Journal of Bioinformatics and Computational Biology
Vol. 18, No. 5 (2020) 2050026 (22 pages)

DOI: 10.1142/S0219720020500262

World Scientific
www.worldscientific.com

Scalable classification of organisms into a taxonomy using hierarchical supervised learners

Gihad N. Sohsah[*,‡], Ali Reza Ibrahimzada[*,§], Huzeyfe Ayaz[*,¶] and Ali Cakmak[†,‖]

[*]*Department of Computer Science*
Istanbul Sehir University, Istanbul, Turkey

[†]*Department of Computer Engineering*
Istanbul Technical University, Istanbul, Turkey
[‡]*gihadsohsah@std.sehir.edu.tr*
[§]*aliibrahimzada@std.sehir.edu.tr*
[¶]*huzeyfeayaz@std.sehir.edu.tr*
[‖]*ali.cakmak@gmail.com*

Received 22 August 2019
Accepted 1 June 2020
Published 29 October 2020

Accurately identifying organisms based on their partially available genetic material is an important task to explore the phylogenetic diversity in an environment. Specific fragments in the DNA sequence of a living organism have been defined as DNA barcodes and can be used as markers to identify species efficiently and effectively. The existing DNA barcode-based classification approaches suffer from three major issues: (i) most of them assume that the classification is done within a given taxonomic class and/or input sequences are pre-aligned, (ii) highly performing classifiers, such as SVM, cannot scale to large taxonomies due to high memory requirements, (iii) mutations and noise in input DNA sequences greatly reduce the taxonomic classification score. In order to address these issues, we propose a multi-level hierarchical classifier framework to automatically assign taxonomy labels to DNA sequences. We utilize an alignment-free approach called spectrum kernel method for feature extraction. We build a proof-of-concept hierarchical classifier with two levels, and evaluated it on real DNA sequence data from barcode of life data systems. We demonstrate that the proposed framework provides higher f1-score than regular classifiers. Besides, hierarchical framework scales better to large datasets enabling researchers to employ classifiers with high classification performance and high memory requirement on large datasets. Furthermore, we show that the proposed framework is more robust to mutations and noise in sequence data than the non-hierarchical classifiers.

Keywords: Taxonomies; classification; scalability; supervised learning.

1. Introduction

Classification of living organisms is a key problem in both biology and computer science. Using traditional morphological keys for classification is often efficient only

‖Corresponding author.

for a particular gender or life stage. Besides, this method is slow and expensive, as it requires the time and effort of highly experienced taxonomists.

DNA barcoding has gained significant attention in the scientific community[1,2] after it was first introduced.[3] Specific gene regions have been chosen as markers that can distinguish between different species.[4–6] For animal groups, cytochrome c oxidase 1 gene (COI) is used as a barcode, while matK and rbcL are used for identifying land plants, and ITS is used for fungi.[7] The problem then translates into classifying barcodes to a known species in a fast and efficient way.[8]

There are a number of methods that tackle with the DNA barcode-based classification problem using the tools of sequence comparison and alignment.[9,10] However, aligning multiple sequences in an optimal way is computationally costly. Using alignment-free kernel methods has been proven efficient for this problem.[11] In this approach, the occurrences of each possible fixed-length substring are counted in each DNA barcode sequence. These substrings are called k-mers, where k is an integer parameter that corresponds to the length of the substring. These k-mers are then used as features with their corresponding count per sequence as feature values.

Several machine learning classification techniques are proposed and compared to determine species given a DNA barcode.[9] More specifically, support vector machines (SVMs),[12] the rule-based method RIPPER,[13] the decision tree C4.5,[14] and the Naïve Bayes are considered.[15] A major drawback in such studies is that only pre-aligned sequences are considered. Besides, the classification is strictly performed within the scope of specific taxonomic classes like bats, birds, fungi, or fishes. Hence, in order to classify a given barcode sequence, it first needs to be aligned, and one also needs to know to which taxonomic class the given sequence belongs to.

Besides, phylogenetic and statistical classification methods are also studied together.[16] More specifically, neighbor joining and PHYML are studied as phylogenetic methods,[17,18] and k-nearest neighbor, classification and regression trees (CART), random forests (RFs), and SVM are evaluated under statistical classification methods. However, a limitation is that *a priori* knowledge about the genus of the sequence is assumed to be available and employed sequences are pre-aligned. Another direction is to exploit a supervised machine learning approach that selects suitable nucleotide positions and then compute the logic formulas for species classification.[10] Nevertheless, the input DNA barcode sequences are required to be pre-aligned.

In later studies, the requirements for pre-aligned sequences are removed. As an example, k-mers are employed for DNA barcode classification and analytics.[11] In particular, 10-mer features are exploited to train classification models using two classes of algorithms: nearest neighbor and SVM.

Another alignment-free approach employs a new set of classification features that are based on covariance of nucleotides in DNA barcodes.[19] The computed features are later exploited in a RF classifier to perform phylogenetic analysis on a particular fungi species.

In addition to real datasets, synthetic datasets are also considered and compared within the domain of DNA barcode-based classification.[20] To this end, different classifiers including (i) simple logistic function,[21] (ii) IBk from lazy classifier,[22] (iii) PART from rule-based classifier,[23] (iv) RF from tree-based classifier,[24] (v) attribute selected classifier, and (vi) bagging from meta classifiers are benchmarked.[25]

In most of the existing studies that focus on the problem of organism classification using DNA sequences, the classification is mainly performed within a specific taxonomic class assuming *a priori* knowledge about the given to-be-classified sequence. This assumption may not always hold true, e.g. when inspecting fossil remains or sequences extracted from mud-samples and earth layers. In such situations, it is hard to identify whether these sequences belong to the class of bats, birds, rodents, fishes, etc. Furthermore, highly performing classifiers, such as SVM, cannot scale to large taxonomies due to high memory requirements. Besides, mutations and noise in input DNA sequences greatly reduce the taxonomic classification.

In order to address the above issues, in this paper, we introduce a hierarchical framework that can be extended into one hierarchical classifier capable of classifying any DNA barcode sequence without any *a priori* knowledge about its taxonomic tree. This framework utilizes support vector classifiers in order to build a two-stage classifier that can predict the species given the DNA barcode sequence only, without the need to compute any sequence alignment. Our framework enables leveraging the strength of the support vector classifiers while overcoming the scalability issues that arise when the number of classes increases or when the data matrix size grows.

In order to establish a proof-of-concept, we test the proposed approach on five different datasets obtained from Barcode of Life Data (BOLD) systems[26]: (i) aves (i.e. birds), (ii) chiroptera (i.e. bats), (iii) rodentia (i.e. rodents), (iv) polypodiopsida (a member of vascular plants), and (v) pucciniomycetes (a member of fungi). For each dataset, the classification performance of different classifiers using varying subsequence lengths are compared. We observe that the SVM classifier with a linear kernel outperforms all the other methods for larger subsequence lengths. The RF classifier, on the other hand, outperforms the SVM-based classifiers when the subsequence length is relatively small. Then, we merge the five datasets to examine the scalability of each classification method. For larger datasets, SVM was not a feasible solution, since the data matrix was not representable. Nevertheless, the RF method did not experience such problems, and provided a reasonable accuracy (f1-score: 90.8%). In order to overcome this scalability drawback and utilize the high classification performance of the linear kernel SVM, we build a hierarchical SVM-based classifier, and demonstrate that it outperforms the non-hierarchical regular classifier (93.0% versus 90.8%). Besides, we also study the robustness of the proposed method by introducing artificial mutations to the sequences with increasing ratios. Our experimental results show that as mutations rates increase, the proposed hierarchical classifier framework exhibits more robustness than the other non-hierarchical classifiers.

