DNA Barcode Classification Using General Regression Neural Network with Different Distance Models

Massimo La Rosa, Antonino Fiannaca, Riccardo Rizzo, and Alfonso Urso

Abstract The "cythosome c oxidase subunits 1" (COI) gene is used for identification of species, and it is one of the so-called *DNA barcode* genes. Identification of species, even using DNA barcoding can be difficult if the biological examples are degraded. Spectral representation of sequences and the General Regression Neural Network (GRNN) can give some interesting results in these difficult cases. The GRNN is based on the distance between the memorized examples of sequence and the input unknown sequence, both represented using a vector space spectral representation. In this paper we will analyse the effectiveness of different distance models in the GRNN implementation and will compare the obtained results in the classification of full length sequences and degraded samples.

Keywords Barcode classification • Alignment-free • GRNN

1 Introduction

The so-called *DNA barcode sequence* is a small segment ($\sim$650 bp) of DNA, usually from "cythosome c oxidase subunits 1" mitochondrial gene (COI) [8, 13]. The sequence is a good marker for DNA and is widely used for identification and taxonomic rank assignment of many species [5].

DNA barcoding is difficult if the biological samples under analysis are degraded: in this case only fragments of the barcode sequence is available. A suitable solution for this problem is studied in [14]: in this work the barcode sequence is analysed in order to find small subsequences that are still useful for identification of the sample specie.

We started from a different point of view: we addressed the identification and rank assignment of degraded barcode sequences, usually sequence fragments of about 200 bp, building a robust classifier based on the spectral representation and a modified version of the General Regression Neural Network (GRNN).

M. La Rosa (✉) • A. Fiannaca • R. Rizzo • A. Urso
ICAR-CNR, viale delle Scienze Ed. 11, 90128 Palermo, Italy
e-mail: larosa@pa.icar.cnr.it; fiannaca@pa.icar.cnr.it; ricrizzo@pa.icar.cnr.it; urso@pa.icar.cnr.it

V. Zazzu et al. (eds.), *Mathematical Models in Biology*,
DOI 10.1007/978-3-319-23497-7_9

Using spectral representation the DNA barcode sequence is represented using the frequency of very short strings of length $k = 3, 4, \ldots,$ called *k-mers*. This sequence representation is often addressed as *k-mers* decomposition or, more generally, as alignment-free sequence decomposition. In this representation the order of k-mers in the sequence is discarded and only their count is considered; if a sequence fragment has a k-mers frequency distribution similar to the one of the whole barcode sequence then the two will have a similar representation.

The set of the frequencies of the k-mers in a sequence constitutes the representing vector for the sequence in a vector space. The dimension of the representation space is 4^k and the distance among these representing vector can be calculated using Euclidean norm in $\Re^{4^k}$.

The GRNN is a neural network originally developed for regression and adapted to classification of DNA sequences in [17]. This modification made the network a prototype-based classification tool that classifies a new input looking at the distance from the memorized training samples. It is clear that different distance models, like *Euclidean*, *manhattan* and so on, can change the performances of the network, as we found in [17].

In this paper we want to go further in this study and analyse and compare the performances of other distance models on the GRNN, considering classification results of both full length sequences and degraded samples.

With regards to barcode classification, very interesting results have been obtained in the works presented in [10, 15, 20]. In particular both the algorithms described in [15, 20] propose alignment-based methods in order to classify barcode specimen. In [20], after the training sequences are aligned, a set of logic rules are extracted in the form "if pos35 is G and pos300 is A then the sequence is classified as …", where pos*X* represents a sequence locus. In [15], first a phylogenetic tree of input sequences is computed; then at each branching node, a set of "characteristic attributes" (CA) is identified for the corresponding leaf nodes. Considering a branch node, CAs are single nucleotide position or multiple nucleotide positions that are shared only by one of the branch descending from that node. Another alignment-free approach more similar to our proposed method is the one presented in [10]. There authors introduce the spectral representation for the barcode sequences and they use two machine learning algorithms, k-Nearest Neighbour (kNN) [2] and Support Vector Machine (SVM) [18], to train different classifiers. In this paper we are going to compare our GRNN approach with the classifiers proposed in [10] because they represent alignment-free approaches, differently from [15] and [20], that also implement the spectral representation. The comparison between our GRNN method and the SVM classifier has been already done in [17], where we demonstrated our method outperforms SVM when dealing with sequence fragments. Therefore in this paper we compare our GRNN method against the k-NN classifier.

2 Methods

Prototype-based classification tools are based on sequence distance; there are many algorithm to evaluate sequence distance besides the evolutionary distance, for example the compression distance used in [11, 12]. The vector space representation is obtained by considering the frequency of all possible 5-letters substring in the DNA barcode sequence (k-mers), these k-mers are obtained by using a sliding window on the sequence. A deeper discussion on this representation can be found in [3, 10]. In the following sections the GRNN modified algorithm is explained and the different distance measures applied are described; moreover the barcode dataset used is introduced.

2.1 The General Regression Neural Network

Artificial neural networks (ANN) are a set of algorithms used to approximate functions or cluster large sets of input values. A neural network usually have a very large set of parameters (the network weights) adapted using a set of training examples and a specific learning algorithm (the training algorithm). The training phase is aimed at reducing the error of the network on a specific task, classification or regression, by changing the weight values.

Among the neural networks the GRNN [19] is a network created for regression i.e. the approximation of the values of a dependent continuous variable y given a set of samples $(\mathbf{x}_i, y_i)$ $i = 1, 2, \ldots N$.

In the following we will discuss the one dimensional output case, the extension to an output vector **y** being straightforward (see [19] for details).

