AIMS Medical Science

http://www.aimspress.com/journal/medicalScience/

AIMS Medical Science, 5(1): 23–32.
DOI:10.3934/medsci.2018.1.23
Received: 12 May 2017
Accepted: 24 November 2017
Published: 23 December 2017

Research article

Bacteria classification using minimal absent words

Gabriele Fici[1,*], **Alessio Langiu**[2,3], **Giosuè Lo Bosco**[1] **and Riccardo Rizzo**[3]

[1] Dipartimento di Matematica e Informatica, Università di Palermo, Palermo, Italy
[2] Department of Informatics, King's College London, London, UK
[3] Istituto di Calcolo e Reti ad Alte Prestazioni, Consiglio Nazionale delle Ricerche, Palermo, Italy.

* **Correspondence:** email: gabriele.fici@unipa.it; Tel: +39 09123891130.

Abstract: Bacteria classification has been deeply investigated with different tools for many purposes, such as early diagnosis, metagenomics, phylogenetics. Classification methods based on ribosomal DNA sequences are considered a reference in this area. We present a new classificatier for bacteria species based on a dissimilarity measure of purely combinatorial nature. This measure is based on the notion of Minimal Absent Words, a combinatorial definition that recently found applications in bioinformatics. We can therefore incorporate this measure into a probabilistic neural network in order to classify bacteria species. Our approach is motivated by the fact that there is a vast literature on the combinatorics of Minimal Absent Words in relation with the degree of repetitiveness of a sequence. We ran our experiments on a public dataset of Ribosomal RNA Sequences from the complex 16S. Our approach showed a very high score in the accuracy of the classification, proving hence that our method is comparable with the standard tools available for the automatic classification of bacteria species.

Keywords: bacteria classification; supervised classification; probabilistic neural network; minimal absent word; combinatorics on words

1. Introduction

Bacteria classification has recently attracted a lot of interest since the starting of Human Microbiome Project (HMP) from the US National Institutes of Health in 2008 [1]. This led in particular to the development of complex bioinformatics software pipelines whose kernel is the bacteria classification algorithm. Although several classifiers based on classical sequence representations are known, e.g., row fasta sequences, symbol frequency histograms, k-mers or q-grams (see, for example [2–5]) only very few contributions are primarily based on purely combinatorial properties of sequences (see, for example [6]).

Among the methodologies that researchers considered for automatic sequence classification, neural networks belong to the most widely used, due to their performances (see, e.g., [7]). In particular, Probabilistic Neural Networks (PNN) [8] constitute a probabilistic model simulating the bayesian classifier, which represents the baseline model for classification.

In this paper we use a dissimilartity measure among sequences based on a purely combinatorial tool, Minimal Absent Words. A sequence U is called a Minimal Absent Word for a sequence S if U is a minimal element in the set of sequences that do not appear in S. From this notion one can derive a dissimilarity measure between two genomic sequences based on the symmetric difference of their sets of MAWs [9]. We incorporated this measure into a PNN in order to classify bacteria species. For our experiments, we used a public dataset of Ribosomal RNA Sequences from the complex 16S. We compared the PNN with the K-Nearest Neighbor (KNN) and the Ribosomal Database Project (RDP) classifiers. The results we obtained show that our approach gives a very high score in the accuracy of the classification, as detailed in the section devoted to the experimental results.

2. Materials and method

We devise a bacteria classification algorithm, using a PNN incorporating the matrix of mutual dissimilarity among the sequences in a dataset. The dissimilarity measure is computed comparing the sets of Minimal Absent Words (MAWs) of two given sequences. We then evaluate the classification performances with the 10-fold cross-validation at every available taxon level in the dataset.

