Deep Learning for Predicting 16S rRNA Copy Number

Jiazheng Miao[1], Tian-Lai Zang[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

16S rRNA is a commonly used technique to identify prokaryotes. However, this approach could only reflect the proportion of sequencing reads, instead of actual cell fraction. To accurately represent actual cell fraction, bioinformatic tools that estimate 16S gene copy number (GCN) of unmeasured species are required. All the existing tools rely on taxonomy or phylogeny to predict 16S GCN. The present 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. The computational analysis revealed the success of the deep learning model built on 16S GCN data retrieved from rrnDB (N = 19,520). The resulting deep learning model outperforms all the major and state-of-the-art algorithms in the literature. Shapley Additive exPlanations (SHAP) shows ANNA16 is capable of detecting informative positions and weighing K-mers unequally according to their informativeness to more effectively utilize the information contained in 16S sequence. For the accessibility of ANNA16, it will be released as a public, an end-to-end, web-based tool that predicts copy number from 16S rRNA sequence.

1 INTRODUCTION

16S rRNA is the small subunit ribosomal RNA molecules of prokaryotic ribosomes. Since ribosome is a self-replicating system, 16S rRNA is readily isolated and its sequence changes slowly with time (Woese and Fox, 1977). This characteristic allows 16S rRNA to be highly conserved within species (STACKEBRANDT and GOEBEL, 1994) and the variance increases as inter-species relatedness decreases (Woese and Fox, 1977). Therefore, 16S rRNA is used to identity prokaryotes, or detect relatedness among distant species, through matching next-generation sequencing (NGS) data to reference database (Louca et al., 2018) such as RDP and SILVA. The relatedness may be represented as genetic distance among 16S rRNAs like mean nucleotide substitutions per site (Louca et al., 2018). This technique, named 16S rRNA metabarcoding, has been established as the gold standard to infer organismal phylogeny and microbial ecology for over 20 years (Woo et al., 2000).

One critical limitation of 16S rRNA metabarcoding is that NGS actually determines the proportion of 16S rRNA read counts, instead of cell counts. Because it is typical for prokaryotes to have more than one copy of the 16S rRNA gene, the proportion of 16S rRNA read counts (also called relative abundance) could not exactly reflect taxonomic composition of a microbial community (Angly et al., 2014). To represent the profile of a microbial community, the proportion of 16S rRNA read counts needs to be weighted by an inverse of 16S rRNA gene copy number (GCN). This step yields a corrected microbial composition as the community profile, which represents the proportion of cell counts of each lineage in the community (Angly et al., 2014). To calculate 16S GCN, scientists rely on experimental methods such as whole genome sequencing (Stoddard et al., 2015) or competitive PCR (Fogel et al., 1999). Nevertheless, experimental methods are usually costly or culture-dependent, and so far, only a limited number of species have GCN data. This situation catalyzes the development of bioinformatic tools that predict GCN of unmeasured species from measured species. Luckily, 16S GCN is correlated with relatedness across species (Angly et al., 2014, Kembel et al., 2012), which means 16S GCN variation is relatively low within species, but increases as relatedness decreases. This trait allow 16S GCN to be estimated from phylogeny or taxonomy (Kembel et

al., 2012). In fact, all current tools all rely on this kind of method (Angly et al., 2014, Louca et al., 2018, Stoddard et al., 2015). Specifically, one correction tool could construct one taxonomy or phylogenetic tree of the 16S sequence and estimate GCN of the unmeasured species from its close measured relatives. For instance, rrnDB (Stoddard et al., 2015) is one well-established tool which correct 16S GCN based on taxonomy. rrnDB calculates the mean 16S GCN of a taxon from the means of its sub-taxa. CopyRighter (Angly et al., 2014) is one relatively novel tool based on phylogeny. In addition to that, CopyRighter constructs a tree of 16S sequences, and estimate GCN of each ancestral node from its daughter species (Angly et al., 2014), weighted by the phylogenetic distance of each sequenced daughter species through phylogenetic independent contrast (PIC) algorithm (Felsenstein, 1985).

Deep learning as a popular machine learning strategy comes with its potential of learning complex patterns from biological sequencing data (Jurtz et al., 2017, Park and Kellis, 2015). This aspect can be benefited as a novel approach for predicting 16S GCN. Deep learning is essentially an Artificial Neural Network (ANN). The concept of ANN is inspired by the way human brain learns and processes information through synaptic interactions between neurons (Lancashire et al., 2009). An ANN is a computational model that is constituted by multiple interconnected processing elements—artificial neurons (Lancashire et al., 2009). Neurons are utilized in groups as the layers of a network. Each artificial neuron receives and processes input with a non-linear function, known as the activation function (Schmidt and Hildebrandt, 2021). Collectively, one ANN model learns to solve a specific task through capturing non-linear trends and relationships from the presented data. Beyond these capabilities, ANN is even competent to disclose relationships or mechanisms that are still unknown to scientists (Stepniewska-Dziubinska et al., 2018). Deep learning gets its name by having more than one (hidden) layer connecting the input and output units. Having deeper networks refer to representing complex input-output relationships while accommodating large amount of data. In that sense, high-dimensional biological data is a popular target for deep learning.

Recently, a broad spectrum of deep learning aproaches have been applied for microbiology research. Various deep learning architectures like Convolutional Neural Network (CNN), and

90 Recurrent Neural Network (RNN), have been used for metagenomic-based taxonomy
91 classification (Fioravanti et al., 2018), host phenotype predication (Lo and Marculescu, 2019),
92 disease prediction (Wang et al., 2021), and microbial community predication (Zha et al., 2022).
93 Some other works focus on data mining for antibiotic resistance genes (Arango-Argoty et al.,
94 2018), antimicrobial peptides (Ma et al., 2022), microbial source (Chong et al., 2022), and raw
95 sequencing data (Zhao et al., 2021). All these applications show great potential of deep
96 Learning for microbial studies. With these present successes, it should be possible to estimate
97 GCN directly from 16S sequence through an ANN, without the construction of taxonomy or
98 phylogeny. To the best of our knowledge, there is not any deep learning-based method utilized
99 for the 16S rRNA-based copy number estimation.

100 This study undergoes three stages. In Stage 1, this study compares deep learning models
101 to taxonomy-based and phylogeny-based algorithms. Two ANNs are used to predict 16S rRNA
102 GCN from its sequence. One ANN is a classic Multi-layer Perceptron (MLP) while the other
103 one is a more advanced and complex Transformer architecture. These ANNs deliver superior
104 performance leading to the next stage. In Stage 2, a number of methods to make the deep
105 models compatible with different subregions of 16S rRNA, are investigated. Due to drawbacks
106 such as relatively high error rate or higher costs, at present it is less common for microbiologist
107 to sequence the full-length of 16S rRNA (Abellan-Schneyder et al., 2021). Full-length 16S
108 could be more commonly used only if new long sequencing methods are invented. Alternatively,
109 microbial studies tend to target the hypervariable regions in 16S rRNA. 16S rRNA contains
110 nine hypervariable regions, interspersed among conserved regions (V1-V9) (Woese et al.,
111 1990). Mostly, microbiologist sequence only one to three adjacent hypervariable regions by
112 NGS to study the microbial ecology (Abellan-Schneyder et al., 2021). This study focuses on
113 whether deep learning could have consistent performance on different combination of the
114 adjacent hypervariable regions in 16S rRNA. The performance of deep learning models on the
115 subregions are also compared with conventional algorithms to determine if deep learning could
116 outperform the latter ones. In Stage 3, a deep learning-based bioinformatic tool for 16S GCN
117 correction is developed. The tool is compared with existing correction tools to assess the

118 performance. One state-of-the-art method of model explanation is also adopted to uncover how
119 the deep learning model understand a 16S sequence.
120 Based on MLP model, the most accurate method, this study presents ANNA16 (Artificial
121 Neural Network Approximator for 16S rRNA Gene Copy Number), an end-to-end tool that
122 predicts copy number from 16S rRNA sequence.
123

124 # 2 MATERIALS AND METHODS

125 ## 2.1. Data Source and Preprocessing

126 16S rRNA sequences and GCN used to train deep learning models in this study is retrieved
127 from rrnDB, version 5.7 (Stoddard et al., 2015). In total, 20,277 16S genes with unique
128 accession number are included in the dataset, involving 1,443 genera. Average sequence length
129 is 1,537.32 nucleotides, with standard deviation of 31.98 nucleotides. Gene copy number
130 ranges from 1 copy per genome to 21 copies per genome, with mean value of 5.14 and standard
131 deviation of 2.77. This dataset is referred as preprocessed dataset in the following content. In
132 addition, to assess the performance of the deep learning-based correction tool developed in this
133 study, an independent dataset is prepared. This study retrieves 16S GCN data collected
134 independently by (Louca et al., 2018), and removes the overlap in Louca dataset with rrnDB
135 dataset. This yields an independent dataset of 40 sequences. This dataset is referred as
136 independent dataset in the following content.
137 Because raw 16S sequence data contains both (+) strand and (-) strand, bias may be caused
138 if the data is directly used for taxonomy-based and phylogeny-based algorithms. Therefore,
139 this study uses BLASTN algorithm in BLAST+ application, version 2.11.0 (24), with the first
140 50 nucleotides of *E. coli* 16S sequence (+) strand (accession number: GCF 002953035.1) being
141 the subject, to determine the orientation of each 16S sequence. The first 50 nucleotides of E.
142 coli belong to conserved region on 16S rRNA (Yarza et al., 2014), so it could serve as a template.
143 Returned HSPs with sstart > send and evalue < 0.01 are considered as (+) strand, and HSPs
144 with sstart < send and evalue < 0.01 are considered as (-) strand. Remaining sequences are
145 labelled as unrecognizable and dropped from dataset. Reverse complement of (-) strand

146 sequences are generated and added to (+) strands. In addition, 16S sequences in the dataset
147 have unidentical starting positions. To simulate NGS data of PCR amplicon, this study employs
148 HyperEx (Ebou et al., 2021), to extract full length sequence between primer 27F and primer
149 1492R (as listed in Table S1). This ensures all 16S sequences have uniform starting and ending
150 position. This preprocessing step yields a dataset of 19,520 sequences.
151

