IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 19, 2021, accepted November 4, 2021, date of publication November 12, 2021,
date of current version December 16, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3127957

NASCUP: Nucleic Acid Sequence Classification by Universal Probability

SUNYOUNG KWON[1], (Member, IEEE), GYUWAN KIM[2], BYUNGHAN LEE[3], (Member, IEEE), JONGSIK CHUN[4,5], SUNGROH YOON[5,6], (Senior Member, IEEE), AND YOUNG-HAN KIM[7], (Fellow, IEEE)

[1]School of Biomedical Convergence Engineering, Pusan National University, Yangsan 50612, South Korea
[2]Department of Computer Science, University of California, Santa Barbara, Santa Barbara, CA 93106, USA
[3]Department of Electronic and IT Media Engineering, Seoul National University of Science and Technology, Seoul 01811, South Korea
[4]School of Biological Sciences, Seoul National University, Seoul 08826, South Korea
[5]Interdisciplinary Program in Bioinformatics, Seoul National University, Seoul 08826, South Korea
[6]Department of Electrical and Computer Engineering, Seoul National University, Seoul 08826, South Korea
[7]Department of Electrical and Computer Engineering, University of California, San Diego, San Diego, CA 92093, USA

Corresponding authors: Sungroh Yoon (sryoon@snu.ac.kr) and Young-Han Kim (yhk@ucsd.edu)

This work was supported in part by the National Research Foundation of Korea (NRF) Grant by the Korean Government through the Ministry of Science and ICT under Grant 2018R1A2B3001628; in part by the BK21 FOUR Program of the Education and Research Program for Future ICT Pioneers, Seoul National University, in 2021; in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant by the Korean Government through the Ministry of Science and ICT (MSIT) by the Artificial Intelligence Convergence Research Center, Pusan National University, under Grant 2020-0-01450.

ABSTRACT Nucleic acid sequence classification is a fundamental task in the field of bioinformatics. Due to the increasing amount of unlabeled nucleotide sequences, fast and accurate classification of them on a large scale has become crucial. In this work, we developed NASCUP, a new classification method that captures statistical structures of nucleotide sequences by compact context-tree models and universal probability from information theory. A comprehensive experimental study involving nine public databases for functional non-coding RNA, microbial taxonomy and coding/non-coding RNA classification demonstrates the advantages of NASCUP over widely-used alternatives in efficiency, accuracy, and scalability across all datasets considered. NASCUP achieved BLAST-like classification accuracy consistently for several large-scale databases in orders-of-magnitude reduced runtime, and was applied to other bioinformatics tasks such as outlier detection and synthetic sequence generation.

INDEX TERMS Bioinformatics, context-tree models, information theory, sequence classification, universal probability.

I. INTRODUCTION

Sequence classification plays a key role in various bioinformatics pipelines by revealing the proximity and membership of a biological sequence to known sequence groups [1]–[5]. Expedited by new sequencing technologies, nucleotide sequence databases are rapidly expanding at a rate that exceeds that of the technologies to handle the bioinformatics around the sequences [6]. Moreover, existing databases sometimes contain mislabeled sequences that can potentially impair the identification accuracy significantly [7]. To address these challenges, we present NASCUP, an accurate and computationally efficient classification method that is scalable for large and growing datasets and robust against mislabeling errors.

Common approaches to sequence classification can be broadly divided into two categories—alignment-based and model-based ones. In alignment-based approaches, the class of the query sequence is determined by sequence-to-sequence comparison. Alignment tools, including BLAST [8], typically exhibit high accuracy, but are vulnerable to errors and often becomes time-consuming as the number of sequences increases. Model-based approaches, such as RDP [9], HMMER [10], and Phymm [11], derive statistical models from each group of sequences and compare the query sequence to these models. Sequence-to-model comparison is more scalable than sequence-to-sequence comparison, but it is often difficult to extract a model from a group

The associate editor coordinating the review of this manuscript and approving it for publication was Vincenzo Conti.

of sequences that is statistically meaningful and does not overfit the data. The proposed NASCUP, which belongs to model-based classification method, combines the merits of existing model-based methods and compression-based methods. By utilizing compact context-tree models along with the notion of universal probability from information theory, NASCUP delivers high accuracy comparable to the sequence-to-sequence comparison approaches, while providing robustness and scalability as a computationally efficient sequence-to-model comparison approach suitable for large-scale, expanding datasets.

NASCUP utilizes the context-tree model constructed from k-mer statistics of training sequences with the notion of minimax universal probability measure used in information theory [12]–[14]. This allows for efficient and effective modeling and classification of multiple sequences in two stages, improving significantly upon the performance and scalability of compression based approaches. In addition to alleviating the efficiency problem of existing k-mer based approaches (sparse vs. dense trees) in representing the probabilistic structure of a sequence, these contexts also capture the textual information of the sequence by only keeping segments of statistical importance.

Since NASCUP does not rely on any sequence-to-sequence comparison, the classification time of NASCUP is independent of the number of sequences per family, which enables a significant speedup. For gigabase-scale databases, such as Greengenes [15] and SILVA [16], many alignment-based methods would take days or even weeks for database-wide classification, whereas the time demand of NASCUP is orders-of-magnitude lower. NASCUP runs faster than HMM-based approaches which require MSA preprocessing and ICM-based approaches which also utilize context trees.

In this study, we experimentally prove highly competitive performance of NASCUP in diverse datasets in terms of classification accuracy and time efficiency. Regardless of the number of families and the level of intra-family sequence similarity, which affected the performance of most of the alternatives, NASCUP successfully maintained its accuracy with sufficiently fast speed, confirming its robustness.

II. BACKGROUND

A. MARKOV AND CONTEXT-TREE MODELS

The simplest probabilistic model for nucleotide sequences is the independent and identical distribution (IID) model that assigns one of the four fixed probabilities $p_{\mathtt{A}}$, $p_{\mathtt{C}}$, $p_{\mathtt{G}}$, and $p_{\mathtt{T}}$ to each symbol, and computes the probability of the entire sequence as the product of those probabilities. More precisely, a sequence $x = \mathtt{X}_1 \cdots \mathtt{X}_n$ with symbols $\mathtt{A}$, $\mathtt{C}$, $\mathtt{G}$, and $\mathtt{T}$ respectively appearing $n_{\mathtt{A}}$, $n_{\mathtt{C}}$, $n_{\mathtt{G}}$, and $n_{\mathtt{T}}$ times has the probability

$$P(x) = p_{\mathtt{A}}^{n_{\mathtt{A}}} p_{\mathtt{C}}^{n_{\mathtt{C}}} p_{\mathtt{G}}^{n_{\mathtt{G}}} p_{\mathtt{T}}^{n_{\mathtt{T}}}. \tag{1}$$

A d-th order Markov model introduces dependence across $d+1$ consecutive symbols (*i.e.*, k-mers for $k = d+1$) by assigning one of the probabilities $p_{\mathtt{A}}(s)$, $p_{\mathtt{C}}(s)$, $p_{\mathtt{G}}(s)$, and $p_{\mathtt{T}}(s)$ to the symbol $\mathtt{X}_i$ at position i if the previous d symbols $\mathtt{X}_{i-d} \cdots \mathtt{X}_{i-1}$, namely, the *state*, is equal to s. For example, a second-order Markov model assigns the probability $p_{\mathtt{G}}(\mathtt{GG})^2 p_{\mathtt{A}}(\mathtt{GG}) p_{\mathtt{C}}(\mathtt{GG}) p_{\mathtt{G}}(\mathtt{AG}) p_{\mathtt{G}}(\mathtt{GA})$ to the sequence $\mathtt{GG|GAGGGC}$ from the third position. In general, a d-th order Markov model assigns the probability $P(x) = \prod_{i=d+1}^{n} p_{\mathtt{X}_i}(\mathtt{X}_{i-d} \cdots \mathtt{X}_{i-1})$ to the sequence $x = \mathtt{X}_1 \cdots \mathtt{X}_d | \mathtt{X}_{d+1} \cdots \mathtt{X}_n$ from the $(d+1)$-st position. By parsing the sequence into subsequences by states, this probability can be equivalently expressed as

$$\begin{aligned} P(x) &= \prod_s P_s(x) \\ &= \prod_s p_{\mathtt{A}}(s)^{n_{\mathtt{A}}(s)} p_{\mathtt{C}}(s)^{n_{\mathtt{C}}(s)} p_{\mathtt{G}}(s)^{n_{\mathtt{G}}(s)} p_{\mathtt{T}}(s)^{n_{\mathtt{T}}(s)}, \end{aligned} \tag{2}$$

where the products are over all states $s \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}^d$, $P_s(x)$ denotes the probability assigned to the subsequence of x in state s (that is, the symbols in x that are preceded by s), and $n_{\mathtt{X}}(s)$ denotes the number of occurrences of symbol $\mathtt{X} \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$ in that subsequence. An IID model corresponds to a zeroth-order Markov (1-mer) model with the empty string as the only state, and that a d-th order Markov model can be decomposed as multiple IID models, each corresponding to one of the 4^d distinct states. A hidden Markov model used in HMMER [10] is a generalization of a Markov model by stochastically transforming a Markov model symbol-by-symbol.