2.1. The dataset

Studies about bacteria species are generally based on the analysis of their 16S ribosomal RNA housekeeping gene. Ribosomes act for the protein synthesis in each living organism, and are mainly composed by two subunits that share a special kind of RNA, called ribosomal RNA (rRNA). The first of these two subunits contains the so-called 16S ribosomal RNA. The presence of hyper variable regions in the 16S rRNA gene provides a species signature sequence which is useful for bacterial identification. Moreover, the 16S rRNA gene is very short (around 1,500 nucleotide bases) so that it can be easily copied and sequenced. The use of 16S rRNA gene sequences to study bacterial phylogeny and taxonomy has been by far the most common housekeeping genetic marker and is currently widely used for several reasons, including its presence in almost all bacteria and its unchanged role function over evolution.

In our experiments, we used a public dataset of Ribosomal RNA Sequences from the complex 16S, the RDP Ribosomal Database Project II repository [10], release 10.27. We randomly selected 3,000 sequences among the sequences of good quality (according to the RDP), i.e., they are the best representatives of their own species. Every sequence in the dataset is tagged with a five-level taxon identifier reporting Phylum, Class, Order, Family, and Genus, respectively. Over all the 3,000 sequences in the dataset, there are 3 different bacteria species in the Phylum taxon, 6 in Class, 22 in Order, 65 in Family and 393 in Genus.

2.2. Minimal absent words

The dissimilarity measure we use in our approach is based on the notion of Minimal Absent Words (also known as Minimal Forbidden Words or Minimal Forbidden Factors), a combinatorial tool introduced for studying properties of strings and sequences. Minimal Absent Words (MAWs) have already found applications in several areas of Computer Science, ranging from symbolic dynamics to string processing (for more details the reader may see [11] and references therein). The theory of MAWs is well developed, both from the combinatorial and the algorithmic point of view. Indeed, MAWs are at the basis of efficient algorithms for manipulating strings in several domains, including text compression [12] and bioinformatics [9, 13, 14]. There exist several efficient algorithms for computing the set $\mathbf{MAW}(S)$ of MAWs of a sequence S [13, 15–17]. Although for a given sequence the cumulative length of all its MAWs can be quadratic with respect to the length of the sequence, the whole set of MAWs can be represented on a trie whose size is linear in the length of the sequence S, thus allowing an efficient algorithmic treatment of the information this set contains.

It is worth noticing – even if we do not use this property in the present paper – that a sequence can be uniquely retrieved from its set of MAWs, and there exist linear-time algorithms performing this operation [18]. In particular, this shows that the set of MAWs of a sequence contains all the information of that sequence.

Formally, given a sequence S, we say that a sequence U is a MAW for S if U does not occur in S but so do the sequences U' and U'' obtained from U removing the first (resp. last) character. For example, the set of MAW of the string $S = AGCTCTCA$ over the alphabet $\Sigma = \{A, T, G, C\}$ is

$$\mathbf{MAW}(S) = \{TT, TG, TA, CC, CG, GT, GG, GA, AT, AC, AA, CAG, GCA, TCTCT, GCTCA\}.$$

2.3. sMAW

In [9], Chairungsee and Crochemore introduced a measure of dissimilarity between two sequences based on the sets of MAWs of the two sequences. They made use of a length-weighted index to measure the dissimilarity between two sequences using sample sets of their MAWs, by considering the length of each member in the symmetric difference of these sample sets. This measure can be trivially computed in time and space $O(m+n)$, where m and n are the lengths of the two sequences, respectively, provided that these sample sets contain MAWs of some bounded length ℓ. For unbounded length, the same measure can be trivially computed in time $O(m^2 + n^2)$ since, as already mentioned, for a given sequence, the cumulative length of all its MAWs can grow quadratically on the length of the sequence. However, in [14] the authors presented an $O(m + n)$-time and $O(m + n)$-space algorithm to compute the dissimilarity measure introduced by Chairungsee and Crochemore by considering *all* minimal absent words of the two sequences; thereby showing that it is indeed possible to compare two sequences in time proportional to their lengths.

