Large Scale Reevaluation of *Influenza A Virus* Classification based on *k*-mer frequencies

MASTERARBEIT

zur Erlangung des akademischen Grades
Master of Science (M. Sc.)
im Studiengang Bioinformatik

FRIEDRICH-SCHILLER-UNIVERSITÄT JENA
Fakultät für Mathematik und Informatik

eingereicht von Alexander Henoch
geb. am 02.02.1995 in Bremen

Betreuer: Prof. Dr. Manja Marz

Jena, den 14.07.2021

Zusammenfassung

Wiederauftretende lokale Ausbrüche von hoch pathogenen *Influenza A Virus* (IAV) Strängen, verweisen auf eine ständig presente Gefahr für die Menschheit, die in der Vergangenheit mehrfach globale Ausmaße mit hohen Todeszahlen zufolge hatte. Da kein verlässliches Medikament vorhanden ist, muss unausweichlich auf Impfungen zurückgegriffen werden, die verschieden hohe Immunisierung gewähren und jährliche Erneuerung erfordern. Eine Erweiterung der Kenntnisse über IAV, ist daher unerlässlich, um eine bessere Vorbereitung auf mögliche zukünftige Pandemien zu ermöglichen. Hohe Evolutionsraten durch die drastischeren Mutationsmechanismen von IAV und eine wenig Einsicht gewährenden Klassifizierung, verkomplizieren dabei allerdings die Gewinnung neuer und exakter Forschungsergebnisse. Diese Thesis dient daher der Herausarbeitung einer Pipeline, zur segmentweisen Klassifizierung aller sequenzierten Genome von IAV. Anstelle von Alignments, nutzt diese Methode besser skalierbare k-mer Frequenz-Vektoren mit dem neuen hybriden Clustering Ansatz von `HDBSCAN`, der hierarchische und auf Dichte basierende Methoden vereint. Geeignete Parameter wurden mit Hilfe verschiedener Validierungs-Techniken ausgesucht und die Dimensionalität der genutzten Vektoren mit bekannten Tools verringert. Die Ergebnisse wurden im Detail verglichen, wodurch ein Workflow kreiert wurde, der eine neue Clustermethode, mit validierten Parametern und einer zuvor erfolgten effizienten Reduzierung der Dimensionen, verbindet.

Abstract

Reoccurring local outbreaks of new, highly pathogenic strains of the *Influenza A Virus* (IAV), picture a unnoticed but still persisting major danger to the whole human population, that reached global extend with high numbers of fatalities, several times in the past. Due to the lack of a cure, resort to vaccines producing varying levels of immunization with yearly expiration is inevitable. For better preparation on possible future pandemics, enlarging the knowledge of the IAV is crucial. High evolution-rates by more drastic mutation mechanisms of the IAV and a classification giving little insight, complicate accurate novel research though. This thesis, serves the elaboration of a segment-wise clustering pipeline, usable on all sequenced genomes of IAV. Instead of being alignment based, this method utilizes the better scalablity of k-mer frequency vectors, with the novel hybrid clustering implementation of `HDBSCAN`, connecting hierarchical with density-based clustering. Appropriate parameters were selected by different validation techniques and well-known tools were used for dimension reduction of the vectors. By in depth comparison of the results, a workflow combining a novel vector clustering method with validated parameters, posterior to an efficient dimension reduction, is proposed.

Contents

List of Abbreviations

Symbols

(-)-ssRNA negative-sense single stranded viral RNA

A

AA aminoacid

AIDS acquired immunodeficiency syndrome

C

COVID-19 coronavirus disease 2019

D

DBCV density based cluster validity

F

FFT fast Fourier transform

H

HA hemagglutinin

I

IAV *Influenza A Virus*

IRD Influenza Research Database

IRES internal ribosome entry site

L

LPAIV low-pathogenic avian influenza virus

M

M1 matrix protein 1

M2 M2 ion channel

mRNA viral messenger RNA

MSA multiple sequence alignment

N

NA neuraminidase

NEP nuclear export protein

NP nucleoprotein

NPC nuclear pore complex

NS1 nonstructural protein 1

P

PA polimerase acid protein

PB1 polimerase basic protein 1

PB2 polimerase basic protein 2

R

RNP viral ribonucleoprotein

S

SARS severe acute respiratory syndrome

U

UPGMA unweighted pair group method with arithmetic mean

W

WHO world health organisation

Danksagung

Zuallererst gebührt mein Dank den Gutachtern meiner Arbeit, Prof. Dr. Manja Marz und Prof. Dr. Matrin Schwemmle. Besonderer Dank gilt dabei Frau Marz, da Sie es mir ermöglicht hat, in Ihrer Arbeitsgruppe dieses Thema zu bearbeiten, an dem ich sehr großes Interesse habe.

Bei Kevin möchte ich mich für die Betreuung und die ständige Unterstützung während des gesamten Projekts bedanken. Ich bin letztendlich sehr froh, dass wir wöchentliche Meetings abgehalten haben, da mir die Diskussionen mit dir und deine vielen Tipps jedes Mal sehr geholfen haben. Dem anderen Insassen des Twilight Rooms, Emanuel, möchte ich ebenfalls danken, da er mir im Laufe meines Studiums viel Wissen vermittelt hat, dass grade für dieses Projekt letztendlich unerlässlich war, allen voran dabei seine Einführung in die Programmierung mit Python.

Insa danke ich für motivierende Briefe und liebe Worte, die mich während der ganzen Zeit immer angespornt haben weiter zu arbeiten und mein bestes zu geben.

Darüber hinaus möchte ich Sarah und Jannes für die schöne Zeit mit euch hier in Jena und häufigen Beistand mit Rat und Tat danken.

Danken möchte ich außerdem Thorben und Simon, die zu jeder Zeit für mich da sind. Ich habe mit jedem von euch inzwischen einiges durchgestanden und dabei nie auch nur eine Sekunde bereut euch kennengelernt zu haben.

Zu guter Letzt möchte ich mich bei meinen Eltern bedanken. Ihr unterstützt mich bereits mein ganzes Leben lang unermüdlich und sagt mir zu jeder Zeit, was ich zwar nicht immer hören möchte aber trotzdem hören muss. Bei all meinen Entscheidungen, ob gut oder schlecht habt ihr mir immer zur Seite gestanden.

Allen genannten Personen dafür an dieser Stelle nochmals: VIELEN DANK!

1 Introduction

1.1 Zoonoses and the risks of pandemics

In times where infectious disease outbreaks become more frequent and sometimes even reach global appearance, unpredictable effects on humans, wildlife and whole ecosystems are inevitable [64]. Growing human population and persisting poverty has harmful impact on the biodiversity and results in degradation of natural habitats and more frequent human-wildlife contacts [64]. Therefore, increasing numbers of zoonoses, transfers of animal pathogens on humans, arise and are a major driving force in pathogen emergence on humans in recent decades [29]. Most of the human pathogens emerging lately are of animal origin, indeed, up to 75% [73]. Well-known examples for diseases, that originate from viral zoonoses are the avian and swine flu, the acquired immunodeficiency syndrome (AIDS) and the severe acute respiratory syndrome (SARS) related to the current circulating coronavirus disease 2019 (COVID-19). [67, 72, 75, 77].

While zoonoses can be of viral, bacterial and parasitic nature, emergences of higher magnitude, like the mentioned well-known examples, are often linked back to viral infections [73]. Harmfulness of viral infections can be diverse, ranging from mostly no sign of infection in the natural hosts to very severe symptoms or death in accidental ones [81]. In contrast to natural hosts, humans, accidental hosts to, e. g. , the West Nile Virus, develop disease patterns upon infection and, thereby, are not able to fully support the virus life cycle [22]. High variety in host circulation and transmission ways from such natural or intermediate onto humans as accidental hosts, with long infectious periods without symptoms and high transmission pace have a high risk of pandemic events [28]. A prominent virus detected in a variety of hosts and known for reoccurring local and global outbreaks in the past is IAV, member of the *Orthomyxoviridae* family and also commonly known as flu virus [81]. Analysis indicate a 1% chance of a pandemic with millions of deaths every year and is, therefore, the pathogen most likely to be responsible for a sudden severe pandemic [28].

1.2 Life-cycle and structure of the *Influenza A Virus*

In humans, infection by IAV affects the upper respiratory tract [31]. General symptoms like fever, cough and headache characterize the infection [31]. In some cases complications can occur, resulting in primary viral pneumonia or secondary bacterial pneumonia by bacterial infection [31]. The virus particles of IAV, called virions, are spherical in shape, 80 – 120 nm in size and enveloped by the hosts cell membrane lipids [12, 49, 54]. These virions can sustain extensive forces in the hosts body and even survive deformation to around 33% of its total diameter [63]. Epithelial cells in the tissue of the respiratory tract, are the main infection area of the virions [54]. The IAV virions attach themselves to the host cells with their surface glycoproteins, the tetrameric neuraminidase (NA) and the trimeric hemagglutinin (HA) and enter the cells by clathrin-mediated endocytosis (Fig. 1.1) [29, 49, 76, 86]. Equilibration of pH in the process of endocytosis involve the M2 ion channel (M2) protein of IAV [60]. The surface glycoprotein tails are also connected to the second layer in the IAV virions, the matrix protein 1 (M1) [3]. Following the endocytosis, the viral ribonucleoprotein (RNP) complexes are carried to the hosts cells nucleus by the nuclear pore complex (NPC) for virus replication [19]. These RNP complexes contain the viral negative-sense single stranded viral RNA ((-)-ssRNA), holding the genetic information, bound to multiple nucleoproteins (NPs) and the polymerase complex [19]. The trimeric polymerase complex including the polimerase basic protein 1 (PB1), polimerase basic protein 2 (PB2) and polimerase acid protein (PA) is essential for viral replication in the hosts nucleus [5, 19]. IAV is a segmented virus, one virion of IAV contains eight different short (-)-ssRNAs, called segments, encoding in total 14 viral proteins [19]. In the nucleus, the eight (-)-ssRNAs are replicated and transcribed to viral messenger RNAs (mRNAs) with the latter translated to the virions proteins in the cytoplasm [19]. The translated nuclear export protein (NEP), M1, NP and proteins of the polymerase complex are imported by the NPC to build new RNPs with the replicated genomes and enable the nuclear exit [19]. By budding through the plasma membrane with help of the translated M2 and NA, while incorporating these proteins including HA into the surface, new virions are released coated in the host cell membrane lipids [19].

1.3 Evolution of the *Influenza A Virus*

Present day research, indicate mallards *(Anas platyrhynchos)* as main reservoir and natural host of less dangerous IAV strains called low-pathogenic avian influenza viruses (LPAIVs) [30]. Studies on Pekin ducks, descended from mallards, have shown minor immune responses and antibody production to the infection with LPAIV strains and the possibility of a reinfection after two months with the same strain [34]. Strains are lines of IAV related to a specific location and time point [12]. These LPAIV strains of IAV, seem to repeatedly circulate in duck species and may evolve into human pathogenic strains by zoonoses [30]. Simple transmission over species is not enough to start a pandemic though, therefore, better understanding of the mostly unknown genetic changes vital for zoonotic events is required [75]. Circulation of these strains in aquatic bird species with continued evolution enable the transmission possibilities to humans, lower animals, and other birds [83]. The evolution occurs in all segments of the IAV but is most prominent in HA and NA [83]. Infection is mostly dependent on these surface glycoproteins, or surface antigens, as they are crucial for antigenic attachment to host cells [12]. Significant variation in the surface antigens mostly occur by reassortment, also called genetic shift and point mutations, also called antigenic drift [83]. Current classification of IAV by the subtypes nomenclature is solely based on characterization of the two antigens [2]. Prior to the current classification, IAV subtypes were separated by host origin, based on defining just major antigenic differences. These previous subtypes consisting of H0 to H3, Hsw1, Heq1, Heq2 and Hav1 to Hav10 were replaced by the known subtypes H1 to H12 for HA and in a similar for NA by replacement of N1, N2, Neq1, Neq2 and Nav1 to Nav6 in favor of N1 to N9 [1]. The infixes denote **eq**uine, **sw**ine and **av**ian origin or otherwise human antigenic character[1]. The subtype nomenclature was changed to include more subtle characterization of the IAV genome based on immunological relationships [61]. Thereby, the IAV subtypes were described by a sequential system involving the antigenic character regardless of host origin. Origin and other informations are included in the strain naming system involving position, number and year of detection [2]. The current subtypes were also grouped according to the sequence homology of the nonstructural protein 1 (NS1) with some major differences [61]. Still, due to missing serological data for NS1, only HA and NA were used for the classification. Other segments were not considered due to being highly conserved [61]. However, future classification involving all the segments was not negated, if appropriate [61].

Point mutations in the segmented genome are very frequent, as the mutation rate in all RNA viruses is very high [16]. Present *poliovirus* research, indicate higher selection for faster replication and, therefore, acceptance of replication errors in favor of faster viral polymerases [16, 58]. This finding is in line with research indicating the short length or segmentation of RNA viruses and the high mutation rates as evolutionary trade-off [8, 80]. Thereby, creating a cloud of offsprings, called quasispecies, with 1-2 mutations in the genome each [8, 80]. The point mutations can affect the offsprings proteins aminoacid (AA) composition in the translation process, by possible missense and nonsense errors or frameshifts [55, 83].

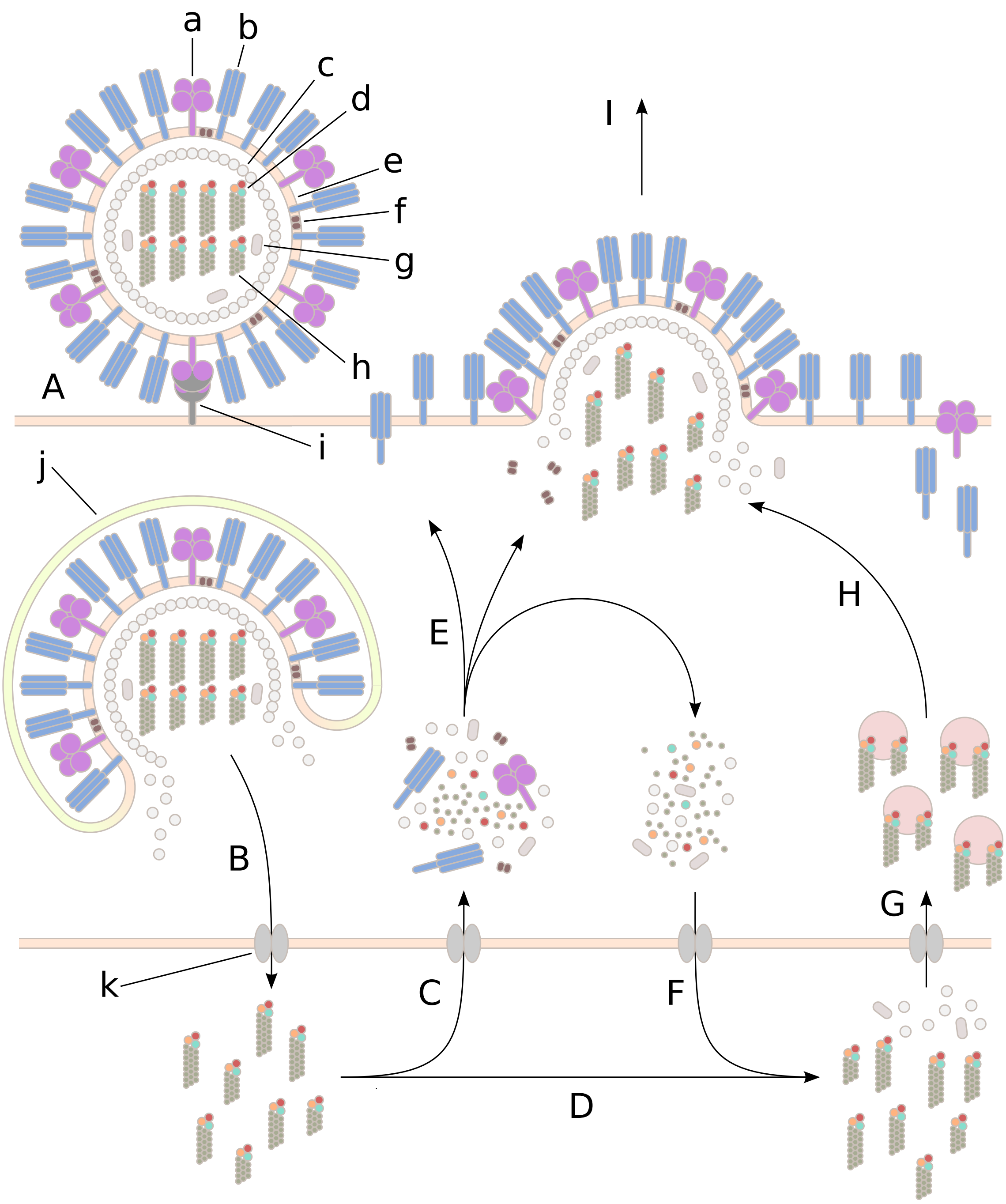


Fig. 1.1 ***Influenza A Virus* life-cycle.** The IAV virion attaches itself to the host cell membrane using HA (**a**) to bind to the cell receptors (**i**). By endocytosis (**A**), the virion infiltrates the host cell. After assimilation into the cell, the virion escapes the endosome (**j**) by raising the pH involving the M2 protein (**f**), spilling the segments of the virus into the cytoplasm. After nuclear import (**B**) by the NPC (**k**), the segments are transcribed and the mRNA is transported to the cytoplasm for translation (**C**). The newly build M1 (**c**) and NEP (**g**) proteins, as well as the NPs (**h**) and the proteins of the polymerase complex (**d**) are transported back into the nucleus (**F**) to create new RNPs with the replicated genomes (**D**) and enable the nuclear exit (**G**). The newly build HA, NA (**b**) and M2 proteins are incorporated into the hosts cell membrane (**E**). Using vesicle transport, the RNPs are transported to the position in the membrane containing the surface proteins and are incorporated into the progeny virion by embedding into M1 proteins (**H**). The virion is released involving NA and M2 by budding the hosts cell membrane (**I**) and coating itself in it (**e**).