Our contributions in this paper are as follows:

- We propose a multi-level hierarchical DNA sequence classification framework, and build a proof-of-concept instance with two taxonomic levels. The proposed framework can be extended to predict the species within larger scopes that go beyond just the taxonomic class level. More specifically, it can be used as a blueprint in building a full supervised classifier that can classify all life forms.
- We demonstrate that the hierarchical classifier framework classifies DNA sequences with higher f1-score than the regular stand-alone classifiers.
- The proposed framework allows taking advantage of SVM's high accuracy prediction power for larger datasets as well by increasing its scalability with multi-level architecture. While regular SVM-based classifiers run out of memory when trained on a large dataset on a decently configured test environment, the hierarchical SVM-based classifier successfully runs on the same test hardware and dataset.
- We demonstrate that with the hierarchical classifier framework, the robustness of classification in the presence of mutations and/or noise in sequence data is higher than the regular non-hierarchical classifiers.

2. Methods

2.1. *Kernel-based alignment-free method for feature extraction*

Kernel-based methods are employed to represent sequences with variable lengths and also to avoid the burden of handling insertions and deletions. Kernel-based methods have been proven to be efficient in a number of similar tasks like protein–protein interaction prediction and protein classification.[27–29] They have been also demonstrated to be effective in tackling the problem of species classification using DNA barcodes.[11] In this method, sequences are represented as collections of short substring kernels of length k. These substrings are called k-mers. Figure 1 illustrates how a sequence can be represented as a vector of k-mers frequencies, where $k = 5$.

The number of k-mers increases exponentially with k. Since we have four bases (i.e. A, C, G, T), the number of all k-mers is 4^k. The occurrence frequency of these k-mers is then used as features. A variation of this method employs mismatch-kernels for feature extraction.[29,30] In this case, at most m, mismatches are allowed within a substring. This can enhance the results of the classification task by making the data

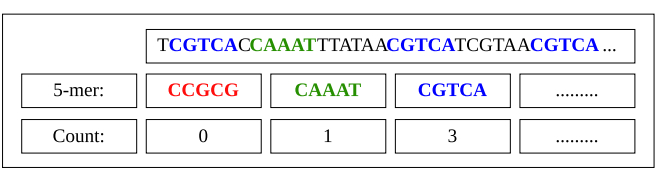

Fig. 1. An example of how 5-mer kernels are used to represent a DNA barcode sequence.

Algorithm 1: k-mer Generation Algorithm

Input: sequence data S and a number k
Output: k-mers along with their frequencies

```
1  create an empty frequence dictionary, freq
2  create all k-mer combinations of A, G, C, and T.
3  insert all generated k-mer combinations into freq with frequency value 0.
4  Loop for each DNA sample D in S:
5      if length(D) <657 then
6          continue
7      end
8      i ← 1
9      while i <= length(D)-k+1 do
10         k-mer ← D[i : i + k]
11         i ←i+1
12         if k-mer contains nucleotides other than A, G, C, T then
13             continue
14         end
15         freq[k-mer] ← freq[k-mer] + 1
16     end
17 end
18 return freq
```

Fig. 2. *K*-mer generation algorithm.

matrix denser which is desirable for most of the classification algorithms. However, in this paper, we do not study the effect of allowing mismatches and the effect of changing the value of m. Instead, we focus on building a hierarchical classifier that allows the classification tasks to be more efficiently performed on large datasets that include different taxonomic classes.

In order to generate k-mers from raw datasets, we use the algorithm shown in Fig. 2. The inputs are the sequence data from BOLD systems and a number k that represents the length of substring kernels. We first create an empty frequency dictionary, and then populate it with all possible k-mer combinations of A, G, C and T along with initial frequency value of 0 (lines 1–3). Next, we loop over the BOLD systems dataset until there are no samples left (lines 4–17). In each iteration, we skip a DNA string if its length is smaller than 657, since the full length of COI segment used as a DNA barcode is 657 bp (lines 5–7).[31] Then, in a sliding window manner, we consider each k-mer in the sequence (lines 9–16). We skip a k-mer if it contains ambiguous characters like "-" or "N" (lines 12–14). Then, we increment the appearance frequency of each qualifying k-mer by 1 (line 15). At the end, the set of all k-mers along with their frequencies is returned (line 18).

2.2. *Scalable supervised learning*

In most of the related works, it has been shown that it is possible to train a supervised classifier that has the ability to predict the species given the DNA barcode sequence. However, there were two factors that kept the effectiveness of such approaches restricted. First, these studies carried out the prediction effort within a specific taxonomic organism rank, e.g. performing the experiments on the taxonomic class level such as chiroptera, rodentia, aves, mammalia, etc.,[9,11] or on the taxonomic genus level as in Ref. 16. Hence, they assume the availability of *a priori* knowledge of the taxonomic class or genus to which a specimen or a sequence belong to. Such an assumption may not hold true in many cases such as when inspecting samples from a lake or soil.

Second, among the supervised classification algorithms, SVMs are commonly employed in taxonomy classification, as it provides more accurate results than many other methods. However, SVM suffers from scalability issues when the number of classes in the dataset increases, as the data matrix grows in size. All these reasons hinder its use as an efficient classification algorithm to train a classifier that predicts the species directly from the DNA barcode sequence. As, in that case, the number of classes would be the number of all known species, and the dataset would be all the data samples available on BOLD systems. Motivated by the above observations, in this paper, we propose a two-stage hierarchical classifier inspired by the hierarchical nature of the taxonomy tree. The first stage predicts the taxonomic class. Then, according to the prediction of the first-stage classifier, the feature vector representing a given DNA barcode sequence is directed to the corresponding classifier trained to predict the species within that taxonomic class. A diagram of this framework is shown in Fig. 3.

The illustrated framework is used to train a classifier capable of predicting the species name for a given DNA barcode sequence out of 1400 species appeared in the datasets used in this work. Although the results were obtained for 1400 species that belong to five different taxonomic classes (aves, rodentia, chiroptera, polypodiopsida,

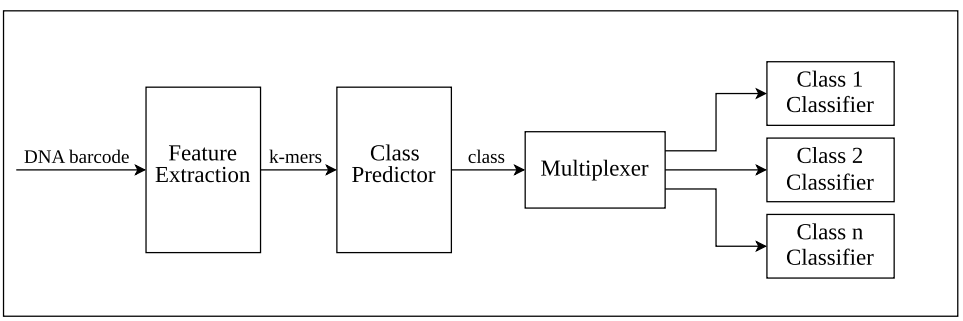

Fig. 3. Two-stage hierarchical classifier for predicting the species without any *a priori* knowledge about its taxonomic class.