A context-tree model (CTM) [17] is both a specialization and generalization of a Markov model, in which states with a common suffix share the same probability assignments and thus are aggregated to form a *context*. For example, the set $\{\mathtt{AA}, \mathtt{CA}, \mathtt{GA}, \mathtt{TA}\}$ consists of all possible states of a second-order Markov chain that has the common suffix $\mathtt{A}$. This set is represented in shorthand notation $\star\mathtt{A}$, where $\star$ denotes "any" symbol. Since the probability assignments are the same for all states in the context $\star\mathtt{A}$, the symbols preceded by $\mathtt{A}$ effectively follow a first-order Markov distribution. Since the effective Markov order varies from one context to another, a CTM is also referred to as a variable-order Markov model [18]. Each CTM is represented by a collection $\mathcal{S}$ of contexts that partition $\{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}^d$ and parameters $p(s) = (p_{\mathtt{A}}(s), p_{\mathtt{C}}(s), p_{\mathtt{G}}(s), p_{\mathtt{T}}(s))$ associated with each context $s \in \mathcal{S}$. Using the contexts in $\mathcal{S}$ in as leaves and merging contexts that share suffixes in a hierarchical manner, we can form a proper (that is, each node has 0 or 4 children) suffix tree with root node $\star^d = \star \cdots \star$ (d times). For example, the contexts $\star\mathtt{A}$, $\mathtt{AC}$, $\mathtt{CC}$, $\mathtt{GC}$, $\mathtt{TC}$, $\star\mathtt{G}$, $\star\mathtt{T}$ partition $\{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}^2$ and form a suffix tree. Consequently, a CTM is equivalently represented by a suffix tree along with probability assignments on its leaves, and thus is referred to as a probabilistic suffix tree model [19] as well. A d-th order Markov model can be viewed as a CTM on a perfect suffix tree that has $\{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}^d$ as its 4^d leaves. A typical CTM of depth d consists of a fewer number of contexts (and corresponding probability parameters) than a d-th order Markov model, providing a more succinct representation of the data. A CTM

can be further generalized by allowing states to be merged to a context at any position, *e.g.*, $\mathtt{A}\star = \{\mathtt{AA}, \mathtt{AC}, \mathtt{AG}, \mathtt{AT}\}$. An interpolated context model used in Phymm [11] is a mixture of such generalized CTMs. NASCUP uses only suffix trees since this restriction does not incur any empirical performance loss compared to CTMs, and it allows computationally efficient comparison among all suffix trees.

B. MAXIMUM LIKELIHOOD AND UNIVERSAL PROBABILITY ESTIMATES

When the parameters $p_{\mathtt{A}}, p_{\mathtt{C}}, p_{\mathtt{G}}, p_{\mathtt{T}}$ of an IID model are not known, the most naive approach to estimating the true probability $P(x)$ of a given sequence x is to find the parameters that maximize the sequence probability expression in (1) and then to compute the probability of the entire sequence using these parameter estimates. It can be easily verified that the empirical probabilities $(p_{\mathtt{A}}, p_{\mathtt{C}}, p_{\mathtt{G}}, p_{\mathtt{T}}) = (n_{\mathtt{A}}/n, n_{\mathtt{C}}/n, n_{\mathtt{G}}/n, n_{\mathtt{T}}/n)$ maximize (1) for any sequence x with symbol counts $n_{\mathtt{A}}, n_{\mathtt{C}}, n_{\mathtt{G}}, n_{\mathtt{T}}$ and length $n = n_{\mathtt{A}} + n_{\mathtt{C}} + n_{\mathtt{G}} + n_{\mathtt{T}}$. The resulting *maximum likelihood (ML) estimate* of the true probability is

$$Q^{\mathrm{ML}}(x) = \left(\frac{n_{\mathtt{A}}}{n}\right)^{n_{\mathtt{A}}} \left(\frac{n_{\mathtt{C}}}{n}\right)^{n_{\mathtt{C}}} \left(\frac{n_{\mathtt{G}}}{n}\right)^{n_{\mathtt{G}}} \left(\frac{n_{\mathtt{T}}}{n}\right)^{n_{\mathtt{T}}}. \tag{3}$$

Note that $Q^{\mathrm{ML}}(x)$ is a function only of the symbol count vector $\mathbf{n} = (n_{\mathtt{A}}, n_{\mathtt{C}}, n_{\mathtt{G}}, n_{\mathtt{T}})$. Hence, with a slight abuse of notation, we will write the righthand side (RHS) of (3) as $Q^{\mathrm{ML}}(\mathbf{n})$. More generally, for a depth-d CTM on a given context set $\mathcal{S}$ with unknown parameters $p(s)$, $s \in \mathcal{S}$, the ML estimate $Q^{\mathrm{ML}}(x)$ of the true probability $P(x) = \prod_s P_s(x)$ (from the $(d+1)$-st position) as in (2) is the product of the ML estimates of subsequence probabilities $P_s(x)$, namely,

$$Q^{\mathrm{ML}}(x) = \prod_{s \in \mathcal{S}} Q^{\mathrm{ML}}(\mathbf{n}(s)), \tag{4}$$

where $Q^{\mathrm{ML}}(\mathbf{n}(s))$ denotes the ML estimate of an IID probability in (3) evaluated with the count vector $\mathbf{n}(s) = (n_{\mathtt{A}}(s), n_{\mathtt{C}}(s), n_{\mathtt{G}}(s), n_{\mathtt{T}}(s))$ for context s. The ML estimate $Q^{\mathrm{ML}}(x)$ overfits the given data x by discounting the symbols that did not appear in it. In fact, $Q^{\mathrm{ML}}(\cdot)$ is not a valid probability assignment since the sum of the estimated probabilities $Q^{\mathrm{ML}}(x)$ over all sequences $x \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}^n$ is greater than 1.

As an alternative, NASCUP relies on the notion of *universal probability* [20], [21] in information theory that is chosen independent of the data x and is close to all unknown probability models in a given class. The most basic example of universal probability is the *Krichevski–Trofimov (KT) estimate* [12] for IID models that assigns the sequence probability

$$Q^{\mathrm{KT}}(x) = \int p_{\mathtt{A}}^{n_{\mathtt{A}}} p_{\mathtt{C}}^{n_{\mathtt{C}}} p_{\mathtt{G}}^{n_{\mathtt{G}}} p_{\mathtt{T}}^{n_{\mathtt{T}}} f(p_{\mathtt{A}}, p_{\mathtt{C}}, p_{\mathtt{G}}, p_{\mathtt{T}}) dp_{\mathtt{A}} dp_{\mathtt{C}} dp_{\mathtt{G}} dp_{\mathtt{T}}, \tag{5}$$

where $f(p_{\mathtt{A}}, p_{\mathtt{C}}, p_{\mathtt{G}}, p_{\mathtt{T}})$ is the Dirichlet prior on the quaternary probability simplex with parameters $1/2, 1/2, 1/2, 1/2$. As a Dirichlet mixture of IID models, the KT estimate is a valid probability assignment (that is, $Q^{\mathrm{KT}}(x) \geq 0$ for every x and $\sum_x Q^{\mathrm{KT}}(x) = 1$). Moreover, Q^{KT} is *uniformly* close to every IID probability model P on quaternary sequences of length n in the sense that both the relative entropy (Kullback–Leibler divergence) [22]

$$D(P \| Q^{\mathrm{KT}}) = \sum_{x \in \{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}^n} P(x) \log \frac{P(x)}{Q^{\mathrm{KT}}(x)}$$

and the maximum log likelihood ratio

$$\max_x \log \frac{P(x)}{Q^{\mathrm{KT}}(x)}$$

are upper bounded by $(3/2)\log n$ plus uniform constants independent of P, which vanishes when normalized by the sequence length n and is essentially tight as no other probability estimate can approximate all IID probability models uniformly closer [23], [24].

Since the Dirichlet distribution is the conjugate prior for the parameters of an IID model, the KT probability estimate has the predictive "add-half" formula for the conditional probability

$$Q^{\mathrm{KT}}(\mathtt{X}_{i+1} = \mathtt{X} | \mathtt{X}_1, \ldots, \mathtt{X}_i) = \frac{i_{\mathtt{X}} + 1/2}{i+2}, \tag{6}$$

when $\mathtt{X} \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$, $i = 0, 1, 2, \ldots$, and the sequence $\mathtt{X}_1 \cdots \mathtt{X}_i$ has symbol counts $i_{\mathtt{A}}, i_{\mathtt{C}}, i_{\mathtt{G}}, i_{\mathtt{T}}$. Applying this predictive estimate sequentially, the KT probability estimate of the entire length-n sequence x in (5) can be expressed as

$$\begin{aligned} Q^{\mathrm{KT}}(x) &= \frac{\prod_{\mathtt{X} \in \{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}} \prod_{i_{\mathtt{X}}=1}^{n_{\mathtt{X}}} (i_{\mathtt{X}} - 1/2)}{\prod_{i=1}^{n} (i+1)} \\ &= \frac{\prod_{\mathtt{X} \in \{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}} \Gamma(n_{\mathtt{X}} + 1/2)}{\pi^2 \Gamma(n+2)}, \end{aligned} \tag{7}$$

where $\Gamma(\cdot)$ is the standard Gamma function. As with the ML estimate, the KT estimate is a function of x only through the symbol count vector $\mathbf{n} = (n_{\mathtt{A}}, n_{\mathtt{C}}, n_{\mathtt{G}}, n_{\mathtt{T}})$ and hence we will write the RHS of (7) as $Q^{\mathrm{KT}}(\mathbf{n})$. Further extending the predictive estimate in (6), we can express the conditional probability of a sequence x with symbol count vector $\mathbf{m}$ given a preceding sequence y with symbol count vector $\mathbf{n}$ as

$$Q^{\mathrm{KT}}(x|y) = \frac{Q^{\mathrm{KT}}(y, x)}{Q^{\mathrm{KT}}(y)} = \frac{Q^{\mathrm{KT}}(\mathbf{m} + \mathbf{n})}{Q^{\mathrm{KT}}(\mathbf{n})}. \tag{8}$$