Reassortment is a more drastical change in the surface proteins and likely to be related to the pasts most horrible IAV pandemic, the spanish flu in 1918 [52]. For reassortment induced zoonoses, there practically always has to be a intermediate host or „mixing vessel“, most likely pigs, able to be infected by IAVs strains of different hosts origin [68]. The hosts cells can then be co-infected by two different IAVs and create offsprings with mixed segments [70]. All segments can be exchanged, but, in case of segments coding for surface proteins of different hosts are mixed together, IAVs can occur that are able to do interspecies transmission [68]. Avian IAV strains can, thereby, evolve to human strains by co-infection of a pig with human transferable and avian originating IAV strains [68].

1.4 Vaccines design and the link to reassortment

Reassortment events are a major source of danger, since no real cure to IAV infection is available and generation of vaccines is straining and not always as effective as expected [81, 87]. Furthermore, the efficacy of vaccines vary in specific populations and there are limitations to the manufacturing and the time frame of the production [87]. The strains most likely to circulate for the season are selected twice a year by the world health organisation (WHO), to be included in vaccines prepared for the winters in both hemispheres [7]. The seasonal used IAV vaccines target the highly mutable head domain of HA surface proteins to stimulate immune response [84, 87]. Therefore, the IAV vaccines efficiency vary depending on the similarity of the HA head domain of the strains used for the vaccines and the ones circulating in the season [84]. The accuracy of the recommendation by the WHO is, therefore, especially crucial for the survival of humans with pre-existing conditions or humans of old age that are more prone to infection. To manufacture the vaccines, reassortment of the selected strain with a master strain is induced in chicken eggs [87]. The resulting hybrid strain contains the selected strains surface proteins and the master strains high-growth properties, necessary for production of the vaccine in the short time frame [87]. Therefore enlarging the knowledge of IAV reassortment is important for the prediction of future pandemic strains, creation of vaccines by high-growth hybrid strains and the overall efficiency of the vaccines against circulating variable strains, most likely to undergo reassortment [14, 87]. For better understanding of IAVs reassortment and estimation of resulting risks, the interaction mechanisms of the genome segments have to be fully discovered [14].

1.5 Importance of secondary structure predictions

The eight IAV (-)-ssRNA segments are single-stranded chains of nucleotides, by inter- and intra-molecular base-pairing, various complex arrangements with different stabilities can be build [14, 26]. Single-stranded RNA viruses can use secondary structures on the (-)-ssRNA, as well as on the transcribed positive mRNA for different mechanisms, like the initiation of the translation on the mRNA by the internal ribosome entry site (IRES) [35]. Segmented viruses are able to perform inter-molecular binding of different (-)-ssRNA segments to each other [14, 46]. Gerber et al. [23] described the selective IAVs packaging by segment interactions with consequences for reassortment. It is assumed, that different interactions favor reassortment, while others prevent specific segment incorporation in the reassortant virus. Fully understanding the conserved structures and interactions of IAVs segments would, thereby, enlarge our knowledge of IAV to a great extend. Prediction of these viral secondary structures is mostly done by lab methods *in virio* and *in vitro* or computationally with *in silico* methods, with the latter mostly based on thermodynamic calculations alone [14, 46]. *In virio* methods involve modification inside a virion and *in vitro* methods involve modification on transcribed RNA in a probe, both can used with SHAPE-MaP and SPLASH methods [14, 69]. Following the lab methods, structure prediction, is performed by tools using the gained insight for better accuracy, in Dadonaite et al. [14] using `IntaRNA` [42]. Prediction of present day secondary structures by *in silico* thermodynamic energy minimization calculations is mostly performed by tools involving the ViennaRNA package, like on single sequences, `RNAfold` [38]. Both lab methods can only be used to analyze secondary structure folding on single viruses at once, but reveal structures at single-nucleotide resolution [14]. *In silico* methods, that are used without support by experiments, are not limited by prediction on single viruses, only require prior sequenced genomes [14, 46]. Since *in silico* methods can be used on a higher number of sequences at once, consensus structures can be predicted together to find conserved, possibly equal folding regions in the sequenced genomes. by prior multiple sequence alignments [46].

1.6 Alignments and clustering

Aligning multiple sequences to each other is possible by a number of different methods nowadays. The core of most multiple alignment methods were created by Needleman and Wunsch [51], with an algorithm usable to aligns two sequences in a pairwise manner using fast dynamic programming. The algorithm was intended to be used solely on proteins but could be transfered to any problems involving pairwise distance and was soon used for nucleotide sequence comparisons [59]. Other algorithms were created in the following years but the one proposed by Needleman and Wunsch was soon used to not only align two sequences but multiple sequences, creating the possibility sequence comparisons of multiple sequences [59]. Due to inability to be used on high numbers of sequences other algorithms, like the one proposed by Feng and Doolittle [21], were created. These algorithms did not offer the same accuracy but could be performed on higher number of sequences, thereby, creating first heuristic multiple sequence alignment methods. The way for present multiple sequence alignments was paved, offering the possibility to align higher numbers of sequences in reasonable time. Present day multiple alignment tools offering global comparisons of the complete sequences are still based on these heuristic methods. Famous ones are up-to-date versions of `CLUSTALW` first proposed in Thompson et al. [74] and `T-COFFEE` proposed in Notredame et al. [53]. While steps in terms of accuracy had been made in the development of newer versions, calculations still have high CPU times and hardware offering high amounts of computational power are still needed [32]. Other alignment methods exist, searching for similar short profiles in the sequences and extend these matches. `MAFFT` first proposed in Katoh [32] and evolved to the present day version as described in Katoh and Standley [33] uses fast Fourier transform (FFT) for similar profile search, resulting in faster and less costly nucleotide sequence comparison. `MUSCLE` also prominent for multiple sequence alignments (MSAs) creation, is based on k-mers profiles instead, also improving speed [18]. More alignments in shorter time-spans enable searching for conserved structures in a higher magnitude. MSAs created by the available present day tools can be used in structure predictions by e. g. , `RNAalifold` [9].

Since *in silico* methods are based on sole predictions involving thermodynamic calculations, the choice of a set to use when aiming to make statements about higher amounts of sequences is crucial. Prediction of conserved consensus structures is, therefore, highly fragile in terms of high genomic differences of the used sequences. Prior clustering to discover related groups can be helpful in improving the accuracy of consensus structures as performed in Moss et al. [46] for prediction of a tetraloop structure. Clustering techniques are broadly used to discover new insights of biological data, not only to predict more accurate structures in IAV segments, but are also used on post-genomic data in all fields of bioinformatics [24]. Clustering methods exist, based on entirely different concepts to separate the data. They can split given data in groups by connectedness, compactness or spatial separation of data points [24]. By separation based on compactness, the method aims to reduce the intra-cluster variation as much as possible, thereby, mostly creating spherical clusters (Fig. 1.2a) [24]. The concept of connectedness connects neighbors to each other, thus, creating chains of associated data points (Fig. 1.2b). The clusters are mostly arbitrarily shaped. Spatial separation is a concept aiming for splitting the data in

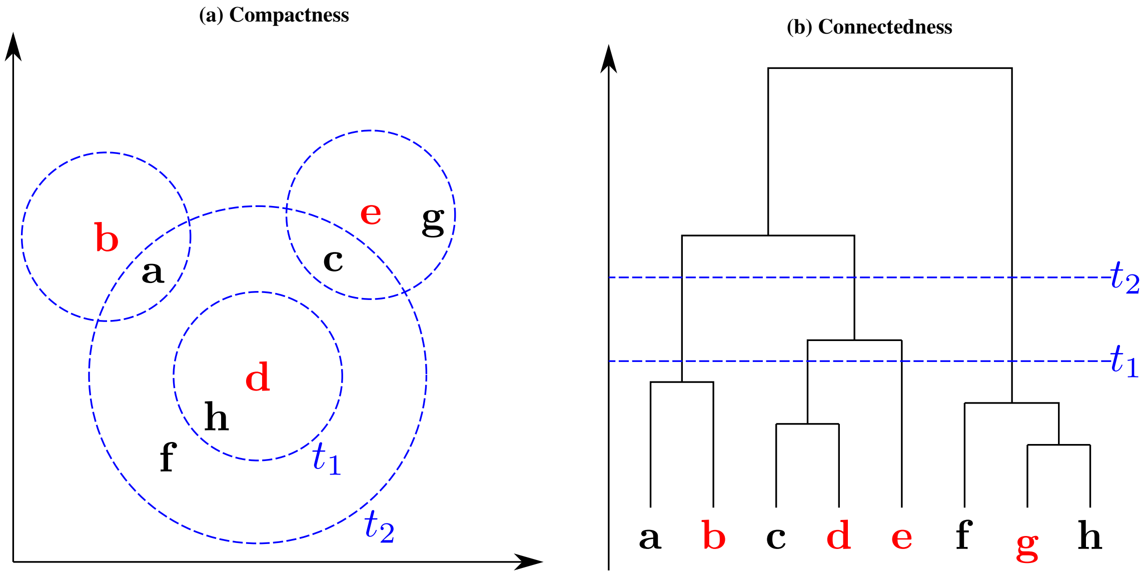


Fig. 1.2 Clustering methods. Clustering is most frequently used by density-based separation, using the compactness concept or agglomerative hierarchical clustering with the connectedness concept. The letters denote datapoints with the clusters most meaningful points as centroids in red, when using threshold t_1. Both methods are highly dependent on the used parameters for the clustering. Therefore, the thresholds t_1 and t_2 illustrate major differences on the resulting clustering. Choosing t_1 or t_2 in the hierarchical connectedness clustering either produce a cluster containing c, d and e or two clusters, one containing c and d and the other containing only e, leaving e unclustered. Setting the threshold to high can on the other hand result in a even smaller number of large clusters containing higher differences. Setting the value to small results in more unclustered datapoints. Threshold definition in the density-based compactness clustering defines the magnitude of the density necessary to induce cluster separation. For the illustration of the compactness clustering, the values for t_1 and t_2 are used as cluster selection by a radius including at least a number of two points for t_1 and five points for t_2. Therefore, t_1 would create three clusters leaving f out unclustered. The value t_2 on the other hand would create a bigger cluster with higher differences, including f but would most likely leave b e and g unclustered.

different regions and is related to the concept of compactness [24]. Existing clustering algorithms try to best separate the data based on these concepts. Still, no existing clustering algorithm can take all of these concepts into consideration. Most clustering algorithms follow the principle of either connectedness, like hierarchical clustering algorithms, or the compactness concept, involving also the spatial separation, like density-based clustering algorithms [24]. Hierarchical clustering algorithms can be used by starting with each datapoint in a single cluster and merging while climbing the hierarchy, called agglomerative or divisive, when starting in on cluster and dividing in smaller ones [50]. Prominent widely used examples for agglomerative hierarchical clustering are the unweighted pair group method with arithmetic mean (UPGMA) method, that is also present in `MAFFT` for guide tree creation [32]. Also the recent published `HDBCSCAN` tool. using the single-linkage hierarchical clustering method [43]. A well-known examples for density-based clustering is the tool `DBSCAN` [40, 65]. Regardless of which clustering methods is used, there is always a threshold that has to be defined in order for the algorithm to work as expected

(Fig. 1.2b) [40]. Visual inspection on the clustering is not always possible making these threshold parameters a tough choice. Hierarchical clustering methods need a threshold to define the cutoff in the tree and density-based methods depend on choosing the size of an area with a given density to be handled as cluster [40]. Estimation of a reasoned number of clusters is possible in some clustering methods by the elbow method, thus, reverse estimating the threshold parameter necessary for the number of clusters [40, 62]. Clustering methods, like USEARCH and CD-HIT, that are especially developed to be used for sequence clustering, require a threshold based on sequence similarity only [17, 37]. However, the majority of cluster tool use data points as vectors for clustering in statistical data analysis and distance measurements instead of sequence similarity [40]. The dimensionality of these vectors is dependent on the amount of information. Using vectors with a high amount of information for clustering requires lowering the dimensionality of the data prior to clustering [6]. Therefore, combination of the clustering with methods, that reduce dimensionality is crucial in the most cases. Widely used methods for reducing the dimensionality of vectors are PCA, t-SNE and UMAP. Choosing the right amount of preserved information in combination with wisely selected thresholds define, thus, a well conducted high-dimensional vector clustering [39, 44, 56].

1.7 The proposed project

The present subtype classification of IAV is solely based on immunological research on the surface proteins. Since release of this classification around 40 years, with enormous progress in computer technology, have passed. Aside from the raw number of new sequenced genomes of IAV in these years, also ways for faster sequence comparisons methods using profiles like k-mer comparisons and more accurate clustering algorithms were paved. Using knowledge from this elapsed time, the current classification will be reevaluated from the perspective of bioinformatics, to possibly find subtle differences to renew the classification with more detailed subgroups [2]. Due to the usage of the raw amount of all high quality sequences available for IAV, the clustering into groups were performed without any alignments, searching for a faster, more scaleable and hopefully more accurate method. Instead of alignments, a distance measurement using vector representation based on genomic k-mers will be used in combination with high dimensional clustering methods. As already mentioned, k-mer distance for genomic comparison was also described in RC Edgar [18] for faster alignments. To handle the high dimensionality of the used k-mer representation vectors used in this project, different dimension reduction methods were described and compared. For clustering, hybrid HDBSCAN will be used, combining compactness and connectedness for the best accuracy possible, aiming for high quality clustering of the huge amount of highly variable IAV sequences. Threshold definition is a complex procedure with high impact on the results and will be solved by different approaches, involving the Kneedle Algorithm, implementing the elbow method for the most appropriate results. This project aims for a clustering based classification of all eight segments of IAV, hopefully paving the way for future research to discover more detailed consensus structures and new insights into the molecular life-cycle of the IAV. The current subtype classification will support the clustering of segment 4 HA in this project and, thereby, create a blueprint to cluster the other segments in a similar way. Nevertheless, due to the lower evolutionary pressure, less clusters for the segments not coding for surface proteins are to be expected. For a simple and usable new classification, less than 100 clusters per segment would be comfortable and are anticipated, when considering the number of reassortment events in H1N1 proposed in Nelson et al. [52].

2 Materials and Methods

2.1 Data and pipeline

The tools listed in Table 2.1 were installed using the Conda package distribution system version 2-2.4.0 [4]. A configuration file for recreation of the used environment is present in the projects GitHub repository[1]

Table 2.1 Pipeline tools. All packages used in the project are listed, including their purpose in the project and their source.

Name	Version	Purpose	Source
BioPython	1.78	alignments and tree construction	[13]
ETE3	3.1.2	tree plotting and labeling	[27]
HDBSCAN	0.8.26	hybrid vector clustering	[43]
kneed	0.7.0	Kneedle Algorithm implementation	[62]
MAFFT	7.475	multiple sequence alignment	[33]
numpy	1.19.5	matrix and vector calculations	[25]
pandas	1.2.2	dataframe creation and management	[45]
seaborn	0.11.1	plotting and data visualization	[82]
scikit-learn	0.24.1	PCA and vector normalization	[57]
SciPy	1.6.0	vector distance calculations	[66]
UMAP	0.4.6	UMAP dimension reduction	[44]

Since its file size exceeds the limits of GitHub, the FASTA file containing all the sequences of the *Influenza A Virus* (IAV), that are used in this project is present on the attached USB stick and in the FSU-Cloud[2]. The FASTA file can be manually retrieved from the Influenza Research Database (IRD)[3] using the settings in Table 2.2 for nucleotide sequence search. The header of the FASTA file has to be formatted as Accession Number, Strain Name, Segment, Protein Symbol, Type, SubType, Date, Host Species, Curation Flag, in the given order before downloading from IRD, for the tool to work as expected. The version used for the proposed results was acquired at 08/11/2020[4]. Newer versions[5] might change the results slightly.