Algorithm 2: Hierarchical Classification Algorithm

Data: k-merized data along with their frequencies
Output: species predictions for all samples
1 **Loop** until the end of stratified 10-Fold:
2 Set the train and test portions of the dataset
3 Train a classification model at class level on the training split
4 **Loop** for each taxonomic class:
5 train a classification model at species level based on training split
6 **end**
7 **Loop** over each sample in the test split:
8 $X \leftarrow$ *the predicted class for the current sample*
9 $Y \leftarrow$ *the species-level model for class X that is built in line 5*
10 run Y to predict the species for the current sample
11 save species prediction
12 **end**
13 **end**
14 **return** the set of species predictions for each sample in the dataset

Fig. 4. Hierarchical classification algorithm.

and pucciniomycetes), the extension of this work into one hierarchical learner capable of classifying all known living things is straightforward.

The algorithm in Fig. 4 provides a more detailed description of the proposed hierarchical classification method in a 10-fold cross-validation setting. For each fold of the dataset that is split in a stratified manner, we first determine the indices of train and test portions of the dataset (line 2). Then, on the current training split, we build a classification model for the first level to predict the class that a given sequence belongs to (line 3). Then, the classifiers for the second level are built. More specifically, in a loop, a separate species classifier for each taxonomic class is built on the current training split (lines 4–6). In the next stage, using the above-created classification models, predictions are computed for the sequences in the test split (lines 7–12). In particular, for each sequence in the test split, we first run the class-level model to predict the class (line 8). Next, we employ the species-level classifier that corresponds to the predicted class by the first-level classifier to predict the species for the test sequence (lines 9–10). Finally, the set of the predicted species for each sample is returned at the end (line 14).

3. Results

In this section, the proposed framework is experimentally assessed in terms of f1-score, scalability, and robustness. We also compare it to regular non-hierarchical approaches. As a proof-of-concept, two popular supervised learning classifiers are

contrasted, namely, SVM and RFs. We study how different classification methods perform and how varying k affects their performance. In Sec. 3.1, the performance within the scope of a taxonomic class level is evaluated using five different datasets (rodentia, aves, chiroptera, polypodiopsida, and pucciniomycetes). Then, the robustness aspects of the classifiers are studied in the presence of mutations or noise in data. Finally, a scalability analysis of the proposed hierarchical classification is performed in comparison with the non-hierarchical classification.

For each experiment, the models are evaluated using 10-folds cross-validation by repeating the dataset randomization, splitting, training, and testing steps 10 times, and the resulting f1-scores are averaged. Moreover, in order to make sure that class and species distributions match in both training and test splits, stratified sampling is applied during 10-fold cross-validation. Here, we report the averages over all runs.

All the experiments for this paper were carried out on a DELL R720 server whose specifications are 24 core vCPU, 80 GB RAM and 2.4 TB storage. All the scripts are coded in Python using Scikit — Learn machine learning library to implement SVM and RF classifiers. Besides, Matplotlib and Seaborn are used for visualization.[32,33]

3.1. *Datasets*

For this study, the datasets are obtained from BOLD systems which is an initiative to support the generation and application of DNA barcode data.[26] It contains 8,132,361 DNA barcode sequences for animals, plants, fungi, and protists. The specimen is collected from different sites by different organizations worldwide. Through the portal of BOLD systems, data for different life forms may be downloaded in various formats including XML and tab-separated text.

In this paper, five datasets were used: chiroptera, aves, rodentia, polypodiopsida, and pucciniomycetes datasets. All major organism kingdoms (i.e. animals, plants, and microbes) are represented in the dataset. As a preprocessing step, all the sequences that are less than 657 in length were removed since the full length of the COI segment used as a DNA barcode sequence is 657 bp.[31] However, sequences with ambiguous letters like "Ns" and dashes "-" were kept. Table 1 presents a summary of datasets after the preprocessing step. Besides, the maximum, minimum, and average frequencies are calculated and reported in Table 2.

Figures 5–9 show the percentage of removed samples for species in each dataset. In particular, the total number of removed samples from chiroptera is 5 which are

Table 1. Class datasets summary.

Dataset	No. of species	No. of samples
Chiroptera	122	4731
Rodentia	127	3653
Aves	841	4192
Pucciniomycetes	34	1905
Polypodiopsida	276	5850

Table 2. Class datasets species frequencies summary.

Dataset	Maximum frequency	Minimum frequency	Average frequency
Chiroptera	0.2	0.0002	0.008
Rodentia	0.1	0.0002	0.008
Aves	0.02	0.0002	0.001
Pucciniomycetes	0.43	0.001	0.03
Polypodiopsida	0.09	0.0002	0.004

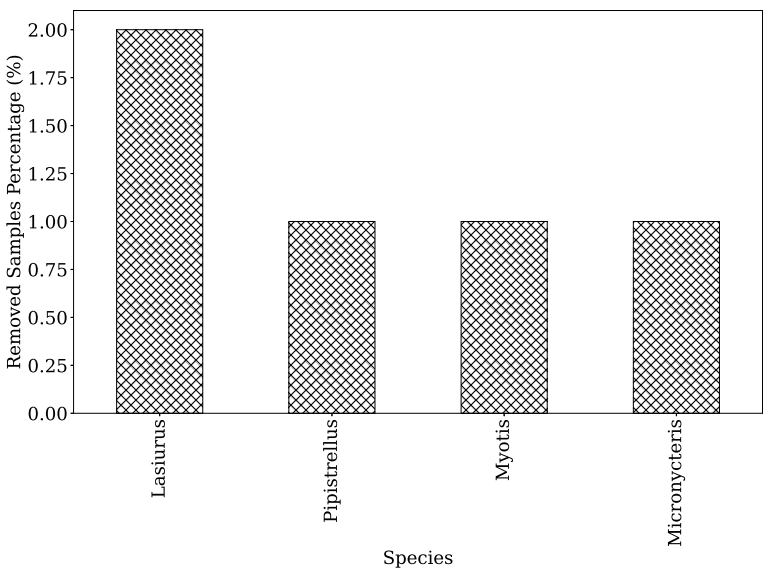

Fig. 5. The percentage of removed samples for species in chiroptera dataset.

distributed over four unique species, as shown in Fig. 5. No species is completely deleted from chiroptera.

The total number of removed samples from aves is 1035 which are distributed over 516 unique species. No species is completely deleted from aves, as shown in Fig. 6. The total number of removed samples from rodentia is 482 which are distributed over 38 species. No species is completely deleted from rodentia, as shown in Fig. 7.

The total number of removed samples from pucciniomycetes is 1001 which are distributed over 30 unique species. Ten species (i.e. insolibasidium, septobasidium, zaghouania, helicobasidium, platygloea, cumminsiella, batistopsora, aecidium, eocronartium and auriculoscypha) are completely removed from pucciniomycetes, as shown in Fig. 8.

The total number of removed samples from polypodiopsida is 945 which are distributed over 82 unique species. No species is completely deleted from polypodiopsida, as shown in Fig. 9.

