Analysis of SARS-CoV-2 RNA-Sequences by Interpretable Machine Learning Models

Marika Kaden¹ , Katrin Sophie Bohnsack¹ , Mirko Weber¹ , Mateusz Kudła^{1,2} , Kaja Gutowska^{2,3,4} , Jacek Blazewicz^{2,3,4} , Thomas Villmann^{1*} ,

1 University of Applied Sciences Mittweida, Saxon Institute for Computational Intelligence and Machine Learning, Technikumplatz 17, 09648 Mittweida, Germany

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

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

4 European Centre for Bioinformatics and Genomics, Piotrowo 2, 60-965 Poznan, Poland

These authors contributed equally to this work.

These authors also contributed equally to this work.

* thomas.villmann@hs-mittweida.de

Abstract

We present an approach to investigate SARS-CoV-2 virus sequences based on alignment-free methods for RNA sequence comparison. In particular, we verify a given clustering result for the GISAID data set, which was obtained analyzing the molecular differences in coronavirus populations by phylogenetic trees. For this purpose, we use alignment-free dissimilarity measures for sequences and combine them with learning vector quantization classifiers for virus type discriminant analysis and classification. Those vector quantizers belong to the class of interpretable machine learning methods, which, on the one hand side provide additional knowledge about the classification decisions like discriminant feature correlations, and on the other hand can be equipped with a reject option. This option gives the model the property of self controlled evidence if applied to new data, i.e. the models refuses to make a classification decision, if the model evidence for the presented data is not given. After training such a classifier for the GISAID data set, we apply the obtained classifier model to another but unlabeled SARS-CoV-2 virus data set. On the one hand side, this allows us to assign new sequences to already known virus types and, on the other hand, the rejected sequences allow speculations about new virus types with respect to nucleotide base mutations in the viral sequences.

Author summary

The currently emerging global disease COVID-19 caused by novel SARS-CoV-2 viruses requires all scientific effort to investigate the development of the viral epidemy, the

properties of the virus and its types. Investigations of the virus sequence are of special interest. Frequently, those are based on mathematical/statistical analysis. However, machine learning methods represent a promising alternative, if one focuses on interpretable models, i.e. those that do not act as *black-boxes*. Doing so, we apply variants of *Learning Vector Quantizers* to analyze the SARS-CoV-2 sequences. We encoded the sequences and compared them in their numerical representations to avoid the computationally costly comparison based on sequence alignments. Our resulting model is interpretable, robust, efficient, and has a self-controlling mechanism regarding the applicability to data. This framework was applied to two data sets concerning SARS-CoV-2. We were able to verify previously published virus type findings for one of the data sets by training our model to accurately identify the virus type of sequences. For sequences without virus type information (second data set), our trained model can predict them. Thereby, we observe a new scattered spreading of the sequences in the data space which probably is caused by mutations in the viral sequences.

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 [2]. Yet there are several indicators that the molecular characteristic evolves during time [3,4]. This evolution is mainly driven by mutations, which play an essential role and maybe accompanied by mechanisms of stabilization [5,6].

The analysis of the genomic structure by sequencing is currently topic of ongoing research to better understand the molecular dynamics [7]. 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 [8].

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 [9]. 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 [10]. Further, mutations are not equally distributed across the SARS-CoV-2 genome [11]. The molecular differences in coronavirus populations were investigated using phylogenetic trees so far resulting in three clusters which are identified as virus types [12]. Yet, SNP-based radial phylogeny-retrieved trees of SARS-CoV-2 genomes result in five major clades [11]. 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 [13,14]. The respective tree has to be reconfigured completely, which frequently leads to major changes in the tree structure [15,16].

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 [17]. However, a general mathematical evaluation of similarities is crucial because respective similarity measures only partially reflect all biological aspects of similarity between RNA sequences [18]. Alignment based methods usually rely on variants of the Levenshtein distance [19], which, however, are computationally costly: $O(n_1 \cdot n_2)$ is the time complexity for both the Needleman-Wunsch-algorithm [20] and for the Smith-Waterman-algorithm [21,22],

where n_1 and n_2 are the sequences length. Hence, if $n_1 = n_2 = n$ the complexity is simply $O(n^2)$. Both approaches solve internally a mathematical optimization problem, i.e. both algorithms belong to the algorithmic class of dynamic programming with high computational complexity. Other alignment based methods consider (multiple) longest common subsequences with similar complexity [23].

Therefore, alignment-free alternatives are promising [18, 24–28]. Most common those approaches are *Bag-of-Words* (BoW, [29]), information theoretic methods based on the *Kolmogorov-Smirnov-complexity* [30] and the related *Normalized Compression Distance* [31, 32]. Recently, similarities based on *Natural Vectors* gained attraction [33–35]. 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 are lost. An overview of prominent measures and their behavior for sequence analysis can be found in [36, 37].

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 [38, 39]. This we do for two data sets: GISAID-data and NCBI-data. For the first one, virus classes (types) were identified by phylogenetic tree analysis in [12], whereas the second one is without class information.

Although deep neural network approaches provide impressive results in sequence classification [16, 40–42], deep architectures are at least difficult to interpret [43]. 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 [44–46]. 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 [45, 47]. 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.