The GRNN do not have a training phase, it is based on the memorization of all the training examples in the hidden layer: one neural unit for each training samples (see Fig. 1). When a new pattern $\mathbf{x}'$ is presented to the network input the output y is calculated using the following equation:

$$y' = \frac{\sum w_i * y_i}{\sum w_i}. \tag{1}$$

where the weight w_i are obtained from each hidden unit as

$$w_i = \exp\left\{-\frac{d(\mathbf{x}', \mathbf{x_i})}{2\sigma^2}\right\} \tag{2}$$

The σ value, called spread factor, is the only parameter of the GRNN network. The weight w_i is considered by some literature the excitation level of the neural unit i corresponding to the input $\mathbf{x}'$.

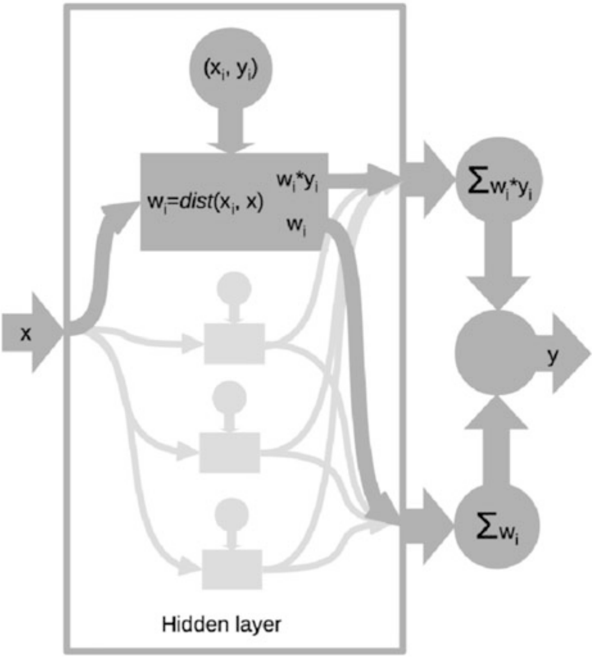


Fig. 1 The representation of the GRNN neural network. The hidden layer contains all the training patterns and calculates the w_i considering the distance from the input pattern x. These weights are used to calculate the output. On the *right* there is the output layer composed by three units: the *upper* one collects all the terms $w_i * y_i$ and the *lower* one collects the terms w_i: these terms are combined in the third unit that generates the output

There are some studies on the optimal value of σ that can be a single value for the whole network or a specific value for each hidden unit. In [7] it is suggested a formula that depends on the maximum distance and number of patterns in the training set.

The GRNN can be used in classification problems: considering a set of classification examples $(\mathbf{x_i}, c_h)$ where $\mathbf{x_i}$ (with $i = 1, 2, \ldots N$) is the input pattern and c_h (with $h = 1, 2, \ldots H$; H is the number of available classes) is the class assigned to the pattern $\mathbf{x_i}$ it is possible to build a set of training examples for a GRNN network as $(\mathbf{x_i}, \mathbf{y_i})$ were $y_i = [y_{i,1}, y_{i,2}, \ldots, y_{i,H}]$ is given by

$$y_{i,j} = \begin{cases} 0 \text{ if } j \neq h \\ 1 \text{ if } j = h \end{cases} \tag{3}$$

where c_h is the class of the pattern x_i.

The set of couples $(\mathbf{x_i}, y_i)$ can be used as a training set for the GRNN and the class of the new input $\mathbf{x}'$ can be calculated as

$$c_h(\mathbf{x}') = \arg\max_j \{y'_j | j = 1, 2, \ldots H\} \tag{4}$$

In order to implement our classification tool for DNA sequences, we obtained the vector representation of the DNA sequences using a k-mer decomposition, as shown in [10], in which sequences are coded as fixed size vectors whose components are the number of occurrences of short DNA snippets of k fixed-length, called k-mers. Considering $k = 5$, as proposed in [10], we have vectors of dimension $4^5 = 1024$ to represent genomic sequences.

The GRNN is used with different distance models, in particular some of the L_p norms, the correlation norm and the cosine norm.

2.2 *The Distance Models*

In this section the L_p norms used are introduced, together with the cosine and correlation distances.

2.2.1 L_p Norms

The norm is a function that assigns a strict positive number to a vector in a vector space $f : (\mathbf{x}) \rightarrow \Re$ that satisfies the following properties:

$$f(\alpha\mathbf{x}) = |\alpha| f(\mathbf{x}) \tag{5}$$

$$f(\mathbf{x} + \mathbf{y}) \leq f(\mathbf{x}) + f(\mathbf{y}) \tag{6}$$

$$\text{if } f(\mathbf{x}) = 0 \text{ then } \mathbf{x} \text{ is the vector zero} \tag{7}$$

the L_p family norms, or *p-norms*, defined as:

$$\|\mathbf{x}\|_p = \left(\sum_i |x_i|^p\right)^{\frac{1}{p}}. \tag{8}$$

The most common norm is the Euclidean norm with $p = 2$, but are also used the $p = 1$ norm namely City-block or Manhattan, and the Chebyshev norm, or L_∞. Although should be $p \geq 1$ there are also fractional norms with $p < 1$, that are interesting in the case of high dimensional spaces.

In case of high-dimensionality data, such as the 1024 sized vectors representing DNA sequences, the Euclidean norm used to define the distance tend to *concentrate* [4]. That means all pairwise distances between high-dimensional objects appear to be very similar. Authors in [4] also state that the concentration phenomenon is intrinsic to the norm. In order to overcome this phenomenon, fractional norms can be used in place of Euclidean norm [1, 9]; whereas with $0 < p < 1$ L_p norms are called fractional norms, which induce fractional distances. Moreover, fractional norms are able to deal with non-Gaussian noise [4]. In this work we adopted fractional norms, considering different values of p, in order to compute Eq. (2) and to limit the effects of the curse of dimensionality.

