Angewandte Computer- und Biowissenschaften

HOCHSCHULE MITTWEIDA
University of Applied Sciences

HOCHSCHULE MITTWEIDA
University of Applied Sciences

HOCHSCHULE MITTWEIDA
University of Applied Sciences

HOCHSCHULE MITTWEIDA
University of Applied Sciences

MASTER THESIS

B. Sc.
Lynn Vivian Reuss

Towards a Sequence Evolutionary Model of Influenza A Neuraminidase based on Evolutionary Coupling Analyses and Interpretable Machine Learning Models

2022

Faculty of **Applied Computer Sciences and Biosciences**

MASTER THESIS

Towards a Sequence Evolutionary Model of Influenza A Neuraminidase based on Evolutionary Coupling Analyses and Interpretable Machine Learning Models

Author:
Lynn Vivian Reuss

Study Programme:
Genomic Biotechnology (M. Sc.)

Seminar Group:
GB19wB-M

First Referee:
Prof. Dr. rer. nat. habil. Thomas Villmann

Second Referee:
M.Sc. Florian Heinke

Mittweida, 01 April 2022

"*Nothing in this world is so small that we cannot find a way to understand it, and nothing is so large that we cannot seek to confront it.*"
Hank Green, Journey to the Microcosmos

Bibliographic Information

Reuss, Lynn Vivian: Towards a Sequence Evolutionary Model of Influenza A Neuraminidase based on Evolutionary Coupling Analyses and Interpretable Machine Learning Models, 91 pages, 39 figures, Hochschule Mittweida, University of Applied Sciences, Faculty of Applied Computer Sciences and Biosciences

Master Thesis, 2022

Satz: LaTeX

Abstract

Influenza A viruses are responsible for the outbreak of epidemics as well as pandemics worldwide. The surface protein neuraminidase of this virus is responsible, among other things, for the release of virions from the cell and is thus of interest in pharmacological research. The aim of this work is to gain knowledge about evolutionary changes in sequences of influenza A neuraminidase through different methods. First, *EVcouplings* is used with the goal of identifying evolutionary couplings within the protein sequences, but this analysis was unsuccessful. This is probably due to the great sequence length of neuraminidase. Second, the natural vector method will be used for sequence embedding purposes, in hopes to visualize sequential progression of the virus protein over time. Last, interpretable machine learning methods will be applied to examine if the data is classifiable by the different years and to gain information if the extracted information conform to the results from the EVcouplings analysis. Additionally to using the class label year, other labels such as groups or subtypes are used in classification with varying results. For balanced classes the machine learning models performed adequately, but this was not the case for imbalanced data. Groups and subtypes can be classified with a high accuracy, which was not the case for the years, continents or hosts. To identify the minimal number of features necessary for linear separation of neuraminidase group 1 subtypes, a logistic regression was performed at last, resulting in the identification of 15 combinations of nine amino acid frequencies. Since the sequence embedding as well as the machine learning methods did not show neuraminidase evolution over time, further research is necessary, for example with focus on one subtype with balanced data.

Kurzbeschreibung

Entwicklung eines Sequenz-Evolutionsmodells der Influenza A Neuraminidase auf Grundlage von evolutionary couplings Analysen und interpretierbaren Modellen des maschinellen Lernens

Influenza A Viren sind weltweit für den Ausbruch von Epidemien und Pandemien verantwortlich. Das Oberflächenprotein Neuraminidase dieses Virus ist u.A. für die Freisetzung der Virionen aus der Zelle verantwortlich und somit Bestandteil pharmakologischer Forschungen. Ziel dieser Arbeit ist es, durch verschiedene Methoden Erkenntnisse über evolutionäre Veränderungen in Sequenzen der Influenza A Neuraminidase zu gewinnen. Zunächst wird *EVcouplings* mit dem Ziel eingesetzt, *evolutionary couplings* innerhalb der Proteinsequenzen zu identifizieren, jedoch war diese Analyse nicht erfolgreich. Dies ist wahrscheinlich auf die Sequenzlänge der Neuraminidase zurückzuführen. Zweitens wird die *natural vectors* Methode auf die Proteinsequenzen angewendet, in der Hoffnung, die sequenzielle Entwicklung des Virusproteins im Laufe der Zeit zu visualisieren. Schließlich werden interpretierbare Methoden des maschinellen Lernens angewandt, um zu untersuchen, ob die Daten nach den verschiedenen Jahren klassifiziert werden können und um Informationen darüber zu gewinnen, ob die extrahierten Informationen mit den Ergebnissen der EVcouplings-Analyse übereinstimmen. Neben der Verwendung der Jahre als Klassenlabel werden auch andere Labels wie Gruppen oder Subtypen bei der Klassifizierung verwendet, mit unterschiedlichen Ergebnissen. Bei balancierten Klassen erzielten die maschinellen Lernmodelle gute Ergebnisse, bei imbalancierten Daten war dies jedoch nicht der Fall. Gruppen und Subtypen können mit einer hohen Genauigkeit klassifiziert werden, was bei den Jahren, Kontinenten oder Wirten nicht zutraf. Um die minimale Anzahl von Merkmalen zu ermitteln, die für eine lineare Trennung der Subtypen der Neuraminidasegruppe 1 erforderlich sind, wurde anschließend eine logistische Regression durchgeführt, die zur Identifizierung von 15 Kombinationen aus neun Aminosäurehäufigkeiten führte. Da die Visualisierung der *natural vectors* als auch die Methoden des maschinellen Lernens keine Evolution der Neuraminidase im Laufe der Zeit aufzeigten, sind weitere Untersuchungen notwendig, zum Beispiel mit Fokus auf einen Subtyp mit gleichgewichtetem Datensatz.

I. Contents

II. List of Figures

III. List of Tables

IV. Nomenclature

.....

A, Ala ..... Alanine

C, Cys ..... Cysteine

D, Asp ..... Aspartate

E, Glu ..... Glutamate

F, Phe ..... Phenylalanine

G, Gly ..... Glycine

H, His ..... Histidine

I, Ile ..... Isoleucine

K, Lys ..... Lysine

L, Leu ..... Leucine

M, Met ..... Methionine

N, Asn ..... Asparagine

P, Pro ..... Proline

Q, Gln ..... Glutamine

R, Arg ..... Arginine

S, Ser ..... Serine

T, Thr ..... Threonine

V, Val ..... Valine

W, Trp ..... Tryptophan

Y, Tyr ..... Tyrosine

Abbreviations ....

ARS ..... Attraction Repulsion Scheme

DSB ..... Disulfide brigde

EC ..... Evolutionary coupling

GLVQ ..... Generalized Learning Vector Quantization

GMLVQ ..... Generalized Matrix Learning Vector Quantization

H, HA ..... Hemagglutinin

IAV ..... Influenza A virus

ID ..... identifier

LVQ ..... Learning Vector Quantization

ML ..... Machine Learning

MSA ..... Multiple Sequence Alignment

N, NA ..... Neuraminidase

NG ..... Neural Gas

PC ..... Principal Component

PCA ..... Principal Component Analysis

RNA ..... Ribonucleic Acid

SGD ..... Stochastic Gradient Descent

WTA rule ..... Winner-Takes-All rule

Mathematical Symbols

α ..... Y intercept

β ..... regression coefficient

$\hat{c} \in C$ ..... Predicted data class

$\mathbf{w} \in W$ ..... Prototype

$\mathbf{x} \in X$ ..... Datapoint

$\mathscr{A}$ ..... set of 20 amino acids, totality of 20 amino acids

μ_a ..... mean position of amino acid a in Sq

a ..... Amino acid

a_i ..... Amino acid at sequence position i

$c \in C$ ..... Data class

C ..... Set of data classes

$d(\mathbf{x}, \mathbf{y})$ ..... Dissimilarity between the vectors $\mathbf{x}$ and $\mathbf{y}$

D_j^a ..... normalized central moment (Distribution of amino acid a in Sq)

n_a ..... Absolute frequency of amino acid a

n_w ..... Number of prototypes

n_x ..... Number of datapoints

Sq ..... Biological Sequence

T ..... Training dataset

W ..... Set of prototypes

X ..... Set of datapoints

Databases and Programs

EMBL-EBI ..... European Molecular Biology Laboratory-European Bioinformatics Institute

NCBI ..... National Center for Biotechnology Information

PDB ..... Protein Data Bank

PFAM ..... Protein Families Database

SIAS ..... Sequence Identity and Similarity

T-Coffee ..... Tree-based consistency objective function for alignment evaluation

Acknowlegment

I would like to take this opportunity to thank all people who have accompanied and supported me throughout my studies and this master thesis.

First of all, I would like to thank my thesis advisor, Professor Dr. rer. nat. habil. Thomas Villmann of the Faculty of Applied Computer- and Biosciences at Hochschule Mittweida for giving me the opportunity to write my master thesis at the HSMW and for his support and exchange during my thesis. Thank you for all of the occasions I was given to further my research and extend my knowledge, e.g. on conferences at which there were plenty of fruitful discussions and horizon-opening moments.

A special thanks is due to my supervisor Florian Heinke of the Faculty of Applied Computer- and Biosciences at Hochschule Mittweida whose expertise and knowledge was and always will be invaluable for me. Your insightful feedback, your enthusiasm and your constructive criticism kept me motivated in these trying times. I would like to thank you very much for your support and understanding over these past years and for your valuable guidance throughout my studies.

Further, I could not have completed this thesis without the support of the whole Villy-team. Marika, Julia, Jensun, Mirko, Daniel, Katrin, Julius, and the Rest of the team: Thank you all for the valuable help, all the stimulating discussions, and coffee by the liter.

To all my fellow students who have become valued friends: Thank you for your support, your ideas and the happy distractions, when I needed them.

Finally, this accomplishment would not have been possible without the patience and trust all of my parents have in me. Thank you!

1 Biological Fundamentals

1.1 Motivation

On earth, there are an estimated 10^{31} viruses, which can be found in every part of the natural world (Breitbart et al., 2005). Of those nonillions of viruses, humans are susceptible for approximately 200 virus species (Woolhouse et al., 2012), with the influenza virus being one of those. Surface proteins of influenza A viruses, such as hemagglutinin and neuraminidase, which make the first contact with the host cell, are subject to frequent changes in their sequences. Those changes on biological sequences, called mutations, can be observed over a period of time. Blackshields et al. (2010) developed a new way of guide tree generation for multiple sequence alignment and by doing so, they visualized 3994 hemagglutinin H3 sequences over a period of 41 years. This visualization (Fig. 1) represents the first three principal components of the embedded vectors using principal component analysis, where the datapoints symbolize the protein sequences, colored from 1967 (blue) to 2008 (red). Thus, the almost linear progression of the protein through time is made visible.

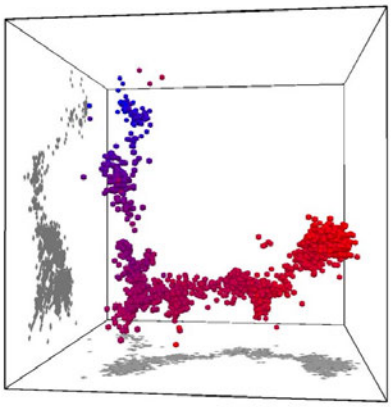


Figure 1: Blackshields et al. (2010) visualization of progressive changes at the amino acid sequence level of 3994 sequences of the influenza A surface protein hemagglutinin from 1967 (blue) up to 2008 (red), with each dot representing a sequence colored by year of isolation.

Based on this idea, this work will focus on the achievement of similar results using the neuraminidase surface protein of influenza A. In a first step, it will be investigated, if similar results can be achieved using the application *EVcouplings* with the goal of identifying evolutionary couplings within the protein sequences. These will provide insight into evolutionary changes and conservation in the sequences over the last 100 years and consequently demonstrate viral movement to facilitate the prediction of future epidemics or pandemics (Shortridge, 1995; Taubenberger, Morens, and Fauci, 2007).

To substantiate the conclusions that emerge from this, the natural vector method will be used for sequence embedding purposes. Similar to Blackshields et al., the multidimensional vectors will be embedded in a low dimensional space using principal component analysis to visualize sequential progression of the virus protein over time. Furthermore, given that the biological sequences are transformed into numerical data, interpretable machine learning methods will be applied. They will be used to examine if the data is classifiable by the different years and to gain information if the extracted information conform to the results from the EVcouplings analysis. In addition to the studies using the years as class labels, it will be investigated, whether machine learning models can accurately classify further characteristics of the neuraminidase, such as individual neuraminidase groups or subtypes. Hence, the different classification problems will be of multi-class or binary nature.
Due to the nature of neuraminidase structure, it is assumed, that the dataset can be classified by groups using machine learning approaches, and based on the specifics of natural vectors, it may be possible that changes in the amino acid sequence stand out in an embedded space or using the machine learning methods.

1.2 Influenza A Virus

Influenza viruses of the family of *Orthomyxoviridae* are classified into four types A, B, C and D. Influenza A viruses (IAVs) are mainly responsible for seasonal influenza epidemics as well as pandemics in humans. Some major human epidemics and pandemics of the 20^{th} century have been the *Spanish Flu* in 1918, the *Asian Flu* 1957 and the *Hong Kong Flu* in 1968 (Ma et al., 2009). Between 1918 and 1920 the *Spanish Flu* caused by H1N1 killed approx. 50 million people (Lina, 2008). The influenza A subtype responsible for this pandemic bears the name of *mother of all pandemics*, not necessarily due to the extraordinary severity of the disease. Rather, the 1918 IAV appears to be the possible genetic ancestor of various human and porcine influenza A subtype strains (Taubenberger and Morens, 2006). It reappeared in the 21^{st} century, in 2009–2010, as H1N1pdm09, called *Swine Flu*, and was less virulent in regard to its overall morbidity and mortality (Lycett et al., 2019).

1.2.1 IAV Subtypes

The influenza A virus (IAV) has its genetic code, eight different ribonucleic acid (RNA) segments, enveloped by a membrane, and those RNA segments code for eleven viral proteins essential for the structure and function of the virus. Figure 2 shows a schematic representation of an influenza A virion. The segments encoding two specific surface proteins are highlighted. These two proteins, hemagglutinin (HA) and neuraminidase (NA), which make the first contact with the host cell, are used to distinguish the different influenza A subtypes. So far, 18 hemagglutinin subtypes (abbreviated H1-H18) have been identified. For neuraminidase, a total of eleven subtypes (abbreviated N1-N11) have been identified (Air, 2012; Q. Li et al., 2012; Zhu et al., 2012). The three influenza A subtypes H1N1, H2N2 and H3N2 have persisted in the human population (Dou et al., 2018) and a further two influenza A subtypes (H17N10 and H18N11) have only been found in bats. These latter do not seem to be able to infect other animal species than bats and therefore do not seem to pose a threat for epidemics or pandemics among humans (Wu et al., 2014; Tong et al., 2012). Wild birds seem to be the predominant hosts for all other IAV subtypes, especially Charadriiformes (gulls, terns, sandpipers) and Anseriformes (ducks, geese, swans) the natural reservoir of IAV (Dou et al., 2018; Lycett et al., 2019). Due to the nature of these waterfowl, they can spread the virus along migratory flyways (Zhang et al., 2014; Olsen et al., 2006).

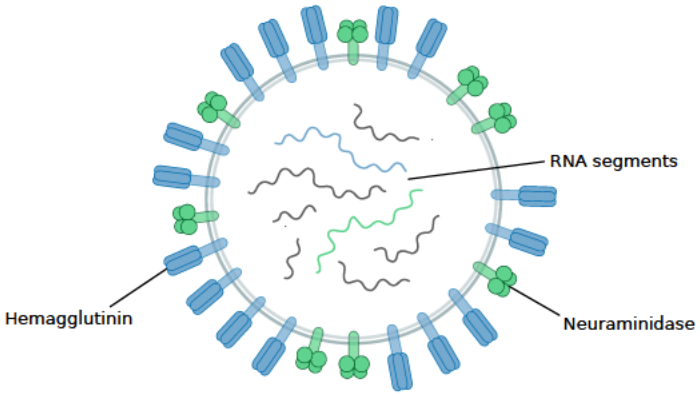


Figure 2: Schematic representation of an influenza A virion. The surface protein hemagglutinin and its coding RNA sequence are highlighted in blue, while the surface protein neuraminidase and its coding sequence are shown in green. There are an additional six RNA segments in the viral cell, which provide the genetic code for nine other viral proteins (McAuley et al., 2019).

The single virus particle, the virion, can undergo two types of genetic changes. In the case of *antigenic shift*, simultaneous infection of a host cell by two or more IAV subtypes can result in genetic rearrangement of the RNA segments, therefore also called *reassortment*. The resulting virion thus contains a combination of the RNA segments of the parent viruses (Labella et al., 2013; Flaherty, 2012). Pigs, in particular, are considered to be so-called *mixed vessels*, because they are susceptible to many influenza A subtypes of avian, porcine, and human origin, and antigenic shifts are increased in these animals (Barry, 2005).
Smaller genetic changes in the viral genome are referred to as *antigen drift*. Here, mutations occur in the genome, such that neuraminidase undergoes changes in its amino acid sequence, but the neuraminidase subtype and function of the protein remain the same (Labella et al., 2013). Those genetic alterations can evade the immune defenses of the infected host and thus reduce vaccine efficacy (Fox et al., 2019), which can result in epidemics and pandemics (Palese, 2004).

1.2.2 Influenza A Neuraminidase

1.2.2.1 Structure and Function

Neuraminidase, as one of the major surface proteins of influenza virus, plays an important role during the release and spread of virions throughout the host, yet it is also responsible for viral entry into the host cell. These functions make the protein one of the main targets for vaccine development, as not only can infection of a host cell be prevented, but also the spread of the virus in the already infected host organism (Labella et al., 2013). The protein consists of four identical polypeptides, which arrange into a tetramer. Each monomer has a sequence length of approximately 470 amino acids (Air, 2012) and can be divided into four domains: a cytoplasmic tail, a transmembrane region, a stem, and a head. The cytoplasmic tail is located within the virus and is conserved across nearly 100% of all subtypes, since mutations of this region result in altered virion morphology and reduced replication yields. The transmembrane region is located in the virion membrane and is thought to be α-helical (McAuley et al., 2019). The stem, which is located outside the virion, can vary in its number of amino acids. The length of the stem is specific to the subtype and has significant implications for viral properties. Commonly observed, for example, is a deletion of 20 amino acids in the transmission of IAV from waterfowl to domestic poultry. That phenomenon is thought to be a viral adaptation to the host being infected and thus species-specific (Y. Li, Chen, et al., 2014). The neuraminidase monomer head sits at the end of the stem and consists of in average 389 amino acids. It is described in its secondary structure as a six-stranded propeller structure (McAuley et al., 2019), with each strand consisting of four antiparallel β-strands connected by loops and stabilized by disulfide bridges.

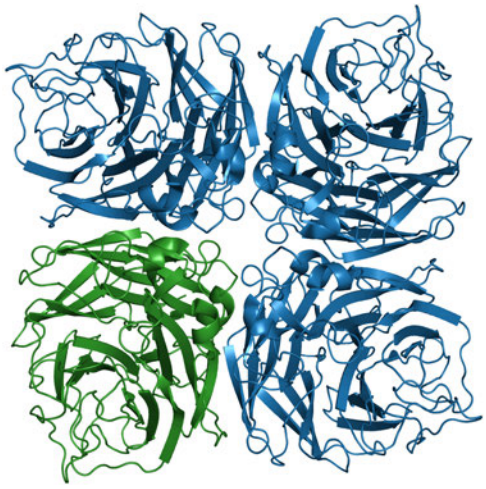
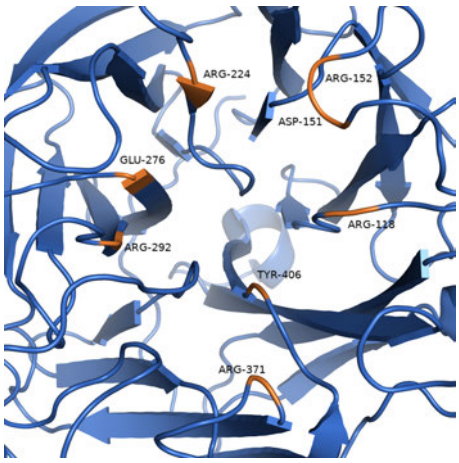
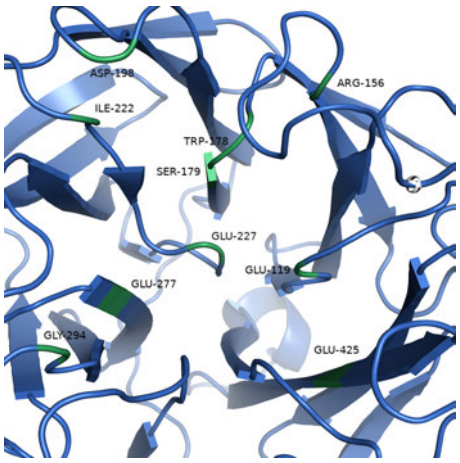


Figure 3: Neuraminidase head region as a tetramer (generated with PyMol, PDB ID: 6hg5). The single monomer is highlighted in green. Each monomer consists of six β-sheets, each with four antiparallel β-strands. The arrangement resembles a propeller, hence the term propeller structure (McAuley et al., 2019).

At the head of the neuraminidase is the active site of the protein. It interacts with the corresponding host cell during viral entry, precisely with terminal sialic acid residues of glycoproteins. The sialidase function is crucial for neuraminidase, as it identifies the correct receptor on the surface of the host cell (H. Wang, 2020). In Figure 3, the neuraminidase head is shown as a tetramer, with a single monomer highlighted in green.
The active site of a monomer consists of eight highly conserved functionally important amino acids, which are Arginine (R/Arg), Aspartate (D/Asp), Glutamate (E/Glu) and Tyrosine (T/Tyr) at the following protein sequence positions (indicated as residue + position by N2 numbering): Arg118, Asp151, Arg152, Arg224, Glu276, Arg292, Arg371, and Tyr406. This 'inner shell' is surrounded by amino acids, which stabilize the structure of the active site and are therefore defined as the 'outer shell' resp. framework residues. They are equally highly conversed and consist of Glutamate (E/Glu), Arginine (R/Arg), Tryptophan (W/Trp), Serine (S/Ser), Aspartate (D/Asp), Isoleucine (I/Ile), Asparagine (N/Asn) at following positions Glu119, Arg156, Trp178, Ser179, Asp198, Ile222, Glu227, Glu277, Asn294, and Glu425 (McAuley et al., 2019). In Figure 4, the functional and structural amino acids are illustrated, with functional residues shown in orange (Fig. 4a) and structural residues in green (Fig. 4b).



(a) Functional residues in active site of neuraminidase N8 (generated with PyMol, PDB ID: 2ht5).



(b) Framework residues in active site of neuraminidase N8 (generated with PyMol, PDB ID: 2ht5).

Figure 4: Functional (orange) and framework (green) residues in active site of neuraminidase N8 (generated with PyMol, PDB ID: 2ht5).

1.2.2.2 Neuraminidase Groups and Subtypes

Neuraminidase subtypes can be divided into three groups depending on the structure of the head region and genetic relationship. The first group includes **N1**, **N4**, **N5**, and **N8**. These have an additional cavity next to the active site, which is formed by the binding of a substrate to the active site. In the second group, which includes **N2**, **N3**, **N6**, **N7**, and **N9**, such a cavity has not been observed yet, and the structural flexibility does not seem to be present (Amaro et al., 2011; Air, 2012). This distinction is pharmacologically relevant. Drugs can be developed that bind in the cavity created, thereby more effectively preventing the virus from spreading (Russell et al., 2006). In the last decade, two additional IAV subtypes H17N10 and H18N11 were detected in New World bats of South America. Because of many differences from previously known IAV, these were first described as *influenza-like viruses*. Both hemagglutinin and neuraminidase show differences in their amino acid sequence and in their structure from previously known protein subtypes (Wu et al., 2014). N10 and N11 share only 19.8%–27.1% sequence identity with other NA subtypes (Tong et al., 2012), but are nevertheless structurally similar to N1-N9. For example, they show the typical tetramer structure of the head domain (see Figure 3), in spite of the fact that they lack some highly conserved functional and structural residues. They possess only three out of eight conserved functional and three out of eleven conserved structurally important residues, resulting in the inability of these neuraminidases to perform the actual neuraminidase function (Q. Li et al., 2012). This was confirmed in *in vitro* experiments, as H17N10 and H18N11 have not yet been able to propagate in cell cultures, which hinders further research (Q. Li et al., 2012; Wu et al., 2014), and therefore it remains unclear what function they perform instead (Zhong et al., 2020) and whether these subtypes may be derived from a previously unknown or extinct influenza A virus (Tong et al., 2012). However, phylogenetic analyses of the whole genome confirmed their genetic relationship to the other neuraminidase subtypes (Ciminski et al., 2017). Figure 5 shows the phylogenetic tree of all known NA subtypes, which illustrates the genetic relationship between them. Relative to the genetic, antigenic, and phenotypic differences of the individual subtypes, the subtypes appear to be genetically relatively conserved and antigenically and phenotypically homogeneous (Webster, Bean, et al., 1992).

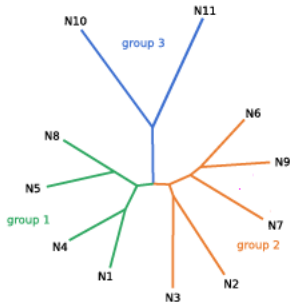


Figure 5: Phylogenetic tree of neuraminidase subtypes according to Wu et al., 2014. Group 1 (N1, N4, N5, N8) is highlighted in green, group 2 (N2, N3, N6, N7, N9) in orange, and group 3 (N10, N11) in blue.

1.3 Outline of this Work

The following chapter "*2 Applied Methods*" is devoted to the methods used in this thesis. First, evolutionary couplings are described in more detail as well as the software programs EVcouplings and plmc are presented. Then, the supervised and unsupervised machine learning algorithms are discussed. Among those, Neural Gas (NG) is applied for unsupervised vector quantization, and its application for dataset balancing is described in Chapter 3. Hereafter, different variants of Learning Vector Quantization (LVQ) as supervised machine learning algorithms are discussed with the purpose to give an overview of the applied methods and to lead to the interpretability of the LVQ variant Generalized Matrix LVQ (GMLVQ), resp. the Λ matrix of GMLVQ. Furthermore, the classification validation is presented, as well as the transformation from biological sequences into vectors via the natural vector method. All analyzed protein sequences are represented by their respective natural vector for the aforementioned machine learning methods. Chapter "*3 Data Acquisition*" describes in detail, how the neuraminidase protein sequences are acquired. Here, the sequences are downloaded from the Protein Data Bank (PDB) an the National Center for Biotechnology Information (NCBI). All proteins acquired from the PDB are structurally resolved and therefore can give an insight into the neuraminidase structure and structural peculiarities. Regardless of the database, nearly all sequences are annotated with additional information regarding the NA group and NA subtype membership, as well as year, place and organism of isolation, sequence length and more. Furthermore, dataset balancing and filtering will be discussed in this chapter, where data filtering is described with emphasis on filtering redundant sequences out of the dataset. In the next chapter "*4 EVcouplings Analysis*", the evolutionary couplings analysis is performed with the software EVcouplings. The mode of operation/functionality of EVcouplings and the data necessary should clarify the coupling results later generated. The chapter "*5 Neuraminidase Natural Vector Embedding*" deals with the transformation of biological sequences into vectors for machine learning analyses. The natural vectors are then visualized by showing the first two or first three principal components after a principal component analysis (PCA) and the datapoints are colored by different labels (e.g. by NA group membership or host). This is followed by the chapter "*6 Classification using the GMLVQ approach*" dedicated to the analysis of the data using the GMLVQ method. There, the GMLVQ model is verified with 5-fold cross validation. The natural vectors of the NA sequences are defined as input and the class labels are set to different additional information concerning the sequences such as the NA group, NA subtype, year of isolation, virus host or continent of isolation. The results are discussed by the interpretation of the output Λ matrix. To interpret this matrix further, the subsequent chapter "*7 Logistic Regression Modeling for GMLVQ interpretation*" takes a closer look on the important features for the classification decision of GMLVQ. There, the NA subtypes of every NA group 1 are tested against the others from the same group for identification of important features to the classification.
Finally, in the last chapter a summary and conclusions as well as remarks to future works with neuraminidase sequences are provided.

2 Applied Methods

2.1 EVcouplings Analysis

For the identification of neuraminidase evolution over the last 100 years, the individual amino acids following evolutionary constraint need to be identified. To achieve this, *EVcouplings* (EVC) is used to determine evolutionary couplings. This chapter is based on the work of Marks, Colwell, et al. (2011), Marks, Hopf, et al. (2012), Ekeberg et al. (2013), Hopf, Schärfe, et al. (2014), and Hopf, Ingraham, et al. (2017). It gives an overview of the functionality of the EVcouplings program as described in Hopf, Green, et al. (2018). More detailed descriptions can be found in the aforementioned publications.

2.1.1 Evolutionary Couplings

The information about the structure and function of biomolecules such as proteins is contained in their sequence, but evolutionary pressure can lead to changes on sequence level. These can lead to improved or reduced functionality of a protein due to conformational alterations in the protein structure (Hopf and Marks, 2017). Since a reduced functionality of a protein would be unfavorable for the persistence of an organism, evolutionary constraint leads to the preservation of the necessary interactions between amino acids, which are indispensable for the formation of stable and functional proteins. This in turn leads to coevolution of interacting amino acids, hence called *evolutionary couplings* (Marks, Colwell, et al., 2011; Hopf, Ingraham, et al., 2017; Hopf and Marks, 2017). Figure 6 illustrates those couplings, where two interacting residues are structurally close to each other. The interaction can occur, for example, through a hydrogen bridge, a disulfide bond or other. In Fig. 6 (A) the two interacting amino acid residues are shown as blue and green circle, and the fold of the protein is visualized as curved line. To maintain this contact, either one of those residues must coevolve with the other or both must remain conserved to sustain functionality of the protein. In Fig. 6 (B) this coevolution in related protein sequences is illustrated schematically, with the sequences from time 1918–2000 in a multiple sequence alignment represented as horizontal lines and the respective residues as aforementioned circles (Marks, Hopf, et al., 2012). A closer look at the MSA reveals the coevolved amino acids, as seen in Fig. 6 (C), so that a correlation between one amino acid to another can be suggested.