[1] https://github.com/ahenoch/Masterthesis.git

[2] https://cloud.uni-jena.de/s/fYkQ2NAwjND8oEM

[3] https://www.fludb.org/brc/home.spg?decorator=influenza

[4] GenBank Genome Sequence/Annotation Update <= 11/2020

[5] GenBank Genome Sequence/Annotation Update >= 05/2021

Table 2.2 Search parameter. The parameters to use on the nucleotide sequence search interface of the IRD.

Field	Parameter
Data Type	Genome Segments
Virus Type	A
Complete Genome	Complete Genome Only
Select Segments	All
Complete	All

Table 2.3 Summary of the clustering methods. For easier separation, the different Methods are listed. The methods PD and PK use the same dimension reduction by Fig. 2.1 workflow **1**, followed by different ε exploration by Fig. 2.2 workflow **3** for method PD and **4** for PK. Method UD and UK, use the Fig. 2.1 workflow **2** for dimension reduction instead.

	Reduction		
Abbreviation	**100 Components**	**30 Components**	**Exploration**
PB	—	`PCA`	DBCV
PK	—	`PCA`	Kneedle Algorithm
UD	`PCA`	`UMAP`	DBCV
UK	`PCA`	`UMAP`	Kneedle Algorithm

In this project four different ways to cluster the segments of IAV are described and discussed (Table 2.3). The methods were compared to each other and analyzed for their capability of IAV clustering. Abbreviations of the four methods were used in the following as indicated in Table 2.3. A combined version of the pipelines in Fig. 2.1, Fig. 2.2 and Fig. 2.3 is available in the projects GitHub repository[6] as a novel clustering tool for IAV genomes (Sec. 3.6). The tool contains the method elaborated as best suitable for IAV clustering and is intended to be used for future research. Execution of the tool on the FASTA file containing 449462 sequences takes around one and a half hours. The sequences are, thereby, clustered segment-wise, based on their 7-mer frequencies. The output consists of database ready CSV files holding the cluster assignment of every sequence, analysis graphics and a labeled clustering tree of each used segment.

The method is based on Viehweger et al. [78]. Instead of using the tool `nanotext`, as proposed in Viehweger et al. [79], a simple 7-mer frequency calculation was implemented and used as described in Sec. 2.2. Similar to Viehweger et al. [78], the calculated vectors were clustered using the same tool `HDBSCAN` but with settings described in the Sec. 2.4 instead. Since `nanotext` was not used in this project, different types of dimension reduction were performed and compared, as described in Sec. 2.3 and discussed in Sec. 3.1. With

[6] https://github.com/ahenoch/Masterthesis.git

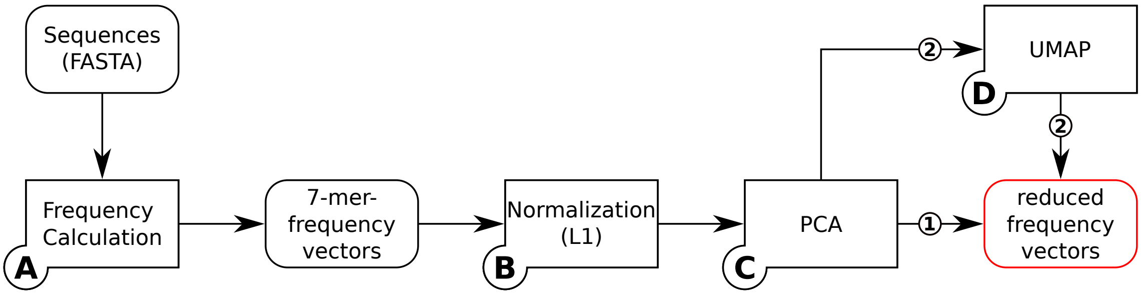


Fig. 2.1 Preprocessing pipeline. To create high-quality vectors representing the sequences, the FASTA file was translated to normalized vectors containing the 7-mer frequencies of the specific sequences (**A** and **B**). By workflow **1**, a low complexity representation of the vectors was obtained for clustering, using PCA (**C**). Workflow **2** describes additional execution of UMAP, that can be used after PCA as intermediate instead of final step (**D**). Reduction with workflow **1** or **2** resulted in reduced frequency vectors with 30 components. A red frame denotes a result of the given workflow.

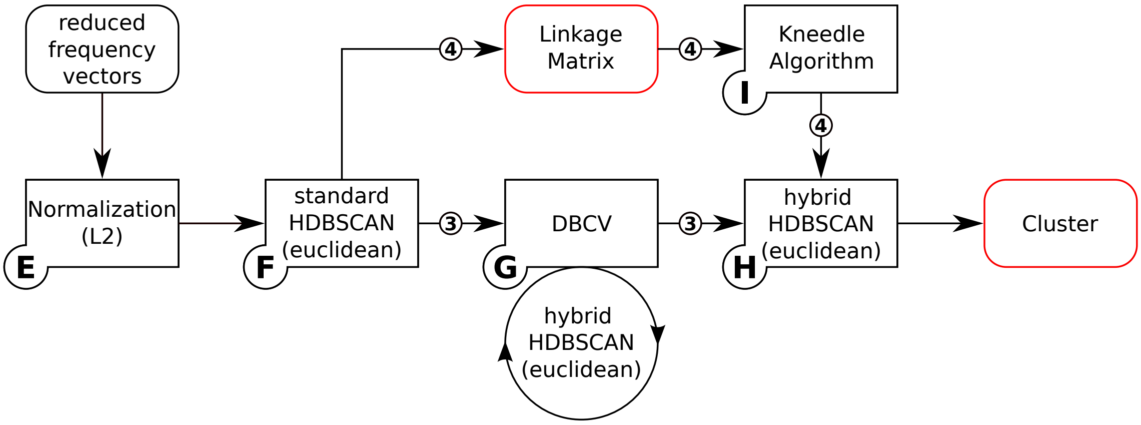

reduced frequency vectors
Normalization (L2)
E
standard HDBSCAN (euclidean)
F
4
Linkage Matrix
4
Kneedle Algorithm
I
4
3
DBCV
G
hybrid HDBSCAN (euclidean)
3
hybrid HDBSCAN (euclidean)
H
Cluster

Fig. 2.2 Clustering pipeline. Following the preprocessing pipeline (Fig. 2.1) normalization was used again with L2-norm as preparation for HDBSCAN (**E**). Initial HDBSCAN clustering (**F**) was performed in preparation to the ε exploration using either workflow **3** or **4**. Final hybrid clustering (**H**) was then executed on the results of the ε exploration using the Kneedle Algorithm (**I** and workflow **4** or DBCV (workflow **3** and **G**). A red frame denotes a result of the given workflow.

reference to the use of cosine similarity $s_{\cos}(\mathbf{x},\mathbf{y})$ as measurement for genomic similarity in `nanotext`, the use of the complementary cosine distance $d_{\cos}(\mathbf{x},\mathbf{y})$ in HDBSCAN was targeted in this project (Eq. 2.1) [79].

$$\begin{aligned} d_{\cos}(\mathbf{x},\mathbf{y}) &= 1 - s_{\cos}(\mathbf{x},\mathbf{y}) \\ &= 1 - \frac{\mathbf{x}^{\top}\mathbf{y}}{\|\mathbf{x}\|_2\|\mathbf{y}\|_2} \\ &= 1 - \cos(\Theta) \end{aligned} \tag{2.1}$$

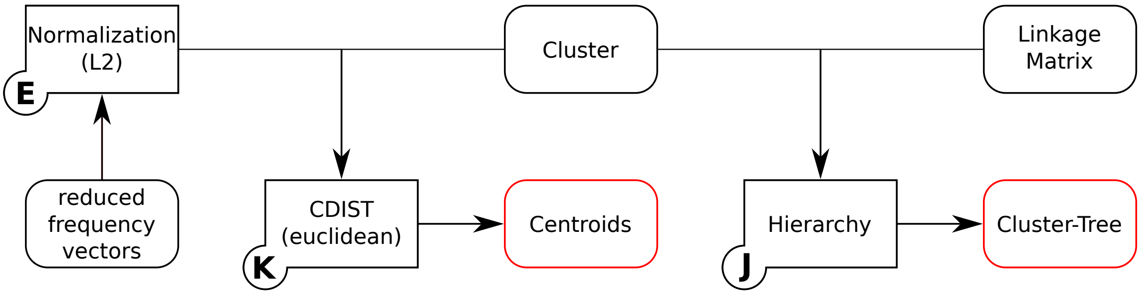


Fig. 2.3 Postprocessing pipeline. Following the clustering pipeline (Fig. 2.2) the cluster-tree is build by `BioPython` and visualized by `ETE3` (**J**). For every cluster the vectors with the smallest distance to the other cluster members are calculated and determined as the clusters centroids (**K**). A red frame denotes a result of the given workflow.

2.2 7-mer frequency calculation

The FASTA file containing the genomes of IAV for clustering, was converted to vectors to enable clustering in high dimension by counting their 7-mer frequency (Fig. 2.1 **A**) [18]. 4^7 possible constellations of the nucleotides A,C,G and T with length seven exist. Therefore, taking every constellation into consideration, the vectors of the sequences had 4^7 components. The numbers in the vectors components were the number of occurrences of the related 7-mer in the sequence. The first constellation of the 4^7 possible ones with length length was AAAAAAA, given a example sequence containing this 7-mer ten times, the first component of the sequences vector would be 10. Fig. 2.4 illustrates the calculation with 3-mers instead of 7-mers.

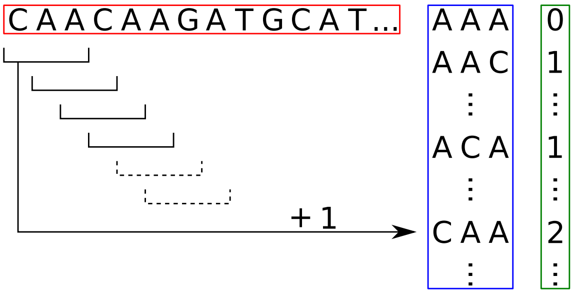


Fig. 2.4 *k*-mer vector creation. An example genomic sequence (red box) is splitted into 3-mers. The sequences 3-mers are then compared to the list of all 4^3 possible 3-mer constellations (blue box) Based on the occurrence number of the lists 3-mers in the sequence, a vector with 4^3 components is created (green box).

To gain the frequency of the 7-mers, all the vectors were normalized to a vector sum of one, according to L1-norm (Eq. 2.2 and Fig. 2.1 **B**).

$$\hat{\mathbf{x}} = \frac{\mathbf{x}}{||\mathbf{x}||_1} \tag{2.2}$$

2.3 Dimension reduction

PCA was used to handle the complexity of the vectors by simplification with the least loss of information possible (Fig. 2.1 **C**) [56] [57].

Without posterior use of UMAP, 30 components were extracted by the PCA (Fig. 2.1 workflow **1**) and otherwise 100 (Fig. 2.1 workflow **2**). Extraction of 30 or 100 components out of 4^7 in total equals $\approx 0.18\%$ or $\approx 0.61\%$. The size limit of the PCA function for calculation with default setting svd_solver='auto' is 500 different vectors, with 500 components and at least 80% of the components to extract. Since every maximum for this standard settings was exceeded, svd_solver='randomized' setting was used automatically.

UMAP was used for dimension reduction similar to PCA, aiming to better preserve the global structure of the data (Fig. 2.1 **D** and workflow **2**) [44]. It is similar to the well-known t-SNE, with better run time performance and better structure preservation in the lower dimension and less restrictions [39, 44]. The used parameters of UMAP match most of the ones listed in the manual under section „UMAP enhanced clustering“[7]. The settings are proposed to be used with UMAP prior to HDBSCAN clustering. Since the goal was clustering, not plotting 30 components were used instead of the proposed two. The neighbors number was also changed. It is recommended to be set in a range of one to 100. Based on the high number of sequences used e. g. , 56617 for segment 4, the highest recommended setting of 100 was used to better preserve the global picture of the data. Also based on the input size n_epochs=200 setting was used automatically.

UMAP was used posterior to dimension reduction with PCA, because of the similarity to t-SNE. As explained in the manual of t-SNE[8], the dimension should be reduced to a reasonable amount prior to execution to reduce noise. Furthermore, in section „What is the difference between PCA / UMAP / VAEs?“ of the UMAP manual[9] a pipeline is proposed, to also reduce from high dimension with PCA, continue with reduction by UMAP and cluster with HDBSCAN and is, therefore, also used as reference. Reduction with UMAP to 30 components posterior to PCA with 100 components also provided a comfortable balancing of computational effort of both methods, while preserving $\approx$ 85% explained variance with PCA [44].

[7] https://umap-learn.readthedocs.io/en/latest/clustering.html (accessed 07/01/2021)

[8] https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html (accessed 07/01/2021)

[9] https://umap-learn.readthedocs.io/en/latest/faq.html (accessed 07/01/2021)

2.4 Hybrid clustering

HDBSCAN was used to cluster the reduced vectors. It is an clustering algorithm proposed by Campello et al. [11], as a novel version of the well-known DBSCAN [10]. Execution of HDBSCAN involves varying values of ε, thus, not one specific threshold is used to define the clusters, but instead clusters of varying densities are extracted based on their stability [43]. HDBSCAN was used with hybrid clustering setting as proposed in Malzer and Baum [41], combining HDBSCAN with DBSCAN. Thereby, some of the disadvantages of using either of these methods could be avoided [43, 47]. Since DBSCAN is a density-based clustering tool, it is dependent on the parameters k and ε for clustering. Vectors not surrounded by k other vectors in a radius defined by this threshold value ε are omitted as single vector clusters or noise [20, 65]. Standard HDBSCAN, on the other hand, tend to create unwanted micro-clusters in areas of high density [43]. Using the hybrid HDBSCAN proposed in [41], a threshold value ε can be used to extract these high density areas as single clusters with DBSCAN, but still use the standard HDBSCAN for the otherwise omitted vectors (Fig. 2.5 and Fig. 2.2 **H**). Hybrid HDBSCAN is, thus, clustering with DBSCAN, resulting in a **raw** cluster number, containing finished clusters and omitted single vector clusters, and subsequent standard HDBSCAN, clustering the omitted vectors, reducing the **final** cluster number.

This method is useful when intending to cluster with a small cluster size value, while still aiming to cluster high-density areas together. Thereby it is well suited for the proposed clustering. Specific strains of IAV were sequenced a lot more, thus, probably creating high-density areas, that should be clustered together with the DBSCAN part. The standard HDBSCAN part of the hybrid clustering is then used with a small minimum cluster size to find clusters of rare sequenced variants, with possibly important mutations in low-density areas [41]. The smallest minimum cluster size of two was used. To declare as least vectors as possible as noise, the minimum samples value k was also set to the minimum of one. The appropriate ε was identified with explorations by different methods. In preparation of the ε exploration, as described in Sec. 2.5, it was necessary to use standard HDBSCAN once alone, without the hybrid setting Fig. 2.2 **F**). The standard HDBSCAN and the hybrid HDBSCAN, involving, as described, also the standard version, were used with the same settings plus a respective ε value in the latter one Fig. 2.2 **H**).

Distance calculations by HDBSCAN were performed with the `metric='euclidean'` setting, due to an open issue in the GitHub Repository of HDBSCAN[10]. In the issue, the inability to use cosine distance metric with HDBSCAN and the approximation of it by euclidean distance of L2-norm normalized vectors, is described (Eq. 2.3 and Fig. 2.2 **E**).