Materials and methods

SARS-CoV-2 Sequence Databases in Use

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

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 4th, 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 [12], 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 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 [12]. 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 the supplementary material.

The complete data information can be found in supplementary files S12 Data.

The NCBI dataset D_N

The second data set including 892 complete genomes has been selected from the National Center for Biotechnology Information (NCBI) Viral Genome database [48], and GenBank [49] by April 19th, 2020, see Tab. 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 have been included in the supplementary material. We refer to this data set 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.

	China	Europe	USA	Others
Dec '19	16	0	0	0
Jan '20	44	4	16	9
Feb '20	2	6	44	7
Mar '20	1	23	706	10
Apr '20	0	0	4	0

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

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

Representation of RNA Sequences for Alignment-free Data Analysis

Several approaches were published to represent sequences adequately for alignment-free comparison. These method range from chaos game representation to standard unary coding or matrix representations. An overview is given in [27] and [36,37]. Here we focus only on two of the most promising approaches – *Natural Vectors* and *Bag-of-Words*.

Natural Vectors

Natural Vectors (NV) for nucleotide sequence comparison is based a statistical sequence description for the distribution of nucleotide positions in a sequence $\mathbf{s}$ based on the alphabet $\mathcal{A} = \{A, C, G, T\}$ [33, 34]. Let $\mu_L^0 = n_L/n$ be the relative number (frequency) of the nucleotide $L \in \mathcal{A}$ and $p_L(j)/n$, $j = 1 \dots n_L$ is the *relative* position of the k th nucleotide L in the sequence. Let $E[r]$ further be the expectation operator of a random quantity r . Thus $\mu_L^0 = E[L]$. Further, we denote by $\mu_L = \mu_L^1 = E[p_L]$ the mean relative position of the nucleotide L in the sequence. The k -th centralized moment μ_L^k for $k \geq 2$ is given as $\mu_L^k = E[(p_L - \mu_L^1)^k]$. Then, the natural vector of order K for a sequence $\mathbf{s}$ is given as

$$\mathbf{x}(K, \mathbf{s}) = (\mu_A^0, \mu_C^0, \mu_G^0, \mu_T^0, \mu_A^1, \mu_C^1, \mu_G^1, \mu_T^1, \dots, \mu_A^K, \mu_C^K, \mu_G^K, \mu_T^K) \quad (1)$$

whereby we again drop the dependencies on K and $\mathbf{s}$, if it is not misleading. Natural vectors are usually compared in terms of the l_p -metric

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

giving the Euclidean distance for $p = 2$. Kendall-statistics, as a kind of correlation measure, was applied in [50].

The NV-description of sequences can also be applied for nucleotide sequences containing ambiguous characters (degenerate bases) [35, 51]. This yields an extended alphabet $\mathcal{A}' = \mathcal{A} \cup \mathcal{E}$. In that case, weights $0 \leq w_L(s_i) \leq 1$ are introduced for each $L \in \mathcal{A}$ with

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

where p_{L, s_i} is the probability that the detected ambiguous character $s_i \in \mathcal{E}$ should be the character L . These weights have to be taken into account during the expectation value calculations [35].

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 $\mathcal{A} = \{A, C, G, T\}$ [24, 25, 27, 41]. 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 is provided in the 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 latter mentioned density property, an alternative

choice is to compare by means of divergence measures [52]. In the investigations presented later, we applied the Kullback-Leibler-divergence [53]

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

for sequence histograms $\mathbf{h}$ and $\mathbf{m}$. Note that the first term in (3) is the negative Shannon entropy $H(\mathbf{h}) = -\sum_{j=1}^n h_j \cdot \log(h_j)$ whereas $Cr(\mathbf{h}, \mathbf{m}) = \sum_{j=1}^n h_j \cdot \log(m_j)$ is the Shannon *cross-entropy*. Yet, other divergences like Rényi-divergences could be used [54]. We refer to [55] for a general overview regarding divergences in the context of machine learning.

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

Machine Learning Methods for Virus Sequence Data Analysis

Median Neural Gas for Data Compression

The *Median Neural Gas* algorithm (MNG) is a neural data quantization algorithm for data compression based on dissimilarities [56, 57]. 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 [58, 59]. 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 [60, 61].

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.

Affinity Propagation for Clustering with Cluster Number Control

Affinity Propagation (AP) introduced by FREY&DUECK in [62] 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

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

where $I : N \rightarrow N$ is the mapping function determining the prototypes for each data point given in (4) and

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

is a penalty function. During the optimization, two kind of messages are iteratively exchanged between the data until convergence: the responsibilities $r(i, k)$ and the availabilities $a(i, k)$. The responsibilities

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

reflect the accumulated evidence that point k serves as prototype for data point i . The availabilities

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

describe the accumulated evidence how appropriate data point k is seen as a potential prototype for the points i . Finally, the prototypes are determined by

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

Hence, $a(i, k)$ and $r(i, k)$ can be taken as log-probability ratios [62]. The iterative alternating calculation of $a(i, k)$ and $r(i, k)$ is caused by the max-sum-algorithm applied for factor graphs [63], which can further be related to spectral clustering [64].