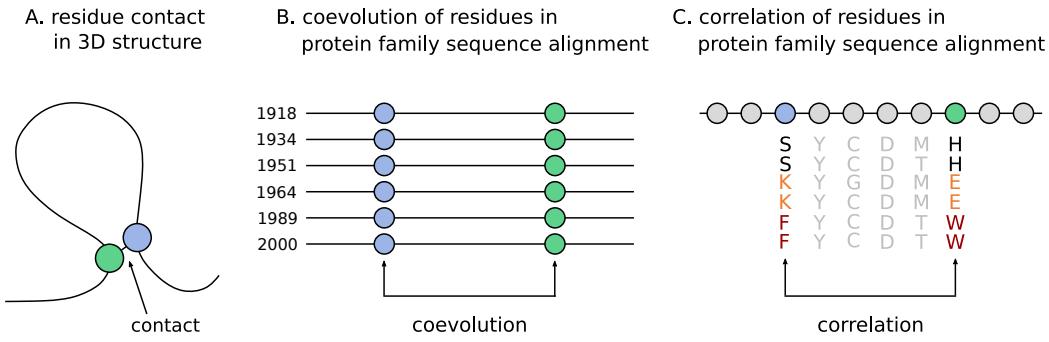


Figure 6: Residues required for the structural stability of a protein are in close 3D proximity to one another and form a residue contact (A). This contact provides a record of residue covariation based on the evolution in the protein sequences of one family (B), where the individual amino acid pair can be identified (C). Therefore, they are also called *evolutionary couplings* (adapted from Hopf and Marks (2017) and Ekeberg et al. (2013)).

2.1.2 EVcouplings

EVcouplings is an open source, integrated pipeline used for predicting structure, function and mutations in protein sequences by means of evolutionary sequence covariation. It was developed in the laboratories of Debora Marks and Chris Sander at Harvard Medical School, with Thomas Hopf as the development lead. EVC is available either on the EVcouplings Website[1] or as command-line application respectively as Python package. An essential feature of EVC is the identification of evolutionary contacts, also called evolutionary couplings (ECs), in proteins. With those ECs, e.g. residue contacts, mutational effects and 3D structures of proteins can be predicted. To achieve this, EVC uses external software tools such as *plmc* (Hopf, Ingraham, et al., 2017) and *HHsuite* (Steinegger et al., 2019) among others, and previously published methods like *EVfold* (Marks, Colwell, et al., 2011), *EVmutation* (Hopf, Ingraham, et al., 2017) and *EVcomplex* (Hopf, Schärfe, et al., 2014).
The complete EVcouplings pipeline is divided into five stages: First, a sequence alignment is either generated from unaligned sequences or an existing alignment loaded. This only applies for the application EVcouplings running on a computer. Using the Website of EVC, one sequence or Uniprot ID needs to be provided. This target sequence is then used to search a redundancy reduced database (e.g. UniRef90) for homologs. The alignment stage also preprocesses the alignment for the following couplings calculation. Secondly, referred to as the couplings stage, the evolutionary couplings are calculated via *plmc*. The output is subsequently processed to achieve the correct numbering of the amino acids and a statistical scoring model is fitted to the ECs. *Plmc*, written by John Ingraham in the laboratory of Debora Marks, is a tool for the deduction of undirected statistical models to describe coevolution and covariation in biological sequence fami-

[1] https://evcouplings.org/

lies. The tool quantifies amino acid or nucleotide coupling strengths between all pairs of sequence positions on the basis of the given MSA. Thereafter, the fold stage being the third stage in the pipeline deals with generating *de novo* protein models utilizing the determined ECs. Fourthly, predicted effects of mutations can be generated during the mutate stage and later visualized. Lastly, the evolutionary couplings are compared to existing structures and a contact map is generated in the compare stage (Hopf, Green, et al., 2018). Figure 7 visualizes the stages of the EVcouplings pipeline.

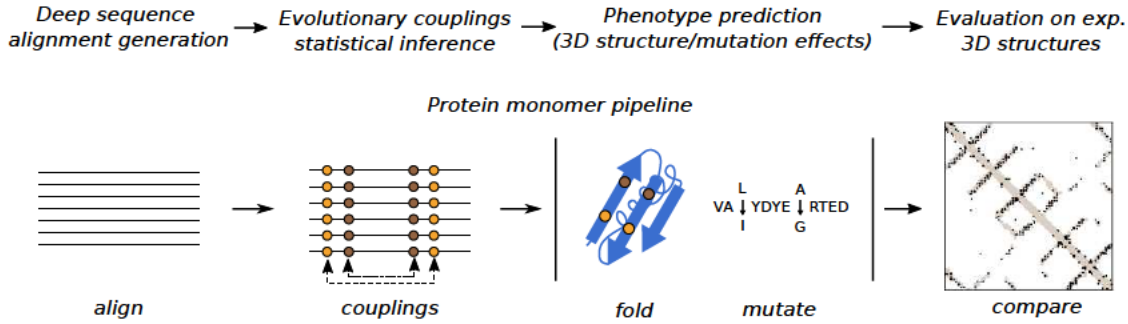


Figure 7: Five stages of the EVcouplings pipeline *align*, *couplings*, *fold*, *mutate*, *compare* (adapted from Hopf, Green, et al. (2018)).

2.2 Natural Vector for Protein Sequence Vector Embedding

Due to the fact that especially distance-based machine learning approaches require some sort of numerical representation of the data, protein sequences can be converted from their primary structure into vectors with the natural vector (NV) method and it is therefore a feature generator (Bohnsack et al., 2022).
Assume a sequence $Sq = (a_1, a_2, \ldots, a_n)$ of length n. The a_i are elements of the set $\mathscr{A} = \{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, U, V, W, Y\}$ and of the 20 amino acids. Even though first described for nucleotide sequences (Deng et al., 2011), the natural vector for a protein sequence consists of the same three groups of parameters (Y. Wang et al., 2019). The first group of parameters includes the absolute frequencies n_a of the 20 amino acids $a \in \mathscr{A}$ in Sq. The second group of parameters are the mean positions μ_a of amino acids a in Sq: $\mu_a = T_a/n_a$ and $T_a = \sum_{i=1}^{n_a} s_{ai}$ with s_{ai} being the position of the i-th occurrence of amino acid a_i within Sq. The third group of parameters is composed of normalized central moments which describe the distribution D_j^a of amino acids a. This distribution is calculated by the following formula:

$$D_j^a = \sum_{i=1}^{n_a} \frac{(s_{ai} - \mu_a)^j}{n_a^{j-1} n^{j-1}} \tag{2.1}$$

with $j = 2, ..., n_a$ (Y. Wang et al., 2019). In practical applications, the minimum order of normalized central moments to be calculated is fixed to j_{max} such, that the data dimension (or length of natural vector l) becomes $l = 20 \cdot (j_{max} + 1)$ for comparability. Frequently, $j_{max} = 2$ is chosen (Y. Li, Tian, et al., 2016). Therefore, the shape of a 60 dimensional natural vector with $j = 2$ is as follows:

$$\mathbf{x} = (n_A, n_C, \ldots, n_Y, \mu_A, \mu_C, \ldots, \mu_Y, D_2^A, D_2^C, \ldots, D_2^Y) \tag{2.2}$$

This method is chosen due to the advantages described by Deng et al., 2011, and Y. Wang et al., 2019. The embedding of the vectors into a high-dimensional space can be used to measure protein similarity using Euclidean distances in that space instead of using common alignment methods. Furthermore, generating natural vectors is faster than computing a multiple sequence alignment and if sequence data is added, there is no need for a realignment of the sequences. Other vector methods like *Bag of Words* (Blaisdell, 1989) are not considered due to the large alphabet of 20 amino acids, which on the one hand is computationally intensive and on the other hand would reduce the interpretability of GMLVQ results.

2.3 Machine Learning Algorithms

Machine Learning (ML), as part of artificial intelligence, refers to mathematical methods for the acquisition of knowledge. ML provides algorithms capable of self-learning from input data, which is therefore called the training dataset. The trained model is then tested on a testing dataset, which is unknown to the model. The learning itself can be differentiated between *supervised learning* and *unsupervised learning*. In *supervised learning*, the data in the training dataset is labeled and the algorithm analyzes the dataset that it is able to apply the gained knowledge to solve the specific learning task. These learning tasks can be classification or regression, which predict the label or class of unlabeled and unseen data (Choi et al., 2020; Srinivasa et al., 2020).
Unlike supervised learning, the data in *unsupervised learning* does not have label information and therefore is usually used to recognize patterns and structures in the data (Choi et al., 2020).
Hence, ML approaches can be used in bioinformatics for prediction of biological data, discrimination and classification of biological sequences or feature selections, as aforementioned biological data are generally multidimensional (Srinivasa et al., 2020).
This chapter is based on the work of Martinetz et al. (1993), Kohonen (1986), Sato et al. (1996), and Geweniger (2012) and gives an overview of the functionality of ML algorithms. Further details and mathematical derivations can be found in the original literature.

2.3.1 Neural Gas for Unsupervised Vector Quantization

One possibility for representing multidimensional data is Vector Quantization (VQ), with Neural Gas (NG) being one possible approach. To achieve this, the Neural Gas algorithm by Martinetz et al. (1993) initializes a set of prototypes $W = \{\mathbf{w}_i\} \in \mathbb{R}^d$, $i = 1, \ldots, n_w$ (with n_w being the number of prototypes) randomly in the data space $\mathbb{R}^d$. This initialization is based on the basic idea of particles propagating in a medium, more precisely on Brownian motion, where the prototypes represent the datapoints $\mathbf{x} \in \mathbb{R}^d$ as best as possible. For this reason, it will be used for dataset balancing purposes.
The cost function of NG is given as

$$E_{NG} = \frac{1}{C(\sigma)} \sum_{i=1}^{n_p} \int h_\sigma(k_i(\mathbf{x}, W)) \cdot d(\mathbf{x}, \mathbf{w}_i) P(X) dx \tag{2.3}$$

Then, stochastic gradient descent (SGD) is performed, which manifests in the random selection of one datapoint $\mathbf{x}_i$. From this datapoint, NG considers the neighborhood of the prototypes by introducing a rank function (see eq. (2.4)). In this function, the distances between all prototypes $\mathbf{w}_i$ and the datapoint $\mathbf{x}$ are calculated and sorted.

$$k_i(\mathbf{x}, W) = |\{\mathbf{w}_k | d(\mathbf{x}, \mathbf{w}_k) < d(\mathbf{x}, \mathbf{w}_i)\}| \tag{2.4}$$

Next, the prototypes are updated according to

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \varepsilon \cdot h_\sigma(k_i(\mathbf{x}, W)) \cdot \frac{\partial\, d(\mathbf{x}, \mathbf{w}_i(t))}{\partial\, \mathbf{w}_i(t)} \tag{2.5}$$

with ε being the learning rate and $h_\sigma(t)$ the neighborhood function, which increases for decreasing values of $k_i(\mathbf{x}, W)$ and returning a maximum value for $k_i(\mathbf{x}, W) = 0$.
The steps of choosing a random datapoint to updating all prototypes are repeated until the algorithm converges or it is manually stopped (Geweniger, 2012; Martinetz et al., 1993).

2.3.2 Supervised Machine Learning Methods

2.3.2.1 Logistic Regression for Linear Classification

Logistic regression is a regression method used for predicting an outcome based on previous observations in data, meaning predicting one variable (the response or dependent variable Y) based on one or more other variables (the predictors or independent variables X) (Peng et al., 2002). It is an effective classification algorithm applied on categorical or binary data and can solve binary classification problems, where the probability of a specific event or class occurring is modeled on the basis of the logistic function

$$y = \frac{1}{1 + e^{-x}}. \tag{2.6}$$

The underlying mathematical concept is the natural logarithm of an odds ratio, where the probability of the event occurring is divided by the probability of the event not occurring. Generally, the occurrence of the event is coded as 1 and its absence as 0. Assume p being the probability of the occurrence of the event (the response) taking the value of 1, then the odds of this event happening is defined as $p/(1-p)$ and therefore, the logarithm of the odds is $ln(p/(1-p))$. For logistic regression with multiple predictors, this results in

$$\text{logit}(Y) = ln\left(\frac{p}{1-p}\right) = \alpha + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k \tag{2.7}$$

where it predicts the logit of Y from X with α is the Y intercept and β_i the regression coefficients. It can be derived from the equation above, that the probability is as follows

$$P(Y=1|X_1=x_1,X_2=x_2,\ldots,X_k=x_k)=\frac{e^{\alpha+\beta_1X_1+\beta_2X_2+\ldots+\beta_kX_k}}{1+e^{\alpha+\beta_1X_1+\beta_2X_2+\ldots+\beta_kX_k}} \tag{2.8}$$

with e the base of the natural logarithm, α the Y intercept, and β_i the regression coefficients, which denotes the direction of the relationship between X and Y (Peng et al., 2002). Logistic regression is used in this thesis to extrapolate the features, in this case the amino acids, important for classification.

2.3.2.2 Generalized Matrix Learning Vector Quantization

Generalized Matrix Learning Vector Quantization (GMLVQ), as a type of relevance learning, has its origin in Learning Vector Quantization (LVQ) (P. Schneider et al., 2009). This chapter briefly describes the path from LVQ to GMLVQ.

Learning Vector Quantization according to Kohonen

First introduced by Kohonen (1986), LVQ is a distance- and prototype-based classification model. It is a heuristic approach which aims to spread the prototypes in a data space by minimizing datapoint misclassification of underlying training data. Each datapoint and each prototype belong to only one class.

Given is a training dataset $T=\left\{\left(\mathbf{x}_i,c(\mathbf{x}_i)\right)\in X\times C,i=1,\ldots,n_x\right\}$, where $X\subset\mathbb{R}^d$ is the set of training vectors with class labels $c\left(\mathbf{x}_i\right)\in C$ and C is the set of classes.
As first step, a set of prototypes $W=\left\{\left(\mathbf{w}_i,c(\mathbf{w}_i)\right)\in W\times C,i=1,\ldots,n_w\right\}$, where $W\subset\mathbb{R}^d$ is the set of prototypes with class labels $c(\mathbf{w}_i)\in C$, is initialized randomly. Then a labeled datapoint $\mathbf{x}_i$ is presented and its closest prototype $\mathbf{w}_{s(\mathbf{x}_i)}$ is determined by

$$s(\mathbf{x}_i)=\arg\min_j\left\|\mathbf{x}_i-\mathbf{w}_j\right\|^2 \tag{2.9}$$

with $\left\|\mathbf{x}_i-\mathbf{w}_j\right\|^2$ being the squared Euclidean distance $d(\mathbf{x}_i,\mathbf{w}_i)$ (Geweniger, 2012).

In the following update step, this winning prototype $\mathbf{w}_{s(x_i)}$ is either moved closer to or farther away from the datapoint, depending on the class label of both prototype and datapoint. If the prototype's class is the same as the chosen datapoint's, the prototype is moved closer to the datapoint. If the prototype and the datapoint differ in their class labels, the prototype is moved further away from the datapoint. According to (Kohonen, 1997), this is called the Attraction Repulsion Scheme (ARS) can be formulated for every

time step t as follows

$$\mathbf{w}_{s(x_i)}(t+1) = \mathbf{w}_{s(x_i)}(t) + \begin{cases} -\varepsilon \cdot (\mathbf{x}_i - \mathbf{w}_{s(x_i)}(t)) & if\ c(\mathbf{x}_i) = c(\mathbf{w}_{s(\mathbf{x}_i)}) \\ \varepsilon \cdot (\mathbf{x}_i - \mathbf{w}_{s(x_i)}(t)) & if\ c(\mathbf{x}_i) \neq c(\mathbf{w}_{s(\mathbf{x_i})}) \end{cases} . \tag{2.10}$$

where $0 < \varepsilon \ll 1$ is the learning rate.

The steps of choosing a random datapoint to updating all prototypes are repeated until the algorithm converges or it is manually stopped (Geweniger, 2012; Kohonen, 1997).

Generalized Learning Vector Quantization according to Sato & Yamada

A new variant of the LVQ, the Generalized LVQ (GLVQ) was introduced by Sato et al. (1996), where a differentiable cost function is introduced. It approximates the classification error, while keeping the ARS principle during learning. This cost function is as follows

$$E_{GLVQ}(X,W) = \frac{1}{2}\sum_{i=1}^{n_x} f(\mu(\mathbf{x}_i)) \tag{2.11}$$

with n_x being the number of datapoints, f a monotonous increasing function, and

$$\mu(\mathbf{x}_i) = \frac{d_{i^+} - d_{i^-}}{d_{i^+} + d_{i^-}} . \tag{2.12}$$

Here, d_{i^+} and d_{i^-} designate the distances between a datapoint $\mathbf{x}_i$ and two specific prototypes $\mathbf{w}_{i^+} \in W^+ \subset W$ and $\mathbf{w}_{i^-} \in W^- \subset W$. The prototype $\mathbf{w}_{i^+}$ denotes the winning prototype with the same class label and $\mathbf{w}_{i^-}$ the winning prototype with a different class label as the datapoint. The squared Euclidean distance can be used as distance measure for GLVQ.
After the random initialization of the prototypes $\mathbf{w}_i \in W$ with $W \subseteq \mathbb{R}^d$, the cost function is optimized with the SGD approach. Consequently, $\mathbf{w}_{s^+(x_i)}$ and $\mathbf{w}_{s^-(x_i)}$ are updated and the ARS is realized following the update rule

$$\Delta\mathbf{w}_{\mathbf{j}\pm} \propto \varepsilon \cdot \frac{f(\mu(\mathbf{x}_i))}{\partial \mathbf{w}_{\mathbf{j}\pm}} \tag{2.13}$$

for the prototypes for a randomly chosen training data point $\mathbf{x}_i$ and ε being the learning rate. The steps of choosing a random datapoint to updating all prototypes are repeated until the algorithm converges or it is manually stopped.

Generalized Matrix Learning Vector Quantization

Generalized Matrix Learning Vector Quantization (GMLVQ) is an extension of GLVQ, where a matrix is adjusted by SGD, besides the prototype adaptation. The GMLVQ cost function is as follows

$$E_{GMLVQ} = \sum_i f(\mu(\mathbf{x}_i, W, \Omega)) \tag{2.14}$$

with f an sigmoid function, $\mu(\mathbf{x})$ as in eq. (2.12) and $\Omega \in \mathbb{R}^{m \times n}$ a matrix, with which the data space is mapped to an embedded data space, before applying the squared Euclidean metric in the mapping space $\mathbb{R}^m$. The matrix Λ of size $n \times n$ is calculated by

$$\Lambda = \Omega^T \Omega \tag{2.15}$$

The Λ matrix, or *classification correlation matrix*, denotes feature correlations, which are significant for the classification (Villmann et al., 2017). Positive or negative correlation values of two features imply a positive or negative correlation important to differentiate classes, whereas the relevance values on the main diagonal provide insight into the importance of each feature on its own for the distinction of the classes (Bittrich et al., 2019). Visualizing the Λ matrix contributes to a faster perception of the classification impacting features and leads to a better model interpretability, which increases the reliability of the prediction, but alas, is not a given aspect of unsupervised ML method (Lisboa et al., 2021). This interpretability is needed to draw conclusions about the biological data.

2.4 Classification Validation

In the case of classification tasks, there exist several measures for evaluating the classification of the model. The classification accuracy (Acc) is such a measure, that denotes the ratio between correctly classified datapoints to the total number of datapoints. It is therefore the probability of the model prediction being correct (Powers, 2020; Grandini et al., 2020).
The accuracy and further classification validation measures (or confusion metrics) can be calculated with the values of a confusion matrix, as seen in Table 1.

Table 1: Confusion matrix of binary classification problem

		Predicted Class	
	Classes	**Positives**	**Negatives**
Actual Class	**Positives**	True Positives	False Negatives
	Negatives	False Positives	True Negatives

The confusion matrix for multi-class classification problems can be seen in Table 2.

Table 2: Confusion matrix of multiple classification problem with class 2 as reference with its True Positive highlighted in green. The False Positives are in the same column, whereas False Negatives are located in the same row as the True Positive. All other cells contain the True Negatives.

		Predicted Class			
	Classes	**1**	**2**	**3**	**4**
Actual Class	**1**	True Negatives	False Positives	True Negatives	True Negatives
	2	False Negatives	True Positives	False Negatives	False Negatives
	3	True Negatives	False Positives	True Negatives	True Negatives
	4	True Negatives	False Positives	True Negatives	True Negatives

with class 2 as reference (Grandini et al., 2020).

These confusion matrices include the occurrences of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN), with TP and TN the two types of correctly classified datapoints, and FP and FN the two types of wrongly classified datapoints (Powers, 2020).

$$\text{TP} = \sum_{x \in X} \mathbf{x_i} \text{ with class } c(\mathbf{x}_i) \text{ correctly classified into class } c(\mathbf{x}_i) \tag{2.16}$$

$$\text{TN} = \sum_{x \in X} \mathbf{x_j} \text{ with class } c(\mathbf{x}_j) \text{ correctly classified into class } c(\mathbf{x}_j) \tag{2.17}$$

$$\text{FP} = \sum_{x \in X} \mathbf{x_j} \text{ with class } c(\mathbf{x}_j) \text{ incorrectly classified into class } c(\mathbf{x}_i) \tag{2.18}$$

$$\text{FN} = \sum_{x \in X} \mathbf{x_i} \text{ with class } c(\mathbf{x}_i) \text{ incorrectly classified into class } c(\mathbf{x}_j)\ . \tag{2.19}$$

Accordingly, the accuracy is calculated as follows:

$$Acc = \frac{\text{number of correct predictions}}{\text{total number of predictions}} = \frac{TP+TN}{TP+TN+FP+FN} \tag{2.20}$$

For imbalanced data, one possible classification validation measure is the *balanced accuracy*. It is more suitable, since otherwise classification errors may occur for classes with fewer datapoints, as these are less relevant compared to classes with numerous datapoints (Grandini et al., 2020).

$$Acc_{\text{balanced}} = \frac{\frac{TP_{c_1}}{\sum c_1} + \frac{TP_{c_2}}{\sum c_2} + \ldots + \frac{TP_{c_n}}{\sum c_n}}{|C|} \tag{2.21}$$

with $|C|$ for the number of classes and $\sum c_i$ for all datapoints of class i.

To evaluate the validity of the model further, additional confusion metrics beside the accuracy can be calculated. The following work will focus on Precision, Sensitivity, Specificity and Matthews Correlation Coefficient (MCC). Precision expresses the amount of datapoints of class $c(\mathbf{x}_i)$ correctly classified as true positives. It is therefore an indication of the degree to which the model can be trusted when it predicts a datapoint to be positive (Grandini et al., 2020).

$$\text{Precision} = \frac{TP}{TP+FP} \tag{2.22}$$

Sensibility measures the probability that a datapoint of class $c(\mathbf{x}_i)$ will be correctly classified as class $c(\mathbf{x}_i)$ (Positive) and therefore captures the prediction accuracy of the model for a positive class (Grandini et al., 2020).

$$\text{Sensitivity} = \frac{TP}{TP+FN} \tag{2.23}$$

In contrast, specificity is the probability of a datapoint of class $c(\mathbf{x}_j)$ being correctly classified as class $c(\mathbf{x}_j)$ (Negative) (Swift et al., 2019).

$$\text{Specificity} = \frac{TN}{TN+FP} \tag{2.24}$$

Calculations of the above mentioned confusion metrics are adapted for multi-class classification to each class accordingly (Grandini et al., 2020).

Moreover, the Matthews Correlation Coefficient can be used in case of imbalanced data (Chicco et al., 2020). It is calculated as follows

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{2.25}$$

with MCC $\in [-1, 1]$. MCC values of 1 indicate perfect classification, whereas values of -1 indicate perfect misclassification. The value 0 denotes coincident classification (Chicco et al., 2020).

3 Data Acquisition

The dataset consists of neuraminidase sequences, which were acquired through the Protein Data Bank (PDB) and through the National Center for Biotechnology Information (NCBI) Influenza Virus Database. After collecting and preprocessing both subdatasets and balancing the subdataset acquired through the NCBI, they are merged into one dataset and identical sequences are removed.
The dataset used for this analysis finally contains 1506 sequences (60 from the PDB and 1446 from the NCBI) ranging from the years 1918 to 2019 and which are isolated from eight different taxa of seven different continents. The neuraminidase proteins were extracted from 89 different influenza subtypes, of which six have an unknown hemagglutinin subtype. In this chapter, the data acquisition, the data balancing as well as the data filtering will be specified.

3.1 Data acquisition from the PDB

The sequences from the PDB are defined as amino acid sequences extracted directly from the .pdb-files instead of selecting the annotated sequences. Since .pdb-files contain the coordinates or location of each atom of a protein sequence in a 3D space including the respective amino acid, this procedure has the advantage of getting the exact amino acids from the experimentally resolved protein structure. These can deviate from the annotated amino acids both in length and in individual amino acids since other experimental methods may have been used for amino acid identification. Thus, only the domain relevant to this analysis is considered. The PDB sequences have a median length of 389 amino acids.
To acquire all NA data from the PDB, the PDB identifiers (IDs) are obtained via the Protein Families Database (PFAM). Under the Protein Family *Glycoside hydrolase family 34* (PF00064[2]), the PDB identifiers for almost every structurally resolved neuraminidase protein are listed and can thus be downloaded from PDB[3]. Only the structures for N10 and N11 have not been added at the time to the PFAM, and due to insufficient data concerning N10 and N11, those NA subtypes are not included in the dataset. The low quantity of sequences may be explained by the recent discovery of the organism of isolation. So far, N10 and N11, isolated from H17N10 and H18N11, have only been detected in New World bats of South America. Furthermore, H17N10 and H18N11 have not yet been able to propagate in cell cultures, which hinders further research (Q. Li et al., 2012; Wu et al., 2014).
In total, 228 PDB IDs are acquired as of 5 May 2021. Identifiers and protein chains not belonging to the influenza A neuraminidase head domain are discarded, e.g. influenza B

[2] https://pfam.xfam.org/family/PF00064
[3] https://www.rcsb.org/downloads

NA or ligands and antibodies bound to NA. After this preprocessing, 184 structures remain in the dataset, from which the sequences are extracted as described.
The 184 sequences are from 23 influenza A subtypes (including five with unknown hemagglutinin subtypes) from 1918 to 2015. These neuraminidase proteins were isolated from 12 distinct hosts of human, avian, porcine, canine, phocid and cetacean origin from five different continents (see Table A.23 in Appendix). Of this initial dataset, a total of 58 sequences belong to neuraminidase group 1 and 126 to group 2. The great number of group 2 data is striking. This group consists, among others, of the neuraminidase subtypes N2 and N9, which were isolated from influenza A subtypes responsible for some human and avian pandemics of the 20th and 21st century (Kilbourne, 2006; Morens et al., 2009; Al Hajjar et al., 2010; Flaherty, 2012). As observed in Table A.23, most N9 data is from 1975. It is to be assumed that in the early 1970's new avian influenza A strains were discovered, therefore being a surplus of isolated neuraminidase from Australia's tern population of that time (Webster, Isachenko, et al., 1974). The oldest sequences of the dataset date back to 1918 and come from the influenza A subtype H1N1. In 2009, H1N1 reemerged as the strain H1N1pdm09, which was responsible for the swine flu pandemic of that year (Taubenberger and Morens, 2010). In total, there are 24 H1N1 sequences in the PDB dataset.
Furthermore, there are five sequences to which the influenza A subtype could not be identified and 14 sequences where only the neuraminidase subtype is known but not the hemagglutinin subtype. Therefore, there is no information on the place of origin, but through literature research the hosts could mostly be identified.
Of the initial PDB dataset containing 184 sequences, sequences with a sequence identity of 100% were removed, leaving 62 sequences. This subset of sequences will be referred to as the **PDB sequences**.

3.2 Data acquisition from the NCBI

To obtain a dataset, which can be used for scientific purposes, additional neuraminidase sequences from the NCBI Influenza Virus Database were downloaded on 20 April 2021. As keyword *influenza* was chosen as well as Type *A*, *any* host, *any* country/region, protein *NA*, with the influenza subtype *any* H and subtype *any* N. As collection date, all sequences from 01 January 1918 to 31 December 2020 were chosen. This query led to 96.897 sequences.
From these sequences, those with unknown or ambiguous amino acids, sequence length of under 100 amino acids or a wrong subtype were discarded, leaving 91.758 sequences. Subsequently, after removing all identical sequences, 38.215 sequences remained, referred to as the **NCBI sequences**. These sequences have a median length of 469 amino acids. This length indicates that the NA sequences are mostly whole protein sequences.
The NCBI sequences are from 129 influenza subtypes. Of those 38.215 sequences, 11.180 sequences are from the influenza subtypes H1N1 and 10.590 from H3N2. All

other influenza subtypes have less than 3.000 sequences (see Table A.24). Consequently, the NA subtypes N1 and N2 are the subtypes with the most sequences (N1: 14784 sequences, N2: 17976 sequences), while all other NA subtypes contain less than 1500 sequences (see Table A.25). Furthermore, the NA proteins were isolated from a wide range of hosts of mammalian and avian origin from six different continents, with most sequences originating in Asia and North America. Aside from the sequences with additional information to their influenza or neuraminidase subtypes, the dataset also contains seven sequences with unknown influenza subtype and nine influenza subtypes with unknown hemagglutinin subtype. No specific host could be identified in 21 cases and no specific continent in eight cases. The isolation years range from 1918 to 2020, with most sequences coming from 2009 when H1N1 was pandemic.