Formally, given two sequences S and T, we define the set $\mathbf{SMAW}(S, T)$ as the symmetric difference $\mathbf{MAW}(S) \triangle \mathbf{MAW}(T)$ of the sets $\mathbf{MAW}(S)$ and $\mathbf{MAW}(T)$, that is, the set of sequences that are MAWs for S or for T but not for both. The dissimilarity measure introduced by Chairungsee and Crochemore is defined by

$$\mathrm{sMAW}(S, T) = \sum_{U \in \mathbf{SMAW}(S,T)} \frac{1}{|U|^2}$$

where $|U|$ is the length of the sequence U.

From this definition, it follows that the smaller is the value of $\mathrm{sMAW}(S, T)$ the more similar are the sequences S and T. As already mentioned, the value of $\mathrm{sMAW}(S, T)$ can be computed in time and space linear on the lengths of S and T [14]. An implementation of this algorithm is contained in the MAW tool, available online at `https://github.com/solonas13/maw` [19]. The MAW tool also provides an implementation of the algorithm for computing the matrix of mutual dissimilarities of a given set of sequences based on the measure sMAW. We used this implementation in our experiments.

2.4. Probabilistic neural network

The Probabilistic Neural Network (PNN) [8] is a special kind of neural network. Differently from a Multi Layer Perceptron, in the PNN there is only one hidden layer, where each of the neural units computes a gaussian on the distance between a prototype and an input element. The classification result is the output of an argmax rule (see Figure 1) that is computed by the output layer.

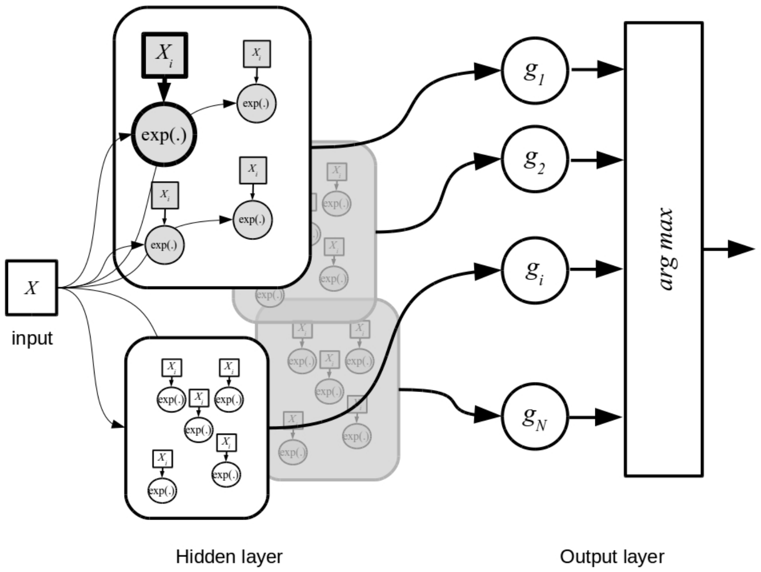

Figure 1. The representation of the PNN. The hidden layer computes a gaussian on the distance between a prototype and an input element. The terms g_i, one for each training class, are used to compute the output.

The first layer of the PNN takes into account each training input in a neural cell, so that the number of hidden units is the same as the training samples. When a new input $\mathbf{x}'$ is submitted to the network, the first layer computes a value that represents the activation $g_i(\mathbf{x}')$ of the single cell for the input $\mathbf{x}'$. These activations are non-linear functions of the distance between the input and the training samples.

The second layer of the network gathers these contributions for each class of the training set and produces an output activation value for each of the N classes. Assuming that $\mathbf{x}_{i,k}$ is the k-th training sample of the class i ($i = 1, 2, \ldots, N$), when a new input $\mathbf{x}'$ is presented to the network the output for the class i is computed by the equation

$$g_i(\mathbf{x}') = \frac{1}{n_i} \sum_{k=1}^{n_i} \exp \frac{-dist(\mathbf{x}', \mathbf{x}_{i,k})^2}{\sigma^2} , \tag{1}$$

where n_i is the number of training samples for the class i, and σ, called *spread factor*, is a parameter of the PNN network. The term $g_i(\mathbf{x}')$ represents the excitation level of the neural unit corresponding to the class i generated by the input $\mathbf{x}'$.