152 **2.2. Conventional Algorithms and Existing Tools**

153 Sub-taxa averaging (TA) is one taxonomy-based algorithm. It is implemented to compare
154 with deep learning methods. The algorithm calculates the mean 16S GCN of a taxon from the
155 means of its sub-taxa. To implement this algorithm, 16S sequences are first input to
156 *assignTaxonomy* function in DADA2 package in R (version 1.21.0) (Callahan et al., 2016) and
157 yield three taxonomies, based on RDP (version 18, formatted by DADA2) (Cole et al., 2013),
158 SILVA (version 138.1, formatted by DADA2) (Quast et al., 2012), PR2 (version 4.14.0)
159 (Guillou et al., 2012) database respectively. The algorithm first receives taxonomy and GCN
160 from database and converts data structure of the taxonomy to an inheritance tree. In the tree,
161 every taxon is viewed as a node on the tree, and it computes the copy number of every node by
162 taking the mean of its sub-nodes. If an unknown query is given, it assigns the value of the
163 query's parent node to the query node as an estimation. This study also employs six phylogeny-
164 based algorithms to compare with deep learning methods: empirical probabilities (EP), subtree
165 averaging (SA), phylogenetic independent contrasts (PIC) (Felsenstein, 1985), Sankoff's
166 maximum-parsimony (MPR) (Sankoff, 1975), maximum-likelihood of fixed-rates continuous-
167 time Markovmodels (MK) (Pagel, 1994), and weighted-squared-change parsimony (WSCP)
168 (Maddison, 1991). To implement the algorithms, 16S sequences in dataset first undergoes
169 multiple sequence alignment by MAFFT v7.490 (Katoh et al., 2002), and is converted to
170 phylogenetic tree by FastTree 2.1.10 (Price et al., 2010). The algorithms are performed by
171 castor package in R (Louca and Doebeli, 2017), adjusted from the R script in Louca, Doebeli,
172 and Parfrey (2018) (Louca et al., 2018) (MK by function *hsp_mk_model*, MPR by function
173 *hsp_max_parsimony*, WSCP by function *hsp_squared_change_parsimony*, SA by function

174 *hsp_subtree_averaging*, EP by function *hsp_empirical_probabilities*, and PIC by
175 *hsp_independent_contrasts*). All functions are set on default parameters.
176 What is more, after the deep learning-based tool is developed, this study compares it with
177 two existing tools–rrnDB and CopyRighter–by applying the three tools to the independent
178 dataset. Because rrnDB and CopyRighter application will directly output microbial community
179 abundance instead of 16S GCN values, this study retrieves their precomputed tables (rrnDB
180 version 5.7) and predict GCN values by searching the precomputed tables for assigned lineages.
181 Taxonomy of full-length sequences and all subregion sequences are generated by
182 *assignTaxonomy* function in DADA2 package through RDP (version 18) and Greengenes
183 (version 13.8, formatted by DADA2) (DeSantis et al., 2006) database. The assigned RDP
184 lineages are searched in rrnDB precomputed table, and Greengenes lineages are searched in
185 CopyRighter precomputed table.
186

187 **2.3. Development of ANNA16**

188 To build our own deep predictive models, we first implemented Multi-Layer Perceptron
189 and Transformer-based models on full-length DNA sequences in Figure 1 b). Sequences were
190 split into training and testing datasets by 3:1. The DNA sequences were preprocessed with K-
191 merization. After testing K from 3 to 7 on a three-layer MLP prototype model, we found that
192 the model performance would improve as K increased, but the requirement for memory size
193 would also expand sharply. Therefore, the value 6 appears to be the optimal length which
194 results in good performance and do not require extremely large memory size. 6-merization is
195 to decompose the DNA sequence into the set of sub-sequences of length 6. An example of a set
196 with three 6-mers is displayed in Figure 1 a). In MLP path, the K-mers are converted into the
197 number of counts in the set via scikit-learn module (version 0.23.2) CountVectorize function.
198 MLP takes the vector of occurrence counts and predict the gene copy number. In Transformer-
199 based path, K-mers are sorted by occurrence frequency in the set astoundingly and assigned
200 with integer values starting from 0. With a vector of integer values as input, the model then
201 predicts copy number. The performance of the two methods is compared with conventional

algorithms through a 5-fold cross-validation on full length DNA sequence, we selected MLP for further training and predicting on hypervariable regions. Shapley Additive exPlanations (SHAP) (Lundberg and Lee, 2017) is introduced here for post-hoc interpretability. The whole study was conducted with Python 3.8 and Tensorflow 2.8.0 on Nividia RTX 3090.

2.3.1. Multi-layer Perceptron

MLP model is the most common form of ANN, in which artificial neurons are organized into layers and the input is processed layer by layer (Lancashire et al., 2009). The calculation of the output for each neuron is as follows (Lancashire et al., 2009):

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

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

where v_k is the net output of neuron k, with $x_1, x_2, \ldots, x_n$ being the input from the last layer, and $\omega_{k,1}, \omega_{k,2}, \ldots, \omega_{k,n}$ being the weights on the input. γ_k is the output of the neuron, where v_{k_0} represents the bias. In our case, $\varphi(.)$ refers to the rectified linear units (ReLU) activation function. The model was trained with Rooted Mean Squared Error (RMSE) loss. We performed hyper-parameter tuning on the number of dense layers, hidden layer size, and learning rate. Layers of the model were tested in the range of [3,7], hidden layer size was selected from the list of (128, 256, 512, 1024). Learning rate was tested from 0.001 to 0.002, with 0.0002 being a step. The final architecture of MLP for full length DNA sequence is shown in Figure 2 a). It contains a dense block of 6 layers with ReLU.

2.3.2. Transformer Encoder

For transformer-based model in Figure 2 b), we leveraged the original transformer encoder architecture with cosine learning rate decay (Loshchilov and Hutter, 2016), followed by global average pooling, and two dense layers. As shown in the Figure 2 c), the transformer encoder takes embedding and position encoding as input. For biological sequences, a dense embedding is considered more effective than sparse tokenization. Each sequence is thus represented as a matrix $\mathbf{M} \in \mathbb{R}^{n \times d}$, by embedding each 6-mer into a numerical vector of dimension d via

230 embedding layer. Cosine positional encoded method are used to incorporate both absolute and
231 relative positional information by constructing the matrix **P** from **M** (40), where each position
232 has:

233 $$p(2k+1, i) = \sin\left(i/1000^{2k/d}\right)$$

234 $$p(2k+2, i) = \cos\left(i/1000^{2k/d}\right)$$

235 The encoder includes multi-head self-attention, position-wise feed-forward network,
236 residual connection, and layer normalization (Vaswani et al., 2017). With the combination of
237 these modules, the transformer encoder can process contextual information parallelly with
238 attention mechanism. For one single attention head, each token embedding e_i is taken as
239 query q_i to compute *scaled dot attention score* with all token embedding in the same input
240 sequence, i.e., $[e_1, e_2, \dots, e_L]$, as key *k* and value *v*. Formally, the output of one attention head
241 is defined as follows:

242 $$\mathrm{head}(\boldsymbol{q}_i, (\boldsymbol{k}_1, \boldsymbol{v}_1), \dots, (\boldsymbol{k}_L, \boldsymbol{v}_L)) = \sum_{j=1}^{L} \alpha(\boldsymbol{q}_i, \boldsymbol{k}_j)\boldsymbol{v}_j$$

243 $$\alpha(\boldsymbol{q}_i, \boldsymbol{k}_j) = \mathrm{softmax}\left(\frac{\boldsymbol{q}_i{}^T \boldsymbol{k}_j}{\sqrt{d}}\right)$$

244 Thus, for the whole input sequence S, the attention score from multi-head mechanism is

245 $$\mathbf{M} = \mathrm{Multihead}(\mathrm{S}) = \mathrm{Concat}(\mathrm{head}_1, \mathrm{head}_2, \dots, \mathrm{head}_n)\mathbf{W}$$

246 where all **W** parameters are learnable for linear transformation. By learning different linear
247 projection independently, it captures the combination of different representation subspaces.
248 The intermediate representation is passed into feed-forward network for further non-linear
249 transformation with ReLU activation. The Layer Normalization is used here to avoid gradient
250 explosion and accelerate convergence, and Dropout method to prevent over-fitting.

251

252 2.3.3. Test of Deep Learning on Hypervariable Regions

253 According to Abellan-Schneyder et al. (2021) (Abellan-Schneyder et al., 2021), V1-V2,
254 V1-V3, V3-V4, V4, V4-V5, V6-V8, and V7-V9 are frequently targeted. This study uses
255 primers listed in Table S1 and HyperEx to extract these adjacent hypervariable regions from
256 preprocessed full-length dataset. To make deep learning model compatible with hypervariable

257 regions, this study test three methods to train the model. Method I extracts all nine
258 hypervariable regions from preprocessed full-length dataset separately with HyperEx based on
259 conserved markers proposed by Yang, Wang, and Qian (Yang et al., 2016), as listed in Table
260 S2. The markers were used as forward primers, and their reverse complements were used as
261 reverse primers. Extracted subregion sequences were added back to the dataset. This in fact
262 augments the dataset by 10 times. A new model on was trained on the augmented dataset.
263 Method II augments the preprocessed full-length dataset by non-overlap splitting each
264 sequence to fragments of 101 nucleotides. A new model will be trained on the augmented
265 dataset. Method III trains the model separately on the adjacent hypervariable regions. This will
266 produce customize models for each region. Hyperparameters of customized architecture is
267 shown in Table S4.
268
269 2.3.4. Case Study and Explainability
270 Based on the MLP method, this study presents ANNA16 (Artificial Neural Network
271 Approximator for 16S rRNA Gene Copy Number), an end-to-end tool that predicts copy
272 number from 16S rRNA sequence. In addition to independent dataset test, this study designs a
273 case study which simulates two mock communities in silico to demonstrate ANNA16's
274 performance. rrnDB and CopyRighter are also applied to estimate the mock communities'
275 microbial profile as a positive control. The simulated communities are two commercially
276 available mock communities, one even community (ATCC MSA-2003) and one staggered
277 community (ATCC MSA-1001). Such mock communities have been used by Moriya (2021) to
278 study methods of quantifying a microbial community (Moriya, 2021). Both mock communities
279 consist of similar 10 bacterial strains, but with different ratios. This study retrieves genome
280 sequences of the 10 strains from NCBI, copy number data from rrnDB (version 5.7), and
281 assigns a cell counts value to each strain by assuming each community have 1,000,000 cells.
282 Table S3 shows basic information of the mock communities. After genome sequences of the
283 ten strains are retrieved from NCBI, HyperEx is used to extract 16S full length sequence from

284 the genomes. The method is the same as described in the Data Source and Preprocessing and
285 Test of Deep Learning on Hypervariable Regions part.