3.3 Dataset Balancing

As neuraminidase subtypes N1 and N2 are overrepresented in the NCBI sequences, the dataset needed to be balanced. The NCBI sequences were split into nine subdatasets according to their neuraminidase subtypes. The balancing was done as described in Chapter 2.3.1 after converting the amino acid sequences into natural vectors as specified in Chapter 2.2.
Neural Gas was applied to every NA subtype subdataset with prototype initialization $k = 200$ and neighborhood range $\lambda = 20$. For further parameter settings, see Table A.26. After running Neural Gas, the nearest datapoint of a prototype was determined via Euclidean distance with the aim of providing a dataset of 200 datapoints for each NA subtype. It became apparent when reviewing the resulting data that some datapoints were assigned multiple prototypes. For example, the datapoint *AAA43363* is simultaneously the nearest datapoint to prototype *Pt_186* and to prototype *Pt_102*. To avoid duplications, all datapoints that occur twice or more have been reduced so that they are only present once in the dataset. Hereby, each NA subtype had between 124 and 200 entries (see table 3), which was considered a balanced dataset.

Table 3: Overview of number of sequences for each NA subtype in dataset NCBI sequences after balancing.

NA subtype	N1	N2	N3	N4	N5	N6	N7	N8	N9
number of sequences	199	200	170	148	124	167	159	176	143

Finally, the balanced NCBI sequence dataset and the PDB sequence dataset were merged together. To avoid identical entries/sequences, redundant sequences were discarded, resulting in a working dataset of 1506 sequences.

3.4 Redundancy Filtering

The majority of the NCBI sequences is much longer than the PDB sequences, due to them being sequences of the whole NA protein, whereas the PDB sequences are mostly the NA head domain. A multiple sequence alignment therefore was carried out for two reasons: first, to identify the NA head in the NCBI sequences (described in detail in Ch. 5) and secondly to determine the sequence identity and similarity of the sequences. For those reasons, a structural alignment of the PDB sequences was conducted using *Tree-based consistency objective function for alignment evaluation* (T-Coffee) Expresso (Notredame et al., 2000). With this specific type of alignment, the amino acid sequences are aligned using information about their secondary structure. Two representative neuraminidases were selected for the secondary structure information, each of which can be assigned to one of two NA group. These are listed in Table 4.

Table 4: Secondary structure representatives of each subdataset for the structural alignment with T-Coffee Expresso.

Dataset	NA group	NA subtype	PDB-ID	Year of Isolation
PDB sequences	group 1	N1	3NSS	2009
	group 2	N9	2B8H	1984

Hereinafter, the NCBI sequences were aligned to the PDB sequence alignment. This multiple sequence alignment was accomplished using the profile alignment method of the Clustal software version *ClustalX* (Larkin et al., 2007). *ClustalX* has a graphical user interface, where the structurally aligned PDB sequences were uploaded as the first profile and the unaligned NCBI sequences as the second profile. A profile alignment was then conducted. This MSA of both sequence datasets was used to discard of any sequence sections not belonging to the NA head domain. To determine where this domain starts, the PDB sequences were used as reference and the NCBI sequences were truncated at this position.
Sequence identity and similarity was subsequently calculated using the web based application *Sequence Identity And Similarity* (SIAS), (Reche, 2021)). The percentage values of identity and similarity, as measures of dis-/similarities, can be used to determine to which degree sequences might be related, and especially the sequence identity can be used for protein classification or modeling (May, 2004). The percentage value of sequence identity or sequence similarity is calculated as follows

$$ID|SIM_{\%} = 100 * \frac{Identical \mid SimilarResidues}{Sequence\ Length} \tag{3.1}$$

where identical residues are the number of exactly matching residues, and similar residues are the number of residues matching considering their physio-chemical properties (Reche, 2021). For the latter, those were customized to aromatic (F, Y, W, H), aliphatic (V, I, L),

Table 5: Overview of NA sequences per group and subtype in the dataset

NA groups	group 1				group 2				
NA subtypes	N1	N4	N5	N8	N2	N3	N6	N7	N9
number of sequences	211	112	123	185	213	172	169	160	161
in total	631				875				

charged positive (R, K), charged negative (D, E), small (S, T, A, G), polar (N, Q) and hydrophobic (V, I, L, M, F, W) after Betts et al. (2003). Furthermore, the sequence length is selectable. In this analysis, as sequence length the length of the multiple sequence alignment was chosen, considering that gaps and the number thereof play an important role in multiple sequence alignments (May, 2004). Additionally to the identity and similarity, SIAS calculates a normalized similarity score S for every pair of aligned sequences that penalizes the presence of gaps as follows

$$S = \frac{(\sum M_{ij}) + oP_o + eP_e}{\sum M_{ii}} \tag{3.2}$$

where M_{ij} are the similarity scores obtained from the BLOSUM62 substitution matrix for the amino acids i and j, o is the number of gaps, e the total extension of the gaps. Likewise, P_o is the penalty for opening a gap (set at 10) and P_e the penalty for extending a gap (set at 0.5).

3.5 Overview of Working Dataset

The dataset consists of neuraminidase sequences, which were acquired through the PDB and through the NCBI Influenza Virus Database. After collecting and preprocessing both subdatasets and balancing the subdataset acquired through the NCBI, they were merged into one dataset and identical sequences are removed.
The dataset used for this analysis finally contains 1506 sequences (60 from the PDB and 1446 from the NCBI) ranging from the years 1918 to 2019 and which are isolated from eight different taxa of seven different continents. The neuraminidase proteins were extracted from 89 different influenza subtypes, of which six have an unknown hemagglutinin subtype. The following chapter gives an overview over the neuraminidase dataset which is used in further analyses.
Overall, there are 631 NA sequences belonging to neuraminidase group 1 and 875 NA sequences in group 2. Due to IAV subtypes that have led to human epidemics in recent decades or are endemic in the human population, more data are available in group 2. These IAV subtypes include, for instance, H2N2 (1957, Asian Flu), H3N2 (1968, Hong Kong Flu) or H7N9 (2013). The dataset, balanced in regard of neuraminidase subtype, has between 112–213 sequences in the respective subtypes. Table 5 gives an overview of the number of sequences in NA subtype per group.

Moreover, IAV has been isolated on every continent. In Figure 8, the amount of sequences per continent are illustrated, with the amount of sequences from Africa colored in pink, those from Antarctica in orange, from Asia in light blue, from Australia in green, from Europe in yellow, from North America in dark blue and those from South America in red. North America is the continent with most sequences with 613 sequences. 511 sequences belong to Asia and 281 to Europe.

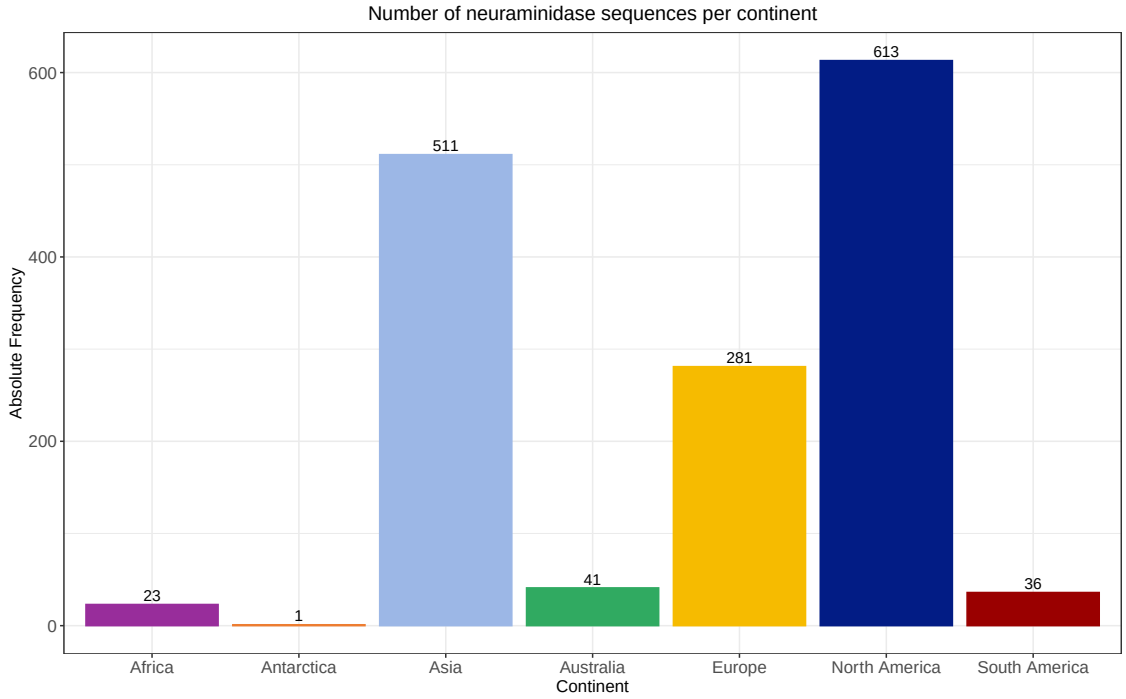


Figure 8: Overview of NA sequences per continent in the dataset

The predominant hosts of IAV are avian hosts, which is mirrored in the amount of avian data in the dataset. In total, 1140 avian NA sequences, 202 human NA sequences and 93 porcine NA sequences are represented in the dataset. All other hosts have less than 40 sequences, with only 2 neuraminidases extracted from musteline hosts (see Fig. 9). Among avian hosts are different species of poultry (e.g. chickens) and wild bird such as waterfowl (e.g ducks, mallards). Albeit cetaceans include all whales, such as humpback whales or sperm whales, and canines include dogs, wolves etc., no exact species information were available in the case of these two hosts. Neuraminidase sequences extracted from IAV collected from the *environment* was included in the dataset as a host, as the samples are from host excretions or from aquatic environment. As no specific host or species could be identified, the sequences remained labeled as *environment*. For NA sequences labeled as coming from equine host, no species was specified, but it is assumed these sequences were isolated from horses, as multiple epidemics among working horses or racehorses (in 1979 and 2003) are documented in literature (Newton et al., 2006). For the musteline host, IAV was isolated from the species ferret and weasel, and for phocids, the virus was sampled from seals, like the harbor seal. Lastly,

the porcine NA were collected from swine and wild boar.

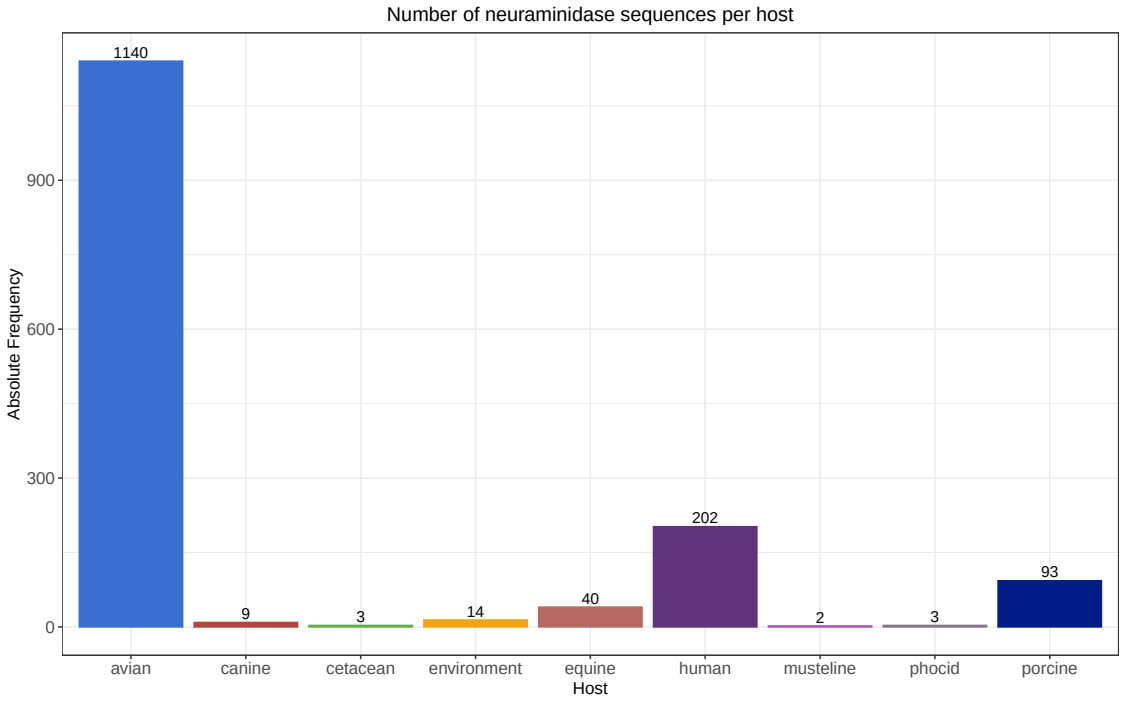


Figure 9: Overview of NA sequences per host in the dataset

4 EVcouplings Analysis

The EVcouplings pipeline, as described in chapter 2.1.2, was used as command-line application as well as online on the EVcouplings server. Furthermore, *plmc* was executed individually, with the goal of identifying evolutionary couplings.
In the following chapter, the execution and the results[4] of the evolutionary couplings analysis are discussed in more detail. All parameter settings can be found in Appendix A.4.

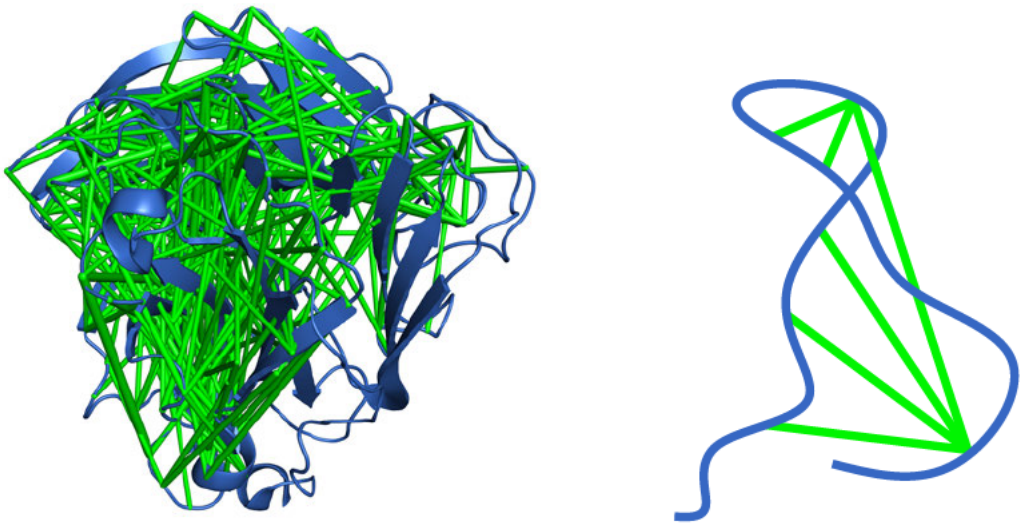
4.1 EVcouplings as command-line application

For using EVcouplings as command-line application, EVcouplings and additional external software tools were downloaded following the instructions in the EVcouplings GitHub repository. Next, the configuration file was adapted to fit the requirements for this analysis. The configuration and additional information regarding the device used for the EVcouplings analysis can be found in the Appendix A.3.
The existing MSA (described in Ch. 3.4) was provided as input, the NA sequence with the PDB-ID 6D96 was set as target sequence, and all pipeline stages not required for the identification of evolutionary couplings were disabled. Those stages included the compare, mutate and fold stage. Running EVcouplings then took 184.1 seconds.
In total, 28 files in two folders were generated as output. The first folder contains the results from the align stage, including MSA statistics, sequence identities and position frequencies.
The *statistics-file* contains information on the alignment properties, such as the minimum column coverage (set at 0.7), the number of sequences in the alignment (1506 sequences), the length of the target sequence 6D96 (387 amino acids), the number of uppercase columns in the final alignment (382 columns) or percentage of coverage (98.7%). The latter is calculated by dividing the number of uppercase columns in the alignment by the length of the target sequence.
In the *sequence-identities-file* all sequence identities of the MSA sequences to the target sequence can be found. For example, all sequences with the same NA subtype N1 as 6D96 have a sequence identity of approx. $70\% - 95\%$ in contrast to less than 50% sequence identity for NA subtypes of NA group 2 e.g. N2.
The *frequencies-file* contains the frequency of each amino acid character at each alignment position according to the target sequence. The column *conservation* indicates the conservation of each amino acid in 6D96 with a value of 1 indicating perfect conservation and a value of 0 very little to no conservation at that specific sequence position. Only two amino acids reach perfect conservation: Threonine (T) at target sequence

[4] All documents mentioned are available at https://github.com/Lreuss/NA_MA after being granted access by the author.

position 144 and Cysteine (C) at target sequence position 210 are shown to have a conservation value of 1.

The coupling files are located in the second folder, including tables with couplings scores and PyMol scripts for the visualization of the ECs in PyMol. The raw results from *plmc's* EC calculations contain APC-corrected Frobenius norm score values quantifying the derived strength of the coupling between each pair of amino acid positions, therefore called *coupling scores*. Those scores are processed by EVcouplings and are output as *coupling-score-file*, which contains additional information such as the probability of a coupling being significant. When reviewing this file, the most obvious result is significant coupling probability of 0 for all ECs. Since this probability is 0%, it can be presumed that the calculation of evolutionary couplings was not successful. Furthermore, the PyMol scripts mentioned above were run to visualize the ECs. A total of 382 ECs were visualized in PyMol, as shown in Figure 10. The green ECs seem to connect residue pairs throughout the whole structure of the neuraminidase head domain, here shown in blue. With a sequence length of 387 amino acids, a total of 382 ECs is very surprising as so many were not expected. It should be noted that multiple couplings could be observed originating from one to several other residues. While biologically possible, the high number of occurrences of this event is surprising and, when considering the significant coupling probabilities, this seems error-prone. These unexpected results led to additional analyses via the EVcouplings server and *plmc*.



(a) Visualization of all evolutionary couplings in neuraminidase N1 (PDB ID 6D96).

(b) Schematic representation of multiple couplings originating from one residue to several other residues.

Figure 10: Visualization of evolutionary couplings. The protein is shown in blue and the couplings between two amino acids in green. (a) All EC in neuraminidase N1 (PDB ID 6D96), generated by EVcouplings. (b) Schematic representation of multiple couplings originating from one residue to several other residues. The protein is shown in blue and the couplings between two amino acids in green.

4.2 EVcouplings Website

EVcouplings analysis was also performed via the EVC website. For this purpose, no predefined alignment could be entered, instead only a UniProt-ID or a protein sequence in *fasta*-format could be specified. For the sake of comparability, the sequence of 6D96 was chosen and EVC workflow was then performed on this sequence. The set default parameters are mentioned in Table A.28. The resulting files are the same as with the command-line application, but with additional visualizations. Due to the calculations being executed for every bitscore and throughout the whole pipeline, the calculation time amounted to 36 hours. The bitscore (Fig. 11 (1)) denotes deeper to shallower evolutionary depth. The resulting outcome is assessed by EVcouplings (see Fig. 11) as a quality score from $0-10$ with values $0-3$ demonstrating low quality results, $4-7$ medium quality results and $8-10$ high quality results. As illustrated, none of the results are satisfying, as the result quality is of 0 for every bitscore (Fig. 11 (5)). Moreover, the alignment does not seem to cover the whole target sequence, as seen by the deep blue bars in contrast to the light blue bar (Fig. 11 (2)). As no alignment could be input, EVC generated an alignment. The number of sequences used can be seen in Fig. 11 (3). These alignments are of low quantity compared to the neuraminidase dataset of 1506 sequences analyzed in this thesis. It is doubtful, that changes in the neuraminidase sequence over an extended period of time, such as 100 years, are computed reliably with less than 40 sequences.

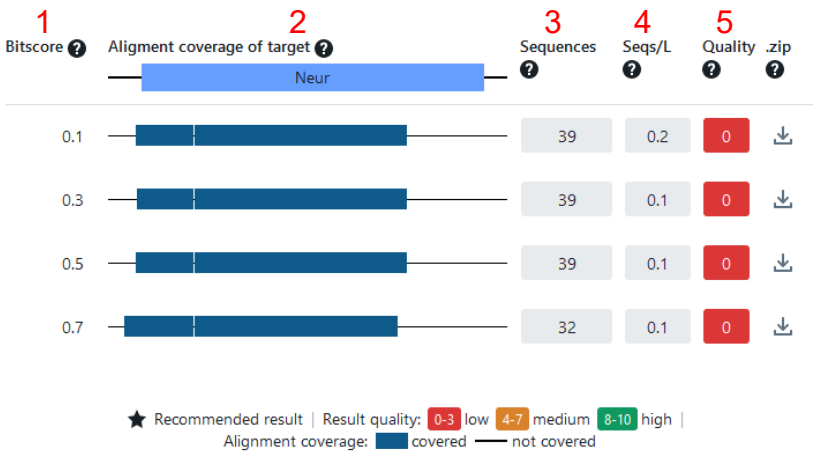


Figure 11: Overview of results from EVcouplings Website with neuraminidase N1

Changing the target sequence or parameters did not produce better results for both EVcouplings analysis methods, and thus will not be discussed further. To ensure no mistakes were made in the handling and parameter setting of EVcouplings, the program was run with the protein ribonuclease A (RNase A, Uniprot ID: Q9BEC3). RNase A has a sequence length of 142 amino acids and is thus significantly shorter than the NA sequences. This did reduce the execution time and produce better results, as the quality

score for bitscore 0.1 was 10, with 22.016 sequences in the alignment. Furthermore, the probabilities for significant couplings were not 0, but contained much more reliable values. These will not be discussed further, but it is assumed, that EVcouplings is more suitable for shorter protein sequences.

4.3 *Plmc* Analysis

Plmc was executed individually and the results were visualized in MATLAB[5]. The first run was executed with the following parameters: strong L_2 regularization for the ECs $\lambda_e = 16.0$, weak L_2 regularization for the fields $\lambda_n = 0.01$ and maximum number of iterations `-m` was set to 100. Furthermore, the target sequence (here `focus` or `-f`) was set to 6D96 and the option `gapignore` (`-g`) was activated. These parameters were selected on the basis of the specified parameters for an exemplary protein in the *plmc* GitHub repository. The parameters were adapted for further runs, as seen in table 7 below.

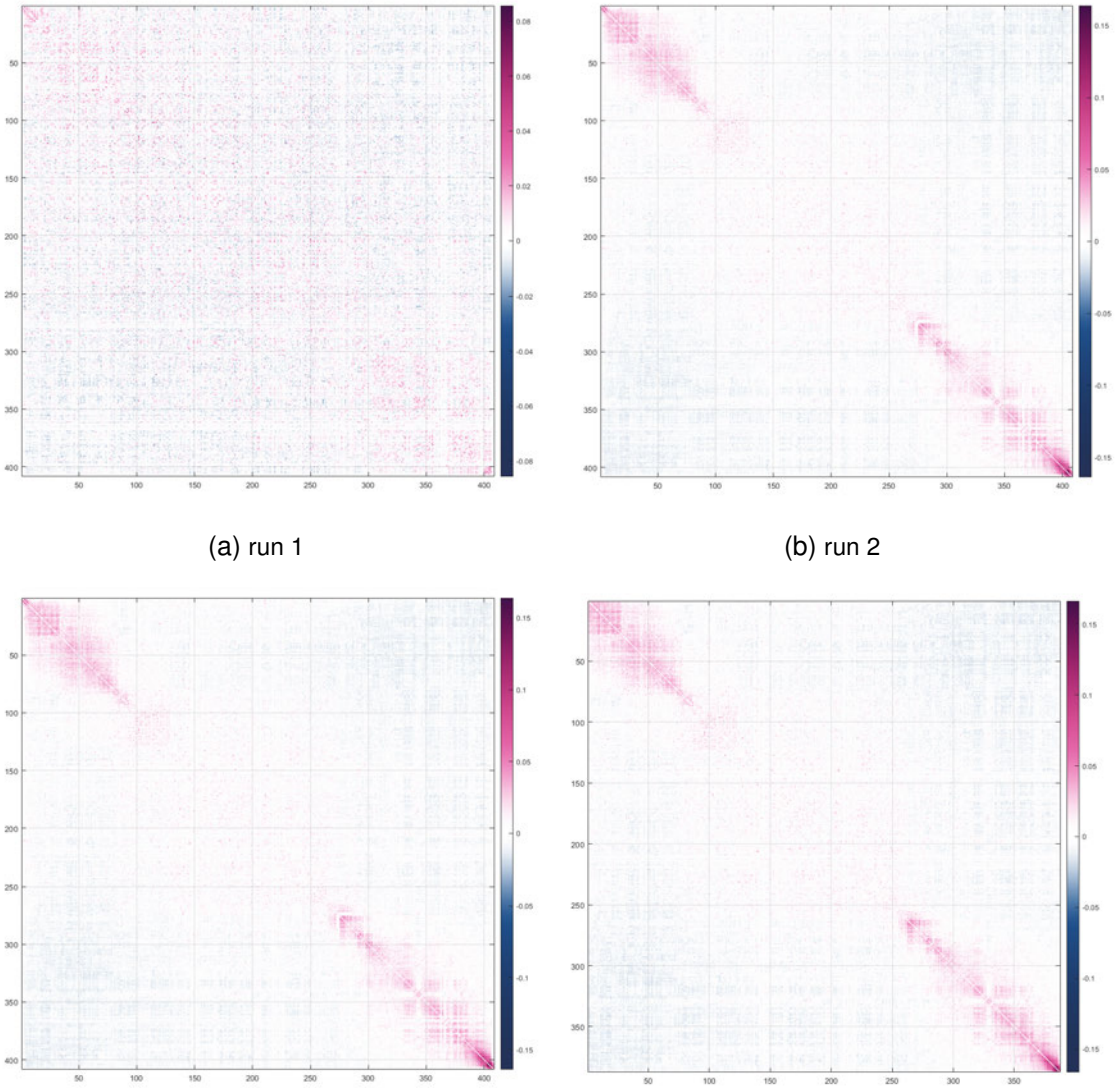
Table 7: *Plmc* parameter settings

		plmc runs							
		run 1	run 2	run3	run 4	run 5	run 6	run 7 group 1	run 7 group 2
parameter settings	λ_e	16.0	16.0	16.0	16.0	32.0	32.0	16.0	16.0
	λ_n	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	-m	100	-	-	-	-	-	-	-
	-f	6D96	-	6D96	-	6D96	-	-	-
	-g								
runtime (in sec)		778.2	1870.9	1736.0	1453.2	2016.7	2203.8	1398.8	1628.2

It was analyzed how the parameters `-m`, `-f`, and `-g` affect the results (run 2 - run 4) and these were visualized. *Plmc* offers scripts for visualization of the coupling values in MATLAB. The generated Figures 12-14 are representations of the coupling strengths between the positions in an alignment or target sequence, with negative coupling strength values in deep blue up to positive coupling strength values in dark pink. The diagonal line running from upper left to lower right corner is similar to the diagonal in contact maps, which portrays the backbone of the protein. No coupling strength value is calculated for a residue to itself and is therefore of value 0. The upper and lower triangular matrices are symmetrical. Figure 12 shows the plotted coupling strengths, which should visualize the strong-coupled pairs/evolutionary couplings in a pink shade. As observable, run 1 appears to consist only of random noise. No significant EC can be detected, as most probably a premature termination of the EC calculations led to these

[5] MATLAB-scripts were provided by *plmc*

results. Therefore, the maximum iterations `-m` were no longer specified. Then, the impact of specifying a target sequence or activating the `gapignore` option was observed. Run 2 and run 3 are almost identical. It can be deduced, that specifying an aligned target sequence or setting the `gapignore` option for this dataset seem to have a similar impact on the calculated couplings. For the fourth run, no parameter, except from λ_e and λ_n were specified. Even though the visualized couplings plot looks slightly different, the couplings strength values lie between 0.135 and -0.021 with a median coupling strength at -0.001, and hence, the values are the same as for run 2 and run 3. From the output couplings file, it could be observed that almost every strong coupling score calculated was between adjacent positions in the target sequence. This means, that amino acids in direct sequential neighborhood share a strong bond. While this is certainly true (Bruice, 2004), it was not an expected outcome and for further research, this information is obsolete.



(a) run 1

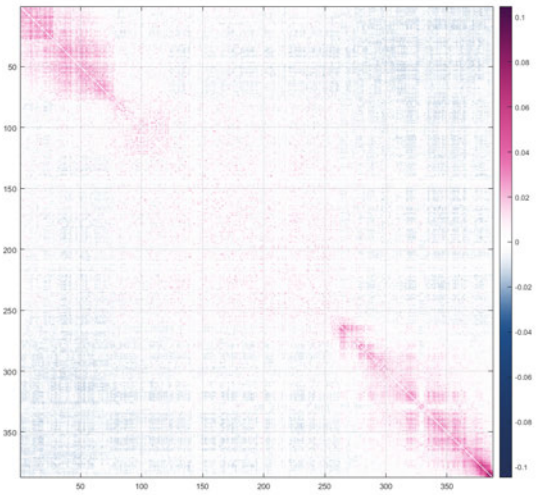
(b) run 2

(c) run 3

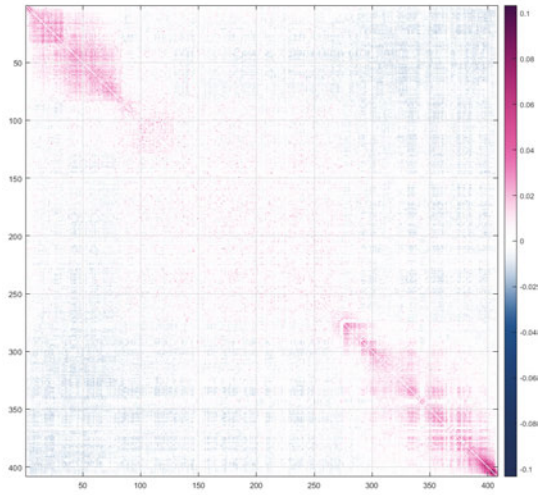
(d) run 4

Figure 12: *Plmc* results from run 1–4

Next, the effect of the strong L_2 regularization for the ECs λ_e on the results was tested. Therefore, λ_e was set to 32.0 and two runs were executed, run 5 with a target sequence (6D96) and run 6 without. No maximum iterations were set due to the aforementioned reasons, neither was the `gapignore` option set. The results are visualized in figure 13.