[10] https://github.com/scikit-learn-contrib/hdbscan/issues/69 (accessed 06/02/21)

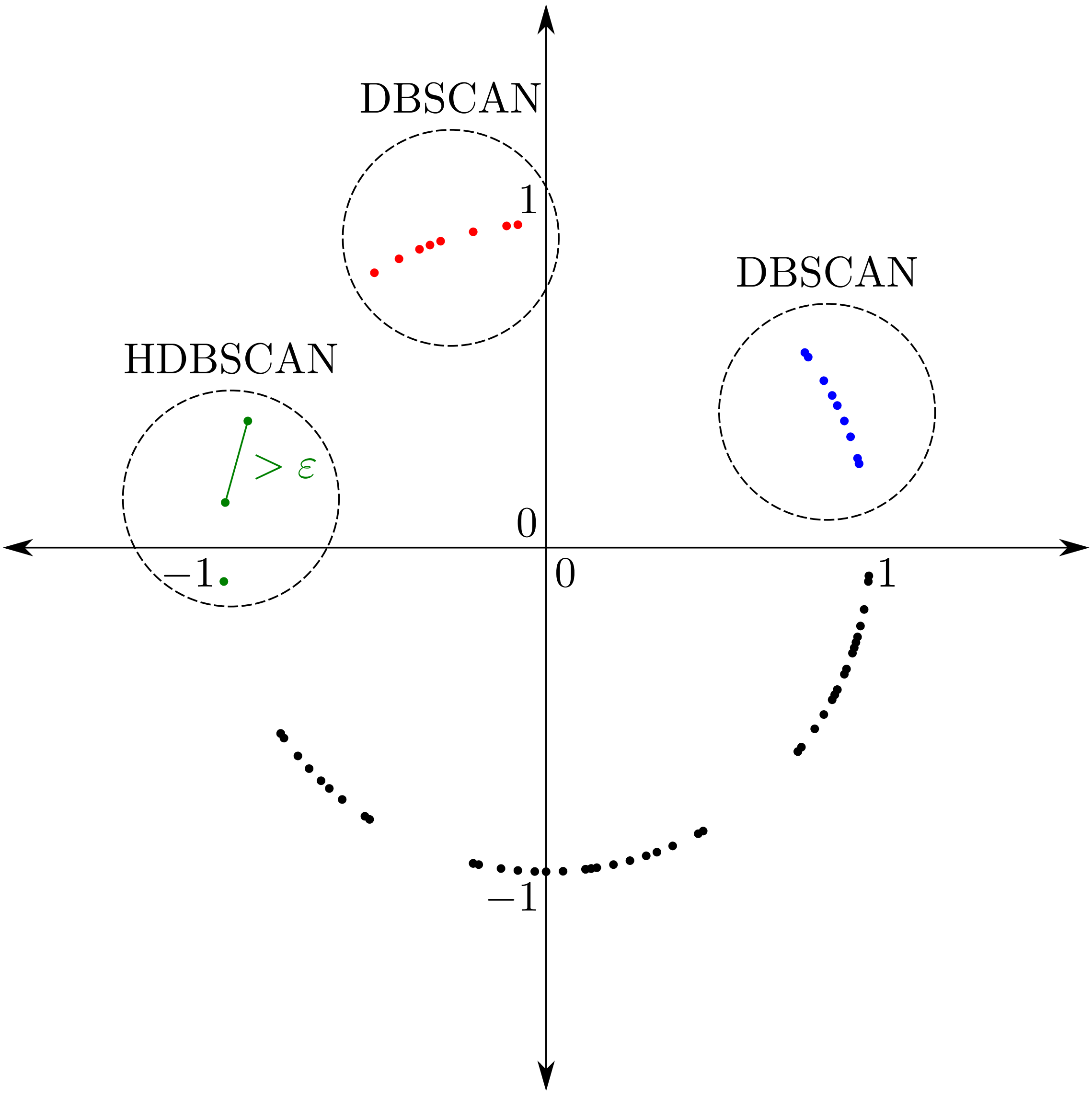


Fig. 2.5 Hybrid clustering threshold. The hybrid clustering `HDBSCAN` differentiate between clusters where vectors are connected by distances smaller and higher than ε. When the distance is smaller, the `DBSCAN` algorithm is used and clusters are generated based on this threshold value ε by combining vectors with smaller distance. Omitted vectors, not having a given number of $k = 1$ vectors in reachable distance of ε and are, therefore, impossible to be clustered by `DBSCAN`, are subsequently clustered with `HDBSCAN` building clusters with higher threshold if appropriate. The graphic is based on „Combining HDBSCAN* with DBSCAN“ in the manual[11]and adapted to the euclidean distance calculations in this project, as a two dimensional example.

$$\hat{\mathbf{x}} = \frac{\mathbf{x}}{||\mathbf{x}||_2} \tag{2.3}$$

[11] https://hdbscan.readthedocs.io/en/latest/how_to_use_epsilon.html (accessed 07/01/2021)

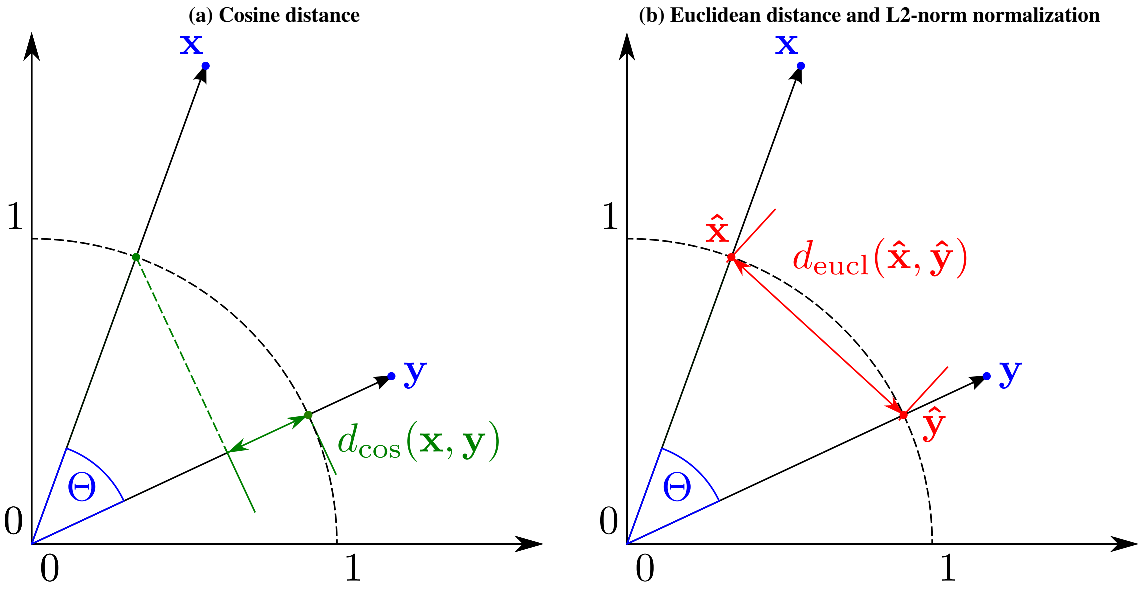

(a) Cosine distance
x
1
0
0
1
y
Θ
$d_{\cos}(\mathbf{x}, \mathbf{y})$
(b) Euclidean distance and L2-norm normalization
x
1
0
0
1
$\hat{\mathbf{x}}$
$d_{\text{eucl}}(\hat{\mathbf{x}}, \hat{\mathbf{y}})$
y
$\hat{\mathbf{y}}$
Θ

Fig. 2.6 Distance differences as graphical example. The cosine distance of two vectors **x** and **y** is based on the cosine of the angle Θ between the vectors (left picture). Since this information is not always available or the alternative calculation using the vectors only is to expensive for a high number of vectors, the euclidean distance can be used for approximation of cosine distance, with the precondition of a L2-norm of both vectors equal to one. L2-norm normalization was used on every vector scaling them to the unit sphere to match this condition (right picture).

Approximation is possible by euclidean distance calculation on vectors with an L2-norm of one Fig. 2.6. Due to the L2-norm normalization, according to Eq. 2.3, all vectors satisfy this condition (Eq. 2.4).

$$||\hat{\mathbf{x}}||_2 = ||\hat{\mathbf{y}}||_2 = 1 \tag{2.4}$$

The euclidean distance of the L2-norm normalized vectors, is in close relation to the cosine distance as proven in Eq. 2.5 and Fig. 3.2. Dividing the squared euclidean distance of the L2-norm normalized vectors by two, results in the cosine distance of the vectors [36].

$$\begin{aligned} d_{\text{eucl}}(\hat{\mathbf{x}}, \hat{\mathbf{y}})^2 &= ||\hat{\mathbf{x}} - \hat{\mathbf{y}}||_2^2 \\ &= (\hat{\mathbf{x}} - \hat{\mathbf{y}})^\top (\hat{\mathbf{x}} - \hat{\mathbf{y}}) \\ &= ||\hat{\mathbf{x}}||_2^2 + ||\hat{\mathbf{y}}||_2^2 - 2\hat{\mathbf{x}}^\top \hat{\mathbf{y}} \\ &= 2(1 - \cos(\Theta)) \\ &= 2 d_{\cos}(\hat{\mathbf{x}}, \hat{\mathbf{y}}) \end{aligned} \tag{2.5}$$

Thus, in this project, approximation of cosine distance with the euclidean distance metric, posterior to the normalization with the L2-norm, which scales the vectors to the unit sphere, was used (Eq. 2.6 and Fig. 2.6b).

$$d_{\text{eucl}}(\hat{\mathbf{x}}, \hat{\mathbf{y}}) = ||\hat{\mathbf{x}} - \hat{\mathbf{y}}||_2 \tag{2.6}$$

Due to the use of the euclidean distance metric in HDBSCAN, the calculation of the euclidean distance of the L2-norm normalized vectors is integrated into the mutual reachability distance calculation of HDBSCAN. The mutual reachability distance is the maximum of the described euclidean distance and the core distances of two L2-norm normalized vectors (Eq. 2.7 and Fig. 2.7). It is the main calculation of HDBSCAN used for clustering. The core distance is the minimum radius necessary to include $k = 1$ other vector around a given vector [43]. Threshold ε is used on the mutual reachability distances between the vectors Fig. 2.5. Due to the parameter choice of $k = 1$, the core distance can never be higher than the euclidean distance, making the mutual reachability distance always equal the euclidean distance of the L2-norm normalized vectors.

$$d_{\text{mreach}-k}(\hat{\mathbf{x}}, \hat{\mathbf{y}}) = \max\{\text{core}_k(\hat{\mathbf{x}}), \text{core}_k(\hat{\mathbf{y}}), d_{\text{eucl}}(\hat{\mathbf{x}}, \hat{\mathbf{y}})\} \tag{2.7}$$

2.5 Epsilon selection

To find an appropriate value for ε, two different methods were used and compared (Fig. 2.2 workflow **3** and **4**). The first method, the density based cluster validity (DBCV) exploration in Fig. 2.2 **G**, is based on repeated execution of hybrid HDBSCAN with different settings for ε and comparison by the DBCV. The DBCV is a calculation based on the minimum spanning tree, to estimate overall cluster density [47]. To enable the calculation hybrid HDBSCAN was performed with gen_min_span_tree=True setting.

Second method for ε exploration was performed using the Kneedle Algorithm on the linkage matrix created by the initially performed standard HDBSCAN clustering without hybrid setting (Fig. 2.2 **F** and **I**) [62]. With increasing cluster number, the distance threshold decreases in hierarchical clustering methods. This describes a decreasing curve of convex type, with distance threshold on the y- and cluster number on the x-axis. The knee is the number of clusters at the point in the polynomial representation of the curve with maximal acceleration. Polynomial representation was used to find the maximum accel-

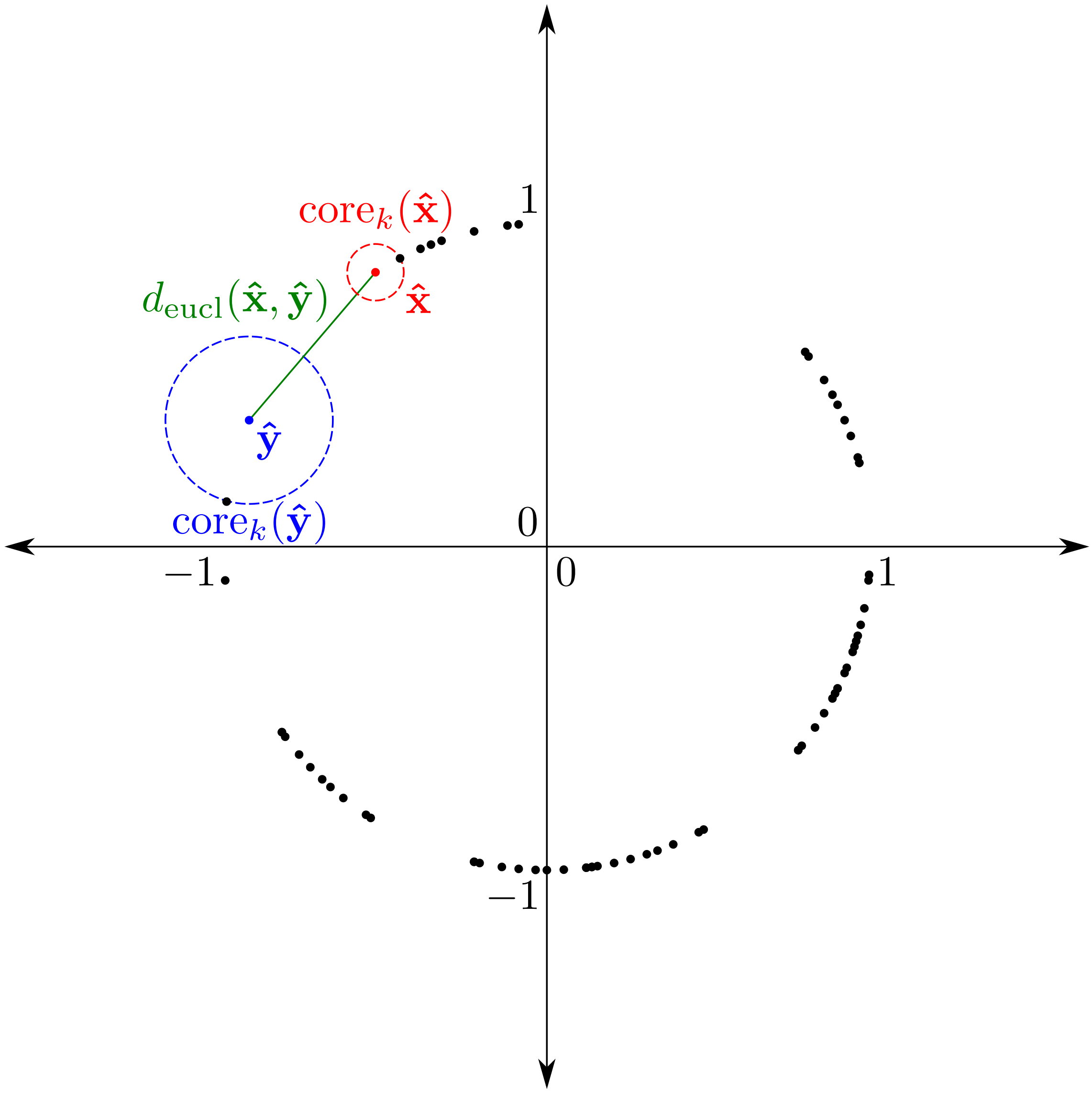

$\text{core}_k(\hat{\mathbf{x}})$
$d_{\text{eucl}}(\hat{\mathbf{x}}, \hat{\mathbf{y}})$
$\hat{\mathbf{x}}$
$\hat{\mathbf{y}}$
$\text{core}_k(\hat{\mathbf{y}})$
1
0
−1
0
1
−1

Fig. 2.7 Mutual reachability calculation. A low dimension representation of the calculation HDBSCAN performed in this project. To calculate the mutual reachability distance, the smallest radius is calculated to include exactly $k = 1$ vectors. This example should demonstrate the calculation for two L2-norm normalized vectors $\hat{\mathbf{x}}$ in blue and $\hat{\mathbf{y}}$ in red, with the value of one for k used in the project. The euclidean distance between these vectors is then calculated and compared to the radii. The maximum of both radii and the euclidean distance is the mutual reachability distance (Eq. 2.7). The used parameters make the mutual reachability distance always fall back to the euclidean distance, since the radius including $k = 1$ other vectors can only reach the euclidean distance at maximum.

eration of a more precise smoothed curve. Therefore, the settings `curve='concave'`, `direction='increasing'` and `interp_method='interp1d'` were used to find the optimal number of clusters. The optimal number of clusters was converted to the respective ε threshold. Knee point selection was restricted to a given area between one and a maximum value of 500. The maximum was chosen to include the area with the highest expected differences. The best Knee point was expected to be less than 100, a higher value was used to prevent bias creation by forcing the number of clusters to be maximal 100.

2.6 Alignments and vector calculations

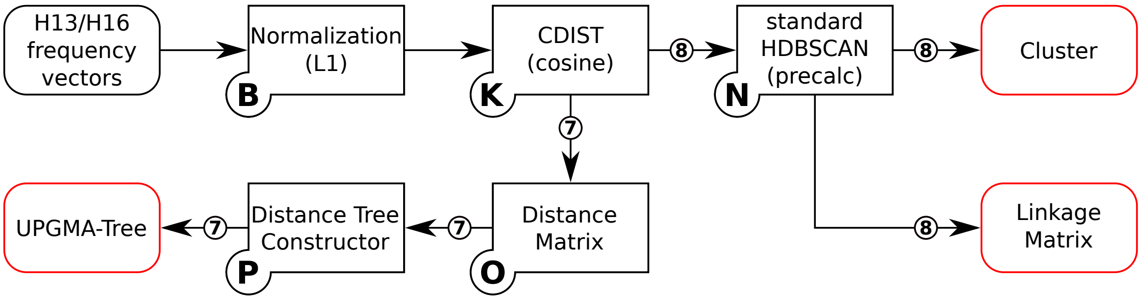


Fig. 2.8 Precalculation pipeline. For the precalculated trees the L1-normalized 7-mer frequency vectors distances were calculated and used for clustering by HDBSCAN (**B**, **K**, **N** and workflow **7**) and on the other hand processed with BioPython for unweighted pair group method with arithmetic mean (UPGMA) tree creation and visualization by ETE3 (**N** and workflow **8**). Results from this pipeline were visualized according to Fig. 2.3. A red frame denotes a result of the given workflow.

