A Novel Method 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

Culture-independent 16S rRNA gene metabarcoding is a commonly used method in microbiome profiling. However, this approach can only reflect the proportion of sequencing reads, rather than the actual cell fraction. To achieve more quantitative cell fraction estimates, we need to resolve the 16S gene copy numbers (GCN) for different community members. Currently, there are several bioinformatic tools available to estimate 16S GCN, either based on taxonomy assignment or phylogeny. Here we develop a novel algorithm, Stacked Ensemble Model (SEM), that estimates 16S GCN directly from the 16S rRNA gene sequence strings, without resolving taxonomy or phylogeny. For accessibility, we developed a public, end-to-end, web-based tool based on the SEM model, named ANNA16. Based on 27,579 16S rRNA gene sequence data (rrnDB database), we show that ANNA16 outperforms the most commonly used 16S GCN prediction algorithms. The prediction error range in the 5-fold cross validation of SEM is completely lower than all other algorithms for the 16S full-length sequence and partially lower at 16S subregions. The final test and a mock community test indicate ANNA16 is more accurate than all currently available tools (i.e., rrnDB, CopyRighter, PICRUSt2, & PAPRICA). SHAP value analysis indicates ANNA16 mainly learns information from extremely variable nucleic positions, rather than merely the hypervariable regions. Compared to the traditional GCN prediction tools, ANNA16 has a simple structure, faster inference speed without precomputing, and higher accuracy. If more 16S GCN data is available, future studies could improve the larger prediction error for high GCN taxa caused by data imbalance.

43 # 1 INTRODUCTION

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

54 One critical limitation of 16S rRNA metabarcoding is that NGS determines the proportion
55 of 16S rRNA read counts, instead of cell counts. Because it is typical for prokaryotes to have
56 more than one copy of the 16S rRNA gene, the proportion of 16S rRNA read counts (also called
57 relative abundance) could not exactly reflect the taxonomic composition of a microbial
58 community [5]. To represent the profile of a microbial community, the proportion of 16S rRNA
59 read counts needs to be weighted by an inverse of 16S rRNA gene copy number (GCN). This
60 step yields a corrected microbial composition as the community profile, which represents the
61 proportion of cell counts of each lineage in the community [5]. To calculate 16S GCN,
62 scientists rely on experimental methods such as whole genome sequencing [6] or competitive
63 PCR [7]. Nevertheless, experimental methods are usually costly or culture-dependent, and so

64 far, only a limited number of species have GCN data. This situation catalyzes the development
65 of bioinformatics tools that predict the GCN of unmeasured species from measured species.
66 Luckily, 16S GCN is correlated with relatedness across species [5, 8], which means 16S GCN
67 variation is relatively low within species but increases as relatedness decreases. This trait
68 allows 16S GCN to be estimated from phylogeny or taxonomy [8]. In general, all current tools
69 rely on this kind of method [3, 5, 6]. Specifically, one correction tool could construct one
70 taxonomy or phylogenetic tree of the 16S sequence and estimate the GCN of the unmeasured
71 species from its close measured relatives. For instance, rrnDB [6] is one well-established tool
72 that corrects 16S GCN based on taxonomy. Its algorithm calculates the mean 16S GCN of a
73 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, ANN is
84 even competent to disclose relationships or mechanisms that are still unknown to scientists

85 [13]. Deep learning gets its name by having more than one (hidden) layer connecting the input
86 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 is not any deep learning-based method 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, at
103 present, it is less common for microbiologists to sequence the full length of 16S rRNA [22].
104 Full-length 16S could be more commonly used only if new long sequencing methods are
105 invented. Alternatively, microbial studies tend to target the hypervariable regions in 16S rRNA.

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

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

117

118 2 MATERIALS AND METHODS

119 2.1. Data Source and Preprocessing

120 16S rRNA sequences and the GCN data used to train deep learning models in this study
121 are retrieved from rrnDB, version 5.8 [6]. In total, 29,267 16S genes with unique accession
122 numbers are included in the dataset. Because raw 16S sequence data contains both (+) strands
123 and (-) strands, bias may be caused if the data is directly used for taxonomy-based and
124 phylogeny-based algorithms. Therefore, this study uses the BLASTN algorithm in the
125 BLAST+ application, version 2.11.0 [24], with the first 50 nucleotides of *E. coli* 16S sequence
126 (+) strand (accession number: GCF_002953035.1) being the subject, to determine the

127 orientation of each 16S sequence. The first 50 nucleotides of E. coli belong to the conserved
128 region on 16S rRNA [25], so it qualifies as a template. Returned HSPs with sstart > send and
129 evalue < 0.01 are considered (+) strands, and HSPs with sstart < send and evalue < 0.01 are
130 considered (-) strands. The remaining sequences are labeled as unrecognizable and dropped
131 from the dataset. The reverse complement of (-) strand sequences is generated and added to the
132 (+) strands.

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

144 Because a previous study points out that the performance of all the existing GCN
145 correction tools is unsatisfactory on clades that are distant from training data [3], this study
146 would like to test if deep learning could perform better on these distant clades. Therefore, this
147 study splits the preprocessed dataset into train and test datasets according to when the

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

161 **2.2. Conventional Algorithms and Existing Tools**

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

169 copy number of every node by taking the mean of its sub-nodes. If an unknown query is given,
170 it assigns the value of the query's parent node to the query node as an estimation.

171 This study also employs six phylogeny-based algorithms to compare with deep learning
172 methods: empirical probabilities (EP), subtree averaging (SA), phylogenetic independent
173 contrasts (PIC) [30], Sankoff's maximum-parsimony (MPR) [31], and weighted-squared-
174 change parsimony (WSCP) [32]. These algorithms first receive a pre-constructed tree of 16S
175 sequences and estimates the GCN of each ancestral node from its daughter species through
176 weighing each labeled daughter node with its phylogenetic distance to the parental node [33].
177 To implement the algorithms, 16S sequences in the dataset first undergo multiple sequence
178 alignment by MAFFT v7.490 [34], and are converted to a phylogenetic tree by FastTree 2.1.10
179 [35]. The algorithms are performed by castor package in R [33], adjusted from the R script in
180 Louca, Doebeli, and Parfrey (2018) [3] (MPR is implemented by function *hsp_max_parsimony*,
181 WSCP by function *hsp_squared_change_parsimony*, SA by function *hsp_subtree_averaging*,
182 EP by function *hsp_empirical_probabilities*, and PIC by *hsp_independent_contrasts*). All
183 functions are set on default parameters.

184 What is more, after the deep learning-based tool is developed, this study compares it with
185 all the existing tools on the test dataset that can be used for 16S GCN correction, which are
186 rrnDB [6], CopyRighter [5], PAPRICA [36], PICRUSt2 [37]. rrnDB is a taxonomy-based tool
187 using sub-taxa averaging, while the latter three tools are phylogeny-based. CopyRighter uses
188 PIC algorithm to pre-compute the copy number of all taxa in the Greengenes database [29] and
189 makes predictions by searching the lineages of the queries in its pre-computed GCN table [5].

190 The original function of PAPRICA and PICRUSt2 is to predict the metabolic profile of a
191 microbial community based on the 16S data, but they possess the function of predicting the
192 copy number of any given gene according to the phylogeny [36, 37]. These two tools first
193 utilize EPA-ng [38] to place the queries on their built-in phylogenetic tree, and use subtree-
194 averaging (PAPRICA) or maximum parsimony (PICRUSt2) algorithms to predict the copy
195 number of the queries [36, 37].

196 Because rrnDB and CopyRighter application will directly output microbial community
197 abundance instead of 16S GCN values, this study retrieves their precomputed tables (rrnDB
198 version 5.7) and predict GCN values by searching the precomputed tables for assigned lineages.
199 Taxonomy of full-length sequences and all subregion sequences are generated by the
200 *assignTaxonomy* function in the DADA2 package through RDP (version 18) and Greengenes
201 (version 13.8, formatted by DADA2) [29] database. The assigned RDP lineages are searched
202 in the rrnDB precomputed table, and Greengenes lineages are searched in CopyRighter
203 precomputed table. As for PAPRICA and PICRUSt2, the DNA sequences in the test dataset are
204 direct input to the tools. For PAPRICA, *paprica-run.sh* is called without additional parameters.
205 For PTCRUSt2, *place_seqs.py* and *hsp.py* is 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. The K-mers are converted into the
212 number of counts in the set via *CountVectorizer* function in the the scikit-learn module (version
213 0.23.2) [39]. Deep learning models will intake the vector of occurrence counts and predicts the
214 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, but the requirement for memory size also expands sharply. Therefore,
217 the value 6 appears to be the optimal length which results in a good performance and does not
218 require an extremely large memory size. 6-merization is to decompose the DNA sequence into
219 a set of sub-sequences of length 6. An example of a set with three 6-mers is displayed in Figure
220 1 a). After determining the length of K-mers, a Stacked Ensemble Model (SEM) is
221 implemented for 16S GCN prediction. The framework is demonstrated in Figure 1 b). After
222 model construction, the performance of the SEM is compared with conventional algorithms
223 through a 5-fold cross-validation at full-length and subregional DNA sequence. Shapley
224 Additive exPlanations (SHAP) [40] is introduced here for post-hoc interpretability. The whole
225 study was conducted with Python 3.8 and Tensorflow 2.8.0 on Nividia RTX 3090.