286 SHAP (Lundberg and Lee, 2017) is used to further explain our model prediction. SHAP is
287 based on both collational game theory (Shapley, 1952) and the classical Shapley method
288 (Štrumbelj and Kononenko, 2014). It is a unified measure of feature importance that determines
289 the individual contribution of each feature to model prediction by viewing each feature as a
290 player in Shapley. The marginal contribution can be estimated under multiple feature subsets,
291 and the final contribution is the sum of all marginal contributions. A positive SHAP value
292 denotes that the feature contributes positively to the model's prediction over the model
293 expectation, whereas the negative value denotes that the feature contributes negatively. In this
294 study, each individual K-mer is considered as a feature. To calculate SHAP value for each K-
295 mer, the preprocessed dataset and ANNA16's pre-trained neural network are input to
296 shap.KernelExplainer function in shap module (Version 0.41.0) in Python. The function views
297 each prediction of every single sequence in the dataset as an independent game and calculates
298 SHAP values in each game. As a result, a list of 19,520 SHAP values will be assigned to every
299 K-mer to represent the K-mer's overall contribution. After SHAP values of K-mers are
300 calculated, the SHAP values are mapped back to 16S sequence to see if there is any positional
301 pattern in contribution. Each sequence is first transformed into an array of K-mers, and next
302 each K-mer is replaced by its corresponding SHAP value. As a result, the contribution of each
303 position in a sequence is represented as an array of SHAP values. What is more, to study the
304 relationship between contribution and nucleotide conservation in a position, information
305 content of each position is calculated. Information content is a measurement of the average
306 information available in a position, and it represents how conserved a position is if applied to
307 analyzed nucleotides (Schneider et al., 1986). This measurement has been widely used in
308 genetics, and one well-known application is to serve as the height of characters in sequence
309 logos (Schneider and Stephens, 1990). Previously aligned 16S sequences during
310 implementation of phylogeny-based algorithms are used to calculate information content. The
311 calculation is as follows:

312 $$H_i = -\sum f_{char,i} \times \log_2(f_{char,i}), char \in [A, T, C, G, -, N]$$

313 $$R'_i = \log_2(6) - H_i - (\frac{1}{\ln 2} \times \frac{s-1}{2n})$$

314 $$R_i = R'_i \times \sum f_{nt,i}, nt \in [A, T, C, G]$$

315 where H_i is the uncertainty of position i, f is the frequency of a character, − represents gaps
316 generated in multiple sequence alignment, N = ambiguous nucleotides, $R_i{}'$ is information
317 content of all characters in position i, s is the number of characters (which is 6 in this
318 calculation), n is the number of sequences in alignment (which is 19520), R_i is the information
319 content of the four nucleotides, and nt represents nucleotides. R_i is used to represent degree of
320 conservation of the position i.

321

322 **3 RESULTS**

323 The first stage of this study tests two deep learning models against all other potential
324 algorithms for the task of 16S GCN correction. After tuning, the optimized hyperparameters of
325 MLP model are demonstrated in Table S4. MLP of full length comprises 7 layers, with 512
326 neurons on input layer and 1 on output layer. The model is trained with a batch size of 100, a
327 learning rate of 0.001 for 50 epoches. All algorithms implemented in this study undergo one 5-
328 fold cross-validation in Stage 1. The performance is measured in rooted mean squared error
329 (RMSE) as follows:

330 $$\text{RMSE} = \sqrt{\frac{\sum_{i=0}^{n}(Y_{true} - Y_{pred})^2}{n}}$$

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

333 Figure 3 and Table 1 summarize the results of 5-fold cross-validation. Sub-taxa averaging
334 based on PR2, RDP, and SILVA exhibit a mean RMSE of 1.26, 1.18, 1.20 copies/genome
335 respectively, with standard deviation of RMSE being 0.0597, 0.0356, and 0.0377. RDP proves
336 to be most accurate database for this task which leads to the lowest error range. Except that
337 MK shows particularly high RMSE (Mean = 5.76, SD = 0.112), all other phylogeny-based

338 algorithms perform better than sub-taxa averaging (EP: Mean = 0.845, SD = 0.0443; MPR:
339 Mean = 0.787, SD = 0.0375; PIC: Mean = 0.789, SD = 0.0243; SA: Mean = 0.774, SD = 0.0329;
340 WSCP: Mean = 0.775, SD = 0.0273). As shown in Figure 3, error range of all phylogeny-based
341 algorithms, except MK, is lower than those of taxonomy-based algorithms with no overlap.
342 With low rmse and small standard deviation, WSCP turns out to be the optimal phylogeny-
343 based algorithm. As for deep learning models, Transformer achieves a Mean RMSE of 0.861
344 copies/genome (SD = 0.0643). The error range is close to that of EP. MLP outperforms all other
345 algorithms with the lowest Mean RMSE (0.729 copies/genome) and a relatively small standard
346 deviation (0.0388). Its error range partially overlaps with WSCP, SA, PIC, MPR, and is the
347 lowest among all algorithms.

348 EP, MPR, PIC, SA, WSCP are selected as references in Stage 2 due to their relatively
349 superior performance. MLP is chosen between the deep learning models, and Method I, II, III
350 for subregions are applied to MLP. Table S5 shows the performance of Method I, II and III. It
351 can be seen that Method I and II performs significantly worse than Method III. Method III
352 achieves a maximum Mean RMSE of 0.8177 copies/genome at V4 region. However, the
353 minimum Mean RMSE of Method I and II is 1.034 and 1.257 copies/genome at V4 and V1-
354 V3 region respectively. Mean RMSE of Method I can even be as high as 4.272 copies/genome
355 at V6-V8 region, and Method II shows a maximum Mean RMSE of 2.239 copies/genome at
356 V7-V9 region. Therefore, Method I and II are abandoned and only the results of Method III are
357 compared with the selected phylogeny-based algorithms. Figure 4 illustrates prediction
358 produced by Method III in the first cycle of cross-validation. It can be seen that predicted values
359 possess a good linear relationship with true values. Predicted values and true values also have
360 similar patterns in data distribution. But one issue is that the models tend to make
361 underestimated prediction when true value > 12. This will be further discussed in the discussion
362 section. Figure 5 and Table 2 shows the performance of EP, MPR, PIC, SA, WSCP and MLP-
363 R (MLP by region, which is MLP on full length and MLP trained in Method III). Compared
364 with performance on 16 full length, all algorithms produce an increased Mean RMSE on
365 subregions. EP experiences the most significant increase, with Mean RMSE rising up to 0.9312

366 copies/genome on V7-V9 region. WSCP is the most stable phylogeny-based algorithm, with
367 Mean RMSE on subregions ranges from 0.7792 (V3-V4) to 0.8227 (V4) copies/genome. Mean
368 RMSE of MLP on subregions is relatively consistent, ranging from 0.7751 (V1-V3) to 0.8127
369 (V4) copies/genome. Notably, as shown in Figure 5, the error range of MLP on subregions is
370 lower than EP and MPR in almost all subregions, and partially lower than SA in V1-V2, V4-
371 V5 and V7-V9, but comparable to PIC or WSCP. This indicates the performance of MLP on
372 subregions is consistent with that on full length. Compared with phylogeny-based algorithms,
373 the performance is better than EP and MPR but indistinguishable with PIC and WSCP.

374 Figure 6, demonstrate the results of independent dataset test in Stage 3. Figure 6 shows
375 ANNA16 achieves the lowest RMSE among the three tools on full length and every subregion.
376 The smallest RMSE of MLP appears on full length, which is 0.7194 copies/genome, and the
377 highest RMSE is at V6-V8 region, being 0.8780 copies/genome. In contrast, the RMSE of
378 CopyRighter ranges from 1.382 (V1-V2) to 1.770 (V4) copies/genome, and rrnDB have a
379 stable RMSE between 1.10 and 1.20 copies/genome. Figure 7, Table S6, and Table S7 shows
380 uncorrected copy fraction, true cell fraction, and cell fraction (microbial profiles) estimated by
381 ANNA16, rrnDB (version 5.7), and CopyRighter. The different in uncorrected copy fraction
382 and true cell correction proves GCN correction is a necessary procedure if researchers aim to
383 present the accurate microbial profile. Cell fraction estimated by ANNA16 has most similar
384 patterns with true cell fraction compared to the counterpart of rrnDB and CopyRighter.

385 To illustrate how ANNA16 makes prediction, shap module in Python is used calculate
386 SHAP value of each K-mer. Figure S1 shows top 20 K-mers ranked by mean absolute SHAP
387 value. Figure 8 shows the distribution of SHAP values on the sequences and indicates that
388 ANNA16 does not simply extract information from hypervariable regions–conserved regions
389 also provide information. To further illustrate the relationship between SHAP value and
390 information content (degree of conservation), Figure 9 is drawn. Figure 9 plots mean absolute
391 SHAP value on each position versus degree of conservation of each position. Interestingly,
392 mean absolute SHAP value reaches the peak when degree of conservation is extremely low.
393 This will be further discussed in Discussion Section.

394

395 # 4 DISCUSSION

396 In summary, this study aims to use deep learning to predict 16S GCN based on 16S
397 sequence and consists of two stages: In Stage 1, two deep learning models, MLP and
398 transformer, are implemented and compared with conventional algorithms on 16S full length;
399 in Stage 2, sequences of different 16S subregions are generated to test if deep learning could
400 achieve a consistent performance on subregions; finally in Stage 3, ANNA16–a bioinformatic
401 tool based on MLP–is subsequently developed and compared to two existing 16S GCN
402 correction tools, rrnDB and CopyRighter, through a case study. SHAP method is used to
403 explain how MLP finishes the task.

404 First, the results in Stage 1 indicates MLP outperforms all other algorithms on 16S full
405 length. As shown in Figure 3, MLP exhibits an error range which is far lower than the error
406 ranges of sub-taxa averaging algorithms and considerably lower than phylogeny-based
407 algorithms. This indicates MLP is capable of more accurately correcting GCN than existing
408 algorithms if applied on 16S full length. Moreover, it is worth noting that the performance of
409 Transformer is unexpectedly inferior to that of MLP. As the computing cost of the self-attention
410 mechanism increases quadratically with sequence length, Transformer-based models are
411 typically incapable of modeling extended sequences (Beltagy et al., 2020). As a reference, in
412 natural language processing, classical models normally take inputs below 512 or 1024 (Devlin
413 et al., 2018). Given biological sequences are more complex than natural language and our input
414 length exceeds one thousand, it's plausible that Transformer-based model gives worse
415 performance. As MLP has simpler structure, faster inference speed, less computing
416 consumption, and satisfactory performance, we decided to use MLP for our final model and
417 other downstream tasks.