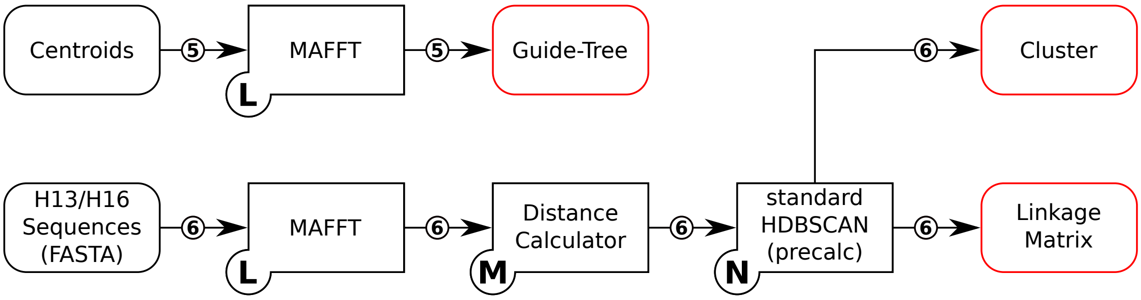


Fig. 2.9 Alignment pipeline. The sequences related to the centroid vectors were aligned using MAFFT resulting in the output as guide-tree visualized by ETE3 (**L** and workflow **5**). A small FASTA subset of H13/H16 was also aligned by MAFFT prior to evolutionary distance calculation with BioPython and clustering with HDBSCAN (**L**, **M**, **N** and workflow **6**). Results from this pipeline were visualized according to Fig. 2.3. A red frame denotes a result of the given workflow.

The labeled clustering trees were created by ETE3 with a newick file, generated according to a feature request, proposed for the SciPy[12] (Fig. 2.3 **J**) [27]. Clusters centroid vectors were selected by calculating euclidean distance between all the vectors of a cluster to each other. The vector with the smallest mean distance was declared as centroid (Fig. 2.3 **K**). The precalculated trees were created using cosine distance calculation on the L1-norm normalized 7-mer frequency vectors of the segment 4 H13 and H16 sequences of the FASTA (Fig. 2.8 **B** and **K**). The calculated distances were used for UPGMA tree building with

[12] https://github.com/scipy/scipy/issues/8274 (accessed 06/02/21)

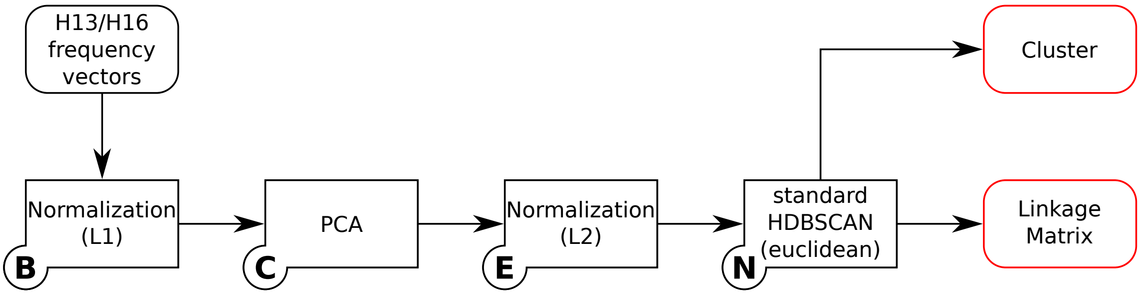


Fig. 2.10 Simplified clustering pipeline. For the simplified clustering on the sequence subset, the the L1-normalized 7-mer vectors were reduced with `PCA` (**B** and **C**). Following the dimension reduction, the vectors were normlized again acording to L2-norm and clustered by `HDBCSCAN` (**E** and **N**). Results from this pipeline were visualized according to Fig. 2.3. A red frame denotes a result of the given workflow.

`BioPython` (Fig. 2.9 **O** and **P**). The vectors were also clustered by standard `HDBSCAN` without hybrid setting, using the precalculated distances with `metric='precalculated'`.

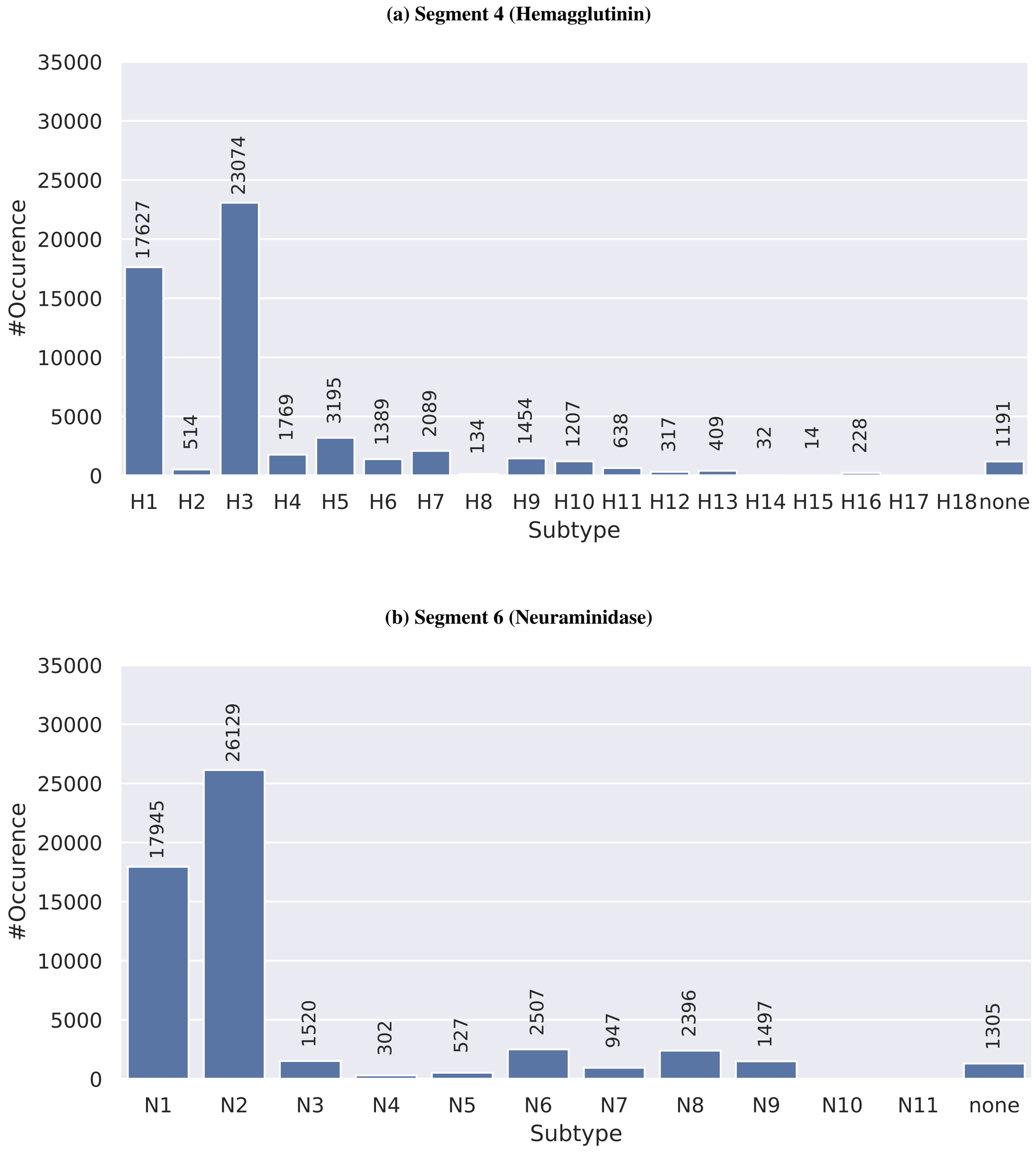
`MAFFT` was used for multiple sequence alignments (MSAs) and guidetree creation with `treeout=True` on the centroid sequences and on the FASTA subset containing H13 and H16 sequences of segment 4 (Fig. 2.9 **L**) [33]. The MSA of the FASTA subset was converted to evolutionary distances with `BioPython` and clustered by standard `HDBSCAN` without hybrid setting (Fig. 2.9 **M** and **N**). Pairwise alignments in Sec. 3.6 were performed using `BioPython`.

3 Results and Discussion

For a basic overview of the sequences in the FASTA file, the numbers of sequences of segment 4 and 6, related to the subtypes of hemagglutinin (HA) and neuraminidase (NA) were counted and visualized (Fig. 3.1). Hereby, the high number of segment 4 sequences of subtype H1 and H3 in Fig. 3.1a and segment 6 sequences of N1 and N2 in Fig. 3.1b was noticeable. Furthermore, the similarity of the number of sequences of segment 4 subtype H1 and N1, as well as of segment 6 H3 and N2 was also remarkable. The similarity seem to originate from the frequency of subtype combinations H1N1 and H3N2 of the full genomes, which is also in line with the most prominent subtypes of *Influenza A Virus* (IAV) mentioned in Deng et al. [15]. In addition to that, a high number of not classified sequences named „none“ is present in the data. These unclassified sequences were possibly sequenced without prior subtype testing or are in fact sequences that do not belong to either of the known subtypes [2].

As described in Chap. 2, the clustering was performed multiple times to compare different methods for their quality and efficiency of IAV clustering. The four methods were compared directly to each other in the following. For better separation, the four methods were called by their abbreviation, to be found in Table 2.3. As a short reminder, PD used direct reduction to 30 components with `PCA`, following ε exploration with the density based cluster validity (DBCV) and PK used the same reduction but exploration with the Kneedle Algorithm instead. Method UD and UK were performed in a similar way but with reduction to 100 components by `PCA` and to 30 afterwards by `UMAP`. The numbered workflows in Fig. 2.1, Fig. 2.2 and Fig. 2.3 represent the different methods.

To choose a reasonable number of components for the `PCA`, mentioned in Sec. 2.3, the 7-mer frequency vectors of segment 4 were reduced by running `PCA` with different settings (Fig. 2.1 and Table 3.1). The sum of explained variance for every setting were calculated and listed in Table 3.1. Because of the high increase of explained variance from 10 to 20 components with 58.7% to 68.7% and likewise to 30 components with 73.9% explained variance, a value of 30 was used as default in this project. A fairly small value was used, due to increasing computational effort of the `PCA` implementation and also to preserve the usability of spanning tree calculation with `HDBSCAN`. Calculation of the spanning tree by



(a) Segment 4 (Hemagglutinin)

(b) Segment 6 (Neuraminidase)

Fig. 3.1 Antigen subtype frequency. The number of occurrences of segment 4 and segment 6 sequences related to HA and NA subtypes was counted. Sequences related to subtypes H1 and H3 were the most frequent ones from segment 4 with a total number of 17627 and 23074. Subtype H5 was also slightly over-represented with 3195 sequences. 1191 sequences from segment 4 in the data had no classification and are listed as „none“. Likewise were sequences related to subtypes N1 and N2 the most frequent ones from segment 6 with a total number of 17945 and 26129. 1305 sequences from segment 6 in the data had no classification and are also listed as „none“.

HDBSCAN is, with the preferred settings, only possible up to ≈ 50 components. For better comparison of the methods, the final number of components of the UMAP reduction in UD and UK was also set to 30. Therefore, clustering with all the methods was performed on vectors with 30 components.

Table 3.1 Explained variance by different PCA settings. The explained variance of specific PCA component settings used on the 7-mer frequency vectors of segment 4. A result of 0.587, as for extracted 10 components, represents 58.7% of the variance explained by the first 10 components of the vector. The more components were extracted the lower the increase in additional explained variance [56].

#Components	**Var**(X)
10	0.587
20	0.687
30	0.739
40	0.771
50	0.793
60	0.809
70	0.821
80	0.831
90	0.839
100	0.846

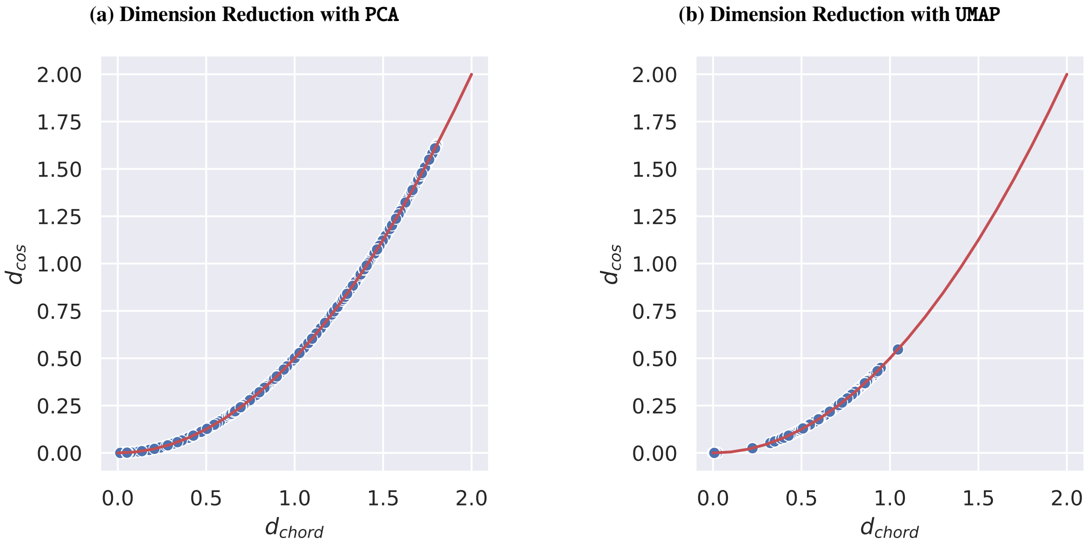


Fig. 3.2 Approximation proof on real data. Calculations for the L2-norm normalized euclidean and the cosine distances were performed on subsets posterior to the PCA and the UMAP workflows and compared. In both cases the difference is exactly matching the relation of squared euclidean distance divided by two equals the cosine distance, as described in Sec. 2.4, further proving the relation of the distances. Crowding of the points in the right graph, representing reduction with UMAP, most likely can be explained by the embedding mechanism of the tool, that further reduces the distance of similar vectors [44].

For the graphical comparison of the demonstrated mathematical proof of the close relation of the euclidean distance of the L2-norm normalized vectors and the cosine distance (Sec. 2.4), subsets of the real data were used. In Fig. 3.2a, a subset of 100 7-mer frequency vectors of segment 4 were reduced by PCA, L2-norm normalized and compared to each other by euclidean and cosine distance, as described in Sec. 2.3 and Sec. 2.4. All the calculated distances for euclidean distance were then plotted against the cosine distances. The red line indicates the relation as calculated in Sec. 2.4. All the points are arranged exactly along the red line which confirmed the calculation and relation of the used euclidean distance on L2-norm normalized vectors to the cosine distance, that was intended to be used in the first place. The same procedure was repeated with the use of the `UMAP` workflow, resulting in the same graphical proof (Fig. 3.2b). In Fig. 3.2b, all the points are crowded much more to a dense area of the curve. This is most likely reasoned by the number of neighbors setting and the embedding behavior of `UMAP` and will be discussed in Sec. 3.5.

3.1 Method selection

Clustering with each of the four methods, resulted in a table containing the used settings and a summary of the clustering. For better visualization, the tables were combined based on the exploration method. Table 3.2 contains the results of the two methods using the Kneedle Algorithm (PK and UK) and Table 3.3 the results that were based on using the DBCV (PD and UD). Every segment of IAV was clustered by each of the four methods. By result comparison of methods using the ε exploration by the Kneedle Algorithm (PK and UK) in Table 3.2, a major difference in the number of raw clusters stood out. Hybrid `HDBSCAN` clustering of the only `PCA` reduced 7-mer vectors (PK) resulted in around 60 to 70 raw clusters (after `DBSCAN` part) and subsequently 40 to 50 final clusters (after standard `HDBSCAN` part) per segment. The number of final clusters was relative close to the state of the art subtype classification with 18 HA and 11 NA antigen subtypes [2]. The UK method, using `UMAP` for reduction, resulted in a higher number of clusters with no difference in raw and final cluster number. Therefore, the hybrid `HDBSCAN` clustering only used the `DBSCAN` part. This can be explained by the overall higher ε threshold compared to the `PCA` version (PK). By a higher ε, more points were included in the `DBSCAN` part of the hybrid clustering. In addition to that, the embedding behavior of `UMAP` probably affected the position of the vectors, decreasing the distances inside of groups of vectors, thereby, leaving less vectors out by the `DBSCAN` part. With the `UMAP` reduction method (UK), all vectors of every segment could be clustered, thus, zero vectors were left out unclustered. The approach using `PCA` alone (PK) was unable to cluster the vectors of around 10 to 30 sequences per segment. However the number of unclustered with PK for, e. g., segment 4 was 6 of 56617 used sequences and, therefore, neglectable $\approx 0.01\%$. As a comparison, 1191 sequences segment 4 are declared unclassified by the current subtype convention making $\approx 2\%$.