226

227 2.3.2. Model Construction

228 The deep learning model employs one strategy of ensemble learning: model stacking,
229 which first trains multiple base-models on the available data, and then uses a meta-model to
230 accept the output of all base-models and make the final prediction. The stacked model would
231 typically perform better than any single base-model [41, 42]. In this study, the Stacked

232 Ensemble Model is a combination of one MLP model with other classic machine learning
233 models. MLP is the most common form of ANN, in which artificial neurons are organized into
234 layers and the input is processed layer by layer [11]. The calculation of the output for each
235 neuron is as follows [11]:

236 $$v_k = \sum_{i=0}^{n} \omega_{k,i} x_i$$

237 $$\gamma_k = \varphi(v_k + v_{k_0})$$

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

242 $$\text{RMSE} = \sqrt{\frac{\sum_{i=0}^{n}(Y_{true} - Y_{pred})^2}{n}}$$

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

245 In addition to MLP, classic machine learning algorithms including Support Vector Machine
246 (SVM) [43], Random Forest [44], K-nearest Neighbors [45], and Gradient Boosting [46] are
247 tested as base-models. Linear Regression, Ridge Regression [47], and Lasso Regression [48]
248 are tested as meta-models. All machine learning algorithms are implemented by the scikit-learn
249 module (version 0.23.2). After testing, the SEM uses one MLP and one SVM as the base-
250 models and Ridge Regression as the meta-model, as shown in Figure 2 c). Principle Component
251 Analysis (PCA) is implemented by *PCA* function in scikit-learn with the value of

252 *n_components* set to 100, SVM is implemented by *SVR* function in scikit-learn, and Ridge
253 Regression is implemented by *Ridge* function in scikit-learn.

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

268

269 2.3.3. Dismissed Machine Learning Algorithms

270 This study also used tested one Transformer model and one Residual Multi-layer
271 Perceptron for 16S GCN prediction. The Transformer model leveraged the original transformer
272 encoder architecture with cosine learning rate decay [50], followed by global average pooling,

273 and two dense layers. Embedding and position encoding are used as input for the Transformer
274 model. Residual Multi-layer Perceptron are MLP models utilizing Residual Connection
275 architectures. In this architecture, skip connections are built to jump over two to three layers.
276 Such design may help avoid the problem of vanishing gradients or degradation [51]. However,
277 because these models do not exhibit a good performance, they are dismissed from the study.
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. In addition to the final test, this study designs a case study
283 that simulates two mock communities *in silico* to demonstrate ANNA16's performance. rrnDB,
284 CopyRighter, PAPRICA, and PTCRUSt2 are also applied to estimate the mock communities'
285 microbial profile as a positive control. The simulated communities are two commercially
286 available mock communities, one even community (ATCC MSA-2003) and one staggered
287 community (ATCC MSA-1001). The information of the mock communities is shown in Table
288 S3. Such mock communities have been used by Moriya (2021) to study methods of quantifying
289 a microbial community [52]. Both mock communities consist of similar 10 bacterial strains,
290 but with different ratios. This study retrieves genome sequences of the 10 strains from NCBI,
291 copy number data from rrnDB (version 5.7), and assigns a cell counts value to each strain by
292 assuming each community has 1,000,000 cells. Table S3 shows the basic information about the
293 mock communities. After genome sequences of the ten strains are retrieved from NCBI,

294 HyperEx is used to extract 16S full-length sequences from the genomes. The method is the
295 same as described in the Data Source and Preprocessing.

296 After estimating the profile of the mock communities, Bray-Curtis Dissimilarity [53] is
297 calculated between estimated cell fraction and true cell fraction to represent the accuracy of
298 community profile prediction. Bray-Curtis Dissimilarity is a method widely used in ecology
299 and biology for quantifying the compositional dissimilarity between two different communities
300 [53]. The larger the Bray-Curtis Dissimilarity value is, the more distinct two communities will
301 be. Its calculation is as follows:

302 $$BC = 1 - \frac{2C_{AB}}{S_A + S_B}$$

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

308

309 **2.5. Deep Learning Model Explanation**

310 SHAP [40] is used to further explain our model prediction. SHAP is based on both
311 collational game theory [54] and the classical Shapley method [55]. It is a unified measure of
312 feature importance that determines the individual contribution of each feature to a model
313 prediction by viewing each feature as a player in Shapley. The marginal contribution can be
314 estimated under multiple feature subsets, and the final contribution is the sum of all marginal

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

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

336 $$H_i = -\sum f_{char,i} \times \log_2(f_{char,i}), char \in [A, T, C, G, -, N]$$

337 $$R'_i = \log_2(6) - H_i - (\frac{1}{\ln 2} \times \frac{s-1}{2n})$$

338 $$R_i = R'_i \times \sum f_{nt,i}, nt \in [A, T, C, G]$$

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

345

346 **3 RESULTS**

347 This study first compares the performance of SEM with all other potential algorithms for
348 the task of 16S GCN correction. Figure 2 and Table 1 summarize the results of 5-fold cross-
349 validation. First, looking at the performance of these algorithms for the 16S full-length, sub-
350 taxa averaging based on RDP and Greengenes exhibit a mean RMSE of 1.18 and 2.52
351 copies/genome respectively, with a standard deviation of RMSE being 0.0356, and 0.0388.
352 Because TA(Greengenes) has much higher Mean RMSE at all regions, the results are not shown
353 in Figure 2. RDP proves to be the more accurate database for this task which produces the
354 lowest error range. All phylogeny-based algorithms perform better than sub-taxa averaging (EP:
355 Mean = 0.845, SD = 0.0443; MPR: Mean = 0.787, SD = 0.0375; PIC: Mean = 0.789, SD =
356 0.0243; SA: Mean = 0.774, SD = 0.0329; WSCP: Mean = 0.775, SD = 0.0273). The error range

357 of all phylogeny-based algorithms is lower than those of taxonomy-based algorithms with no
358 overlap. With low RMSE and small SD, WSCP turns out to be the optimal phylogeny-based
359 algorithm. As for deep learning models, SEM outperforms all other algorithms with the lowest
360 Mean RMSE (0.685 copies/genome) with a standard deviation of 0.0379. Its error range on the
361 16S full-length is the lowest among all algorithms and has no overlap with the phylogeny-
362 based algorithms.

363 Next, Figure 2 and Table 1 also show the performance of the algorithms for the 16S
364 subregions. Compared with the performance for 16 full-length, all algorithms produce an
365 increased Mean RMSE on subregions. EP experiences the most significant increase, with the
366 Mean RMSE rising to 0.9312 copies/genome on the V7-V9 region. WSCP is the most stable
367 phylogeny-based algorithm, with Mean RMSE on subregions ranging from 0.7792 (V3-V4) to
368 0.8227 (V4) copies/genome. The mean RMSE of SEM on subregions is relatively consistent,
369 ranging from 0.731 (V1-V3) to 0.783 (V4) copies/genome. Notably, as shown in Figure 2, the
370 error range of SEM on subregions is lower than EP and MPR in all subregions, and partially
371 lower than SA, PIC and WSCP. This indicates the performance of SEM on subregions is
372 consistent with that on full-length, completely better than EP and MPR, and partially better
373 than SA, PIC, and WSCP.

374 Figure 3 illustrates the prediction produced by SEM in the first cycle of cross-validation.
375 It can be seen that the predicted values possess a good linear relationship with the true values.
376 The predicted values and true values also have similar patterns in data distribution. But one

377 issue is that the models tend to make underestimated predictions when the true value > 12. This
378 will be further discussed in the discussion section.