(a) run 5

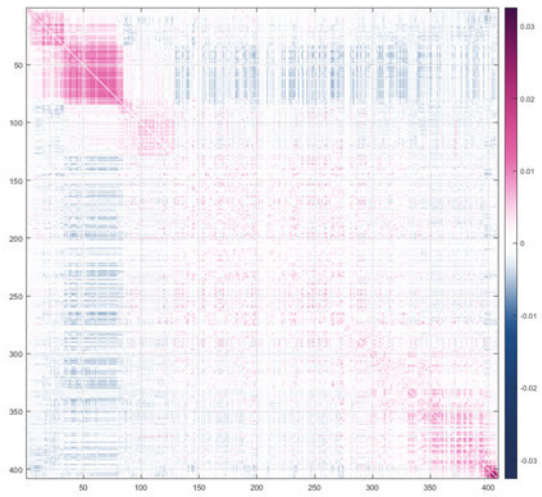


(b) run 6

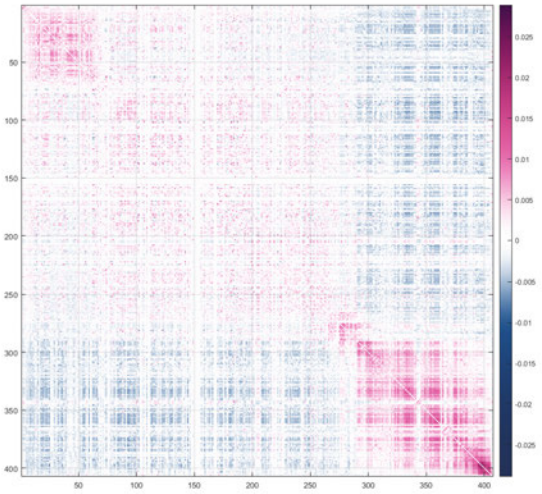
Figure 13: *Plmc* results run 5 and run 6

Increasing λ_e did not improve the results. As already noted for run 2-4, the highest coupling scores are between neighboring amino acids. For both runs, run 5 and 6, the coupling score values lie between 0.082 and -0.017 with a median of -0.001.
As a last attempt, the dataset was separated according to the neuraminidase groups, with the same parameter settings as run 4. Although this has also not led to the desired results, it became apparent that the couplings of group 1 and group 2 seem to differ. While group 1 seems to have strong coupled pairs of amino acids at the beginning of the protein sequence, group 2 seems to have those at the end of the protein sequence

(see Fig. 14).



(a) run 7: NA group 1



(b) run 7: NA group 2

Figure 14: *Plmc* results run 7

This leads to the assumptions, that neuraminidase is probably either too long for an analysis with *plmc* or EVcouplings, or that the parameters need to be fine-tuned for interpretable results. Moreover, it can be deduced from Fig. 14 that the neuraminidase could eventually be classified by NA group due to the nature of the different couplings positions, even though the results from *plmc* cannot be used in further investigations.

5 Natural Vectors for Sequence Embedding

After the unsuccessful analysis using EVcouplings, the next step is the preparation to vectorize the sequences for subsequent machine learning methods, because especially distance-based machine learning methods require a transformation into vectors, more precisely into numerical data (Bohnsack et al., 2022). This chapter is devoted to the feature generation with the NV method, which encodes the neuraminidase sequences into 60-dimensional vectors as described in chapter 2.2. Thereafter, the vectors were subjected to Principle Components Analysis (PCA) and the first principal components (PC) are used to visualize the protein sequences in 2D space.

5.1 Application of the Natural Vector Method

Based on the visualization of sequence embedding from Blackshields et al. (2010), an embedding of 1506 neuraminidase sequences was performed.
After generating natural vectors, PCA was performed and the first two principal components were visualized. To get a general overview of the data, the datapoints were colored by the years of isolation of their respective protein sequence (see figure 15). Since these cover a time period of 100 years, the years are defined as intervals ranging from 1918–1959, from 1960–1999, from 2000–2005, from 2006–2010, from 2011–2015 and from 2015–2019, with most IAV neuramindases being isolated between 2006 and 2010 due to the H1N1pdm09 pandemic of 2009.

The datapoints in Figure 15 are colored from the oldest sequences (1918) in blue, changing progressively to red for the most recent sequences from 2019. Alas, the visualization as seen in Blackshields et al. (2010) does not seem to be reproducible with neuraminidase natural vectors. The reasons for this are manifold. Firstly, Blackshields et al. were looking at hemagglutinin subtype H3, instead of all neuraminidase subtypes. Secondly, they examined roughly 2.5 times more sequences from a smaller time range. Lastly, the probably most striking contrast in the workflow is due to the different vectors that were used. Blackshields et al. generated 121 dimensional vectors with *mBed* in contrast to 20 dimensional neuraminidase natural vectors.

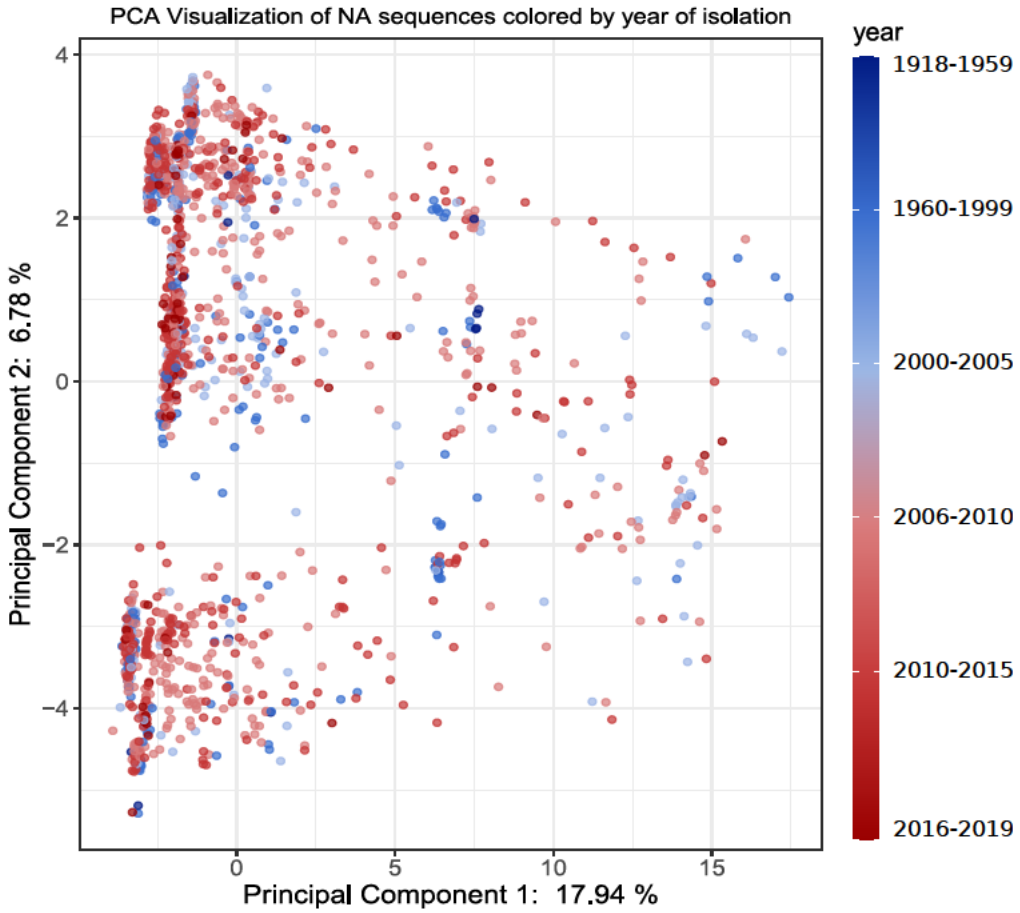


Figure 15: PCA Visualization of NA sequences colored by year of isolation from 1918 in blue to 2019 in red. The amino acid sequences were processed according to the natural vector method by Y. Wang et al. (2019). The figure represents the two principal components of the principal component analysis.

Additionally, the datapoints were colored by other labels, such as *subdataset*, *NA groups*, *NA subtypes*, *hosts* or *continents*. After coloring the datapoints by subdataset (see Fig. 16), the NCBI sequences in blue and the PDB sequences in black, it becomes evident, that the majority of NCBI sequences builds two point clouds (left-hand side in Fig. 16) whereas the PDB sequences located separately from them in the middle of the Figure. To determine why the majority of the NCBI sequences differs from the PDB sequences, the datapoints were colored according to sequence length. This revealed that the PDB sequences are shorter than the NCBI sequences, thus leading to the conclusion that the length of a sequence seems crucial for natural vectors (Deng et al., 2011).

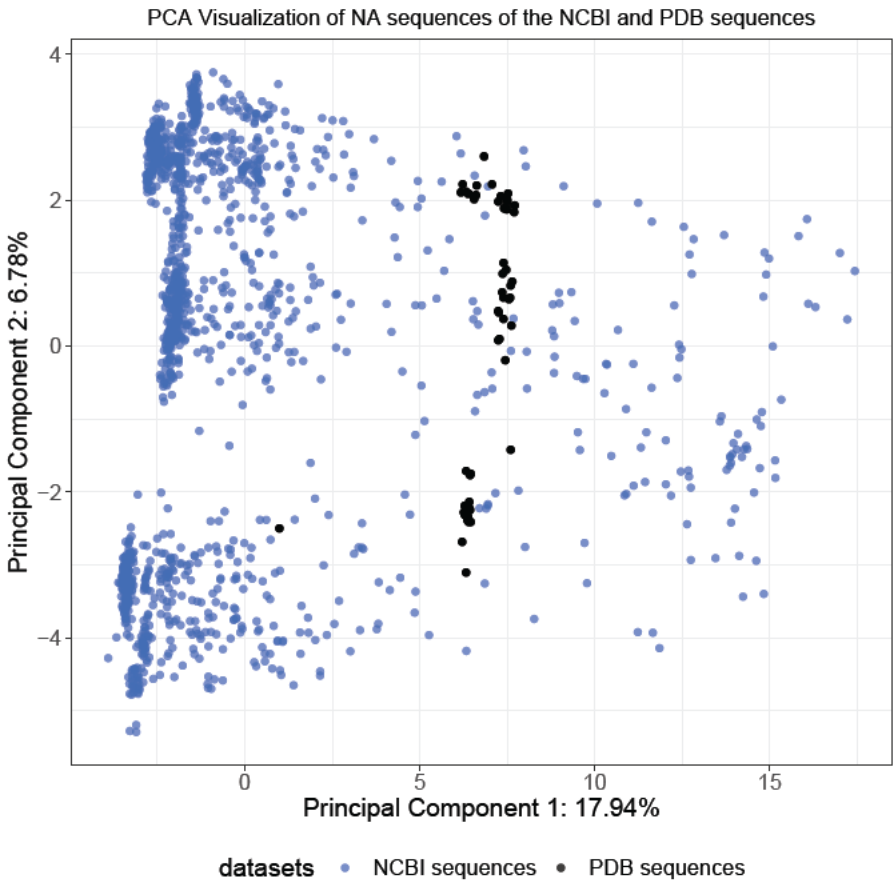


Figure 16: Visualization of 1506 neuraminidase natural vectors colored by subdataset: natural vectors of NCBI sequences are shown in blue and natural vectors of PDB sequences are shown in black. The amino acid sequences were processed according to the natural vector method by Y. Wang et al. (2019) - the figure represents the two principal components of the principal component analysis.

In Fig. 17, the sequences are colored by their sequence length, ranging from red for shorter (300 residues) to blue for longer sequences (475 residues). The majority of sequences, as seen in this figure, have a sequence length between 435 and 475 amino acids. This corresponds with the length of whole neuraminidase proteins, which is approximately 469 amino acids. On one hand, this is dependent of the NA subtype. As mentioned, the stalk region can undergo a deletion of 20 amino acids as adaption to the host to be infected (see Chapter 1.2). On the other hand, sequence length depends on the focus of the researcher doing the sequencing and the respective research task. The Figures 16 and 17 show that the majority of the NCBI sequences is much longer than the PDB sequences.

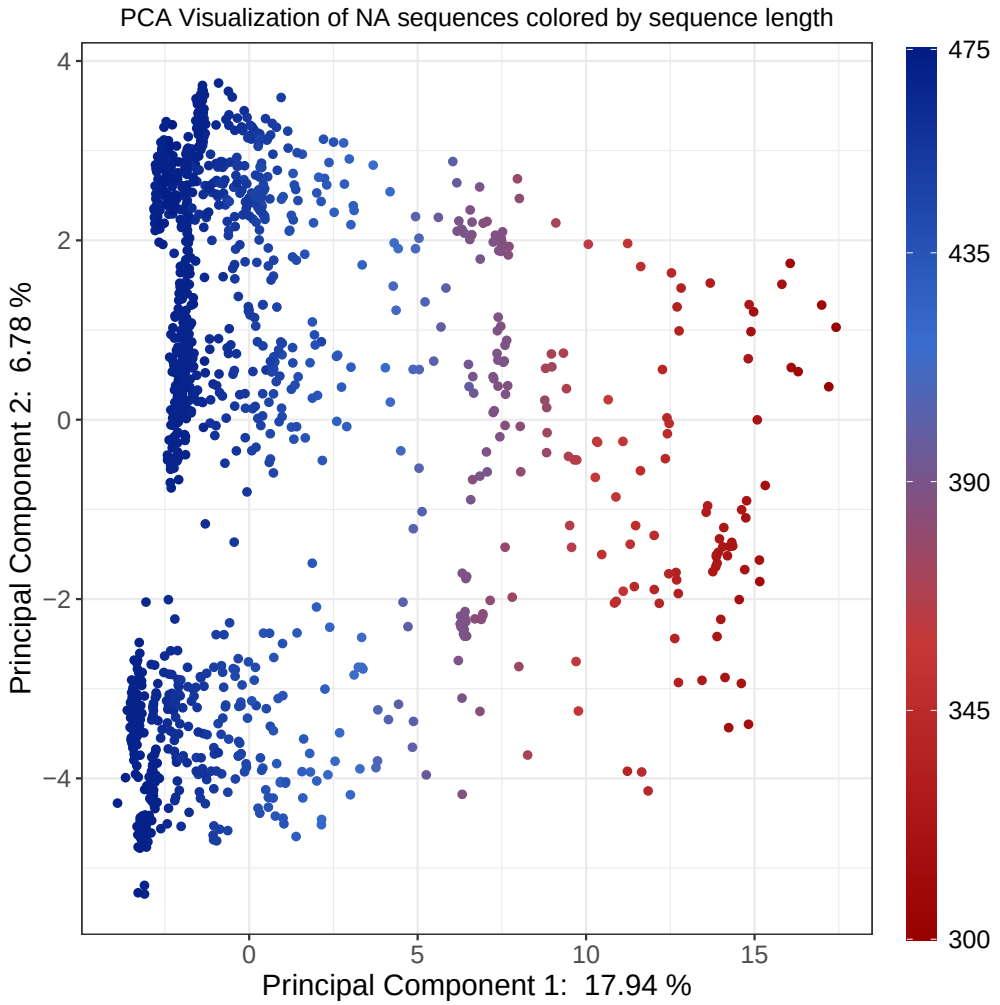


Figure 17: Visualization of 1506 neuraminidase sequences colored by sequence length ranging from red for shorter sequences to blue for longer sequences. The amino acid sequences were processed according to the natural vector method by Y. Wang et al. (2019). The figure represents the two first principal components of the principal component analysis.

Due to this, a multiple sequence alignment was performed as described in Chapter 3.4. Once the MSA contained only the NA head domain of every sequence, the sequences were once again transformed into natural vectors and PCA was again performed to visualize the first two principal components (see Fig. 18). Now, the majority of NCBI sequences has approximately the same sequence length as the PDB sequences. The work was continued with this updated dataset, meaning the natural vectors of the truncated sequences were used for methods needing numerical data and for the following PCA visualizations of the natural vectors colored by other labels, such as *NA groups*, *NA subtypes*, *hosts* and *continents*.

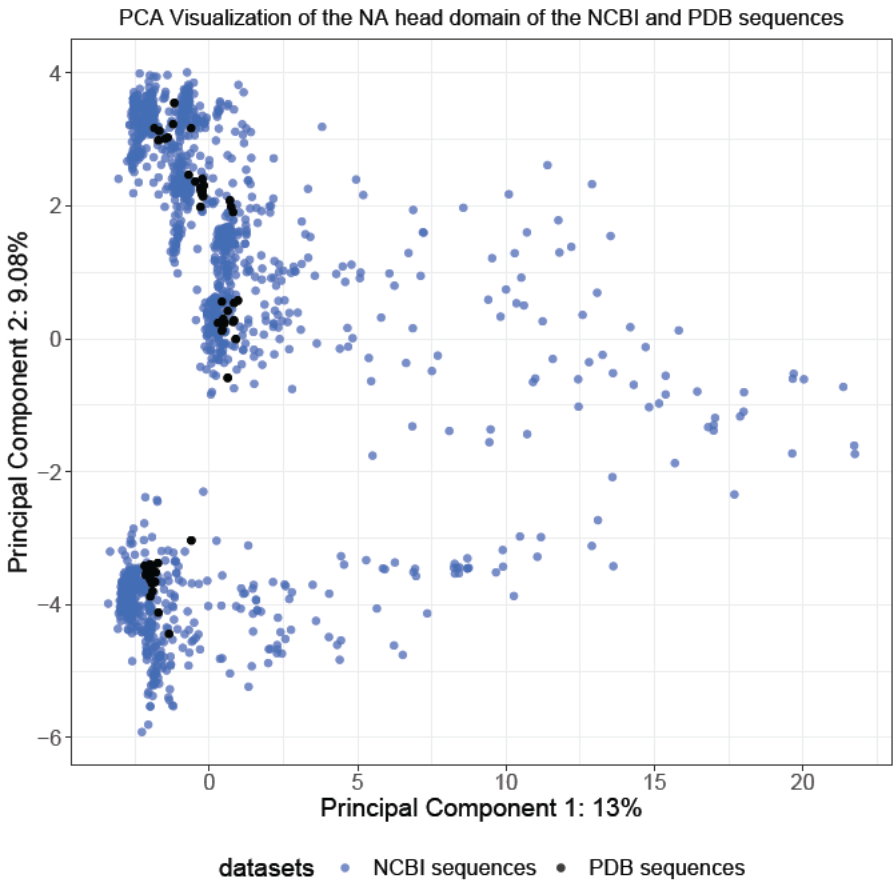
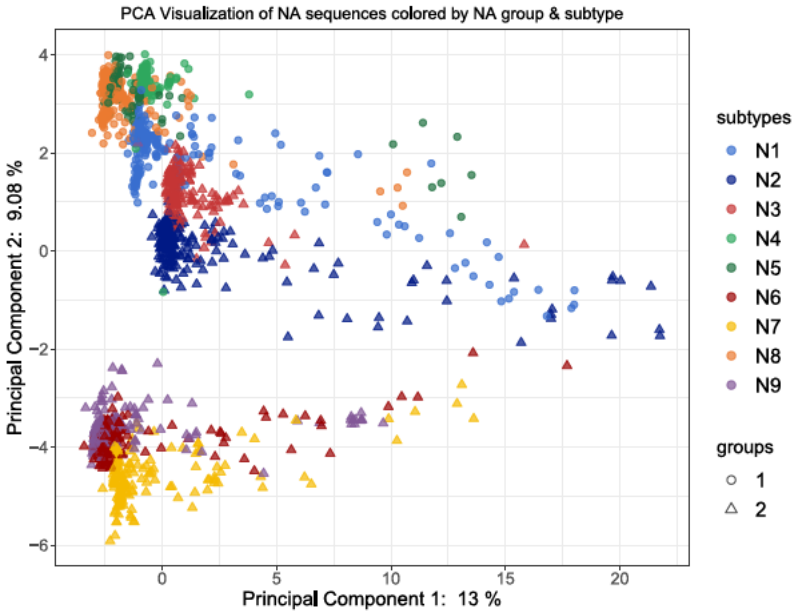


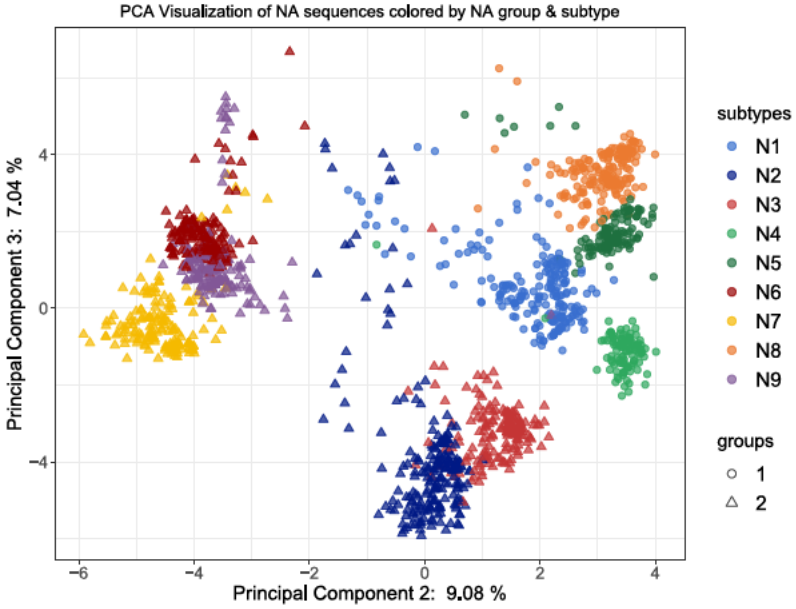
Figure 18: Visualization of 1506 neuraminidase natural vectors colored by subdataset: natural vectors of NCBI sequences are shown in blue and natural vectors of PDB sequences are shown in black. The amino acid sequences were processed according to the natural vector method by Y. Wang et al. (2019). The figure represents the two principal components of the principal component analysis.

The following Figure 19 illustrates the first principal components of PCA. Figure 19a represents PC1 and PC2 and Figure 19b PC2 and PC3 with the datapoints colored by their respective NA subtype. The shape of the datapoints determines the NA group to which they belong. Datapoints of group 1 are shown as circles, whereas those of group 2 are shown as triangles. As illustrated, datapoints of the same subtype lie close to each other. Likewise, datapoints belonging to the same group also lie closer together, as demonstrated in Fig. 19b. It may be presumed, that the distances of the subtypes and groups mirror the phylogenetic tree of neuraminidase subtypes as seen in Fig. 5. Thus e.g. for group 2, the dark blue and the salmon datapoints of the subtypes N2 and N3 are on one branch in the phylogenetic tree and close to one another in the PCA visualization. Same applies to the red, yellow and lilac datapoints of NA subtype N6, N7 and N9. Those subtypes are also located on one branch in the phylogenetic tree, which demonstrates closer relatedness, with N6 and N9 being closer related to each other. This is reflected in the overlapping red and lilac datapoints. The visualization of relatedness is equally true for datapoints of group 1. Even though the datapoints

of neuraminidase subtypes seem to intertwine to some degree in this illustrations, the groups seem to be fairly linear separable. To determine if a machine learning model can classify the data by group or subtype, the GMLVQ will be applied to this dataset (see Chapter 6).



(a) PCA Visualization of PC1 and PC2 of NA sequences colored by NA subtypes



(b) PCA Visualization of PC2 and PC3 of NA sequences colored by NA subtypes

Figure 19: PCA Visualization of 1506 neuraminidase sequences colored by NA groups with group 1 represented in blue and group 2 in green. The amino acid sequences were processed according to the natural vector method by Y. Wang et al. (2019). The figures represent two principal components of the principal component analysis.

Fig. 20 illustrates the datapoints of PC2 and PC3 colored by continent. Thus, the corresponding datapoints for the sequences from Africa are colored in pink, those from Antarctica in orange, from Asia in light blue, from Australia in green, from Europe in yellow, from North America in dark blue and those from South America in red. The first thing to mention is the sheer number of datapoints from Asia, Europe and North America. Although the datapoints of the same continent are mostly located together in the data space, a clear separation between the individual continents is not recognized. It is noteworthy that the natural vectors are 60 dimensional and that those 60 dimensions are reduced for illustrative purposes to two dimensions. This means, that the data may be classifiable in a higher space, even though the 2D representation does not reflect this. Hence, the data with class label *continents* will be used as input for GMLVQ (see Chapter 6).

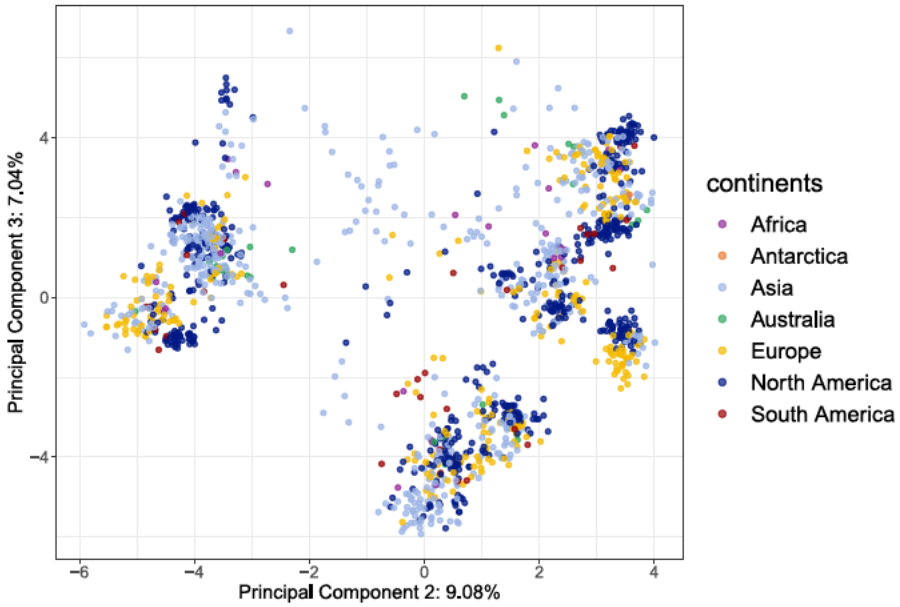


Figure 20: Visualization of 1506 neuraminidase sequences colored by continent of isolation. The amino acid sequences were processed according to the natural vector method by Y. Wang et al. (2019). The figure represents two principal components of the principal component analysis.

At last, the datapoints were colored according to the respective virus host of the NA sequences (Fig. 21). In total, eight organisms of isolation plus isolation from environment were determined. Datapoints from avian hosts are colored blue and are the most abundant datapoints. Moreover, datapoints from canines are colored in red, from cetaceans in green, from equine hosts in orange, from humans in lilac, from mustelines in violet, from phocids in pink, from porcine hosts in grey. Viruses sampled from the environment are shown in yellow. The datapoints of equine hosts (orange) are conspicuous because most of them are close together in the 2D space. In contrast, datapoints belonging to environment seem scattered. Because no host could be identified in the latter, it is speculated that GMLVQ may classify these sequences to the appropriate host. For in-

stance, it can be taken into consideration, that the corresponding protein sequences of environment datapoints located within avian datapoints may actually be from an avian hosts.

Another striking observation is that the data points from porcine hosts (grey) are close to those from human hosts (lilac). This could be explained by the fact that pigs are considered mixed hosts and can become infected with human IAV. This also means that they can transmit human and porcine IAV subtypes to humans. This happens, for example, when humans and livestock live closely together (Ma et al., 2009).

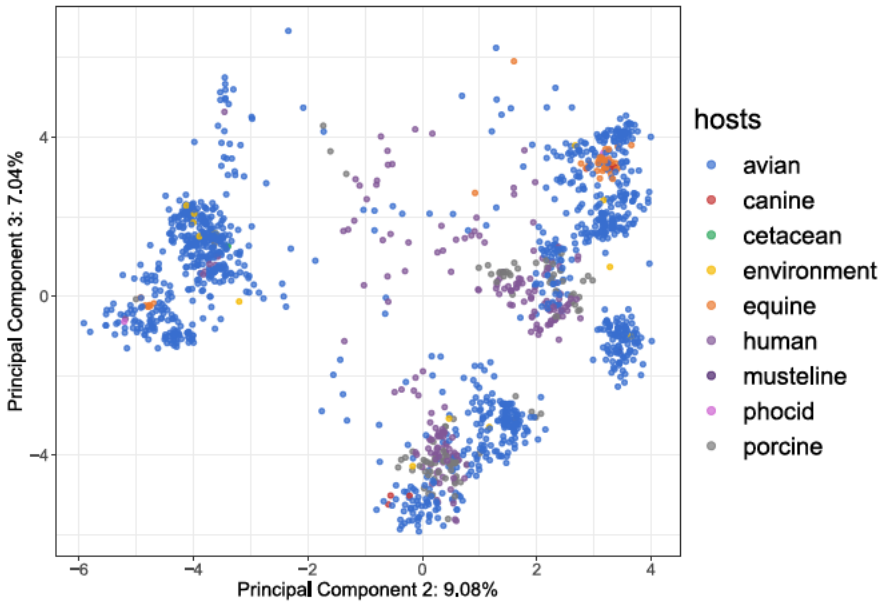


Figure 21: Visualization of 1506 neuraminidase natural vectors colored by host. The amino acid sequences were processed according to the natural vector method by Y. Wang et al. (2019). The figure represents principal component 2 and principal component 3 of the principal component analysis.