The last layer (output layer) compares all the output from the second layer and links the activation functions g_i to the class with the maximum activation signal. The class c of the new input $\mathbf{x}'$ can be computes as

$$c(\mathbf{x}') = \arg\max_i \{g_i(\mathbf{x}'), i = 1, 2, \ldots, N\}. \tag{2}$$

This network only needs:

- the spread factor σ, which regulates each contribution;
- the computation of the dissimilarity among all the training samples.

The value of the spread factor σ can be a single value for the whole network or a particular value for each layer [21].

According to equation (2) the network will classify an unknown sequence in one of the existing classes.

The main advantage of the PNN is that it does not need any training phase, since the algorithm is based on the memorization of all the training samples in the hidden layer, one neural unit for each training sample (see Figure 1). The computational cost of the method therefore depends only on the number of training samples.

3. Testing procedure

In our experiments, we used the MAW tool [19] to compute the matrix of mutual dissimilarities among all the pairs of sequences in the dataset, according to the sMAW dissimilarity measure defined above. Figure 2 shows how dense is the dissimilarity space inside the intra Phylum region.

The MAW dissimilarity matrix constitutes the only input for our PNN. We used a different PNN for the classification of bacteria in each of the taxa levels. We evaluated at first the impact of the σ parameter of the neural network. In Figures 3(a) and 3(b) we report the accuracy of the PNN for different values of σ. The classifier shows good and stable accuracy for values of σ between 4 and 8, with the best performance for the value 8. The accuracy values have been computed as the average of all the runs of a Leave-One-Out procedure, i.e., we presented all the sequences in the dataset, one by one, to the PNN, which used the knowledge of all the other sequences to identify the class of the input sequence.

We ran experiments comparing two different dissimilarity measures, the sMAW dissimilarity and the euclidean distance computed on the q-gram representations of the sequences (see below for details). Each measure has been incorporated into a PNN and into a KNN. The rationale is to show that the sMAW can be more useful, in the special case of bacteria classification, than a classical euclidean distance.

Moreover, we aim at showing that sMAW can lead to the design of classifiers that achieve performances no far from those of classifiers based on classical sequence representations, such as row fasta sequences, symbol frequency histograms, k-mers or q-grams [2–4]. In our trial, we used the q-gram representation with $q = 5$ and computed the euclidean mutual distance matrix. In the q-gram representation, a sequence is mapped to a vector of length $|\Sigma|^q$ (4^5 in our case) representing the frequencies of each of the q-grams of the sequence.

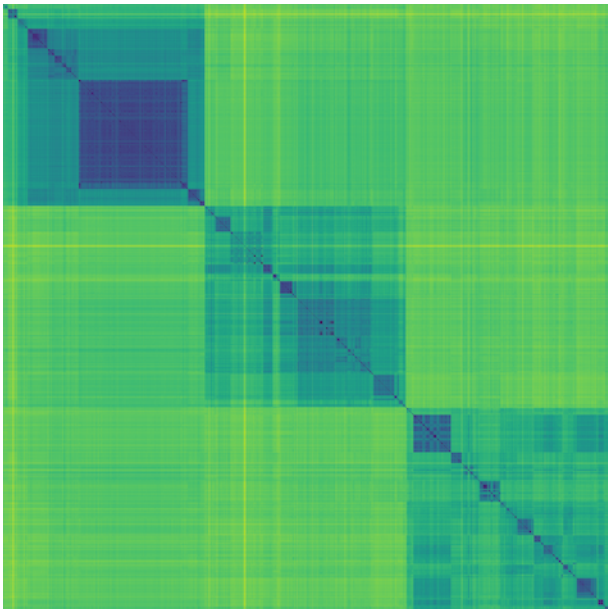

Figure 2. The plot of the MAW dissimilarity matrix. Darkest pixels on the principal diagonal represent zero values, i.e., the identity.

Furthermore, we compared our classifier against an RDP Naive Bayes Classifier [20], since this is a commonly used method for bacteria ribosomal sequence classification. The KNN was used as a baseline classifier. For the number of neighbors, K, we used the same value obtained from the Leave-One-Out validation process that we used for the spread factor σ of the PNN (see above), i.e., we set $K = 5$.

