Neural Computing and Applications (2022) 34:67–78
https://doi.org/10.1007/s00521-021-06018-2

S.I. : WSOM 2019

Learning vector quantization as an interpretable classifier for the detection of SARS-CoV-2 types based on their RNA sequences

Marika Kaden[1,2] · Katrin Sophie Bohnsack[1,2] · Mirko Weber[1,2] · Mateusz Kudła[1,3] · Kaja Gutowska[3,4,5] · Jacek Blazewicz[3,4,5] · Thomas Villmann[1,2]

Received: 13 July 2020 / Accepted: 7 April 2021 / Published online: 27 April 2021

Abstract
We present an approach to discriminate SARS-CoV-2 virus types based on their RNA sequence descriptions avoiding a sequence alignment. For that purpose, sequences are preprocessed by feature extraction and the resulting feature vectors are analyzed by prototype-based classification to remain interpretable. In particular, we propose to use variants of learning vector quantization (LVQ) based on dissimilarity measures for RNA sequence data. The respective matrix LVQ provides additional knowledge about the classification decisions like discriminant feature correlations and, additionally, can be equipped with easy to realize reject options for uncertain data. Those options provide self-controlled evidence, i.e., the model refuses to make a classification decision if the model evidence for the presented data is not sufficient. This model is first trained using a GISAID dataset with given virus types detected according to the molecular differences in coronavirus populations by phylogenetic tree clustering. In a second step, we apply the trained model to another but unlabeled SARS-CoV-2 virus dataset. For these data, we can either assign a virus type to the sequences or reject atypical samples. Those rejected sequences allow to speculate about new virus types with respect to nucleotide base mutations in the viral sequences. Moreover, this rejection analysis improves model robustness. Last but not least, the presented approach has lower computational complexity compared to methods based on (multiple) sequence alignment.

Keywords Learning vector quantization · Interpretable models · Genomic sequence analysis · Reject options

1 Introduction

The coronavirus disease 2019 (COVID-19) caused by SARS-CoV-2 viruses, whose origin lies probably in Wuhan (China), is a severe respiratory disease [1]. Currently (May 2020), it is spreading rapidly all over the world [88]. Yet there are several indicators that its molecular characteristics evolve during time [2, 89]. This evolution is mainly driven by mutations, which play an essential role and may be accompanied by mechanisms of stabilization [70, 71].

Marika Kaden and Thomas Villmann contributed equally to this work.

Katrin Sophie Bohnsack, Mirko Weber, Mateusz Kudla, Kaja Gutowska and Jacek Blazewicz also contributed equally to this work.

✉ Thomas Villmann
thomas.villmann@hs-mittweida.de

[1] University of Applied Sciences Mittweida, Technikumplatz 17, 09648 Mittweida, Germany

[2] Saxon Institute for Computational Intelligence and Machine Learning, Technikumplatz 17, 09648 Mittweida, Germany

[3] Institute of Computing Science, Poznan University of Technology, Piotrowo 2, 60-965 Poznan, Poland

[4] Institute of Bioorganic Chemistry, Polish Academy of Sciences, Noskowskiego 12/14, 61-704 Poznan, Poland

[5] European Centre for Bioinformatics and Genomics, Piotrowo 2, 60-965 Poznan, Poland

Therefore, an analysis of virus sequences is essential to understand the spreading and the behavior of the virus population. One aspect is to distinguish several types of the virus, which may force different symptoms and medical conditions. Thus, sequences have to be compared regarding their genomic structure. This can be done by alignment methods or by alignment-free approaches, both coming with pros and cons. Further, the sequences have to be distinguished or classified with respect to their virus types. For this purpose, interpretable models are favored in comparison with black-box approaches like deep networks, because a medical interpretation of the classification decision process is highly desirable. In fact, this could help to detect new virus variants.

1.1 Biological basics regarding SARS-CoV-2

The analysis of the genomic structure by sequencing is currently topic of ongoing research to better understand the molecular dynamics [53]. Obviously, changing the genomic structure may cause new properties and, hence, could increase the difficulties in finding drugs for treatment. For example, changes may lead to behavioral changes, such as the increased binding of the SARS-CoV-2 surface glycoprotein to human ACE2 receptors [37].

Viruses of the family *Coronaviridae* possess a single-stranded, positive-sense RNA genome ranging from 26 to 32 kilobases in length and frequently are extremely similar [44]. Therefore, the analysis of those sequences to understand the genetic evolution in time and space is very difficult. This problem is magnified by incorrect or inaccurate sequencing [75]. Further, mutations are not equally distributed across the SARS-CoV-2 genome [28]. The molecular differences in corona virus populations were investigated using phylogenetic trees so far resulting in three clusters which are identified as virus types [23]. Yet, SNP-based radial phylogeny-retrieved trees of SARS-CoV-2 genomes result in five major clades [28]. Generally, a disadvantage of those decision-tree-like approaches is the problem of out-of-sample considerations, i.e., new data cannot easily be integrated [55, 86]. The respective tree has to be reconfigured completely, which frequently leads to major changes in the tree structure [56, 72].

Frequent mutations in SARS-CoV-2 genomes are in the genes encoding the S-protein and RNA polymerase, RNA primase, and nucleoprotein. Applying a sequence alignment and similarity comparison using the Jaccard index, a method for monitoring and tracing SARS-CoV-2 mutations was established in [90]. However, a general mathematical evaluation of similarities is crucial because respective similarity measures only partially reflect all biological aspects of similarity between RNA sequences [87].

Alignment-based methods usually rely on variants of the Levenshtein distance [38], which, however, are computationally costly: $O(l_1 \cdot l_2)$ is the time complexity for both the Needleman–Wunsch algorithm [51] and for the Smith–Waterman algorithm [26, 68], where l_1 and l_2 are the sequence lengths. Hence, if $l_1 = l_2 = l$, the complexity is simply $O(l^2)$. Both approaches solve internally a mathematical optimization problem, i.e., both algorithms belong to the algorithmic class of dynamic programming with high computational complexity.

In case of multiple sequence alignments (MSAs), the dissimilarity problem is NP-hard [31]. Currently used MSA implementations such as ClustalW [73], MAFFT [33], or MUSCLE [18] therefore rely on the progressive alignment technique [20], which reduces the computational complexity to polynomial time [47]. In the example of MUSCLE, the time complexity amounts to $O(N^4 + N \cdot l^2)$ with N being the number of sequences and l is the uniform sequence length. Other alignment-based methods for SARS-CoV-2 data consider (multiple) longest common subsequences with similar complexity [41].

Therefore, alignment-free alternatives are promising to avoid this algorithmic complexity [7, 8, 83, 84, 87, 91]. Commonly used approaches are *Bag-of-Words* (BoW [67]), information theoretic methods based on the *Kolmogorov–Smirnov complexity* [35] and the related *Normalized Compression Distance* [13, 40]. Recently, similarities based on *Natural Vectors* gained attraction [17, 42, 92]. These methods have in common that the sequences are considered in terms of their statistical properties and distributions of the nucleotides. However, local information like precise nucleotide positions as well as specific motifs is lost. An overview of prominent measures and their behavior for sequence analysis can be found in [94, 95]. The time complexity for this data coding is only $O(N \cdot l)$ and, hence, much lower than for alignment methods.