418 Second, Stage 2 proves ANNA16, as a state-of-the-art GCN correction tool, is more
419 accurate than rrnDB and CopyRighter. On the one hand, the advantage of MLP over rrnDB, is
420 clearly illustrated through cross-validation (Figure 3) or independent dataset test (Figure 6). It
421 could be safely concluded that ANNA16 turns out to be a more accurate than rrnDB. On the

422 other, though Figure 5 and Table 2 suggest MLP has a comparable performance to PIC (the
423 algorithm of CopyRighter), independent dataset test gives a different indication. As can be seen
424 in Figure 6, RMSE of CopyRighter is significantly higher than that of ANNA16 on all regions.
425 This suggests CopyRighter is actually less accurate than ANNA16. If this is the case, the
426 microbial profile of the two mock communities estimated by ANNA16 should be closer to true
427 profile than the counterpart presented by CopyRighter. Figure 7, Table S6, and Table S7
428 confirms this indication, where both true profiles are more similar to those presented by
429 ANNA16, especially for the staggered community. Independent dataset test and case study
430 collectively suggest ANNA16 is more accurate in 16S GCN prediction than CopyRighter. This
431 inconsistency in PIC and CopyRighter performance could be a result of CopyRighter's old
432 database. According to CopyRighter's GitHub Page, by the time this manuscript is written,
433 CopyRighter has not got any update for 9 years. This means the database of CopyRighter is
434 relatively small and out-of-date. If developers of CopyRighter could update the pre-computed
435 table of CopyRighter using enlarged 16S GCN dataset, its performance would probably
436 experience a promising improvement. But for the time being, ANNA16 turns out to be the most
437 accurate 16S GCN correction tool. In addition to its superior performance, another advantage
438 of ANNA16 is that it is taxonomy-independent. Current 16S GCN tools are dependent on pre-
439 computation and are bound to one specific ribosomoal database: rrnDB and CopyRighter pre-
440 compute the GCN value for every taxon in RDP database or Greengene database respective (4,
441 5). When a user submits a correction request, a pre-computed tool actually searches the input
442 lineages from the pre-computed GCN table, instead of computing the value from scratch. This
443 strategy means a pre-computed correction tool will be bound to its database. Due to the
444 existence of differences in nomenclature among databases (for example, SILVA merges
445 *Allorhizobium*, *Neorhizobium*, *Pararhizobium*, *Rhizobium* as one genus, named *Allorhizobium-*
446 *Neorhizobium-Pararhizobium-Rhizobium*, but in RDP these are separated genera), increased
447 error might be caused if a pre-computed tool is applied to a taxonomy generated by a
448 miscellaneous database. Table S8 shows the results when RDP taxonomy of preprocessed
449 dataset is applied to SILVA taxonomic tree, and vice versa. The RMSE will increase by 5.30%

450 and 4.00% respectively. On the contrary, the deep learning method implemented in this study
451 is free of the problem. Users could predict GCN of each 16S sequence and assign taxonomy
452 separately. This feature allows the deep learning method to cooperate with any 16S database in
453 microbial community composition analysis. After proving the supremacy of ANNA16, it
454 become an interesting question–how the model perceives the complex relationship between K-
455 mer counts and 16S GCN. Conventionally, microbiologists believe the nine hypervariable
456 regions are the most informative regions on 16S sequence that could reflect the phylogeny of
457 one species. Therefore, one hypothesis is that K-mers from hypervariable regions would
458 contribution more to prediction than those from conserved regions. However, Figure 8 shows
459 that distribution of SHAP values does not completely fit in the location of hypervariable regions.
460 Though high SHAP values indeed tend to aggregate in V2, V4, V6, and V8 region, conserved
461 regions between V2-V3, V4-V5, V8-V9 also contribute considerably. Since Figure 8 alone is
462 not evident enough to accept or reject the hypothesis, Figure 9 is drawn. By plotting
463 information content against mean absolute SHAP values on each position, we expect to see a
464 linear relationship. However, the scatters are actually L-shaped, with high SHAP value only
465 appearing on positions with extremely low information content. SHAP values is generally
466 constant on other positions. This trend indicates extremely variable positions or positions with
467 insertion of extremely low frequency are most informative for ANNA16. The other positions
468 provide nearly same level of information, no matter how conserved they are. Deducing from
469 this indication, this means ANNA16 is able to detect informative positions and assigning high
470 weights to the positions. This may help explain why ANNA16 or MLP has better performance
471 than conventional algorithms. When implementing conventional algorithms, variance in
472 sequence is converted to sequence similarity, and the sequence similarity is to generate
473 phylogeny or taxonomy. This approach in fact weighs every position equally. Therefore, it is
474 possible for informative positions to be underrepresented and non-informative region
475 overrepresented in conventional algorithms. However, by weighing K-mers unequally
476 according to their informativeness, ANNA16 could overcome this problem and utilize the
477 information contained in 16S sequence more effectively. This finding reveals the great potential

of deep learning in genetics. On the one hand, deep learning could accurately predict a trait based on sequencing data. On the other hand, by investigating contribution of each feature, scientists could discover which genes or loci are most related with the trait.

One limitation of ANNA16 is underestimation of species with high GCN. As can be seen in Figure 4, ANNA16 would make underestimated prediction when true GCN $\geq$ 14 and those predicted values can be far from true values. This limitation could be attributed to unbalance in the dataset. Figure S2 shows the distribution of GCN in the preprocessed dataset. Sequences with GCN $\geq$ 12 accounts for only a small proportion of the whole dataset, and sequences with GCN > 15 is extremely rare. This makes it difficult for the model to learn features of high GCN. The most effective method to solve this problem is to wait for increase of data. Thanks to invention of new sequencing techniques, nowadays genomic data is accumulating at an exponential rate (Navarro et al., 2019). If more data about high GCN species could be collected, the underestimation would be correspondingly relieved.

CONFLICT OF INTEREST STATEMENT

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

AUTHOR 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 manuscript writing.

FUNDING

This work was supported by Duke Kunshan University (DKU) Summer Research Scholars program and startup funds to Y.L. and M.M..

ACKNOWLEDGMENTS

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

506

507 **DATA AVAILABILITY STATEMENT**

508 16S rRNA sequence and GCN data analyzed during the current study are available in the rrnDB

509 repository (https://rrndb.umms.med.umich.edu/).

510 Scripts needed to implement ANNA16 are available on

511 https://github.com/Merlinaphist/Anna16. We will soon release a notebook on Colab.

512

REFERENCES

ABELLAN-SCHNEYDER, I., MATCHADO, M. S., REITMEIER, S., SOMMER, A., SEWALD, Z., BAUMBACH, J., LIST, M. & NEUHAUS, K. 2021. Primer, Pipelines, Parameters: Issues in 16S rRNA Gene Sequencing. *mSphere,* 6.

ANGLY, F. E., DENNIS, P. G., SKARSHEWSKI, A., VANWONTERGHEM, I., HUGENHOLTZ, P. & TYSON, G. W. 2014. CopyRighter: a rapid tool for improving the accuracy of microbial community profiles through lineage-specific gene copy number correction. *Microbiome,* 2**,** 11.

ARANGO-ARGOTY, G., GARNER, E., PRUDEN, A., HEATH, L. S., VIKESLAND, P. & ZHANG, L. 2018. DeepARG: a deep learning approach for predicting antibiotic resistance genes from metagenomic data. *Microbiome,* 6**,** 23.

BELTAGY, I., PETERS, M. E. & COHAN, A. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004. 05150*.

CALLAHAN, B. J., MCMURDIE, P. J., ROSEN, M. J., HAN, A. W., JOHNSON, A. J. & HOLMES, S. P. 2016. DADA2: High-resolution sample inference from Illumina amplicon data. *Nat Methods,* 13**,** 581-3.

CHONG, H., ZHA, Y., YU, Q., CHENG, M., XIONG, G., WANG, N., HUANG, X., HUANG, S., SUN, C., WU, S., CHEN, W.-H., COELHO, L. P. & NING, K. 2022. EXPERT: transfer learning-enabled context-aware microbial community classification. *Briefings in Bioinformatics***,** bbac396.

COLE, J. R., WANG, Q., FISH, J. A., CHAI, B., MCGARRELL, D. M., SUN, Y., BROWN, C. T., PORRAS-ALFARO, A., KUSKE, C. R. & TIEDJE, J. M. 2013. Ribosomal Database Project: data and tools for high throughput rRNA analysis. *Nucleic Acids Research,* 42**,** D633-D642.

DESANTIS, T. Z., HUGENHOLTZ, P., LARSEN, N., ROJAS, M., BRODIE, E. L., KELLER, K., HUBER, T., DALEVI, D., HU, P. & ANDERSEN, G. L. 2006. Greengenes, a chimera-checked 16S rRNA gene database and workbench compatible with ARB. *Appl Environ Microbiol,* 72**,** 5069-72.

DEVLIN, J., CHANG, M.-W., LEE, K. & TOUTANOVA, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810. 04805*.

EBOU, A., KOUA, D. & ZEZE, A. 2021. HyperEx: A Tool to Extract Hypervariable Regions from 16S rRNA Sequencing Data. *bioRxiv***,** 2021.09.03.455391.

FELSENSTEIN, J. 1985. Phylogenies and the Comparative Method. *The American Naturalist,* 125**,** 1-15.

FIORAVANTI, D., GIARRATANO, Y., MAGGIO, V., AGOSTINELLI, C., CHIERICI, M., JURMAN, G. & FURLANELLO, C. 2018. Phylogenetic convolutional neural networks in metagenomics. *BMC Bioinformatics,* 19**,** 49.

FOGEL, G. B., COLLINS, C. R., LI, J. & BRUNK, C. F. 1999. Mini-Review Article: Prokaryotic Genome Size and SSU rDNA Copy Number: Estimation of Microbial Relative Abundance from a Mixed Population. *Microbial Ecology,* 38**,** 93-113.

554 GUILLOU, L., BACHAR, D., AUDIC, S., BASS, D., BERNEY, C., BITTNER, L., BOUTTE,
555 C., BURGAUD, G., DE VARGAS, C., DECELLE, J., DEL CAMPO, J., DOLAN, J.
556 R., DUNTHORN, M., EDVARDSEN, B., HOLZMANN, M., KOOISTRA, W. H. C.
557 F., LARA, E., LE BESCOT, N., LOGARES, R., MAHÉ, F., MASSANA, R.,
558 MONTRESOR, M., MORARD, R., NOT, F., PAWLOWSKI, J., PROBERT, I.,
559 SAUVADET, A.-L., SIANO, R., STOECK, T., VAULOT, D., ZIMMERMANN, P. &
560 CHRISTEN, R. 2012. The Protist Ribosomal Reference database (PR2): a catalog of
561 unicellular eukaryote Small Sub-Unit rRNA sequences with curated taxonomy. *Nucleic*
562 *Acids Research,* 41**,** D597-D604.
563 JURTZ, V. I., JOHANSEN, A. R., NIELSEN, M., ALMAGRO ARMENTEROS, J. J.,
564 NIELSEN, H., SØNDERBY, C. K., WINTHER, O. & SØNDERBY, S. K. 2017. An
565 introduction to deep learning on biological sequence data: examples and solutions.
566 *Bioinformatics,* 33**,** 3685-3690.
567 KATOH, K., MISAWA, K., KUMA, K. I. & MIYATA, T. 2002. MAFFT: a novel method for
568 rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids*
569 *Research,* 30**,** 3059-3066.
570 KEMBEL, S. W., WU, M., EISEN, J. A. & GREEN, J. L. 2012. Incorporating 16S Gene Copy
571 Number Information Improves Estimates of Microbial Diversity and Abundance. *PLOS*
572 *Computational Biology,* 8**,** e1002743.
573 LANCASHIRE, L. J., LEMETRE, C. & BALL, G. R. 2009. An introduction to artificial neural
574 networks in bioinformatics—application to complex microarray and mass spectrometry
575 datasets in cancer studies. *Briefings in Bioinformatics,* 10**,** 315-329.
576 LO, C. & MARCULESCU, R. 2019. MetaNN: accurate classification of host phenotypes from
577 metagenomic data using neural networks. *BMC Bioinformatics,* 20**,** 314.
578 LOSHCHILOV, I. & HUTTER, F. 2016. SGDR: Stochastic Gradient Descent with Restarts.
579 *CoRR,* abs/1608.03983.
580 LOUCA, S. & DOEBELI, M. 2017. Efficient comparative phylogenetics on large trees.
581 *Bioinformatics,* 34**,** 1053-1055.
582 LOUCA, S., DOEBELI, M. & PARFREY, L. W. 2018. Correcting for 16S rRNA gene copy
583 numbers in microbiome surveys remains an unsolved problem. *Microbiome,* 6**,** 41.
584 LUNDBERG, S. M. & LEE, S.-I. 2017. A Unified Approach to Interpreting Model Predictions.
585 *In:* GUYON, I., LUXBURG, U. V., BENGIO, S., WALLACH, H., FERGUS, R.,
586 VISHWANATHAN, S. & GARNETT, R. (eds.) *Advances in Neural Information*
587 *Processing Systems.* Curran Associates, Inc.
588 MA, Y., GUO, Z., XIA, B., ZHANG, Y., LIU, X., YU, Y., TANG, N., TONG, X., WANG, M.,
589 YE, X., FENG, J., CHEN, Y. & WANG, J. 2022. Identification of antimicrobial peptides
590 from the human gut microbiome using deep learning. *Nature Biotechnology,* 40**,** 921-
591 931.
592 MADDISON, W. P. 1991. Squared-Change Parsimony Reconstructions of Ancestral States for
593 Continuous-Valued Characters on a Phylogenetic Tree. *Systematic Zoology,* 40**,** 304-
594 314.

