 conservation), the mean absolute SHAP value on each position against the degree of
413 conservation of each position is plotted in Figure 7. Interestingly, the mean absolute SHAP
414 value reaches its peak when the degree of conservation is extremely low. This phenomenon
415 will be further discussed in the **Discussion** section.

416

417 **4 DISCUSSION**

418 This study presents a novel approach for predicting 16S GCN based on 16S sequence using
419 the Stacked Ensemble Model (SEM), a deep learning model. The method outperforms
420 taxonomy- and phylogeny-based algorithms in predicting GCN at both the 16S full-length and
421 subregions. ANNA16, a bioinformatic tool based on SEM, is subsequently developed and
422 compared to existing 16S GCN correction tools, rrnDB, CopyRighter, PAPRICA, and
423 PICRUSt2, through the final test and a case study. The SHAP method is used to explain how
424 ANNA16 achieves good performance in predicting GCN.

425 First and foremost, ANNA16 emerges as the most accurate and state-of-the-art GCN
426 correction tool for the 16S full-length and all subregions tested in this study. As demonstrated
427 in Figure 2, the algorithm used by ANNA16—Stacked Ensemble Model of MLP, SVM, and
428 Ridge (SEM)—outperforms all taxonomy- and phylogeny-based algorithms at 16S full-length,
429 exhibiting a consistently lower error range. As for subregions, SEM still maintains the lowest
430 error ranges despite some overlap with the counterparts of PIC, SA, WSCP. These results
431 underscore the superior accuracy of SEM in GCN correction compared to existing algorithms
432 across all regions tested, with 16S full-length being its main strength. In addition, as Figure 4
433 and Table 2 show, ANNA16 achieves the lowest RMSE among the tools at the full-length and
434 every subregion in the final test, indicating its superior ability to predict GCN in newly
435 discovered species. The case study results in Figure 5, Table S4, and Table S5 also support this
436 finding, showing that the true cell fraction of both communities more closely resembles those
437 presented by ANNA16, especially for the staggered community. Overall, the results confirm
438 that ANNA16 is currently the most accurate 16S GCN correction tool.

439 In addition to its superior performance, ANNA16 also offers an advantage in inference
440 speed without the need for pre-computation. Existing GCN correction tools can be classified
441 into two categories based on their inference strategies: pre-computation or real-time
442 phylogenetic placement. On the one hand, pre-computational tools (i.e., rrnDB & CopyRighter)
443 compute the GCN value for every taxon in the database beforehand, and the tools retrieve the
444 GCN values from the database when a query is given. On the other hand, real-time phylogenetic
445 placement tools (i.e., PAPRICA and PICRUSt2) place a query sequence into their built-in
446 phylogenetic trees to predict the GCN value at runtime. Compared with real-time phylogenetic
447 placement, pre-computation tools have short inference time, but are also bound to their
448 database, which can lead to errors if used with databases having differences in nomenclature.
449 For example, the SILVA database [58] merges *Allorhizobium*, *Neorhizobium*, *Pararhizobium*,
450 and *Rhizobium* as one genus, named *Allorhizobium-Neorhizobium-Pararhizobium-Rhizobium*,
451 but in RDP these are independent genera. ANNA16, however, is free from this limitation as it
452 predicts GCN values directly for each 16S sequence and allows users to assign taxonomy
453 separately, leading to faster prediction time and database-independence.

454 After demonstrating the supremacy of ANNA16, an interesting question arises: how does
455 the model understand the complex relationship between the 16S sequence and 16S GCN?
456 Traditionally, microbiologists have considered the nine hypervariable regions to be the most
457 informative for inferring the phylogeny of one species. One hypothesis, therefore, is that K-
458 mers from hypervariable regions would contribute more to prediction than those from
459 conserved regions. However, Figure 6 shows that the distribution of SHAP values does not

460 entirely align with the location of hypervariable regions. Though high SHAP values tend to
461 cluster in the V2, V3, V5, V6, V7, and V9 regions, conserved regions between V2-V3, V4-V5,
462 and V8-V9 also contribute considerably.

463 To further explore how ANNA16 perceives the relationship the 16S sequence and 16S
464 GCN, Figure 7 is created. Though a linear relationship is expected between information content
465 and mean absolute SHAP values at each position, the scatter plot actually shows an L-shaped
466 trend. High SHAP values aggregate at positions with extremely low information content, while
467 SHAP values are generally constant in other positions. This trend indicates extremely variable
468 positions or positions with insertion of extremely low frequency are most informative for
469 ANNA16, and ANNA16 can identify these positions and assign high weights to them. This
470 may help explain why ANNA16 or MLP has better performance than conventional algorithms.
471 Conventional algorithms typically weigh every position equally by converting the variance in
472 the sequence-to-sequence similarity, which can underrepresent informative positions. However,
473 by weighing K-mers unequally according to how much information they contain, ANNA16
474 could more effectively utilize the information contained in the 16S sequence. This finding
475 underscores the potential of deep learning in genetics. Not only can deep learning accurately
476 predict traits based on sequencing data, but by investigating the contribution of each feature,
477 scientists can also discover which genes or loci are most closely related to the trait.

478 Although ANNA16 has shown remarkable performance in 16S GCN prediction, several
479 aspects could still be improved. First, one limitation of ANNA16 is its tendency to
480 underestimate the GCN of species with high GCN. As shown in Figure 3, ANNA16's

481 predictions become increasingly underestimated when the true GCN is greater than or equal to
482 14. This limitation could be attributed to the imbalanced distribution of GCN in the dataset.
483 Figure S2 shows the distribution of GCN in the preprocessed dataset. Sequences with GCN ≥
484 12 account for only a small proportion of the whole dataset, and sequences with GCN > 15 are
485 extremely rare. One solution to this problem is to wait for more sequence data with high GCN
486 to be collected, as new sequencing techniques are rapidly generating more genomic data.
487 Alternatively, data imbalance methods, such as class re-balancing, information augmentation,
488 and module improvement [59] could be applied to ANNA16 to address the issue of
489 underestimation. Furthermore, due to the limited amount of 16S GCN data currently available,
490 transfer learning could be explored as a potential solution. Future studies could collect a large
491 amount of unlabeled 16S rRNA sequences and pre-train a model with an autoregression or
492 autoencoder task, similar to models like BERT [60] or GPT-2/3/4 [61], to further improve the
493 accuracy of 16S GCN prediction.
494