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 [62]. 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.

Interpretable Prototype-based Classifier – the Generalized Learning Vector Quantizer

Learning Vector Quantization (LVQ) was is an adaptive prototype-based classifier introduced by T. KOHONEN [65]. A cost-function-based variant is known as *generalized LVQ* [66]. This cost function approximates the classification error [67]. In particular, an LVQ classifier requires training data $T = \{(\mathbf{x}_j, c(\mathbf{x}_j)) \in X \times \mathcal{C}, j = 1 \dots N\}$ where $X \subseteq \mathbb{R}^n$ and $\mathcal{C} = \{1, \dots, C\}$ is the set of available class labels. Further, the model assumes a set of prototypes $W = \{\mathbf{w}_k \in \mathbb{R}^n, k = 1 \dots M\}$ 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

$$\mathbf{x} \mapsto c(\mathbf{w}_{\omega(W)}) \quad (5)$$

with

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

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

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

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

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

where $d^\pm(\mathbf{x}_k) = d^\pm(\mathbf{x}_k, \mathbf{w}^\pm)$ and $\mathbf{w}^+ = \mathbf{w}_{\omega(W_c(\mathbf{x}_k))}$ is the so-called best matching correct prototype and $\mathbf{w}^- = \mathbf{w}_{\omega(W \setminus W_c(\mathbf{x}_k))}$ 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) [68, 69] 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 [70].

The resulting LVQ variant is denoted as *divergence-based GLVQ* (GDLVQ).

Another popular choice is the squared Euclidean mapping distance

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

proposed in [71] with the mapping matrix $\Omega \in \mathbb{R}^{m \times n}$ and m being the projection dimension usually chosen $m \leq n$ [72]. 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) [73]. Note that SGDL for Ω -optimization usually requires a careful regularization technique [74].

After training, the adapted projection matrix Ω provides additional information. The resulting matrix $\Lambda = \Omega^T \Omega \in \mathbb{R}^{n \times n}$ allows an interpretation as *classification correlation matrix*, i.e. the matrix entries Ω_{ij} give the correlations between data features i and j , which contribute to the class discrimination [39, 75]. A non-linear mapping could be realized applying kernel distances instead [76, 77].

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 [78, 79]. If the dissimilarity of the best matching prototype to a data point is greater than a given threshold τ , it is rejected for classification, i.e. this optional tool equips the LVQ with a so-called *self-controlled*

evidence (SCE) [45]. 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).

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 [80]. 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 [81], which leads to inaccurate visualizations. To overcome this problem, the so-called *t-distributed* SNE (*t*-SNE) was developed [82].

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 $\mathcal{A}' = \mathcal{A} \cup \mathcal{E}$ with alphabet extension $\mathcal{E} = \{B, D, H, K, M, N, R, S, V, W, Y\}$ due to ambiguous characters in the data sets.
 - 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 $\mathcal{E}$.
 - 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 $\mathcal{A} = \{A, C, G, T\}$ paying attention to the alphabet extension $\mathcal{E}$
2. Training of LVQ-classifiers for D_G -data to evaluate the results from [12] obtained by phylogentic 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 10-fold cross validation to determine the best LVQ architecture for the given problem.
 - Training of W using the GLVQ for NV representation.
 - * GDLVQ is not applicable for this sequence representation due to mathematical reasons.

- Training of W using the GLVQ for BoW representation. 296
- Training of W and Ω using the GMLVQ for BoW representation. 297
- Final training of the best LVQ architecture with optimum training schedule 298
to achieve best prototype configuration W . 299
 - If GMLVQ architecture is selected for final training: training of both W 300
and Ω , determination of the classification correlation matrix $\Lambda = \Omega^T \Omega$. 301
 - Determination of the reject thresholds for each prototype for 302
self-controlled of evidence use based on the 95%-percentile rule. 303
- 3. Clustering D_N -data 304
 - Compression of the subset of 706 US-sequences of March by MNG to achieve 305
50 representatives by MNG using 50 prototypes. 306
 - generating a balanced subsets consisting of all China samples (63), all Europe 307
samples (33), and USA samples (114) for cluster analysis. The US samples 308
comprise the 50 representatives from MNG and all US samples from January 309
and February. The sample from other regions are not considered for cluster 310
analysis. We denote this balanced data set extracted from D_N by D_{NB} . 311
 - Clustering and identification of stable cluster solutions using affinity 312
propagation by means of the control parameter $\varsigma = \varsigma(k, k) \forall k$. 313
- 4. Classification of the D_{NB} -data as well as the full D_N -data using the best LVQ 314
classifier with integrated self-controlled evidence 315
 - Classification of the D_{NB} -data by the final LVQ classifier with reject option 316
using the determined thresholds to realize the self-controlled evidence (SCE). 317
 - Evaluation of the data rejected by the SCE rule. 318