If $p = 1$ in Eq. (8) the norm is called the *Manhattan norm*, or *taxicab norm*, and is defined as

$$L_1 = \sum_i \left|x'_i - x_i\right| . \tag{9}$$

both the names are related to the distance a taxi as to drive in a city with a rectangular grid.

The Chebyshev distance is obtained from the formula:

$$d(\mathbf{x}', \mathbf{x}_i) = \max_i(\left|x'_i - x_i\right|). \tag{10}$$

this is usually considered as L_∞ norm.

2.2.2 Cosine and Correlation Distance

Cosine and correlation distance are both based on scalar product $\mathbf{x}' \cdot \mathbf{x}_i$, instead of the difference $\mathbf{x}' - \mathbf{x}_i$. The cosine distance is defined by the following equation:

$$d(\mathbf{x}', \mathbf{x}_i) = 1 - \frac{\mathbf{x}' \cdot \mathbf{x}_i}{\|\mathbf{x}'\| \, \|\mathbf{x}_i\|} \tag{11}$$

where the $\|.\|$ is the Euclidean norm. The correlation distance is defined by:

$$d(\mathbf{x}', \mathbf{x}_i) = 1 - \frac{(\mathbf{x}' - \overline{\mathbf{x}'}) \cdot (\mathbf{x}_i - \overline{\mathbf{x}_i})}{\|\mathbf{x}' - \overline{\mathbf{x}'}\| \, \|\mathbf{x}_i - \overline{\mathbf{x}_i}\|} \tag{12}$$

where $\overline{\mathbf{x}'}$ is the mean of the input vectors $\mathbf{x}'$ and $\overline{\mathbf{x}_i}$ is the mean of the training samples.

2.3 *Barcode Dataset*

We downloaded barcode sequences from the Barcode of Life Database (BOLD) [16]. In our study, we considered 10 barcode datasets belonging to different BOLD projects and living organisms. These datasets have been selected according to some criteria: we chose only *barcode compliant* dataset, i.e certified by BOLD as true barcode sequences, with sequence length not shorter than 500 bp and not longer than 800 bp. These datasets differ each other on the basis of the number of species and specimen, the sequence length and the sequence quality (in terms of undefined nucleotides). Following these criteria, we collected 2210 sequences. The dataset composition, in terms of number of different taxa and number of specimen for each taxa, is summarized in Table 1, where it is possible to note how the dataset is unbalanced.

Table 1 Barcode dataset composition at each taxonomic level

Sequence distribution for each taxa								
Phylum			Class			Order		
# Classes	# Seqs	% Seqs	# Classes	# Seqs	% Seqs	# Classes	# Seqs	% Seqs
1	1361	61.9 %	1	1361	61.9 %	1	1049	47.46 %
2	[219,386]	27.4 %	2	[219,286]	22.85 %	3	[209,286]	32.30 %
2	[111,133]	11.0 %	3	[100,133]	15.56 %	4	[100,133]	20.22 %
Family			Genus			Species		
# Classes	# Seqs	% Seqs	# Classes	# Seqs	% Seqs	# Classes	# Seqs	% Seqs
1	885	40.04 %	1	386	17.46 %	1	279	12.64 %
3	[209,274]	31.76 %	3	[209,290]	32.48 %	4	[105,140]	22.30 %
4	[103,164]	23.12 %	6	[103,164]	35.15 %	30	[14,92]	49,50 %
7	[4,46]	5.06 %	15	[4,71]	14.91 %	35	[1,11]	15.56 %

Numbers between square brackets represent range of values

3 Results and Discussion

In this section, we describe the parameter setup for the GRNN algorithm and the adopted training/testing procedure. Then we report classification results in terms of accuracy, precision and recall scores, and finally we discuss those results.

3.1 Experimental Setup

The only parameter of the GRNN algorithm is the spread factor σ (Eq. 2). In our experiments, we tuned the σ value by means of a ten fold cross validation procedure, considering as training set the dataset composed of the full length sequences. This procedure has been carried out implementing each distance model (see Sect. 2.2), and for values of σ ranging from 0.5 to 0.8, with a step of 0.1. For each value of σ we noticed that the behaviour of the GRNN was substantially the same regardless the distance model, and the best results, in terms of error rate, were obtained with $\sigma = 0.6$. As for the fractional distances, Eq. (8) with $p < 1$, we considered three values for p: 0.3, 0.5, 0.7. All the experiments have been done using Python scripts on a Windows 7 machine equipped with i7 Intel CPU at 2.8 GHz with 8 GB of RAM. Computational times of the GRNN algorithm are about 1 min for a single experiment.

The classification performances of the GRNN algorithm have been tested considering full length barcode sequences and sequence fragments of 200 consecutive bp randomly extracted from the original sequences. We want to assess the GRNN predictive power and its robustness with regards to the sequence sizes. In fact, in the study of environmental species, for example, usually only small portions of the barcode sequences are available.

For each distance model, the training and testing procedures have been done in two ways. In the first case, we adopted a ten fold cross validation method: in each fold, we trained the GRNN with the 90 % of the full-length sequences and we used as test set the remaining 10 % of both the full-length sequences and their corresponding sequence fragments of 200 bp. In the second case, we trained the GRNN with the whole dataset of the full-length sequences and then we tested it with all the sequence fragments. In the first scenario, we want to assess the classification performances of the GRNN considering full-length sequences and its generalization degree when used to classify sequence fragments whose corresponding original sequence does not belong to the training set. In the second scenario, we supposed the GRNN is used to recognize small random fragments, by "knowing" all the original full-length sequences. Comparison with the k-NN classifier has been carried out following the same training and testing procedure. We used the k-NN implementation provided by the Weka Experimenter Platform [6], considering $k = 1$ and $k = 3$, as done similarly in [10].