495 **5 CONCLUSIONS**

496 In conclusion, this study developed ANNA16 as a deep learning-based new bioinformatic
497 tool for predicting the gene copy number of 16S rRNA. Compared to existing taxonomy-based
498 and phylogeny-based algorithms, ANNA16 showed significantly lower error ranges at 16S
499 full-length and all 16S subregions in the cross-validation. Furthermore, in the final test and
500 case study, ANNA16 outperformed existing tools such as rrnDB, CopyRighter, PICRUSt2, and
501 PAPRICA. ANNA16's simple structure also offers several advantages, including faster

502 inference speed, independence of pre-computation, and less demand for computing resources.
503 The SHAP analysis implies ANNA16 could utilize the information contained in the 16S
504 sequence more effectively by weighing K-mers unequally according to how much information
505 they contain. Future studies may combine strategies such as pre-training the model on
506 unlabeled data and addressing data imbalance to further decrease the error of copy number
507 correction.
508

509 **LIST OF ABBREVIATIONS**

510 GCN: Gene Gopy Number
511 MLP: Multi-layer Perceptron
512 SEM: Stacked Ensemble Model
513 SHAP: Shapley Additive exPlanation
514

515 **DECLARATIONS**

516 ***Ethics approval and consent to participate***

517 Not applicable.

518 ***Consent for publication***

519 Not applicable.

520 ***Availability of data and materials***

521 16S rRNA sequences and the GCN data analyzed during the current study are available in the
522 rrnDB repository, https://rrndb.umms.med.umich.edu/. The codes and model files of ANNA16

are available at https://github.com/Merlinaphist/ANNA16. ANNA16 is deployed on Google Colaboratory, https://colab.research.google.com/drive/1XwpTMCHSfTmzpHyKrmiD8aC8C_1nndUV#scrollTo=M0jLLsHC9W4M.

***Competing interests***

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

***Funding***

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

***Authors' contributions***

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

***Acknowledgments***

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

544 **REFERENCES**

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

547 2. STACKEBRANDT E, GOEBEL BM: **Taxonomic Note: A Place for DNA-DNA**
548 **Reassociation and 16S rRNA Sequence Analysis in the Present Species Definition**
549 **in Bacteriology**. *International Journal of Systematic and Evolutionary Microbiology*
550 1994, **44**(4):846-849.

551 3. Louca S, Doebeli M, Parfrey LW: **Correcting for 16S rRNA gene copy numbers in**
552 **microbiome surveys remains an unsolved problem**. *Microbiome* 2018, **6**(1):41.

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

556 5. Angly FE, Dennis PG, Skarshewski A, Vanwonterghem I, Hugenholtz P, Tyson GW:
557 **CopyRighter: a rapid tool for improving the accuracy of microbial community**
558 **profiles through lineage-specific gene copy number correction**. *Microbiome* 2014,
559 **2**(1):11.

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

563 7. Fogel GB, Collins CR, Li J, Brunk CF: **Mini-Review Article: Prokaryotic Genome**
564 **Size and SSU rDNA Copy Number: Estimation of Microbial Relative Abundance**
565 **from a Mixed Population**. *Microbial Ecology* 1999, **38**(2):93-113.
566 8. Kembel SW, Wu M, Eisen JA, Green JL: **Incorporating 16S Gene Copy Number**
567 **Information Improves Estimates of Microbial Diversity and Abundance**. *PLOS*
568 *Computational Biology* 2012, **8**(10):e1002743.
569 9. Jurtz VI, Johansen AR, Nielsen M, Almagro Armenteros JJ, Nielsen H, Sønderby CK,
570 Winther O, Sønderby SK: **An introduction to deep learning on biological sequence**
571 **data: examples and solutions**. *Bioinformatics* 2017, **33**(22):3685-3690.
572 10. Park Y, Kellis M: **Deep learning for regulatory genomics**. *Nature Biotechnology* 2015,
573 **33**(8):825-826.
574 11. Lancashire LJ, Lemetre C, Ball GR: **An introduction to artificial neural networks in**
575 **bioinformatics—application to complex microarray and mass spectrometry**
576 **datasets in cancer studies**. *Briefings in Bioinformatics* 2009, **10**(3):315-329.
577 12. Schmidt B, Hildebrandt A: **Deep learning in next-generation sequencing**. *Drug*
578 *Discov Today* 2021, **26**(1):173-180.
579 13. Stepniewska-Dziubinska MM, Zielenkiewicz P, Siedlecki P: **Development and**
580 **evaluation of a deep learning model for protein–ligand binding affinity prediction**.
581 *Bioinformatics* 2018, **34**(21):3666-3674.

582 14. Fioravanti D, Giarratano Y, Maggio V, Agostinelli C, Chierici M, Jurman G, Furlanello
583 C: **Phylogenetic convolutional neural networks in metagenomics**. *BMC*
584 *Bioinformatics* 2018, **19**(2):49.
585 15. Lo C, Marculescu R: **MetaNN: accurate classification of host phenotypes from**
586 **metagenomic data using neural networks**. *BMC Bioinformatics* 2019, **20**(12):314.
587 16. Wang Y, Bhattacharya T, Jiang Y, Qin X, Wang Y, Liu Y, Saykin AJ, Chen L: **A novel**
588 **deep learning method for predictive modeling of microbiome data**. *Brief Bioinform*
589 2021, **22**(3).
590 17. Zha Y, Chong H, Qiu H, Kang K, Dun Y, Chen Z, Cui X, Ning K: **Ontology-aware**
591 **deep learning enables ultrafast and interpretable source tracking among sub-**
592 **million microbial community samples from hundreds of niches**. *Genome Medicine*
593 2022, **14**(1):43.
594 18. Arango-Argoty G, Garner E, Pruden A, Heath LS, Vikesland P, Zhang L: **DeepARG:**
595 **a deep learning approach for predicting antibiotic resistance genes from**
596 **metagenomic data**. *Microbiome* 2018, **6**(1):23.
597 19. Ma Y, Guo Z, Xia B, Zhang Y, Liu X, Yu Y, Tang N, Tong X, Wang M, Ye X *et al*:
598 **Identification of antimicrobial peptides from the human gut microbiome using**
599 **deep learning**. *Nature Biotechnology* 2022, **40**(6):921-931.
600 20. Chong H, Zha Y, Yu Q, Cheng M, Xiong G, Wang N, Huang X, Huang S, Sun C, Wu S
601 *et al*: **EXPERT: transfer learning-enabled context-aware microbial community**
602 **classification**. *Briefings in Bioinformatics* 2022:bbac396.