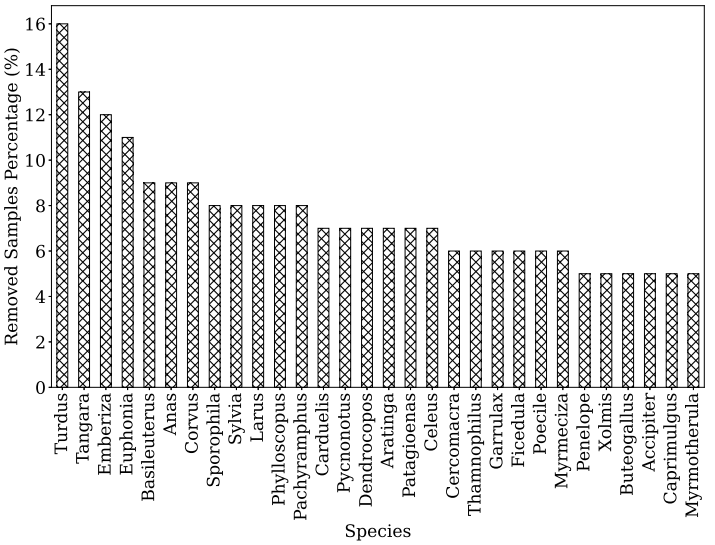

Fig. 6. The percentage of removed samples for top-30 species with the highest removal rate in aves dataset.

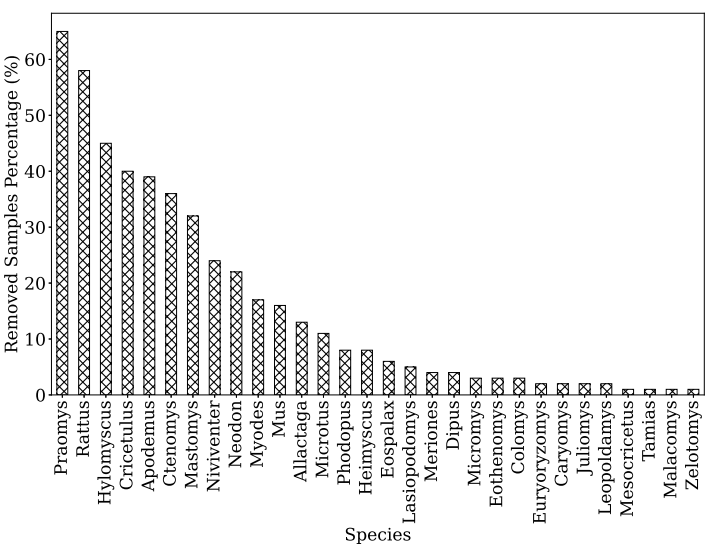

Fig. 7. The percentage of removed samples for top-30 species with the highest removal rate in rodentia dataset.

3.2. *The evaluation of non-hierarchical classifiers*

In this part of our experiments, the effect of changing the length of the subsequence kernel k-mers on the f1-scores of non-hierarchical classifiers is studied. To this end, on each of the datasets, three different classifiers (i.e. RF with 10 estimators, SVM with

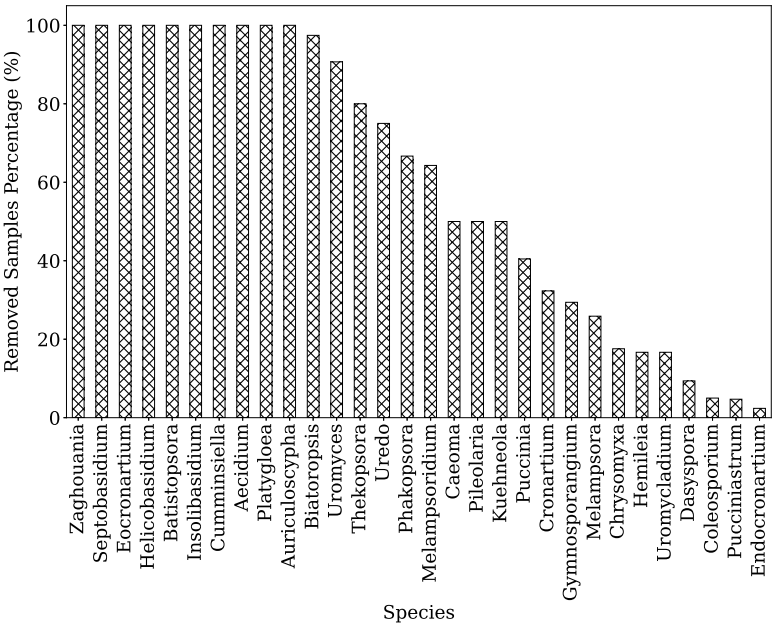

Fig. 8. The percentage of removed samples for top-30 species with the highest removal rate in pucciniomycetes dataset.

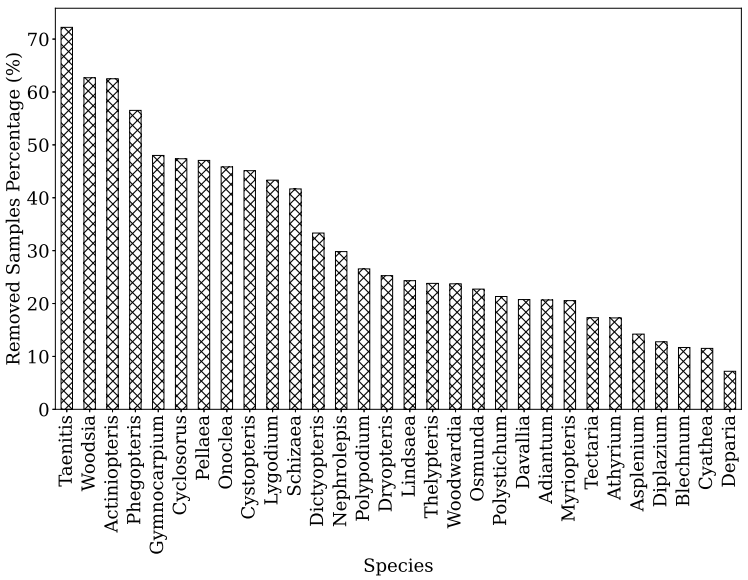

Fig. 9. The percentage of removed samples for top-30 species with the highest removal rate in polypodiopsida dataset.

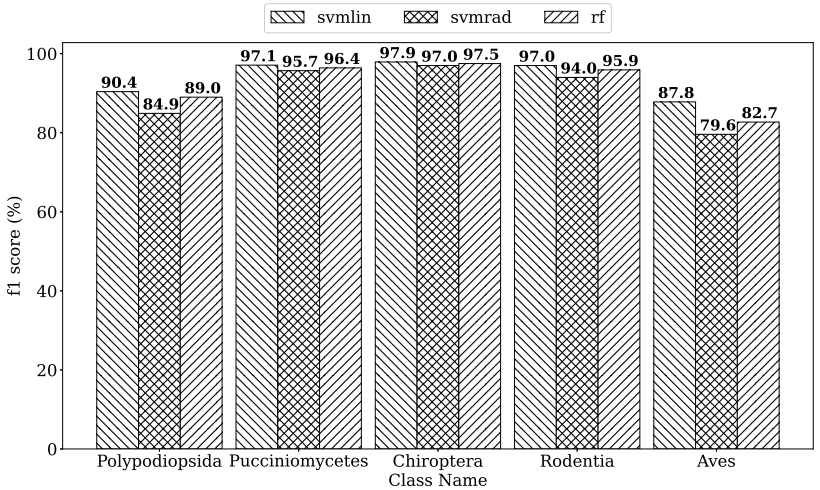

Fig. 10. Maximum mean f1-score of test folds from different models.

linear kernel, and SVM with radial kernel) are trained and tested using 10-folds cross-validation.

Figure 10 presents a summary of how each classification algorithm performs on different datasets. In all experiments, for SVM with radial kernel, the kernel width is set to the number of samples in the training split at each iteration. In all charts, boxplots are also included in order to demonstrate the variability of f1-scores across different folds during cross-validation. Please note that in order to prevent the overlap among box plots as well as the original f1-score lines, box plots are slightly shifted so that they do not block each other in the visualization.