Results 319

According to the processing workflow we trained several LVQ-classifier variants for the 320
 D_G -data. By 10-fold cross-validation, we achieved the averaged accuracies depicted in 321
Tab. 2 together with their respective standard deviations. According to these results, 322
GMLVQ performs best using the BoW-coding of the sequences together with the 323
Euclidean mapping distance $d_\Omega(\mathbf{x}, \mathbf{w})$ from (9). Thus, we finally trained a GMLVQ 324
network for both the prototype set W containing one prototype per class and the 325
mapping matrix Ω using the sequence BoW-coding. For this final network a 326
classification accuracy of 100% is obtained while rejecting 7 samples for classification 327
according to the SCE-decision. The resulting classification correlation matrix $\Lambda = \Omega^T \Omega$ 328
is depicted in S1 Fig. Because $\Omega \in \mathbb{R}^{2 \times n}$, it can serve for a data mapping into a 329
two-dimensional visualization space. Accordingly, all D_G -data together with the 330
GMLVQ-prototypes are visualized in S2 Fig. An additional visualization of the learned 331
prototypes is given in S3 Fig. 332

The list of rejected sequences is provided in the supplementary material S14 GMLVQ 333
Mapping for D_N . 334

The clustering of the D_{NB} -data set suggests cluster solutions with either 2, 4, or 5 335
clusters according to the stability range of the control parameter ς , see S4 Fig. We 336
visualized the 4-cluster solution using the t -SNE as depicted in S5 Fig. The respective 337
cluster centroids are visualized in S6 Fig. 338

	NV			BoW	
	GLVQ	GMLVQ	GLVQ	GDLVQ	GMLVQ
averaged accuracy	53.1%	56.4%	81.7%	87.7%	97.4%
standard deviation	$\pm 9.8\%$	$\pm 6.3\%$	$\pm 4.4\%$	$\pm 6.2\%$	$\pm 1.5\%$

Table 2. Classification results of trained LVQ-variants for the D_G -dataset obtained by 10-fold cross-validation.

Applying the trained GMLVQ classifier to the D_{NB} -data set leads to the classification of 37 data points to class A , 95 data points to class B , 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 Ω -mapping from the GMLVQ model delivers S8 Fig.

The distribution of the sequence data from the D_{NB} -data set 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 data set D_G is shown in S10 Fig.

The classification of the full D_N data set 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 RCE decision can be found in the supplementary material S14 GMLVQ Mapping for D_N .

Discussion

The classification analysis of the D_G -data by means of the machine learning model GMLVQ verifies the class determination suggested in [12]. Only 7 data samples are not classified accordingly due to the model self-controlled evidence decision. Thereby, the GMLVQ model shows a stable performance in learning (see Tab. 2), which underlies its well-known robustness [47]. Thus, we observe an overall precise agreement supporting the findings in [12].

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 $d_{\Omega}(\mathbf{x}, \mathbf{w})$. Yet, the computational complexity of dissimilarity calculations for the encoded sequences is only $O(64 \cdot m)$ with $m = 2$ being the mapping dimension of Ω . The BoW sequence coding takes $O(n)$ such that we have an overall complexity of $O(n + 64 \cdot m) = O(n)$ for dissimilarity calculations needed in our approach in comparison to at least $O(n^2)$ in case of alignment based dissimilarity calculations.

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, see S3 Fig, the Ω 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 [11, 12]. The AP-centroids differ slightly more than the GMLVQ prototypes, see 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 (6). However, in general the prototypes are not identical with the mean vectors of the class distribution, as emphasized in [83].

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 can not be detected as separate regions in that case. However, applying the Ω -mapping S8 Fig, the class structure becomes visible also for this data set. The reason for this discrepancy could be that both t -SNE as well as AP implicitly reflect data densities in the data space. Class densities, however, do not have to coincide with the overall data density. Thus, the Ω -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} -data set from NCBI contains only a few samples from Europe all occurring from February onward, i.e. no European data samples from December/January are available. We observe that class C for the D_G data is mainly represented in January for European samples, which confirms the findings in [12]. Thus, the small number of class C samples in 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.

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* (GMLVQ) classifier model (GMLVQ) for a data set with given virus type information, which was obtained by phylogenetic tree analysis [12]. 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 phylogentic trees, 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 data set. 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 data set 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.

Future work will consider to replace the WTA-rule (6) by a fuzzy variant (winner ranking) resulting in a probabilistic class/type assignment replacing 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.

Supporting information

S1 Fig.

Visualization of the classification-correlation matrix. The classification-correlation matrix provides the linear correlations among the triplet frequencies of the sequences, which contribute to the class discrimination.

S2 Fig.

Visualization of the GMLVQ result for D_G -data. The data as well as the the GMLVQ-prototypes are mapped using the learned Ω -matrix. The data points are colored either regarding their class assignments or regarding their reject decision. The GMLVQ-prototypes serve as class representatives. However, they are not identical with the mean vectors of the classes.

S3 Fig.

Visualization of the GMLVQ prototypes. The color of the prototypes is in agreement with the class coloring in S2 Fig. Further, the prototypes are vertically shifted by small offsets for better visualization and separation.

S4 Fig.

457

Stability of AP cluster solutions for D_{NB} -data. The number of clusters in dependence on the control parameter ζ is depicted. Plateaus refer to stable cluster solutions. Accordingly, we identify 2-, 4-, and 5-cluster solutions as most recommended.

458

459

460

461

S5 Fig.