5.2 Comparison between BLAST and Natural Vector Distance

After generating a working dataset, the 1506 sequences were divided by PDB sequences and NCBI sequences, so that there are 60 PDB sequences and 1446 NCBI sequences. For readability, the 60 PDB sequences will be referred to as **subdataset-PBD** (SDS-PDB) and the 1446 NCBI sequences as **subdataset-NCBI** (SDS-NCBI). The two were then used in the *basic local alignment search tool* (BLAST). With BLAST, a biological sequence can be compared to a database to identify those sequences in the database, that correspond to the input sequence based on sequence identity.
In this thesis, SDS-PDB sequences were used as the database against which the SDS-NCBI sequences were run. Due to the objective being the comparison of BLAST results to an alignment free method, BLAST will not be discussed in detail. For information on BLAST, please consider Altschul et al. (1990). BLAST was executed via the R-Package *rBLAST* after downloading and installing the BLAST software separately. The sequence data needs to be in fasta-format and the type of the database was set to “protein” as well as the blast-type to “blastp” and the algorithm took 58 seconds. The percentage identity of all SDS-NCBI sequences to the SDS-PBB sequence is in median 48.69%. The minimum sequence identity is of 18.37% for an avian N8 NA (ID: 5HUN) and a canine N8 NA (ID: ABA46974). The maximum percentage identity is of 100.00%, for example for two sequences of same subtype, same host and same year of isolation.
As the natural vector method can be used as an alignment free method for sequence comparison (Bohnsack et al., 2022), the Euclidean distances between the NV from SDS-NCBI to SDS-PDB were calculated. This took only a fraction of a second and is much faster than the BLAST algorithm. As a result, the median Euclidian distance was of 131.59, with a minimum distance of 10.65 and a maximum distance of 412.41.
The results from BLAST and from vector distance calculation were then compared. Therefore, both were ranked: BLAST from highest percentage identity to lowest and similar NV distances from closest to farthest distance. It is expected that sequences with high percentage identity also have a close NV distance and therefore have the same ranking score, but the juxtaposition of the ranking scores yielded some interesting results.
Some BLAST results match with the NV distance results, which means that the sequences most similar according to BLAST are also the respective natural vectors with the closest distance to each other. Interestingly, there were also discrepancies between both methods, which were categorized into four types. The first type describes the first two highest ranking sequence pair in BLAST being inverted in the NV distance ranking, meaning the sequences with BLAST rank 1 have NV rank 2 and the entry at BLAST rank 2 has NV rank 1. Therefore, this conspicuity will be called *inversion*. In the second type of discrepancies, the two highest matching sequences in BLAST are crossed over in the two closest natural vectors. In detail, this means, that the two closest natural vectors are a combination of one sequence from the highest ranking BLAST pair

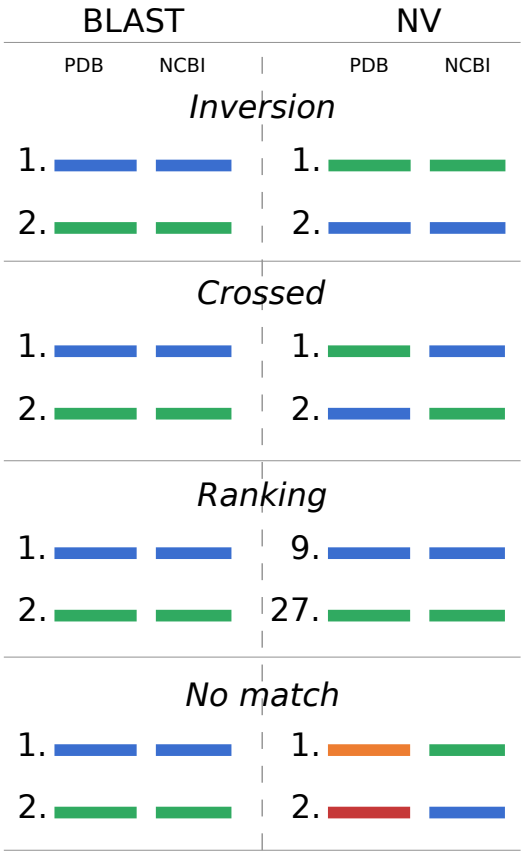


Figure 22: Comparison of BLAST and NV distance results, ranked from most similar (BLAST) or closest distance (NV) to least similar/farthest distance. Four types of discrepancies between BLAST and NV ranking are identified: *Inversion*, *Crossed*, *Ranking* and *No match*. The colors indicate a sequence or NV pair in the ranking.

with one of the second highest ranking BLAST pair. Figure 22 illustrates this type as *crossed* type. Furthermore, as seen in the figure, two additional types were identified. The third, called *ranking*, indicates that the ranking of BLAST and NV differ from each other. It could only differ in some minor ranking scores, but did appear to differ greatly in five cases. The last type deals with no match at all between the individual compared sequences (see Figure 22, *no match*).

Overall, the embedded vector distances closely approximate the sequence distances only in a few cases and it could not be determined, which method rather reflects the reality. At this point, further studies are needed to analyze why these discrepancies occur.

6 Classification using GMLVQ

Since EVcouplings did not provide satisfactory results in this particular case and the task is interpreted as a classification problem given that label information is provided, GMLVQ was used for classification. The class labels depend on the additional information to each neuraminidase sequence and were chosen to be *years*, *group*, NA *subtype*, *continent* and *host*. Since vectors of numerical data are required, especially for distance-based machine learning methods, the amino acid sequences were transformed into vectors. This was accomplished using the natural vector method as described in Ch. 2.2, which resulted in vectors of equal length of 60 dimensions. These vectors were used as input for the GMLVQ. The GMLVQ model code used in this thesis was acquired through SICIM's GitHub and adapted for the specific classification task.
For every class, one prototype was initialized and the learning rate of the prototypes was set to 0.01, as well as the maximum epochs to 1000. Further parameter settings are listed in the Appendix Table A.27. The GMLVQ model is verified with 5-fold cross validation. Based on the values of the confusion matrices (see Tables 8-12), various confusion metrics have been calculated for each class according to the equations in Chapter 2.4. The classification validation measures *accuracy* (Eq. 2.20), *precision* (Eq. 2.22), *sensitivity* (Eq. 2.23) and *specificity* (Eq. 2.24) are listed in Tables 9, 11 and 13. In addition, the most important features for the classification were discovered with the generated Λ matrix.

6.1 Classification by NA years

As Fig. 15 showed no indication of sequence evolution over time, GMLVQ was performed and the year intervals were used as the class labels as listed in Chapter 5. Year intervals were chosen instead of the year of isolation itself as this would have led to 63 classes, some with only one or two datapoints. This would otherwise have resulted in a severely imbalanced dataset and in a decreased model performance (Gupta et al., 2014). The intervals were chosen to include data points corresponding to sequences from epidemic or pandemic periods. These are described to some extent in Chapter 1. Additionally, as of the year 2000, the intervals are of size between 4-5 years, due to the amount of data from these time periods in the dataset. The mean accuracy of the GMLVQ model verified with 5-fold cross validation is 18.39%, which amounts to 277 out of 1506 correctly classified datapoints. The accuracies of each fold did not differ much from this value and therefore will not be discussed in detail, but can be found in Table A.29. As the dataset is imbalanced, the balanced accuracy was calculated by Eq. 2.21 and is 16.39%. Both the accuracy and balanced accuracy are so low that it cannot be assumed that the model has performed adequately.
This can also be deduced from Table 8. There, the true positives are highlighted in

green and the sum of all datapoints in blue. The relative frequencies of TP range from 7.05% for the years *2011–2015* to 26.70% for the years *2006–2010*. Only a fraction of each class is classified correctly as TP by the model. The significant number of 409 FP datapoints in class *1918–1959* could indicate relatedness between the sequences, especially since all influenza A subtypes are speculated to be related to 1918 influenza A H1N1 (Taubenberger and Morens, 2006). It could equally be that IAV subtypes from before 1960 reemerged in the following years, therefore being classified as class *1918–1959*.

Table 8: Confusion matrix of GMLVQ with class label *years*. The classes in the classification tasks represent the year intervals of virus isolation. In addition to Positives and Negatives, the sums of the predicted positives ($\sum$ predicted) and the datapoints of each class ($\sum$ true) are listed. True positives are highlighted in green and the sum of all datapoints in blue.

		Predicted						
		1918–1959	1960–1999	2000–2005	2006–2010	2011–2015	2016–2019	$\sum$ true
True	1918–1959	2	12	1	1	0	1	17
	1960–1999	72	61	33	40	19	23	248
	2000–2005	62	38	22	50	12	35	219
	2006–2010	156	132	30	153	27	75	573
	2011–2015	104	94	35	65	27	58	383
	2016–2019	15	16	7	15	1	12	66
	$\sum$ predicted	411	353	128	324	86	204	1506

Based on the values in this confusion matrix, precision, sensitivity and specificity have been calculated for each class. They are listed in Table 9.

Table 9: Confusion Metrics Precision, Sensitivity and Specificity of every class in neuraminidase years in percent

	Precision	Sensitivity	Specificity
1918–1959	0.49%	11.76%	72.53%
1960–1999	17.28%	24.60%	76.79%
2000–2005	17.19%	10.05%	91.76%
2006–2010	47.22%	26.70%	81.67%
2011–2015	31.40%	7.05%	94.75%
2016–2019	5.88%	18.18%	86.67%

The specificity being relatively high in contrast to the other classification validation measures means that the model can, to a certain degree, correctly classify datapoints not belonging to the respective class (TN). Further, the precision of every class is less than 50%. The lowest precision overall is 0.49% for the class *1918–1959*. This means that a maximum of 0.49% of positive datapoints of this class are correctly classified as true positives. It should be noted that only 17 out of 1506 or 1.13% of the dataset belongs to class *1918–1959*, which means that this class is strongly underrepresented.
The sensitivity is lowest for class *2011–2015* with only 7.05% and highest for class *2006–2010* with 26.70%. These values show, that the model is not capable to correctly identify TP. Since these results do not suffice in showing evolutionary correlation, the

classification task will be expanded to other class labels, such as *groups*, *subtypes*, *hosts* and *continents*.

6.2 Classification by NA Groups

Considering the PCA results of Figure 19, it was determined, that the dataset could be classified into NA groups. This leads to a binary classification problem with the class labels being *group 1* and *group 2*. The parameter settings for this classification task were the same as in Ch. 6.1 and the GMLVQ model achieved a mean accuracy of 99.73%. Figure 23 shows the datapoints in the latent dimension space of the model. The groups seem to be linear separable. Datapoints of class *group 1* are represented as blue circles and datapoints of class *group 2* in green circles. The respective prototypes are illustrated as diamonds. The visualization was attained at the end of the model training.

Table 10: Confusion matrix of GMLVQ with class label *groups*. True positives are highlighted in green and the sum of all datapoints in blue.

		Predicted		
		NA Group 1	NA Group 2	∑ true
True	NA Group 1	630	1	631
	NA Group 2	3	872	875
	∑ predicted	633	873	1506

The confusion matrix (Table 10) shows that the GMLVQ is well trained for this binary classification problem. Almost all datapoints of both classes were correctly classified. Out of 631 datapoints in class *group 1*, 630 are correctly classified, and out of 875 datapoints in class *group 2*, 872 are correctly classified. The TP are highlighted in green, while the sum of all datapoints is emphasized in blue. Based on this confusion matrix, the classification validation measures precision, sensitivity and specificity were calculated and are listed in Table 11.

Table 11: The Confusion Metrics Precision, Sensitivity and Specificity of every class in neuraminidase groups in percent

Precision	Sensitivity	Specificity
99.53%	99.84%	99.66%

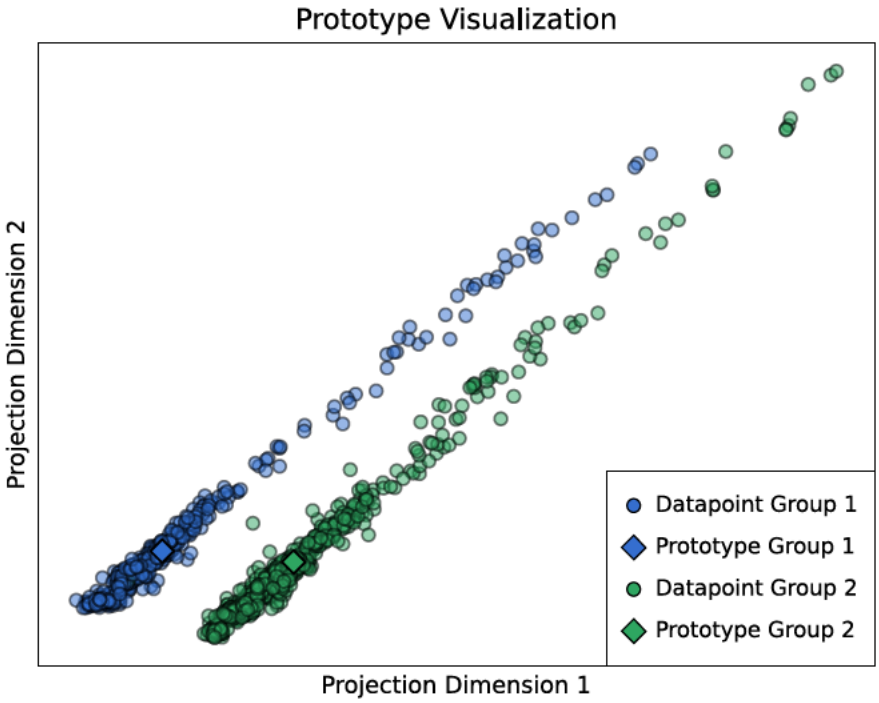


Figure 23: Two dimensional visualization of datapoints and prototypes with labels *group 1* and *group 2* at the end GMLVQ training. Datapoints of class *group 1* are represented as blue circles and datapoints of *group 2* in green circles. The respective prototypes are illustrated as diamonds.

The precision, indicating how much the model can be trusted to predict a datapoint as positive (Grandini et al., 2020), is 99.53% for this binary classification problem. The sensitivity of 99.84% denotes, that almost all positive datapoints were correctly classified (TP), and further, the specificity of 99.66% shows, that almost all negative datapoints were correctly identified (TN). Both sensitivity and specificity together suggest, that the model can differentiate both classes well from one another.
To verify which features are important for such an accurate classification, the Λ matrix is described and interpreted in the following. This matrix shows positive as well as negative correlated features. In Figure 24, the first 20 dimensions, n_a, of the respective Λ matrix are depicted, as only those seem to be more or less relevant for classification. The entire matrix is shown in Fig. A.34 The relevance is coded as negative correlation (blue) to positive correlation (red) for all features besides the main diagonal. For example, n_F and n_C are negative correlated, as denoted by the shade of blue, which means that if the absolute frequency of one residue is high, it has to be low for the other to be decisive for the classification. The opposite is true for positive correlated features such as n_H and n_C (as denoted by the shade of red), meaning both features need to be either of high or low values to be conclusive for the classification. The most remarkable feature is the absolute frequency of cysteines n_C in dark red on the main diagonal. The greater the absolute value, the more important the respective feature is in the classification process. Values illustrated in a white color or with weak coloring can be neglected in the interpretation of the Λ matrix, as these are of small to none

importance for the classification. This means, that the quantity of cysteines (C) in the neuraminidase sequences plays an important role in the classification decision. This is also denoted by the correlation from n_C to other features. Nevertheless, the combination of all features is important for classification, even though only the most important ones will be further investigated. The biological explanation will be given in Ch. 6.6. All following Λ matrices can be interpreted following the described procedure above. Other classifications with class labels *NA subtypes* and moreover class labels *NA subtypes* divided in their respective group will be investigated in the following chapters.

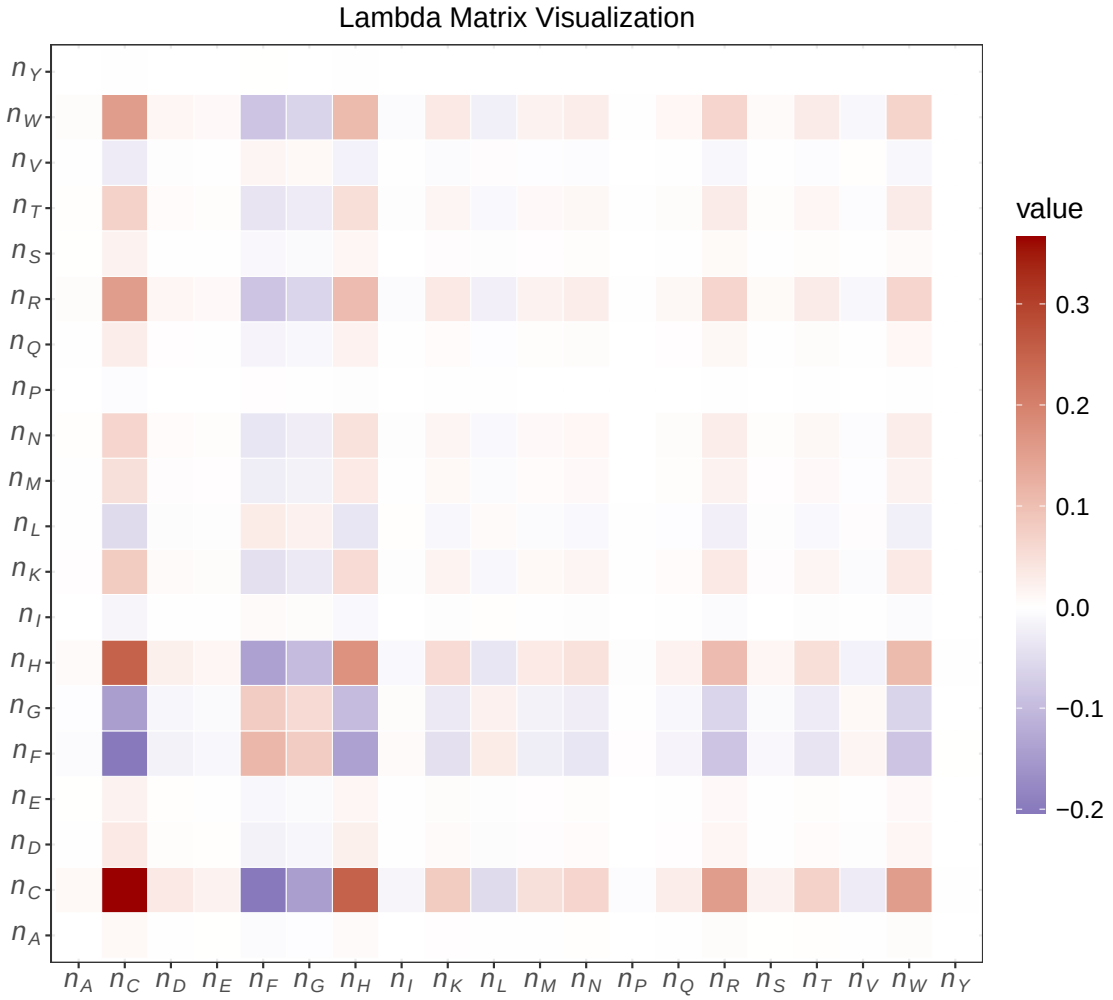


Figure 24: Visualization of first 20 dimensions of Λ Matrix after classification by *groups*. The color gradient from blue to red indicates negative to positive values. Blue values denote negative correlation, while red values denote positive correlation.

6.3 Classification by NA subtypes

For classification by NA subtypes, GMLVQ was used in the same way and with the same parameters as described above. The class labels were set to the neuraminidase subtypes *N1–N9*. The accuracy of classification is 99.23%, and as seen in the latent dimension space of the model in Fig. 25, the nine point clouds are colored by NA subtype and fairly separable. Datapoints are represented as circles, while the respective prototypes are illustrated as diamonds. Class *N1* datapoints are shown in light blue, class *N2* in dark blue, class *N3* in salmon, class *N4* in yellow, class *N5* in light green, class *N6* in dark green, class *N7* in orange, class *N8* in lilac and class *N9* datapoints in red.

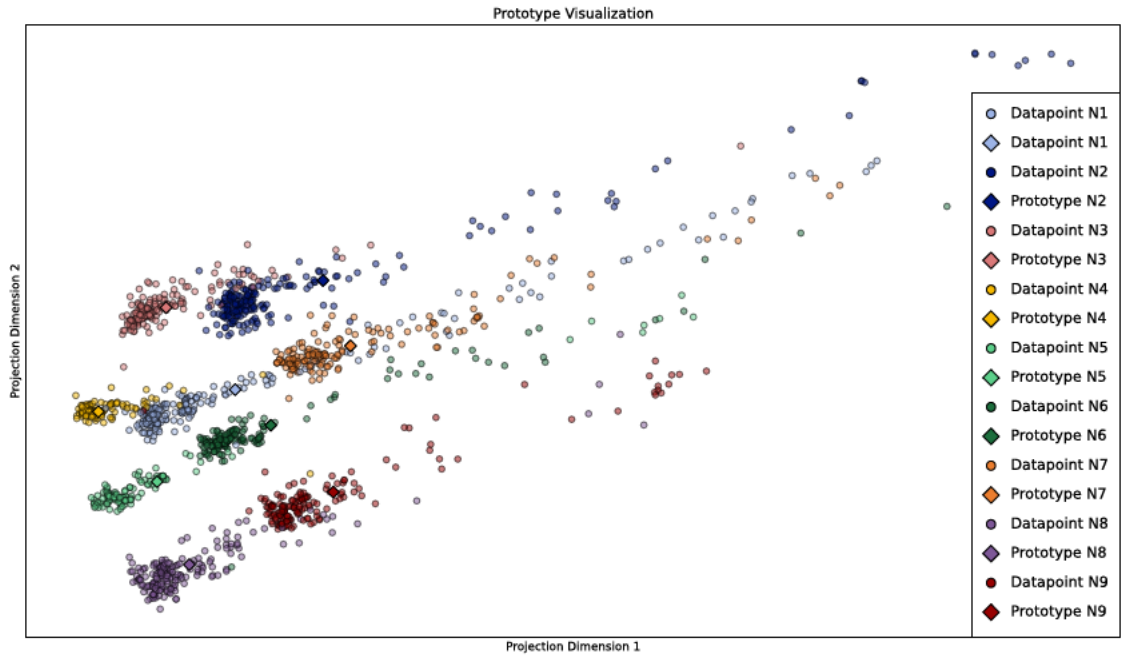


Figure 25: Two dimensional visualization of datapoints and prototypes with class labels *subtypes*. Datapoints are represented as circles and the respective prototypes are illustrated as diamonds.

The confusion matrix for the neuraminidase subtype classification (Table 12) shows that the GMLVQ is well trained for this multi-class classification problem. Almost all datapoints of all classes were correctly classified. The TP are highlighted in green, while the sum of all datapoints is emphasized in blue.

Based on the confusion matrix, the classification validation measures precision, sensitivity and specificity were calculated and are listed in Table 13. The precision, sensitivity and specificity for every class have percentages over 97%. The precisions over 97% indicate that the model is able to predict a datapoint as positive. The sensitivities over 98% denote, that almost all positive datapoints were correctly classified (TP), and further, the specificities of over 99% show, that almost all negative datapoints were

Table 12: Confusion matrix of GMLVQ with class label *subtypes*

		Predicted									
		N1	N2	N3	N4	N5	N6	N7	N8	N9	∑ true
True	N1	210	0	0	0	0	0	0	0	1	211
	N2	0	212	0	1	0	0	0	0	0	213
	N3	0	0	170	2	0	0	0	0	0	172
	N4	1	0	0	110	0	0	0	0	1	112
	N5	0	0	0	0	123	0	0	0	0	123
	N6	0	0	0	0	0	168	0	1	0	169
	N7	0	0	0	0	0	1	159	0	0	160
	N8	1	0	0	0	0	0	0	184	0	185
	N9	2	0	0	0	0	0	0	0	159	161
	∑ predicted	214	212	170	113	123	169	159	185	161	1506

correctly identified (TN). Even though the model is slightly better at identifying TN, sensitivity and specificity suggest that the model can differentiate all classes well from one another. For determination of the important features for the classification, the Λ matrix is described and interpreted in the following.

Table 13: The Confusion Metrics Precision, Sensitivity and Specificity of every class in neuraminidase subtypes in percent

Class	Precision	Sensitivity	Specificity
N1	98.13%	99.53%	99.69%
N2	100.00%	99.53%	100.00%
N3	100.00%	98.84%	100.00%
N4	97.35%	98.21%	99.78%
N5	100.00%	100.00%	100.00%
N6	99.41%	99.41%	99.93%
N7	100.00%	99.38%	100.00%
N8	99.46%	99.46%	99.92%
N9	98.77%	98.77%	99.85%

The Λ matrix, as shown in Fig. 26, represents the first 20 dimensions of the Λ matrix, as only these seem to be important for the classification task. For a 60 dimensional visualization, see Fig. A.35. n_W and n_C are negative correlated. The quantities of the amino acid phenylalanine (F) as well as proline (P), glutamine (Q) and tryptophan (W) play a small role. Similar to the classification by NA groups, n_C colored in dark red shows a strong positive correlation and it can therefore be assumed, that cysteine plays a major role in differentiating the classes *N1–N9*.

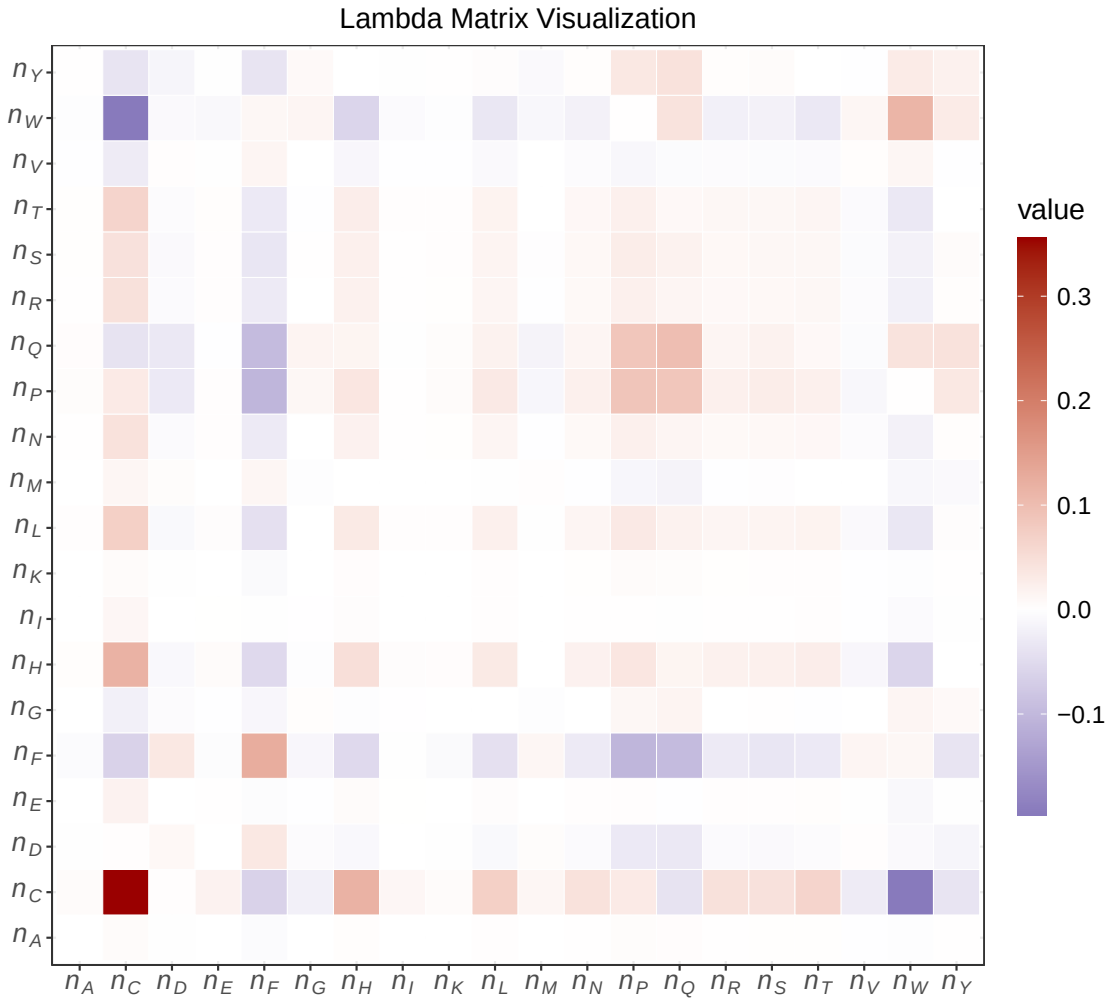
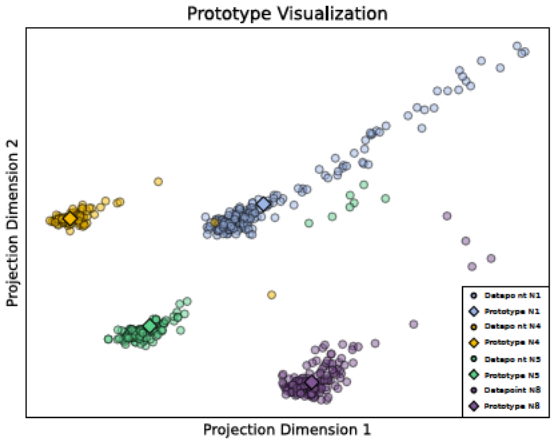


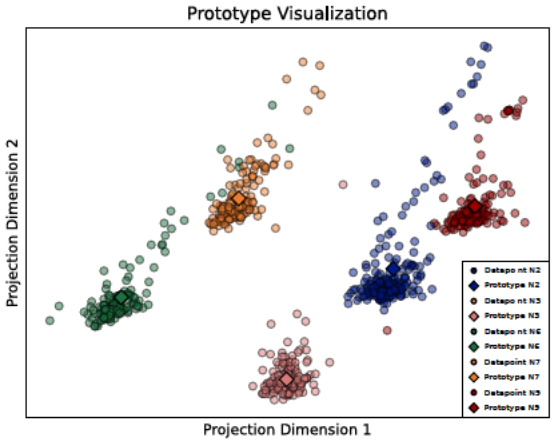
Figure 26: Visualization of first 20 dimensions of Λ Matrix after classification by *subtypes*. The color gradient from blue to red indicates negative to positive values. Blue values denote negative correlation, while red values denote positive correlation.

6.4 Classification by NA Subtypes divided in their respective Group

Although classification occurs through the interaction of various features, the strong positive correlation of cysteine is remarkable. The dataset is divided into corresponding groups in this chapter, to see whether the subtypes in their respective groups are also classifiable almost exclusively by n_C. GMLVQ was executed for each group individually as described before with the same parameters and prototype initialization. Further parameter settings are listed in Table A.27. For group 1, the class labels were set to the different subtypes of this group *N1, N4, N5* and *N8*. The accuracy of the model is 99.68%. For NA subtypes in group 2 *N2, N3, N6, N7* and *N9* the accuracy is 99.89%. This comes to no surprise, as the previous GMLVQ model could classify all NA subtypes with an accuracy of 99.23%.