603 21. Zhao Z, Woloszynek S, Agbavor F, Mell JC, Sokhansanj BA, Rosen GL: **Learning,**
604 **visualizing and exploring 16S rRNA structure using an attention-based deep**
605 **neural network**. In: *PLoS computational biology.* vol. 17; 2021: e1009345.
606 22. Abellan-Schneyder I, Matchado MS, Reitmeier S, Sommer A, Sewald Z, Baumbach J,
607 List M, Neuhaus K: **Primer, Pipelines, Parameters: Issues in 16S rRNA Gene**
608 **Sequencing**. *mSphere* 2021, **6**(1).
609 23. Woese CR, Kandler O, Wheelis ML: **Towards a natural system of organisms:**
610 **proposal for the domains Archaea, Bacteria, and Eucarya**. *Proc Natl Acad Sci U S*
611 *A* 1990, **87**(12):4576-4579.
612 24. Camacho C, Coulouris G, Avagyan V, Ma N, Papadopoulos J, Bealer K, Madden TL:
613 **BLAST+: architecture and applications**. *BMC Bioinformatics* 2009, **10**(1):421.
614 25. Yarza P, Yilmaz P, Pruesse E, Glöckner FO, Ludwig W, Schleifer K-H, Whitman WB,
615 Euzéby J, Amann R, Rosselló-Móra R: **Uniting the classification of cultured and**
616 **uncultured bacteria and archaea using 16S rRNA gene sequences**. *Nature Reviews*
617 *Microbiology* 2014, **12**(9):635-645.
618 26. Ebou A, Koua D, Zeze A: **HyperEx: A Tool to Extract Hypervariable Regions from**
619 **16S rRNA Sequencing Data**. *bioRxiv* 2021:2021.2009.2003.455391.
620 27. Callahan BJ, McMurdie PJ, Rosen MJ, Han AW, Johnson AJ, Holmes SP: **DADA2:**
621 **High-resolution sample inference from Illumina amplicon data**. *Nat Methods* 2016,
622 **13**(7):581-583.

623 28. Cole JR, Wang Q, Fish JA, Chai B, McGarrell DM, Sun Y, Brown CT, Porras-Alfaro
624 A, Kuske CR, Tiedje JM: **Ribosomal Database Project: data and tools for high**
625 **throughput rRNA analysis**. *Nucleic Acids Research* 2013, **42**(D1):D633-D642.
626 29. DeSantis TZ, Hugenholtz P, Larsen N, Rojas M, Brodie EL, Keller K, Huber T, Dalevi
627 D, Hu P, Andersen GL: **Greengenes, a chimera-checked 16S rRNA gene database**
628 **and workbench compatible with ARB**. *Appl Environ Microbiol* 2006, **72**(7):5069-
629 5072.
630 30. Felsenstein J: **Phylogenies and the Comparative Method**. *The American Naturalist*
631 1985, **125**(1):1-15.
632 31. Sankoff D: **Minimal Mutation Trees of Sequences**. *SIAM Journal on Applied*
633 *Mathematics* 1975, **28**(1):35-42.
634 32. Maddison WP: **Squared-Change Parsimony Reconstructions of Ancestral States**
635 **for Continuous-Valued Characters on a Phylogenetic Tree**. *Systematic Zoology*
636 1991, **40**(3):304-314.
637 33. Louca S, Doebeli M: **Efficient comparative phylogenetics on large trees**.
638 *Bioinformatics* 2017, **34**(6):1053-1055.
639 34. Katoh K, Misawa K, Kuma Ki, Miyata T: **MAFFT: a novel method for rapid multiple**
640 **sequence alignment based on fast Fourier transform**. *Nucleic Acids Research* 2002,
641 **30**(14):3059-3066.
642 35. Price MN, Dehal PS, Arkin AP: **FastTree 2 – Approximately Maximum-Likelihood**
643 **Trees for Large Alignments**. *PLOS ONE* 2010, **5**(3):e9490.

644 36. Bowman JS, Ducklow HW: **Microbial Communities Can Be Described by**
645 **Metabolic Structure: A General Framework and Application to a Seasonally**
646 **Variable, Depth-Stratified Microbial Community from the Coastal West Antarctic**
647 **Peninsula**. *PLOS ONE* 2015, **10**(8):e0135868.
648 37. Douglas GM, Maffei VJ, Zaneveld JR, Yurgel SN, Brown JR, Taylor CM, Huttenhower
649 C, Langille MGI: **PICRUSt2 for prediction of metagenome functions**. *Nature*
650 *Biotechnology* 2020, **38**(6):685-688.
651 38. Barbera P, Kozlov AM, Czech L, Morel B, Darriba D, Flouri T, Stamatakis A: **EPA-ng:**
652 **Massively Parallel Evolutionary Placement of Genetic Sequences**. *Systematic*
653 *Biology* 2019, **68**(2):365-369.
654 39. Buitinck L, Louppe G, Blondel M, Pedregosa F, Mueller A, Grisel O, Niculae V,
655 Prettenhofer P, Gramfort A, Grobler J *et al*: **API design for machine learning software:**
656 **experiences from the scikit-learn project**. *CoRR* 2013, **abs/1309.0238**.
657 40. Lundberg SM, Lee S-I: **A Unified Approach to Interpreting Model Predictions**. In:
658 *Advances in Neural Information Processing Systems: 2017*: Edited by Guyon I,
659 Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R. Curran
660 Associates, Inc. 2017.
661 41. Wolpert DH: **Stacked generalization**. *Neural Networks* 1992, **5**(2):241-259.
662 42. Breiman L: **Stacked regressions**. *Machine Learning* 1996, **24**(1):49-64.
663 43. Cortes C, Vapnik V: **Support-vector networks**. *Machine Learning* 1995, **20**(3):273-
664 297.