379 Due to its superior performance, a bioinformatics tool, ANNA16, is developed based on
380 SEM and compared with other existing tools in the final test. Figure 4 and Table 2 demonstrate
381 the results of the final test. Because PICRRUSt2 has a much higher RMSE (Table 2), it is not
382 shown in Figure 4 and excluded from the case study. Figure 4 and Table 2 show that ANNA16
383 achieves the lowest RMSE among the tools at full-length and every subregion. The smallest
384 RMSE of SEM appears at full length, which is 0.683 copies/genome, and the highest RMSE is
385 at the V4 region, being 0.780 copies/genome. In contrast, the RMSE of CopyRighter ranges
386 from 1.822 (V4-V5) to 2.181 (V7-V9) copies/genome, rrnDB from 1.079 (V4-V5) to 1.207
387 (V4) copies/genome, PAPRICA from 0.773 (V3-V4) to 1.321 (V4-V5), and PICRUSt2 from
388 3.459 (V4-V5) to 4.731 (V6-V8). The performance of ANNA16 is better than other tools.

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

398 staggered community. For the even community, PAPRICA and CopyRighter exhibit very close
399 dissimilarity values and are the second accurate tools. For the staggered community, PAPRICA
400 is the second accurate tool, followed by rrnDB, but CopyRighter becomes the most accurate
401 one. In summary, ANNA16 has the most similar patterns with true cell fraction compared to
402 the counterparts of other tools.

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

418

419 **4 DISCUSSION**

420 In summary, this study aims to use the Stacked Ensemble Model (SEM), one deep learning
421 model, to predict 16S GCN based on 16S sequence. The performance of SEM is tested at the
422 16S full-length and subregions and compared with taxonomy- and phylogeny-based algorithms.
423 Finally, ANNA16–a bioinformatic tool based on SEM–is subsequently developed and
424 compared to existing 16S GCN correction tools, rrnDB, CopyRighter, PAPRICA, PICRUSt2,
425 through the final test and a case study. The SHAP method is used to explain how ANNA16
426 finishes the task.

427 First and foremost, ANNA16 is also the most accurate and state-of-the-art GCN correction
428 tools for 16S full-length, as well as for all subregions tested in this study. As shown in Figure
429 2, the algorithm used by ANNA16—Stacked Ensemble Model of MLP, SVM, and Ridge
430 (SEM)—exhibits an error range that is completely lower than the error ranges of all taxonomy-
431 based and phylogeny-based algorithms at 16S full-length. As for subregions, SEM still
432 maintains the lowest error ranges although having overlaps with the counterparts of PIC, SA,
433 WSCP. These results indicate SEM is capable of more accurately correcting GCN than existing
434 algorithms at all regions tested, and its main advantage lies at 16S full-length. In addition, as
435 Figure 4 and Table 2 shows, the RMSE of ANNA16 is the lowest among the tools at full-length
436 and every subregion in the final test. This indicates ANNA16 has a stronger ability of predicting
437 the GCN newly discovered species than other tools. Figure 5, Table S4, and Table S5 of the
438 case study also confirms this indication, where the true cell fraction of both communities is
439 more similar to those presented by ANNA16, especially for the staggered community.

440 Altogether, the results prove that ANNA16 is the most accurate 16S GCN correction tool for
441 the time being.

442 In addition to its superior performance, another advantage of ANNA16 is that it achieves
443 a fast inference speed without pre-computation. In general, existing GCN correction tools can
444 be classified as two types by inference strategies: pre-computation or real-time phylogenetic
445 placement. On the one hand, rrnDB and CopyRighter are pre-computational tools—they pre-
446 compute the GCN value for every taxon in the RDP database or Greengenes database respective
447 [5, 6]. On the other hand, PAPRICA and PICRUSt2 utilize real-time phylogenetic placement.
448 When a query is given, the tools first place the query sequence to its built-in phylogenetic tree
449 and make the prediction according to the new tree. Compared with real-time phylogenetic
450 placement, pre-computation strategy has a much short inference time, because the tools only
451 need to searche the input lineages from the pre-computed GCN table, instead of computing the
452 value from scratch. However, this pre-computation strategy also means a pre-computed
453 correction tool will be bound to its database, which is a critical disadvantage of pre-computed
454 tools. Due to the existence of differences in nomenclature among databases (for example, the
455 SILVA database [58] merges *Allorhizobium*, *Neorhizobium*, *Pararhizobium*, and *Rhizobium* as
456 one genus, named *Allorhizobium-Neorhizobium-Pararhizobium-Rhizobium*, but in RDP these
457 are independent genera), increased error might be caused if a pre-computed tool is applied to a
458 taxonomy generated by a miscellaneous database. Fortunately, the deep learning method
459 implemented in this study is free of this *accuracy-or-time* dilemma. Users could predict the

460 GCN of each 16S sequence and assign taxonomy separately at a much shorter time than using
461 PAPRICA or PICRUSt2.

462 After proving the supremacy of ANNA16, it becomes an interesting question–how the
463 model perceives the complex relationship between K-mer counts and 16S GCN?
464 Conventionally, microbiologists believe the nine hypervariable regions are the most
465 informative regions on the 16S sequence that could reflect the phylogeny of one species.
466 Therefore, one hypothesis is that K-mers from hypervariable regions would contribute more to
467 prediction than those from conserved regions. However, Figure 6 shows that the distribution
468 of SHAP values does not completely fit in the location of hypervariable regions. Though high
469 SHAP values indeed tend to aggregate in V2, V3, V5, V6, V7, and V9 regions, conserved
470 regions between V2-V3, V4-V5, and V8-V9 also contribute considerably.

471 Since Figure 6 alone is not evident enough to accept or reject the hypothesis, Figure 7 is
472 drawn. By plotting information content against mean absolute SHAP values on each position,
473 we expect to see a linear relationship. However, the scatters are unexpectedly L-shaped, with
474 high SHAP values aggregating at positions with extremely low information content. SHAP
475 values are generally constant in other positions. This trend indicates extremely variable
476 positions or positions with insertion of extremely low frequency are most informative for
477 ANNA16. The other positions provide nearly the same level of information, no matter how
478 conserved they are. Deducing from this indication, this means ANNA16 can detect informative
479 positions and assign high weights to the positions. This may help explain why ANNA16 or
480 MLP has better performance than conventional algorithms. When implementing conventional

481 algorithms, variance in the sequence is converted to sequence similarity, and the sequence
482 similarity is to generate phylogeny or taxonomy. This approach weighs every position equally.
483 Therefore, informative positions can be underrepresented and non-informative regions
484 overrepresented in conventional algorithms. However, by weighing K-mers unequally
485 according to how much information they contain, ANNA16 could overcome this problem and
486 utilize the information contained in the 16S sequence more effectively. This finding reveals the
487 great potential of deep learning in genetics. On the one hand, deep learning could accurately
488 predict a trait based on sequencing data. On the other hand, by investigating the contribution
489 of each feature, scientists could discover which genes or loci are most related to the trait.

490 Several aspects of this study could still be improved. First, one limitation of ANNA16 is
491 the underestimation of species with high GCN. As can be seen in Figure 3, ANNA16 would
492 make underestimated prediction when the true GCN $\geq$ 14, and those predicted values can be
493 far from true values. This limitation could be attributed to the imbalance in the dataset. Figure
494 S2 shows the distribution of GCN in the preprocessed dataset. Sequences with GCN $\geq$ 12
495 account for only a small proportion of the whole dataset, and sequences with GCN > 15 are
496 extremely rare. This makes it difficult for the model to learn features of high GCN sequences.
497 The most feasible method to solve this problem is to wait for the increase in data. Thanks to
498 the invention of new sequencing techniques, nowadays genomic data is accumulating at an
499 exponential rate [59]. If more data about high GCN species could be collected, the
500 underestimation would be correspondingly relieved. Apart from waiting, the data science
501 community has developed a number of methods to relieve data imbalance, including class re-

502 balancing, information augmentation, and module improvement [60]. These methods could be
503 applied to ANNA16 and may solve the problem of underestimation. Lastly, as noted, currently
504 the amount of 16S GCN data is still limited. In this case, it is worth trying transfer learning.
505 Future studies could collect a massive amount of unlabeled 16S rRNA sequences and pre-train
506 a model on these data with an autoregression or autoencoder task, like the famous BERT [61]
507 or GPT-2/3/4 [62] models did. The pre-trained model may further improve the accuracy of 16S
508 GCN prediction.
509

510 **5 CONCLUSIONS**

511 In conclusion, this study developed ANNA16, a deep learning-based new bioinformatic
512 tool for predicting the gene copy number of 16S rRNA. The error range of ANNA16 at cross-
513 validation is significantly lower than the error ranges of all taxonomy-based and phylogeny-
514 based algorithms at 16S full-length and is partially lower at all 16S subregions. The
515 performance of ANNA16 in the final test and the case study is also significantly better than all
516 existing tools (i.e., rrnDB, CopyRighter, PICRUSt2, PAPRICA). As a deep learning tool of
517 simple structure, ANNA16 also possesses other advantages such as faster inference speed,
518 independence of pre-computation, and less demand for computing resources. SHAP analysis
519 implies ANNA16 could utilize the information contained in the 16S sequence more effectively
520 by weighing K-mers unequally according to how much information they contain. Future
521 studies may combine the strategy of pre-training model on unlabeled data with the strategies
522 for solving data imbalance to further decrease the error of copy number correction.