In the present publication, we investigate whether alignment-free dissimilarities are suitable for the identification of SARS-CoV-2 clusters/classes in combination with *interpretable machine learning methods* for clustering and classification [4, 5]. This we do for two datasets: GISAID data and NCBI data, see Sect. 2.1. For the first one, virus classes (types) were identified by phylogenetic tree analysis in [23], whereas the second one is without class information.

1.2 Motivation to use an interpretable classifier

Although deep neural network approaches provide impressive results in sequence classification [9, 21, 69, 72], deep architectures are at least difficult to interpret. Therefore, many attempts are made to explain deep architectures

[59]. However, it is claimed that restricting models to be interpretable does not necessarily lead to weaker performance and, hence, should be favored if possible [58, 82]. Moreover, particularly in the medical domain, knowledge regarding decision processes is strongly required for correct interpretation of the results [76].

Therefore, we focus on applying prototype-based methods using alignment-free dissimilarity measures for sequence comparison. In fact, prototype-based machine learning models for data classification and representation are known to be interpretable and robust [6, 82, 93]. Using such methods for the SARS-CoV-2 sequence data, first we verify the classification results for the GISAID data. In particular, we classify the sequences by a learning vector quantizer, which is proven to be robust and interpretable [60, 82]. Thereafter, we use this model to classify the new data from the NCBI. Moreover, this interpretable classifier provides correlation information regarding data features contributing to a class discrimination. This additional knowledge allows a further characterization of the virus classes. Additionally, the model is equipped with a reject option following [22]. This allows to refuse outliers by the model, which could give hints for new virus types.

2 Materials and methods

2.1 SARS-CoV-2 sequence databases in use

In order to investigate SARS-CoV-2 viruses in terms of sub-type spreading, two virus sequence datasets were considered.

2.1.1 The GISAID dataset D_G

The first one, abbreviated by D_G, is from the GISAID coronavirus repository (GISAID—Global Initiative on Sharing Avian Influenza Data). It consists by March 4, 2020, of 254 coronavirus genomes, isolated from 244 humans, nine Chinese pangolins, and one bat *Rhinolophus affinis*. After preprocessing, 160 complete human sequences are obtained as described in [23], where these genomes of SARS-CoV-2 have been used to create a phylogenetic network. The resulting network analysis distinguished three types of the virus (cluster) $A, B,$ and C : $A$is most similar to the bat virus, whereas B and C are sequences obtained from A by two mutations: the synonymous mutation T8782C and the non-synonymous mutation C28144T changing a leucine to a serine. A further non-synonymous mutation G26144T changing a glycine to a valine lead from B to type C. In this sense, the classes (virus types) code implicitly the evolution in time of the virus.

In our data analysis, we removed two sequences, whose accession numbers occur twice in the data record, and another two, which we identified as not human resulting in 156 final sequences. Additionally, we take the type/class information as label for the virus genome sequences and, hence, as reference. A detailed data description as well as complete list of sequences can be found in [23]. The virus type assignments and additional data (country, collection date) as well as accession numbers for all 156 sequences in use are additionally provided in supplementary material.

The complete data information is found in supplementary files S12 Data.

2.1.2 The NCBI dataset D_N

The second dataset including 892 complete genomes has been selected from the National Center for Biotechnology Information (NCBI) Viral Genome database [10] and GenBank [14] by April 19, 2020, as given in Table 1. These data are human-based sequences and provide additionally the country information from which the sequences originate, as well as their collection date. For each sequence, we have also derived a more general assignment to regions based on the country information, which includes the following values: USA, China, Europe, and Others. The accession number and the additional data used in the analysis are included in supplementary material. We refer to this dataset by D_N.

Remark, although the SARS-CoV-2 virus is an RNA virus, the sequences provided by databases are given using the DNA coding. In the following, we take over this convention and do not explicitly refer to that later.

Again, the complete data information is found in supplementary files S12 Data.

Table 1 Distribution of the NCBI data D_N regarding regions and month of collection date

	China	Europe	USA	Others
December 2019	16	0	0	0
January 2020	44	4	16	9
February 2020	2	6	44	7
March 2020	1	23	706	10
April 2020	0	0	4	0

2.2 Representation of RNA sequences for alignment-free data analysis

Several approaches were published to represent sequences adequately for alignment-free comparison. These methods range from chaos game representation to standard unary coding or matrix representations. An overview is given in [84, 94, 95]. Here, we focus only on two of the most promising approaches—*Natural Vectors* and *Bag-of-Words*.

2.2.1 Natural vectors

Natural Vectors (NV) for nucleotide sequence comparison are based on a statistical sequence description for the distribution of nucleotide positions within a sequence $\mathbf{s} = [s_,\ldots,s_n]$ based on the alphabet $\mathscr{A} = \{A, C, G, T\}$ [17, 42]. Let $\mu_L^0 = n_L/n$ be the relative number (frequency) of the nucleotide $L \in \mathscr{A}$ and $p_L(j)/n$, $j = 1\ldots n_L$ is the *relative* position of the kth nucleotide L in the sequence. Let $E[r]$ further be the expectation operator of a random quantity r. With this convention, we get $\mu_L^0 = E[L]$ for the frequency of the nucleotide L. Further, we denote by $\mu_L = \mu_L^1 = E[p_L]$ the mean relative position of the nucleotide L in the sequence. The kth centralized moment μ_L^k for $k \geq 2$ is given as $\mu_L^k = E\left[\left(p_L - \mu_L^1\right)^k\right]$. Then, the natural vector of order K for a sequence $\mathbf{s}$ is defined as

$$\mathbf{x}(K,\mathbf{s}) = \left(\mu_A^0, \mu_C^0, \mu_G^0, \mu_T^0, \mu_A^1, \mu_C^1, \mu_G^1, \mu_T^1, \ldots, \mu_A^K, \mu_C^K, \mu_G^K, \mu_T^K\right) \tag{1}$$

whereby we again drop the dependencies on K and $\mathbf{s}$ for simplicity, if it is not misleading.

Natural vectors are usually compared in terms of the l_p-metric

$$d_p(\mathbf{x},\mathbf{y}) = \sqrt[p]{\sum_{j=0}^{K}\sum_{L\in\mathscr{A}}\left(\mu_L^j(\mathbf{x}) - \mu_L^j(\mathbf{y})\right)^p} \tag{2}$$

giving the Euclidean distance for $p = 2$. The Kendall statistics, as a kind of correlation measure, was used in [43].

The NV description of sequences can also be applied to nucleotide sequences containing ambiguous characters (degenerate bases) collected in the extension set $\mathscr{E}$ [15, 92]. This yields an extended alphabet $\mathscr{A}' = \mathscr{A} \cup \mathscr{E}$. In that case, weights $0 \leq w_L(s_i) \leq 1$ are introduced for each $L \in \mathscr{A}$ with

$$w_L(s_i) = \begin{cases} 1 & \text{if } s_i \in \mathscr{A} \wedge s_i = L \\ 0 & \text{if } s_i \in \mathscr{A} \wedge s_i \neq L \\ p_{L,s_i} & \text{otherwise} \end{cases}$$

where p_{L,s_i} is the probability that the detected ambiguous character $s_i \in \mathscr{E}$ should be the character L. These weights have to be taken into account during the expectation value calculations [92].