Paralleling (2) and (4), we can generalize the KT estimate to a CTM on a given context set $\mathcal{S}$ with unknown parameters $p(s)$. In this case, the KT estimate of the probability of the sequence x with symbol count vectors $\mathbf{n}(s)$, $s \in \mathcal{S}$, is

$$Q^{\mathrm{KT}}(x) = \prod_{s \in \mathcal{S}} Q^{\mathrm{KT}}(\mathbf{n}(s)) \tag{9}$$

and the KT estimate of the conditional probability of x with symbol count vectors $\mathbf{m}(s)$ given y with symbol count vectors $\mathbf{n}(s)$ is

$$Q^{\mathrm{KT}}(x|y) = \prod_{s \in \mathcal{S}} \frac{Q^{\mathrm{KT}}(\mathbf{m}(s) + \mathbf{n}(s))}{Q^{\mathrm{KT}}(\mathbf{n}(s))}. \tag{10}$$

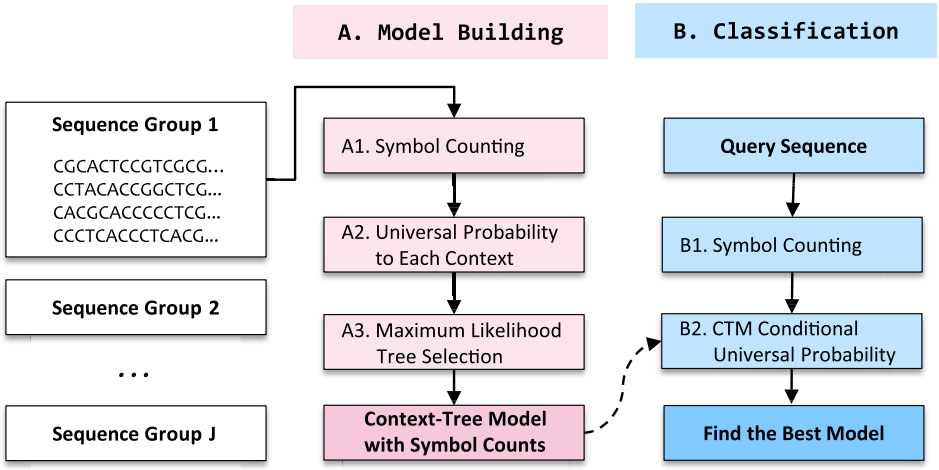

FIGURE 1. Overall flow of NASCUP. The proposed NASCUP methodology consists of (A) model-building and (B) classification pipelines that leverage the notion of universal probability in steps A2 and B2 in place of the unknown true probability. For each sequence group, the context-tree model with the highest universal probability is found and the query sequence is classified to the group with the highest conditional universal probability given the context tree model.

As in the IID case, the KT probability estimate is universal in the sense that Q^{KT} is uniformly close to all CTMs on the given context set $\mathcal{S}$.

III. METHODS

Similar to most model-based sequence classification tools, the NASCUP pipeline consists of two stages as illustrated in (**Fig 1**): model building and classification. In the first, model-building stage, NASCUP learns the statistical structure of each nucleotide sequence group in a database from the occurrence counts of all k-mers (substrings of length k) in the sequences and builds a corresponding *context-tree model* (CTM) [17] (alternatively referred to as variable-order Markov models [18] or probabilistic suffix trees [19]) that represents the data best. Such context-tree models, as reported for protein sequence classification [25] are simple enough for fast and scalable processing (as in k-mer count models of RDP), yet rich enough for accurate modeling of the data (as in hidden Markov models of HMMER or interpolated context models of Phymm). In the second, classification stage of its pipeline, NASCUP evaluates the likelihood of a test sequence under the context-tree model of each sequence group and chooses the group that maximizes the likelihood.

In both model-building and classification stages, NASCUP utilizes the notion of *universal probability* [20], [21] from information theory. In a nutshell, universal probability approximates all probability distributions in a given class of models, and serves as a close proxy to an unknown true probability distribution of given data without unnecessary overfitting. With theoretical performance guarantee and practical low-complexity implementations, universal probability has found many successful applications, including compression and prediction of sequential data of a priori unknown statistics. NASCUP measures how likely a sequence group fits a context-tree model and how likely a test sequence fits the chosen context-tree model of a sequence group by evaluating the universal probabilities of the sequences. The inference approach based on universal probability in particular and the information-theoretic principle in general has an additional benefit of having no tunable parameter [26] except the maximum depth of the context-tree models.

A. MODEL BUILDING

Given a collection of sequence groups in a database, NASCUP models each sequence group by a CTM of an unknown context set $\mathcal{S}$ and unknown parameters $p(s)$, $s \in \mathcal{S}$. NASCUP estimates the probability of such an unknown CTM by selecting the ''maximum likelihood'' context set $\mathcal{S}^*$ based on the sequences y in the group. Since the parameters are unknown, NASCUP evaluates the universal probability of the *Krichevski–Trofimov (KT)* [12] estimate $Q^{\mathrm{KT}}(y;\mathcal{S})$ for all possible $\mathcal{S}$, and chooses $\mathcal{S}^*$ that attains the maximum. This context tree selection procedure of NASCUP is intimately related to the model selection method of the *minimum description length principle* [27], [28] and other information criteria used in statistics and information theory. The resulting context set $\mathcal{S}^*$ and the count vectors are used for the subsequent classification stage.

The detail of the model-building stage is as follows. NASCUP initially counts the nucleotide symbols A,C,G,T for all sequences in the group that follow each of length-d contexts $s \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}, \star\}^d$ and forms count vectors $\mathbf{n}(s)$. $\star$ implies any possible symbols. These count vectors can be calculated by merging bottom–up from the leaves of the perfect suffix tree (the d-th order Markov model) to the root (the IID model) iteratively. As an example, for $d = 2$, $\mathbf{n}(\star\mathtt{A}) = \mathbf{n}(\mathtt{AA}) + \mathbf{n}(\mathtt{CA}) + \mathbf{n}(\mathtt{GA}) + \mathbf{n}(\mathtt{TA})$ and $\mathbf{n}(\star\star) = \mathbf{n}(\star\mathtt{A}) + \mathbf{n}(\star\mathtt{C}) + \mathbf{n}(\star\mathtt{G}) + \mathbf{n}(\star\mathtt{T})$. NASCUP then finds the

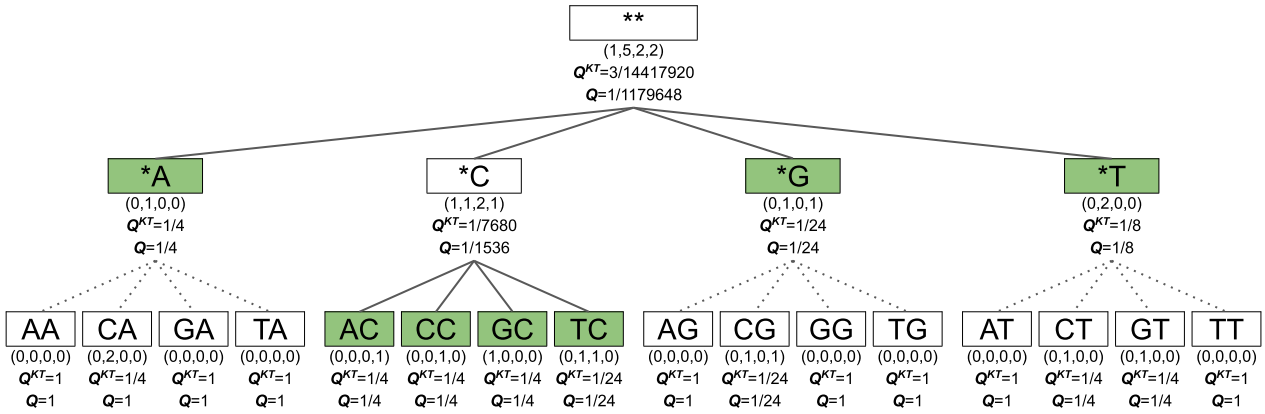

FIGURE 2. Example of a context-tree model. A complete suffix tree consists contexts from every-order Markov model of $\{\star\}^j \times \{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}^{d-j}$ for some j, where $\star$ denotes "any". Leaf nodes consist of $\{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}^d$ and root node consists of $\{\star\}^d$. Parent nodes have one more $\star$ replacing the $\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}$ in child nodes, *e.g.*, $\star\mathtt{C}$ is a parent node of child-nodes $\mathtt{AC}$, $\mathtt{CC}$, $\mathtt{GC}$, and $\mathtt{TC}$. Root node $\star\star$ is a parent node of child-nodes $\star\mathtt{A}$, $\star\mathtt{C}$, $\star\mathtt{G}$, and $\star\mathtt{T}$. CTM has contexts disjointly, *e.g.*, $\star\mathtt{A}$,$\mathtt{AC}$,$\mathtt{CC}$,$\mathtt{GC}$,$\mathtt{TC}$,$\star\mathtt{G}$,$\star\mathtt{T}$.

context set $\mathcal{S}$ that maximizes the KT estimate probability, which can be viewed as a proxy for all unknown CTMs on $\mathcal{S}$. This maximization can be performed efficiently in a recursive manner [29], [30] by starting with the root in the perfect suffix tree in a depth-first manner and computing for each node $s \in \{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T},\star\}^d$ in the tree

$$Q(s) = \max\left\{Q^{\mathrm{KT}}(\mathbf{n}(s)), \prod_{s'\text{: children of } s} Q(s')\right\}.$$

If s is a leaf (at depth d), then $Q(s) = Q^{\mathrm{KT}}(\mathbf{n}(s))$. As an example, for $d = 2$,

$$Q(\star\star) = \max\{Q^{\mathrm{KT}}(\mathbf{n}(\star\star)), Q(\star\mathtt{A})Q(\star\mathtt{C})Q(\star\mathtt{G})Q(\star\mathtt{T})\}$$

and

$$Q(\star\mathtt{A}) = \max\{Q^{\mathrm{KT}}(\mathbf{n}(\star\mathtt{A})), Q^{\mathrm{KT}}(\mathbf{n}(\mathtt{AA}))Q^{\mathrm{KT}}(\mathbf{n}(\mathtt{CA}))Q^{\mathrm{KT}}(\mathbf{n}(\mathtt{GA}))Q^{\mathrm{KT}}(\mathbf{n}(\mathtt{TA}))\}.$$

A branch that does not attain the maximum is pruned. A tie between a parent node and its children is broken against branching for the sparsity of the resulting model. Upon completion of this maximizing and pruning step, at the end of model building for each sequence group, NASCUP produces a valid context set $\mathcal{S}^*$ and symbol count vectors $\mathbf{n}(s)$ for all contexts $s \in \mathcal{S}^*$. **Fig 2** illustrates an example of obtained context tree model by the model building process explained.