LIST OF ABBREVIATIONS

GCN: Gene Gopy Number

MLP: Multi-layer Perceptron

SEM: Stacked Ensemble Model

SHAP: Shapley Additive exPlanation

DECLARATIONS

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Availability of data and materials

16S rRNA sequences and the GCN data analyzed during the current study are available in the rrnDB repository, https://rrndb.umms.med.umich.edu/. The codes of this study are available at https://github.com/Merlinaphist/Anna16. ANNA16 is deployed on Google Colaboratory, https://colab.research.google.com/drive/1XwpTMCHSfTmzpHyKrmiD8aC8C_1nndUV#scrollTo=M0jLLsHC9W4M. The preprint of this study has been released on *bioRxiv* [63]: https://www.biorxiv.org/content/10.1101/2022.11.26.518038v3.

544

545 ***Competing interests***

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

547

548 ***Funding***

549 This work was supported by the Summer Research Scholars (SRS) program at Duke Kunshan

550 University (DKU), the startup funds of Y.L. and M.M. granted by DKU, and the Oracle Cloud

551 Starter Award by the Oracle Corporation.

552

553 ***Authors' contributions***

554 J.M. conceived the study. J.M. and T.C. analyzed the data and implemented the models. Y.L.

555 and M.M. provided mentorship for the study. All authors contributed to the manuscript writing.

556

557 ***Acknowledgments***

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

559

560

561

562

563

564

565 **REFERENCES**

566 1. Woese CR, Fox GE: **Phylogenetic structure of the prokaryotic domain: the primary**
567 **kingdoms**. *Proc Natl Acad Sci U S A* 1977, **74**(11):5088-5090.
568 2. STACKEBRANDT E, GOEBEL BM: **Taxonomic Note: A Place for DNA-DNA**
569 **Reassociation and 16S rRNA Sequence Analysis in the Present Species Definition**
570 **in Bacteriology**. *International Journal of Systematic and Evolutionary Microbiology*
571 1994, **44**(4):846-849.
572 3. Louca S, Doebeli M, Parfrey LW: **Correcting for 16S rRNA gene copy numbers in**
573 **microbiome surveys remains an unsolved problem**. *Microbiome* 2018, **6**(1):41.
574 4. Woo PC, Leung PK, Leung KW, Yuen KY: **Identification by 16S ribosomal RNA**
575 **gene sequencing of an Enterobacteriaceae species from a bone marrow transplant**
576 **recipient**. *Mol Pathol* 2000, **53**(4):211-215.
577 5. Angly FE, Dennis PG, Skarshewski A, Vanwonterghem I, Hugenholtz P, Tyson GW:
578 **CopyRighter: a rapid tool for improving the accuracy of microbial community**
579 **profiles through lineage-specific gene copy number correction**. *Microbiome* 2014,
580 **2**(1):11.
581 6. Stoddard SF, Smith BJ, Hein R, Roller BR, Schmidt TM: **rrnDB: improved tools for**
582 **interpreting rRNA gene abundance in bacteria and archaea and a new foundation**
583 **for future development**. *Nucleic Acids Res* 2015, **43**(Database issue):D593-598.
584 7. Fogel GB, Collins CR, Li J, Brunk CF: **Mini-Review Article: Prokaryotic Genome**
585 **Size and SSU rDNA Copy Number: Estimation of Microbial Relative Abundance**
586 **from a Mixed Population**. *Microbial Ecology* 1999, **38**(2):93-113.
587 8. Kembel SW, Wu M, Eisen JA, Green JL: **Incorporating 16S Gene Copy Number**
588 **Information Improves Estimates of Microbial Diversity and Abundance**. *PLOS*
589 *Computational Biology* 2012, **8**(10):e1002743.
590 9. Jurtz VI, Johansen AR, Nielsen M, Almagro Armenteros JJ, Nielsen H, Sønderby CK,
591 Winther O, Sønderby SK: **An introduction to deep learning on biological sequence**
592 **data: examples and solutions**. *Bioinformatics* 2017, **33**(22):3685-3690.
593 10. Park Y, Kellis M: **Deep learning for regulatory genomics**. *Nature Biotechnology* 2015,
594 **33**(8):825-826.
595 11. Lancashire LJ, Lemetre C, Ball GR: **An introduction to artificial neural networks in**
596 **bioinformatics—application to complex microarray and mass spectrometry**
597 **datasets in cancer studies**. *Briefings in Bioinformatics* 2009, **10**(3):315-329.
598 12. Schmidt B, Hildebrandt A: **Deep learning in next-generation sequencing**. *Drug*
599 *Discov Today* 2021, **26**(1):173-180.
600 13. Stepniewska-Dziubinska MM, Zielenkiewicz P, Siedlecki P: **Development and**
601 **evaluation of a deep learning model for protein–ligand binding affinity prediction**.
602 *Bioinformatics* 2018, **34**(21):3666-3674.
603 14. Fioravanti D, Giarratano Y, Maggio V, Agostinelli C, Chierici M, Jurman G, Furlanello
604 C: **Phylogenetic convolutional neural networks in metagenomics**. *BMC*
605 *Bioinformatics* 2018, **19**(2):49.

15. Lo C, Marculescu R: **MetaNN: accurate classification of host phenotypes from metagenomic data using neural networks**. *BMC Bioinformatics* 2019, **20**(12):314.
16. Wang Y, Bhattacharya T, Jiang Y, Qin X, Wang Y, Liu Y, Saykin AJ, Chen L: **A novel deep learning method for predictive modeling of microbiome data**. *Brief Bioinform* 2021, **22**(3).
17. Zha Y, Chong H, Qiu H, Kang K, Dun Y, Chen Z, Cui X, Ning K: **Ontology-aware deep learning enables ultrafast and interpretable source tracking among sub-million microbial community samples from hundreds of niches**. *Genome Medicine* 2022, **14**(1):43.
18. Arango-Argoty G, Garner E, Pruden A, Heath LS, Vikesland P, Zhang L: **DeepARG: a deep learning approach for predicting antibiotic resistance genes from metagenomic data**. *Microbiome* 2018, **6**(1):23.
19. Ma Y, Guo Z, Xia B, Zhang Y, Liu X, Yu Y, Tang N, Tong X, Wang M, Ye X *et al*: **Identification of antimicrobial peptides from the human gut microbiome using deep learning**. *Nature Biotechnology* 2022, **40**(6):921-931.
20. Chong H, Zha Y, Yu Q, Cheng M, Xiong G, Wang N, Huang X, Huang S, Sun C, Wu S *et al*: **EXPERT: transfer learning-enabled context-aware microbial community classification**. *Briefings in Bioinformatics* 2022:bbac396.
21. Zhao Z, Woloszynek S, Agbavor F, Mell JC, Sokhansanj BA, Rosen GL: **Learning, visualizing and exploring 16S rRNA structure using an attention-based deep neural network**. In: *PLoS computational biology.* vol. 17; 2021: e1009345.
22. Abellan-Schneyder I, Matchado MS, Reitmeier S, Sommer A, Sewald Z, Baumbach J, List M, Neuhaus K: **Primer, Pipelines, Parameters: Issues in 16S rRNA Gene Sequencing**. *mSphere* 2021, **6**(1).
23. Woese CR, Kandler O, Wheelis ML: **Towards a natural system of organisms: proposal for the domains Archaea, Bacteria, and Eucarya**. *Proc Natl Acad Sci U S A* 1990, **87**(12):4576-4579.
24. Camacho C, Coulouris G, Avagyan V, Ma N, Papadopoulos J, Bealer K, Madden TL: **BLAST+: architecture and applications**. *BMC Bioinformatics* 2009, **10**(1):421.
25. Yarza P, Yilmaz P, Pruesse E, Glöckner FO, Ludwig W, Schleifer K-H, Whitman WB, Euzéby J, Amann R, Rosselló-Móra R: **Uniting the classification of cultured and uncultured bacteria and archaea using 16S rRNA gene sequences**. *Nature Reviews Microbiology* 2014, **12**(9):635-645.
26. Ebou A, Koua D, Zeze A: **HyperEx: A Tool to Extract Hypervariable Regions from 16S rRNA Sequencing Data**. *bioRxiv* 2021:2021.2009.2003.455391.
27. Callahan BJ, McMurdie PJ, Rosen MJ, Han AW, Johnson AJ, Holmes SP: **DADA2: High-resolution sample inference from Illumina amplicon data**. *Nat Methods* 2016, **13**(7):581-583.
28. Cole JR, Wang Q, Fish JA, Chai B, McGarrell DM, Sun Y, Brown CT, Porras-Alfaro A, Kuske CR, Tiedje JM: **Ribosomal Database Project: data and tools for high throughput rRNA analysis**. *Nucleic Acids Research* 2013, **42**(D1):D633-D642.