(a) Visualization of datapoints and prototypes colored by NA subtype in group 1



(b) Visualization of datapoints and prototypes colored by NA subtype in group 2

Figure 27: Visualizations of datapoints and prototypes colored by NA subtype in group 1 and group 2 as follows: (a) N1 in light blue, N4 in yellow, N5 in light green and N8 in lilac; (b) N2 in dark blue, N3 in salmon, N6 in dark green, N7 in orange and N9 in red.

The Figures 27a and 27b show visualized datapoints in a 2D space at the end of the training. The point clouds datapoints colored by subtype seem fairly separable. Here again, datapoints are represented as circles, whereas the respective prototypes are illustrated as diamonds. In Fig. 27a the group 1 class *N1* datapoints are shown in light blue, class *N4* in yellow, class *N5* in light green and class *N8* datapoints in lilac. In Fig. 27b, the group 2 class *N2* datapoints are colored in dark blue, class *N3* in salmon, class *N6* in dark green, class *N7* in orange and class *N9* datapoints in red.
The confusion matrices of both GMLVQ models are shown in Table 14. In both cases, GMLVQ seems well trained for these two multi-class classification problems. Almost all datapoints of all classes were correctly classified. The TP are highlighted in green, while the sum of all datapoints is emphasized in blue. The misclassification of two datapoints of class label *N4* to class *N1* (Tab. 14a) is most likely due to the close relatedness of the NA subtypes N1 and N4 (see Fig. 5).

Table 14: Confusion matrix of GMLVQ with class labels *NA subtype* in both group 1 and group 2

		Predicted				
		N1	N4	N5	N8	∑ true
True	N1	211	0	0	0	211
	N4	2	110	0	0	112
	N5	0	0	123	0	123
	N8	0	0	0	185	185
	∑ predicted	213	110	123	185	631

(a) Confusion matrix of GMLVQ with class label *subtype in group 1*

		Predicted					
		N2	N3	N6	N7	N9	∑ true
True	N2	213	0	0	0	0	213
	N3	0	172	0	0	0	172
	N6	0	0	169	0	0	169
	N7	0	0	0	160	0	160
	N9	1	0	0	0	160	161
	∑ predicted	214	172	169	160	160	875

(b) Confusion matrix of GMLVQ with class label *subtype in group 2*

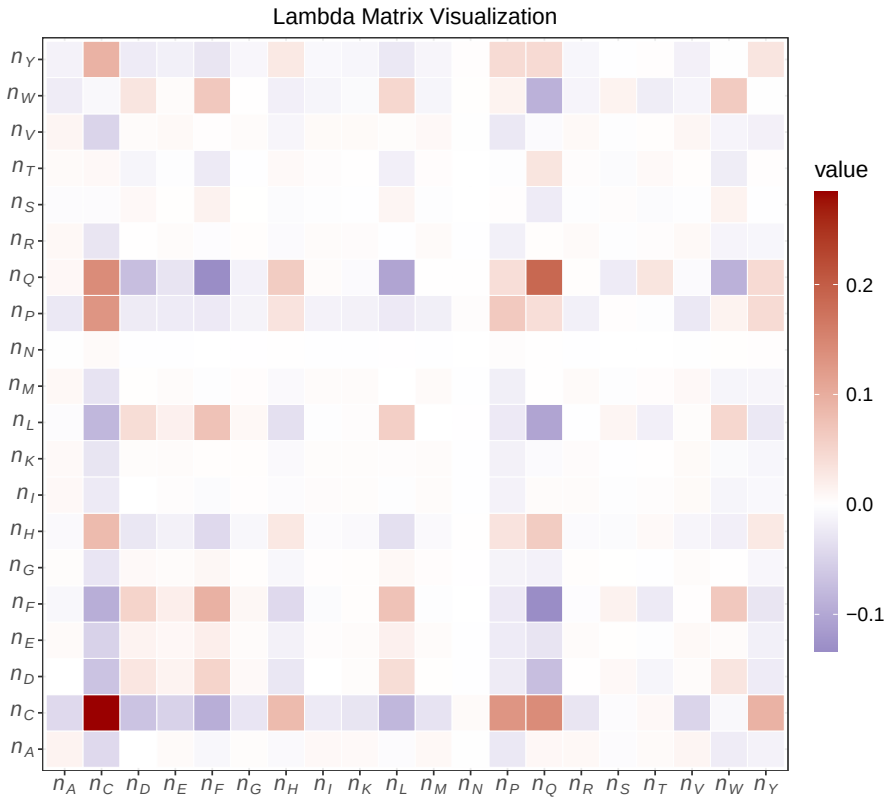
Based on the values in both confusion matrices, the aforementioned classification validation measures were calculated. They are listed in Table A.30. The results of all three metrics for both cases display values of 99% or higher. Both GMLVQ models can accurately classify the NA subtypes of each individual group.

The visualization in Figure 28 displays the first 20 dimensions of the Λ matrices of the NA subtypes of group 1 (Fig. 28a) and of the NA subtypes of group 2 (Fig. 28b). The complete Λ matrices are illustrated in Fig. A.36 and Fig. A.37.

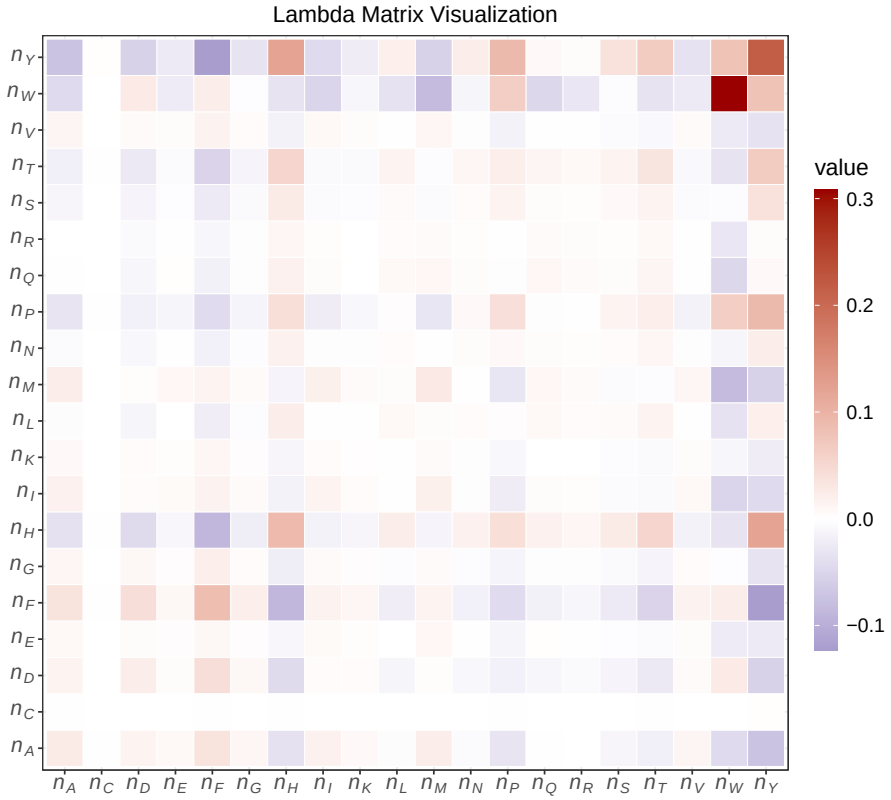
In Fig. 28a, once again the absolute frequency of cysteine (dark red) stands out. Additionally, n_Q seems to be important for classification and it is positive correlated to n_C, whereas n_Q and n_F, and n_Q and n_L are negative correlated. Furthermore, n_C and n_F are in positive correlation to each other.

Surprisingly, in Fig. 28b, cysteine is not important for the classification, but rather the frequencies of tryptophan n_W and tyrosine n_Y, as seen by their darker shades of red in the main diagonal. In addition, n_Y and n_F are negative correlated as well as n_H and n_F.

In conclusion, the NA subtypes of group 1 are differentiable in particular by the absolute frequency of cysteines in the protein sequences, in contrast to NA subtypes of group 2, which are notably distinguishable by the absolute frequency of tryptophans and tyrosines in their respective protein sequences. Further interpretation of these results is discussed in Chapter 6.6.



(a) Visualization of first 20 dimensions of Λ Matrix after classification by *subtypes* of group 1.



(b) Visualization of first 20 dimensions of Λ Matrix after classification by *subtypes* of group 2.

Figure 28: Visualization of first 20 dimensions of Λ Matrix after classification by *subtypes* of group 1 and group 2. The color gradient from blue to red indicates negative to positive values. Blue values denote negative correlation, while red values denote positive correlation.

6.5 Problematic with Classifications by Hosts and by Continent

Classifications by hosts and by continents were performed, but poor results were obtained due to imbalanced class representation. The hosts are limited to eight taxa plus "environment". For the latter, the organism of isolation cannot be determined beyond doubt, as the samples were extracted from aquatic environments or similar. Moreover, there are 1140 sequences (75.7%) in the dataset that were isolated from avian hosts. The accuracy of the model is 69.63%, with the accuracies of the five folds varying from 3.31% - 91.36% (see Table 15).

Table 15: Fold accuracy of classification by host

folds	fold 1	fold 2	fold 3	fold 4	fold 5
accuracy	3.31%	80.06%	90.70%	91.36%	82.72%

Even though the majority of accuracies is over 80% (apart from fold 1 accuracy), it cannot be trusted to reflect a well trained model. This is because not all classes were represented in all folds, resulting in incomplete confusion matrices, which in turn leads to the inability to calculate confusion metrics like precision, sensitivity and specificity. The dataset is strongly imbalanced in favor of the class *avian*, thus that accuracies of over 90% denote rather a coincidence of classifying a datapoint correctly.
Similar is the case for classification by continent. Asia and North America are overrepresented. Both account for 74.63% of the dataset with a total of 1124 sequences. Here again, not all classes were represented in all fold stages, so that no confusion matrix could be generated. The accuracies of each fold range from 1.99% - 58.47% (see Table 16).

Table 16: Fold accuracy of classification by continent

folds	fold 1	fold 2	fold 3	fold 4	fold 5
accuracy	1.99%	46.18%	58.47%	42.19%	46.84%

The models not being able to classify the dataset according to the organism or continent of origin is not surprising, as the dataset is imbalanced in both cases. Furthermore, most sequences were isolated from birds, which are the predominant host for IAV. Those sequences came from domesticated poultry and also wild waterfowl and other migratory birds. Figure 29 illustrates the eight flyways of wild birds in the world. These routes trend in a north-south direction. Especially waterfowl travels wide distances yearly, crossing multiple continents and thus spreading the influenza virus along migratory routes (Zhang et al., 2014; Olsen et al., 2006). Those are, as illustrated, the Pacific Americas

(light blue), the Central Americas (pink), the Atlantic Americas (violet), the East Atlantic (yellow), the Black Sea/Mediterranean (dark blue), the West Asian/East African (red), the Central Asian (orange) and East Asian/Australasian flyway (lilac). Solely Antarctica does not seem to be a destination of migratory birds, even though IAV has been found in penguins at that location.
Thus, IAV can be transmitted to local wild birds as well as to domesticated livestock, such as chickens or swine, since IAV is also found in excreta and aquatic environments such as lakes (Webster, Bean, et al., 1992). This spread of the virus makes it difficult to distinguish between NA from different continents. Furthermore, the majority of the eight migratory routes pass North America and Asia (Zhang et al., 2014; Schäfer, 2019), which explains why many data were collected from these continents.

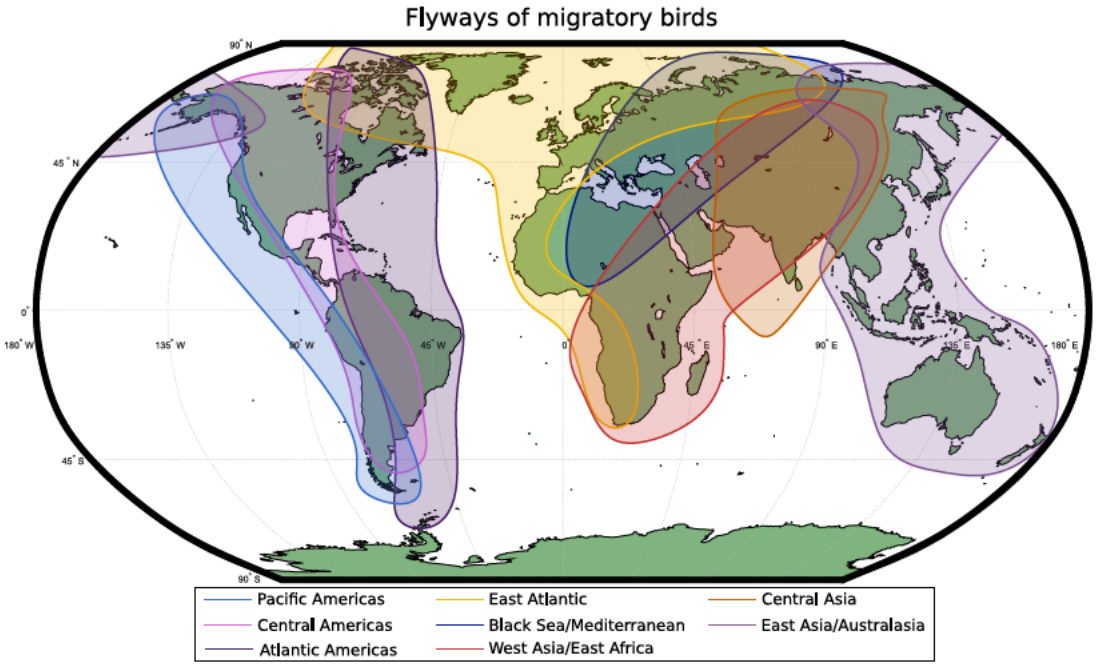


Figure 29: Migration routes for wild birds in the world, adapted from Zhang et al., 2014. Each colored shape represents one flight path (World map is drawn by Matlab R2021b and the routes are added with Inkscape).

To obtain more reliable results with these two classification problems, the dataset needs to be balanced with focus on the number of sequences for either host or continent. Based on the Fig. 21, it could be possible to differentiate between the individual hosts once the classes have been balanced. One other possibility for classification by host would be to separate mammals and birds with the same amount of sequences in each, creating a binary classification problem. To construct a binary classification problem for the continents, the data from Asia and North America can be used. To obtain a multi-class classification problem, data from Europe could be added, with subsequent balancing. Another idea would be to only consider the avian data and classify them by continents.

6.6 Interpretability of Λ Matrices

Based on the results, the question now arises how the positive correlated features of the Λ matrices diagonal can be explained biologically. In the following chapter, the focus will be limited to the analysis on the results from Chapter 6.2, Chapter 6.3 and Chapter 6.4, since the models show the best classification results for these two classification tasks.

6.6.1 Analysis of Cysteine Occurrence and Disulfide Bridges in Neuraminidase

Based on the results of the GMLVQ models discussed in Chapter 6.2 and Chapter 6.3, the cysteine occurrences per NA group and per NA subtype were extracted from the resp. natural vectors and analyzed in more detail. The amino acid cysteine has a sulfur atom located in its side chain, which is responsible for the formation of disulfide bridges (DSB) between two cysteines (Basler et al., 1999). The median absolute frequency of this amino acid in the whole dataset is 18 cysteines per sequences, which is approx. 4.47% of amino acids per sequence. These cysteines were located by generating a sequence logo using *WebLogo* (Crooks et al., 2004), which is a graphical representation of consensus sequences (T. D. Schneider et al., 1990). The sequence logo of the neuraminidase sequences is visualized in Figure 30, with the cysteines (C) highlighted in orange (see Figure A.38 for larger visualization).

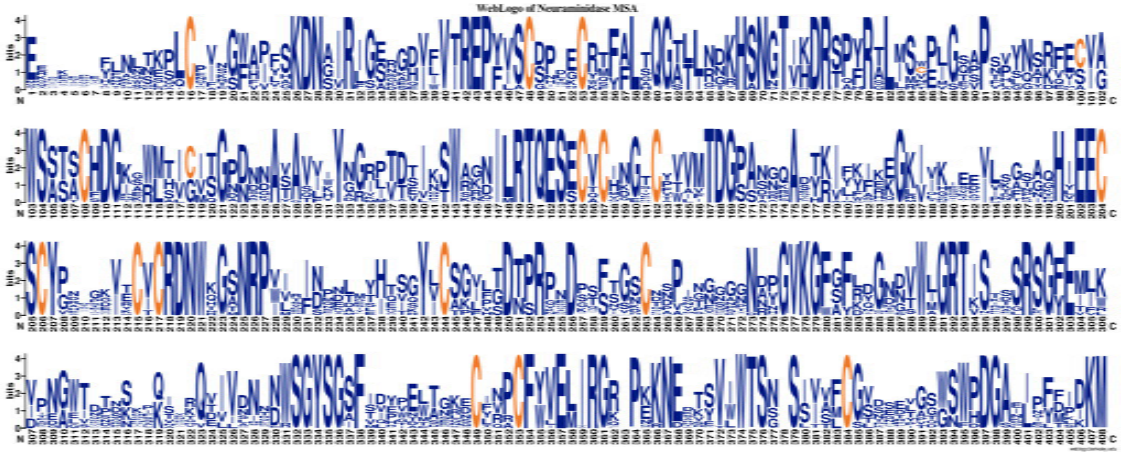


Figure 30: Sequence logo of neuraminidase dataset, with cysteine residues highlighted in orange.

In total, 16 cysteines are highly conserved, which is represented by the height of C. Two additional cysteines at alignment positions 100 and 118 are not as highly conserved throughout the NA sequences. Table 17 provides an explanation for the latter two C. It shows the average occurrences of cysteine residues per group and per subtype. In median, group 2 has one to two cysteines more than group 1. The number of cysteines may consequently play a major role in differentiating the neuraminidase groups from one another, which subsequently could be the decisive feature for the classifier in this classification task.

Table 17: Absolute frequencies of cysteines in neuraminidase sequences per NA group and per NA subtype

	group 1				group 2				
	N1	N4	N5	N8	N2	N3	N6	N7	N9
median cysteine occurrence	17	17	16	16	18	18	18	18	18
max. cysteine occurrence	18	18	17	17	19	20	19	20	20
min. cysteine occurrence	13	14	13	13	11	15	13	15	15

Since cysteines are known to form disulfide bridges and those bonds are fundamental components for the molecular structure of proteins (Wiedemann et al., 2020), the frequency of DSB in NA was also examined. For this purpose, all DSB annotations were loaded according to the *European Molecular Biology Laboratory–European Bioinformatics Institute* (EMBL-EBI) *Protein API* page for searching protein sequence features of a type DISULFID in UniProt. For all neuraminidase UniProt accession IDs, the annotated information to DSB and position in the respective sequence was downloaded.

Table 18: Absolute and relative frequencies of cysteines in neuraminidase sequences per NA group and per NA subtype

	DSB annotation		no DSB annotation		in total
	per group	per subtype	per group	per subtype	
group 1	377 59.8%	N1 115 N4 72 N5 73 N8 117	254 40.2%	N1 96 N4 40 N5 50 N8 68	631 100%
group 2	532 60.8%	N2 138 N3 120 N6 106 N7 81 N9 87	343 39.2%	N2 75 N3 52 N6 63 N7 79 N9 74	875 100%
in total	909 60.4%		597 39.4%		1506 100%

Information on DSB positions in neuraminidase sequences was not available for all sequences of the dataset, so only those for which this was the case are considered here. As seen in Table 18, information on DSB is available for a total of 909 sequences of the dataset (as of January 2022), which is approx. 60.4% of the dataset. Of all group 1 neuraminidase sequences, 377 sequences (59.8%) and of all group 2 neuraminidase sequences, 532 sequences (60.8%) have annotation to their DSB. For 597 NA sequences (39.6%) (254 group 1 neuraminidases (40.2%) and 343 group 2 neuraminidases (39.2%)) no information on the DSB positions could be acquired. The number of DSB varies greatly in the dataset, but it is suspected that this is not due to a different quantity in neuraminidase in general, but rather due to the conducted experiments for identification of disulfide bridges. Mostly, chemical and crystallographic analyses are used to extract information on DSB, which is always dependent on the experiment and the scientific question (Colman, Varghese, et al., 1983; Krug, 1989). Unfortunately, this is why class affiliation cannot be deduced from the frequencies of disulfide bridges. Thus, the number of DSB varies from one to nine disulfide bridges per sequence for this dataset, as seen in Table 19.

Table 19: Number of disulfide bridges per sequence

number of DSB	1	2	3	4	5	6	7	8	9	in total
number of sequences	1	2	114	2	312	230	79	128	41	909

According to literature, neuraminidase possesses eight conserved DSB and one additional in the NA subtypes N2, N7 and N9 (Krug, 1989). Moreover, Asian IAV N2 neuraminidases possess 19 cysteines in the NA head domain alone. This can be confirmed by the number of cysteines in group 2 neuraminidases, as they possess in median one more cysteine for DSB formation (see Table 17). Figure 31 illustrates the DSB of N2 neuraminidase, with the propeller structure of the protein being shown in blue, the protein sequence in black and the disulfide bridges as orange lines. The respective cysteines are shown as orange dots with their sequence position in N2 indicated. Of these positions, the DSB regions 230–237 and 278–291 are close to functional residues (Colman and Ward, 1985), which shows the importance of those DSB. Mutations of the cysteines would result in altered or lost protein structure and/or function (Basler et al., 1999).

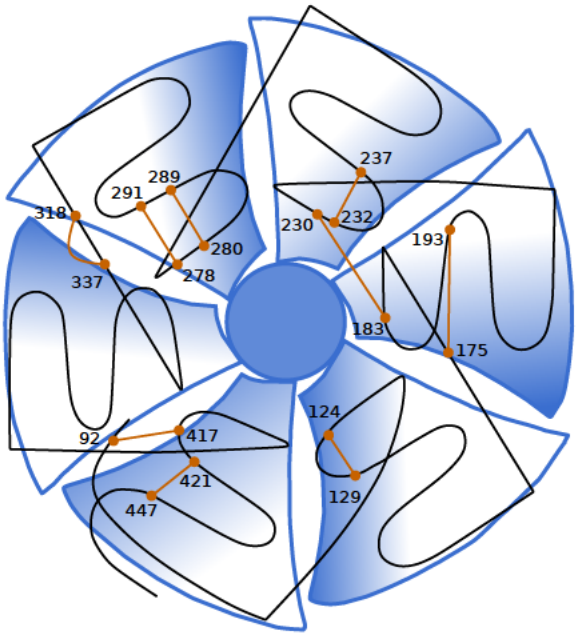


Figure 31: Schematic representation of disulfide bridges in N2 neuraminidase (adapted from Colman and Ward (1985)). The schematic protein structure (propeller) is shown in blue and the sequence is represented as black line. The disulfide bridges are shown as orange lines linking the respective cysteines as orange dots. Their sequence position is indicated.

As demonstrated, the amino acid cysteine is highly conserved over all neuraminidase sequences in the dataset, which leads to the conclusions that, foremost, it is indispensable to maintain the structure and function of the protein due to its ability of forming disulfide bridges. Ultimately, due to varying quantities of cysteines per NA subtype, cysteine frequency can be used to discriminated between NA groups by GMLVQ classification. Nevertheless, it is not possible to deduced class affiliation from the frequencies of disulfide bridges, as the information on those are incomplete. To identify the residues important in classification of each NA subtype in the respective group, a logistic regression will be applied.

6.6.2 Analysis of Tryptophan Occurrence

Due to the results of the GMLVQ models discussed in Chapter 6.4, the amino acid occurrence of tryptophan in NA group 2 will subsequently be analyzed in more detail. Tryptophan is the largest and least abundant of all 20 amino acids. The binuclear ring structure and the associated hydrophobicity result in this amino acid being strategically located in the protein structure (Barik, 2020). For these reasons, among others, it accounts for only 1.1% of the amino acids in proteins on average (Bruice, 2004). In the case of neuraminidase, this would account for roughly four to five tryptophan in the whole protein. Astonishingly, the median absolute frequency of this amino acid in the

whole dataset is 12 tryptophans per sequences, which accounts for approx. 3.09% of amino acids per sequence. It should be noted, that the dataset in use is composed of only the head domain sequence of the NA protein, and therefore these values are only valid for this specific sequence length.
The tryptophans were located by generating a sequence logo using *WebLogo* (Crooks et al., 2004). The sequence logo of the neuraminidase sequences is visualized in Figure 32, with the tryptophan (T) highlighted in orange (see Figure A.39 for larger visualization).

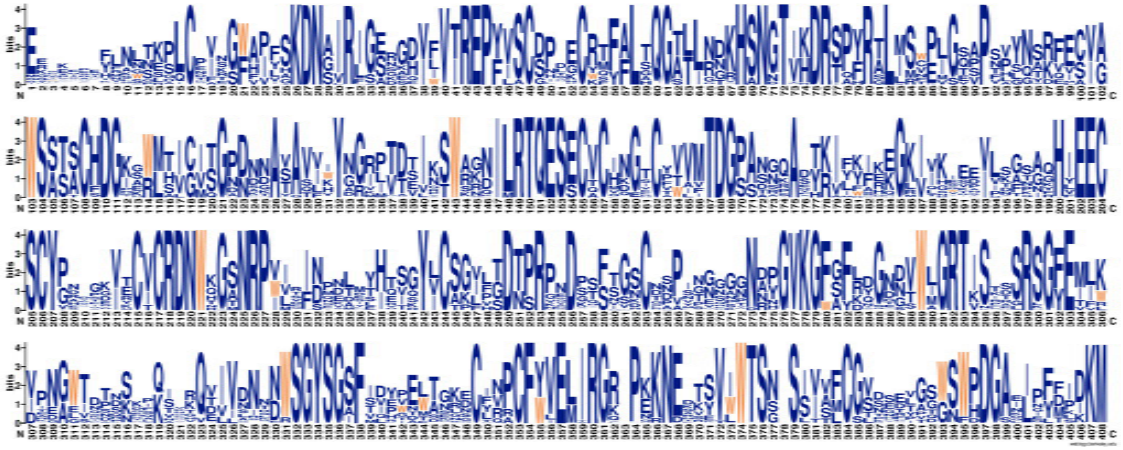


Figure 32: Sequence logo of neuraminidase dataset, with tryptophan residues highlighted in orange.

In total, six tryptophans are highly conserved, which is represented by the height of T. Further six are moderately conserved throughout the NA sequences and no conservation of tryptophan can be seen at 14 positions. In total, 12 tryptophans per sequence occurred in 627 out of 1506 sequences, which is approximately 41.63% of the dataset. Table 20 shows the absolute frequencies of tryptophan in sequences of group 2 subtypes. In median, this amino acid occurs 14 times in a neuraminidase N9 sequence, but only 12 times in N2, N3 and N6 sequences. N7 is the subtype with the least abundance of tryptophan per sequence, with in median only nine of this residue.

Since these findings are not sufficient to say unequivocally that the individual group 2 subtypes can be classified on the basis of tryptophan frequency alone, further features must be considered. Especially N2, N3 and N6 can hardly be differentiated solely based on the median frequency of tryptophan. The consideration of tyrosine, for example, is suitable for this purpose. In the Λ matrix in Figure 28b, tyrosine has a particularly high relevance compared to the other features, except tryptophan. In this case, both features might be decisive in the classification process without any feature, but the automation of every possible feature combination facilitates this task. Therefore, to identify the most important residues in classification of each NA subtype in the respective group, a logistic regression could be applied as described in Chapter 7 in future.

Table 20: Absolute frequencies of tryptophan in neuraminidase sequences per NA group and per NA subtype

	group 2				
	N2	N3	N6	N7	N9
median tryptophan occurrence	12	12	12	9	14
max. tryptophan occurrence	13	13	12	10	15
min. tryptophan occurrence	6	8	6	5	8

6.7 Final Thoughts and Discussion

Concluding, the individual GMLVQ models showed that it is crucial for a well performing model to have a balanced dataset according to the classification task. In three cases the dataset was very imbalanced leading to poor accuracies or, even worse, no interpretable Λ matrix at all. To circumvent this problematic, it might be necessary to either duplicate data in the training or to reduce classes with lots of datapoints. In four other cases the GMLVQ models performed adequately due to balanced class representation according to the individual classification task. For classification by groups, subtypes and by the subtypes in group 1, one of the decisive features is the frequency of the amino acid cysteine. This was validated by analysis of the cysteine occurrences, with on average one or two cysteines more in group 2 than in group 1. Cysteines are responsible for the formation of disulfide bridges, which stabilize the overall protein structure (Wiedemann et al., 2020). However, for the classification by subtypes in group 2, the absolute frequency of tryptophan was one of the decisive features. This specific residue is one of the framework residues and occurs relatively often and conserved in the neuraminidase dataset, although it is the rarest and largest amino acid (Barik, 2020). Both residues are fairly important for protein structure stabilization and moreover, for the structure of the active site. Interestingly the classification shows that their frequencies are one of the decisive features for class separation. To identify all relevant feature combinations for group 1 subtype separation, a logistic regression was subsequently applied.

7 Logistic Regression Modelling for GMLVQ Model Interpretation

As the GMLVQ Λ matrix showed strong positive correlations for the frequencies of the respective amino acids, only the first 20 dimensions of the natural vectors containing those were used in the logistic regression analysis. For this purpose, every NA subtype of group 1 gets discriminated against all other NA subtypes of this NA group. The results were then analyzed to identify the amino acids relevant for classification of each subtype.

At first, a logistic regression was performed with labeled data. For group 1, four logistic regression models were trained, and to obtain binary classification models, the labels were set according to the classification task. For example, the first task was the logistic regression of N1 and all other NA subtypes of group 1, so the class labels were set to *is N1* and *not N1*. The NA subtype labels were changed for every other classification task accordingly. All group 1 subtypes seem to be linear separable, as seen in the confusion matrices in Table 21. Furthermore, the logistic regression algorithm did not converge, as adjusted probabilities with numerical value 0 or 1 occurred. In the following, the logistic regression approach is exemplified by N1 classification.