2.2.2 Bag-of-words

Another popular method to compare RNA/DNA sequences is the method *Bag-of-words* (BoW) based on 3-mers, where the set S of words contains all possible 64 triplets defined by the nucleotide alphabet $\mathscr{A} = \{A, C, G, T\}$ [7, 8, 21, 84]. Thus, all sequences $\mathbf{s}$ are coded as (normalized) histogram vectors of dimensionality $n = 64$, such that we have for each sequence the corresponding histogram vector $\mathbf{h}(\mathbf{s}) \in \mathbb{R}^n$ with the constraints $h_k(\mathbf{s}) \geq 0$ and $\sum_{k=1}^n h_k(\mathbf{s}) = 1$. Mathematically speaking, these vectors are discrete representations of *probability densities*. If the latter constraint is dropped, we have discrete representations of *positive functions*. The assignments of the triplets to the vector components h_i are provided in supplementary material. If it is not misleading, we drop the dependence on $\mathbf{s}$ and simply write $\mathbf{h}$ instead of $\mathbf{h}(\mathbf{s})$. As for NV, nucleotide sequences with ambiguous characters can be handled using appropriate expectation values.

Obviously, comparison of those histogram vectors can be done using the usual Euclidean distance. However, motivated by the already mentioned density property, an alternative choice is to compare them by means of divergence measures [46]. In the investigations presented later, we applied the Kullback–Leibler divergence [36]

$$D_{KL}(\mathbf{h},\mathbf{m}) = \sum_{j=1}^{n} h_j \cdot \log\left(h_j\right) - \sum_{j=1}^{n} h_j \cdot \log\left(m_j\right) \tag{3}$$

for sequence histograms $\mathbf{h}$ and $\mathbf{m}$. Note that the first term in (3) is the negative Shannon entropy $H(\mathbf{h}) = -\sum_{j=1}^n h_j \cdot \log\left(h_j\right)$, whereas $Cr(\mathbf{h},\mathbf{m}) = \sum_{j=1}^n h_j \cdot \log\left(m_j\right)$ is the Shannon *cross-entropy*. Yet, other divergences like Rényi divergences could be used [85]. We refer to [79] for a general overview regarding divergences in the context of machine learning.

The assignment of the nucleotide triplets to the histogram dimension is found in supplementary material S13 Histogram Coding of Nucleotide Triplets.

2.3 Machine learning approach for virus sequence data analysis

2.3.1 Median neural gas for data compression

The *Median Neural Gas* algorithm (MNG) is a neural data quantization algorithm for data compression based on (dis-

)similarities [3, 16]. It is a stable variant of the k-median centroid method improved by neighborhood cooperativeness enhanced learning, where k is the predefined number of representatives [39, 49]. In this context, median approaches only assume a dissimilarity matrix for the data and restrict the data centroids to be data points. Thus, after training, MNG provides k data points to serve as representatives of the data. Thereby, the data space is implicitly sampled according to the underlying data density in consequence of the so-called magnification property of neural gas quantizers [48, 78].

It should be emphasized that despite the weak assumption of a given similarity matrix, MNG always delivers exact data objects as representatives. Hence, any averaging for prototype generation like in standard vector quantizers is avoided here. This is essential, if averaged data objects are meaningless like for texts, music data, or RNA/DNA sequences, for example.

2.3.2 Affinity propagation for clustering with cluster cardinality control

Affinity propagation (AP) introduced by Frey and Dueck in [24] is an iterative cluster algorithm based on message passing where the current cluster nodes, in the AP setting denoted as prototypes or exemplars, interact by exchanging real-valued messages. Contrary to methods like c-means or neural maps, where the number c of prototypes has to be chosen beforehand, AP starts assuming that all N data points are potential exemplars and reduces the number of valid prototypes (cluster centroids) iteratively. More precisely, AP realizes an exemplar-dependent probability model where the given similarities $\varsigma(i,k)$ between data points $\mathbf{x}_i$ and $\mathbf{x}_k$ (potential exemplars) are identified as log-likelihoods of the probability that the data points assume each other as a prototype. For example, the similarities $\varsigma(i,k)$ simply could be negative dissimilarities like the negative Euclidean distance.

The cost function $C_{AP}(I)$ minimized by AP is given by

$$C_{AP}(I) = -\sum_i \zeta\left(\mathbf{x}_i, \mathbf{x}_{I(i)}\right) - \sum_j \delta_j(I)$$

where $I : N \to N$ is the mapping function determining the prototypes for each data point given by means of

$$I(i) = \arg\max_j \{a(i,k) + r(i,k)\}. \tag{4}$$

and

$$\delta_j(I) = \begin{cases} -\infty & \text{if } \exists\, j,k\; I(j) \neq j,\; I(k) = j \\ 0 & \text{otherwise} \end{cases}$$

is a penalty function. The quantity $r(i,k)$ is denoted as responsibility reflecting the accumulated evidence that point k serves as prototype for data point i. The availabilities $a(i,k)$ describe the accumulated evidence how appropriate data point k is seen as a potential prototype for the points i.

During the optimization, both kinds of messages are iteratively exchanged between the data by means of the alternating calculations according to

$$r(i,k) = \varsigma(i,k) - \max_{j \neq k}\{a(i,j) + \zeta(i,j)\}$$

and

$$a(i,k) = \min\left\{0, r(k,k) + \sum_{j \neq i,k} \max\{0, r(j,k)\}\right\}$$
$$a(k,k) = \max_{j \neq k}\{\max\{0, r(j,k)\}\}$$

until convergence. Finally, the prototypes are determined according to (4).

Hence, $a(i,k)$ and $r(i,k)$ can be taken as log-probability ratios [24]. The iterative alternating calculation of $a(i,k)$ and $r(i,k)$ is caused by the max-sum-algorithm applied for factor graphs [54], which can further be related to spectral clustering [45].

The number of resulting clusters is implicitly determined by the self-similarities $\varsigma(k,k)$ also denoted as preferences. The larger the self-similarities the finer is the granularity of clustering [24]. Common choices are the median or the minimum of the similarities between all inputs. Otherwise, the self-similarities can be seen as a control parameter for the granularity of the clustering. Variation of this parameter provides information regarding stable cluster solutions in dependence of plateau regions of the resulting minimum cost function value.

2.3.3 The generalized learning vector quantizer: an interpretable prototype-based classifier

Learning Vector Quantization (LVQ) is an adaptive prototype-based classifier introduced by T. Kohonen [34]. A cost-function-based variant is known as *generalized LVQ* [62]. This cost function approximates the classification error [32]. In particular, an LVQ classifier requires training data $T = \left\{\left(\mathbf{x}_j, c\left(\mathbf{x}_j\right)\right) \in X \times \mathscr{C},\ j = 1 \ldots N\right\}$ where $X \subseteq \mathbb{R}^n$ and $\mathscr{C} = \{1, \ldots, C\}$ is the set of available class labels. Further, the model assumes a set of prototypes $W = \{\mathbf{w}_k \in \mathbb{R}^n,\ k = 1 \ldots M\}$ with class labels $c(\mathbf{w}_k)$ such that at least one prototype is assigned to each class. Hence, we have a partitioning of the prototype set $W = \cup_{j=1}^{C} W_j$ with $W_j = \{\mathbf{w}_k \in W | c(\mathbf{w}_k) = j\}$. Further, a dissimilarity measure $d(\mathbf{x}, \mathbf{w})$ is supposed, which has to be differentiable with respect to the second argument. For a given LVQ configuration, a new data point $\mathbf{x}$ is assigned to a class by the mapping

$$\mathbf{x} \mapsto c\left(\mathbf{w}_{\omega(W)}\right) \tag{5}$$

with

$$\omega(W) = \mathsf{argmin}_{\mathbf{w}_k \in W} d(\mathbf{x}, \mathbf{w}_k) \tag{6}$$

is known as the winner-takes-all rule (WTA) in prototype-based vector quantization. The prototype $\mathbf{w}_\omega$ is denoted as winner of the competition.