29. DeSantis TZ, Hugenholtz P, Larsen N, Rojas M, Brodie EL, Keller K, Huber T, Dalevi D, Hu P, Andersen GL: **Greengenes, a chimera-checked 16S rRNA gene database and workbench compatible with ARB**. *Appl Environ Microbiol* 2006, **72**(7):5069-5072.
30. Felsenstein J: **Phylogenies and the Comparative Method**. *The American Naturalist* 1985, **125**(1):1-15.
31. Sankoff D: **Minimal Mutation Trees of Sequences**. *SIAM Journal on Applied Mathematics* 1975, **28**(1):35-42.
32. Maddison WP: **Squared-Change Parsimony Reconstructions of Ancestral States for Continuous-Valued Characters on a Phylogenetic Tree**. *Systematic Zoology* 1991, **40**(3):304-314.
33. Louca S, Doebeli M: **Efficient comparative phylogenetics on large trees**. *Bioinformatics* 2017, **34**(6):1053-1055.
34. Katoh K, Misawa K, Kuma Ki, Miyata T: **MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform**. *Nucleic Acids Research* 2002, **30**(14):3059-3066.
35. Price MN, Dehal PS, Arkin AP: **FastTree 2 – Approximately Maximum-Likelihood Trees for Large Alignments**. *PLOS ONE* 2010, **5**(3):e9490.
36. Bowman JS, Ducklow HW: **Microbial Communities Can Be Described by Metabolic Structure: A General Framework and Application to a Seasonally Variable, Depth-Stratified Microbial Community from the Coastal West Antarctic Peninsula**. *PLOS ONE* 2015, **10**(8):e0135868.
37. Douglas GM, Maffei VJ, Zaneveld JR, Yurgel SN, Brown JR, Taylor CM, Huttenhower C, Langille MGI: **PICRUSt2 for prediction of metagenome functions**. *Nature Biotechnology* 2020, **38**(6):685-688.
38. Barbera P, Kozlov AM, Czech L, Morel B, Darriba D, Flouri T, Stamatakis A: **EPA-ng: Massively Parallel Evolutionary Placement of Genetic Sequences**. *Systematic Biology* 2019, **68**(2):365-369.
39. 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**.
40. Lundberg SM, Lee S-I: **A Unified Approach to Interpreting Model Predictions**. In: *Advances in Neural Information Processing Systems: 2017*: Edited by Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R. Curran Associates, Inc. 2017.
41. Wolpert DH: **Stacked generalization**. *Neural Networks* 1992, **5**(2):241-259.
42. Breiman L: **Stacked regressions**. *Machine Learning* 1996, **24**(1):49-64.
43. Cortes C, Vapnik V: **Support-vector networks**. *Machine Learning* 1995, **20**(3):273-297.
44. 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.

45. Cover TM, Hart PE: **Nearest neighbor pattern classification**. *IEEE Trans Inf Theory* 1967, **13**:21-27.
46. Friedman JH: **Greedy function approximation: a gradient boosting machine**. *Annals of statistics* 2001:1189-1232.
47. Hoerl AE, Kennard RW: **Ridge Regression: Biased Estimation for Nonorthogonal Problems**. *Technometrics* 1970, **12**(1):55-67.
48. Tibshirani R: **Regression Shrinkage and Selection via the Lasso**. *Journal of the Royal Statistical Society Series B (Methodological)* 1996, **58**(1):267-288.
49. 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.
50. Loshchilov I, Hutter F: **SGDR: Stochastic Gradient Descent with Restarts**. *CoRR* 2016, **abs/1608.03983**.
51. He K, Zhang X, Ren S, Sun J: **Deep Residual Learning for Image Recognition**. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR): 27-30 June 2016 2016*. 770-778.
52. Moriya S: **Simple mapping-based quantification of a mock microbial community using total RNA-seq data**. *PLoS One* 2021, **16**(7):e0254556.
53. Bray JR, Curtis JT: **An Ordination of the Upland Forest Communities of Southern Wisconsin**. *Ecological Monographs* 1957, **27**(4):325-349.
54. Shapley LS: **A Value for N-Person Games**. Santa Monica, CA: RAND Corporation; 1952.
55. Štrumbelj E, Kononenko I: **Explaining prediction models and individual predictions with feature contributions**. *Knowledge and Information Systems* 2014, **41**(3):647-665.
56. Schneider TD, Stormo GD, Gold L, Ehrenfeucht A: **Information content of binding sites on nucleotide sequences**. *J Mol Biol* 1986, **188**(3):415-431.
57. Schneider TD, Stephens RM: **Sequence logos: a new way to display consensus sequences**. *Nucleic Acids Res* 1990, **18**(20):6097-6100.
58. Quast C, Pruesse E, Yilmaz P, Gerken J, Schweer T, Yarza P, Peplies J, Glöckner FO: **The SILVA ribosomal RNA gene database project: improved data processing and web-based tools**. *Nucleic Acids Research* 2012, **41**(D1):D590-D596.
59. Navarro FCP, Mohsen H, Yan C, Li S, Gu M, Meyerson W, Gerstein M: **Genomics and data science: an application within an umbrella**. *Genome Biology* 2019, **20**(1):109.
60. Zhang Y, Kang B, Hooi B, Yan S, Feng J: **Deep Long-Tailed Learning: A Survey**. *CoRR* 2021, **abs/2110.04596**.
61. Devlin J, Chang M-W, Lee K, Toutanova K: **Bert: Pre-training of deep bidirectional transformers for language understanding**. *arXiv preprint arXiv:1810 04805* 2018.
62. Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A *et al*: **Language Models are Few-Shot Learners**. *CoRR* 2020, **abs/2005.14165**.
63. Miao J, Chen T, Misir M, Lin Y: **Deep Learning for Predicting 16S rRNA Copy Number**. *bioRxiv* 2022:2022.2011.2026.518038.