4. Results

In Figures 4(a) and 4(b) we show the accuracies of the three classifiers for the euclidean (a) and the sMAW distance (b). Note that the RDP classifier does not make any use of dissimilarities, but is reported in the figures for comparison purposes. The results have been computed using a 10-fold cross-validation scheme so that the histogram bars show the mean values over all the 10 runs. In particular, the whole dataset of sequences X was split into 10 equal-sized subsets S_i, $i = 1, .., 10$. At each run i, the classification method was retrained on $X \setminus S_i$ and the accuracy was then determined based on the performance in the test set S_i. We also computed the standard deviation, shown in red above each bar. The dashed black lines show the mean accuracies of the considered classifiers all over the taxa. The results show that the sMAW is more accurate than the euclidean distance in order to classify bacteria species (see Table 1 for details), giving also the possibility to boost up the PNN classifier in order to reach the RDP performances.

We also computed the running times of the above experiments (see Table 2) in order to give a rough idea about the computationally efficiency of the PNN-based classifier with respect to the RDP. For all the experiments, we used a standard laptop PC. The PNN and the KNN do not have a training phase. The test was done by computing the matrix of the mutual dissimilarity measures between a fold of 300 sequences and the remaining 2,700 sequences. The PNN-based classifier, as well as the KNN-based

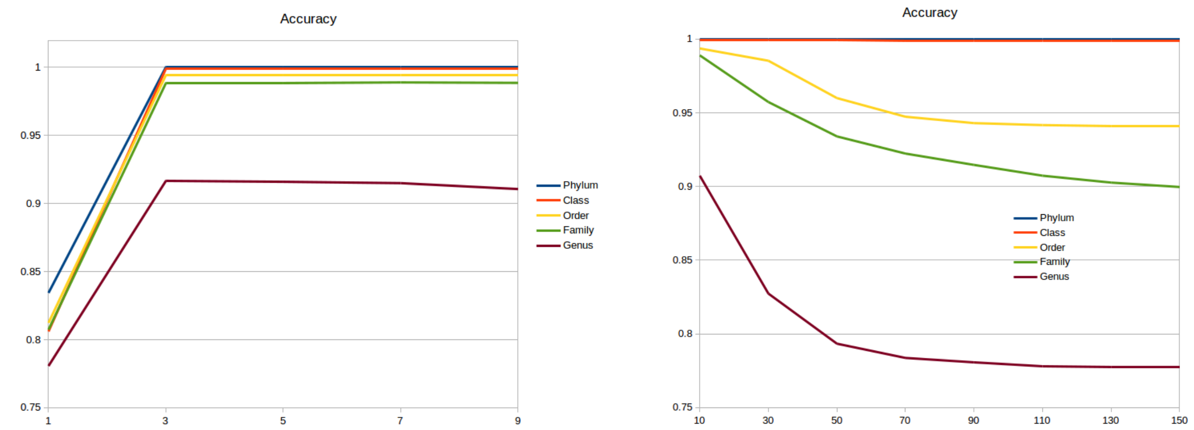

Figure 3. The accuracy of the classification system for values of σ ranging between 1 and 9 (a) and between 10 and 150 (b).

Table 1. Accuracy values of the three classifiers with euclidean and sMAW distances.

	Phylum	Class	Order	Family	Genus
Knn-euclidean	0.99	0.99	0.98	0.96	0.81
Pnn-euclidean	0.99	0.98	0.96	0.94	0.69
Knn-sMAW	1.00	0.99	0.99	0.98	0.87
Pnn-sMAW	1.00	0.99	0.99	0.98	0.91
RDP	1.00	0.99	0.99	0.98	0.91

one, was about 10 times slower than the RDP in the testing phase, overall running in a very reasonable time on the chosen dataset. However, our implementation has not been optimized, hence the time comparison is provided only as an estimate of the order of magnitude for the execution times of the presented experiments.