B. CLASSIFICATION

The notion of universal probability plays a pivotal role in the classification stage as well. If the true CTM P_j for sequence group j in the database were known, then the maximum likelihood classifier would compute $P_j(x)$ of the query sequence x for all groups j and select the group j^* that maximizes the likelihood. As in the model-building stage, NASCUP relies on the first principle of using the universal probability when the true probability is unknown, and computes the maximum using the universal probability instead of P_j. More concretely, NASCUP first generates count vectors $\mathbf{m}(s)$ from the query sequence x for all contexts $s \in \{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T},\star\}^d$ bottom–up. For each sequence group j with context tree $\mathcal{S}_j^*$ and count vectors $\mathbf{n}(s)$, $s \in \mathcal{S}_j^*$, NASCUP then compute the KT estimate $Q^{\mathrm{KT}}(x|y_j)$ of the conditional probability of the query sequence x given the existing sequences y_j in group j and its count vectors $\mathbf{m}(s)$ by

$$Q^{\mathrm{KT}}(x|y) = \prod_{s\in\mathcal{S}} \frac{Q^{\mathrm{KT}}(\mathbf{m}(s)+\mathbf{n}(s))}{Q^{\mathrm{KT}}(\mathbf{n}(s))}.$$

Assuming that the correct context tree $\mathcal{S}_j^*$ was found, this KT estimate $Q^{\mathrm{KT}}(x|y_j)$ is universal and thus is uniformly close to the true conditional probability $P_j(x|y)$, which is in turn uniformly close to $P_j(x)$ due to the Markov property of the CTM. Hence, we can perform ML classification approximately without knowing the true probability distributions. NASCUP thus compares $Q^{\mathrm{KT}}(x|y_j)$ over all sequence groups j and selects the one with the maximum KT conditional probability. The idea of using universal probability in sequence classification traces back to the information theory literature [31], [32] and NASCUP extends it to CTMs and associated universal probability estimates. Note that the measure, $\log 1/P_j(\mathbf{x}|\mathbf{y}_j)$, can be interpreted as the code length for lossless compression of the sequence [22] under the probability model P_j. In this sense, NASCUP can be viewed as a refinement of the nucleotide sequence classification methods based on compression [33], [34] in that the code length from NASCUP is essentially optimal and NASCUP provides the probability model itself instead of the code length.

C. SCORES, RANKING, AND MULTI-CANDIDATE CLASSIFICATION

For each sequence group $j = 1,\ldots,J$ in the database, NASCUP computes the estimate $Q^{\mathrm{KT}}(x|y_j)$ of the likelihood that the query sequence is generated from group j. This likelihood estimate serves as a score for each group, which can be rank-ordered to form a small list of candidate groups

of the query sequence. Such a candidate list can boost the accuracy of classification, for example, as an input to a slower yet more accurate classifier, or even as a focused target for biological experiments. The simplest approach to forming a list is to sorting all J groups in the database by the likelihood and choosing the top K groups for a fixed number K. Alternatively, the list size K can be adjusted adaptively by estimating the overall accuracy of the list. By the Bayes rule, under the uniform prior on the sequence groups, the posterior probability that the sequence x belongs group j is approximately

$$\frac{Q^{\mathrm{KT}}(x|y_j)}{\sum_{j=1}^{J} Q^{\mathrm{KT}}(x|y_j)}.$$

The sum of these posterior probabilities of the candidates in a list provides an estimate of the classification accuracy, which can be used to control the size of the list. Note that this approach can incorporate an arbitrary prior on the sequence groups.

D. OUTLIER DETECTION

The CTM that NASCUP finds in the model-building stage and the resulting KT estimate of the conditional probability can be utilized beyond classification of query sequences. Suppose that there are a few outliers in a sequence group. Since the symbol counts used by NASCUP reflects the statistical behavior of the entire group, the generated model is rather immune to a small number of outliers and other errors in the data. Once the model is built, we can detect outliers within the group by evaluating the universal probability $Q^{\mathrm{KT}}(x|y)$ trained by all sequences y in the group with each individual sequence x in the group. We measure the degree of conformance to the model of each sequence x of length m by its normalized negative log-likelihood (NLL)

$$\frac{1}{m}\log\frac{1}{Q^{\mathrm{KT}}(x|y)}.$$

The smaller the NLL value is, the better the conformance to the model is. Conversely, the larger the value, the greater the difference from the model, indicating a high likelihood of being an outlier (**Fig 6**).

E. SYNTHETIC SEQUENCE GENERATION

Let d be the depth of the context tree and l be the average length of the sequences y in a sequence group. First, we generate the starting string $\mathtt{X}_1\cdots\mathtt{X}_d$ of length d by copying the most frequent starting string of the same length among the existing sequences in the group. Using a suffix of $\mathtt{X}_1\cdots\mathtt{X}_d$ as the context s, we generate the next symbol $\mathtt{X}_{d+1}$ according to

$$Q^{KT}(\mathtt{X}_{d+1}=\mathtt{X}|\mathtt{X}_1,\ldots,\mathtt{X}_d,y)=\frac{n_{\mathtt{X}}(s)+1/2}{n(s)+2}$$

as in (6). Subsequently, we generate $\mathtt{X}_i$, $i=d+2,\ldots,l$, each according to a similar predictive probability estimate with a suffix of $\mathtt{X}_{i-d}\cdots\mathtt{X}_{i-1}$ as the context and the corresponding count vector from the existing sequences y as well as the preceding symbols $\mathtt{X}_{d+1}\cdots\mathtt{X}_{i-1}$. This sliding-window sequence generation procedure is an extension of Polya's urn process in the standard Bayesian statistics to a CTM. Due to the conjugacy of the Dirichlet prior, the distribution of the generated sequence x (from the $(d+1)$-st position and on) is equivalent to a CTM with *random* parameters $p(s)$ drawn from the Dirichlet prior with parameters $n_{\mathtt{A}}(s)+1/2$, $n_{\mathtt{C}}(s)+1/2$, $n_{\mathtt{G}}(s)+1/2$, $n_{\mathtt{T}}(s)+1/2$ for each $s\in\mathcal{S}$.

IV. EXPERIMENTS

We measured performance of NASCUP and compared it with four main alternative methods (BLAST, HMMER, RDP, and USEARCH) in classification accuracy and classification time (**Fig 3**). We also examined additional methods (Phymm, gzip, UBLAST, caBLAST, BLAT, and three methods provided by QIIME [37] — Naive Bayes in QIIME-2, and UCLUST and Mothur in QIIME-1) (**Supplementary Table 2** and **Supplementary Table 3**). For BLAST and its variants based on sequence alignment (USEARCH, UBLAST, caBLAST, and BLAT), the class of a query sequence was determined by the best hit. For gzip, the class was determined by the smallest difference between the lengths of compressed representations of a sequence group and the group appended by the query sequence.

Experiments were done using a Linux machine (Ubuntu 12.04, 2.2 GHz Intel Xeon E5-4620, and 512 GB memory) without any parallelization. All command scripts of NASCUP and other methods are provided as **Supplementary Table 1**. Source code of NASCUP and the dataset used for experiments are available at https://github.com/nascup/nascup.

A. DATASET

Real sequence datasets from a variety of sources, organized on a functional or taxonomic basis with varying degrees of inter-group similarity (**TABLE 1**) are used to validate the classification accuracy and efficiency of NASCUP: a function-based RNA family database (Rfam [3]), taxonomy-based rRNA databases (RDP [9], [35], Greengenes [15], and SILVA-SSU/LSU [16]), and pyrosequencing databases (Artificial/Divergent [36]).

We excluded sequences containing symbols other than `ACGT(U)` and sequence groups of size less than ten for 10-fold cross-validation. We compacted Greengenes and SILVA with cd-hit-est [38] by 97% similarity and limited the number of sequences in a group to 2,000. The datasets thus obtained had diverse characteristics: the number of groups from 23 to 1,320, the sequence length from 20 to almost 5,000, and the average normalized intra-group pairwise sequence distance from 0.08 to 0.33.

B. CLASSIFICATION ACCURACY AND COMPUTATION SPEED

NASCUP achieves superb performance with respect to accuracy and speed among the five classification methods

TABLE 1. Details of the dataset used in the experiments.

	Category	ID	Dataset	AIGD‡	Number of Groups *	Number of Sequences †	Sequence length	Ground truth
Functional Non-coding RNA		RF	Rfam v11.0 [3]	0.33	1,320	170,881	20–1,875	Accession
Microbial Taxonomy	rRNA (16S, 18S, 23S/28S)	RD	RDP v10.0 [9], [35]	0.08	134	3,838	320–1,833	Taxonomy (Genus Level)
		GG	Greengenes v13.5 [15]	0.12	464	23,142	1,254–2,146	
		SS	SILVA-SSU v119.1 [16]	0.15	313	17,625	902–3,749	
		SL	SILVA-LSU v119 [16]	0.21	107	4,593	1,900–4,954	
	Pyrosequencing (16S rRNA)	AR	Artificial [36]	0.18	60	44,407	40–294	Reference Sequences
		DV	Divergent [36]	0.14	23	55,466	38–521	

* the number of groups with more than 10 preprocessed sequences
† the total number of sequences after the preprocessing
‡ average intra-group distance (the normalized pairwise distance between the sequences within a group), ranging from 0 to 1. The AIGD being close to zero means that the group consists of similar sequences.