3.2 Classification Results

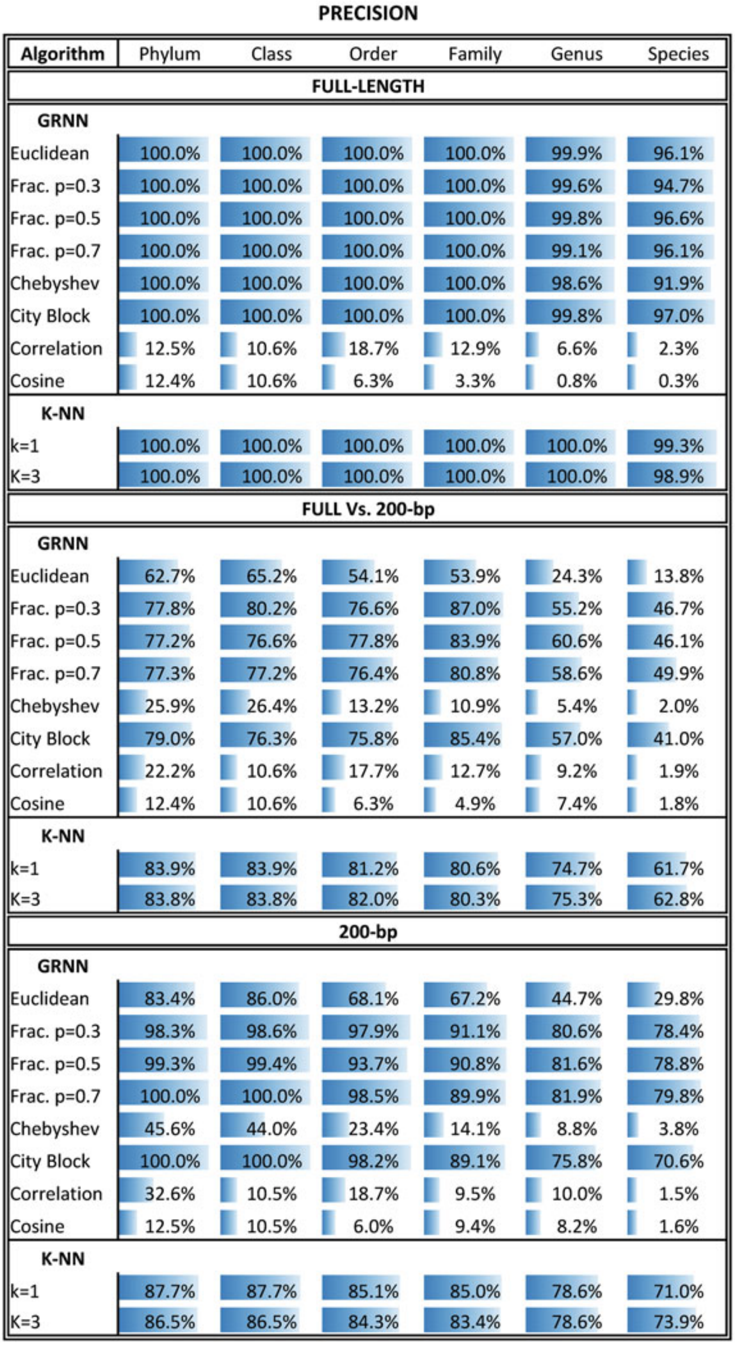
Classification scores have been evaluated by means of the accuracy, precision and recall performance measures.

These scores are summarized in Tables 2, 3, and 4, respectively. Each table is composed of three parts, according to the adopted training/testing procedure. "Full-length" means the classification results are obtained through a ten fold cross validation scheme considering full length sequences both for training and testing; the scores are averaged over the ten folds. "Full vs. 200-bp" means the classification results are obtained through a ten fold cross validation scheme considering full-length sequences for training and 200 bp fragments for testing; once again the scores are averaged over the ten folds. "200-bp" means the classification results are obtained training with the whole dataset of full-length sequences and tested with all the sequence fragments. In each table, in the first column there is the distance model used to train the GRNN, and in the second row there is the taxonomic level, from Phylum to Species. The last two rows of each table part show the results obtained from the k-NN classifiers.

3.3 Discussion

From the classification results shown in Tables 2, 3, and 4, it is evident that the GRNN and the k-NN algorithms are able to correctly classify full-length barcode sequences, with scores around 100 % at each taxonomic level. The GRNN reaches those scores with all the distance models except for the correlation and the cosine distances.