During the learning, the cost-based LVQ minimizes the expected classification error $E_X[E(\mathbf{x}_k, W)]$ where

$$E(\mathbf{x}_k, W) = f(\mu(\mathbf{x}_k)) \tag{7}$$

is the local classification error depending on the choice of the monotonically increasing function f and the classifier function

$$\mu(\mathbf{x}_k) = \frac{d^+(\mathbf{x}_k) - d^-(\mathbf{x}_k)}{d^+(\mathbf{x}_k) + d^-(\mathbf{x}_k)} \in [-1, 1] \tag{8}$$

where $d^\pm(\mathbf{x}_k) = d^\pm(\mathbf{x}_k, \mathbf{w}^\pm)$ and $\mathbf{w}^+ = \mathbf{w}_{\omega\left(W_{c(\mathbf{x}_k)}\right)}$ is the so-called best matching correct prototype and $\mathbf{w}^- = \mathbf{w}_{\omega\left(W \setminus W_{c(\mathbf{x}_k)}\right)}$ is the corresponding best matching incorrect prototype. Frequently, the squashing function f is chosen as sigmoid: $f_\sigma(z) = \frac{1}{1+\exp(-\sigma z)}$. Learning takes place as stochastic gradient descent learning (SGDL) [27, 57] of $E_X[E(\mathbf{x}_k, W)]$ with respect to the prototype set W to obtain an optimum prototype configuration in the data space.

The dissimilarity $d(\mathbf{x}, \mathbf{w})$ can be chosen arbitrarily supposing differentiability with respect to $\mathbf{w}$ to ensure SGDL. Frequently, the squared Euclidean distance $d_E(\mathbf{x}, \mathbf{w}) = (\mathbf{x} - \mathbf{w})^2$ is applied resulting in the *standard generalized LVQ* (GLVQ). If both $\mathbf{x}$ and $\mathbf{w}$ are assumed as discrete representations of density functions, divergences like the Kullback–Leibler divergence $D_{KL}(\mathbf{x}, \mathbf{w})$ from (3) come into play instead [50]. It should be emphasized here that the non-symmetry of general divergences is not affecting the algorithm if it is used consistently in the predefined manner. Taking the variant $D_{KL}(\mathbf{x}, \mathbf{w})$ leads to the computational advantage that the Shannon entropy $H(\mathbf{x})$ of the data according to (3) is not required to be calculated because the derivative with respect to a prototype $\mathbf{w}$ vanishes and, hence, does not contribute to the learning. In consequence, only the derivative of the cross-entropy $Cr(\mathbf{x}, \mathbf{w})$ affects the learning as it is also known from classification learning by deep neural networks [25].

The resulting LVQ variant is denoted as *divergence-based GLVQ* (GDLVQ). We refer to [79] for further considerations and mathematical analysis.

Another popular choice is the squared Euclidean mapping distance

$$\begin{aligned} d_{\mathbf{\Omega}}(\mathbf{x}, \mathbf{w}) &= (\mathbf{\Omega}(\mathbf{x} - \mathbf{w}))^2 \\ &= (\mathbf{\Omega}(\mathbf{x} - \mathbf{w}))^T \mathbf{\Omega}(\mathbf{x} - \mathbf{w}) \\ &= (\mathbf{x} - \mathbf{w})^T \mathbf{\Omega}^T \mathbf{\Omega}(\mathbf{x} - \mathbf{w}) \end{aligned} \tag{9}$$

proposed in [66] with the mapping matrix $\mathbf{\Omega} \in \mathbb{R}^{m \times n}$ and m being the projection dimension usually chosen $m \leq n$ [12]. Here, the data are first mapped linearly by the mapping matrix and then the Euclidean distance is calculated in the mapping space $\mathbb{R}^m$. The mapping matrix can be optimized again by SGDL to achieve a good separation of the classes in the mapping space. The respective algorithm is known as *Generalized Matrix LVQ* (GMLVQ) [65]. Note that SGDL for $\mathbf{\Omega}$-optimization usually requires a careful regularization technique [64].

After training, the adapted projection matrix $\mathbf{\Omega}$ provides additional information. The resulting matrix $\mathbf{\Lambda} = \mathbf{\Omega}^T \mathbf{\Omega} \in \mathbb{R}^{n \times n}$ allows an interpretation as *classification correlation matrix*, i.e., the matrix entries Λ_{ij} give only those correlation information between data features i and j, which contribute to the class discrimination [5, 77]. Thus, it is not comparable with the data correlation matrix (or covariance), which does not reflect class discriminating correlations. Moreover, because the $\mathbf{\Omega}$-matrix, and therefore also the $\mathbf{\Lambda}$-matrix, is optimized to maximize the classifier accuracy, bias effects as known from covariance estimation as explained in [19] are not problematic in this context.

Instead of the linear $\mathbf{\Omega}$ mapping, nonlinear mappings could be considered explicitly as suggested in [81] or implicitly by means of kernel distances [63, 80].

A trained LVQ model can be applied to newly incoming data of unknown distribution. However, care must be taken to ensure that the model remains applicable and that there is no inconsistency with the new data. Therefore, each LVQ can be equipped with a reject option for the application phase [22, 29]. If the dissimilarity of the best matching prototype to a data point is greater than a given threshold τ, it is refused for classification, i.e., this optional tool equips the LVQ with a so-called *self-controlled evidence* (SCE) [82]. The threshold τ is determined during model training for each prototype individually, e.g., 95% percentile of the dissimilarity value for those data, which are assigned to the considered prototype by the WTA rule (6) together with the class assignment (5).

In fact, this reject option improves the robustness of the model [61].

2.4 Stochastic neighbor embedding for visualization

The method of stochastic neighbor embedding (SNE) was developed to visualize high-dimensional data in a typically two-dimensional visualization space [30]. For this purpose, each data point $\mathbf{x}_k$ in the data space is associated with a visualization vector $\mathbf{v}_k \in \mathbb{R}^2$. The objective of the respective embedding algorithm is to distribute the visualization data in a way that the density of original data distances in the high-dimensional data space is preserved as good as possible for the respective density of the distances in the visualization space (embedding space). The quality criterion is the Kullback–Leibler divergence between them, which is minimized by SGDL with respect to the visualization vectors $\mathbf{v}_k$.

Yet, SNE suffers from the fact that the distance densities in the original data space are frequently heavy-tailed [11], which leads to inaccurate visualizations. To overcome this problem, the so-called *t-distributed* SNE (*t*-SNE) was developed [74].

2.5 Data processing workflow

In the following, we describe and motivate the steps of data processing and analysis.