732 **TABLES**

733 **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

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

737

738

739

740 **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

741

FIGURES

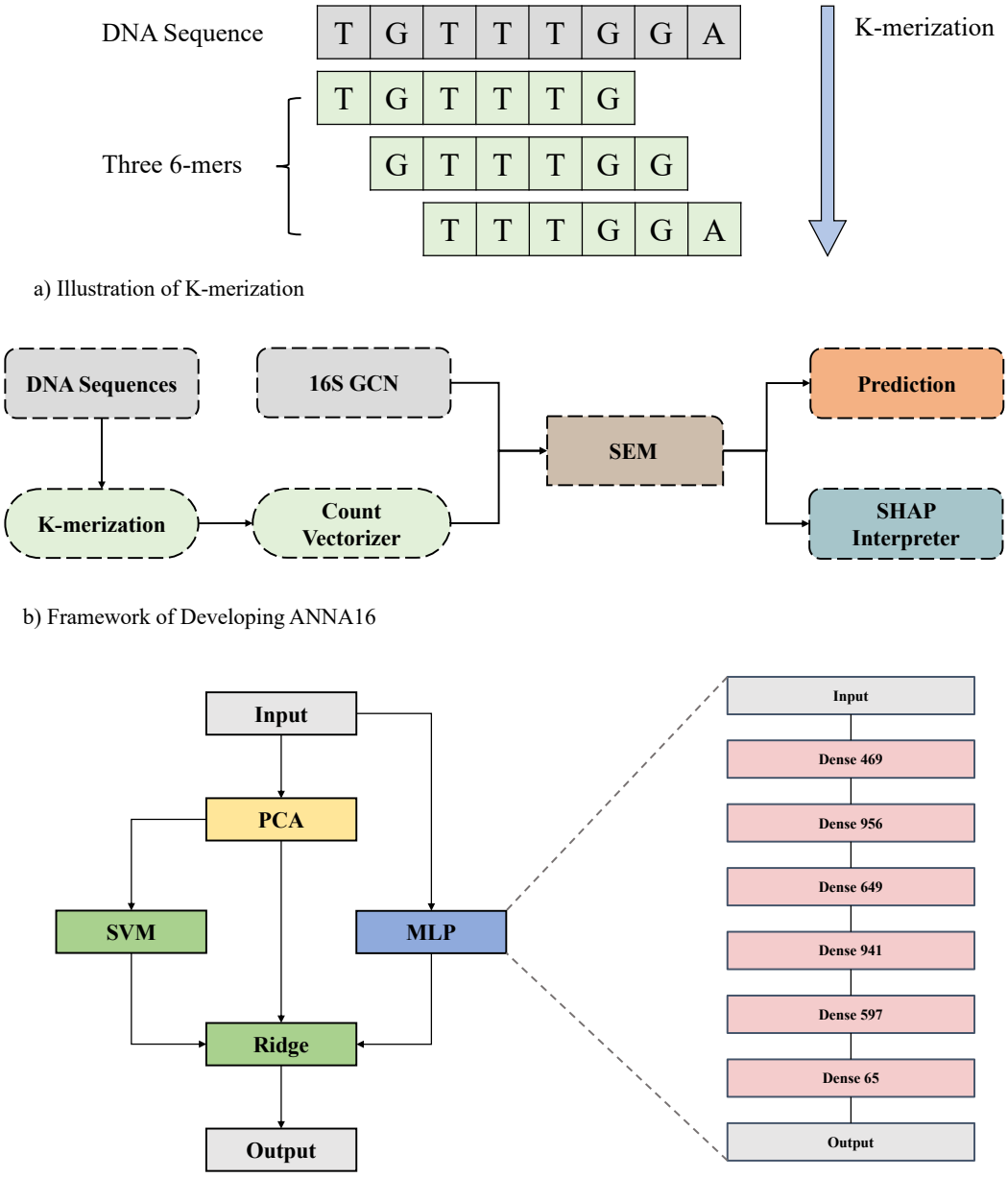

Figure 1. Illustration of the Framework and Architecture of the Deep Learning Models. a) K-mer strategy (K = 6). b) Framework of Designing ANNA16. c) Architecture of the Stacked Ensemble Model. d) Architecture of the MLP model inside the Stacked Ensemble Model.

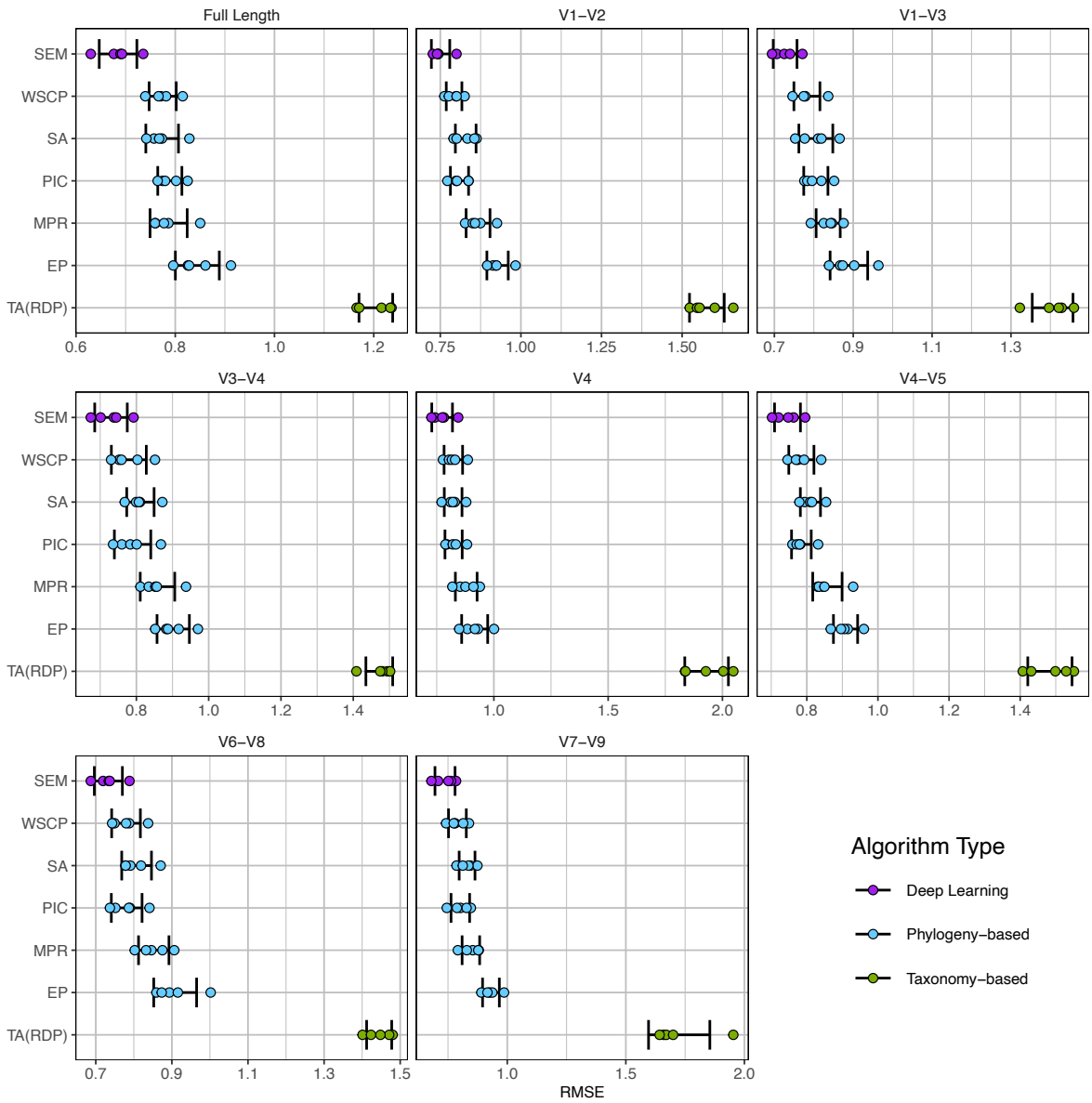

Figure 2. Performance of all algorithms or models at 16S full-length and subregions. EP: empirical probabilities; MPR: Sankoff's maximum-parsimony; PIC: phylogenetic independent contrast; SA: Subtree Averaging; WSCP: weighted-squared-change parsimony; TA(RDP): sub-taxa averaging based on RDP database;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