Comparison of the methods PK and UK to the ones using the DBCV for ε exploration, instead of the Kneedle Algorithm (Table 3.3, did not result in much difference for the `UMAP`

Table 3.2 Clustering results with the Kneedle Algorithm. The results of the clustering methods using the Kneedle Algorithm (PK and UK). Listed is every used segment with the number of raw clusters and the final cluster number by hybrid clustering with the given value of ε. The mixed cluster numbers of H and N denotes number of clusters that contained vectors related to more than one subtype. The variance was calculated as the sum of the explained variance by the PCA.

		#Cluster			#Mixed				
Method	**Segment**	**Final**	**Raw**	**Normalized**	**H**	**N**	**#Unclustered**	ε	**Var**(X)
PK	1	29	65	0.128	20	21	30	0.258	0.773
	2	39	63	0.124	18	18	21	0.285	0.756
	3	42	72	0.142	20	20	20	0.310	0.777
	4	56	67	0.132	2	41	6	0.191	0.739
	5	32	71	0.140	18	20	27	0.277	0.800
	6	44	58	0.114	30	3	9	0.215	0.754
	7	41	72	0.142	20	21	22	0.348	0.822
	8	37	61	0.120	18	19	20	0.311	0.826
UK	1	255	255	0.509	103	107	0	0.064	0.859
	2	222	222	0.443	88	89	0	0.066	0.849
	3	268	268	0.535	92	94	0	0.055	0.865
	4	266	266	0.531	2	104	0	0.043	0.846
	5	309	309	0.617	94	95	0	0.056	0.882
	6	271	271	0.541	90	4	0	0.035	0.855
	7	437	437	0.874	100	112	0	0.093	0.903
	8	360	360	0.719	111	111	0	0.089	0.899

reduction with DBCV exploration method (UD). The numbers of clusters found were a little higher compared to the results with the Kneedle Algorithm exploration (UK). Using the DBCV to find the optimal ε with the PCA workflow (PD) on the other hand changed the results drastically in comparison to PK. The numbers of final clusters were between 9000 and 12000 depending on the segment, with the exception of segment 4 and the raw cluster numbers were nearly equivalent to the total number of sequences for the other segments. Also the numbers of unclustered sequences were increased by a major amount to around 20% for most segments, making the method in any case unusable for IAV clustering. Only the clustering of segment 4 with DBCV ε exploration (PD) seemed to be as stable as with the Kneedle Algorithm (PK). The numbers of clusters for the other segments were higher, because the DBCV method was searching for the ε value setting that results in the highest DBCV possible. In all the segments except 4 this resulted in a value of zero. Hybrid clustering with a ε value of zero results in clustering with the standard HDBSCAN part only, without prior DBSCAN and is, considering the amount of unclustered sequences, not suited for IAV clustering. Based on the results in Table 3.2 and Table 3.3, the best method for IAV clustering seemed to be the combination of PCA and the Kneedle Algorithm (PK). The method resulted in an appropriate number of clusters and was the only combination using all the benefits of the hybrid HDBSCAN clustering.

Table 3.3 Clustering results with the DBCV. The results of the clustering methods using the DBCV (PD and UD). Listed is every used segment with the number of raw clusters and the final cluster number by hybrid clustering with the given value of ε. The mixed cluster numbers of H and N denotes number of clusters that contained vectors related to more than one subtype. The variance is calculated as the sum of the explained variance by the PCA.

		#Cluster		#Mixed					
Method	**Segment**	**Final**	**Raw**	**H**	**N**	**#Unclustered**	ε	**DBCV**	**Var**(X)
PD	1	11,599	55,436	1,152	1,225	13,548	0.000	0.404	0.773
	2	11,596	55,292	1,067	1,169	13,457	0.000	0.417	0.756
	3	11,521	55,351	1,080	1,168	13,017	0.000	0.422	0.777
	4	51	58	2	38	4	0.200	0.572	0.739
	5	10,717	32,818	1,074	1,158	11,478	0.000	0.479	0.800
	6	11,143	34,600	713	3	12,065	0.000	0.455	0.754
	7	8,866	55,620	1,119	1,227	8,753	0.000	0.513	0.822
	8	9,189	55,563	1,125	1,225	9,794	0.000	0.493	0.826
UD	1	279	279	110	116	0	0.041	0.860	0.859
	2	242	242	92	93	0	0.052	0.838	0.849
	3	265	265	92	94	0	0.060	0.897	0.865
	4	261	261	2	103	0	0.049	0.891	0.846
	5	283	283	87	88	0	0.094	0.868	0.882
	6	255	255	85	4	0	0.061	0.919	0.855
	7	462	463	103	115	0	0.050	0.869	0.903
	8	364	364	111	111	0	0.079	0.868	0.899

To backup this assumption, the results were visualized for better understanding and analysis. Since investigation of all segments would overfill this section, only the methods PK, UK and UD and their clustering behavior on segment 4 were discussed in detail. As described, the PCA method in combination with the DBCV (PD) used only standard HDBSCAN, resulting in a very high number of clusters and 20% unclustered sequences. Therefore, the method was already rejected and not included in the following discussions. Nevertheless, similar graphics for PD with segment 4, as well as, all used method for the other segments can be found in the Appendix A.

A big difference in the cluster size distribution stood out, when comparing the method with PCA (PK) to the ones with UMAP (UK and UD). Only the method using PCA and the Kneedle Algorithm created clusters with more than 10000 vectors (Fig. 3.3c). The different cluster sizes using UMAP were spreaded more equally with no cluster containing 3000 vectors or more (Fig. A.2c and Fig. A.3c). Since the UMAP methods also used prior PCA reduction, the major difference was the additional use of UMAP (Sec. 3.5). Therefore, the difference in cluster size distribution was most likely caused by UMAP itself. As already mentioned UMAP not only reduces the dimension of the data but also changes the position of the vectors in the embedded dimension according to the used settings. The n_neighbors=100 setting seemed to be most likely the cause of a change of this magnitude. By this high number

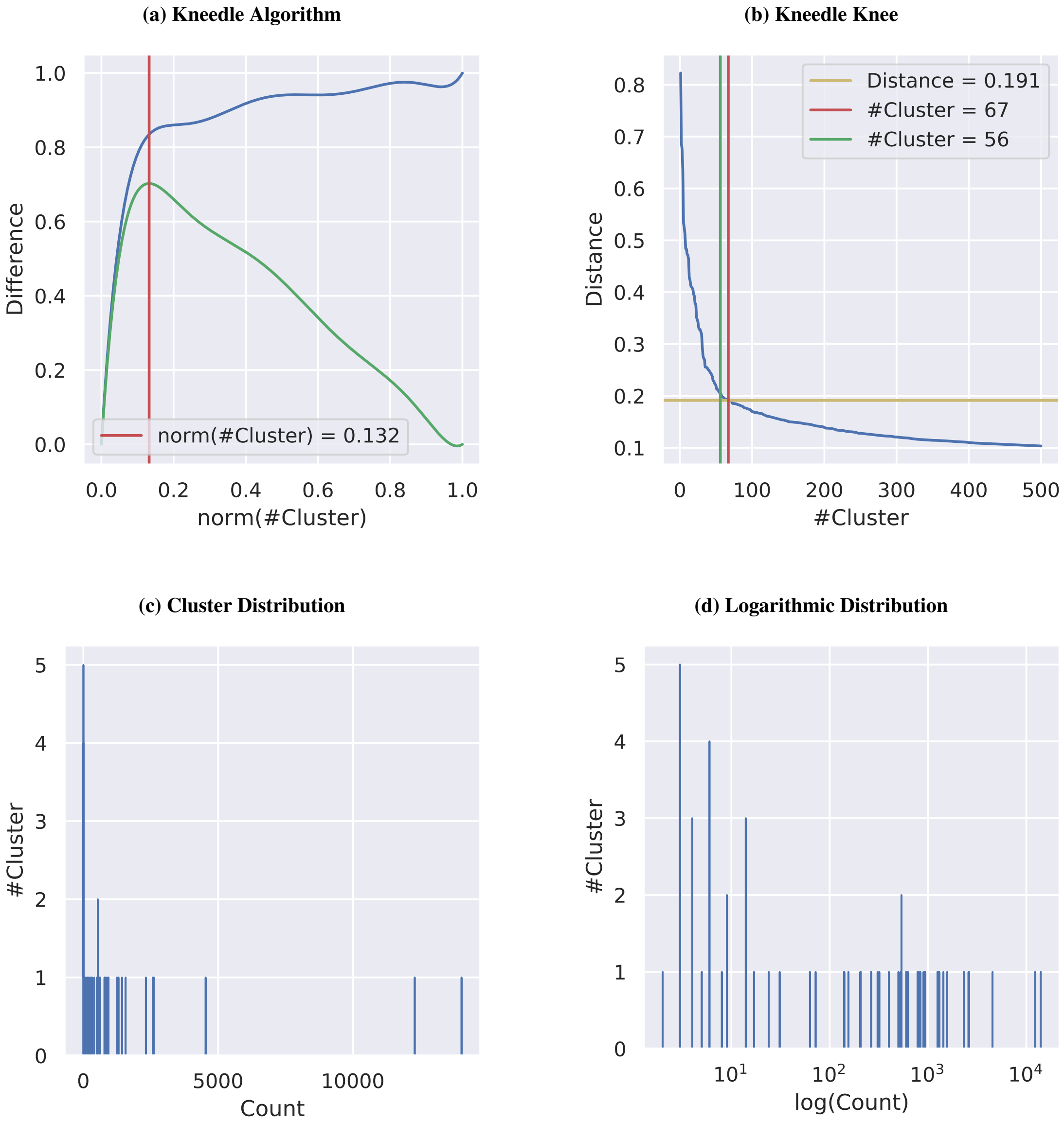


Fig. 3.3 Clustering of segment 4 with PK. Segment 4 clustering using the combination of PCA and the Kneedle Algorithm (PK) results in the given figure. The green curve in the top left subfigure describes the change of the distance in the single linkage tree with increasing normalized cluster number and, therefore, the location of the knee, at the maximum, highlighted by the red line. The blue line represents the inverse polynomial representation of the blue line in top right subfigure. The top right subfigure shows the absolute relation of the distance in the single linkage tree to the total number of clusters as the blue line. The red line, indicates the number of raw clusters, by the DBSCAN part of the hybrid HDBSCAN clustering and the final cluster number in green. The yellow line describes the threshold, extracted from the knee and, therefore, the ε value used to perform the hybrid clustering. The normalized cluster number in the red line in the top left subfigure is equivalent to the raw cluster number in the top right subfigure. The bottom subfigures give information about the distribution of the clusters sizes, by plotting the number of clusters containing a given counted number of sequences in continuous and logarithmic scale.

number of neighbors, the vectors were more crowded in large groups to support the bigger picture of the data and, therefore, seemed to build more crowded clusters of similar sizes.

With the distribution of segment 4 sequences in the data in Fig. 3.1a in mind, it was expected that a distribution of cluster sizes in segment 4 clustering would in fact approximate the distribution of the sequences in the former one. Therefore, a clustering with PCA in combination with Kneedle Algorithm exploration seemed to give the expected results and appeared again as the best method for IAV clustering. Taking also the relation of the cluster number and the distance into consideration, Fig. 3.3b indicated continuous merging of clusters with decreasing cluster number in the linkage tree. Thereby, exponential behavior, with a knee point that is easily distinguishable, even without computationally methods, developed. The behavior is not present in Fig. A.2b and Fig. A.3b in a similar degree, as at least two knees occurred.

The results of the method using PCA in combination with the Kneedle Algorithm (PK) were visualized by the clustering tree in Fig. 3.4. Labeling of the tree was performed based on the HA antibody subtype, as shown in Fig. 3.1a. Therefore, clusters only containing sequences of a given subtype were labeled as such. If a cluster only contained sequences of one subtype, plus some not classified sequences, the not classified sequences were declared as the subtype too. That way, a clear presentation of the subtype distribution by labeling was possible, since the not classified sequences were very likely to actually belong to an existing subtype when clustered that way. If they actually did not belong to a existing subtype, a cluster only containing these not classified sequences would most likely have occurred, which did not happen. Furthermore, the chance to eventually break the classification of subtypes, by declaring not classified sequences to a existing subtype was not given. The clusters were not based on subtypes in any way and the visualization was only for guidance and not possible for any segments other than 4 and 6 anyway. Also, without this assignment of the not classified sequences, no presentation would be possible since they were present in a high number and distributed over mostly all clusters.

If a cluster contained sequences of more than one subtype plus some not classified sequences, no labeling was performed, since the cluster was not homogeneous for one subtype and a declaration of the unclassified was not possible. In the following the term vector will be mostly replaced by the term sequence, to aid the discussion in terms of tree placement and subtype relation. Since the vectors represented the sequences, the terms were linked to each other and used as synonyms.

The prime example of the not classified sequence annotation was the yellow labeled cluster 29 of subtype H9 (Fig. 3.4 **A**). While the vectors of all sequences of subtype H9 were accumulated in this single cluster, the cluster size of 1569 was bigger than the number of H9 sequences in Fig. 3.1a with 1454. This was justified by the presence of

Table 3.4 Unclassified sequences in segment 4 cluster 29 with PK. The multiple sequence alignments (MSAs) mean distance of the given sequences in comparison to a sample of H9 sequences of the same cluster and a sample of unclassified sequences from other clusters was calculated. Only the first 20 columns are presented, the full table can be found in the projects GitHub Repository[1].

Accession	H9	unclassified
>CY125733	0.167	0.499
>KP286620	0.104	0.509
>KP286635	0.105	0.510
>CY206227	0.166	0.499
>MG042357	0.140	0.500
>MG957616	0.139	0.501
>MG957513	0.138	0.501
>MG957519	0.139	0.500
>MH791733	0.141	0.501
>MN209446	0.145	0.501
>MN209378	0.145	0.502
>KP416339	0.104	0.509
>KP416375	0.104	0.510
>KP416404	0.104	0.509
>KP415847	0.104	0.510
>KP416464	0.105	0.511
>KP415453	0.106	0.509
>KP286584	0.112	0.503
>KP414685	0.113	0.506
>KP415520	0.105	0.511
...	...	...

115 not classified sequences in the cluster, which were declared as H9, since only H9 and unclassified sequences were present in the cluster. The declaration of the unclassified sequences to be the H9 sequences was supported by the very small evolutionary distance of the unclassified sequences to a sample of H9 sequences from the same cluster (Table 3.4). The sequences were also compared to unclassified sequences from other clusters to prove the smaller evolutionary distance to H9 sequences. This comparison was performed to prove that, even when only used for visualization reasons, the annotation of not classified sequences in the described way was most likely appropriate.

The number of clusters for a given subtype of HA seemed to correspond roughly to the overview of sequences in Fig. 3.1a. Clusters of very low represented subtypes, like H15, contained mostly all the subtypes sequences, while the high represented subtypes sequences, like H1 and H3, were spreaded over more clusters.