MORIYA, S. 2021. Simple mapping-based quantification of a mock microbial community using total RNA-seq data. *PLoS One,* 16**,** e0254556.

NAVARRO, F. C. P., MOHSEN, H., YAN, C., LI, S., GU, M., MEYERSON, W. & GERSTEIN, M. 2019. Genomics and data science: an application within an umbrella. *Genome Biology,* 20**,** 109.

PAGEL, M. 1994. Detecting correlated evolution on phylogenies: a general method for the comparative analysis of discrete characters. *Proceedings of the Royal Society of London. Series B: Biological Sciences,* 255**,** 37-45.

PARK, Y. & KELLIS, M. 2015. Deep learning for regulatory genomics. *Nature Biotechnology,* 33**,** 825-826.

PRICE, M. N., DEHAL, P. S. & ARKIN, A. P. 2010. FastTree 2 – Approximately Maximum-Likelihood Trees for Large Alignments. *PLOS ONE,* 5**,** e9490.

QUAST, C., PRUESSE, E., YILMAZ, P., GERKEN, J., SCHWEER, T., YARZA, P., PEPLIES, J. & GLÖCKNER, F. O. 2012. The SILVA ribosomal RNA gene database project: improved data processing and web-based tools. *Nucleic Acids Research,* 41**,** D590-D596.

SANKOFF, D. 1975. Minimal Mutation Trees of Sequences. *SIAM Journal on Applied Mathematics,* 28**,** 35-42.

SCHMIDT, B. & HILDEBRANDT, A. 2021. Deep learning in next-generation sequencing. *Drug discovery today,* 26**,** 173-180.

SCHNEIDER, T. D. & STEPHENS, R. M. 1990. Sequence logos: a new way to display consensus sequences. *Nucleic Acids Res,* 18**,** 6097-100.

SCHNEIDER, T. D., STORMO, G. D., GOLD, L. & EHRENFEUCHT, A. 1986. Information content of binding sites on nucleotide sequences. *J Mol Biol,* 188**,** 415-31.

SHAPLEY, L. S. 1952. *A Value for N-Person Games,* Santa Monica, CA, RAND Corporation.

STACKEBRANDT, E. & GOEBEL, B. M. 1994. Taxonomic Note: A Place for DNA-DNA Reassociation and 16S rRNA Sequence Analysis in the Present Species Definition in Bacteriology. *International Journal of Systematic and Evolutionary Microbiology,* 44**,** 846-849.

STEPNIEWSKA-DZIUBINSKA, M. M., ZIELENKIEWICZ, P. & SIEDLECKI, P. 2018. Development and evaluation of a deep learning model for protein–ligand binding affinity prediction. *Bioinformatics,* 34**,** 3666-3674.

STODDARD, S. F., SMITH, B. J., HEIN, R., ROLLER, B. R. & SCHMIDT, T. M. 2015. rrnDB: improved tools for interpreting rRNA gene abundance in bacteria and archaea and a new foundation for future development. *Nucleic Acids Res,* 43**,** D593-8.

ŠTRUMBELJ, E. & KONONENKO, I. 2014. Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems,* 41**,** 647-665.

VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, L. & POLOSUKHIN, I. 2017. Attention Is All You Need. *CoRR,* abs/1706.03762.

636 WANG, Y., BHATTACHARYA, T., JIANG, Y., QIN, X., WANG, Y., LIU, Y., SAYKIN, A. J.
637 & CHEN, L. 2021. A novel deep learning method for predictive modeling of
638 microbiome data. *Brief Bioinform,* 22.
639 WOESE, C. R. & FOX, G. E. 1977. Phylogenetic structure of the prokaryotic domain: the
640 primary kingdoms. *Proceedings of the National Academy of Sciences of the United*
641 *States of America,* 74**,** 5088-5090.
642 WOESE, C. R., KANDLER, O. & WHEELIS, M. L. 1990. Towards a natural system of
643 organisms: proposal for the domains Archaea, Bacteria, and Eucarya. *Proc Natl Acad*
644 *Sci U S A,* 87**,** 4576-9.
645 WOO, P. C., LEUNG, P. K., LEUNG, K. W. & YUEN, K. Y. 2000. Identification by 16S
646 ribosomal RNA gene sequencing of an Enterobacteriaceae species from a bone marrow
647 transplant recipient. *Mol Pathol,* 53**,** 211-5.
648 YANG, B., WANG, Y. & QIAN, P. Y. 2016. Sensitivity and correlation of hypervariable regions
649 in 16S rRNA genes in phylogenetic analysis. *BMC Bioinformatics,* 17**,** 135.
650 YARZA, P., YILMAZ, P., PRUESSE, E., GLÖCKNER, F. O., LUDWIG, W., SCHLEIFER,
651 K.-H., WHITMAN, W. B., EUZÉBY, J., AMANN, R. & ROSSELLÓ-MÓRA, R. 2014.
652 Uniting the classification of cultured and uncultured bacteria and archaea using 16S
653 rRNA gene sequences. *Nature Reviews Microbiology,* 12**,** 635-645.
654 ZHA, Y., CHONG, H., QIU, H., KANG, K., DUN, Y., CHEN, Z., CUI, X. & NING, K. 2022.
655 Ontology-aware deep learning enables ultrafast and interpretable source tracking
656 among sub-million microbial community samples from hundreds of niches. *Genome*
657 *Medicine,* 14**,** 43.
658 ZHAO, Z., WOLOSZYNEK, S., AGBAVOR, F., MELL, J. C., SOKHANSANJ, B. A. &
659 ROSEN, G. L. 2021. Learning, visualizing and exploring 16S rRNA structure using an
660 attention-based deep neural network. *PLoS computational biology* [Online], 17.
661 Available: http://europepmc.org/abstract/MED/34550967 [Accessed 2021/09//].

662

663

664 **TABLES**

665

666 **Table 1.** Performance of all algorithms.

Algorithm Category	Model	SD	Mean RMSE
Phylogeny-based	EP	0.0443	0.845
	MK	0.112	5.76
	MPR	0.0375	0.787
	PIC	0.0243	0.789
	SA	0.0329	0.774
	WSCP	0.0273	0.775
Taxonomy-based	PR2+TA	0.0597	1.26
	RDP+TA	0.0356	1.18
	SILVA+TA	0.0377	1.20
Deep Learning	Transformer	0.0643	0.861
	MLP	0.0388	0.729

667 MK: maximum-likelihood of fixed-rates continuous-time Markov models; EP: empirical
668 probabilities; MPR: Sankoff's maximum-parsimony; PIC: phylogenetic independent contrast;
669 SA: Subtree Averaging; WSCP: weighted-squared-change parsimony; RDP+TA: sub-taxa
670 averaging based on RDP database; SILVA+TA: sub-taxa averaging based on SILVA database;
671 PR2+TA: sub-taxa averaging based on PR2 database; MLP: multi-layer perceptron model;
672 Transformer: Transformer model.

673

674 **Table 2.** Performance of selected algorithms on 16S subregions.

Region	Model	SD	Mean RMSE
V1-V2	MLP-R	0.02367864	0.7978728
	EP	0.03312260	0.9276227
	MPR	0.03703488	0.8670806
	PIC	0.02811679	0.8095574
	SA	0.03205930	0.8288667
	WSCP	0.02420925	0.7927451
V1-V3	MLP-R	0.02417787	0.7751063
	EP	0.04763446	0.8889738
	MPR	0.03050384	0.8363964
	PIC	0.03068616	0.8051317
	SA	0.04302227	0.8050986
	WSCP	0.03299523	0.7824740
V3-V4	MLP-R	0.04929372	0.7811680
	EP	0.04471593	0.9018608
	MPR	0.04769014	0.8583522
	PIC	0.05032324	0.7894828
	SA	0.03785498	0.8111553
	WSCP	0.04830682	0.7791547
V4	MLP-R	0.03688014	0.8127494
	EP	0.05693514	0.9160779
	MPR	0.04745741	0.8791149
	PIC	0.03767273	0.8238156
	SA	0.03875001	0.8218996
	WSCP	0.04046582	0.8226994
V4-V5	MLP-R	0.03665258	0.7887458
	EP	0.03397454	0.9089205

	MPR	0.04128272	0.8579178
	PIC	0.02769405	0.7847597
	SA	0.02830729	0.8101834
	WSCP	0.03535666	0.7847232
V6-V8	MLP-R	0.03607741	0.7850310
	EP	0.05631475	0.9086637
	MPR	0.03998800	0.8523850
	PIC	0.04035999	0.7812806
	SA	0.03908318	0.8072924
	WSCP	0.03771673	0.7794703
V7-V9	MLP-R	0.03250365	0.7937580
	EP	0.03535698	0.9311839
	MPR	0.03702084	0.8464945
	PIC	0.03925002	0.8019705
	SA	0.03337741	0.8304700
	WSCP	0.03747741	0.7896712

EP: empirical probabilities; MPR: Sankoff's maximum-parsimony; PIC: phylogenetic independent contrast; SA: Subtree Averaging; WSCP: weighted-squared-change parsimony; MLP-R: multi-layer perceptron model by region.