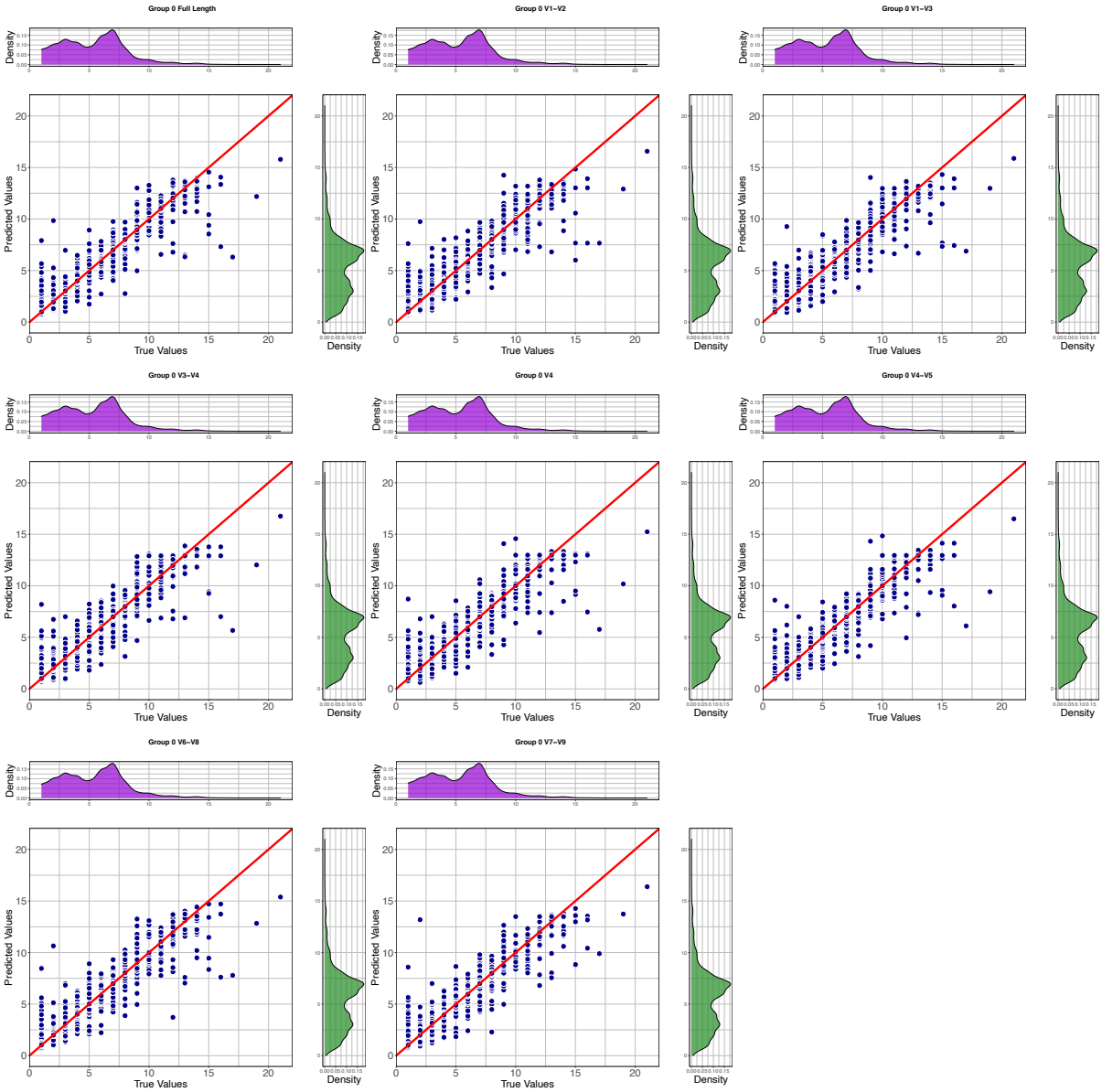

Figure 3. Prediction by SEM on 16S full length and subregions in the first cycle of 5-fold cross-validation. The red line is a diagonal line illustrating to which extent the predicted values are derived from the true values. The density plot filled in violet shows the distribution of the true values,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

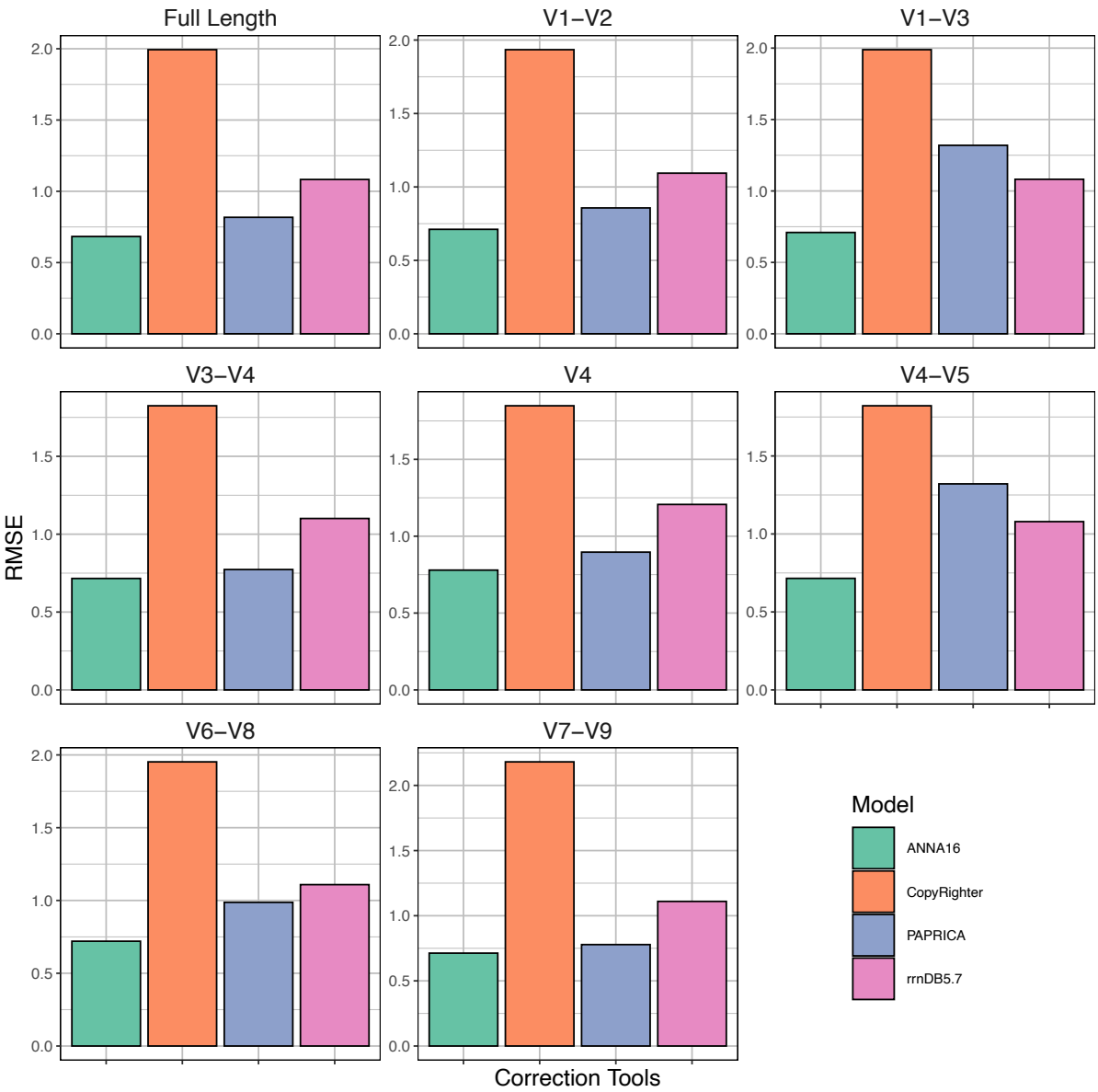

Figure 4. Comparison of performance on the test dataset among CopyRighter, rrnDB, ANNA16, and PAPRICA.