Table 2. Running time comparison of KNN, PNN, and RDP classifiers.

	KNN	PNN	RDP
Training over $2,700$ seq.	0s	0s	46s
Test on 300 seq.	3m 24s	3m 23s	21s
Overall 10-fold experiment	33m 54s	33m 53s	9m 25s
Test time per sequence	680ms	677ms	71ms

The PNN incorporating the sMAW dissimilarity measure shows good performances in solving the classification problem at every taxon level, compared to the other two classification methods considered (KNN and RDP). As a result of our experiments, we have that the PNN performance is the same as the RDP. Moreover, the performances of the classifiers with the sMAW measure are better than with the euclidean distance, showing the interest of using more sophisticated combinatorial tools in this problem. Table 1 shows the accuracy values of the considered classifier, where it is possible to observe the improvements with respect to the euclidean distance when using the sMAW dissimilarity.

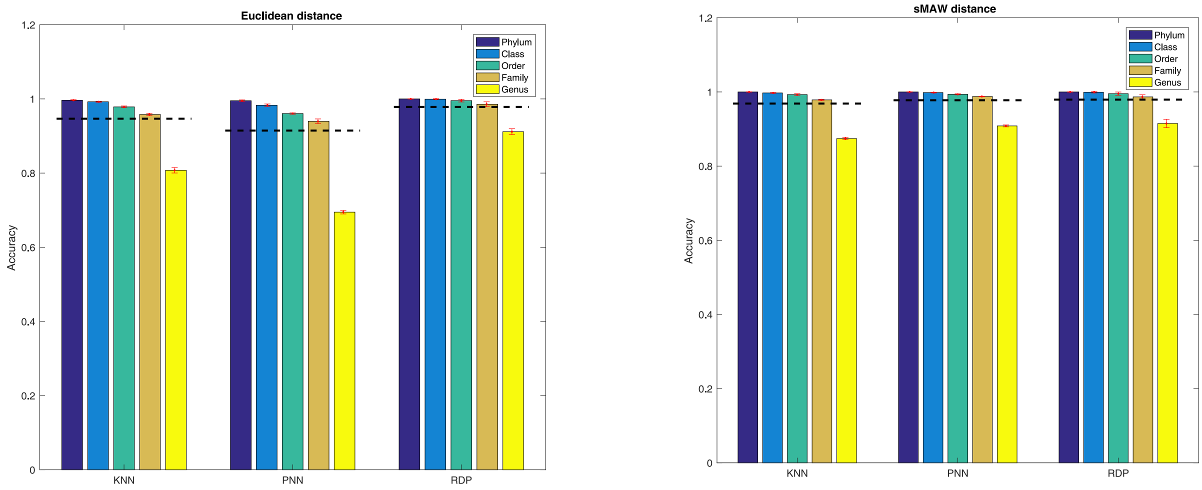

Figure 4. Accuracies of the PNN and KNN classifiers on the different taxa, when adopting euclidean (a) and sMAW (b) distances. The accuracies of RDP classifiers are also reported in each figure. The standard deviation is shown in red above each bar. The dashed black lines show the average performance over the taxa.

5. Conclusions

In this paper we devised a classifier for 16S bacterial genomic sequences based on the sMAW dissimilarity measure, i.e., a mutual dissimilarity measure for sequences based on a purely combinatorial definition. This measure takes into account the length and the number of MAWs (Minimal Absent Words) of two given sequences.

We obtained very good results (over 98%) in the first four taxa levels and good results (about 91%) at the Genus level. The main aim of this paper was to provide an example of a concrete application of the theoretical background on MAWs to a biological classification system. For this, it should mainly be considered as an exploratory approach. From the practical point of view, the devised system may have a point of weakness represented by the size of the dataset. Indeed, in a scenario in which a new sequence has to be presented to the classifier, there is the need to compute the dissimilarity of this new sequence with respect to every sequence already present in the dataset. The computation of such dissimilarity values takes time proportional to the size of the dataset. This observation leads us to ask for an efficient method for selecting a subset of the dataset, the smaller the better, for which the performances in terms of accuracy do not decrease significantly. It is desirable to fix the minimum accuracy level to be satisfied and then proceed by reducing the dataset. These two considerations may lead to a valuable classifier usable in many practical scenarios.