462

Visualization of the AP clustering result for D_{NB} -data using 4 clusters. The data as well as the cluster centroids are depicted using the t -SNE.

463

464

465

S6 Fig.

Visualization of the AP cluster centroids for D_{NB} -data using the 4-cluster solution. The color of the cluster centroids is in agreement with the cluster coloring in S5 Fig. Further, the centroids are vertically shifted by small offsets for better visualization and separation.

S7 Fig.

Visualization of GMLVQ classification for the D_{NB} -data by t -SNE. The class coloring is as in S2 Fig.

S8 Fig.

Visualization of GMLVQ classification for the D_{NB} -data by Ω -mapping. The data as well as the GMLVQ prototypes are depicted. The class coloring is as in S2 Fig.

S9 Fig.

Distribution of the D_{NB} -data depending on the geographic origin and the collection date. A distribution of the data from the D_{NB} -data set with respect to the geographic sequence origin and the collection date together with the class assignments. Again, the mappings are realized by the Ω matrix. The class coloring is as in S2 Fig.

S10 Fig.

Distribution of the D_G -data depending on the geographic origin and the collection date. A distribution of the data from the D_G -data set with respect to the geographic sequence origin and the collection date together with the class assignments. Again, the mappings are realized by the Ω matrix. The class coloring is as in S2 Fig.

S11 Fig.

Visualization of GMLVQ classification for the full D_N -data by Ω -mapping. The data as well as the GMLVQ prototypes are depicted. The class coloring is as in S2 Fig.

S12 Data Data Files The data file 'S12 Data.xlsx' (Excel) contains the accession numbers of both data sets D_G and D_N . Further collection date and origin (region) are attached. For the D_G data set, additionally, the type information (class assignment) is given.

S13 Histogram Coding of Nucleotide Triplets Assignment of the histogram dimensions to the nucleotide triplets For each of the 64 nucleotide

combinations, the coding by the histogram dimensions is given in the file 'S13 Histogram Coding.xlsx'.

S14 GMLVQ Mapping for D_N Virus type assignments for the

D_N -sequences obtained by GMLVQ For each sequence in D_N , the class/type assignment obtained by the GMLVQ model is given as well as if a rejection decision was made according to the SCE of the GMLVQ model. Additionally, we provide the sequences from D_G , which were rejected by the SCE decision of the GMLVQ model. The respective file is 'S14 GMLVQ Mapping.xlsx'.

Acknowledgments

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

References

1. Andersen KG, Rambaut A, Lipkin WI, ECH, Garry RF. The proximal origin of SARS-CoV-2. *Nature Medicine*. 2020;26:450–452. doi:<https://doi.org/10.1038/s41591-020-0820-9>. (document)
2. Wu JT, Leung K, Leung GM. Nowcasting and forecasting the potential domestic and international spread of the 2019-nCoV outbreak originating in Wuhan, China: a modelling study. *The Lancet*. 2020;395:689–697. doi:[https://doi.org/10.1016/S0140-6736\(20\)30260-9](https://doi.org/10.1016/S0140-6736(20)30260-9). (document)
3. Bai Y, Jiang D, Lon JR, Chen X, Hu M, Lin S, et al. Evolution and molecular characteristics of SARS-CoV-2 genome. *bioRxiv*. 2020;(2020.04.24.058933). doi:<https://doi.org/10.1101/2020.04.24.058933>. (document)
4. Yang HC, Chen CH, Wang JH, Liao HC, Yang CT, Chen CW, et al. Genomic, geographic and temporal distributions of SARS-CoV-2 mutations. *bioRxiv*. 2020;(2020.04.22.055863). doi:<https://doi.org/10.1101/2020.04.22.055863>. (document)
5. Szostak N, Wasik S, Blazewicz J. Hypercycle. *Plos Computational Biology*. 2016;12(4):e1004853. doi:10.1371/journal.pcbi.1004853. (document)
6. Szostak N, Synak J, Borowski M, Wasik S, Blazewicz J. Simulating the origins of life: The dual role of RNA replicases as an obstacle to evolution. *PLoS ONE*. 2017;12(7):1–28. doi:10.1371/journal.pone.0180827. (document)
7. Paden CR, Tao Y, Queen K, Zhang J, Li Y, Uehara A, et al. Rapid, sensitive, full genome sequencing of Severe Acute Respiratory Syndrome Virus Coronavirus 2 (SARS-CoV-2). *bioRxiv*. 2020;(2020.04.22.055897). doi:<https://doi.org/10.1101/2020.04.22.055897>. (document)
8. Lan J, Ge J, Yu J, Shan S, Fan HZS, Zhang Q, et al. Structure of the SARS-CoV-2 spike receptorbinding domain bound to the ACE2 receptor. *Nature*. 2020;(10.1038/s41586-020-2180-5). doi:<https://doi.org/10.1038/s41586-020-2180-5>. (document)