Table 21: Confusion matrix of logistic regression results of group 1 NA subtypes

	is N1	not N1	$\sum$ true
is N1	211	0	211
not N1	0	420	420
$\sum$ predicted	211	420	631

(a) Confusion matrix of classes *is N1* and *not N1*

	is N4	not N4	$\sum$ true
is N4	111	1	112
not N4	0	519	519
$\sum$ predicted	111	520	631

(b) Confusion matrix of classes *is N4* and *not N4*

	is N5	not N5	$\sum$ true
is N5	123	0	123
not N5	0	508	508
$\sum$ predicted	123	508	631

(c) Confusion matrix of classes *is N5* and *not N5*

	is N8	not N8	$\sum$ true
is N8	185	0	185
not N8	0	446	446
$\sum$ predicted	185	446	631

(d) Confusion matrix classes *is N8* and *not N8*

To determine the minimal feature combination for linear separability, the data was labeled according to group affiliation, with natural vectors belonging to N1 subtype labeled as *is N1* and all others as *not N1*. With a total of 20 amino acids, 2^{20} possible feature combinations of amino acid frequencies were tested. This was done by iterating over all feature combination possibilities. For each combination, the amino acid frequencies are procured, a logistic regression model computed and the MCC calculated, with a computation time of approx. four hours. The combination with the least number of features and with the MCC equal of 1 was then determined.

The goal was to determine, which amino acid frequencies or combinations thereof enable linear separability. As seen in the Λ matrices, cysteines play an important role, but only the combination of multiple features enables class separability. In total, 15 possible combinations of nine amino acid frequencies for linear separability were identified, and none of the 15 combination possibilities is better than the others. All combinations are listed in Table 22.

Table 22: Determination of amino acid combinations for linear separability of N1 to other group 1 NA subtypes. The combinations are numbered consecutively, the amino acids are written in single letter abbreviation.

Combination	Amino Acids								
Combination 1	C	E	G	H	L	Q	T	V	Y
Combination 2	D	G	H	L	Q	R	S	T	V
Combination 3	G	H	K	L	P	Q	S	T	V
Combination 4	E	G	H	K	L	Q	S	T	V
Combination 5	D	G	H	K	L	Q	S	T	V
Combination 6	C	G	H	K	L	Q	S	T	V
Combination 7	C	G	H	I	L	Q	R	T	V
Combination 8	C	E	G	H	L	Q	R	T	V
Combination 9	C	D	F	H	L	Q	R	T	V
Combination 10	C	F	G	H	K	L	Q	T	V
Combination 11	C	E	G	H	K	L	Q	T	V
Combination 12	C	D	F	H	L	Q	R	S	V
Combination 13	E	G	H	K	L	P	Q	S	V
Combination 14	C	G	H	K	L	P	Q	S	V
Combination 15	D	F	H	K	L	N	Q	S	V

As discussed in Ch. 6.4, one of the discriminant features for classification of NA subtypes in group 1 is the frequency of cysteines. These are also found in the possible combinations of nine amino acids imperative for linear separation. Cysteine (C) is represented in nine out of 15 combinations. The amino acids hystidine (H), leucine (L), glutamine (Q) and valine (V) are found in every one of the 15 combinations, followed by glycine (G) in 12 and threonine (T) in 11 combinations. Isoleucine (I) is only represented in combination 7, asparagine (N) in combination 15 and tyrosine (Y) in combination 1. Further, in the Λ matrix Fig. 28a, the frequency of amino acid Q, n_Q, seems also to weigh into the classification decision. The amino acid Q is also represented in every possible combination. The amino acids alanine, methionine and tryptophan are not represented at all in the possible feature combinations. However, it should be noted that the presence of a single amino acid in the 15 combinations is not necessarily evidence of its relevance. Only in combination with other amino acids, they support the separation task.

In conclusion, this analysis illustrates that not all 60 dimensions of natural vectors are needed to achieve linear class separability.

8 Conclusion and Outlook

Influenza A viruses have been and still are responsible for seasonal outbreaks and pandemics among humans and different species of animals. The surface protein neuraminidase of this virus is responsible, among other things, for the release of virions from the cell (Air, 2012; McAuley et al., 2019). Neuraminidase is subject to mutation in its sequences, which can facilitate the adaptation to specific hosts, enhance viral replication, or reduce the effects of vaccination. Thus, it is of interest in pharmacological research (Wohlbold et al., 2014; Krammer et al., 2019; Creytens et al., 2021). Due to these sequence variations, the aim of this work was to gain knowledge about evolutionary changes in sequences of influenza A neuraminidase through different methods. Firstly, *EVcouplings* is used with the goal of identifying evolutionary couplings within the protein sequences, but this analysis was unsuccessful. The evolutionary couplings would have given insights into coevolving amino acids in neuraminidase over a certain period of time (Hopf, Green, et al., 2018). It is probable, that the alignment length is too great for coupling score calculations.
Secondly, the natural vector method is used for sequence embedding purposes to visualize sequential progression and relationships between sequences of the virus protein over time. While the embedded sequences could not illustrate the evolution of all neuraminidase subtypes in the visualizations, they showed that separability per NA group or NA subtype could be possible. Lastly, interpretable machine learning methods are applied to examine if the data is nevertheless classifiable by the different years. Additionally to using the class label *year* and as result from sequence embedding visualizations, other labels such as *groups* or *subtypes* are used in classification with varying outcomes. For a balanced dataset, the classes seem to be well separable, but this was not the case for imbalanced data. Groups and subtypes can be classified with a high accuracy, which was not the case for the years, continents or hosts.
To explain the high accuracy, the visualized Λ matrices are used to determine the decisive features of the classifications. Interestingly, only the first 20 dimensions of the natural vectors seem to be important. As these represent the absolute frequencies of the amino acids in the protein sequences, it was concluded that firstly, in the case of NA these features are the only ones needed to successfully accomplish the classification task. Secondly, the frequency of cysteines (or in one case tryptophan) per sequence plays an important role. To biologically explain these results, the amino acids and their particularities were examined more closely. This lead to the discovery, that group 1 sequences have in median one to two cysteines less than group 2 sequences. Cysteines play an important role in the stability of neuraminidase (Basler et al., 1999) and therefore, they are highly conserved over all NA subtypes, with differences in the quantity of cysteines over the individual NA groups. The absolute frequency of cysteines seemed to be an important feature for achieving class separability in three cases, while the GM-LVQ model trained with data of class label *group 2 subtypes* showed that the quantity of

tryptophan is important. Tryptophan, the least abundant and largest of the amino acids (Bruice, 2004; Barik, 2020), occurs 3x more often in the neuraminidase head domain than usually on average in other proteins. Interestingly, in median 12 tryptophan per sequence were identified, with six of those being highly conserved. The NA subtype N7 has the lowest and N9 the highest quantity of the residue in median, but N2, N3 and N6 cannot be discriminated only by their frequency of tryptophan. This illustrates, that for the classification more than one feature is needed, which in this case could be also the frequency of the amino acid tyrosine.
To identify the minimal number of features necessary for linear separation (in this case of neuraminidase group 1 subtypes), a logistic regression was performed with 20 dimensional natural vectors. Fascinatingly, only nine features are necessary to achieve linear separation of the NA subtypes in group 1, showing that, similar to the GMLVQ results, not all 60 dimensions are necessary for class separation. In total, 15 combinations were identified. Overall, using the natural vectors as feature vectors in ML methods has the advantage of very interpretable results. ML models give an insight into the most relevant amino acids to differentiate between protein sequences, which characteristics can then be analyzed in more detail.
Since the sequence embedding as well as the machine learning methods did not show neuraminidase evolution over time, further research is necessary, for example with focus on one subtype with balanced data. Notably, interpretable ML approaches require a balanced dataset for efficient training and reliable results. Especially the hosts can probably be classified, if the dataset is balanced. This could be achieved by reducing the high quantity of avian sequences. Moreover, only the avian sequences could be classified by continent of origin, if only North America, Asia and eventually Europe are considered as class labels. These three classes are those with the most sequences in the dataset and moreover, are the continents through which migratory flyways pass. Results could give insight into the spreading of IAV globally across bird species, and if the years are taken into account, knowledge could be gained of the viral mutations or reemerging IAV subtypes. Even though the identification of neuraminidase evolution was not successful, neither by sequence embedding nor by ML methods, this was probably due to class imbalance. It could be analyzed, if splitting the dataset into the individual groups or subtypes, and classifying these by years does generate better and interpretable results. Likewise, in addition to year label, the NA subtypes could be stated as additional feature, in hopes of classification success. Overall, the classification performance may be improved by using additional features, but it is assumed that improving the central moments of natural vector does not, as only the first 20 dimensions seemed relevant in classification and linear separation was achievable with only nine features. Additionally, the complete neuraminidase protein should be investigated, because species specific adaptations and mutation can occur for example also in the stalk domain (Y. Li, Chen, et al., 2014). Moreover, the whole protein sequence characterizes the NA subtypes more distinctively. The other domains of the protein may have additional characteristics that were not considered here, but may be specific of a particular subtype to individual years. Transforming neuraminidases of nearly the same length of 469 amino acids has the ad-

vantage, that the natural vectors extracted through the NV methods are more reliable. If other feature generation methods like *Bag of Words* are considered, it is advisable to consider the neuraminidase genome instead of proteome. This could also give insight into mutations on genome level, which might not affect the phenotype.
Concluding, linear separation of neuraminidase sequences is possible for balanced data. Through the visualization capabilities of GMLVQ Λ matrix paired with natural vectors, the results can be easily grasped, analyzed and interpreted by a wide range of scientists. Even though the viral progression over time could not be analyzed and made visible, it could be possible to achieve this by elaborate class balancing. At last, the trained GMLVQ models can be used to identify the groups or subtypes of neuraminidases of unknown origin.

Appendix A: Appendix

A.1 Overview od preprocessed Dataset

Table A.23: Overview initial PDB dataset

year	number of sequences	Influenza A subtype	continent	host
unknown	5	-	-	-
1918	4	H1N1	North America	human
1956	8 5 3	H1N6 H11N6	Europe	duck
1957	10 9 1	H2N2	North America Asia	human
1963	12	H3N8	Europe	duck
1967	12 9 3	H2N2 HxN2	Asia	human
1975	40 33 4 3	H11N9 H1N9 HxN9	Australia	tern
1976	4	H12N5	North America	mallard
1984	3 2 1	H13N9 H1N9	North America	whale
1996	2	H10N7	Europe	mallard
1998	12 10 2	H1N9 H3N2	Australia North America	tern human
2000	2	H3N6	Asia	chicken
2004	3	H5N1	Asia	human
2006	14 6 5 2 1	H2N3 HxN8 HxN1 HxN4	North America unknown unknown unknown	pig
2009	19 17 2	H1N1 H3N2	North America	human human
2010	11 8 3	H3N2 H1N1	Africa, North America Europe	human human
2011	3	H3N8	North America	harbor seal
2013	15	H7N9	Asia	human
2014	4 1 1 1 1	H5N1 H5N2 H5N8 H5N6	North America North America North America Asia	green-winged teal pintail duck gyrfalcon chicken
2015	1	H3N2	North America	canine
in total	184	23	5	12

Table A.24: Overview initial NCBI dataset

influenza subtype	number of sequences	influenza subtype	number of sequences	influenza subtype	number of sequences	influenza subtype	number of sequences
H1N1	11180	H13N2	61	H13N9	14	H2N6	4
H3N2	10590	HXN1	58	H11N6	13	H5N4	4
H5N1	2764	HXN2	58	H4N9	13	H6N7	4
H1N2	2731	H10N8	55	H11N8	12	H7N5	4
H9N2	2541	H6N5	55	H1N5	12	H9N6	4
H3N8	904	H10N1	44	H2N8	11	H9N8	4
H4N6	676	H1N3	41	H11N7	10	H13N3	3
H5N2	673	H10N4	38	H4N7	10	H14N5	3
H6N2	597	H2N9	38	H5N7	10	H14N6	3
H6N1	383	H11N1	36	HXN6	10	HXN4	3
H10N7	359	H10N2	35	H3N7	9	H12N9	2
H7N3	290	H11N3	34	H4N4	9	H14N7	2
H7N2	272	H2N1	32	H6N9	9	H14N8	2
H11N9	263	H10N6	27	H12N4	8	H15N8	2
H7N9	261	H7N6	27	H6N4	8	H17N10	2
H7N7	240	H13N8	24	H9N5	8	H18N11	2
H6N6	208	H7N4	23	H9N7	8	H3N4	2
H2N2	168	H4N3	22	unknown	7	H8N8	2
H3N6	167	H9N1	22	H3N9	7	H9N4	2
H6N8	147	H1N9	21	H9N9	7	HXN7	2
H2N3	145	H2N7	21	H12N2	6	HXN9	2
H4N8	140	H3N3	21	H1N7	6	H11N4	1
H7N1	130	H1N8	19	H6N3	6	H13N1	1
H4N2	124	H5N9	18	H9N3	6	H14N2	1
H11N2	118	H7N8	18	HXN8	6	H14N4	1
H5N6	118	H10N9	17	H11N5	5	H15N7	1
H12N5	110	H4N5	17	H12N7	5	H16N9	1
H5N3	103	H5N5	17	H12N8	5	H8N1	1
H3N1	101	H12N1	16	H1N4	5	H8N2	1
H5N8	88	H1N6	16	H12N3	4	H8N3	1
H8N4	84	H2N5	16	H12N6	4	H8N5	1
H16N3	71	H3N5	16	H14N3	4	H8N7	1
H13N6	70	H4N1	16	H15N9	4	HXN3	1
H10N3	69	H10N5	15	H2N4	4	HXN5	1

Table A.25: Number of sequences per neuraminidase subtypes in NCBI sequences

NA Subtypes	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11
number of sequences	14784	17976	821	192	280	1347	688	1439	677	2	2

A.2 PDB IDs of PDB sequences

1A14, 1ING, 1INY, 1L7G, 1L7H, 1NCA, 1NCB, 1NCC, 1NCD, 1NMA, 1NMB, 1NNA, 1V0Z, 2AEP, 2B8H, 2BAT, 2HT5, 3CKZ, 3CL2, 3NSS, 3SAL, 3TIA, 4B7J, 4B7Q, 4D8S, 4GZO, 4GZS, 4H53, 4HZV, 4HZY, 4HZZ, 4K1J, 4KS1, 4M3M, 4MJU, 4MJV, 4MWJ, 4MWL, 4NN9, 4QN3, 4QN4, 4WA3, 4WA4, 5HUG, 5HUK, 5HUM, 5HUN, 5NN9, 5NWE, 5NZ4, 5NZE, 5NZF, 5NZN, 6BR5, 6CRD, 6D96, 6N4D, 6N6B, 6NN9, 6Q20
It was renounced to enumerate all 1446 IDs of NCBI sequences.

A.3 General System Information and Performance Specifications

For EVcouplings and *plmc*:
Desktop-PC with Linux Mint 20.1
Intel Core i7-3770 CPU @ 3.40 GHz
16 GB RAM

For GMLVQ and general data analysis:
Desktop-PC with Microsoft Windows 10 Pro
Intel Core i5-4670 CPU @ 3.40 GHz
8 GB RAM

A.4 Parameter Settings

Table A.26: Neural Gas parameter settings

name of hyperparameter	setting
prototype initialization	200
neighborhood range	20
neighborhood range reduction	0.5
learn rate	0.1
maximum iteration	number of rows of each subdataset $\cdot$ 1000

Table A.27: GMLVQ parameter settings

name of hyperparameter	setting
input dimension	60
number of folds	5
prototype per class	1
prototype initializer	SMCI
prototype learn rate	0.01
matrix learn rate	0.001
stochastic optimization	Adam
latent dimension	2
maximum epochs	1000

Table A.28: EVcouplings Website parameter settings

	Evcoupling Stages				
	Alignment		Evolutionary couplings	Folding	Result evaluation
	Homology search	Sequence and position filters			
Parameters	Alignment threshold type: Bitscore	Position filter: 70%	Statistical inference method: Pseudo-likelihood maximization	3D structure prediction from ECs: Enabled	Contact distance cutoff: 5.0Å
	Search iterations: 5	Removing similar sequences: 90%		Number of generated models: 10	PDB structure search method: Conservative
	Sequence database: UniRef90	Downweighting similar sequences: 80%			

A.5 EVcouplings additional result

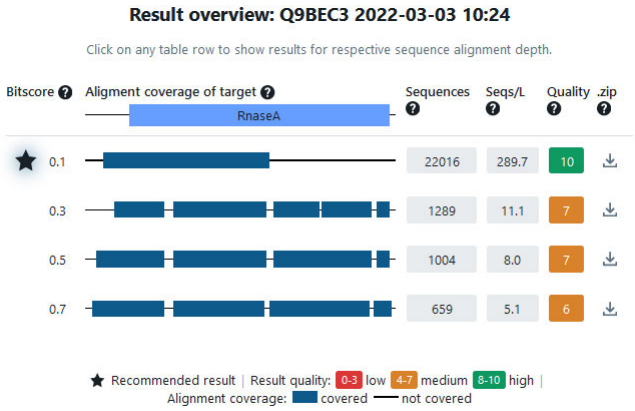


Figure A.33: Overview of results from EVcouplings Website with RNase A.

A.6 GMLVQ additional results

Table A.29: Fold accuracy of classification by host

folds	fold 1	fold 2	fold 3	fold 4	fold 5
accuracy	7.28%	26.58%	26.25%	19.93%	11.96%

Table A.30: The Confusion Metrics Precision, Sensitivity and Specificity of every class in neuraminidase groups in percent

Class	Precision	Sensitivity	Specificity
N1	99.53%	100.00%	99.85%
N4	100.00%	100.00%	100.00%
N5	100.00%	100.00%	100.00%
N8	100.00%	100.00%	100.00%

(a) Confusion Metrics of NA subtype in group 1

Class	Precision	Sensitivity	Specificity
N2	99.53%	100.00%	99.85%
N3	100.00%	100.00%	100.00%
N6	100.00%	100.00%	100.00%
N7	100.00%	100.00%	100.00%
N9	100.00%	99.38%	100.00%

(b) Confusion Metrics of NA subtype in group 2

A.6.1 Lambda Matrices

Figure A.34: Visualized Λ Matrix of classification by *groups*. he color gradient from blue to red indicates negative to positive values. Blue values denote negative correlation, while red values denote positive correlation.

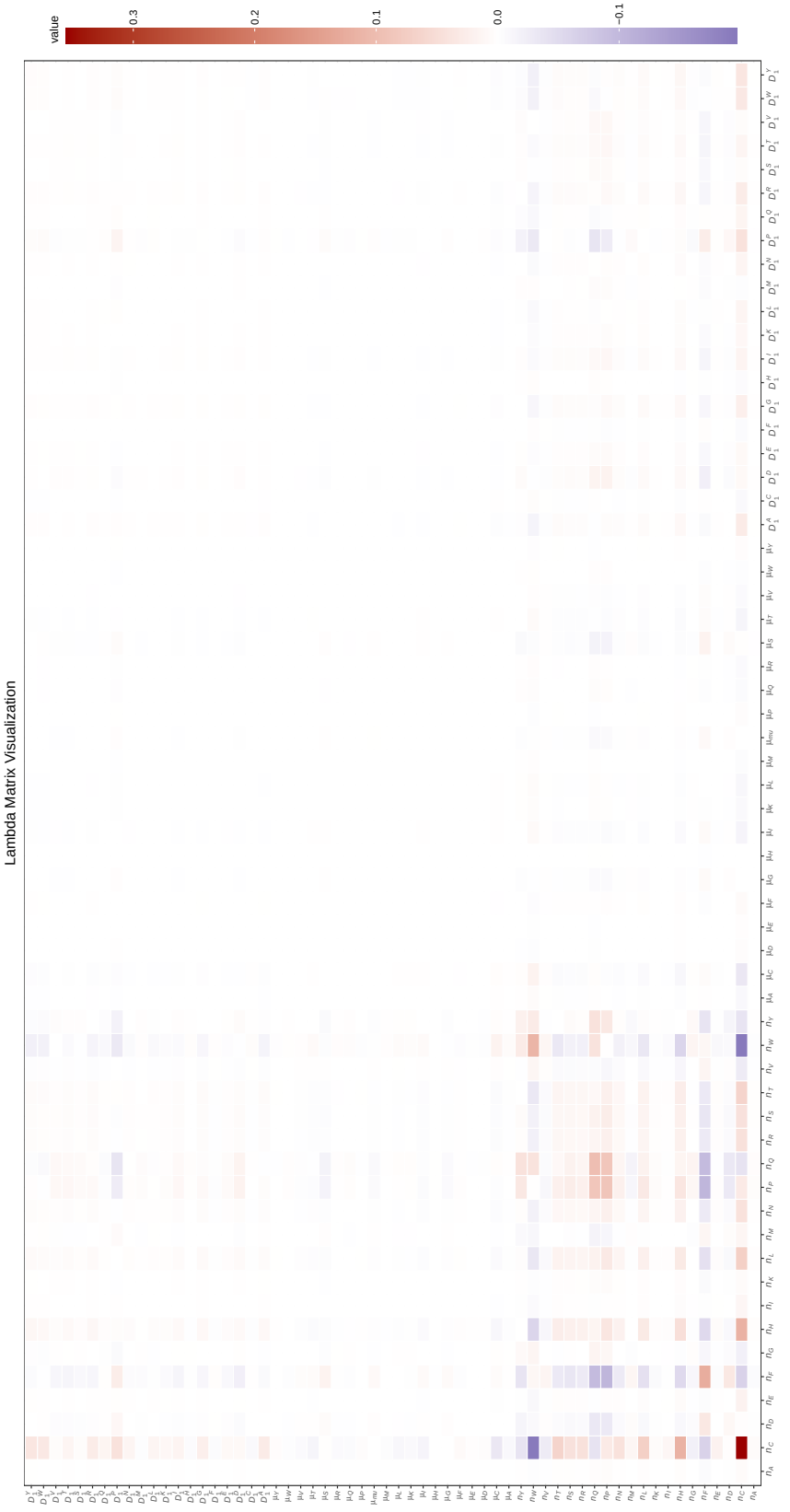


Figure A.35: Visualized Λ Matrix of classification by *subtype*. The color gradient from blue to red indicates negative to positive values. Blue values denote negative correlation, while red values denote positive correlation.

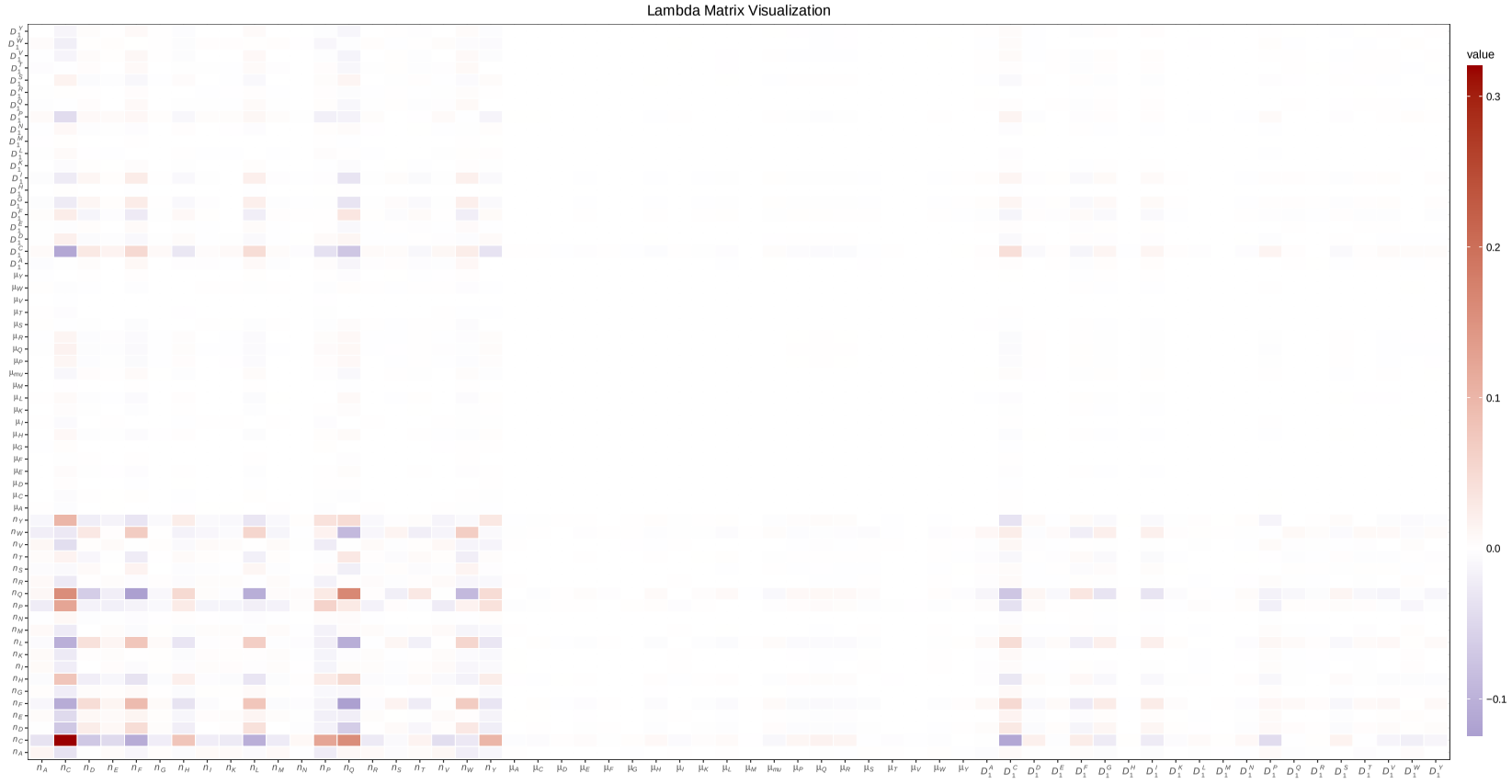


Figure A.36: Visualized Λ Matrix of classification by *group 1 subtypes*. The color gradient from blue to red indicates negative to positive values. Blue values denote negative correlation, while red values denote positive correlation.

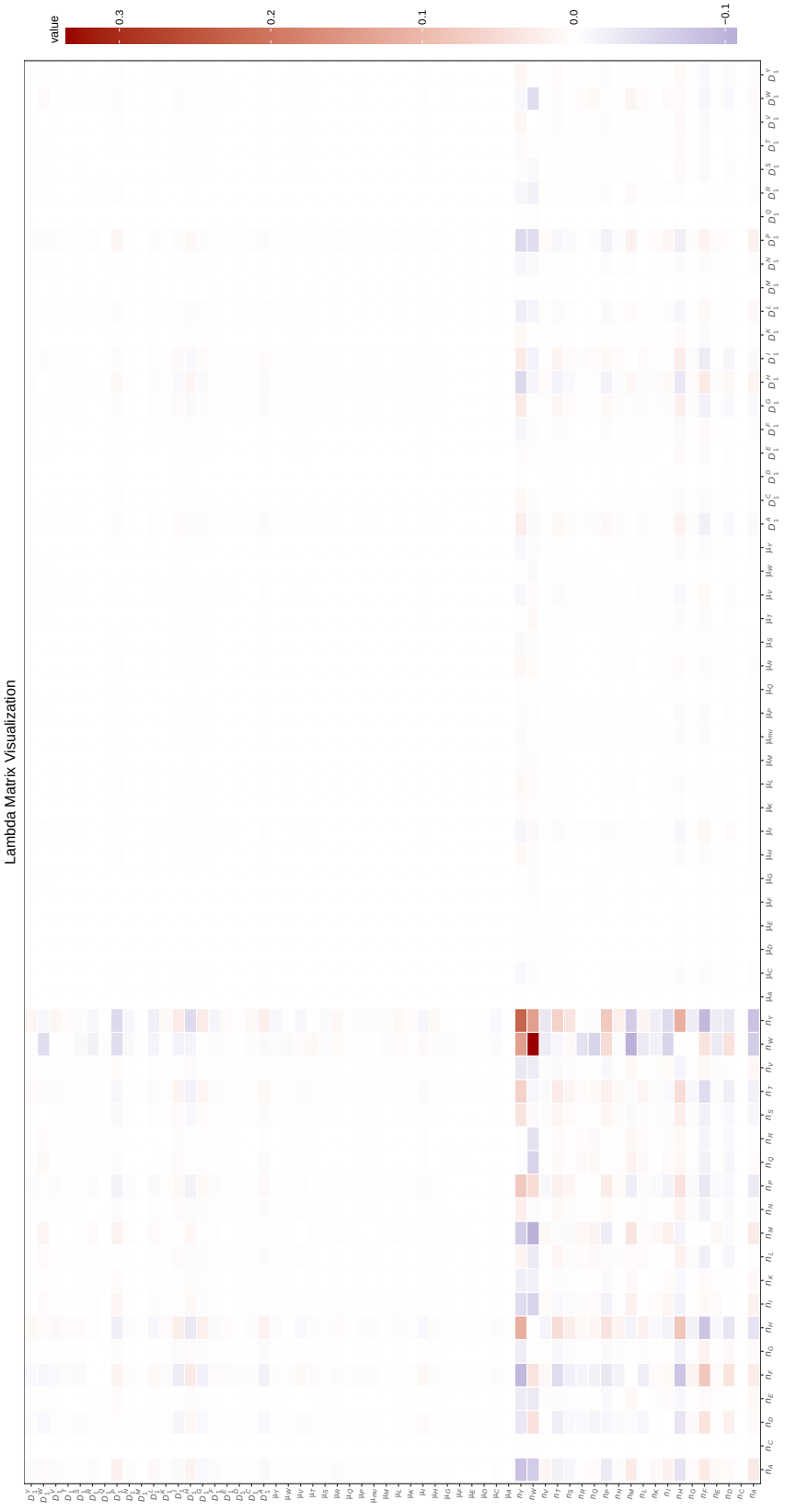


Figure A.37: Visualized Λ Matrix of classification by *group 2 subtypes*. The color gradient from blue to red indicates negative to positive values. Blue values denote negative correlation, while red values denote positive correlation.