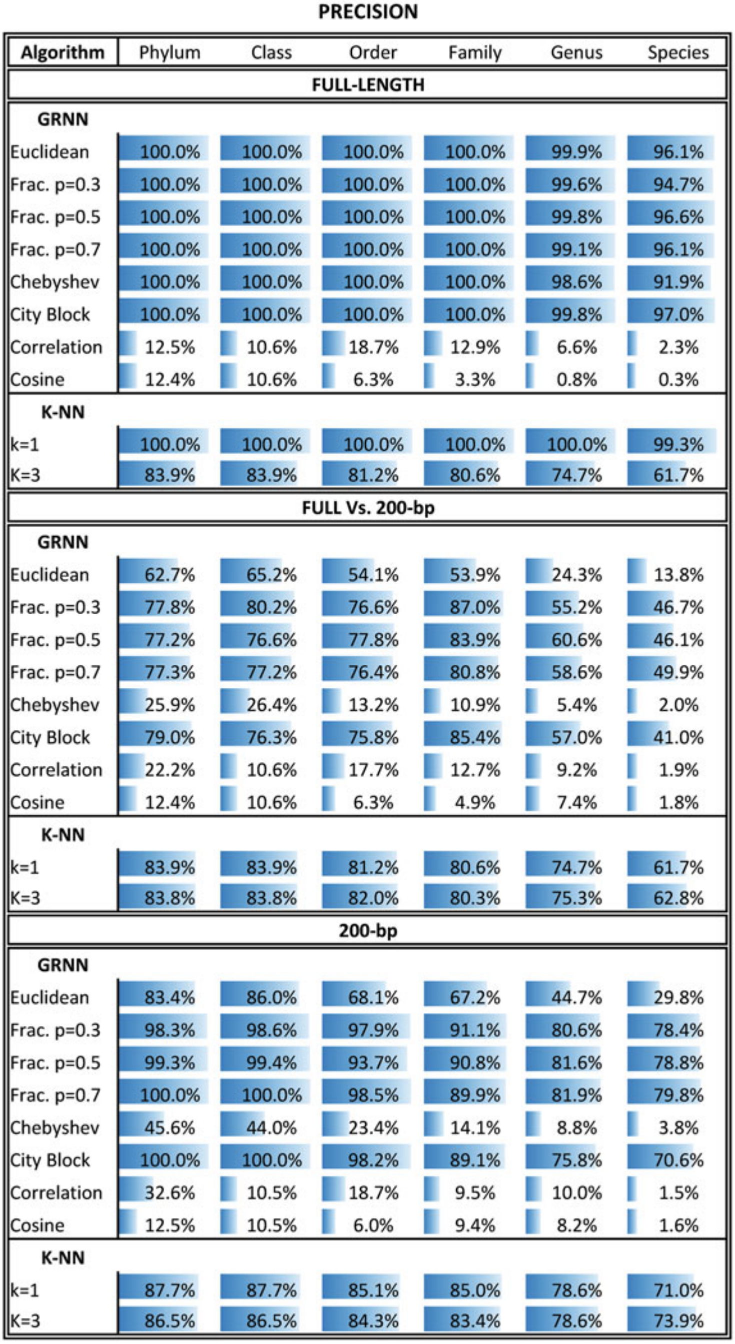
Table 2 Accuracy scores at each taxonomic level the GRNN algorithm, considering each distance model, and the k-NN classifier



PRECISION

Algorithm	Phylum	Class	Order	Family	Genus	Species
FULL-LENGTH						
GRNN						
Euclidean	100.0%	100.0%	100.0%	100.0%	99.9%	96.1%
Frac. p=0.3	100.0%	100.0%	100.0%	100.0%	99.6%	94.7%
Frac. p=0.5	100.0%	100.0%	100.0%	100.0%	99.8%	96.6%
Frac. p=0.7	100.0%	100.0%	100.0%	100.0%	99.1%	96.1%
Chebyshev	100.0%	100.0%	100.0%	100.0%	98.6%	91.9%
City Block	100.0%	100.0%	100.0%	100.0%	99.8%	97.0%
Correlation	12.5%	10.6%	18.7%	12.9%	6.6%	2.3%
Cosine	12.4%	10.6%	6.3%	3.3%	0.8%	0.3%
K-NN						
k=1	100.0%	100.0%	100.0%	100.0%	100.0%	99.3%
K=3	100.0%	100.0%	100.0%	100.0%	100.0%	98.9%
FULL Vs. 200-bp						
GRNN						
Euclidean	62.7%	65.2%	54.1%	53.9%	24.3%	13.8%
Frac. p=0.3	77.8%	80.2%	76.6%	87.0%	55.2%	46.7%
Frac. p=0.5	77.2%	76.6%	77.8%	83.9%	60.6%	46.1%
Frac. p=0.7	77.3%	77.2%	76.4%	80.8%	58.6%	49.9%
Chebyshev	25.9%	26.4%	13.2%	10.9%	5.4%	2.0%
City Block	79.0%	76.3%	75.8%	85.4%	57.0%	41.0%
Correlation	22.2%	10.6%	17.7%	12.7%	9.2%	1.9%
Cosine	12.4%	10.6%	6.3%	4.9%	7.4%	1.8%
K-NN						
k=1	83.9%	83.9%	81.2%	80.6%	74.7%	61.7%
K=3	83.8%	83.8%	82.0%	80.3%	75.3%	62.8%
200-bp						
GRNN						
Euclidean	83.4%	86.0%	68.1%	67.2%	44.7%	29.8%
Frac. p=0.3	98.3%	98.6%	97.9%	91.1%	80.6%	78.4%
Frac. p=0.5	99.3%	99.4%	93.7%	90.8%	81.6%	78.8%
Frac. p=0.7	100.0%	100.0%	98.5%	89.9%	81.9%	79.8%
Chebyshev	45.6%	44.0%	23.4%	14.1%	8.8%	3.8%
City Block	100.0%	100.0%	98.2%	89.1%	75.8%	70.6%
Correlation	32.6%	10.5%	18.7%	9.5%	10.0%	1.5%
Cosine	12.5%	10.5%	6.0%	9.4%	8.2%	1.6%
K-NN						
k=1	87.7%	87.7%	85.1%	85.0%	78.6%	71.0%
K=3	86.5%	86.5%	84.3%	83.4%	78.6%	73.9%