9. Lu R, Zhao X, L J, Niu P, Yang B, Wu H, et al. Genomic characterisation and epidemiology of 2019 novel coronavirus: implications for virus origins and receptor binding. *The Lancet*. 2020;395(10224):565–574. doi:[https://doi.org/10.1016/S0140-6736\(20\)30251-8](https://doi.org/10.1016/S0140-6736(20)30251-8). (document)
10. Vasilariou M, Alachiotis N, Garefalaki J, Beloukas A, Pavlidis P. Population genomics insights into the recent evolution of SARS-CoV-2. *bioRxiv*. 2020;(2020.04.21.054122). doi:<https://doi.org/10.1101/2020.04.21.054122>. (document)
11. Guan Q, Sadykov M, Nugmanova R, Carr MJ, Arold ST, Pain A. The genomic variation landscape of globally-circulating clades of SARS-CoV-2 defines a genetic barcoding scheme. *bioRxiv*. 2020;(2020.04.21.054221). doi:<https://doi.org/10.1101/2020.04.21.054221>. (document)
12. Foster P, Foster L, Renfrew C, Forster M. Phylogenetic network analysis of SARS-CoV-2 genomes. *Proc of the National Academy of Science of the United States of America (PNAS)*. 2020;doi:<https://doi.org/10.1073/pnas.2004999117>. (document), 2
13. Quinlan JR. Induction of decision trees. *Machine Learning*. 1986;1:81–106. (document)
14. Warrow T. Computational phylogenetics: An introduction to designing methods for phylogeny estimation. Cambridge University Press; 2017. (document)
15. Quinlan JR. C4.5: Programs for Machine Learning. Morgan Kaufmann; 1993. (document)
16. Tampuu A, Bzhalava Z, Dillner J, Vicente R. ViraMiner: Deep learning on raw DNA sequences for identifying viral genomes in human samples. *PloS One*. 2019;14(9):e0222271. doi:<https://doi.org/10.1371/journal.pone.0222271>. (document)
17. Yin C. Genotyping coronavirus SARS-CoV-2: methods and implications. *arXiv*. 2020;(2003.10965v1). (document)
18. Wasik S, Szostak N, Kudla M, Wachowiak M, Krawiec K, Blazewicz J. Detecting Life Signatures with RNA Sequence Similarity Measures. *Journal of Theoretical Biology*. 2019;463:110–120. (document)
19. Levenshtein VI. Binary codes capable of correcting deletions, insertions, and reversals. *Doklady Akademii Nauk SSSR*. 1965;163(4):845–848. (document)
20. Needleman SB, Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*. 1970;48(3):443–453. (document)
21. Smith TF, Watermann MS. Identification of Common Molecular Subsequences. *Journal of Molecular Biology*. 1981;147(1):195–197. (document)
22. Gotoh O. An improved algorithm for Matching biological sequences. *Journal of Molecular Biology*. 1982;162:705–708. (document)
23. Li Y, Liu B, Cui J, Wang Z, Shen Y, Xu Y, et al. Similarities and Evolutionary Relationships of COVID-19 and Related Viruses. *arXiv*. 2020;(2003.05580). (document)

24. Blaisdell BE. A measure of the similarity of sets of sequences not requiring sequence alignment. *Proceedings of the National Academy of Sciences USA*. 1986;83:5155–5159. (document)
25. Blaisdell BE. 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. *Journal of Molecular Evolution*. 1989;29:538–547. (document)
26. Vinga S. Information theory applications for biological sequence analysis. *Bioinformatics*. 2004;15(3):376–389. doi:10.1093/bib/bbt068. (document)
27. Vinga S, Almeida JS. Alignment-free sequence comparison – a review. *Bioinformatics*. 2004;20(2):206–215. doi:10.1093/bioinformatics/bfg392. (document)
28. Yin C, Chen Y, Yau SST. A measure of DNA sequence similarity by Fourier Transform with applications on hierarchical clustering. *Journal of Theoretical Biology*. 2014;359:18–28. doi:http://dx.doi.org/10.1016/j.jtbi.2014.05.043. (document)
29. Sievers A, Bosiek K, Bisch M, Dreessen C, Riedel J, Froß P, et al. *K*-mer content, correlation, and position analysis of genome DNA sequences for the identification of function and evolutionary features. *Genes*. 2017;8(122):1–18. doi:doi:10.3390/genes8040122. (document)
30. Kolmogorov AN. Three Approaches to the Quantitative Definition of Information. *Problems of Information Transmission*. 1996;1(1):1–7. (document)
31. Cilibrasi R, Vitányi PMB. Clustering by Compression. *IEEE Transactions on Information Theory*. 2005;51(4):1523–1545. (document)
32. Li M, Chen X, Li X, Ma B, Vitányi PMB. The Similarity Metric. *IEEE Transactions on Information Theory*. 2004;50(12):3250–3264. (document)
33. Deng M, Yu C, Liang Q, He RL, Yau SST. A novel method of characterizing sequences: Genome space with biological distance and applications. *PLoS One*. 2011;6(3):e17293. (document)
34. Li Y, Tian K, Yin C, He RL, Yau SST. Virus classification in 60-dimensional protein space. *Molecular Phylogenetics and Evolution*. 2016;99:53–62. doi:http://dx.doi.org/10.1016/j.ympev.2016.03.009. (document)
35. Yu C, Hernandez T, Zheng H, Yau SC, Huang HH, He RL, et al. Real Time Classification of Viruses in 12 Dimensions. *Plos One*. 2013;8(5):e64328. (document)
36. Zielezinski A, Vinga S, Almeida J, Karlowski WM. Alignment-free sequence comparison: benefits, applications, and tools. *Genome Biology*. 2017;18(186):1–17. doi:10.1186/s13059-017-1319-7. (document)
37. Zielezinski A, Girgis HZ, Bernard G, Leimeister CA, Tang K, Dencker T, et al. Benchmarking of alignment-free sequence comparison methods. *Genome Biology*. 2019;20(144):1–18. doi:https://doi.org/10.1186/s13059-019-1755-7. (document)