A.6.2 Sequence Logos

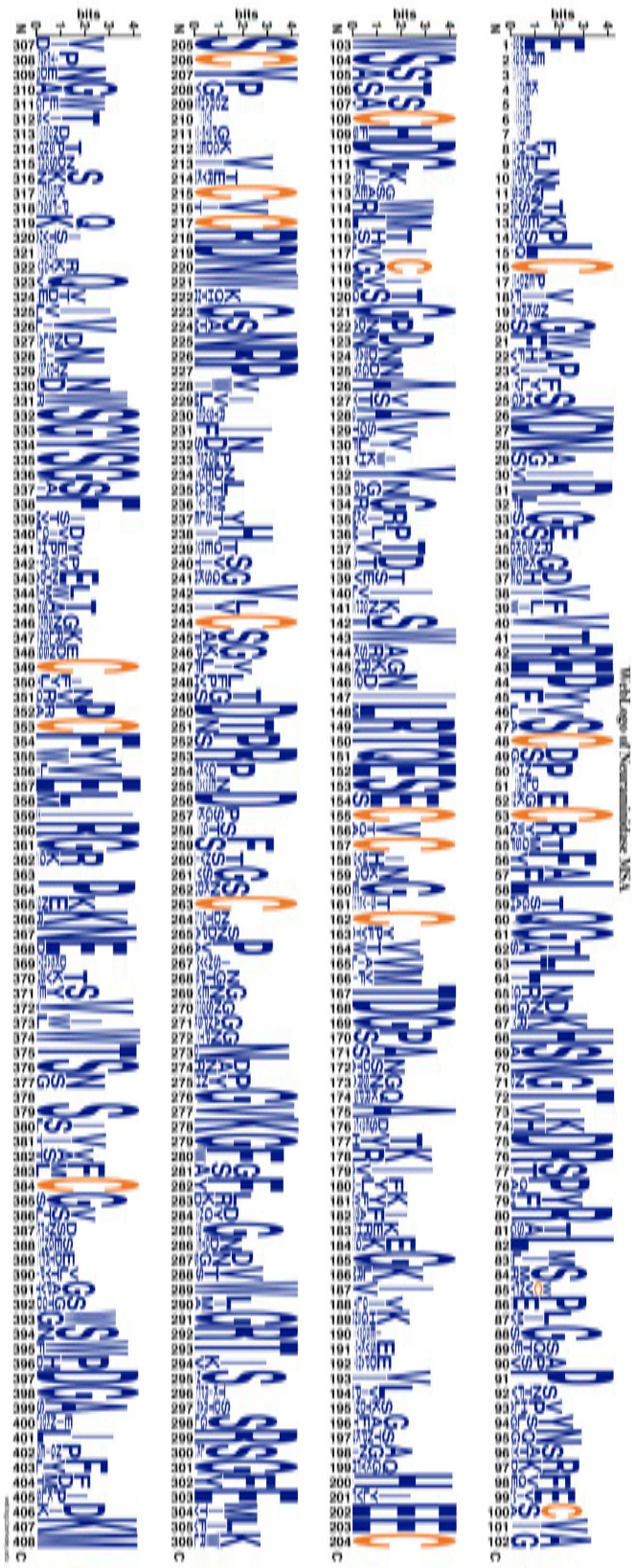


Figure A.38: Sequence logo of neuraminidase dataset, with cysteine residues highlighted in orange.

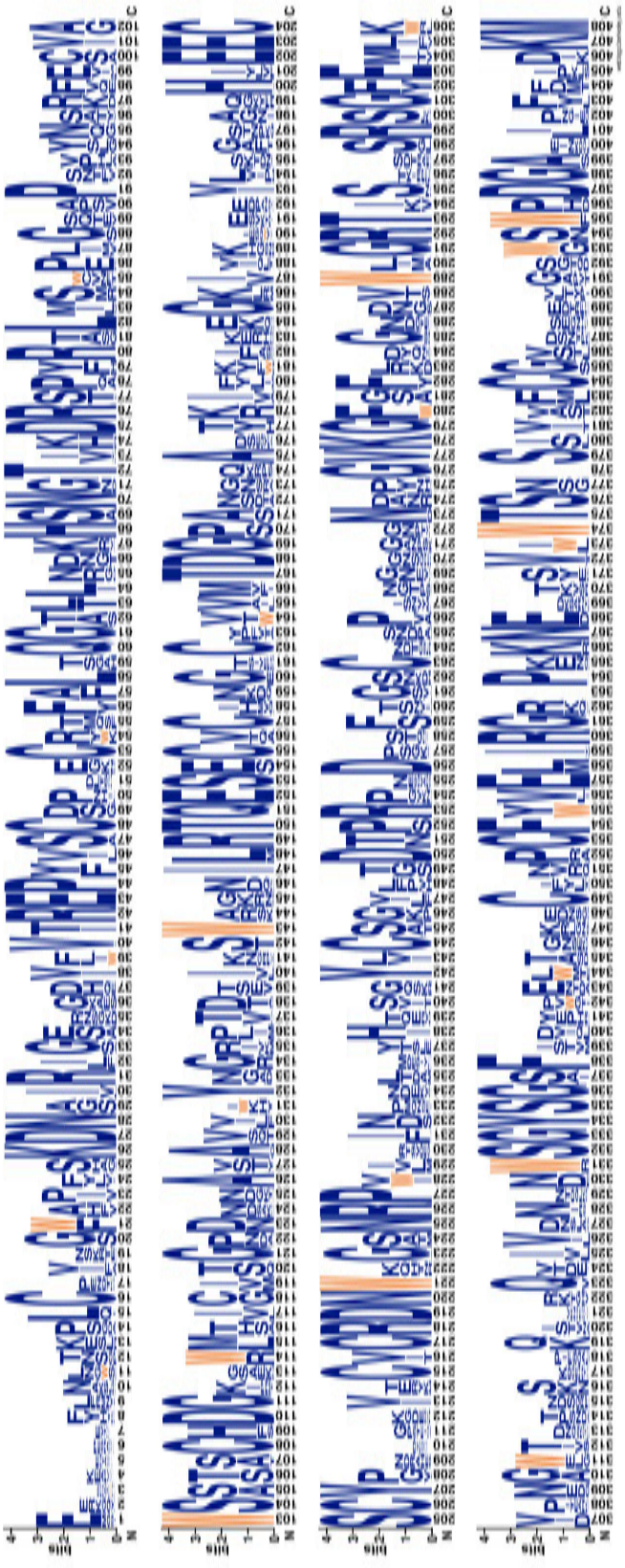


Figure A.39: Sequence logo of neuraminidase dataset, with tryptophan residues highlighted in orange.

A.7 Poster

Interpretation of a GMLVQ Λ Matrix: Explaining Classification Results from a Biological Perspective

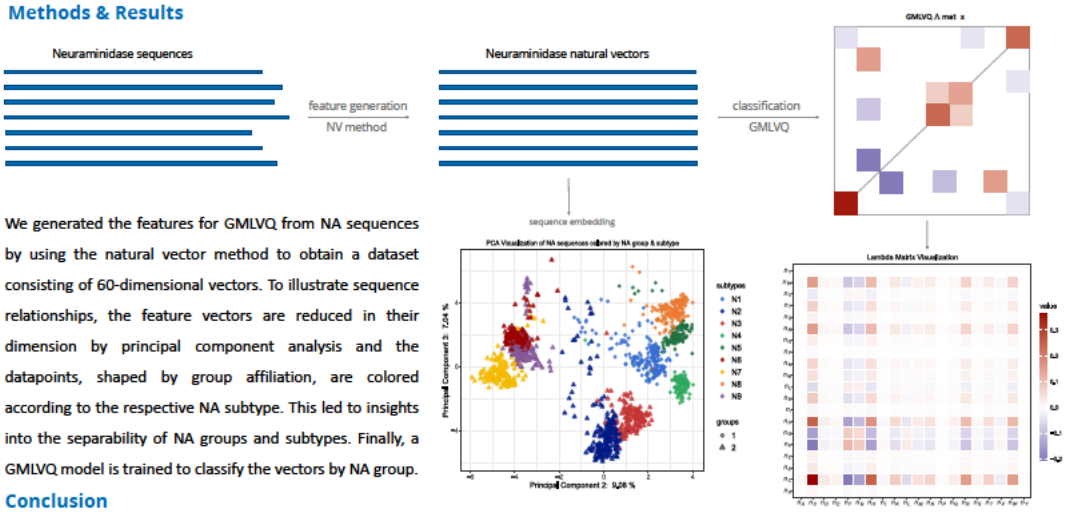
HOCHSCHULE MITTWEIDA
University of Applied Sciences

Lynn Vivian Reuss, Florian Heinke & Thomas Villmann

Introduction

Influenza A viruses (IAV) are responsible for the outbreak of epidemics and pandemics worldwide. The surface protein neuraminidase (NA) of IAV is categorized into eleven neuraminidase subtypes, which are divided into three phylogenic groups: group 1 consists of the subtypes N1, N4, N5 and N8. Group 2 includes the subtypes N2, N3, N6, N7 and N9, and group 3 is composed of N10 and N11. Only the first to groups are used in this work. Apart from group 3, all NA subtypes are predominant in avian hosts with seasonal outbreaks of IAV among humans and their livestock and/or poultry [1]. The function of NA is mostly the release of virions from the host cell and thus, this protein is of interest in pharmacological research [2]. Here, we analyze the sequences of the neuraminidase head domain by means of the interpretable machine learning classification method *Generalized Matrix Learning Vector Quantization* (GMLVQ) verified by 5-fold cross validation. For machine learning purposes, we generate numerical data from the biological sequences using the *natural vector method* [3]. The interpretation of the GMLVQ result is accomplished through the GMLVQ Λ matrix.

Methods & Results



We generated the features for GMLVQ from NA sequences by using the natural vector method to obtain a dataset consisting of 60-dimensional vectors. To illustrate sequence relationships, the feature vectors are reduced in their dimension by principal component analysis and the datapoints, shaped by group affiliation, are colored according to the respective NA subtype. This led to insights into the separability of NA groups and subtypes. Finally, a GMLVQ model is trained to classify the vectors by NA group.

Conclusion

With an accuracy for the binary classification by NA group of 99.73%, the data seems linear separable. To explain such a high accuracy, we used the visualized Λ matrix to show which are the decisive features of the classification. Interestingly, only the first 20 dimensions of the natural vectors are important in the classification. These represent the absolute frequencies of the amino acids in the protein sequences. We concluded that firstly, these features are the only ones needed to successfully accomplish the classification task, and secondly, that the frequency of cysteine per sequence is important for classification. On closer inspection, group 1 sequences have in median one to two cysteines less than group 2 sequences. Cysteines play in important role in stabilizing the NA structure by forming disulfide bonds. These findings could facilitate the identification of group affiliation of unknown neuraminidase .

Groups	Group 1				Group 2				
Subtypes	N1	N4	N5	N8	N2	N3	N6	N7	N9
Median cysteine occurrences	17	17	16	16	18	18	18	18	18

References

[1] McAuley J.L. et al. *Frontiers in Microbiology* **2019** 10 26-32
[2] Creytens S. et al. *Frontiers in Microbiology* **2021** 12 1-19
[3] Deng M. et al. *PLoS One* **2011** 83 5155-5159

Hochschule Mittweida
Technikumplatz 17 09648 Mittweida
Lynn Vivian Reuss
lreuss@hs-mittweida.de

Bibliography

Air, G. M. (2012). “Influenza neuraminidase”. In: *Influenza Other Respi. Viruses* 6.4, pp. 245–256. DOI: 10.1111/j.1750-2659.2011.00304.x.

Al Hajjar, S. and McIntosh, K. (2010). “The first influenza pandemic of the 21st century”. In: *Ann. Saudi Med.* 30.1, pp. 1–10. DOI: 10.4103/0256-4947.59365.

Altschul, S. F. et al. (1990). “Basic local alignment search tool”. In: *Journal of molecular biology* 215.3, pp. 403–410. DOI: 10.1016/S0022-2836(05)80360-2.

Amaro, R. E. et al. (2011). “Mechanism of 150-cavity formation in influenza neuraminidase”. In: *Nat. Commun.* 2.1, pp. 387–388. DOI: 10.1038/ncomms1390.

Barik, S. (2020). “The uniqueness of tryptophan in biology: properties, metabolism, interactions and localization in proteins”. In: *International journal of molecular sciences* 21.22, p. 8776. DOI: 10.3390/ijms21228776.

Barry, J. M. (2005). *Influenza - The Story of the Deadliest Pandemic in History*. New York: Penguin Books. ISBN: 978-0-14-303649-4.

Basler, C. F. et al. (1999). “Mutation of Neuraminidase Cysteine Residues Yields Temperature-Sensitive Influenza Viruses”. In: *J. Virol.* 73.10, pp. 8095–8103. DOI: 10.1128/jvi.73.10.8095-8103.1999.

Betts, M. J. and Russell, R. B. (2003). “Amino Acid Properties and Consequences of Substitutions”. In: *Bioinformatics for Geneticists*. John Wiley Sons, Ltd. Chap. 14, pp. 289–316. DOI: 10.1002/0470867302.ch14.

Bittrich, S. et al. (2019). “Application of an interpretable classification model on Early Folding Residues during protein folding”. In: *BioData Min.* 12.1, pp. 1–16. DOI: 10.1186/s13040-018-0188-2.

Blackshields, G. et al. (2010). “Sequence embedding for fast construction of guide trees for multiple sequence alignment”. In: *Algorithms Mol. Biol.* 5.1, pp. 1–11. DOI: 10.1186/1748-7188-5-21.

Blaisdell, B. E. (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”. In: *Journal of molecular evolution* 29.6, pp. 538–547.

Bohnsack, K. S. et al. (2022). “Alignment-free sequence comparison : A systematic survey from a machine learning perspective”. In: *IEEE Trans. Comput. Biol. Bioinforma.* DOI: 10.1109/TCBB.2022.3140873.

Breitbart, M. and Rohwer, F. (June 2005). *Here a virus, there a virus, everywhere the same virus?* DOI: 10.1016/j.tim.2005.04.003. URL: https://pubmed.ncbi.nlm.nih.gov/15936660/.

Bruice, P. Y. (2004). *Organische Chemie: Studieren kompakt*. Pearson Deutschland GmbH.

Chicco, D. and Jurman, G. (2020). “The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation”. In: *BMC genomics* 21.1, pp. 1–13. DOI: 10.1186/s12864-019-6413-7.

Choi, R. Y. et al. (2020). “Introduction to machine learning, neural networks, and deep learning”. In: *Transl. Vis. Sci. Technol.* 9.2, pp. 1–12. DOI: 10.1167/tvst.9.2.14.

Ciminski, K. et al. (2017). “Novel insights into bat Influenza A Viruses”. In: *J. Gen. Virol.* 98.10, pp. 2393–2400. DOI: 10.1099/jgv.0.000927.

Colman, P. M., Varghese, J. N., et al. (1983). “Structure of the catalytic and antigentiv sites in influenza virus neuraminidase”. In: *Nature* 303, pp. 41–44. DOI: 10.1038/303041a0.

Colman, P. M. and Ward, C. W. (1985). “Structure and Diversity of Influenza Virus Neuraminidase”. In: *Curr. Top. Microbiol. Immunol.* 114, pp. 178–255. DOI: 10.1007/978-3-642-70227-3_5.

Creytens, S. et al. (2021). “Influenza Neuraminidase Characteristics and Potential as a Vaccine Target”. In: *Front. Immunol.* 12.November, pp. 1–19. DOI: 10.3389/fimmu.2021.786617.

Crooks, G. E. et al. (2004). “WebLogo: a sequence logo generator”. In: *Genome research* 14.6, pp. 1188–1190. DOI: 10.1101/gr.849004. URL: https://weblogo.berkeley.edu/.

Deng, M. et al. (2011). “A novel method of characterizing genetic sequences: Genome space with biological distance and applications”. In: *PLoS One* 6.3. DOI: 10.1371/journal.pone.0017293.

Dou, D. et al. (2018). “Influenza A virus cell entry, replication, virion assembly and movement”. In: *Frontiers in immunology* 9, p. 1581. DOI: 10.3389/fimmu.2018.01581/full.

Ekeberg, M. et al. (2013). “Improved contact prediction in proteins: Using pseudolikelihoods to infer Potts models”. In: *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.* 87.1, pp. 1–16. DOI: 10.1103/PhysRevE.87.012707.

“Vaccine-Preventable Diseases” (2012). In: *Immunol. Pharm.* Ed. by D. K. Flaherty. St. Louis, Missouri: Mosby, inc., affiliate of Elsevier Inc., pp. 197–213. DOI: 10.1016/B978-0-323-06947-2.10025-2.

Fox, A. and Carolan, L. (2019). “Neuraminidase escape attempts”. In: *Nat. Microbiol.* 4.12, pp. 2031–2032. DOI: 10.1038/s41564-019-0615-2.

Geweniger, T. (2012). “Fuzzy variants of prototype based clustering and classification algorithms”. PhD thesis. Rijksuniversiteit Groningen. ISBN: 9789036757492.

Grandini, M. et al. (2020). “Metrics for multi-class classification: an overview”. In: *ArXiv* abs/2008.05756. DOI: 10.48550/arXiv.2008.05756.

Gupta, M. R. et al. (2014). “Training highly multiclass classifiers”. In: *The Journal of Machine Learning Research* 15.1, pp. 1461–1492. DOI: 10.5555/2627435.2638582.

Hopf, T. A., Green, A. G., et al. (2018). “The EVcouplings Python framework for coevolutionary sequence analysis”. In: *Bioinformatics* 35.9, pp. 1582–1584. DOI: 10.1093/bioinformatics/bty862. URL: https://evcouplings.org/%20and%20https://github.com/debbiemarkslab/EVcouplings.

Hopf, T. A., Ingraham, J. B., et al. (2017). “Mutation effects predicted from sequence co-variation”. In: *Nat. Biotechnol.* 35.2, pp. 128–135. DOI: 10.1038/nbt.3769. URL: https://github.com/debbiemarkslab/plmc.

Hopf, T. A. and Marks, D. S. (2017). “Protein Structures, Interactions and Function from Evolutionary Couplings”. In: *From Protein Structure to Function with Bioinformatics*. Ed. by D. J. Ridgen. 2nd Edition, pp. 37–58. DOI: 10.1007/978-94-024-1069-3.

Hopf, T. A., Schärfe, C. P., et al. (2014). “Sequence co-evolution gives 3D contacts and structures of protein complexes”. In: *eLife* 3.e03430. DOI: 10.7554/eLife.03430.

Kilbourne, E. D. (2006). “Influenza pandemics of the 20th century”. In: *Emerg. Infect. Dis.* 12.1, pp. 9–14. DOI: 10.3201/eid1201.051254.

Kohonen, T. (1986). “Learning Vector Quantization for Pattern Recognition”. In: *Technical Report* TKK-F-A601.Helsinski Universtiy of Technology.

Kohonen, T. (1997). “Learning Vector Quantization”. In: *Self-Organizing Maps*. 2nd. Berlin, Heidelberg: Springer. Chap. 6, pp. 203–217. DOI: 10.1007/978-3-642-97966-8.

Krammer, F. et al. (2019). “Emerging from the Shadow of Hemagglutinin: Neuraminidase Is an Important Target for Influenza Vaccination”. In: *Cell Host Microbe* 26.6, pp. 712–713. DOI: 10.1016/j.chom.2019.11.006.

Krug, R. M. (1989). *The Influenza Viruses*. 1st ed. DOI: 10.1007/978-1-4613-0811-9.

Labella, A. M. and Merel, S. E. (2013). “Influenza”. In: *Med. Clin. North Am.* 97.4, pp. 621–645. DOI: 10.1016/j.mcna.2013.03.001.

Larkin, M. et al. (2007). “Clustal W and Clustal X version 2.0”. In: *Bioinformatics* 23.21, pp. 2947–2948. DOI: 10.1093/bioinformatics/btm404. URL: http://www.clustal.org/clustal2.

Li, Q. et al. (2012). “Structural and functional characterization of neuraminidase-like molecule N10 derived from bat influenza A virus”. In: *Proc. Natl. Acad. Sci. U. S. A.* 109.46, pp. 18897–18902. DOI: 10.1073/pnas.1211037109.

Li, Y., Chen, S., et al. (2014). “A 20-Amino-Acid Deletion in the Neuraminidase Stalk and a Five-Amino-Acid Deletion in the NS1 Protein Both Contribute to the Pathogenicity of H5N1 Avian Influenza Viruses in Mallard Ducks”. In: *PLoS One* 9.4. DOI: 10.1371/journal.pone.0095539.

Li, Y., Tian, K., et al. (2016). “Virus classification in 60-dimensional protein space”. In: *Mol. Phylogenet. Evol.* 99, pp. 53–62. DOI: 10.1016/j.ympev.2016.03.009.

Lina, B. (2008). *History of Influenza Pandemics*. Ed. by D. Raoult and M. Drancourt. Berlin Heidelberg: Springer-Verlag, pp. 199–211. DOI: 10.1007/978-3-540-75855-6_12.

Lisboa, P. et al. (2021). “The coming of age of interpretable and explainable machine learning models”. In: *ESANN 2021 proceedings European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*. ESANN Ciaco.

Lycett, S. J. et al. (2019). “A brief history of bird flu”. In: *Philosophical Transactions of the Royal Society B* 374.1775, p. 20180257. DOI: 10.1098/rstb.2018.0257.

Ma, W. et al. (2009). “The pig as a mixing vessel for influenza viruses: human and veterinary implications”. In: *Journal of molecular and genetic medicine: an international journal of biomedical research* 3.1, p. 158. DOI: 10.4172/1747-0862.1000028.

Marks, D. S., Colwell, L. J., et al. (2011). “Protein 3D Structure Computed from Evolutionary Sequence Variation”. In: *PLOS ONE* 6, pp. 1–20. DOI: 10.1371/journal.pone.0028766.

Marks, D. S., Hopf, T. A., et al. (2012). “Protein structure prediction from sequence variation”. In: 30.11, pp. 1072–1080. DOI: 10.1038/nbt.2419.

Martinetz, T. M. et al. (1993). “"Neural-Gas" Network for Vector Quantization and its Application to Time-Series Prediction”. In: *IEEE Trans. Neural Networks* 4.4, pp. 558–569. DOI: 10.1109/72.238311.

May, A. C. (2004). “Percent sequence identity: the need to be explicit”. In: *Structure* 12.5, pp. 737–738. DOI: 10.1016/j.str.2004.04.001.

McAuley, J. L. et al. (2019). “Influenza virus neuraminidase structure and functions”. In: *Frontiers in Microbiology* 10.JAN, pp. 26–32. DOI: 10.3389/fmicb.2019.00039.

Morens, D. M. et al. (2009). “The Persistent Legacy of the 1918 Influenza Virus”. In: *N Engl J Med.* 361.3, pp. 225–229. DOI: 10.1056/NEJMp0904819.

Newton, J. et al. (2006). “Description of the outbreak of equine influenza (H3N8) in the United Kingdom in 2003, during which recently vaccinated horses in Newmarket developed respiratory disease”. In: *Veterinary Record* 158.6, pp. 185–192. DOI: 10.1136/vr.158.6.1185.

Notredame, C. et al. (2000). “T-coffee: A novel method for fast and accurate multiple sequence alignment”. In: *J. Mol. Biol.* 302.1, pp. 205–217. DOI: 10.1006/jmbi.2000.4042. URL: https://tcoffee.crg.eu/apps/tcoffee/do:expresso.

Olsen, B. et al. (2006). “Global patterns of influenza A virus in wild birds”. In: *Science* 312.5772, pp. 384–388. DOI: 10.1126/science.1122438.

Palese, P. (2004). “Influenza: Old and new threats”. In: *Nat. Med.* 10.12S, S82–S87. DOI: 10.1038/nm1141.

Peng, C. Y. J. et al. (2002). “An introduction to logistic regression analysis and reporting”. In: 96.1, pp. 3–14. DOI: 10.1080/00220670209598786.

Powers, D. M. (2020). “Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation”. In: *arXiv*. DOI: 10.48550/arXiv.2010.16061.

Reche, P. (2021). *SIAS: Sequence identities and similarities*. URL: http://imed.med.ucm.es/Tools/sias.html (visited on 06/18/2021).

Russell, R. J. et al. (2006). “The structure of H5N1 avian influenza neuraminidase suggests new opportunities for drug design”. In: *Nature* 443.7107, pp. 45–49. DOI: 10.1038/nature05114.

Sato, A. and Yamada, K. (1996). “Generalized Learning Vector Quantization”. In: *Advances in neural information processing systems* 8, pp. 423–429.

Schäfer, M. (2019). “Understanding and predicting global change impacts on migratory birds”. PhD thesis. Universität Potsdam.

Schneider, P. et al. (2009). “Adaptive relevance matrices in learning vector quantization”. In: *Neural Comput.* 21.12, pp. 3532–3561. DOI: 10.1162/neco.2009.11-08-908.

Schneider, T. D. and Stephens, R. M. (1990). “Sequence logos: a new way to display consensus sequences”. In: *Nucleic acids research* 18.20, pp. 6097–6100. DOI: 10.1093/nar/18.20.6097.

Shortridge, K. F. (1995). “The next pandemic influenza virus?” In: *Lancet* 346.8984, pp. 1210–1212. DOI: 10.1016/S0140-6736(95)92906-1.

Srinivasa, K. et al. (2020). *Statistical Modelling and Machine Learning Principles for Bioinformatics Techniques, Tools, and Applications*. Springer Nature. DOI: 10.1007/978-981-15-2445-5.

Steinegger, M. et al. (2019). “HH-suite3 for fast remote homology detection and deep protein annotation”. In: *BMC Bioinformatics* 20.473. DOI: 10.1186/s12859-019-3019-7.

Swift, A. et al. (2019). “What are sensitivity and specificity?” In: *Evidence-Based Nursing* 23.1, pp. 2–4. DOI: 10.1136/ebnurs-2019-103225.

Taubenberger, J. K., Morens, D. M., and Fauci, A. S. (2007). “The next influenza pandemic: can it be predicted?” In: *Jama* 297.18, pp. 2025–2027.

Taubenberger, J. K. and Morens, D. M. (2006). “1918 Influenza: The mother of all pandemics”. In: *Emerg. Infect. Dis.* 12.1, pp. 15–22. DOI: 10.3201/eid1209.05-0979.

Taubenberger, J. K. and Morens, D. M. (2010). “Influenza: The once and future pandemic”. In: *Public Health Rep.* 125.SUPPL. 3, pp. 15–26. DOI: 10.1177/00333549101250s305.

Tong, S. et al. (2012). “A distinct lineage of influenza A virus from bats”. In: *Proc. Natl. Acad. Sci. U. S. A.* 109.11, pp. 4269–4274. DOI: 10.1073/pnas.1116200109.

Villmann, T. et al. (2017). “Can Learning Vector Quantization be an Alternative to SVM and Deep Learning?” In: *Journal of Artificial Intelligence and Soft Computing Research* 7.1, pp. 65–81.

Wang, H. (2020). “Influenza Neuraminidase - Novel mechanisms of influenza NA that enable adaptation and promote diversification”. Academic dissertation. Stockholm University, Sweden, p. 66. ISBN: 9789179112141.

Wang, Y. et al. (2019). “Protein Sequence Classification Using Natural Vector and Convex Hull Method”. In: *J. Comput. Biol.* 26.4, pp. 315–321. DOI: 10.1089/cmb.2018.0216.

Webster, R. G., Isachenko, V. A., et al. (1974). “A new avian influenza virus from feral birds in the USSR: recombination in nature?” In: *Bull. World Health Organ.* 51.4, pp. 325–332.

Webster, R. G., Bean, W. J., et al. (1992). “Evolution and ecology of influenza A viruses”. In: *Microbiological reviews* 56.1, pp. 152–179. DOI: 10.1128/mr.56.1.152-179.1992.

Wiedemann, C. et al. (2020). “Cysteines and Disulfide Bonds as Structure-Forming Units: Insights From Different Domains of Life and the Potential for Characterization by NMR”. In: *Front. Chem.* 8.April, pp. 1–8. DOI: 10.3389/fchem.2020.00280.

Wohlbold, T. J. and Krammer, F. (2014). “In the shadow of hemagglutinin: A growing interest in influenza viral neuraminidase and its role as a vaccine antigen”. In: *Viruses* 6.6, pp. 2465–2494. DOI: 10.3390/v6062465.

Woolhouse, M. et al. (2012). “Human viruses: Discovery and emergence”. In: *Philos. Trans. R. Soc. B Biol. Sci.* 367.1604, pp. 2864–2871. DOI: 10.1098/rstb.2011.0354.

Wu, Y. et al. (2014). “Bat-derived influenza-like viruses H17N10 and H18N11”. In: *Trends Microbiol.* 22.4, pp. 183–191. DOI: 10.1016/j.tim.2014.01.010.

Zhang, J. et al. (2014). “Determination of original infection source of H7N9 avian influenza by dynamical model”. In: *Sci. Rep.* 4.May. DOI: 10.1038/srep04846.

Zhong, G. et al. (2020). “Mutations in the Neuraminidase-Like Protein of Bat Influenza H18N11 Virus Enhance Virus Replication in Mammalian Cells, Mice, and Ferrets”. In: *Journal of Virology* 94.5. Ed. by S. Schultz-Cherry. DOI: 10.1128/JVI.01416-19.

Zhu, X. et al. (2012). “Crystal structures of two subtype N10 neuraminidase-like proteins from bat influenza A viruses reveal a diverged putative active site”. In: *Proc. Natl. Acad. Sci. U. S. A.* 109.46, pp. 18903–18908. DOI: 10.1073/pnas.1212579109.

Erklärung

Hiermit erkläre ich, dass ich meine Arbeit selbstständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und die Arbeit noch nicht anderweitig für Prüfungszwecke vorgelegt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Mittweida, 01.04.2022

Lynn Vivian Reuss