665 44. Ho TK: **Random Decision Forests**. In: *Proceedings of the Third International*
666 *Conference on Document Analysis and Recognition (Volume 1) - Volume 1: 1995*. M:
667 IEEE Computer Society 1995: 278-282.
668 45. Cover TM, Hart PE: **Nearest neighbor pattern classification**. *IEEE Trans Inf Theory*
669 1967, **13**:21-27.
670 46. Friedman JH: **Greedy function approximation: a gradient boosting machine**.
671 *Annals of statistics* 2001:1189-1232.
672 47. Hoerl AE, Kennard RW: **Ridge Regression: Biased Estimation for Nonorthogonal**
673 **Problems**. *Technometrics* 1970, **12**(1):55-67.
674 48. Tibshirani R: **Regression Shrinkage and Selection via the Lasso**. *Journal of the Royal*
675 *Statistical Society Series B (Methodological)* 1996, **58**(1):267-288.
676 49. Lindauer M, Eggensperger K, Feurer M, Biedenkapp A, Deng D, Benjamins C,
677 Ruhkopf T, Sass R, Hutter F: **SMAC3: A Versatile Bayesian Optimization Package**
678 **for Hyperparameter Optimization**. In: *ArXiv: 210909831: 2021*. 2021.
679 50. Loshchilov I, Hutter F: **SGDR: Stochastic Gradient Descent with Restarts**. *CoRR*
680 2016, **abs/1608.03983**.
681 51. He K, Zhang X, Ren S, Sun J: **Deep Residual Learning for Image Recognition**. In:
682 *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR): 27-30*
683 *June 2016 2016*. 770-778.
684 52. Moriya S: **Simple mapping-based quantification of a mock microbial community**
685 **using total RNA-seq data**. *PLoS One* 2021, **16**(7):e0254556.

686 53. Bray JR, Curtis JT: **An Ordination of the Upland Forest Communities of Southern**
687 **Wisconsin**. *Ecological Monographs* 1957, **27**(4):325-349.
688 54. Shapley LS: **A Value for N-Person Games**. Santa Monica, CA: RAND Corporation;
689 1952.
690 55. Štrumbelj E, Kononenko I: **Explaining prediction models and individual predictions**
691 **with feature contributions**. *Knowledge and Information Systems* 2014, **41**(3):647-665.
692 56. Schneider TD, Stormo GD, Gold L, Ehrenfeucht A: **Information content of binding**
693 **sites on nucleotide sequences**. *J Mol Biol* 1986, **188**(3):415-431.
694 57. Schneider TD, Stephens RM: **Sequence logos: a new way to display consensus**
695 **sequences**. *Nucleic Acids Res* 1990, **18**(20):6097-6100.
696 58. Quast C, Pruesse E, Yilmaz P, Gerken J, Schweer T, Yarza P, Peplies J, Glöckner FO:
697 **The SILVA ribosomal RNA gene database project: improved data processing and**
698 **web-based tools**. *Nucleic Acids Research* 2012, **41**(D1):D590-D596.
699 59. Zhang Y, Kang B, Hooi B, Yan S, Feng J: **Deep Long-Tailed Learning: A Survey**.
700 *CoRR* 2021, **abs/2110.04596**.
701 60. Devlin J, Chang M-W, Lee K, Toutanova K: **Bert: Pre-training of deep bidirectional**
702 **transformers for language understanding**. *arXiv preprint arXiv:1810 04805* 2018.
703 61. Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, Neelakantan A, Shyam
704 P, Sastry G, Askell A *et al*: **Language Models are Few-Shot Learners**. *CoRR* 2020,
705 **abs/2005.14165**.
706

## TABLES

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

| | 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 | 0.928 ± 0.0331 | 0.889 ± 0.0476 | 0.902 ± 0.0447 | 0.916 ± 0.0569 | 0.910 ± 0.0340 | 0.909 ± 0.0563 | 0.931 ± 0.0354 |
| MPR | 0.787 ± 0.0375 | 0.867 ± 0.0370 | 0.836 ± 0.0305 | 0.858 ± 0.0477 | 0.879 ± 0.0475 | 0.858 ± 0.0413 | 0.852 ± 0.0400 | 0.846 ± 0.0370 |
| PIC | 0.789 ± 0.0243 | 0.810 ± 0.0281 | 0.805 ± 0.0307 | 0.789 ± 0.0503 | 0.824 ± 0.0377 | 0.785 ± 0.0277 | 0.781 ± 0.0404 | 0.802 ± 0.0393 |
| SA | 0.774 ± 0.0329 | 0.829 ± 0.0321 | 0.805 ± 0.0430 | 0.811 ± 0.0379 | 0.822 ± 0.0388 | 0.810 ± 0.0283 | 0.807 ± 0.0391 | 0.830 ± 0.0334 |
| WSCP | 0.775 ± 0.0273 | 0.793 ± 0.0242 | 0.782 ± 0.0330 | 0.779 ± 0.0483 | 0.823 ± 0.0405 | 0.785 ± 0.0354 | 0.779 ± 0.0377 | 0.790 ± 0.0375 |
| TA(RDP) | 1.18 ± 0.0356 | 1.58 ± 0.0538 | 1.41 ± 0.0517 | 1.47 ± 0.0370 | 1.93 ± 0.0954 | 1.48 ± 0.0622 | 1.44 ± 0.0329 | 1.73 ± 0.129 |
| TA(Greengenes) | 2.52 ± 0.0388 | 2.08 ± 0.122 | 2.63 ± 0.147 | 1.85 ± 0.0939 | 2.12 ± 0.0590 | 2.00 ± 0.0961 | 2.19 ± 0.232 | 2.54 ± 0.434 |