1. Coding of all sequences of D_G data and D_N data.
   - Alphabet $\mathscr{A}' = \mathscr{A} \cup \mathscr{E}$ with alphabet extension $\mathscr{E} = \{B, D, H, K, M, N, R, S, V, W, Y\}$ due to ambiguous characters in the datasets.
   - A natural vector representation $\mathbf{x}(4, \mathbf{s}) \in \mathbb{R}^{20}$ of order $K = 4$ is generated for each sequence $\mathbf{s}$ according to (1) paying attention to the alphabet extension $\mathscr{E}$.
   - A BoW-representation for 3-mers is generated for each sequence $\mathbf{s}$: $\mathbf{h}(\mathbf{s}) \in \mathbb{R}^{64}$ according to the possible nucleotide triplets of the alphabet $\mathscr{A} = \{A, C, G, T\}$ paying attention to the alphabet extension $\mathscr{E}$
2. Training of LVQ-classifiers for D_G data to evaluate the results from [23] obtained by phylogenetic trees
   - Training data are all samples of D_G with the additional virus type assignment *A*, *B*, or *C* taken as class labels.
   - For all LVQ variants, we take only one prototype per class.
   - For GMLVQ, the projection matrix is chosen as $\mathbf{\Omega} \in \mathbb{R}^{2\times n}$, i.e., the mapping dimension is $m = 2$.
   - SGDL training as tenfold cross-validation to determine the best LVQ architecture for the given problem.
     - Training of W using the GLVQ for NV representation.
     - Training of W and $\mathbf{\Omega}$ using the GMLVQ for NV representation.

       GDLVQ is not applicable for this sequence representation due to mathematical reasons.
     - Training of W using the GLVQ for BoW representation.
     - Training of W and $\mathbf{\Omega}$ using the GMLVQ for BoW representation.
   - Final training of the best LVQ architecture with optimum training schedule to achieve best prototype configuration W.
     - If GMLVQ architecture is selected for final training: training of both W and $\mathbf{\Omega}$, determination of the classification correlation matrix $\mathbf{\Lambda} = \mathbf{\Omega}^T\mathbf{\Omega}$.
     - Determination of the reject thresholds for each prototype for self-controlled evidence use based on the 95% percentile rule.
3. Clustering D_N data
   - Compression of the subset of 706 US sequences of March by MNG to achieve 50 representatives by MNG using 50 prototypes.
   - Generating a balanced subset consisting of all China samples (63), all Europe samples (33), and USA samples (114) for cluster analysis. The US samples comprise the 50 representatives from MNG and all US samples from January and February. The samples from other regions are not considered for cluster analysis. We denote this balanced dataset extracted from D_N by D_{NB}.
   - Clustering and identification of stable cluster solutions using affinity propagation by means of the control parameter $\varsigma = \varsigma(k, k)\ \forall k$.
4. Classification of the D_{NB} data as well as the full D_N data using the best LVQ classifier with integrated self-controlled evidence
   - Classification of the D_{NB} data by the final LVQ classifier with reject option using the determined thresholds to realize the self-controlled evidence (SCE).
   - Evaluation of the data rejected by the SCE rule.

3 Results

According to the processing workflow, we trained several LVQ classifier variants for the D_G data. By tenfold cross-validation, we achieved the averaged accuracies depicted in Table 2 together with their respective standard deviations. According to these results, GMLVQ performs best using the BoW coding of the sequences together with the Euclidean mapping distance $d_{\mathbf{\Omega}}(\mathbf{x}, \mathbf{w})$ from (9). Thus, we finally trained a GMLVQ network for both the prototype set W containing one prototype per class and the mapping matrix $\mathbf{\Omega}$ using the sequence BoW coding. For this final network, a classification accuracy of 100% is obtained while rejecting seven samples for classification according to the SCE decision. The resulting classification correlation matrix $\mathbf{\Lambda} = \mathbf{\Omega}^T\mathbf{\Omega}$ is depicted in S1 Fig. Because $\mathbf{\Omega} \in \mathbb{R}^{2\times n}$, it can serve for a data mapping into a two-dimensional visualization space. Accordingly, all D_G data together with the GMLVQ prototypes are visualized in S2 Fig. An additional visualization of the learned prototypes is given in S3 Fig.

The list of rejected sequences is provided in supplementary material S14 GMLVQ Mapping for D_N.

The clustering of the D_{NB} dataset suggests cluster solutions with either 2, 4, or 5 clusters according to the stability range of the control parameter ς, as shown in S4 Fig. We visualized the four-cluster solution using the t-SNE as depicted in S5 Fig. The respective cluster centroids are visualized in S6 Fig.

Applying the trained GMLVQ classifier to the D_{NB} dataset leads to the classification of 37 data points to class A, 95 data points to class B, and 2 data points to class C. According to the SCE decision, 59 data points were rejected from classification by the learned GMLVQ classifier. The result is given in S7 Fig using the t-SNE as visualization scheme. The visualization of the classification result by means of the $\mathbf{\Omega}$ mapping from the GMLVQ model delivers S8 Fig.

The distribution of the sequence data from the D_{NB} dataset with respect to the geographic sequence origins (regions) and the respective collection dates together with the class assignments is presented in S9 Fig. A respective visualization of the distribution for the dataset D_G is shown in S10 Fig.

The classification of the full D_N dataset assigns 154 data points to class A, 293 data points to class B, and 20 data points to class C, whereas 495 data points are rejected according to the SCE rule. The class assignments are visualized in S11 Fig.

The predicted virus type or the rejection decision for each sequence from D_N according to the GMLVQ class assignment or the SCE decision is found in supplementary material S14 GMLVQ Mapping for D_N.

4 Discussion

The classification analysis of the D_G data by means of the machine learning model GMLVQ verifies the class determination suggested in [23]. Only seven data samples are not classified accordingly due to the model self-controlled evidence decision. Thereby, the GMLVQ model shows a stable performance in learning (Table 2), which underlines its well-known robustness [60]. Thus, we observe an overall precise agreement supporting the findings in [23].

This agreement, however, is obtained by alignment-free sequence comparisons. More precisely, the nucleotide-based BoW sequence coding delivers a perfect separation of the given classes for the learned mapping distance $d_{\mathbf{\Omega}}(\mathbf{x}, \mathbf{w})$.

Yet, the computational complexity of a single dissimilarity calculation for the encoded sequences is only $O(64 \cdot m \cdot N_W)$ with $m = 2$ being the mapping dimension of $\mathbf{\Omega}$ and $N_W = |W|$ is the number of all prototypes in GLVQ/GMLVQ. The overall BoW sequence coding takes $O(l \cdot N)$. Paying attention to the fact that the GLVQ/GMLVQ training time scales with the number N of data, we have an overall complexity of $O(64 \cdot m \cdot N_W \cdot N)$ for model learning based on the coded data. Together with the time complexity $O(l \cdot N)$ for BoW-coding of all data with the sequence length l, we finally obtain an overall complexity of $O(N \cdot (64 \cdot m \cdot N_W + l))$ which usually is much lower than $O(N^4 + N \cdot l^2)$ for alignment-based methods [18], because $N_W \ll N$ and $n \ll l$ is valid.