Striking anomalies divergent from the expected nearly uniform allocation of the subtypes in Fig. 3.3 are annotated by **B**, **C** and **D** and will be discussed in the following. For final

[1] https://github.com/ahenoch/Masterthesis.git

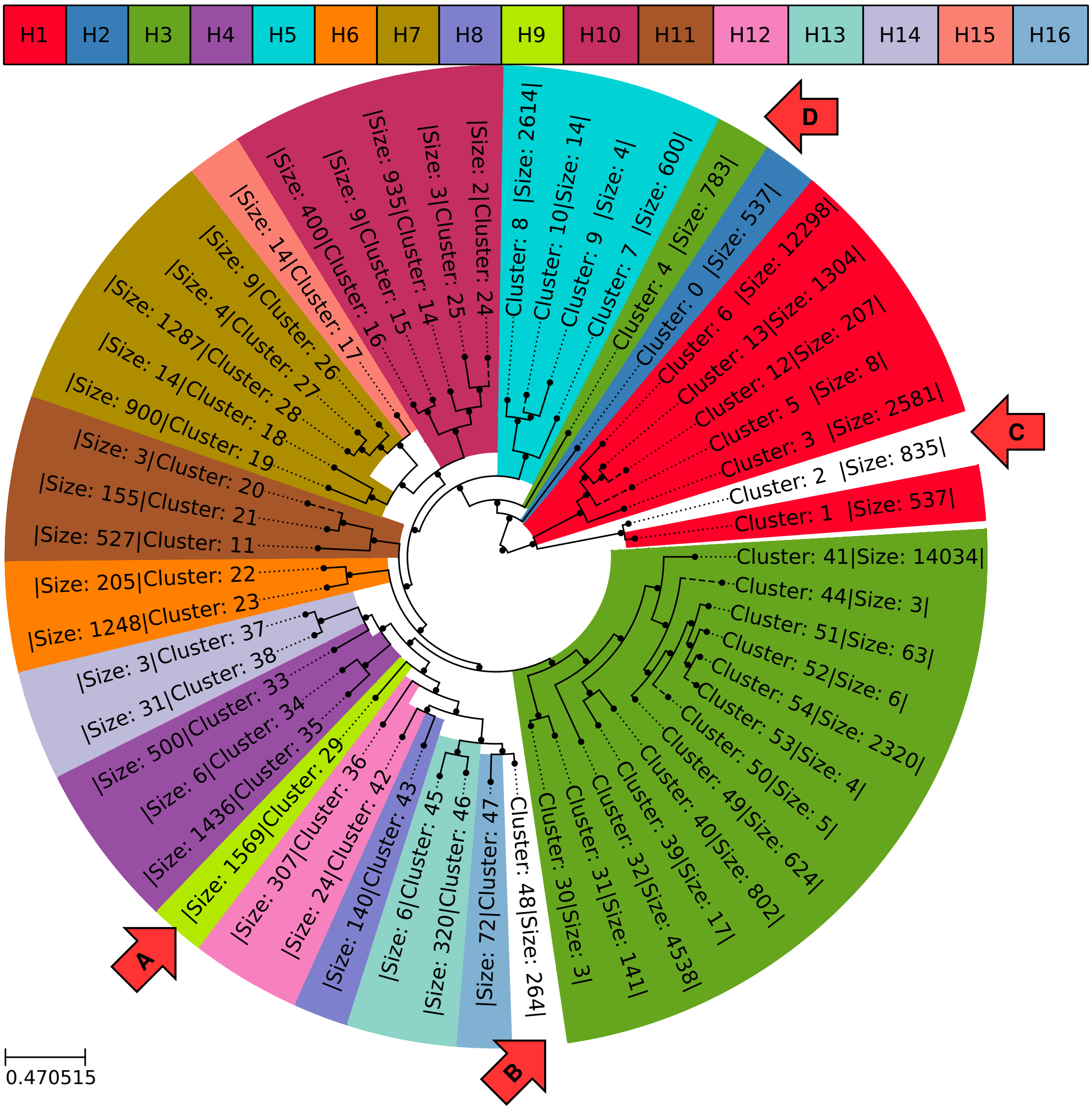


Fig. 3.4 Clustering tree of segment 4 with PK. The clustering tree of segment 4 clustering, using the combination of PCA and the Kneedle Algorithm (PK) (Fig. 3.3). The labeling of the clusters in the tree is based on the subtype of the contained sequences. Unclassified sequences of a cluster were reclassified as a given subtype if sequences of only this subtype were present in the cluster, in addition to the unclassified ones. Unlabeled clusters contain sequences from at least two subtypes and zero or more unclassified sequences. Two clusters were mixed since they contained sequences of more than one subtype (Table 3.2). These non homogeneous clusters are marked by **B** and **C**. The cluster 29 marked by **A** was an example for a cluster consisting of all sequences from a given subtype. The misplaced cluster from subtype H3 is marked by **D**. Dotted lines in the tree indicate the same host.

acceptance of the method PK as the prime IAV clustering method proposed in this project, the clustering tree was compared to a similar one created on the results from method UK (Fig. A.4). While the labeling of the clustering tree of method PK resemble the subtype classification of IAV very closely, no recognizable subtype separation is present in the clustering tree of UK.

3.2 Database annotation errors

To evaluate the anomalies in Fig. 3.3 **B** and **C**, an evolutionary distance was calculated by MSA for every eventually misplaced sequence to ten other sequences (Sec. 2.6). A sample of five sequences from the same cluster, related to the dominant subtype of the cluster, and a sample of five sequences with subtype equal to the misplaced sequence but from other clusters. The mean of the evolutionary distance was then calculated for both sample comparisons independently to rate the assignment and reveal possible misannotations in the Influenza Research Database (IRD).

Table 3.5 Anomalies in segment 4 cluster 2 with PK. The MSAs mean distance of the given sequences in comparison to a sample of H1 sequences of the same cluster and a sample of H10 sequences from other clusters was calculated.

Accession	H1	H10
>MK237334	0.049	0.488

Table 3.6 Anomalies in segment 4 cluster 48 with PK. The MSAs mean distance of the given sequences in comparison to a sample of H16 sequences of the same cluster and a sample of H13 sequences from other clusters was calculated. Only the first 20 columns are presented, the full table can be found in the projects GitHub Repository[2].

Accession	H16	H13
>MH498778	0.278	0.217
>MF682848	0.299	0.244
>KX979327	0.298	0.238
>KX978913	0.300	0.241
>KX979541	0.300	0.240
>KX978076	0.301	0.241
>KX978980	0.299	0.242
>KX979063	0.298	0.238
>KX978876	0.299	0.239
>KX979544	0.301	0.241
>KX979380	0.300	0.240
>MF682844	0.300	0.240
>KX978929	0.298	0.238
>MF575207	0.297	0.239
>KR087564	0.290	0.251
>MF147289	0.298	0.241
>MF147869	0.298	0.241
>CY185569	0.287	0.258
>CY185489	0.287	0.250
>MF461180	0.297	0.244
...	...	...

[2] https://github.com/ahenoch/Masterthesis.git

In case of Fig. 3.3 **C**, a single sequence with subtype H10 was classified as belonging to cluster 2, which other than that, completely consists of H1 and unclassified sequences. By investigation on this possible misplacement, comparison with MSA was used. The results for this comparison in Table 3.5 pointed to the fact, that the as H10 annotated sequence with accession MK237334 is related to subtype H1. The mean of evolutionary distance based on MSAs with the sequence and a sample of cluster 0 H1 sequences was very low. Considering the large size of cluster 0, a higher difference was expected, pointing in direction of many very similar sequences in the cluster. In addition, the evolutionary distance of the sequence in comparison to a sample of random H10 sequences was much higher (Table 3.5). Furthermore, only this sole sequence, annotated as subtype H10, was present in a cluster of over 900 sequences of H1, with a very low evolutionary distance to a sequence sample of the cluster, rendering the error most likely as a misannotation.

When comparing the distances for the same calculation performed on Fig. 3.3 **C** in Table 3.6, no decision for misannotation can be made. The dominant subtype in the cluster 48 is H16 but the sequences of subtype H13 that seem to be misplaced in the cluster had a smaller distance to the sample of sequences from subtype H13. The difference in distance to sample sequences of H13 as well as to sequences of H16 gave indeed no clear finding. Both results are quite similar and the misclustered sequences seemed to share much sequence similarity with both subtypes. The misclustered sequences in cluster 48 possibly pointed to a more complex classification. Cluster 48 remained a mixed cluster with many sequences from subtype H13 and H16 and, thus, was treated as clustering error. Therefore, subtype H13 and H16 will be the focus of investigations of the clustering behavior in the following sections to reveal possible subdivisions responsible for the clustering error.

The last error involved cluster 4, that was homogeneous for subtype H3 but split off the other H3 clusters by nearly all the non H3 clusters Fig. 3.3 **D**. By evaluation of the clusters relation with a MSA created guidetree generated from the centroid vectors sequences, the error persists (Fig. 3.5). Every of the 55 clusters had a sequence intended to best represent the whole cluster, the centroid sequence, calculated as described in Sec. 2.6. When using the guidetree as comparison, the uniform labeled distribution stood out. Even the centroid of the mentioned cluster 4 is arranged in a line with all the centroids of clusters homogeneous for subtype H3. This subset of centroid sequences used for the guidetree was possibly to small for a sure proof but still, the arrangement pointed to a clear subtype separation. Therefore, cluster 4 and 48 (Fig. 3.3 **C** and **D**) remained as identified clustering mistakes and will be further examined in the following.

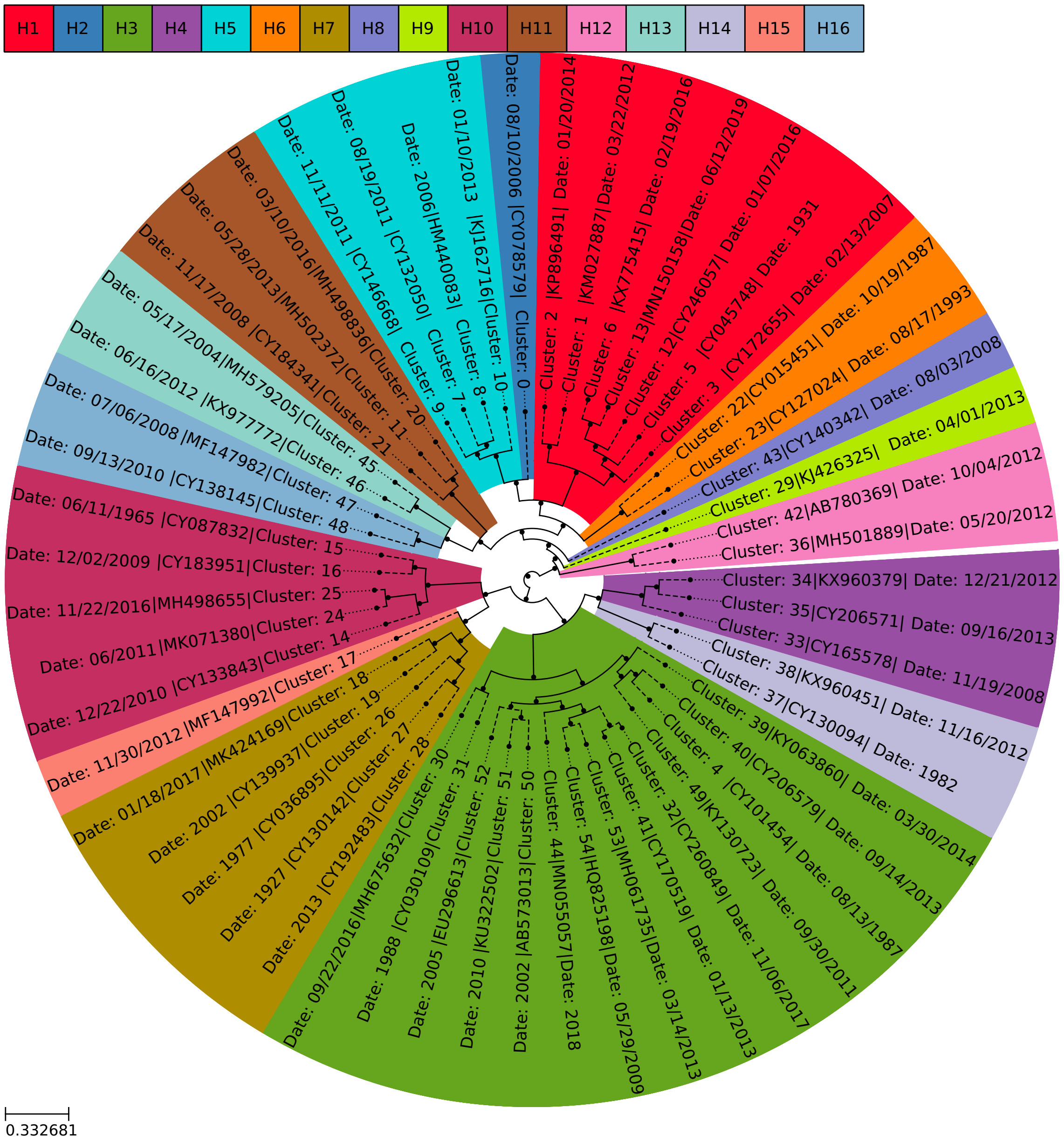

H1
H2
H3
H4
H5
H6
H7
H8
H9
H10
H11
H12
H13
H14
H15
H16
Date: 08/10/2006 |CY078579| Cluster: 0
Cluster: 2 |KP896491| Date: 01/20/2014
Cluster: 1 |KM027887| Date: 03/22/2012
Cluster: 6 |KX775415| Date: 02/19/2016
Cluster: 13|MN150158|Date: 06/12/2019
Cluster: 12|CY246057| Date: 01/07/2016
Cluster: 5 |CY045748| Date: 1931
Cluster: 3 |CY172655| Date: 02/13/2007
Cluster: 22|CY015451| Date: 10/19/1987
Cluster: 23|CY127024| Date: 08/17/1993
Cluster: 43|CY140342| Date: 08/03/2008
Cluster: 29|KJ426325| Date: 04/01/2013
Cluster: 42|AB780369| Date: 10/04/2012
Cluster: 36|MH501889|Date: 05/20/2012
Cluster: 34|KX960379| Date: 12/21/2012
Cluster: 35|CY206571| Date: 09/16/2013
Cluster: 33|CY165578| Date: 11/19/2008
Cluster: 38|KX960451| Date: 11/16/2012
Cluster: 37|CY130094| Date: 1982
Cluster: 39|KY063860| Date: 03/30/2014
Cluster: 40|CY206579| Date: 09/14/2013
Cluster: 4 |CY101454| Date: 08/13/1987
Cluster: 49|KY130723| Date: 09/30/2011
Cluster: 32|CY260849| Date: 11/06/2017
Cluster: 41|CY170519| Date: 01/13/2013
Cluster: 53|MH061735|Date: 03/14/2013
Cluster: 54|HQ825198|Date: 05/29/2009
Cluster: 44|MN055057|Date: 2018
Date: 2002 |AB573013|Cluster: 50
Date: 2010 |KU322502|Cluster: 51
Date: 2005 |EU296613|Cluster: 52
Date: 1988 |CY030109|Cluster: 31
Date: 09/22/2016|MH675632|Cluster: 30
Date: 2013 |CY192483|Cluster: 28
Date: 1927 |CY130142|Cluster: 27
Date: 1977 |CY036895|Cluster: 26
Date: 2002 |CY139937|Cluster: 19
Date: 01/18/2017 |MK424169|Cluster: 18
Date: 11/30/2012 |MF147992|Cluster: 17
Date: 12/22/2010 |CY133843|Cluster: 14
Date: 06/2011|MK071380|Cluster: 24
Date: 11/22/2016|MH498655|Cluster: 25
Date: 12/02/2009 |CY183951|Cluster: 16
Date: 06/11/1965 |CY087832|Cluster: 15
Date: 09/13/2010 |CY138145|Cluster: 48
Date: 07/06/2008 |MF147982|Cluster: 47
Date: 06/16/2012 |KX977772|Cluster: 46
Date: 05/17/2004|MH579205|Cluster: 45
Date: 11/17/2008 |CY184341|Cluster: 21
Date: 05/28/2013|MH502372|Cluster: 11
Date: 03/10/2016|MH498836|Cluster: 20
Date: 11/11/2011 |CY146668| Cluster: 9
Date: 08/19/2011 |CY132050| Cluster: 7
Date: 2006|HM440083| Cluster: 8
Date: 01/10/2013 |KJ162716|Cluster: 10
0.332681

Fig. 3.5 Centroid guidetree of segment 4 with PK. The guidetree was created by building a MSA on the centroid sequences of clusters resulting from the PCA and Kneedle Algorithm (PK) workflow. The labeling of the tree is based on the related subtype of the centroid sequence used for the MSA. The leaves are annotated by the used centroid sequences accession and the cluster it represents.

3.3 *k*-mer representation quality

Investigation on the anomalies resulted in two persistent clustering errors (Fig. 3.3 **B** and **D**). To evaluate if the method is suitable for the clustering of IAV possible error sources will be discussed in the following.