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

715 **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 |

716

## FIGURES

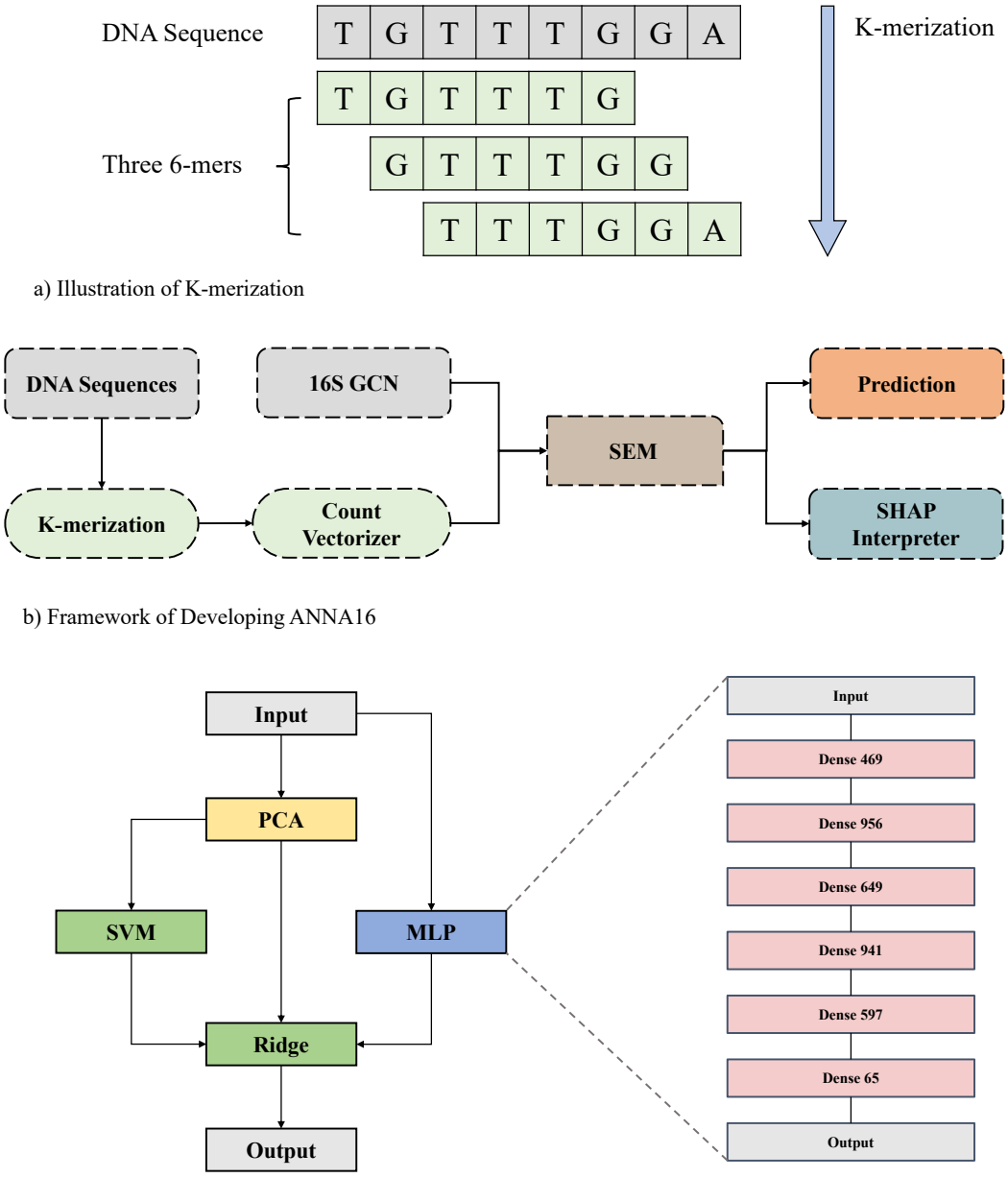


**Figure 1.** Illustration of the Framework and Architecture of the Deep Learning Models. a) K-mer strategy (K = 6). b) Framework of Designing ANNA16. c) Architecture of the Stacked Ensemble Model. d) Architecture of the MLP model inside the Stacked Ensemble Model.

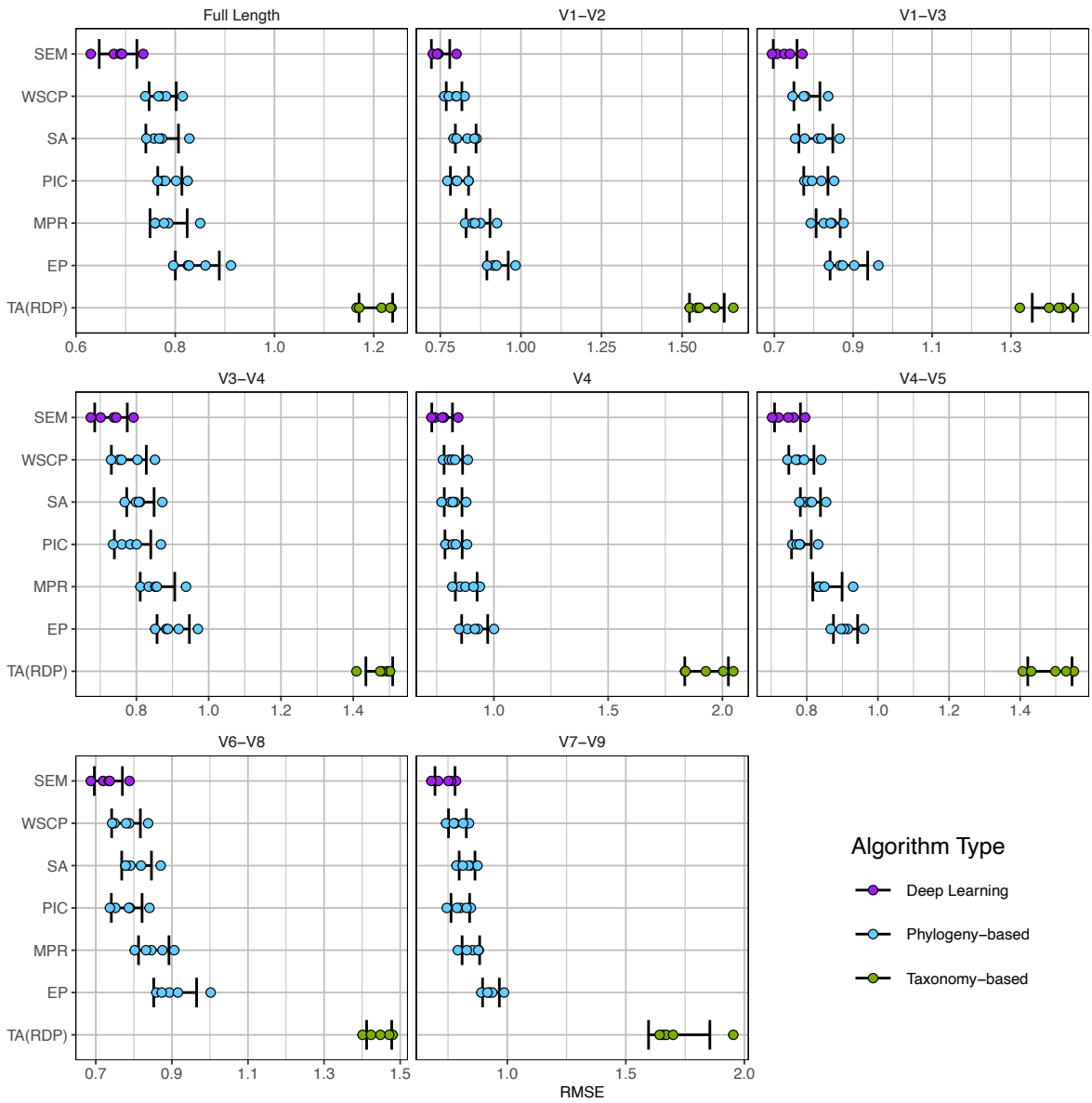


**Figure 2.** Performance of all algorithms or models at 16S full-length and subregions. EP: empirical probabilities; MPR: Sankoff's maximum-parsimony; PIC: phylogenetic independent contrast; SA: Subtree Averaging; WSCP: weighted-squared-change parsimony; TA(RDP): sub-taxa averaging based on RDP database; SEM: stacked ensemble model.