Table 2 Classification results of trained LVQ variants for the D_G dataset obtained by tenfold cross-validation

	NV		BoW		
	GLVQ	GMLVQ	GLVQ	GDLVQ	GMLVQ
Averaged accuracy	53.1%	56.4%	81.7%	87.7%	97.4%
Standard deviation	±9.8%	±6.3%	±4.4%	±6.2%	±1.5%

Further, because GMLVQ is an interpretable classifier, we can draw further conclusions from the trained model: The resulted classification correlation matrix $\mathbf{\Lambda}$ depicted in S1 Fig suggests that particularly the histogram dimensions 27 and 28 are important in correlation with the other dimensions. These dimensions refer to the frequency of the triplets "CGG" and "CGT" in the sequences. Moreover, both dimensions should be negatively correlated for good class separation. This discrimination is a key feature of GMLVQ. Although the prototypes look very similar, as shown in S3 Fig, the $\mathbf{\Omega}$ is sensitive to smallest deviations in the histograms. Yet, we cannot expect greater deviations, because the sequences differ only in few characters according to the special mutations [23, 28]. The AP centroids differ slightly more than the GMLVQ prototypes, as shown in S6 Fig. This can be dedicated to larger overall scattering of the D_{NB} data.

Further, the GMLVQ prototypes serve as class "detectors." If the encoded sequences are most similar to them with respect to the mapping distance, the sequences are assigned to the respective classes according to the WTA rule (6). However, in general the prototypes are not identical with the mean vectors of the class distribution, as emphasized in [52].

Application of the GMLVQ to the D_N and D_{NB} data from the NCBI offers new insights. First, coloring of the data in the *t*-SNE visualization S7 Fig of D_{NB} according to the obtained class assignments seems to be confusing: The classes cannot be detected as separate regions in that case. However, applying the $\mathbf{\Omega}$ mapping S8 Fig, the class structure becomes visible also for this dataset. The reason for this discrepancy could be that both *t*-SNE and AP implicitly reflect data densities in the data space. Class densities, however, do not have to coincide with the overall data density. Thus, the $\mathbf{\Omega}$ mapping, which is optimized during GMLVQ training for best classification performance, offers the better visualization option and, hence, disclosures the class distribution more appropriately.

Comparing the class distributions of the sequences with respect to origins (regions) and collection dates for D_{NB} in S9 Fig and D_G in S10 Fig, both class distributions within the cells show a similar behavior. The D_{NB} dataset from NCBI contains only a few samples from Europe, all occurring from February onward, i.e., no European data samples from December/January were available. We observe that class C for the D_G data is mainly represented in January for European samples, which confirms the findings in [23]. Thus, the small number of class C samples in the D_{NB} classification may be addressed to this peculiarity in Europe. Further, the GMLVQ, which was trained by D_G data, rejects a large amount of data from D_{NB}, particularly in March. We suspect an accumulation of mutations which could explain the scattering. Accordingly, the GMLVQ is able to detect this behavior by means of the SCE decision rule.

We observe from the visualization S11 Fig of the classification for the D_N data that the data points rejected for classification scatter around the dense class regions. Thus, we can conclude that the nucleotide base mutations in the viral sequences, which cause the scattering, do not show a new coherent profile, at least at this time.

5 Conclusion

In this contribution, we investigate the application of interpretable machine learning methods to identify types of SARS-CoV-2 virus sequences based on alignment-free methods for RNA sequence comparison. In particular, we trained a *generalized matrix learning vector quantizer* classifier model (GMLVQ) for a dataset with given virus type information, which was obtained by phylogenetic tree analysis [23]. GMLVQ supposes vectorial data representations and compares vectors in terms of a well-defined dissimilarity measure. In this application, the GMLVQ training is based on the Bag-of-Words coded sequences and yields class specific prototype vectors as well as an optimum class/typus separating dissimilarity measure in the data space of encoded sequences. Compared to phylogenetic trees or multiple sequence alignment, which require high computational costs due to the involved sequence alignment process, the GMLVQ approach has lower complexity and allows an easy out-of-training generalization.

By means of the trained GMLVQ, we first verified the SARS-CoV-2 virus types determined in this first dataset. Further, considering a classification correlation matrix delivered by GMLVQ optimization, we are able to identify features which contribute decisively to a type separation.

Second, we applied the trained GMLVQ to another dataset obtained from the NCBI database without virus type information. Using the self-controlled evidence property of the GMLVQ, we are able to classify these sequences to the previously identified types, avoiding the application of the model to inconsistent data compared to the training data. Further, the rejected data allow speculations about new virus types with respect to nucleotide base mutations in the viral sequences.

Yet, an appropriate training and data coding for successful GMLVQ application require a careful and precise data handling as well as model training regime, i.e., respective expert knowledge.

Future work will consider the replacement of the WTA rule (6) by a fuzzy variant (winner ranking) resulting in a probabilistic class/type assignment instead of the crisp rule (5). This probabilistic view could be further integrated into the SCE-based rejection decision to differentiate between

rejected sequences regarding their consistence to the GMLVQ version in use. Thus, the user can decide whether to retrain the model adding a new class or continue with the current configuration.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00521-021-06018-2.

Acknowledgements M.K., K.S. B., M.W., and M.K. acknowledge support by a Grant of the European Social Fund (ESF).