FIGURES

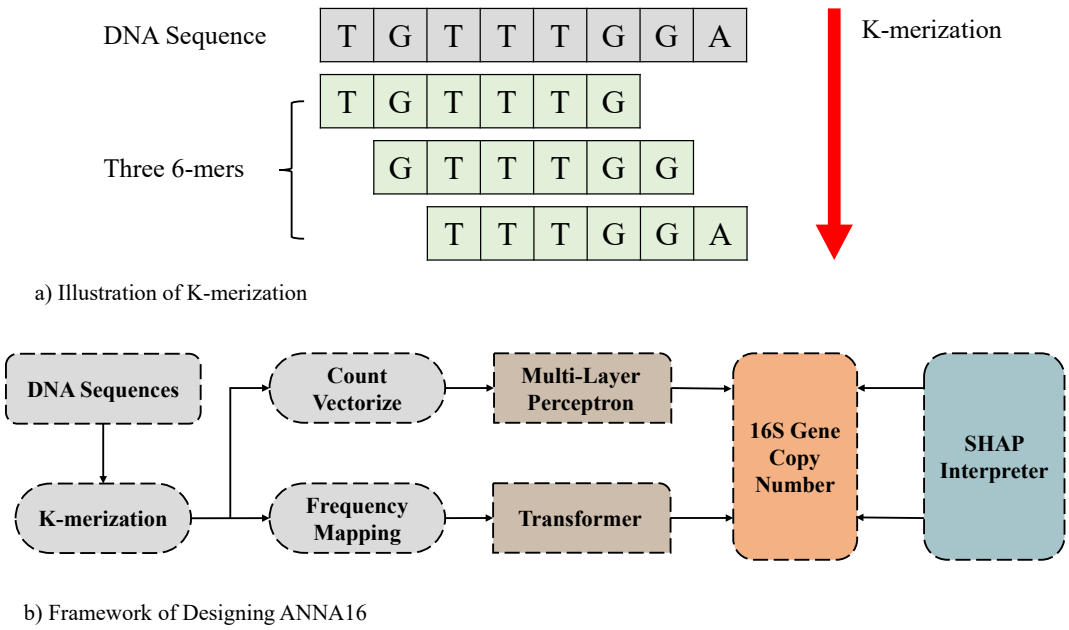


Figure 1. (a) K-mer strategy (K = 6). (b) Workflow of proposed MLP model.

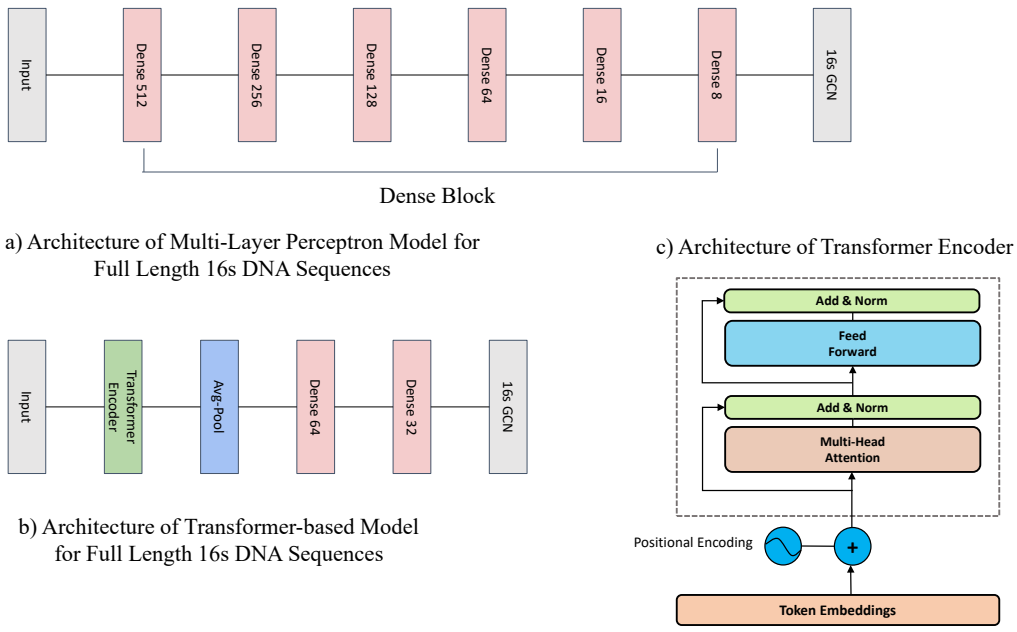
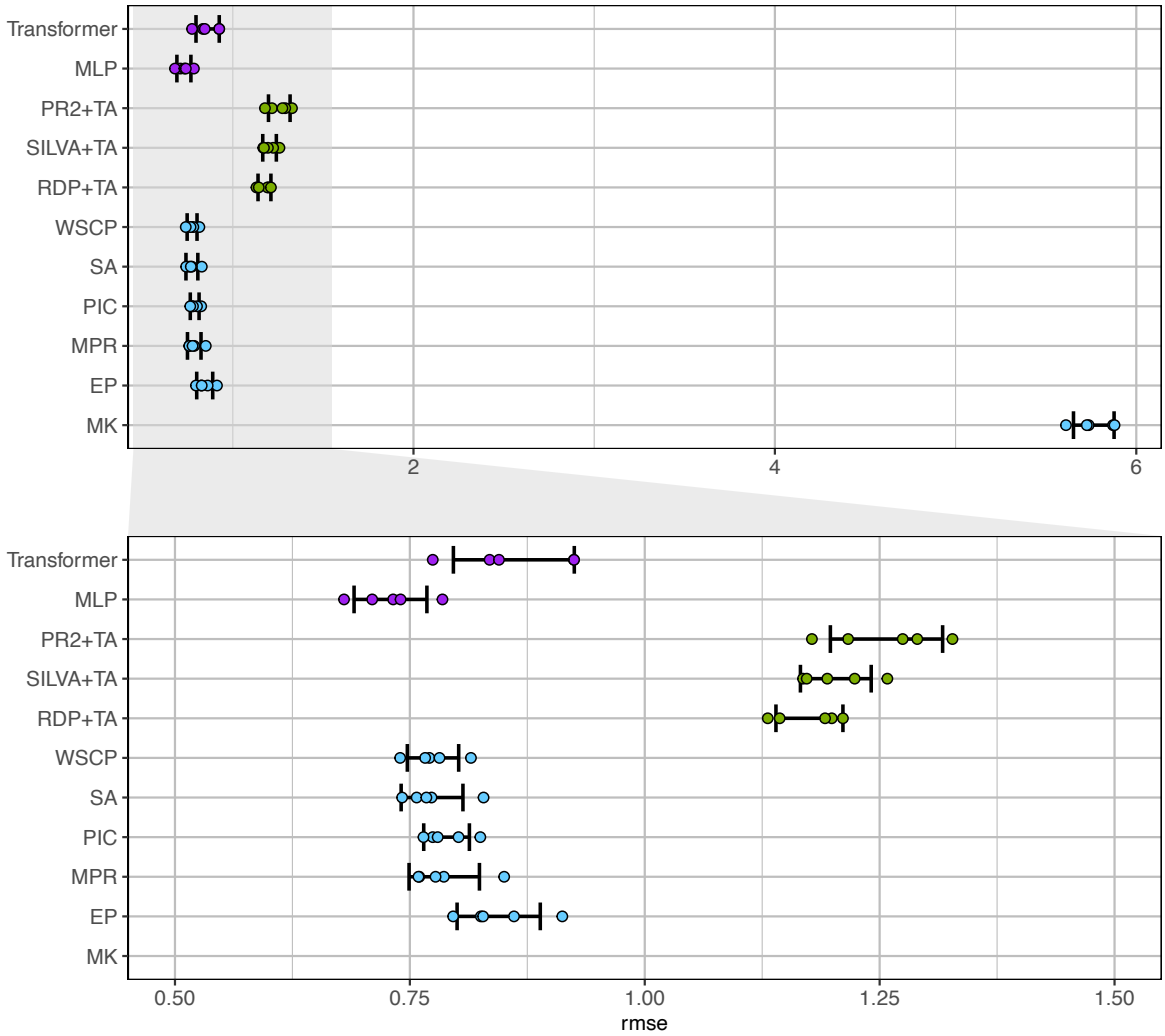


Figure 2. Illustration of Architecture of Deep Learning Models.



686

687 **Figure 3.** Performance of all algorithms or models. MK: maximum-likelihood of fixed-rates
688 continuous- time Markov models; EP: empirical probabilities; MPR: Sankoff's maximum-
689 parsimony; PIC: phylogenetic independent contrast; SA: Subtree Averaging; WSCP: weighted-
690 squared-change parsimony; RDP+TA: sub-taxa averaging based on RDP database; SILVA+TA:
691 sub-taxa averaging based on SILVA database; PR2+TA: sub-taxa averaging based on PR2
692 database; MLP: multi-layer perceptron model; Transformer: Transformer model. The top panel
693 exhibits a global view from 0 to 6 copies/genome, and the bottom panel is a zoom-in view from
694 0.5 to 1.5 copies/genome.