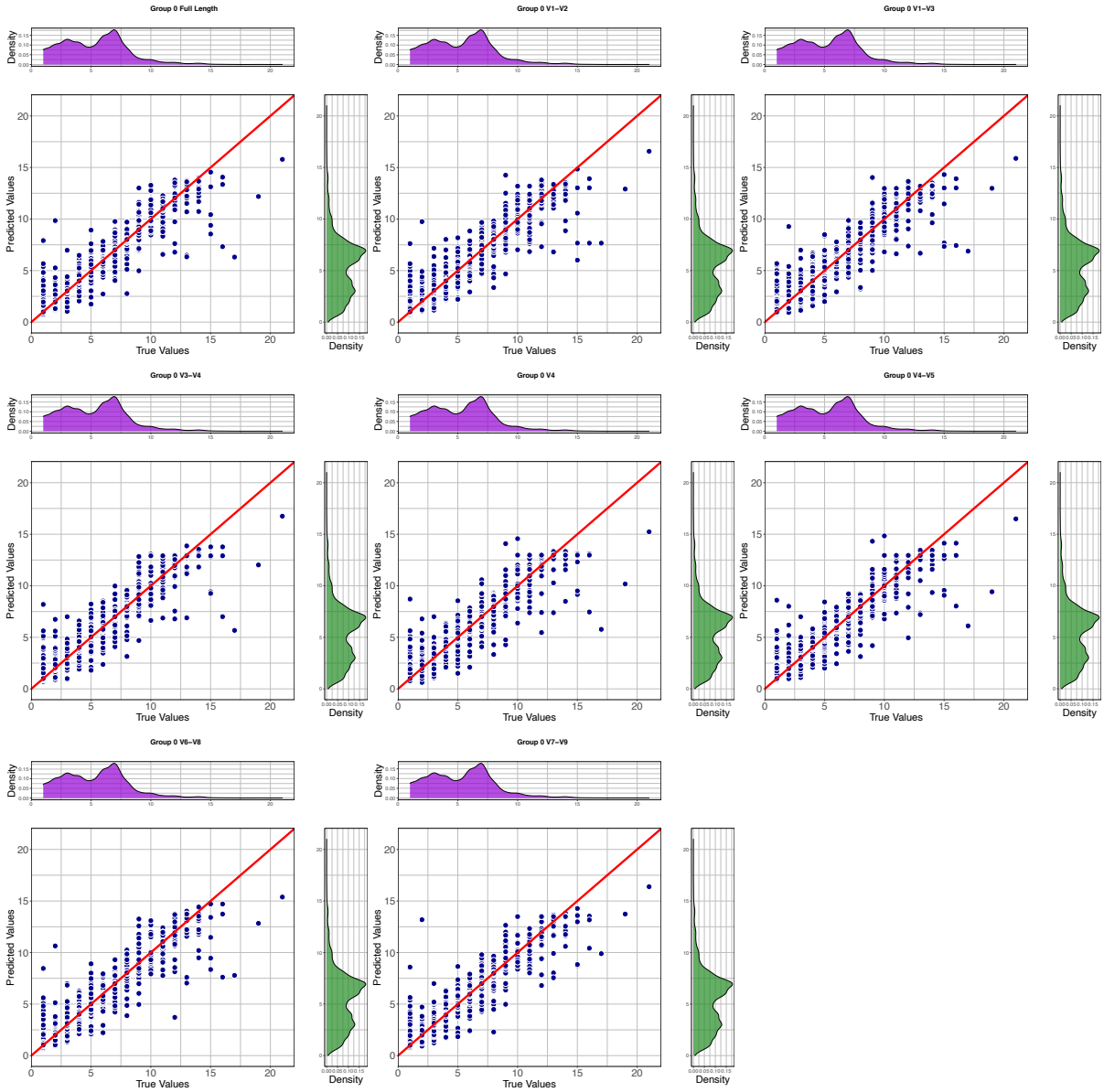


**Figure 3.** Prediction by SEM on 16S full length and subregions in the first cycle of 5-fold cross-validation. The red line is a diagonal line illustrating to which extent the predicted values are derived from the true values. The density plot filled in violet shows the distribution of the true values,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