38. Bhanot G, Biehl M, Villmann T, Zühlke D. Biomedical data analysis in translational research: Integration of expert knowledge and interpretable models. In: Verleysen M, editor. Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'2017). Louvain-La-Neuve, Belgium: i6doc.com; 2017. p. 177–186. (document)
39. Biehl M, Hammer B, Villmann T. Prototype-based models in machine learning. Wiley Interdisciplinary Reviews: Cognitive Science. 2016;(2):92–111. (document)
40. Bosco GL, diGangi MA. Deep Learning Architectures for DNA Sequence Classification. In: Petrosino A, Loia V, Pedrycz W, editors. Fuzzy Logic and Soft Computing Applications: Proceedings of the International Workshop on Fuzzy Logic and Applications (WILF 2016). vol. 10147 of LNCS. Cham: Springer; 2016. p. 162–171. (document)
41. Fianacca A, LaPaglia L, LaRosa M, LoBosco G, Renda G, Rizzo R, et al. Deep learning models for bacteria taxonomic classification of metagenomic data. BMC Bioinformatics. 2018;19(Suppl. 7):198. (document)
42. Sun Y, Zhu S, Ma K, Liu W, Yue Y, Hu G, et al. Identification of 12 cancer types through genome deep learning. Nature Scientific Reports. 2019;9(17256). (document)
43. Rudin C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence. 2019;1(5):206–215. (document)
44. Bittrich S, Kaden M, Leberecht C, Kaiser F, Villmann T, Labudde D. Application of an interpretable classification model on early folding residues during protein folding. BioData Mining. 2019;12(1). doi:<https://doi.org/10.1186/s13040-018-0188-2>. (document)
45. Villmann T, Saralajew S, Villmann A, Kaden M. Learning Vector Quantization Methods for Interpretable Classification Learning and Multilayer Networks. In: Sabourin C, Merele JJ, Barranco AL, Madani K, Warwick K, editors. Proceedings of the 10th International Joint Conference on Computational Intelligence (IJCCI), Sevilla. Lissabon, Portugal: SCITEPRESS - Science and Technology Publications, Lda.; 2018. p. 15–21. (document)
46. Zeng J, Ustun B, Rudin C. Interpretable classification models for recidivism prediction. Journal of the Royal Statistical Society, Series A. 2017;(0964–1998/17/180000):1–34. (document)
47. Saralajew S, Holdijk L, Rees M, Villmann T. Robustness of Generalized Learning Vector Quantization Models against Adversarial Attacks. In: Vellido A, Gibert K, Angulo C, Guerrero JDM, editors. Advances in Self-Organizing Maps, Learning Vector Quantization, Clustering and Data Visualization – Proceedings of the 13th International Workshop on Self-Organizing Maps and Learning Vector Quantization, Clustering and Data Visualization, WSOM+2019, Barcelona. vol. 976 of Advances in Intelligent Systems and Computing. Springer Berlin-Heidelberg; 2019. p. 189–199. (document)
48. Brister JR, Ako-adjai D, Bao Y, Blinkova O. NCBI Viral Genomes Resource. Nucleic Acids Research. 2014;43(D1):D571–D577. doi:10.1093/nar/gku1207. (document)