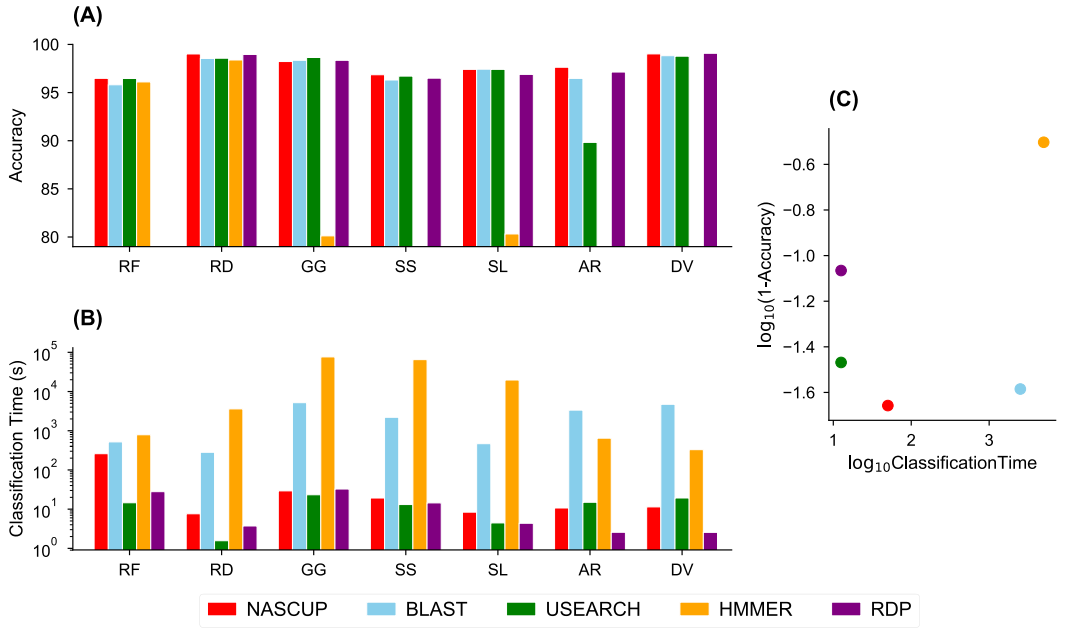

FIGURE 3. Performance comparision of NASCUP and alternatives. (A) Classification accuracy and (B) classification time in log scale of NASCUP and the four main alternatives (BLAST, USEARCH, HMMER, and RDP) on seven sequence datasets. Each value is the average of 10-fold cross validation. (C) 2-D plot to assess methods by comparing accuracy and speed simultaneously. A method closer to the left bottom corner is the better.

compared, consistently across the diverse set of seven datasets (**Fig 3 (C)**).

In terms of accuracy, NASCUP achieved the highest average accuracy of 97.8% (in terms of both arithmetic and geometric means) among an expanded collection of thirteen alternative classification methods, which is trailed slightly by the BLAST-based classification method (**Fig 3 (A)** and **Supplementary Table 2**). In particular, NASCUP showed the highest accuracy on RF dataset, which was the most difficult dataset to classify due to its largest number of classes and widest intra-group sequence distance (AIGD). More than half of the thirteen classification methods (UBLAST, BLAT, caBLAST, RDP, gzip, UCLUST, and Mothur) exhibited unsatisfactory results of below 80% of accuracy on RF. NASCUP and BLAST maintained the accuracy of above 95% across all datasets considered, whereas the performance of the other methods varied often significantly from dataset to dataset.

Except for NASCUP and BLAST, the classification accuracy varied significantly over the datasets. In particular, HMMER was accurate on the functional RNA datasets but not on metagenomic microbial datasets, while RDP worked well on microbial datasets but showed unsatisfactory results on functional RNAs.

NASCUP, RDP, USEARCH, and QIIME package based methods ran significantly (often by orders of magnitude) faster for most of the datasets (**Fig 3 (B)** and **Supplementary Table 3**).

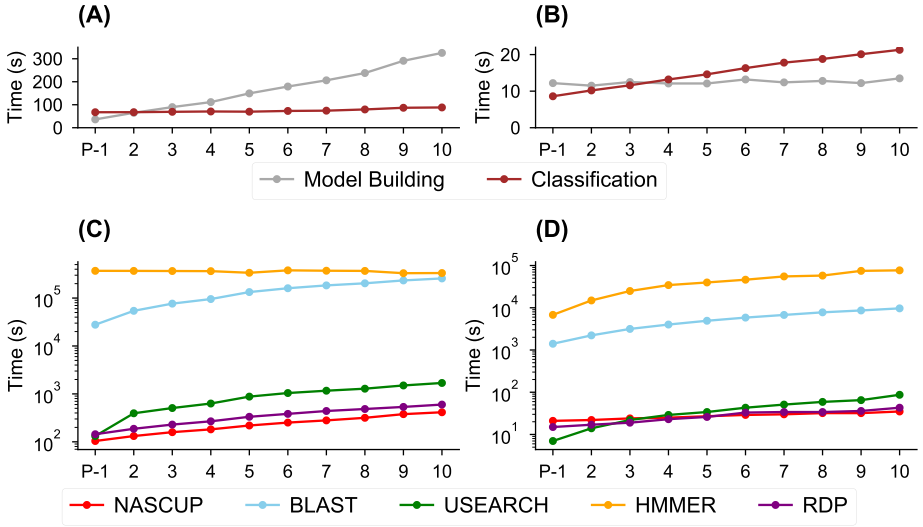

FIGURE 4. **Time comparision to check scalability Model building and classification time of NASCUP for (A) sequencewise and (B) groupwise expanding datasets. Total runtime of NASCUP and the four main alternatives on (C) sequencewise and (D) groupwise expanding datasets.**

Statistical significance test (Wilcoxon signed-rank test) was performed in the SciPy library on normalized and unnormalized aggregate datasets as well as individual datasets (**Supplementary Table 4**). The normalized aggregate dataset was formed by combining the seven datasets in **TABLE 1**. In order to balance the impacts of the individual datasets on the aggregate, we drew bootstrap samples of the same size from each. The sample size of 2,900 was used, which is one tenth of the average size of the component datasets excluding the smallest (RD) and the largest (RF). We used the averaged p-value over 100 signed-rank tests repeated on independently generated bootstrap samples. For the unnormalized aggregate dataset, we used all samples in the seven datasets without normalization of their sizes. All pairs of the five main classification methods (NASCUP, BLAST, USEARCH, HMMER, RDP) were compared for statistical significance.

C. SCALABILITY

To emulate the usage of classification tools for a realistic environment in which sequence databases scale over time, we prepared two types of artificial expanding datasets—one by increasing the number of sequences for a fixed number of groups and the other by increasing the number of groups for a fixed number of sequences per group.

For sequencewise expansion, the model building (first stage) time of NASCUP grew linearly as the number of sequences increased. The actual classification time, however, was affected only marginally since the second-stage classification operation is almost independent of the number of sequences in a group once the modeling has been completed (**Fig 4 (A)**). The total runtime of NASCUP (the sum of modeling and classification times) was lower than the other four classification methods regardless of the data size (**Fig 4 (C)**).

For groupwise expansion, the classification time of NASCUP grew as the number of groups (as well as the total number of sequences) increased. The modeling time did not increase since the model building procedure had to be performed only for newly added groups (**Fig 4 (B)**). The performance of NASCUP was among the top under groupwise expansion, especially for very large databases (**Fig 4 (D)**).

In both sequencewise and groupwise expansion experiments, NASCUP was orders-of-magnitude faster than BLAST, the only method that achieved a comparable level of accuracy, across all database sizes.

D. ROBUSTNESS TO MISLABELING ERRORS

We tested the robustness of NASCUP against mislabeling errors in the sequence database, which, in principle, exists in any real dataset labeled in the absence of the ground truth and especially in pyrogenetic datasets. We prepared a dataset with classification errors at a rate ranging from 1% to 20% and compared the classification accuracy of five alternative methods (**Fig 5**). The accuracy of NASCUP was robust with marginal performance degradation even when 20% of the sequences in the database were mislabeled to arbitrary groups. RDP exhibited a similar level of robustness, whereas the performance of the other three methods degraded as the error rate increased.