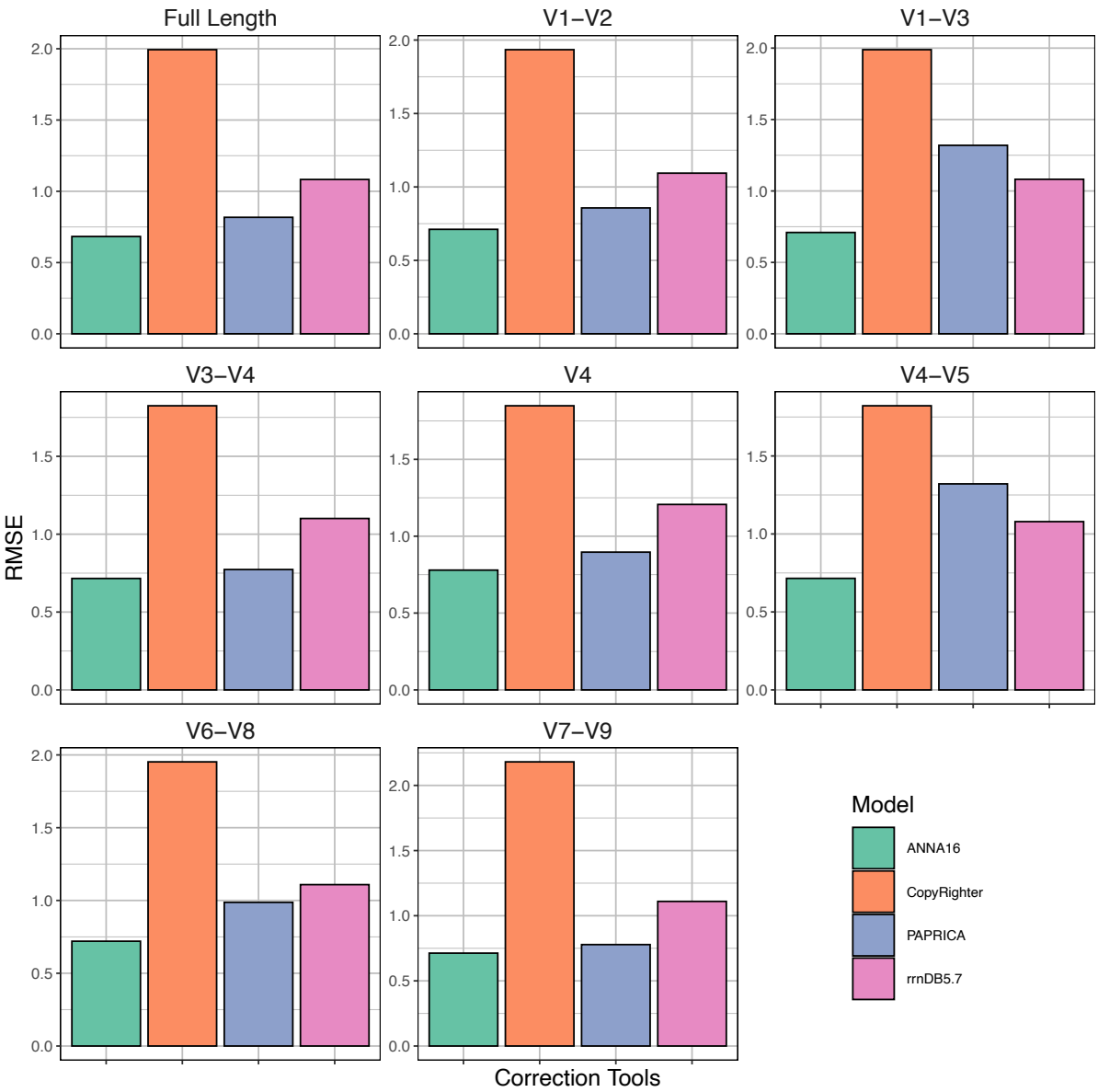


**Figure 4.** Comparison of performance on the test dataset among CopyRighter, rrnDB, ANNA16, and PAPRICA.

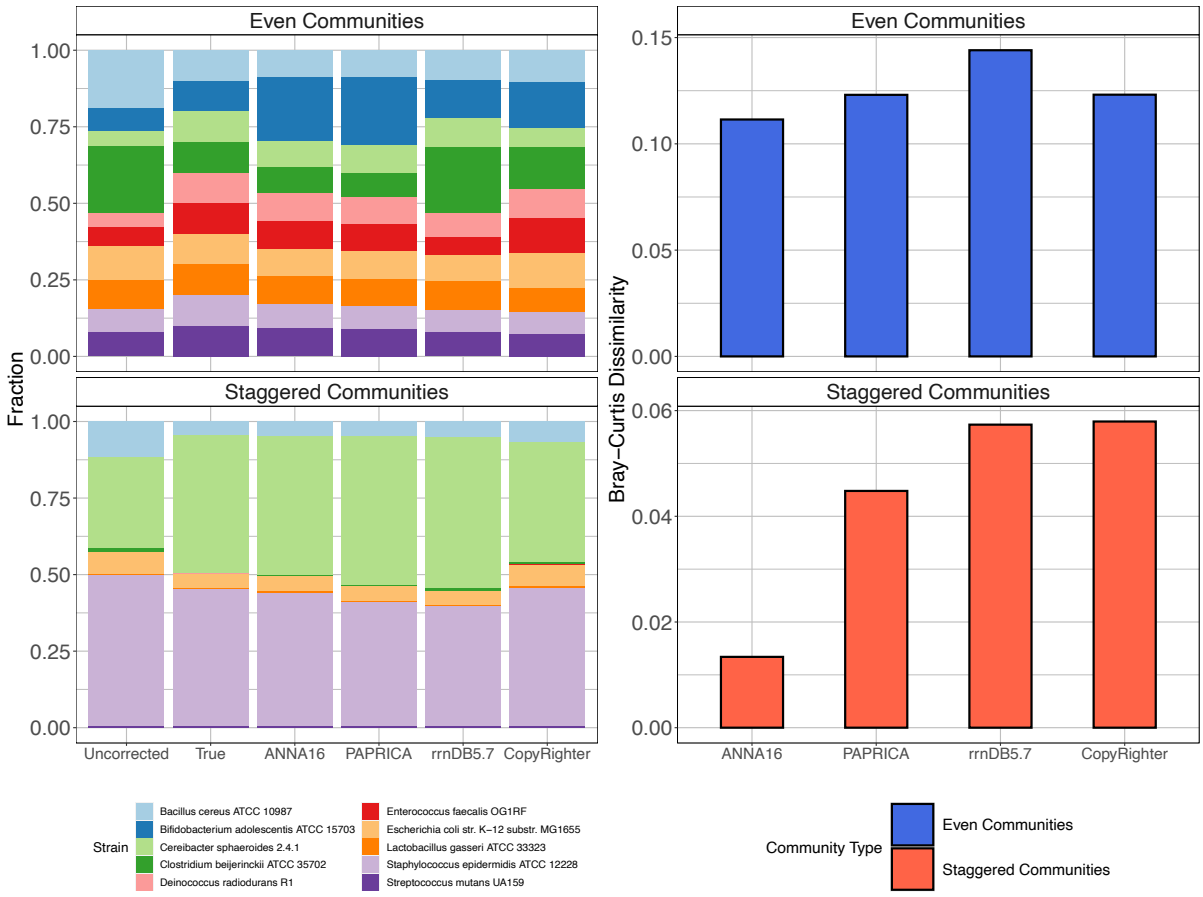


**Figure 5.** Uncorrected copy fraction, true cell fraction, and cell fraction (microbial profiles) estimated by ANNA16, rrnDB (version 5.7), CopyRighter, and PAPRICA. The right panel shows the value of Bray-Curtis Dissimilarity of each estimated profile to the true cell fraction. Uncorrected: Uncorrected 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.