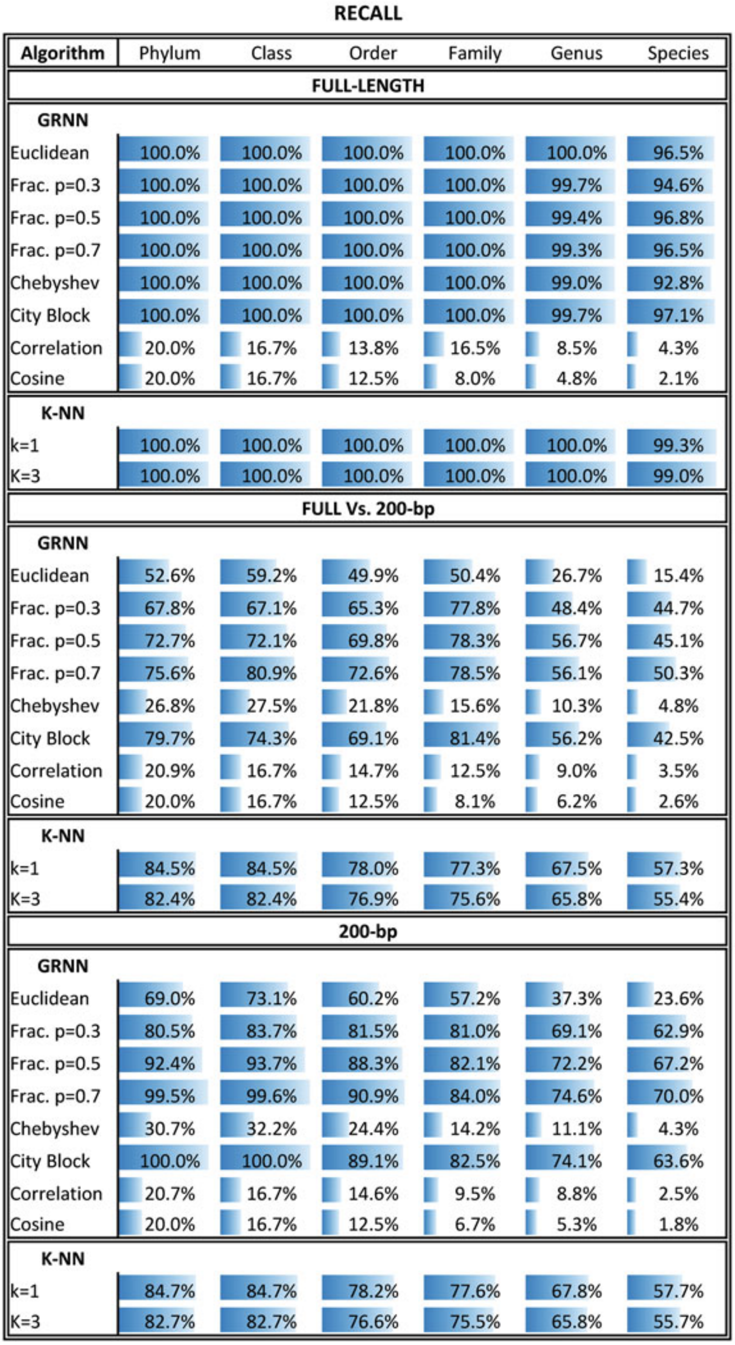
Table 3 Precision scores at each taxonomic level the GRNN algorithm, considering each distance model, and the k-NN classifier



PRECISION

Algorithm	Phylum	Class	Order	Family	Genus	Species
FULL-LENGTH						
GRNN						
Euclidean	100.0%	100.0%	100.0%	100.0%	99.9%	96.1%
Frac. p=0.3	100.0%	100.0%	100.0%	100.0%	99.6%	94.7%
Frac. p=0.5	100.0%	100.0%	100.0%	100.0%	99.8%	96.6%
Frac. p=0.7	100.0%	100.0%	100.0%	100.0%	99.1%	96.1%
Chebyshev	100.0%	100.0%	100.0%	100.0%	98.6%	91.9%
City Block	100.0%	100.0%	100.0%	100.0%	99.8%	97.0%
Correlation	12.5%	10.6%	18.7%	12.9%	6.6%	2.3%
Cosine	12.4%	10.6%	6.3%	3.3%	0.8%	0.3%
K-NN						
k=1	100.0%	100.0%	100.0%	100.0%	100.0%	99.3%
K=3	83.9%	83.9%	81.2%	80.6%	74.7%	61.7%
FULL Vs. 200-bp						
GRNN						
Euclidean	62.7%	65.2%	54.1%	53.9%	24.3%	13.8%
Frac. p=0.3	77.8%	80.2%	76.6%	87.0%	55.2%	46.7%
Frac. p=0.5	77.2%	76.6%	77.8%	83.9%	60.6%	46.1%
Frac. p=0.7	77.3%	77.2%	76.4%	80.8%	58.6%	49.9%
Chebyshev	25.9%	26.4%	13.2%	10.9%	5.4%	2.0%
City Block	79.0%	76.3%	75.8%	85.4%	57.0%	41.0%
Correlation	22.2%	10.6%	17.7%	12.7%	9.2%	1.9%
Cosine	12.4%	10.6%	6.3%	4.9%	7.4%	1.8%
K-NN						
k=1	83.9%	83.9%	81.2%	80.6%	74.7%	61.7%
K=3	83.8%	83.8%	82.0%	80.3%	75.3%	62.8%
200-bp						
GRNN						
Euclidean	83.4%	86.0%	68.1%	67.2%	44.7%	29.8%
Frac. p=0.3	98.3%	98.6%	97.9%	91.1%	80.6%	78.4%
Frac. p=0.5	99.3%	99.4%	93.7%	90.8%	81.6%	78.8%
Frac. p=0.7	100.0%	100.0%	98.5%	89.9%	81.9%	79.8%
Chebyshev	45.6%	44.0%	23.4%	14.1%	8.8%	3.8%
City Block	100.0%	100.0%	98.2%	89.1%	75.8%	70.6%
Correlation	32.6%	10.5%	18.7%	9.5%	10.0%	1.5%
Cosine	12.5%	10.5%	6.0%	9.4%	8.2%	1.6%
K-NN						
k=1	87.7%	87.7%	85.1%	85.0%	78.6%	71.0%
K=3	86.5%	86.5%	84.3%	83.4%	78.6%	73.9%

Table 4 Recall scores at each taxonomic level the GRNN algorithm, considering each distance model, and the k-NN classifier