E. SCORING

NASCUP computes the likelihood of a test sequence belonging to each sequence group. This numerical likelihood value provides additional soft information that can augment the hard classification outcome. As the simplest application of such likelihood values, we evaluated how the accuracy of NASCUP improves when it produces a ranked list of likely groups, instead of a single most likely group, for a given test sequence. By increasing the list size, NASCUP achieved

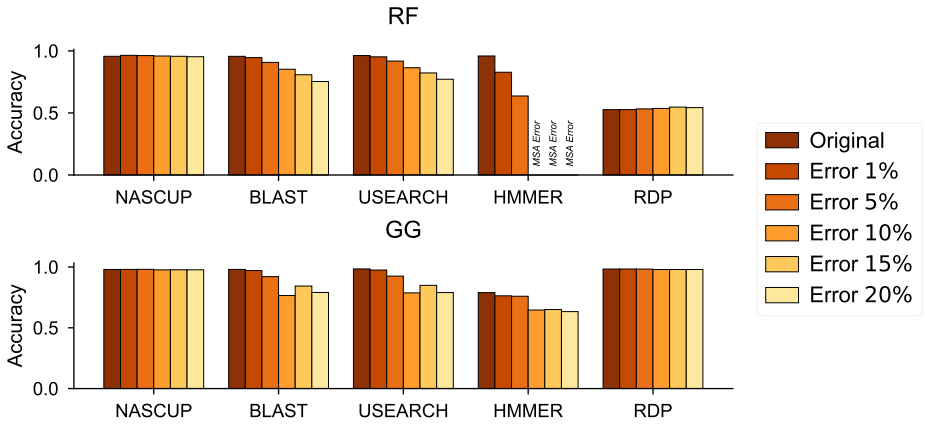

FIGURE 5. **Performance degradation from mislabeling errors. Classification accuracy was measured when a fraction of database sequences in RF and GG datasets were mislabeled. The mislabeling error rate increased from 1% to 20% and the bar color gradually faded as the error rate increased. HMMER was not able to build models on RF datasets of higher mislabeling error rate, because the multiple sequence alignment (MSA) could not be performed properly due to no consensus columns from too diverse sequences.**

TABLE 2. **Accuracy comparison from candidate extraction. Two sections represent different extracting methods about extracting candidate groups as varying (A) number of candidates, fixed to *K* and (B) threshold *T* on the sum of the posterior probabilities, respectively. Each value in the seven datasets is the average accuracy of 10-fold cross-validation. In the second section, the numbers in parentheses mean the average number of candidate groups.**

Method / Data	RF	RD	GG	SS	SL	AR	DV	Arith. Mean	Geom. Mean
NASCUP	**96.5%**	**99.0%**	98.2%	**96.9%**	97.4%	**97.6%**	99.0%	**97.8%**	**97.8%**
BLAST	95.8%	98.5%	98.3%	96.3%	**97.4%**	96.5%	98.8%	97.4%	97.4%
USEARCH	96.5%	98.6%	**98.6%**	96.7%	97.4%	89.8%	98.8%	96.6%	96.6%
UBLAST	79.9%	98.5%	97.9%	95.9%	97.1%	96.4%	98.7%	94.9%	94.7%
BLAT	79.1%	97.2%	92.1%	92.3%	95.2%	94.7%	98.9%	92.8%	92.6%
caBLAST	39.4%	97.1%	86.9%	90.5%	93.9%	95.5%	97.0%	85.8%	82.6%
UCLUST	23.7%	97.0%	94.4%	85.4%	72.6%	96.8%	98.8%	81.3%	74.6%
Mothur	52.2%	99.0%	98.2%	96.3%	96.8%	95.1%	99.0%	90.9%	89.1%
HMMER	96.1%	98.4%	80.1%	14.9%	80.3%	46.3%	64.0%	68.6%	59.7%
RDP	52.6%	99.0%	98.3%	96.5%	96.9%	97.1%	**99.1%**	91.4%	89.5%
QIIME2	83.3%	98.7%	97.1%	94.2%	95.6%	97.0%	99.0%	95.0%	94.8%
Phymm	93.6%	77.5%	76.7%	39.5%	93.2%	95.0%	98.9%	82.1%	79.0%
gzip	62.7%	96.3%	90.3%	80.1%	77.6%	80.9%	96.3%	83.5%	82.7%

near-perfect accuracy (**TABLE 2 (A)**). In particular, the ten most likely sequence groups produced by NASCUP included the correct group over 99% of the time for all datasets. Instead of producing a fixed number of candidate groups, NASCUP can produce a variable-size list according to the target accuracy, which can be estimated by the likelihood values and the Bayes rule (**TABLE 2 (B)**). The list of top candidate groups can be fed into subsequent bioinformatics pipelines (such as cross-validation by BLAST) or actual biological experiments on a far reduced set of groups than before preprocessed by NASCUP.

F. OUTLIER DETECTION

As another application of the likelihood value computed by NASCUP, we performed one-class classification that identifies whether a test sequence belongs to a single target sequence group or not. For NASCUP and HMMER, both of which build generative models for the target group and provide the likelihood values of the test sequence. We used normalized negative log likelihood value to distinguish member sequences in the target group from outliers. NASCUP showed a clear separation of outliers, which manifested in the much larger area under the precision–recall curve (**Fig 6**).

G. SYNTHETIC SEQUENCE GENERATION

To ascertain the quality of synthetic sequences in our procedure, we generated synthetic sequences from RF, RD, and GG datasets, one per group, and classified them using BLAST (**Fig 7**). The combination of context-tree models and universal probability in NASCUP finds statistical generative models of nucleotide sequence groups that are parsimonious and consequently are expected to better represent the ground truth by Occam's razor (**Fig 8** and **Fig 9**). In order to demonstrate the interpretive power of such generative models, we generated synthetic sequences randomly according to four generative models—a combination of context-tree vs. Markov models and universal vs. maximum-likelihood probabilities—and measured how often these sequences looked real by

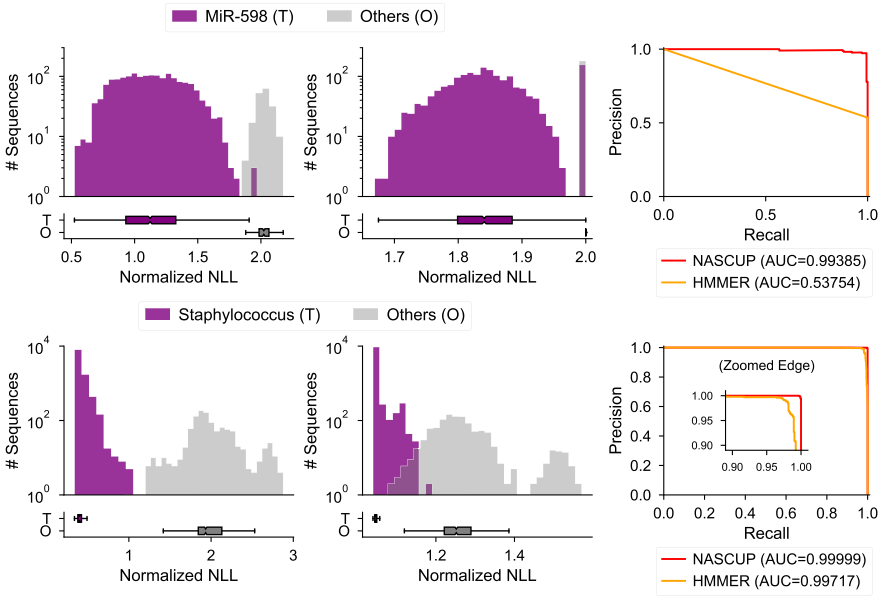

FIGURE 6. **Outlier detection with normalized NLL. Distribution of predicted normalized negative log-likelihood (NLL) values with histogram and boxplot of NASCUP (left column) and HMMER (center column). Precision-recall (PR) curves and their area under the curve (AUC) values of NASCUP and HMMER are drawn in the right column. The top three plots are experimental results on staphylococcus mixed with outliers from GG dataset, and the bottom three plots are results on miR-598 mixed with outliers from RF dataset. The datasets were mixed with 10:1 ratio of sequences from targets and others (outliers). The sequences of outliers were randomly taken from each group except the target group. The number of sequences in the staphylococcus and others were 10,390 and 1,039, respectively. The number of sequences in miR-598 and others were 1,800 and 180, respectively.**

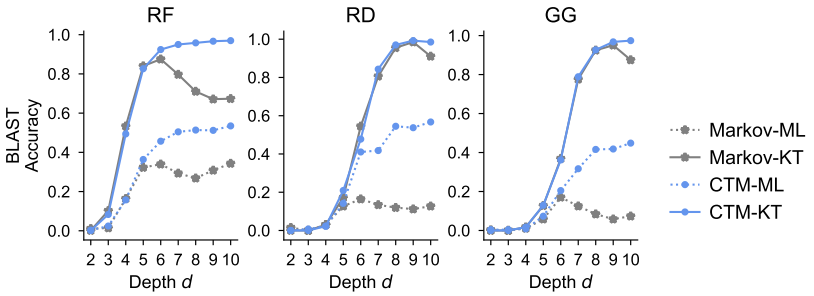

FIGURE 7. **Quality assessment of synthetic sequence generation. For three datasets of RF, RD, GG, a total of respectively 1,320, 134, and 464 synthetic sequences (one synthetic sequence per group) were generated and classified by the combination of the modeling methods (Markov vs. CTM) and sampling methods (universal probability (KT) and maximum likelihood (ML)). The depth varied from 2 to 10. NASCUP and Markov models are expressed in blue and gray colors, respectively. KT and ML estimators are represented by a solid line and dotted line, respectively.**

classifying them using BLAST. As the maximum context size increased, BLAST could classify the synthetic sequences generated by NASCUP (context-tree models and universal probability) correctly with very high accuracy, while the synthetic sequences generated by the other three models failed to emulate real sequences due to inadequate modeling and overfitting (**Fig 7**). Similar trends were observed when BLAST was replaced by other classification methods.

H. NASCUP DESIGN ALTERNATIVES

Universal probability and context-tree models are two key features of NASCUP. In order to demonstrate their combined benefit, we compared classification accuracy among a few design alternatives by varying the model depth (**Fig 8**). As an alternative to a CTM, a Markov model provides a baseline before pruning context trees based on recursive likelihood maximization. For probability assignment, the maximum likelihood (ML) estimator provides a naive alternative to the KT estimator in assigning the conditional probability in each context.