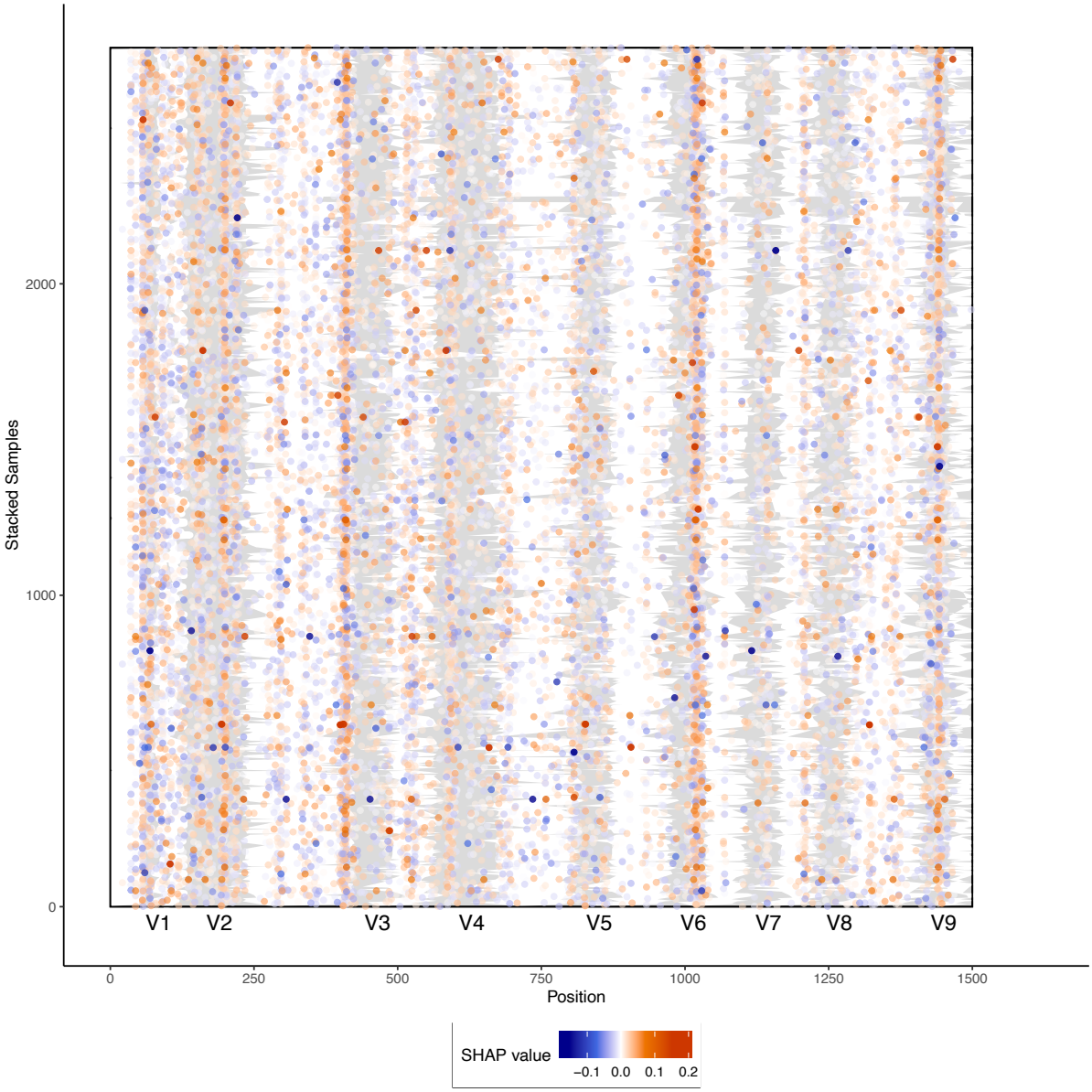


**Figure 6.** Distribution of SHAP values at 16S full-length sequences. To make the figure interpretable, only the value of the top 100 K-mers ranked by mean absolute SHAP value is shown. Hypervariable regionsare labeled as grey shadows. V-region labels on the x-axis show the general position of *E. coli* V-regions according to coordinates from Yarza et al. [25].

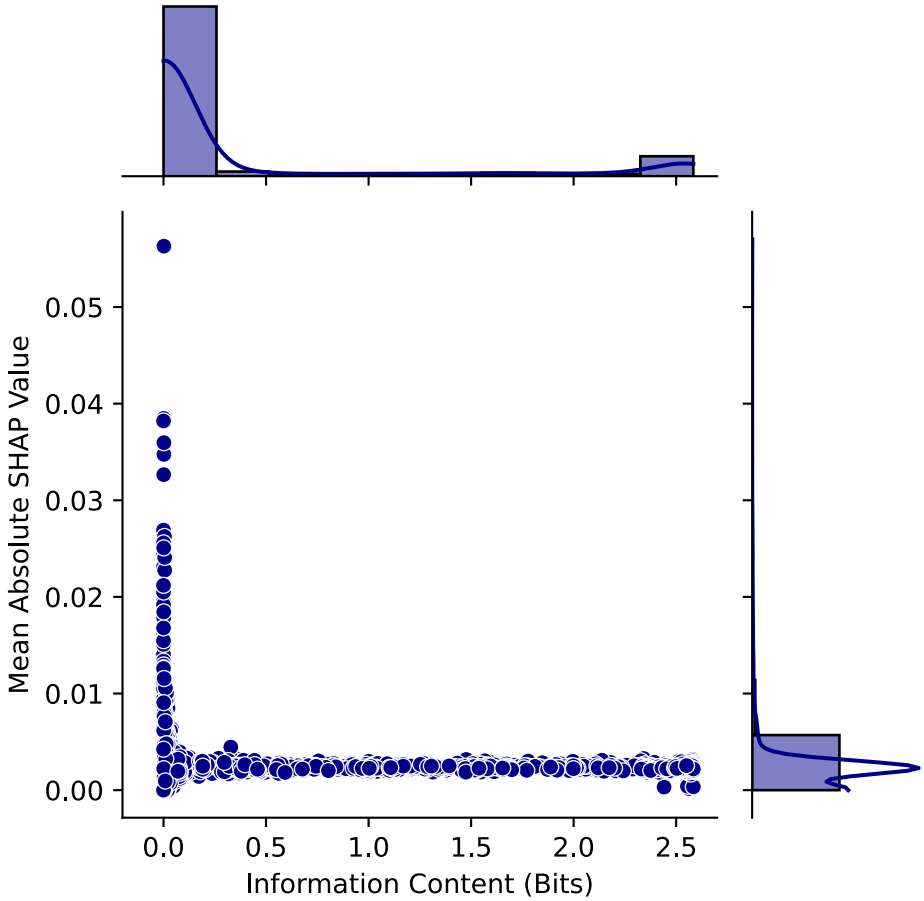


**Figure 7.** Relationship of mean absolute SHAP values and the information content (unit: bits) on each position of aligned 16S full-length sequence.