The results for the chiroptera dataset are visualized in Fig. 11. The SVM classifier with a linear kernel, for larger k values, outperforms the other classification methods, while the RF classifier performs better for lower k values. It can also be noted that the f1-scores obtained with the SVM classifier with a linear kernel increase as k increases in the range $[1, 7]$ without any drop, unlike the SVM classifier with a radial kernel which experiences f1-score drop for larger values of k.

Similar observations are made for the rodentia dataset, as illustrated in Fig. 12.

As for the aves dataset, although the test f1-score is lower than the other studied datasets (see Fig. 10), the relative rank and behavior of the classifiers are similar to the above results with the change of k, as shown in Fig. 13.

Similar observations are made for the pucciniomycetes and polypodiopsida datasets, as illustrated in Figs. 14 and 15.

It can be concluded that as long as the memory resources allow larger values of k, it is possible to train an SVM classifier with a linear kernel that achieves better classification scores than both an SVM classifier with a radial kernel and an

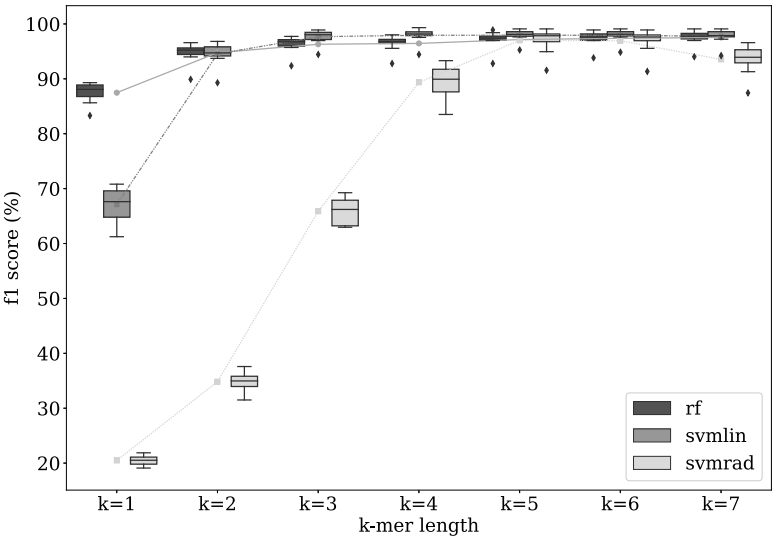

Fig. 11. The effect of changing the value of k on the classification f1-scores for the chiroptera dataset.

RF classifier. On the other hand, if the memory limitations hindered the increase of k, one may opt for RF classifier, as RF requires less memory.

To sum up, the best f1-scores for all datasets are provided by an SVM classifier with a linear kernel trained with subsequence length $k = 7$. As Fig. 10 shows, the

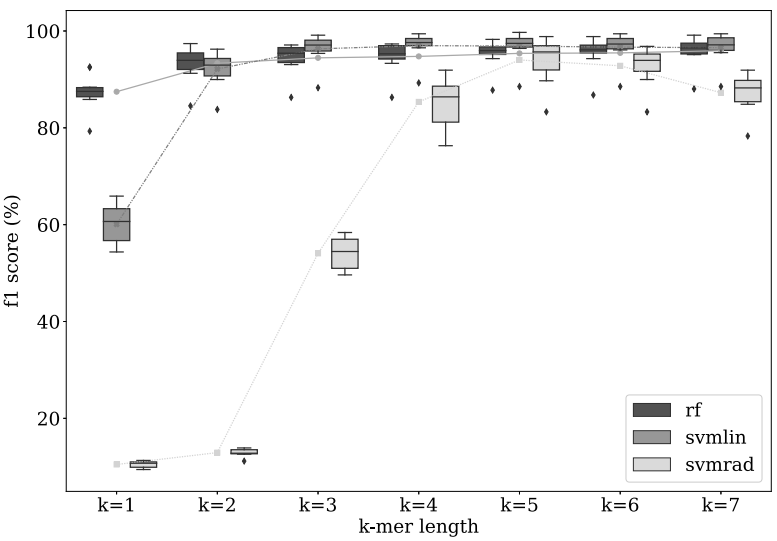

Fig. 12. The effect of changing the value of k on the classification f1-scores for the Rodentia dataset.

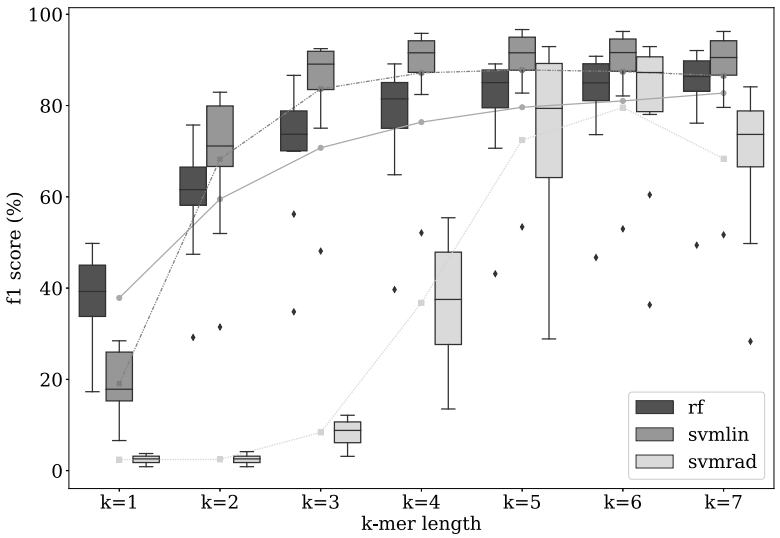

Fig. 13. The effect of changing the value of k on the classification f1-scores for the aves dataset.

classification f1-score for the aves dataset is comparably low. The reason behind this is mainly the insufficient number of samples per species. As summarized in Table 1, the average number of samples per species in aves dataset is significantly lower than that of the other datasets.

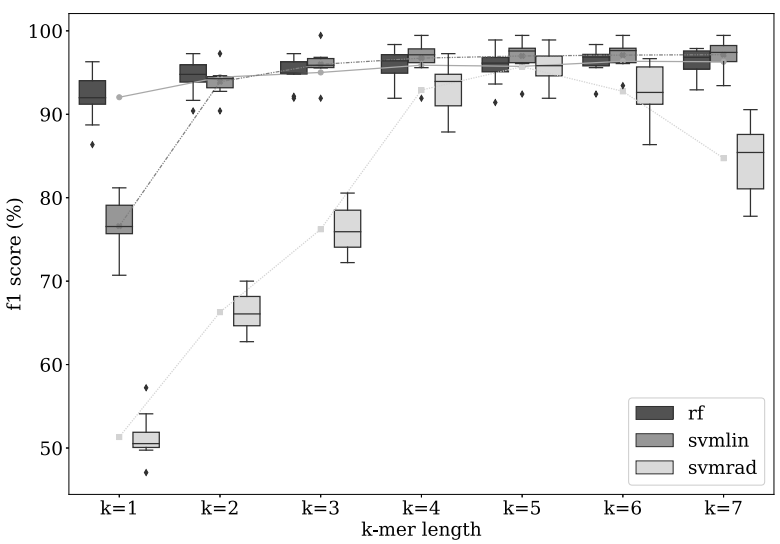

Fig. 14. The effect of changing the value of k on the classification f1-scores for the Pucciniomycetes dataset.