RECALL

Algorithm	Phylum	Class	Order	Family	Genus	Species
FULL-LENGTH						
GRNN						
Euclidean	100.0%	100.0%	100.0%	100.0%	100.0%	96.5%
Frac. p=0.3	100.0%	100.0%	100.0%	100.0%	99.7%	94.6%
Frac. p=0.5	100.0%	100.0%	100.0%	100.0%	99.4%	96.8%
Frac. p=0.7	100.0%	100.0%	100.0%	100.0%	99.3%	96.5%
Chebyshev	100.0%	100.0%	100.0%	100.0%	99.0%	92.8%
City Block	100.0%	100.0%	100.0%	100.0%	99.7%	97.1%
Correlation	20.0%	16.7%	13.8%	16.5%	8.5%	4.3%
Cosine	20.0%	16.7%	12.5%	8.0%	4.8%	2.1%
K-NN						
k=1	100.0%	100.0%	100.0%	100.0%	100.0%	99.3%
K=3	100.0%	100.0%	100.0%	100.0%	100.0%	99.0%
FULL Vs. 200-bp						
GRNN						
Euclidean	52.6%	59.2%	49.9%	50.4%	26.7%	15.4%
Frac. p=0.3	67.8%	67.1%	65.3%	77.8%	48.4%	44.7%
Frac. p=0.5	72.7%	72.1%	69.8%	78.3%	56.7%	45.1%
Frac. p=0.7	75.6%	80.9%	72.6%	78.5%	56.1%	50.3%
Chebyshev	26.8%	27.5%	21.8%	15.6%	10.3%	4.8%
City Block	79.7%	74.3%	69.1%	81.4%	56.2%	42.5%
Correlation	20.9%	16.7%	14.7%	12.5%	9.0%	3.5%
Cosine	20.0%	16.7%	12.5%	8.1%	6.2%	2.6%
K-NN						
k=1	84.5%	84.5%	78.0%	77.3%	67.5%	57.3%
K=3	82.4%	82.4%	76.9%	75.6%	65.8%	55.4%
200-bp						
GRNN						
Euclidean	69.0%	73.1%	60.2%	57.2%	37.3%	23.6%
Frac. p=0.3	80.5%	83.7%	81.5%	81.0%	69.1%	62.9%
Frac. p=0.5	92.4%	93.7%	88.3%	82.1%	72.2%	67.2%
Frac. p=0.7	99.5%	99.6%	90.9%	84.0%	74.6%	70.0%
Chebyshev	30.7%	32.2%	24.4%	14.2%	11.1%	4.3%
City Block	100.0%	100.0%	89.1%	82.5%	74.1%	63.6%
Correlation	20.7%	16.7%	14.6%	9.5%	8.8%	2.5%
Cosine	20.0%	16.7%	12.5%	6.7%	5.3%	1.8%
K-NN						
k=1	84.7%	84.7%	78.2%	77.6%	67.8%	57.7%
K=3	82.7%	82.7%	76.6%	75.5%	65.8%	55.7%

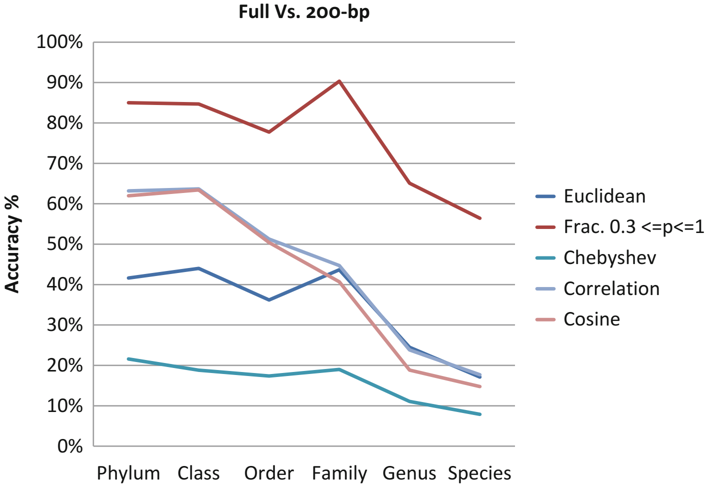


Fig. 2 Accuracy scores at each taxonomic level for the "Full vs. 200-bp" training/testing scheme of the GRNN classifier with different distance models

Using those distances, the performances of the GRNN drop significantly, reaching about 62 % in terms of accuracy at phylum level, and only about 20 and 12 % in terms of recall and precision respectively. That means distances based on scalar product of the patterns are not suitable with the GRNN algorithm. The most interesting results are therefore the ones obtained during the classification evaluation of the sequence fragments. First of all, the performances decrease with respect to taxonomic level, as it is also evident in the chart of Fig. 2. As the taxonomic rank goes down, indeed, the number of categories to classify increases (see Table 1) and, as a consequence, it is more difficult to correctly classify the patterns. Considering the "Full vs. 200-bp" part, the only meaningful scores are provided by the GRNN implementing fractional and city block distances. In particular while the correlation and the cosine distances keep on giving low scores as in the case of full-length sequences, the Chebyshev and the Euclidean distance have a strong drop of performances, with scores about 40 % for Euclidean distance at Phylum level and about 20 % for Chebyshev distance at Phylum level. The same drop of performances also affects the k-NN classifiers, with very similar scores regardless the value of k. On the other hand, considering fractional and city block distances, the GRNN is still able to provide acceptable classification results for sequence fragments, with scores ranging from about 85 % at phylum level to abut 57 % at Species level. These results further confirm that fractional norms contrast the effects of distance concentration. It is important to remember that in the case of "Full vs. 200-bp" the GRNN network classify the sequence fragments without "knowing" the corresponding full length sequences during the training phase. It is interesting to note (see Fig. 2) that at the family level there are the best scores: that because the distribution of specimen at family level is very unbalanced, with one family collecting about the 40 % of available samples, as reported in Table 1. Finally, considering the “200-bp” part of Tables 2, 3, and 4, once again only the GRNN implementing the fractional and the city block distances are able to provide a proper classification for sequence fragments. In this last case, the performance scores are higher than the “Full vs. 200-bp” scenario, because in this situation we carried out a complete training procedure of the GRNN considering all full-length sequences. Of course, because the spectral representation of full-length and sequence fragments are different from each other, no sequence fragment used in the test set belong to the training set.