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Andersen KG, Rambaut A, Lipkin WI, Holmes EC, Garry RF (2020) The proximal origin of SARS-CoV-2. Nat Med 26:450–452
2. Bai Y, Jiang D, Lon J, Chen X, Hu M, Lin S, Chen Z, Meng Y, Du H (2020) Evolution and molecular characteristics of SARS-CoV-2 genome. *bioRXiv*, (2020.04.24.058933)
3. Bauer H-U, Herrmann M, Villmann T (1999) Neural maps and topographic vector quantization. Neural Netw 12(4–5):659–676
4. Bhanot G, Biehl M, Vilmann T, Zühlke D (2017) Biomedical data analysis in translational research: Integration of expert knowledge and interpretable models. In M. Verleysen, editor, *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'2017)*, pages 177–186, Louvain-La-Neuve, Belgium. i6doc.com
5. Biehl M, Hammer B, Villmann T (2016) Prototype-based models in machine learning. Wiley Interdisciplinary Rev Cogn Sci 2:92–111
6. Bittrich S, Kaden M, Leberecht C, Kaiser F, Villmann T, Labudde D (2019) Application of an interpretable classification model on early folding residues during protein folding. BioData Min 12(1):1–16
7. Blaisdell B (1986) A measure of the similarity of sets of sequences not requiring sequence alignment. Proc Natl Acad Sci USA 83:5155–5159
8. Blaisdell B (1989) Average values of a dissimilarity measure not requiring sequence alignment are twice the averages of conventional mismatch counts requiring sequence alignment for a computer-generated model system. J Mol Evol 29:538–547
9. Bosco G, diGangi M (2016) Deep learning architectures for DNA sequence classification. In A. Petrosino, V. Loia, and W. Pedrycz, editors, *Fuzzy Logic and Soft Computing Applications: Proceedings of the International Workshop on Fuzzy Logic and Applications (WILF 2016)*, volume 10147 of *LNCS*, pages 162–171, Cham. Springer
10. Brister JR, Ako-adjei D, Bao Y, Blinkova O (2014) NCBI viral genomes resource. Nucleic Acids Res 43(D1):D571–D577
11. Bryson M (1974) Heavy-tailed distributions: properties and tests. Technometrics 16(1):61–68
12. Bunte K, Schneider P, Hammer B, Schleif F-M, Villmann T, Biehl M (2012) Limited rank matrix learning, discriminative dimension reduction and visualization. Neural Netw 26(1):159–173
13. Cilibrasi R, Vitányi P (2005) Clustering by compression. IEEE Trans Inf Theory 51(4):1523–1545
14. Clark K, Karsch-Mizrachi I, Lipman D, Ostell J, Sayers EW (2015) GenBank. Nucleic Acids Res 44(D1):D67–D72
15. Cornish-Bowden A (1985) Nomenclature for incompletely specified bases in nucleic acid sequences: recommendations 1984. Nucleic Acids Res 13(9):3021–3030
16. Cottrell M, Hammer B, Hasenfuß A, Villmann T (2006) Batch and median neural gas. Neural Netw 19:762–771
17. Deng M, Yu C, Liang Q, He R, Yau S-T (2011) A novel method of characterizing sequences: genome space with biological distance and applications. PLoS One 6(3):e17293
18. Edgar RC (2004) MUSCLE: a multiple sequence alignment method with reduced time and space complexity. BMC Bioinform 5(1):113
19. Fan J, Liao Y, Liu H (2016) An overview of the estimation of large covariance and precision matrices. Econom J 19:C1–C32
20. Feng D-F, Doolittle RF (1987) Progressive sequence alignment as a prerequisite to correct phylogenetic trees. J Mol Evol 25(4):351–360
21. Fianacca A, LaPaglia L, LaRosa M, LoBosco G, Renda G, Rizzo R, Galio S, Urso A (2018) Deep learning models for bacteria taxonomic classification of metagenomic data. BMC Bioinform 19(Suppl. 7):198
22. Fischer L, Hammer B, Wersing H (2015) Efficient rejection strategies for prototype-based classification. Neurocomputing 169:334–342
23. Foster P, Foster L, Renfrew C, Forster M (2020) Phylogenetic network analysis of SARS-CoV-2 genomes. *Proc. of the National Academy of Science of the United States of America (PNAS)*
24. Frey B, Dueck D (2007) Clustering by message passing between data points. Science 315:972–976
25. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge
26. Gotoh O (1982) An improved algorithm for matching biological sequences. J Mol Biol 162:705–708
27. Graf S, Lushgy H (2000) Foundations of quantization for probability distributions, vol 1730. Lect. Notes in Mathematics, Springer, Berlin
28. Guan Q, Sadykov M, Nugmanova R, Carr M, Arold S, Pain A (2020) The genomic variation landscape of globally-circulating clades of SARS-CoV-2 defines a genetic barcoding scheme. *bioRXiv*, (2020.04.21.054221)
29. Herbei R, Wegkamp M (2006) Classification with reject option. Can J Stat 34(4):709–721
30. Hinton G, Roweis S (2002) Stochastic neighbor embedding. Advances in Neural Information Processing Systems. volume 15. The MIT Press, Cambridge, MA, USA, pp 833–840

31. Just W (2001) Computational complexity of multiple sequence alignment with SP-score. J Comput Biol 8(6):615–623
32. Kaden M, Lange M, Nebel D, Riedel M, Geweniger T, Villmann T (2014) Aspects in classification learning—review of recent developments in learning vector quantization. Found Comput Decis Sci 39(2):79–105
33. Katoh K, Standley DM (2013) MAFFT multiple sequence alignment software version 7: improvements in performance and usability. Mol Biol Evol 30(4):772–780
34. Kohonen T (1988) Learning vector quantization. Neural Netw 1(Supplement 1):303
35. Kolmogorov A (1965) Three approaches to the quantitative definition of information. Probl Inf Transm 1(1):1–7
36. Kullback S, Leibler R (1951) On information and sufficiency. Annal Math Stat 22:79–86
37. Lan J, Ge J, Yu J, Shan S, Fan HZS, Zhang Q, Shi X, Wang Q, Zhang L, Wang X (2020) Structure of the SARS-CoV-2 spike receptorbinding domain bound to the ACE2 receptor. Nature. https://doi.org/10.1038/s41586-020-2180-5)
38. Levenshtein V (1965) Binary codes capable of correcting deletions, insertions, and reversals. Doklady Akademii Nauk SSSR 163(4):845–848
39. Li J, Song S, Zhang Y, Zhou Z (2016) Robust k-median and k-means clustering algorithms for incomplete data. *Mathematical Problems in Engineering*, 2016(Article ID 4321928):1–8
40. Li M, Chen X, Li X, Ma B, Vitányi P (2004) The similarity metric. IEEE Trans Inf Theory 50(12):3250–3264
41. Li Y, Liu B, Cui J, Wang Z, Shen Y, Xu Y, Yao K, Guan Y (2020) Similarities and evolutionary relationships of COVID-19 and related viruses. *arXiv*, (2003.05580)
42. Li Y, Tian K, Yin C, He R, Yau S-T (2016) Virus classification in 60-dimensional protein space. Mol Phylogenetics Evol 99:53–62
43. Lin J, Adjeroh D, Jiang B-H, Jiang Y (2018) k_2 and k_2^*?: efficient alignment-free sequence similarity measurement based on Kendall statistics. Bioinformatics 34(10):1682–1689
44. ...Lu R, Zhao X, Juan L, Niu P, Yang B, Wu H, Wang W, Song H, Huang B, Zhu N, Bi Y, Ma X, Zhan F, Wang L, Hu T, Zhou H, Hu Z, Zhou W, Zhao L, Chen J, Meng Y, Wang J, Yuan YL, Xie Z, Ma J, Liu W, Wang D, Xu W, Holmes E, Gao G, Wu G, Chen W, Shi W, Tan W (2020) Genomic characterisation and epidemiology of 2019 novel coronavirus: implications for virus origins and receptor binding. Lancet 395(10224):565–574
45. Luxburg UV (2007) A tutorial on spectral clustering. Stat Comput 17(4):395–416
46. Mackay D (2003) Information Theory. Cambridge University Press, Inference and Learning Algorithms
47. Maiolo M, Zhang X, Gil M, Anisimova M (2018) Progressive multiple sequence alignment with indel evolution. BMC Bioinform 19(1):331
48. Martinetz TM, Berkovich SG, Schulten KJ (1993) Neural-gas network for vector quantization and its application to time-series prediction. IEEE Trans Neural Netw 4(4):558–569
49. Miyamoto S, Ichihashi H, Honda K (2008) *Algorithms for Fuzzy Clustering*, volume 229 of *Studies in Fuzziness and Soft Computing*. Springer
50. Mwebaze E, Schneider P, Schleif F-M, Aduwo J, Quinn J, Haase S, Villmann T, Biehl M (2011) Divergence based classification in learning vector quantization. Neurocomputing 74(9):1429–1435
51. Needleman S, Wunsch C (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol 48(3):443–453
52. Oehler KL, Gray RM (1995) Combining image compression and classification using vector quantization. IEEE Trans Pattern Anal Mach Intell 17:461–473
53. Paden C, Tao Y, Queen K, Zhang J, Li Y, Uehara A, Tong S (2020) Rapid, sensitive, full genome sequencing of severe acute respiratory syndrome virus coronavirus 2 (SARS-CoV-2). *bioRXiv*, (2020.04.22.055897)
54. Pearl J (1988) Probabilistic reasoning in intelligent system. Morgan Kaufmann, Burlington
55. Quinlan J (1986) Induction of decision trees. Mach Learn 1:81–106
56. Quinlan J (1993) *C4.5: Programs for Machine Learning*. Morgan Kaufmann
57. Robbins H, Monro S (1951) A stochastic approximation method. Ann Math Stat 22:400–407
58. Rudin C (2019) Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nat Mach Intell 1(5):206–215
59. Samek W, Monatvon G, Vedaldi A, Hansen L, Müller K-R (eds) (2019) Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, number 11700 in LNAI. Springer
60. Saralajew S, Holdijk L, Rees M, Villmann T (2019) Robustness of generalized learning vector quantization models against adversarial attacks. In: Vellido A, Gibert K, Angulo C, Guerrero J (Eds) Advances in Self-Organizing Maps. Learning Vector Quantization, Clustering and Data Visualization. In: Proceedings of the 13th International Workshop on Self-Organizing Maps and Learning Vector Quantization, Clustering and Data Visualization, WSOM+2019, Barcelona, volume 976 of Advances in Intelligent Systems and Computing. Springer, Berlin-Heidelberg, pp 189–199
61. Saralajew S, Holdijk L, Villmann T (2020) Fast adversarial robustness certification of nearest prototype classifiers for arbitrary seminorms. In: *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, page in press. MIT Press
62. Sato A, Yamada K (1996) Generalized learning vector quantization. In: Touretzky DS, Mozer MC, Hasselmo ME (Eds) Advances in Neural Information Processing Systems 8. In: Proceedings of the (1995) Conference. MIT Press, Cambridge, MA, USA, pp 423–9
63. Schleif F-M, Villmann T, Hammer B, Schneider P (2011) Efficient kernelized prototype based classification. Int J Neural Syst 21(6):443–457
64. Schneider P, Bunte K, Stiekema H, Hammer B, Villmann T, Biehl M (2010) Regularization in matrix relevance learning. IEEE Trans Neural Netw 21(5):831–840
65. Schneider P, Hammer B, Biehl M (2009) Adaptive relevance matrices in learning vector quantization. Neural Comput 21:3532–3561
66. Schneider P, Hammer B, Biehl M (2009) Distance learning in discriminative vector quantization. Neural Comput 21:2942–2969
67. Sievers A, Bosiek K, Bisch M, Dreessen C, Riedel J, Froß P, Hausmann M, Hildenbrand G (2017) k-mer content, correlation, and position analysis of genome DNA sequences for the identification of function and evolutionary features. Genes 8(122):1–18
68. Smith T, Watermann M (1981) Identification of common molecular subsequences. J Mol Biol 147(1):195–197
69. Sun Y, Zhu S, Ma K, Liu W, Yue Y, Hu G, Lu H, Chen W (2019) Identification of 12 cancer types through genome deep learning. Nat Sci Rep 9(1):1–9
70. Szostak N, Synak J, Borowski M, Wasik S, Blazewicz J (2017) Simulating the origins of life: the dual role of RNA replicases as an obstacle to evolution. PLoS ONE 12(7):1–28
71. Szostak N, Wasik S, Blazewicz J (2016) Hypercycle. PLOS Comput Biol 12(4):e1004853
72. Tampuu A, Bzhalava Z, Dillner J, Vicente R (2019) ViraMiner: deep learning on raw DNA sequences for identifying viral genomes in human samples. Plose One 14(9):e0222271
73. Thompson JD, Higgins DG, Gibson TJ (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence

alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acids Res 22(22):4673–4680
74. van der Maaten L, Hinton G (2008) Visualizing high-dimensional data using *t*-SNE. J Mach Learn Res 9:2579–2605
75. Vasilarou M, Alachiotis N, Garefalaki J, Beloukas A, Pavlidis P (2020) Population genomics insights into the recent evolution of SARS-CoV-2. *bioRXiv*, (2020.04.21.054122)
76. Vellido A (2019) The importance of interpretability and visualization in machine learning for applications in medicine and health care. Neural Netw Appl. https://doi.org/10.1007/s00521-019-04051-w
77. Villmann T, Bohnsack A, Kaden M (2017) Can learning vector quantization be an alternative to SVM and deep learning? J Artif Intell Soft Comput Res 7(1):65–81
78. Villmann T, Claussen J-C (2006) Magnification control in self-organizing maps and neural gas. Neural Comput 18(2):446–469
79. Villmann T, Haase S (2011) Divergence based vector quantization. Neural Comput 23(5):1343–1392
80. Villmann T, Haase S, Kaden M (2015) Kernelized vector quantization in gradient-descent learning. Neurocomputing 147:83–95
81. Villmann T, Ravichandran J, Engelsberger A, Villmann A, Kaden M (2020) Quantum-inspired learning vector quantizers for prototype-based classification. Neural Comput Appl. https://doi.org/10.1007/s00521-020-05517-y
82. Villmann T, Saralajew S, Villmann A, Kaden M (2018) Learning vector quantization methods for interpretable classification learning and multilayer networks. In C. Sabourin, J. Merelo, A. Barranco, K. Madani, and K. Warwick (Eds). Proceedings of the 10th International Joint Conference on Computational Intelligence (IJCCI), Sevilla
83. Vinga S (2004) Information theory applications for biological sequence analysis. Bioinformatics 15(3):376–389
84. Vinga S, Almeida J (2004) Alignment-free sequence comparison—a review. Bioinformatics 20(2):206–215
85. Vinga S, Almeida J (2004) Rényi continuous entropy of DNA sequences. J Theor Biol 231:377–388
86. Warrow T (2017) Computational phylogenetics: an introduction to designing methods for phylogeny estimation. Cambridge University Press, Cambridge
87. Wasik S, Szostak N, Kudla M, Wachowiak M, Krawiec K, Blazewicz J (2019) Detecting life signatures with RNA sequence similarity measures. J Theor Biol 463:110–120
88. Wu J, Leung K, Leung G (2020) Nowcasting and forecasting the potential domestic and international spread of the 2019-nCoV outbreak originating in Wuhan, China: a modelling study. Lancet 395:689–697
89. Yang H-C, Chen C-H, Wang J-H, Liao H-C, Yang C-T, Chen C-W, Lin Y-C, Kao C-H, Liao J (2020) Genomic, geographic and temporal distributions of SARS-CoV-2 mutations. *bioRXiv*, (2020.04.22.055863)
90. Yin C (2020) Genotyping coronavirus SARS-CoV-2: methods and implications. *arXiv*, (2003.10965v1)
91. Yin C, Chen Y, Yau S-T (2014) A measure of DNA sequence similarity by Fourier Transform with applications on hierarchical clustering. J Theor Biol 359:18–28
92. Yu C, Hernandez T, Zheng H, Yau S-C, Huang H-H, He R, Yang J, Yau S-T (2013) Real time classification of viruses in 12 dimensions. Plos One 8(5):e64328
93. Zeng J, Ustun B, Rudin C (2017) Interpretable classification models for recidivism prediction. J R Stat Soc Series A 180:1–34
94. Zielezinski A, Girgis H, Bernard G, Leimeister C-A, Tang K, Dencker T, Lau A, Röhling S, Choi J, Waterman M, Comin M, Kim S-H, Vinga S, Almeida J, Chan C, James B, Sun F, Morgenstern B, Karlowski W (2019) Benchmarking of alignment-free sequence comparison methods. Genome Biol 20(144):1–18
95. Zielezinski A, Vinga S, Almeida J, Karlowski W (2017) Alignment-free sequence comparison: benefits, applications, and tools. Genome Biol 18(186):1–17