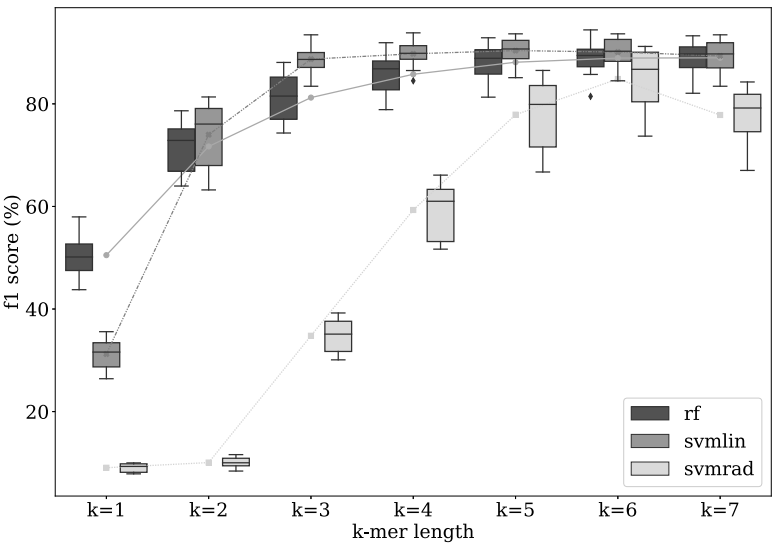

Fig. 15. The effect of changing the value of k on the classification f1-scores for the Polypodiopsida dataset.

As consistent with our results in this section, the previous studies also reported a similar observation that the classification performance improves as the value of k increases in all datasets, and the percentage of improvement significantly drops for the higher values of k.[11] Besides, similarly, in our results, birds (i.e. aves) have the highest error rate as reported earlier in the literature.[11] Finally, SVM's better performance is also in line with the previously published results.[9,16] As an example, in Ref. 16, the method that obtained the best score varied according to the dataset. However, the support vector machine method attained the best score at three out of six datasets while each of the other methods achieved the best score for at most two datasets.

3.3. *Scalability of non-hierarchical classifiers*

In order to study how efficiently different classification methods scale to larger datasets, the five datasets used in this paper, chiroptera, rodentia, aves, pucciniomycetes, and polypodiopsida datasets, are merged into a single dataset. Then, the above three classifiers are trained using the same settings used in the previous Sec. 3.2. Unfortunately, the attempts to train the SVM classifiers (with both linear and radial kernels) failed due to memory limitations, despite the decent memory size of the test machine. However, the RF classifier was trained successfully and provided the results, as illustrated in Fig. 16. The maximum test score (91.1% approximately) was obtained at $k = 7$.

In order to be able to leverage the strength of SVM classifiers, while overcoming their scalability issues, we employ our proposed hierarchical framework as demonstrated next.

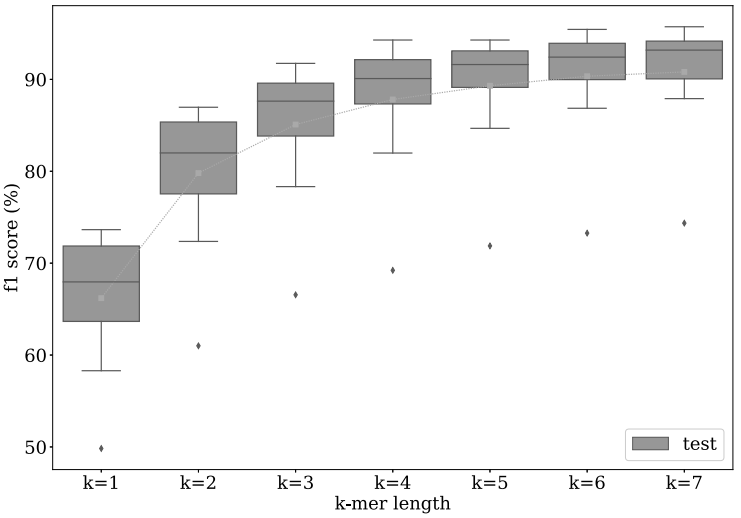

Fig. 16. The effect of changing the value of k on the classification f1-score of a RF classifier with 10 estimators trained and tested on all the five datasets merged together.

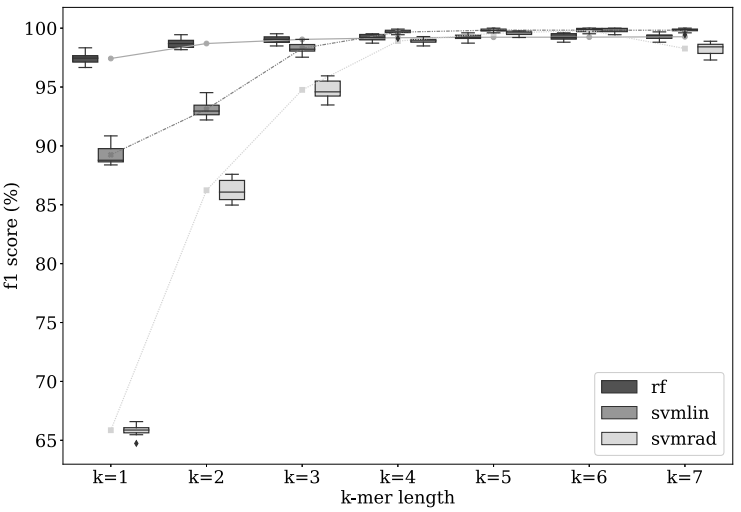

Fig. 17. The effect of changing the value of k on the f1-scores of three different classification methods trained to predict the taxonomic class on the merged dataset.

3.4. *Evaluation of the hierarchical classification framework*

3.4.1. *Taxonomic class predictor*

The first stage of the proposed framework (see Fig. 3) involves the training of a taxonomic class predictor. Figure 17 presents the effect of changing the value of k on the f1-scores of three different classification methods (RF with 10 estimators, SVM

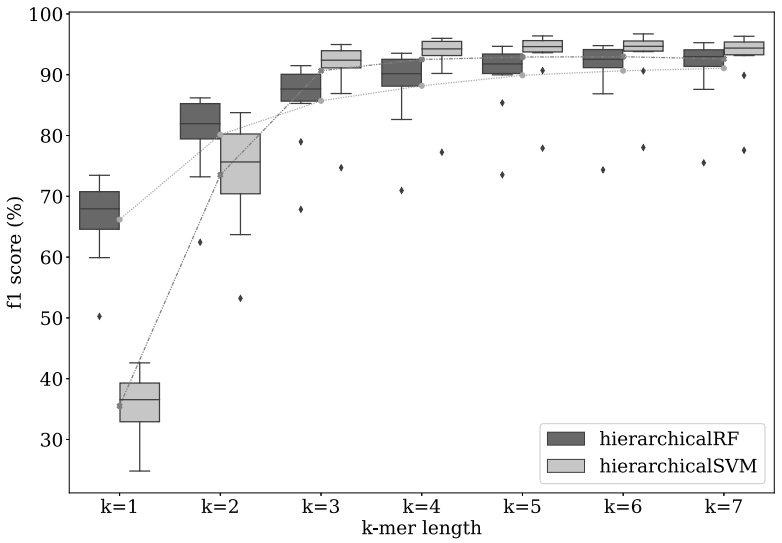

Fig. 18. The effect of changing the value of k on the f1-scores of the hierarchical classifier that employs linear kernel SVM and RF subclassifiers.

with linear kernel, and SVM with radial kernel) trained to predict the taxonomic class on the merged dataset (that involves chiroptera, aves, rodentia, polypodiopsida, and pucciniomycetes datasets).