695

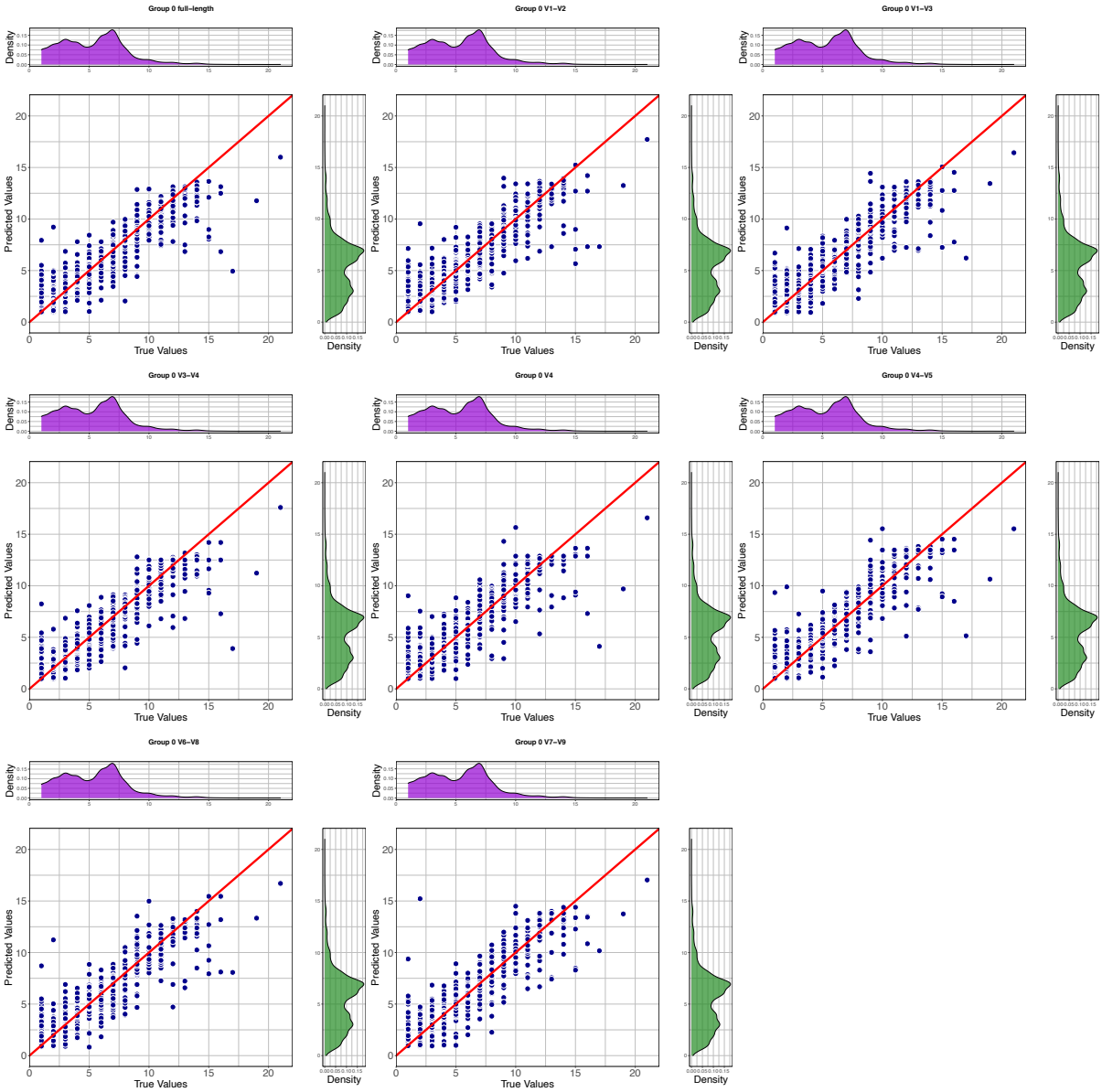


Figure 4. Prediction by MLP on 16S full length and subregions in the first cycle of 5-fold cross-validation. The red line is a diagonal line for illustrating how much predicted values derive from true values. The density plot filled in violet shows the distribution of true values, and the green density plot shows the distribution of predicted values.

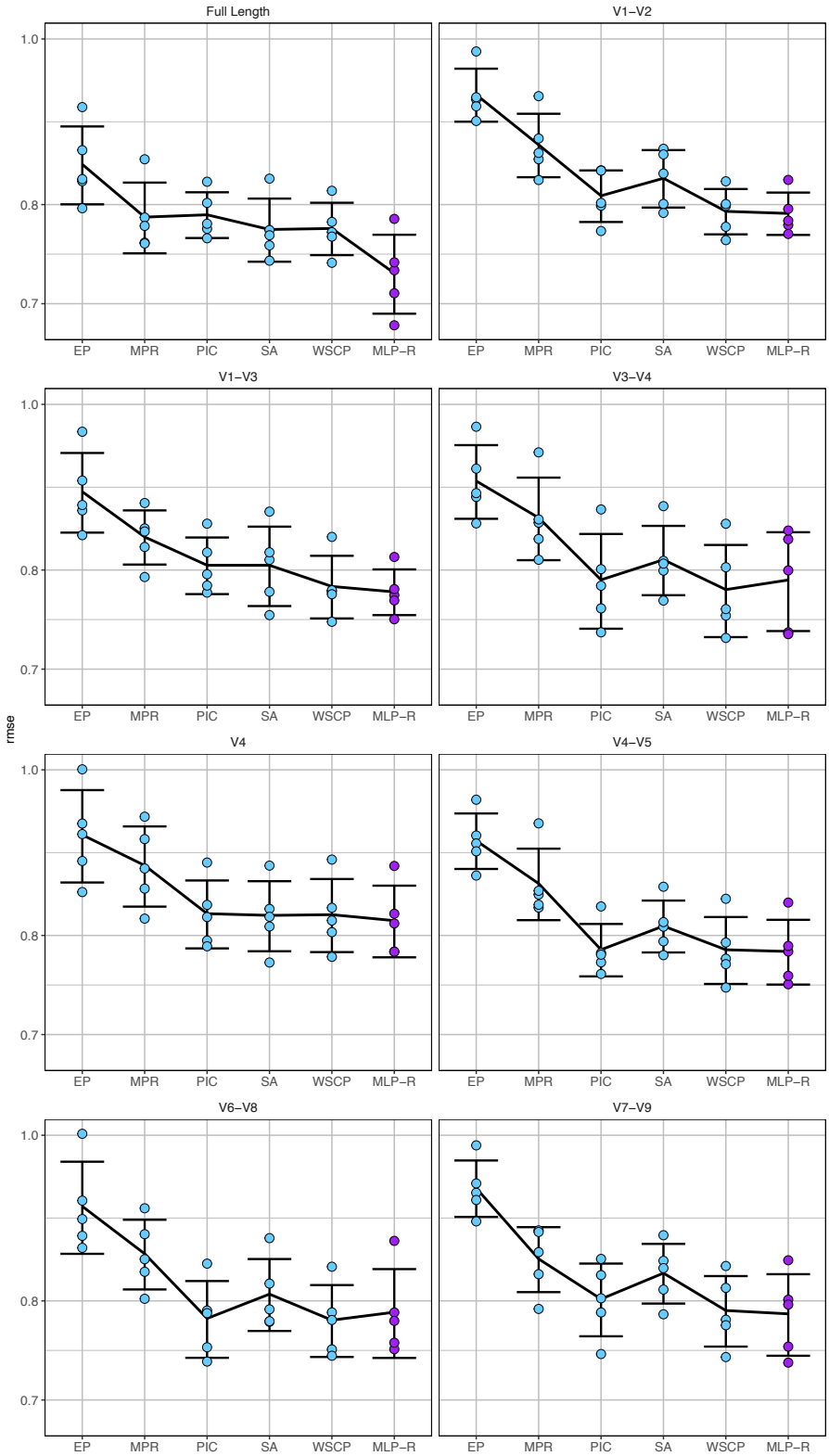


Figure 5. Performance of selected algorithms or models on 16S full length and subregions. EP: empirical probabilities; MPR: Sankoff's maximum-parsimony; PIC: phylogenetic independent contrast; SA: Subtree Averaging; WSCP: weighted-squared-change parsimony; MLP-R: multi-layer perceptron model by regions

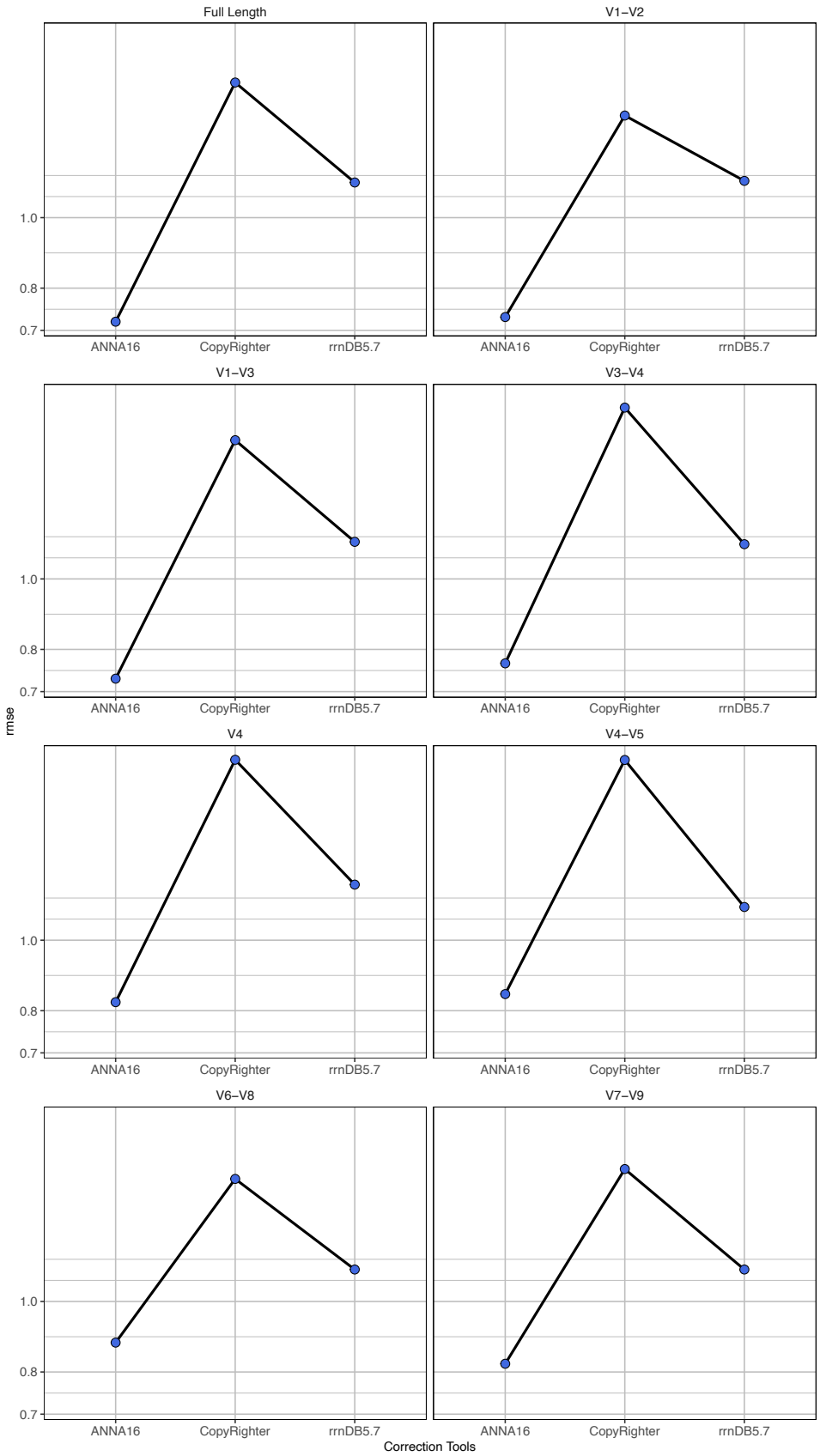


Figure 6. Comparison of performance on independent dataset among CopyRighter, rrnDB, and ANNA16