4 Conclusion

In this work, a modified version of the GRNN algorithm implementing different distance models for barcode sequence classification is presented. The GRNN classification performances have been assessed with regards to sequence sizes. Experimental trials have been carried out considering full-length sequences and sequence fragments that simulate a very common scenario in which only environmental samples are available. In the case of full-length sequences, 6 out of 8 distance models provided near perfect results, in terms of accuracy, precision and recall, with scores ranging between 100 % at Phylum level and 90 % at Species level. The same scores are reached using the k-NN classifier. Only correlation and cosine distance did not provide acceptable results. In the case of sequence fragments, fractional and city block distances only gave meaningful results: in the “Full vs. 200-bp” scenario, accuracy ranged from 85 % at Phylum level to 57 % at Species level; in the “200-bp” scenario, accuracy ranged from 95 to 100 % at Phylum level to 70–79 % at Species level. In both scenarios our GRNN approach outperformed the k-NN classifier. That means GRNN implementing fractional and city block distances was able to correctly predict the similarity between original full-length sequences and their corresponding sequence fragments. All the other distance model were affected by a strong classification performance drop.

References

1. Aggarwal, C., Hinnenburg, A., Keim, D.: On the surprising behavior of distance metrics in high dimensional space. In: Van den Bussche, J., Vianu, V. (eds.) Database Theory ICDT 2001. Lecture Notes in Computer Science, vol. 1973, pp. 420–434. Springer, Berlin/Heidelberg (2001)
2. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theory **13**(1), 21–27 (1967)
3. Fiannaca, A., La Rosa, M., Rizzo, R., Urso, A.: Analysis of DNA barcode sequences using neural gas and spectral representation. In: Iliadis, L., Papadopoulos, H., Jayne, C. (eds.) Engineering Applications of Neural Networks. Communications in Computer and Information Science, vol. 384, pp. 212–221. Springer, Berlin/Heidelberg (2013)
4. Francois, D., Wertz, V., Verleysen, M.: The concentration of fractional distances. IEEE Trans. Knowl. Data Eng. **19**(7), 873–886 (2007)

5. Hajibabaei, M., Singer, G.A.C., Hebert, P.D.N., Hickey, D.A.: DNA barcoding: how it complements taxonomy, molecular phylogenetics and population genetics. Trends Genet. **23**(4), 167–172 (2007)
6. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software. ACM SIGKDD Explorations Newsletter **11**(1), 10–18 (2009)
7. Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd edn. Prentice Hall, Upper Saddle River (1998)
8. Hebert, P.D.N., Ratnasingham, S., DeWaard, J.R.: Barcoding animal life: cytochrome c oxidase subunit 1 divergences among closely related species. Proc. R. Soc. Ser. B, Biol. Sci. **270 Suppl**, S96–S99 (2003)
9. Hinnenburg, A., Aggarwal, C., Keim, D.: What is the nearest neighbor in high dimensional spaces? In: Proceedings of the 26th International Conference on Very Large Data Bases, VLDB '00, pp. 506–515. Morgan Kaufmann, San Francisco (2000)
10. Kuksa, P., Pavlovic, V.: Efficient alignment-free DNA barcode analytics. BMC Bioinf. **10**(Suppl.14), S9 (2009)
11. La Rosa, M., Fiannaca, A., Rizzo, R., Urso, A.: A study of compression-based methods for the analysis of barcode sequences. In: Peterson, L.E., Masulli, F., Russo, G. (eds.) Computational Intelligence Methods for Bioinformatics and Biostatistics. Lecture Notes in Computer Science, vol. 7845, pp. 105–116. Springer, Berlin/Heidelberg (2013)
12. La Rosa, M., Fiannaca, A., Rizzo, R., Urso, A.: Alignment-free analysis of barcode sequences by means of compression-based methods. BMC Bioinf. **14**, S4 (2013)
13. Marshall, E.: Taxonomy. Will DNA bar codes breathe life into classification? Science (New York, N.Y.) **307**(5712), 1037 (2005)
14. Meusnier, I., Singer, G.A.C., Landry, J.F., Hickey, D.A., Hebert, P.D.N., Hajibabaei, M.: A universal DNA mini-barcode for biodiversity analysis. BMC Genomics **9**, 214 (2008)
15. Rach, J., Desalle, R., Sarkar, I.N., Schierwater, B., Hadrys, H.: Character-based DNA barcoding allows discrimination of genera, species and populations in Odonata. Proc. Biol. Sci. R. Soc. **275**(1632), 237–247 (2008)
16. Ratnasingham, S., Hebert, P.D.N.: Bold: the barcode of life data system (http://www.barcodinglife.org). Mol. Ecol. Notes **7**(3), 355–364 (2007)
17. Rizzo, R., Fiannaca, A., La Rosa, M., Urso, A.: The general regression neural network to classify barcode and mini-barcode DNA. In: Computational Intelligence Methods for Bioinformatics and Biostatistics. Lecture Notes in Computer Science. Springer, Berlin/Heidelberg (2015)
18. Scholkopf, B., Smola, A.: Learning with Kernels. MIT, Cambridge (2002)
19. Specht, D.F.: A general regression neural network. IEEE Trans. Neural Netw. **2**(6), 568–576 (1991)
20. Weitschek, E., Van Velzen, R., Felici, G., Bertolazzi, P.: BLOG 2.0: a software system for character-based species classification with DNA barcode sequences. What it does, how to use it. Mol. Ecol. Resour. **13**(6), 1043–1046 (2013)