From Fig. 17, we observe that the best f1-scores are obtained when using SVM classifier with a linear kernel and setting the subsequence length k to be 5. The test f1-score in this case is 99.9% which means that we are able to predict the taxonomic class with an error of 0.1%, and then pass the sequence to a class-based classifier capable of predicting the species with the f1-scores given in Fig. 19.

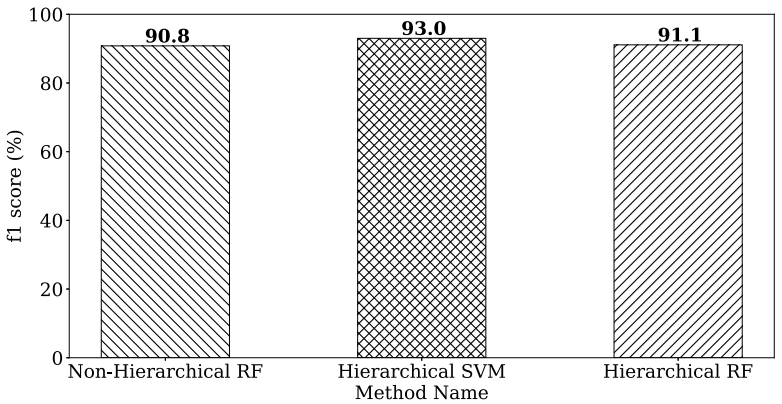

Fig. 19. Comparison between hierarchical and non-hierarchical models based on their testing f1-score.

In the hierarchical classifier, two SVM classifiers with linear kernels are combined in a hierarchical manner. The first one assigns a taxonomic class to each DNA barcode sequence, and passes the sequence down to the second level species classifier to assign a species. The framework for the hierarchical method is illustrated in methods section. SVM classifiers with linear kernels are chosen due to their relatively high performance as shown with the above experimental results. Figure 18 shows how the performance of the two-stage SVM-hierarchical classifier and RF-hierarchical classifier change with the value of k. Figure 19 also compares the f1-scores of the non-hierarchical classifier (with the RF classifier) trained on the merged dataset (with $k = 7$) against the f1-score of the hierarchical classifier on the same dataset.

The above results demonstrate that the proposed hierarchical classification framework provides superior f1-score performance than the non-hierarchical classifier. It also overcomes the memory limitation that is discussed above.

Earlier studies mostly focused on the comprehensive evaluation of phylogenetic and statistical learning models for DNA barcode-based classification, and pointed out in agreement that SVM outperforms other alternatives consistently in most of the studied datasets.[9,16] However, a major limitation point was that SVM could not scale for datasets that contain multiple species, classes, etc. Our results in this section demonstrate that with the proposed hierarchical classification framework, SVM's superior performance now become available for datasets with multiple species, classes, etc. as well.

3.5. *Robustness analysis*

In this section, we study the robustness of hierarchical and non-hierarchical classification frameworks in the presence of mutations and/or noise in DNA sequences. In order to simulate the mutations or sequencing noise, we randomly introduce artificial mutations in the DNA barcode sequences with different ratios. More specifically, the mutation ratios are varied in the range [0, 1] with a step of 0.1, and the f1-score for each classifier is reported. For each mutation ratio, the number of mutation positions is calculated by multiplying the ratio by the sequence length, and then that many mutations are introduced at random positions. Replacement characters are chosen randomly from the set A, G, C, T.

As discussed in the above experiments, one single SVM classifier could not be trained using the merged dataset due to high memory requirements. However, the RF algorithm could scale to train one classifier capable of predicting the species for a given DNA barcode sequence regardless of the taxonomic class in the merged dataset. Here, we compare the robustness of the proposed hierarchical classification framework (with SVM and RF subclassifiers, separately) to that of non-hierarchical classifier (built with RF). The kernel that is employed in all classifiers is a linear kernel due to the relative efficiency of the SVM-linear classifiers as illustrated in Figs. 11–15. All studied classifiers are trained and tested with 10-folds

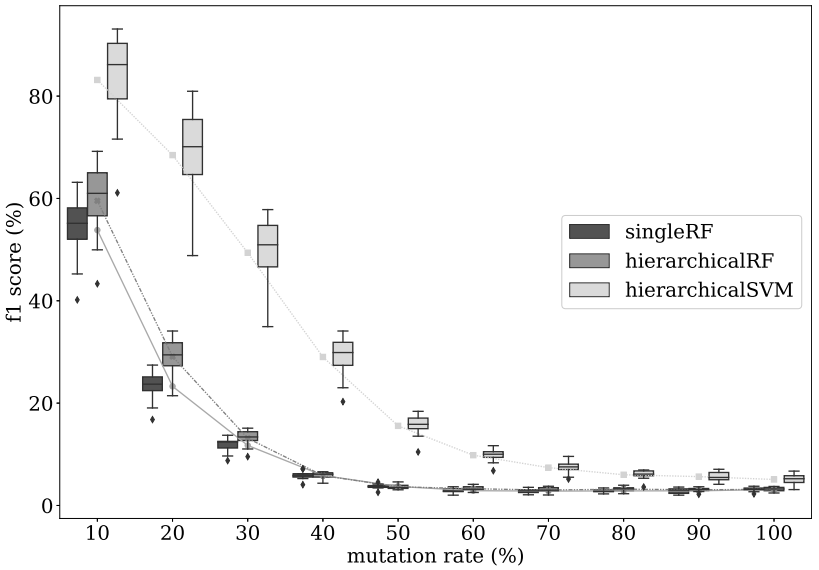

Fig. 20. The effect of introducing artificial mutations on the classification test samples, f1-scores of a hierarchical RF classifier with the number of trees set to 10, non-hierarchical RF classifier with the number of trees set to 10, a hierarchical SVM classifier with linear kernel.

cross-validation using the combined dataset that merges the five taxonomic classes considered in this work.

Figure 20 presents the test f1-scores of classifiers. These results demonstrate that the proposed hierarchical framework provides more robust f1-score performance in the existence of mutations of noise in data in comparison to the conventional non-hierarchical structure.

4. Conclusions

In this paper, the problem of assigning taxonomic labels to DNA sequences using supervised learning is studied. A multi-level hierarchical classification framework, which combines multiple classifiers built for predicting a label (e.g. class, genus, species, etc.) at different levels in organism taxonomy, is proposed. The proposed framework is evaluated on real data of 1400 species from BOLD systems. We demonstrate that in comparison to the conventional supervised classifiers, the proposed method provides the following advantages: (i) better f1-scores, (ii) improved scalability, (iii) more robustness against mutations or noise in sequence data.

The recent works in natural language processing field have shown promising progress in understanding text given sequences of characters. We believe that similar techniques may be employed to achieve better results in the problem of classifying living organisms taxonomy. As part of our future work, we plan to investigate the use of deep learning within the proposed hierarchical taxonomy classification framework.

In particular, we will explore the possible adaptation of long sort-term memory and convolutional neural networks architectures.

Availability: https://github.com/sehir-bioinformatics-database-lab/Hierarchical-Supervised-Learners.