Since the number of possible contexts grows exponentially as the model depth increases, the occurrence of each context becomes sparse with a restricted number of training data. Consequently, the Markov model and ML estimator are

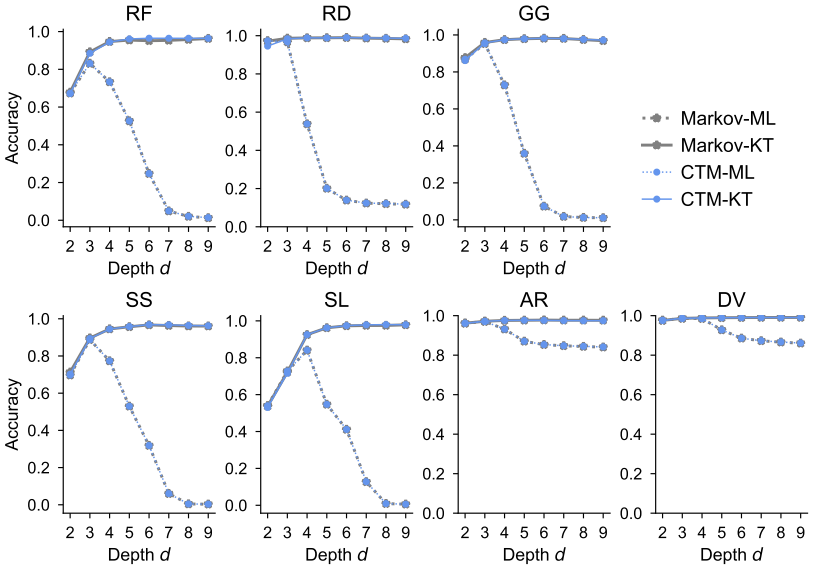

FIGURE 8. **Accuracy of NASCUP variants.** For each dataset, we examined classification accuracy for the combination of modeling methods (Markov vs. CTM) and probability estimators (ML vs. KT). The depth varied from 2 to 9. NASCUP and Markov models are expressed in blue and gray colors, respectively. KT and ML estimators are represented by a solid line and dotted line, respectively. NASCUP (CTM–KT) performed consistently better than other combinations, without any performance degradation when depth becomes too large.

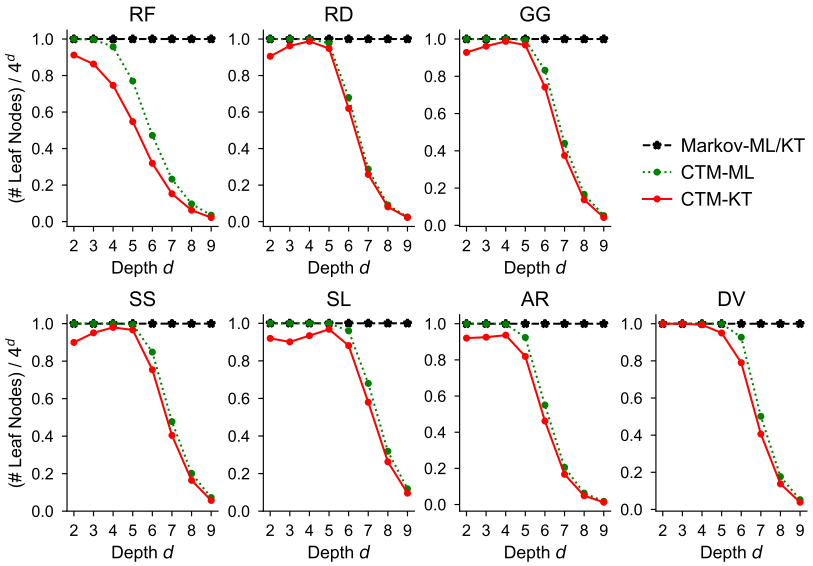

FIGURE 9. **The ratio of the numbers of leaf nodes.** The number of leaf nodes in a CTM was much smaller than that of the corresponding Markov model, which demonstrates that CTMs find a sparse and meaningful structure of the sequence groups. For each dataset, the ratio of the average number of leaf nodes of the CTM selected in the model-building stage to the number 4^d of all leaf nodes in a Markov model is plotted as depth changes. For model building with CTMs, both KT and ML probability estimators were tested, represented by blue dotted and red dashed lines, respectively.

prone to overfitting in general. In our accuracy comparison experiments, both Markov and context-tree models with ML estimator performed rather poorly. While the performance advantage of CTMs over Markov models was not very pronounced, the numbers of leaves in CTMs did not increase exponentially as those in Markov models and their ratios dropped rapidly as the depth increased (**Fig 9**). Moreover, CTMs using KT estimator always have fewer leaf nodes than CTMs using ML estimator. In conclusion, the combination of KT and CTM used in NASCUP was the most parsimonious. This sparsity can be interpreted as being closer to the ground truth in principle (which was cross-examined by the synthetic sequence generation experiment), but also leads to faster and more efficient classification as a practical benefit.

V. DISCUSSION

As a model-based method, NASCUP can be easily adapted to test similarity between families or a membership of a sequence or a sequence family in a group of families. We expect that this modified version of NASCUP can be applied to problems with inherent hierarchical structures such as taxonomy classification and phylogenetic tree construction. A phylogenetic tree is typically constructed by measuring sequence similarities using heuristics such as progressive alignment and then by applying linkage methods to produce a dendrogram structure. In this conventional approach, results may vary (often dramatically) according to which linkage method is used. In contrast, the modified version of NASCUP would allow direct model-to-model comparison, alleviating the aforementioned limitations of alignment–linkage methods. Going one step further, we expect that the modified NASCUP, fully utilizing the simple Bayesian update structure of universal probability assignments in the classification stage, will be applicable to incremental clustering, in which an initial model is constructed from close sequences and then is incrementally augmented by adding additional sequences.

Instead of unidirectional contexts (that precede symbols), we can use bidirectional contexts (that precede and succeed symbols). More generally, the directionality of nucleotide sequences or the lack thereof can be incorporated into modeling and classification stages. Additionally, the maximum depth of context-tree models can be adjusted more flexibly in a data-dependent manner. A similar technique has been developed in the context of data compression [39], which is expected to be applicable in our problem. The use of bidirectional contexts and adaptive maximal depths is expected to boost accuracy further at some cost of time efficiency.

ACKNOWLEDGMENT

(Sunyoung Kwon and Gyuwan Kim contributed equally to this work.)

REFERENCES

[1] M. H. van Regenmortel, C. M. Fauquet, D. H. Bishop, E. Carstens, M. Estes, S. Lemon, J. Maniloff, M. Mayo, D. McGeoch, and C. Pringle, *Virus Taxonomy: Classification and Nomenclature of Viruses: Seventh Report of the International Committee on Taxonomy of Viruses*. New York, NY, USA: Academic, 2000.

[2] J. Lu, E. A. Miska, E. Alvarez-Saavedra, J. Lamb, D. Peck, A. Sweet-Cordero, B. L. Ebert, R. H. Mak, and A. A. Ferrando, "Microrna expression profiles classify human cancers," *Nature*, vol. 435, no. 7043, pp. 834–838, Jun. 2005.

[3] S. Griffiths-Jones, A. Bateman, M. Marshall, A. Khanna, and S. R. Eddy, "Rfam: An RNA family database," *Nucleic Acids Res.*, vol. 31, no. 1, pp. 439–441, Jan. 2003.

[4] P. Simmonds, J. Kolberg, and M. Urdea, "Classification of hepatitis C virus into six major genotypes and a series of subtypes by phylogenetic analysis of the NS-5 region," *J. Gen. Virol.*, vol. 74, no. 11, pp. 2391–2399, Nov. 1993.

[5] R. L. Tatusov, D. A. Natale, I. V. Garkavtsev, T. A. Tatusova, U. T. Shankavaram, B. S. Rao, B. Kiryutin, M. Y. Galperin, N. D. Fedorova, and E. V. Koonin, "The COG database: New developments in phylogenetic classification of proteins from complete genomes," *Nucleic Acids Res.*, vol. 29, no. 1, pp. 22–28, Jan. 2001.

[6] G. Cochrane, B. Alako, C. Amid, L. Bower, A. Cerdeño-Tárraga, I. Cleland, R. Gibson, N. Goodgame, M. Jang, and S. Kay, "Facing growth in the European nucleotide archive," *Nucleic Acids Res.*, vol. 41, no. D1, pp. D30–D35, Nov. 2012.

[7] A. C. Normand, A. Packeu, C. Cassagne, M. Hendrickx, S. Ranque, and R. Piarroux, "Nucleotide sequence database comparison for routine dermatophyte identification by internal transcribed spacer 2 genetic region DNA barcoding," *J. Clin. Microbiol.*, vol. 56, no. 5, May 2018, Art. no. e00046.

[8] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *J. Molecular Biol.*, vol. 215, no. 3, pp. 403–410, Oct. 1990.

[9] J. R. Cole, Q. Wang, E. Cardenas, J. Fish, B. Chai, R. J. Farris, A. S. Kulam-Syed-Mohideen, D. M. McGarrell, T. Marsh, G. M. Garrity, and J. M. Tiedje, "The ribosomal database project: Improved alignments and new tools for rRNA analysis," *Nucleic Acids Res.*, vol. 37, pp. D141–D145, Jan. 2009.

[10] T. J. Wheeler and S. R. Eddy, "Nhmmer: DNA homology search with profile HMMs," *Bioinformatics*, vol. 29, no. 19, pp. 2487–2489, Oct. 2013.

[11] A. Brady and S. L. Salzberg, "Phymm and PhymmBL: Metagenomic phylogenetic classification with interpolated Markov models," *Nature Methods*, vol. 6, no. 9, pp. 673–676, Sep. 2009.

[12] R. Krichevsky and V. Trofimov, "The performance of universal encoding," *IEEE Trans. Inf. Theory*, vol. 27, no. 2, pp. 199–207, Mar. 1981.

[13] J. Rissanen, "Stochastic complexity and modeling," *Ann. Statist.*, vol. 14, no. 3, pp. 1080–1100, Sep. 1986.

[14] B. S. Clarke and A. R. Barron, "Jeffreys' prior is asymptotically least favorable under entropy risk," *J. Stat. Planning Inference*, vol. 41, no. 1, pp. 37–60, Aug. 1994.