49. Clark K, Karsch-Mizrachi I, Lipman DJ, Ostell J, Sayers EW. GenBank. Nucleic Acids Research. 2015;44(D1):D67–D72. doi:10.1093/nar/gku1207. (document)
50. Lin J, Adjeroh DA, Jiang BH, Jiang Y. K_2 and K_2^* : efficient alignment-free sequence similarity measurement based on Kendall statistics. Bioinformatics. 2018;34(10):1682–1689. doi:10.1093/bioinformatics/btx809. (document)
51. Cornish-Bowden A. Nomenclature for incompletely specified bases in nucleic acid sequences: recommendations 1984. Nucleic Acids Research. 1985;13(9):3021–3030. doi:doi:10.1093/nar/13.9.3021. (document)
52. Mackay DJC. Information Theory, Inference and Learning Algorithms. Cambridge University Press; 2003. (document)
53. Kullback S, Leibler RA. On information and sufficiency. Annals of Mathematical Statistics. 1951;22:79–86. (document)
54. Vinga S, Almeida JS. Rényi continuous entropy of DNA sequences. Journal of Theoretical Biology. 2004;231:377–388. (document)
55. Villmann T, Haase S. Divergence based vector quantization. Neural Computation. 2011;23(5):1343–1392. (document)
56. Cottrell M, Hammer B, Hasenfuß A, Villmann T. Batch and Median Neural Gas. Neural Networks. 2006;19:762–771. (document)
57. Bauer HU, Herrmann M, Villmann T. Neural Maps and Topographic Vector Quantization. Neural Networks. 1999;12(4–5):659–676. (document)
58. Miyamoto S, Ichihashi H, Honda K. Algorithms for Fuzzy Clustering. vol. 229 of Studies in Fuzziness and Soft Computing. Springer; 2008. (document)
59. Li J, Song S, Zhang Y, Zhou Z. Robust K -Median and K -Means Clustering Algorithms for Incomplete Data. Mathematical Problems in Engineering. 2016;2016(Article ID 4321928):1–8. doi:http://dx.doi.org/10.1155/2016/4321928. (document)
60. Martinetz TM, Berkovich SG, Schulten KJ. 'Neural-Gas' Network for Vector Quantization and its Application to Time-Series Prediction. IEEE Trans on Neural Networks. 1993;4(4):558–569. (document)
61. Villmann T, Claussen JC. Magnification Control in Self-Organizing Maps and Neural Gas. Neural Computation. 2006;18(2):446–469. (document)
62. Frey BJ, Dueck D. Clustering by Message Passing Between Data Points. Science. 2007;315:972–976. (document)
63. Pearl J. Probabilistic Reasoning in Intelligent System. Morgan Kaufmann; 1988. (document)
64. Luxburg UV. A Tutorial on Spectral Clustering. Statistics and Computing. 2007;17(4):395–416. (document)
65. Kohonen T. Learning Vector Quantization. Neural Networks. 1988;1(Supplement 1):303. (document)
66. Sato A, Yamada K. Generalized learning vector quantization. In: Touretzky DS, Mozer MC, Hasselmo ME, editors. Advances in Neural Information Processing Systems 8. Proceedings of the 1995 Conference. Cambridge, MA, USA: MIT Press; 1996. p. 423–9. (document)

67. Kaden M, Lange M, Nebel D, Riedel M, Geweniger T, Villmann T. Aspects in Classification Learning - Review of Recent Developments in Learning Vector Quantization. *Foundations of Computing and Decision Sciences*. 2014;39(2):79–105. (document)
68. Graf S, Lushgy H. *Foundations of Quantization for Probability Distributions*. vol. 1730 of *Lect. Notes in Mathematics*. Berlin: Springer; 2000. (document)
69. Robbins H, Monro S. A stochastic approximation method. *Ann Math Stat*. 1951;22:400–407. (document)
70. Mwebaze E, Schneider P, Schleif FM, Aduwo JR, Quinn JA, Haase S, et al. Divergence based classification in Learning Vector Quantization. *Neurocomputing*. 2011;74(9):1429–1435. (document)
71. Schneider P, Hammer B, Biehl M. Distance learning in discriminative vector quantization. *Neural Computation*. 2009;21:2942–2969. (document)
72. Bunte K, Schneider P, Hammer B, Schleif FM, Villmann T, Biehl M. Limited Rank Matrix Learning, discriminative dimension reduction and visualization. *Neural Networks*. 2012;26(1):159–173. (document)
73. Schneider P, Hammer B, Biehl M. Adaptive Relevance Matrices in Learning Vector Quantization. *Neural Computation*. 2009;21:3532–3561. (document)
74. Schneider P, Bunte K, Stiekema H, Hammer B, Villmann T, Biehl M. Regularization in Matrix Relevance Learning. *IEEE Transactions on Neural Networks*. 2010;21(5):831–840. (document)
75. Villmann T, Bohnsack A, Kaden M. Can Learning Vector Quantization be an Alternative to SVM and Deep Learning? *Journal of Artificial Intelligence and Soft Computing Research*. 2017;7(1):65–81. (document)
76. Schleif FM, Villmann T, Hammer B, Schneider P. Efficient kernelized prototype based classification. *International Journal of Neural Systems*. 2011;21(6):443–457. (document)
77. Villmann T, Haase S, Kaden M. Kernelized Vector Quantization in Gradient-Descent Learning. *Neurocomputing*. 2015;147:83–95. (document)
78. Fischer L, Hammer B, Wersing H. Efficient rejection strategies for prototype-based classification. *Neurocomputing*. 2015;169:334–342. (document)
79. Herbei R, Wegkamp MH. Classification with reject option. *The Canadian Journal of Statistics*. 2006;34(4):709–721. (document)
80. Hinton GE, Roweis ST. Stochastic Neighbor Embedding. In: *Advances in Neural Information Processing Systems*. vol. 15. Cambridge, MA, USA: The MIT Press; 2002. p. 833–840. (document)
81. Bryson MC. Heavy-Tailed Distributions: Properties and Tests. *Technometrics*. 1974;16(1):61–68. doi:10.2307/1267493. (document)
82. van der Maaten LJP, Hinton GE. Visualizing High-Dimensional Data Using *t*-SNE. *Journal of Machine Learning Research*. 2008;9:2579–2605. (document)
83. Oehler KL, Gray RM. Combining Image Compression and Classification Using Vector Quantization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1995;17:461–473. (document)