References

1. Osmundson TW, Robert VA, Schoch CL, Baker LJ, Smith A, Robich G, Mizzan L, Garbelotto MM, Filling gaps in biodiversity knowledge for macrofungi: Contributions and assessment of an herbarium collection DNA barcode sequencing project, *PLoS One* **8**(4):e62419, 2013.
2. Porter TM, Gibson JF, Shokralla S, Baird DJ, Golding GB, Hajibabaei M, Rapid and accurate taxonomic classification of insect (class Insecta) cytochrome c oxidase subunit 1 (COI) DNA barcode sequences using a naïve Bayesian classifier, *Mol Ecol Resour* **14**(5):929–942, 2014.
3. Hebert PD, Cywinska A, Ball SL, Dewaard JR, Biological identifications through DNA barcodes, *Proc R Soc Lond B Biol Sci* **270**(1512):313–321, 2003.
4. Hu L, Zhao Y, Yang Y, Niu D, Yang R, LSU rDNA D5 region: The DNA barcode for molecular classification and identification of demodex, *Genome* **62**(5):295–304, 2019.
5. Rahman MM, Norén M, Mollah AR, Kullander SO, Building a DNA barcode library for the freshwater fishes of Bangladesh, *Sci Rep* **9**(1):1–10, 2019.
6. Mortágua A, Vasselon V, Oliveira R, Elias C, Chardon C, Bouchez A, Rimet F, Feio MJ, Almeida SF, Applicability of DNA metabarcoding approach in the bioassessment of Portuguese rivers using diatoms, *Ecol Indic* **106**:105470, 2019.
7. International Barcode of Life, *iBOL* Available at http://ibol.org/, 2008.
8. Emu M, Sakib S, Species identification using DNA barcode sequences through supervised learning methods, *2019 Int. Conf. Electrical, Computer and Communication Engineering (ECCE)*, IEEE, pp. 1–6, 2019.
9. Weitschek E, Fiscon G, Felici G, Supervised DNA barcodes species classification: Analysis, comparisons and results, *BioData Min* **7**(1):4, 2014.
10. Weitschek E, Van Velzen R, Felici G, bertolazzi P, BLOG 2.0: A software system for characterbased species classification with DNA Barcode sequences, what it does, how to use it, *Mol Ecol Resour* **13**(6):1043–1046, 2013.
11. Kuksa P, Pavlovic V, Efficient alignment-free DNA barcode analytics, *BMC Bioinf.* **10**(S14):S9, 2009.
12. Suthaharan S, Machine learning models and algorithms for big data classification, *Integr Ser Inf Syst* **36**:1–2, 2016.
13. Asadi S, Shahrabi J, RipMC: RIPPER for multiclass classification, *Neurocomputing* **191**:19–33, 2016.
14. Tanha J, van Someren M, Afsarmanesh H, Semi-supervised self-training for decision tree classifiers, *Int J Mach Learn Cybern* **8**(1):355–370, 2017.
15. Zhou X, Wang S, Xu W, Ji G, Phillips P, Sun P, Zhang Y, Detection of pathological brain in MRI scanning based on wavelet-entropy and naive Bayes classifier, *Int Conf Bioinf Biomed Eng*, Springer, Cham, pp. 201–209, 2015.
16. Austerlitz F, David O, Schaeffer B, Bleakley K, Olteanu M, Leblois R, Veuille M, Laredo C, DNA barcode analysis: A comparison of phylogenetic and statistical classification methods, *BMC Bioinf* **10**(14):S10, 2009.
17. Saitou N, Nei M, The neighbor-joining method: A new method for reconstructing phylogenetic trees, *Mol Biol Evol* **4**(4):406–425, 1987.

18. Guindon S, Gascuel O, A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood, *Syst Biol* **52**(5):696–704, 2003.
19. Zhao X, Tian K, Yau SS, A new efficient method for analyzing fungi species using correlations between nucleotides, *BMC Evol Biol* **18**(1):1–13, 2018.
20. Kabir T, Shemonti AS, Rahman AH, Species identification using partial DNA sequence: A machine learning approach, *2018 IEEE 18th Int Conf Bioinformatics and Bioengineering (BIBE)*, IEEE, pp. 235–242, 2018.
21. Landwehr N, Hall M, Frank E, Logistic model trees, *Mach Learn* **59**(1–2):161–205, 2005.
22. Aha DW, Kibler D, Albert MK, Instance-based learning algorithms, *Mach Learn* **6**(1):37–66, 1991.
23. Frank E, Witten IH, Generating accurate rule sets without global optimization. In *Proceedings of the Fifteenth International Conference on Machine Learning*, 1998, pp. 144–151.
24. Breiman L, Random forests, *Mach Learn* **45**(1):5–32, 2001.
25. Breiman L, Bagging predictors, *Mach Learn* **24**(2):123–140, 1996.
26. Barcode of Life Data (BOLD) Systems, Available at http://www.boldsystems.org/index.php/databases, 2014.
27. Ben-Hur A, Noble WS, Kernel methods for predicting protein–protein interactions, *Bioinformatics* **21**(suppl 1):i38–i46, 2005.
28. Leslie C, Eskin E, Noble WS, The spectrum kernel: A string kernel for SVM protein classification, *Biocomputing*, pp. 564–575, 2002.
29. Leslie CS, Eskin E, Cohen A, Weston J, Noble WS, Mismatch string kernels for discriminative protein classification, *Bioinformatics* **20**(4):467–476, 2004.
30. Kuksa PP, Huang PH, Pavlovic V, Scalable algorithms for string kernels with inexact matching, *Advances in Neural Information Processing Systems*, pp. 881–888, 2009.
31. Ward RD, Holmes BH, O'Hara TD, DNA barcoding discriminates echinoderm species, *Mol Ecol Resour* **8**(6):1202–1211, 2008.
32. Dimitriadou E, Hornik K, Leisch F, Meyer D, Weingessel A, Misc functions of the department of statistics (e1071), TU Wien, *R Package* **1**:5–24, 2008.
33. Liaw A, Wiener M, Classification and regression by randomForest, *R News* **2**(3):18–22, 2002.

Gihad N. Sohsah received her Bachelor of Computer Engineering from the Tanta University (Tanta, Egypt) in 2011. She has recently received her M.Sc. degree in Data Science from Istanbul Sehir University (Istanbul, Turkey). Gihad is mainly interested in both theoretical and applied machine learning and artificial intelligence.

Ali Reza Ibrahimzada is currently studying his B.Sc. degree in Computer Science and Engineering at Istanbul Sehir University (Istanbul, Turkey). Ali Reza is mainly interested in machine learning, artificial intelligence, and data science. He is a member of Bioinformatics and Databases Lab at Istanbul Sehir University.

Huzeyfe Ayaz is a Sophomore Student in Computer Science and Engineering department at the Istanbul Sehir University (Istanbul, Turkey). He is interested in data science, machine learning, and AI. Currently, he works on several research projects to improve the public health within the Bioinformatics and Databases Lab at Istanbul Sehir University.

Ali Cakmak received his B.Sc. degree in 2003 from the Computer Engineering Department at the Bilkent University (Ankara, Turkey), and his Ph.D. degree in 2008 from the Electrical Engineering and Computer Science Department at Case Western Reserve University (Cleveland, OH). Then, he moved to the Silicon Valley, and worked as a Senior Software Engineer as part of the Query Optimization Group at Oracle, Inc (Redwood Shores, CA). His research interests include bioinformatics, machine learning, data mining, databases, and data science. Dr. Cakmak is a recipient of TUBITAK (The Scientific and Technological Research Council of Turkey) Career Grant.