[15] T. Z. DeSantis, P. Hugenholtz, N. Larsen, M. Rojas, E. L. Brodie, K. Keller, T. Huber, D. Dalevi, P. Hu, and G. L. Andersen, "Greengenes, a chimera-checked 16S rRNA gene database and workbench compatible with ARB," *Appl. Environ. Microbiol.*, vol. 72, no. 7, pp. 5069–5072, Jul. 2006.

[16] C. Quast, E. Pruesse, P. Yilmaz, J. Gerken, T. Schweer, P. Yarza, J. Peplies, and F. O. Glöckner, "The SILVA ribosomal RNA gene database project: Improved data processing and web-based tools," *Nucleic Acids Res.*, vol. 41, no. D1, pp. D590–D596, Nov. 2012.

[17] F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens, "The context-tree weighting method: Basic properties," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 653–664, May 1995.

[18] P. Buhlmann and A. J. Wyner, "Variable length Markov chains," *Ann. Statist.*, vol. 27, no. 2, pp. 480–513, 1999.

[19] D. Ron, Y. Singer, and N. Tishby, "The power of amnesia: Learning probabilistic automata with variable memory length," *Mach. Learn.*, vol. 25, nos. 2–3, pp. 117–149, 1997.

[20] P. Algoet, "Universal schemes for prediction, gambling and portfolio selection," *Ann. Probab.*, vol. 20, no. 2, pp. 901–941, 1992.

[21] J. Jiao, H. H. Permuter, L. Zhao, Y.-H. Kim, and T. Weissman, "Universal estimation of directed information," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6220–6242, Oct. 2013.

[22] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. New York, NY, USA: Wiley, 2006.

[23] J. Rissanen, "Universal coding, information, prediction, and estimation," *IEEE Trans. Inf. Theory*, vol. 30, no. 4, pp. 629–636, Jul. 1984.

[24] Y. M. Shtar'kov, "Universal sequential coding of single messages," *Problemy Peredachi Informatsii*, vol. 23, no. 3, pp. 3–17, 1987.

[25] G. Bejerano and G. Yona, "Variations on probabilistic suffix trees: Statistical modeling and prediction of protein families," *Bioinformatics*, vol. 17, no. 1, pp. 23–43, Jan. 2001.

[26] E. Keogh, S. Lonardi, and C. A. Ratanamahatana, "Towards parameter-free data mining," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2004, pp. 206–215.

[27] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, no. 5, pp. 465–471, 1978.

[28] J. R. Quinlan and R. L. Rivest, "Inferring decision trees using the minimum description length principle," *Inf. Comput.*, vol. 80, no. 3, pp. 227–248, Mar. 1989.

[29] F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens, "Context weighting: General finite context sources," in *Proc. 14th Symp. Inf. Theory Benelux*, Veldhoven, The Netherlands, May 1993, pp. 120–127.

[30] R. Nohre, "Some topics in descriptive complexity," Ph.D. dissertation, Linköping Univ., Linköping, Sweden, 1994.

[31] J. Ziv, "On classification with empirically observed statistics and universal data compression," *IEEE Trans. Inf. Theory*, vol. 34, no. 2, pp. 278–286, Mar. 1988.

[32] M. Gutman, "Asymptotically optimal classification for multiple tests with empirically observed statistics," *IEEE Trans. Inf. Theory*, vol. 35, no. 2, pp. 401–408, Mar. 1989.

[33] D. Loewenstern, H. Hirsh, P. Yianilos, and M. Noordewier, "DNA sequence classification using compression-based induction," DIMACS, Tech. Rep. 95-04, 1995.
[34] A. Kocsor, A. Kertesz-Farkas, L. Kajan, and S. Pongor, "Application of compression-based distance measures to protein sequence classification: A methodological study," *Bioinformatics*, vol. 22, no. 4, pp. 407–412, Feb. 2006.
[35] P. D. Schloss, S. L. Westcott, T. Ryabin, J. R. Hall, M. Hartmann, E. B. Hollister, R. A. Lesniewski, B. B. Oakley, D. H. Parks, C. J. Robinson, J. W. Sahl, B. Stres, G. G. Thallinger, D. J. Van Horn, and C. F. Weber, "Introducing mothur: Open-source, platform-independent, community-supported software for describing and comparing microbial communities," *Appl. Environ. Microbiol.*, vol. 75, no. 23, pp. 7537–7541, Dec. 2009.
[36] C. Quince, A. Lanzén, T. P. Curtis, R. J. Davenport, N. Hall, I. M. Head, L. F. Read, and W. T. Sloan, "Accurate determination of microbial diversity from 454 pyrosequencing data," *Nature Methods*, vol. 6, no. 9, pp. 639–641, Sep. 2009.
[37] J. G. Caporaso *et al.*, "QIIME allows analysis of high-throughput community sequencing data," *Nature Methods*, vol. 7, no. 5, pp. 335–336, 2010.
[38] W. Li and A. Godzik, "Cd-hit: A fast program for clustering and comparing large sets of protein or nucleotide sequences," *Bioinformatics*, vol. 22, no. 13, pp. 1658–1659, 2006.
[39] F. M. J. Willems, "The context-tree weighting method: Extensions," *IEEE Trans. Inf. Theory*, vol. 44, no. 2, pp. 792–798, Mar. 1998.

SUNYOUNG KWON (Member, IEEE) received the B.S. degree in computer science from Pusan National University, South Korea, in 2001, and the M.S. degree in bioinformatics and the Ph.D. degree in electrical and computer engineering from Seoul National University, South Korea, in 2014 and 2018, respectively. From 2018 to 2020, she was with Naver Corporation, South Korea. Previously, she was with Korea Communication Agency and LG Electronics Inc., South Korea. She is currently an Assistant Professor with the School of Biomedical Convergence Engineering, Pusan National University. Her research interests include AI, bioinformatics, drug discovery, graph neural networks, machine learning, and big-data analytics.

GYUWAN KIM received the B.S. degree in electrical and computer engineering, the B.S. degree in mathematical sciences, and the M.S. degree in electrical and computer engineering from Seoul National University, South Korea, in 2014 and 2017, respectively. He is currently pursuing the Ph.D. degree with the Department of Computer Science, University of California, Santa Barbara. Previously, he was a Research Scientist at NAVER Clova & AI LAB. His research interests include machine learning and natural language processing.

BYUNGHAN LEE (Member, IEEE) received the B.S. degree in electrical engineering from Korea University, South Korea, in 2011, and the Ph.D. degree in electrical and computer engineering from Seoul National University, South Korea, in 2018. He is currently an Assistant Professor with the Department of Electronic and IT Media Engineering, Seoul National University of Science and Technology. His research interests include machine learning, artificial intelligence, and their biomedical applications.

JONGSIK CHUN received the B.S. degree from Seoul National University, Seoul, South Korea, and the Ph.D. degree from the Newcastle University School of Medicine. He completed his postdoctoral work at the Research Center for Molecular Microbiology, Seoul National University. He is currently a Professor of biology with Seoul National University. He serves on Seoul National University's Interdisciplinary Program in Bioinformatics Steering Committee. He is associated with a number of institutes at Seoul National University, including the Institute of Microbiology Research, the Genetic Engineering Combustion Institute, and the International Vaccine Institute. He also assists the Rural Development Administration and the Institute of Agriculture and Life Sciences. He has previously served as a Research Associate at the Center of Marine Biotechnology, University of Maryland, and a Senior Researcher at the Korea Research Institute of Bioscience and Biotechnology. He was an Associate Editor of the *International Journal of Systematic and Evolutionary Microbiology*. He serves as an Editorial Board Member for *Antonie van Leeuwenhoek* (Dutch Kluwer four issues) and *Microbes and Environments*.

SUNGROH YOON (Senior Member, IEEE) received the B.S. degree in electrical engineering from Seoul National University, South Korea, in 1996, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA, in 2002 and 2006, respectively. From 2006 to 2007, he was with Intel Corporation, Santa Clara. From 2007 to 2012, he was an Assistant Professor with the School of Electrical Engineering, Korea University. From 2016 to 2017, he was a Visiting Scholar with the Department of Neurology and Neurological Sciences, Stanford University. He held research positions at Stanford University and Synopsys, Inc., Mountain View. He is currently a Professor with the Department of Electrical and Computer Engineering, Seoul National University. His current research interests include machine learning and artificial intelligence. He was a recipient of the SNU Education Award, in 2018, the IBM Faculty Award, in 2018, the Korean Government Researcher of the Month Award, in 2018, the BRIC Best Research of the Year, in 2018, the IMIA Best Paper Award, in 2017, the Microsoft Collaborative Research Grant, in 2017 and 2020, the SBS Foundation Award, in 2016, the IEEE Young IT Engineer Award, in 2013, and many other prestigious awards. Since February 2020, he has been serving as the Chairperson (minister-level position) of the Presidential Committee on the Fourth Industrial Revolution established by the Korean Government.

YOUNG-HAN KIM (Fellow, IEEE) received the B.S. degree (Hons.) in electrical engineering from Seoul National University, Seoul, South Korea, in 1996, and the M.S. degree in electrical engineering, the M.S. degree in statistics, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, USA, in 2001, 2006, and 2006, respectively. In 2006, he joined the University of California at San Diego, La Jolla, CA, USA, where he is currently a Professor with the Department of Electrical and Computer Engineering. He has coauthored the book *Network Information Theory* (Cambridge University Press, 2011) and the monograph *Fundamentals of Index Coding* (Now Publishers, 2018). His current research interests include information theory, data science, machine learning, and communication engineering. He was a recipient of the 2008 NSF Faculty Early Career Development Award, the 2009 U.S.–Israel Binational Science Foundation Bergmann Memorial Award, the 2012 IEEE Information Theory Paper Award, and the 2015 IEEE Information Theory Society James L. Massey Research and Teaching Award for Young Scholars. He served as an Associate Editor for the IEEE TRANSACTIONS ON INFORMATION THEORY and a Distinguished Lecturer for the IEEE Information Theory Society.