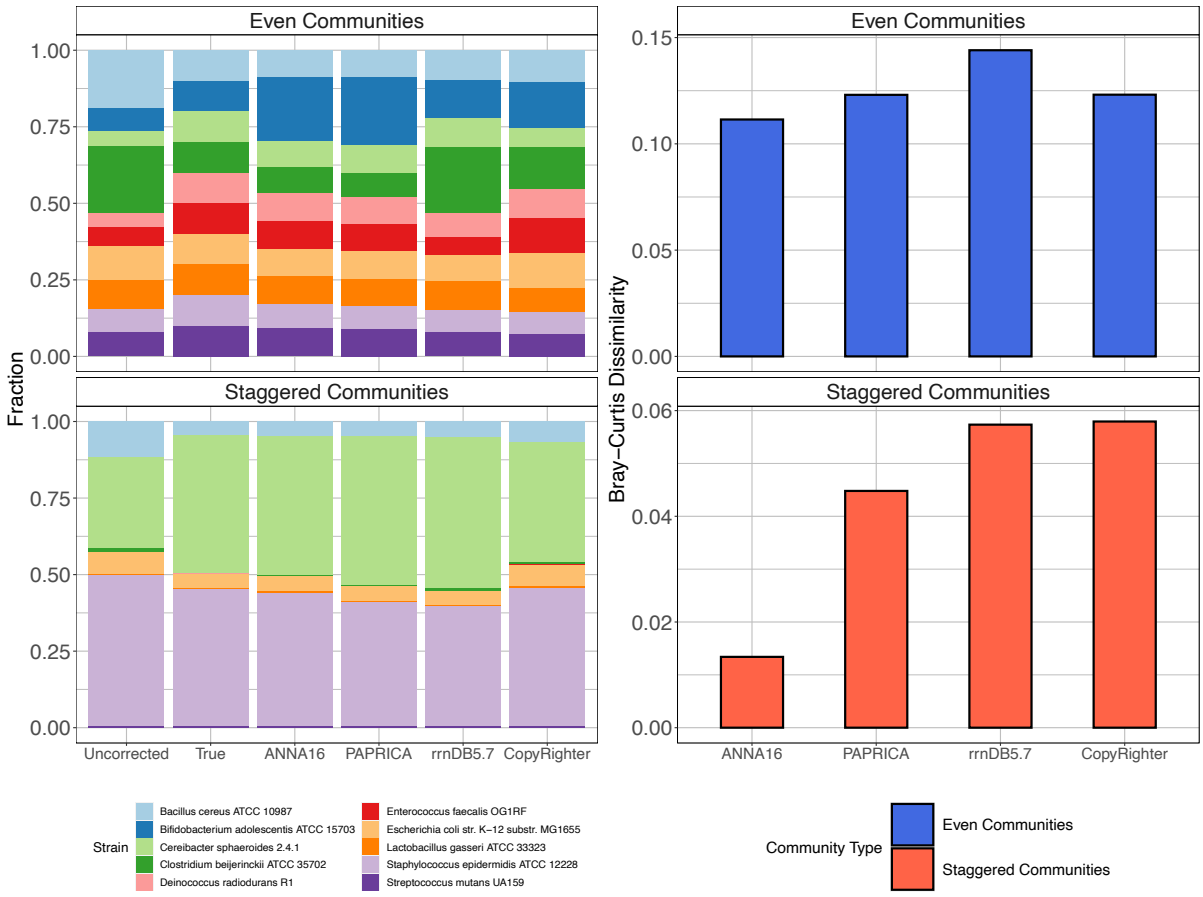

Figure 5. Uncorrected copy fraction, true cell fraction, and cell fraction (microbial profiles) estimated by ANNA16, rrnDB (version 5.7), CopyRighter, and PAPRICA. The right panel shows the value of Bray-Curtis Dissimilarity of each estimated profile to the true cell fraction. Uncorrected: Uncorrected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

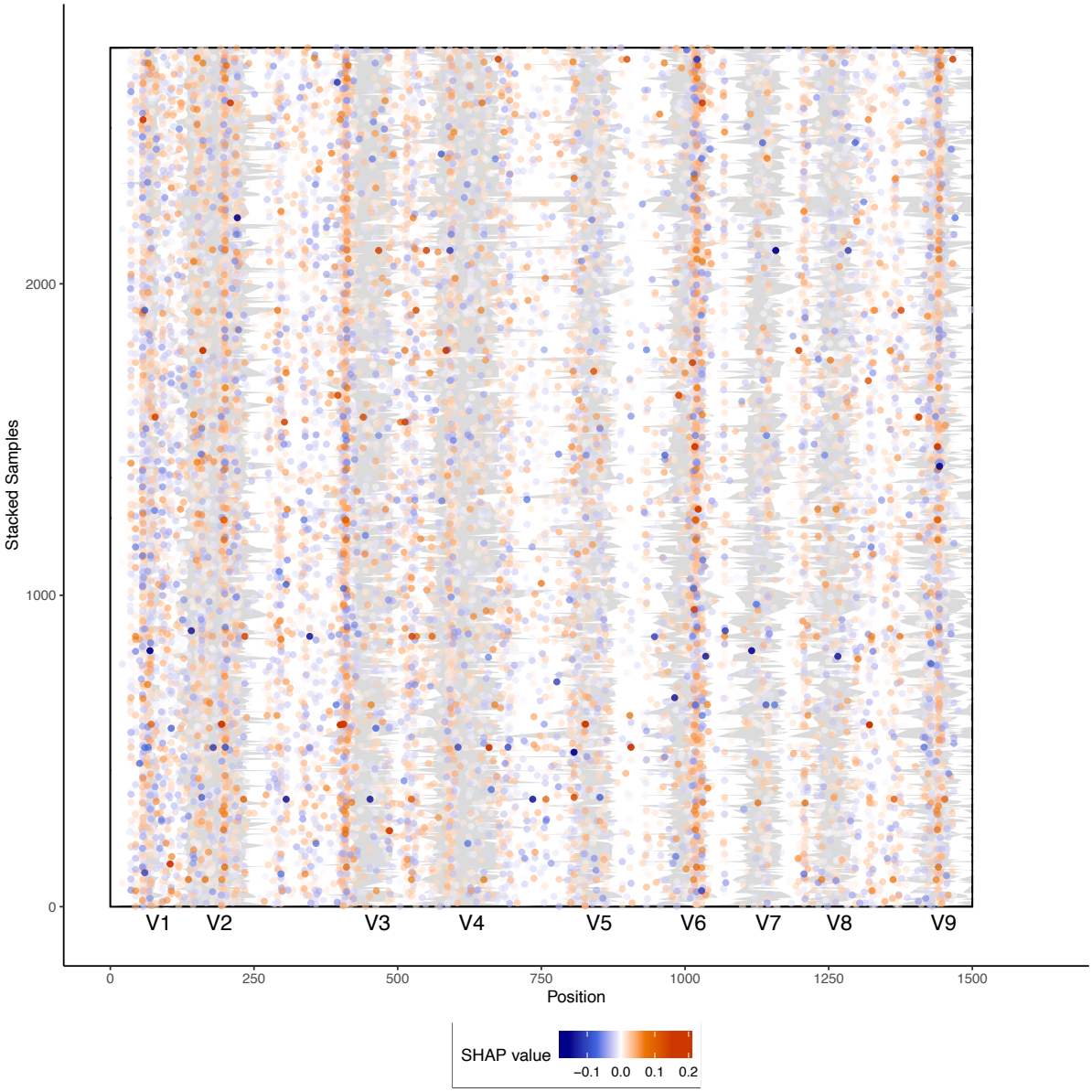

Figure 6. Distribution of SHAP values at 16S full-length sequences. To make the figure interpretable, only the value of the top 100 K-mers ranked by mean absolute SHAP value is shown. Hypervariable regionsare labeled as grey shadows. V-region labels on the x-axis show the general position of *E. coli*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

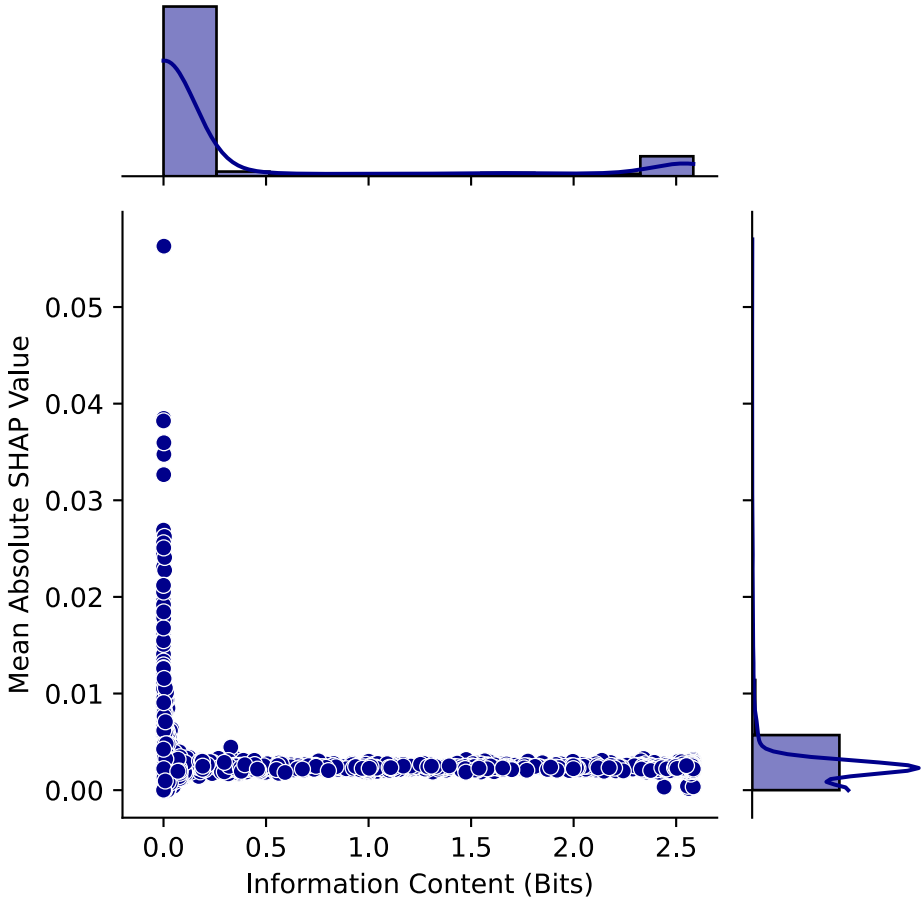

Figure 7. Relationship of mean absolute SHAP values and the information content (unit: bits) on each position of aligned 16S full-length sequence.