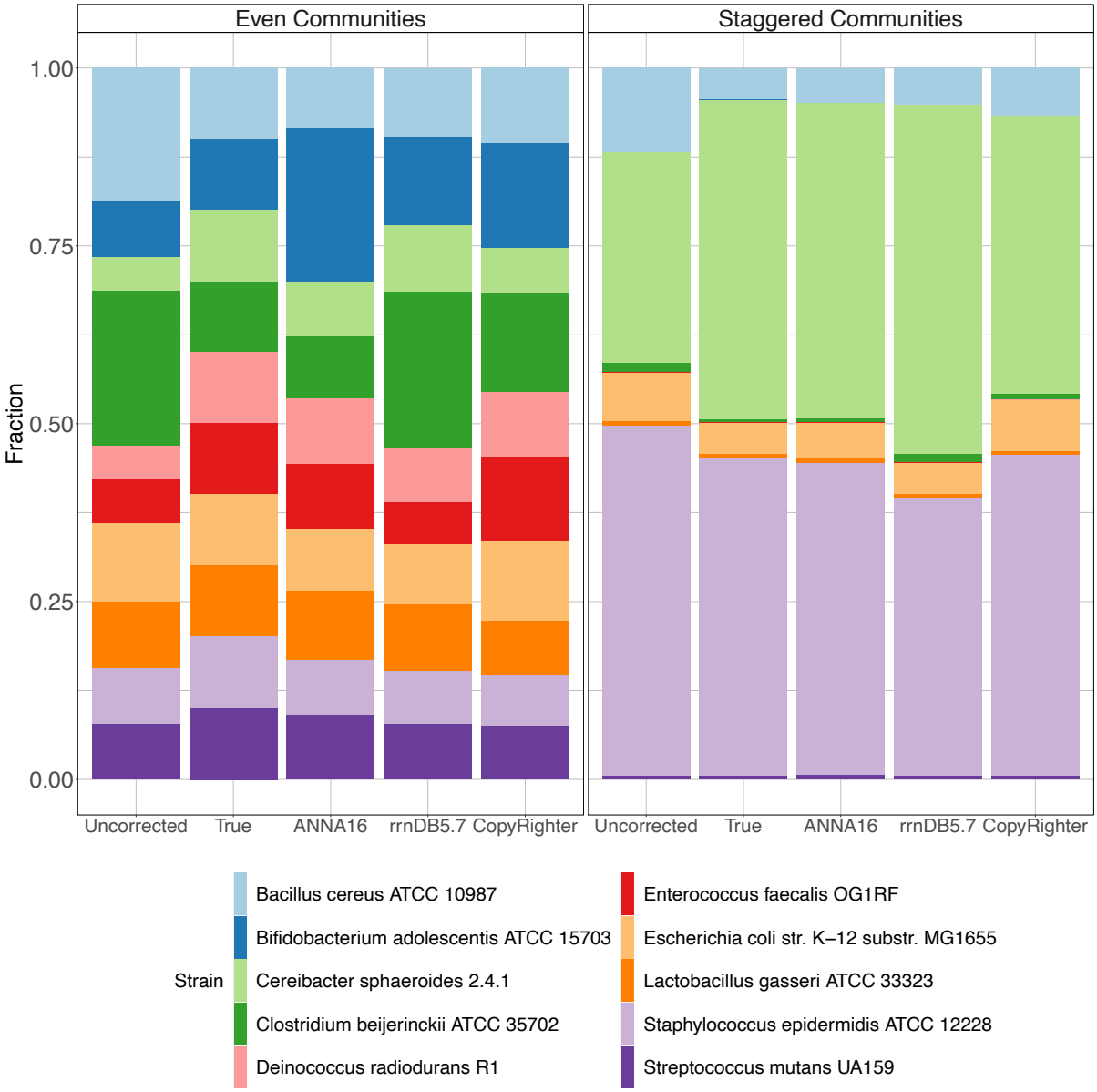


Figure 7. Uncorrected copy fraction, true cell fraction, and cell fraction (microbial profiles) estimated by ANNA16, rrnDB (version 5.7), and CopyRighter. Uncorrected: Uncorrected copy fraction; True: True cell fraction; ANNA16: Cell fraction estimated by ANNA16; rrnDB5.7: Cell fraction estimated by rrnDB (version 5.7); CopyRighter: Cell fraction estimated by CopyRighter.

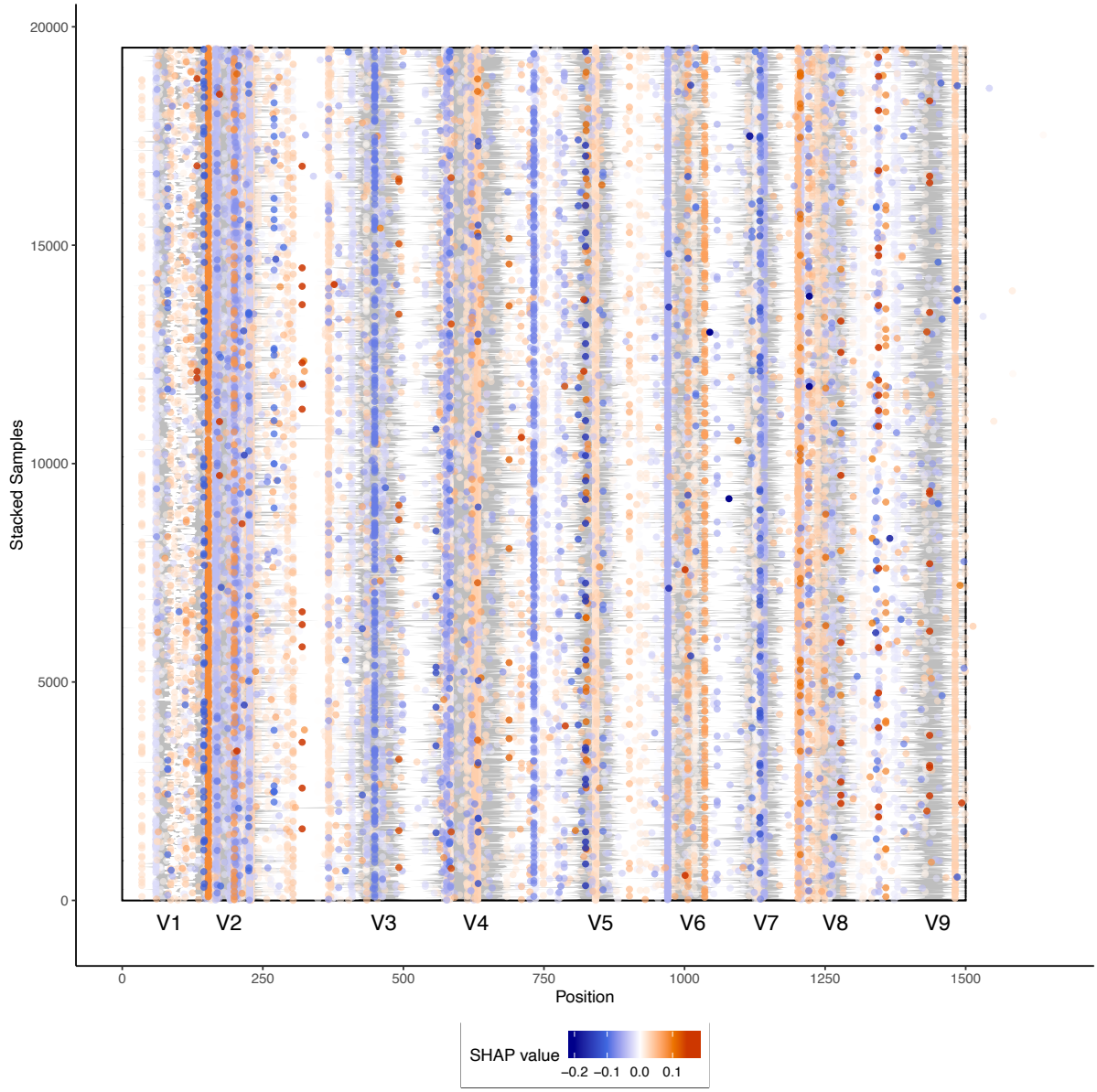


Figure 8. Distribution of SHAP values on 16S full length sequences. To make the figure interpretable, only the value of top 100 K-mers ranked by mean absolute SHAP value is shown. Hypervariable regionsare labelled as grey shadows. To determine the range of hypervariable regions, sequence of conserved regions of E. coli is first extracted according to region coordinates provided by Yarza et al. (Yarza et al., 2014). All sequences in the preprocessed dataset undergo BLASTN (version 2.11.0) against E. coli conserved sequences to determine the range of conserved regions in each sequence. Range of hypervariable regions is thus deduced from conserved regions. To make the figure less complicated, only 5,000 randomly chosen sequences are labelled. V-region labels on x-axis shows the general position of *E. coli* V-regions according to coordinates from Yarza et al. (Yarza et al., 2014).

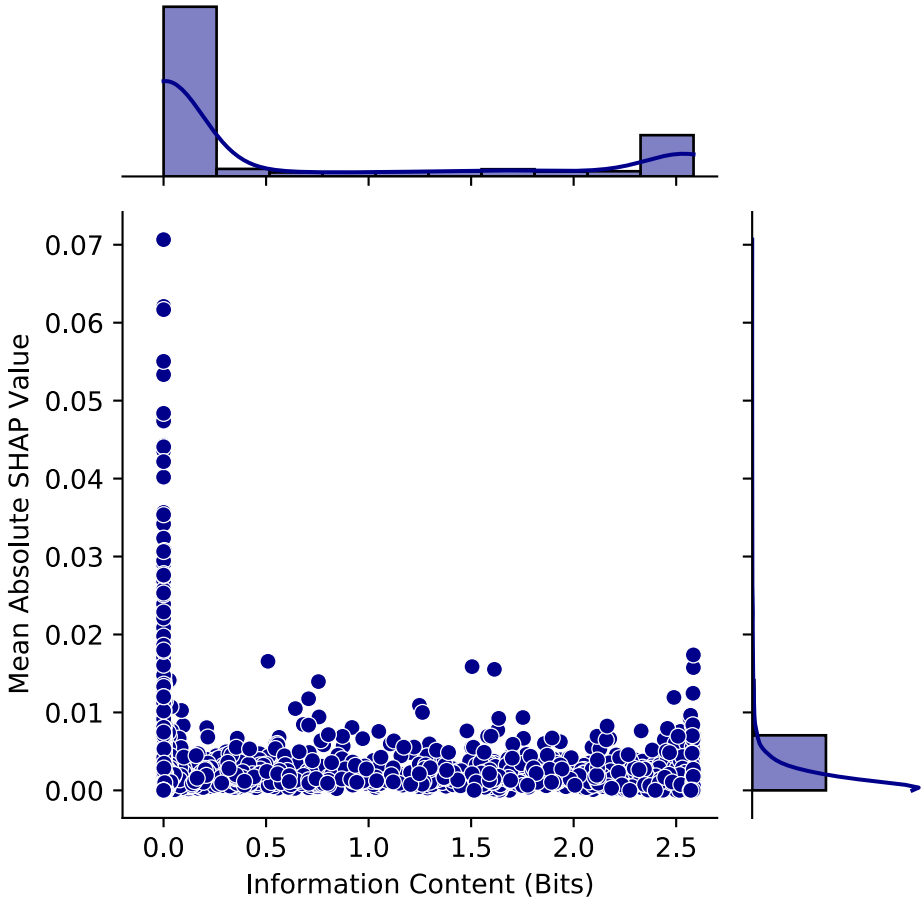


Figure 9. Relationship of mean absolute SHAP values and information content (unit: bits) on each position of aligned 16S full length sequence.