In conclusion, we showed that a sophisticated combinatorial property of sequences, namely their sets of MAWs, can be efficiently exploited in a classical bacteria classification problem. The results presented here may constitute the basis for further investigations on the role that MAWs can play in the automatic classification of DNA sequences.

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1 Nelson KE, Weinstock GM (2010) A catalog of reference genomes from the human microbiome. *Science* 328: 994–999.

2 Soueidan H, Nikolski M (2016) Machine learning for metagenomics: methods and tools. *Metagenomics* 1: 1–19.

3 La Rosa M, Fiannaca A (2015) Probabilistic topic modeling for the analysis and classification of genomic sequences. *BMC Bioinformatics* 16.

4 Lo Bosco G, Pinello L (2015) A new feature selection methodology for K-mers representation of DNA sequences. In *CIBB 2015* : 8623 of *LNCS*, 99–108.

5 Pinello L, Lo Bosco G, Hanlon B, et al. (2011) A motif-independent metric for DNA sequence specificity. *BMC Bioinformatics* 12.

6 Ferragina P, Giancarlo R, Greco V, et al. (2007) Compression-based classification of biological sequences and structures via the Universal Similarity Metric: experimental assessment. *BMC Bioinformatics* 8: 252.

7 Fiannaca A, La Rosa M, Rizzo R, et al. (2015) A *k*-mer-based barcode DNA classification methodology based on spectral representation and a neural gas network. *Artificial Intelligence Med* 64: 173–184.

8 Specht DF (1990) Probabilistic neural networks. *Neural Networks* 3: 109–118.

9 Chairungsee S, Crochemore M (2012) Using minimal absent words to build phylogeny. *Theoretical Computer Sci* 450: 109–116.

10 Cole JR, Wang Q, Cardenas E, et al. (2009) The Ribosomal Database Project: improved alignments and new tools for rRNA analysis. *Nucleic Acids Res* 37: D141–D145.

11 Fici G (2006) *Minimal Forbidden Words and Applications*. PhD thesis, Université de Marne-la-Vallée.

12 Crochemore M, Mignosi F, Restivo A, et al., (1999) Text compression using antidictionaries. In *ICALP 1999* Volume: 1644 of *LNCS*, 261–270.

13 Barton C, Héliou A, Mouchard L, et al. (2014) Linear-time Computation of Minimal Absent Words Using Suffix Array. *BMC Bioinformatics* 15: 388.

14 Crochemore M, Fici G, Mercaş R, et al. (2016) Linear-Time Sequence Comparison Using Minimal Absent Words & Applications. In *LATIN 2016* Volume: 9644 of *LNCS*, 334–346.

15 Pinho AJ, Ferreira PJSG, Garcia SP, et al. (2009) On finding minimal absent words. *BMC Bioinformatics* 10: 137.

16 Barton C, Héliou A, Mouchard L, et al. (2015) Parallelising the computation of minimal absent words. In *PPAM 2015*, Volume: 9574 of *LNCS*, 243–253. Springer.

17 Fukae H, Ota T, Morita H (2012) On fast and memory-efficient construction of an antidictionary array. In: *ISIT 2012*, 1092–1096. IEEE.

18 Crochemore M, Mignosi F, Restivo A (1998) Automata and forbidden words. *Information Processing Letters* 67: 111–117.

19 **Online content:** MAW: a suite on the computation and application of Minimal Absent Words. Available from: `https://github.com/solonas13/maw.`

20 Wang Q, Garrity GM, Tiedje JM, et al. (2007) Naive Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy, *Applied Environmental Microbiology* 73: 5261–5267.

21 Mao KZ, Tan KC, Ser W (2000) Probabilistic neural-network structure determination for pattern classification IEEE Transactions on neural networks. 11: 1009–1016.