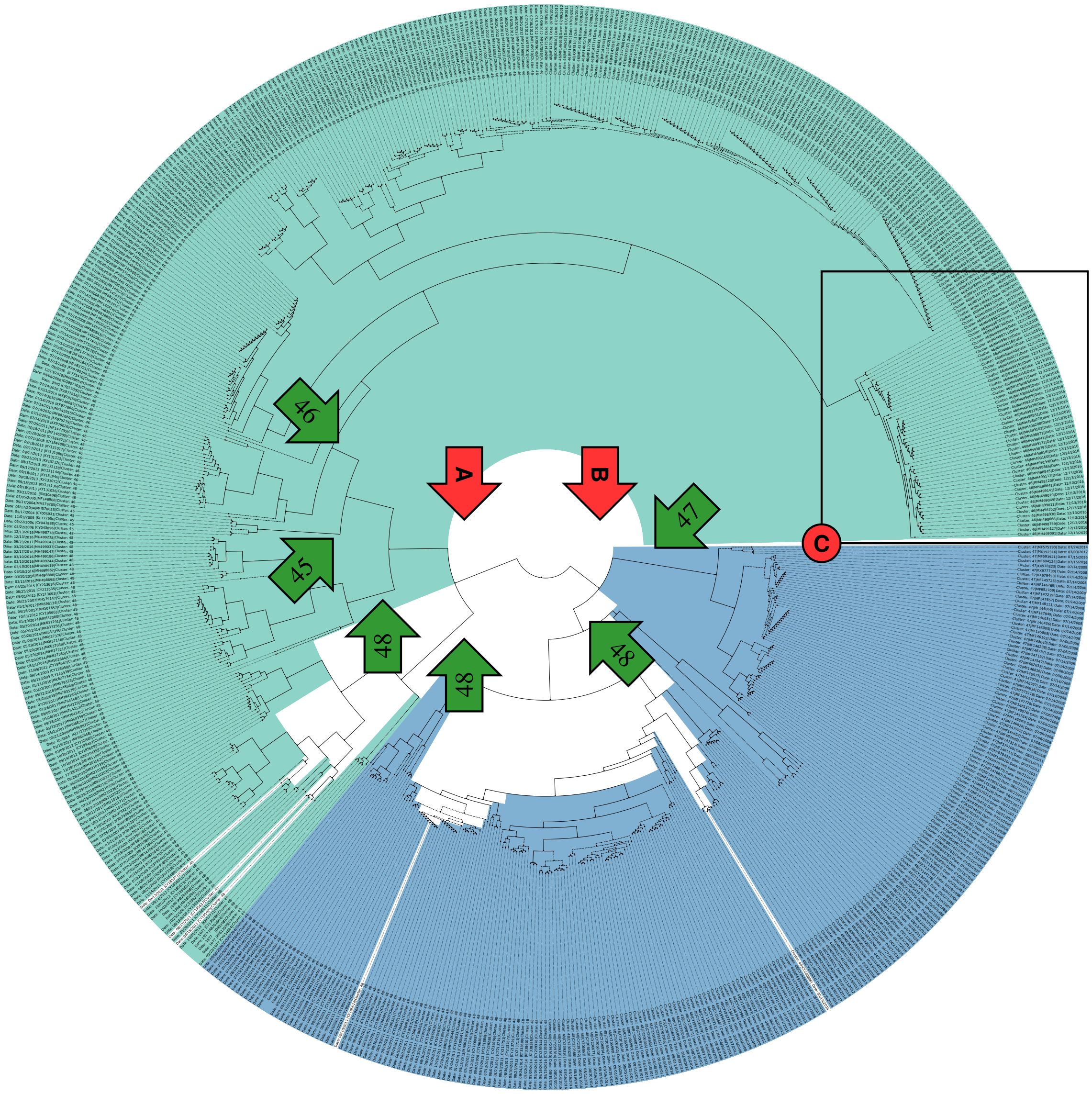


Fig. 3.6 UPGMA tree of H13/H16 with cosine distance. Calculated cosine distance between the k-mer frequency vectors of sequences related to subtype H13 and H16 clusters in Fig. 3.4 were used to build a unweighted pair group method with arithmetic mean (UPGMA) tree. The same labeling of the previous tree was used to clarify the difference between H13 and H16 sequences. Unlabeled sequences were unclassified sequences from the mixed cluster 48 and, thereby, not assigned to a single subtype. The numbers in the green arrows indicate the cluster number of the subtrees sequences in Fig. 3.4. The red arrows will be used in the following to point to the trees division into H13 **A** and H16 **B**. Fig. 3.7 is a enlarged view on the highlighted square at **C**.

By building a UPGMA tree with cosine distance calculation on the non-reduced segment 4 H13 and H16 k-mer frequency vectors with 4^7 components, the unbiased relation of sequences from these subtypes were analyzed (Fig. 2.8 workflow **7**). These calculations required high computation power and were only possible due to the small amount of segment 4 H13 and H16 sequences. Since no component reduction was performed in this case the fundamental use of k-mer frequencies could be validated or rejected. The tree was labeled in a similar way to the previous ones. Therefore, aiding the visualization, unclassified sequences were declared as a given subtype based on the clusters of Fig. 3.4 again. The small amount of not labeled sequences in the UPGMA tree were unclassified sequences, that also could not be assigned to a given subtype in the previous sections (not assigned sequences from mixed cluster 48). The labeling based on the clusters in Fig. 3.4 was used to better visualize the separation and also reversely evaluate the labeling too. Outstanding labeling mistakes breaking the uniform separation would, thus, reject the assignment of not classified sequences, that was performed. All sequences in the UPGMA tree were also annotated by their cluster number in Fig. 3.4 to enable better comparison. The green arrows indicate subtrees in the UPGMA tree containing the sequences assigned to a given cluster in the previous section. In Fig. 3.6, sequences from cluster 46, 45 and 47 are contained in well separated subtrees, while the sequences of cluster 48 are spreaded over half the UPGMA tree. This finding is in line with clustering error Fig. 3.3 **B** and Fig. 3.5 indicating the existence of a clear separation of both subtypes, even when a degree of similarity exist.

While both subtypes in Fig. 3.6 are completely separated directly after the trees root, there are also subsequent subdivisions for both subtypes directly after that. This early subdivision possibly point to the existence of more subtle variations, with major difference to each other than the existing subtype classification reveals. In this case it appeared as if at least two subgroups for H13 and at least two or three subgroups for H16 exist. The threshold was difficult to define, as no clustering was performed here. Thereby, this statement was based only on the early subdivision in the UPGMA tree (Fig. 3.6 **A** and **B**). The exact separation from the root is in line with the centroid guidetree based on MSA where both subtypes were completely separated too (Fig. 3.5). Since this is also in line with the subtype classification, the k-mer frequency approach seemed to work as expected in this project. Furthermore, when focusing on a portion of the UPGMA tree with very small distance in Fig. 3.6 **C**, the similar collection date of all these sequences (12/13/2016) stood out. The only sequences not from this collection date but, nevertheless, included in this subtree are MH499057, MH498978, MH346972, MH499085, and MH499160. Two of the first three mentioned sequences are from the same collection date but some days prior to the rest, while the third was collected some days after the 12/13/2016. These three sequences are the last linked sequences in the subtree with the highest distance in comparison to the rest and their collection date is nearly the same as for the rest. The other two sequences with different date MH499085 and MH499160 are in the middle of the subtree but also collected just one day after the rest. These findings pointed in the direction, that even small differences were noticed by the k-mer approach. The collection date was used as comparison here because many of the sequences in the subtree have very different strain

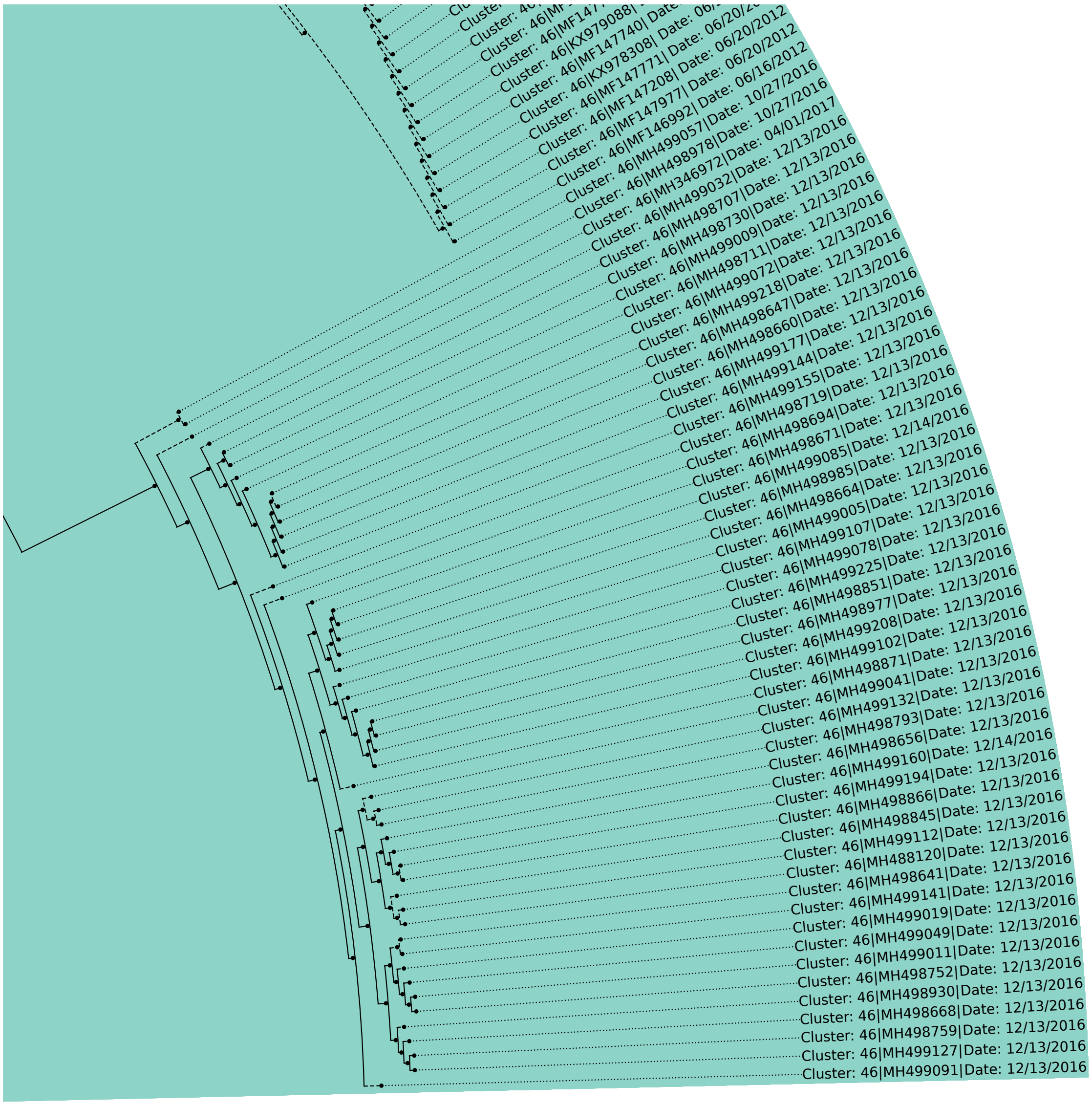

Cluster: 46|MF147
Cluster: 46|KX979088|
Cluster: 46|MF147740| Date: 06/
Cluster: 46|KX978308| Date: 06/20/20
Cluster: 46|MF147771| Date: 06/20/2012
Cluster: 46|MF147208| Date: 06/20/2012
Cluster: 46|MF147977| Date: 06/16/2012
Cluster: 46|MF146992| Date: 10/27/2016
Cluster: 46|MH499057|Date: 10/27/2016
Cluster: 46|MH498978|Date: 04/01/2017
Cluster: 46|MH346972|Date: 12/13/2016
Cluster: 46|MH499032|Date: 12/13/2016
Cluster: 46|MH498707|Date: 12/13/2016
Cluster: 46|MH498730|Date: 12/13/2016
Cluster: 46|MH499009|Date: 12/13/2016
Cluster: 46|MH498711|Date: 12/13/2016
Cluster: 46|MH499072|Date: 12/13/2016
Cluster: 46|MH499218|Date: 12/13/2016
Cluster: 46|MH498647|Date: 12/13/2016
Cluster: 46|MH498660|Date: 12/13/2016
Cluster: 46|MH499177|Date: 12/13/2016
Cluster: 46|MH499144|Date: 12/13/2016
Cluster: 46|MH499155|Date: 12/13/2016
Cluster: 46|MH498719|Date: 12/13/2016
Cluster: 46|MH498694|Date: 12/13/2016
Cluster: 46|MH498671|Date: 12/14/2016
Cluster: 46|MH499085|Date: 12/13/2016
Cluster: 46|MH498985|Date: 12/13/2016
Cluster: 46|MH498664|Date: 12/13/2016
Cluster: 46|MH499005|Date: 12/13/2016
Cluster: 46|MH499107|Date: 12/13/2016
Cluster: 46|MH499078|Date: 12/13/2016
Cluster: 46|MH499225|Date: 12/13/2016
Cluster: 46|MH498851|Date: 12/13/2016
Cluster: 46|MH498977|Date: 12/13/2016
Cluster: 46|MH499208|Date: 12/13/2016
Cluster: 46|MH499102|Date: 12/13/2016
Cluster: 46|MH498871|Date: 12/13/2016
Cluster: 46|MH499041|Date: 12/13/2016
Cluster: 46|MH499132|Date: 12/13/2016
Cluster: 46|MH498793|Date: 12/13/2016
Cluster: 46|MH498656|Date: 12/13/2016
Cluster: 46|MH499160|Date: 12/14/2016
Cluster: 46|MH499194|Date: 12/13/2016
Cluster: 46|MH498866|Date: 12/13/2016
Cluster: 46|MH498845|Date: 12/13/2016
Cluster: 46|MH499112|Date: 12/13/2016
Cluster: 46|MH488120|Date: 12/13/2016
Cluster: 46|MH498641|Date: 12/13/2016
Cluster: 46|MH499141|Date: 12/13/2016
Cluster: 46|MH499019|Date: 12/13/2016
Cluster: 46|MH499049|Date: 12/13/2016
Cluster: 46|MH499011|Date: 12/13/2016
Cluster: 46|MH498752|Date: 12/13/2016
Cluster: 46|MH498930|Date: 12/13/2016
Cluster: 46|MH498668|Date: 12/13/2016
Cluster: 46|MH498759|Date: 12/13/2016
Cluster: 46|MH499127|Date: 12/13/2016
Cluster: 46|MH499091|Date: 12/13/2016

Fig. 3.7 Relation of collection date and k-mer vector distance. Enlarged view of the by **C** highlighted square in Fig. 3.6. The sequences were labeled according to their collection date to indicate the correlation of the close k-mer frequency vector distance in the tree with their sequences similarity. Labeling with the strain name could misleadingly point in a false direction, since the strain names indicate major difference while the sequences having a very high degree of similarity. Thereby the close distance of the vectors of sequences collected on the same date with high sequence similarity point to the precise representation by the k-mer frequency vectors.

names but are almost or completely similar and, thereby, usage of the strain name instead could cause a misinterpretation. MH499085 of strain A/environment/Chile/C20369/2016 and MH498671 of strain A/white_backed_stilt/Chile/C20090/2016 differ by their strain names, as the virus was collected apparently completely different but the sequenced genomes are in fact 100% identical. Around the tree in nearly every case similar collection dates have a small distance based on the k-mer frequency vectors, supporting the statement of the usability of k-mer frequencies for IAV clustering.

3.4 A ground truth for clustering

Since the use of k-mer frequencies proved to be valid for IAV clustering, further investigation on the source of the persisting errors Fig. 3.3 **B** and **D** was performed. For investigation standard HDBSCAN clustering was used without hybrid setting and ε exploration on the same small subset of H13 and H16 sequences used in the previous section. Standard HDBCSCAN was used for simplification and minimization of error sources. Also, the subset used with 662 sequences was smaller than e. g. , the one for full segment 4 clustering with 56617 sequences, making the use of hybrid HDBSCAN unnecessary.

Two different clusterings on this small subset, without the necessity of any dimension reduction were performed and compared to find a ground truth. Subsequently the results were compared to a clustering on the same subset with a simple version of the PK method and, therefore, involvement of dimension reduction. The first input was the non-reduced set of k-mer frequency vectors used in Sec. 3.3 as precalculated cosine distance matrix (Fig. 2.8 workflow **8**). HDBSCAN can use precalculated distances as input instead of vectors. Therefore, no distance calculation is performed by HDBSCAN. Precalculated distances on n vectors create matrices of size $n \times n$, therefore, precalculation is very RAM intensive and not usable on a high number of sequences. Still, since this approach involved no dimension reduction and less calculation by the clustering tool, thereby less error sources, the resulting clustering could be used as ground truth. The result of the clustering was visualized as clustering tree (Fig. 3.8).

In a similar manner to the precalculated UPGMA tree in Fig. 3.6, the subtypes in Fig. 3.8 are completely separated and split on both sides in two subgroups. This pointed to the fact, that the HDBSCAN clustering of the precalculated cosine distances of the k-mer frequencies are as usable as the k-mer frequencies itself to draw a clear line to separate the subtypes. This finding is in line with the second clustering tree based on similar clustering on the same sequences with evolutionary distances of a MSA instead (Fig. 2.9 workflow **6**). There, the same separation is even more obvious, as the subtypes subtrees are farther away from the separation at the trees root in Fig. 3.9. On the side of the H13 sequences, a subdivision is also clearly noticeable. Subgroups in the H16 sequences are, on the other hand, not that clear separated. The different distances between the subtypes and the subgroups in Fig. 3.8 and Fig. 3.9 were most likely caused by evolutionary aspects integrated in the calculation of distances by the MSA. By the k-mer frequencies, the pure constellation of nucleotides was used, evolutionary aspects were neglected. However, clustering with the precalculated approach as well as with the MSAs evolutionary distances used the full information available from the sequences themselves. No reduction with PCA or UMAP

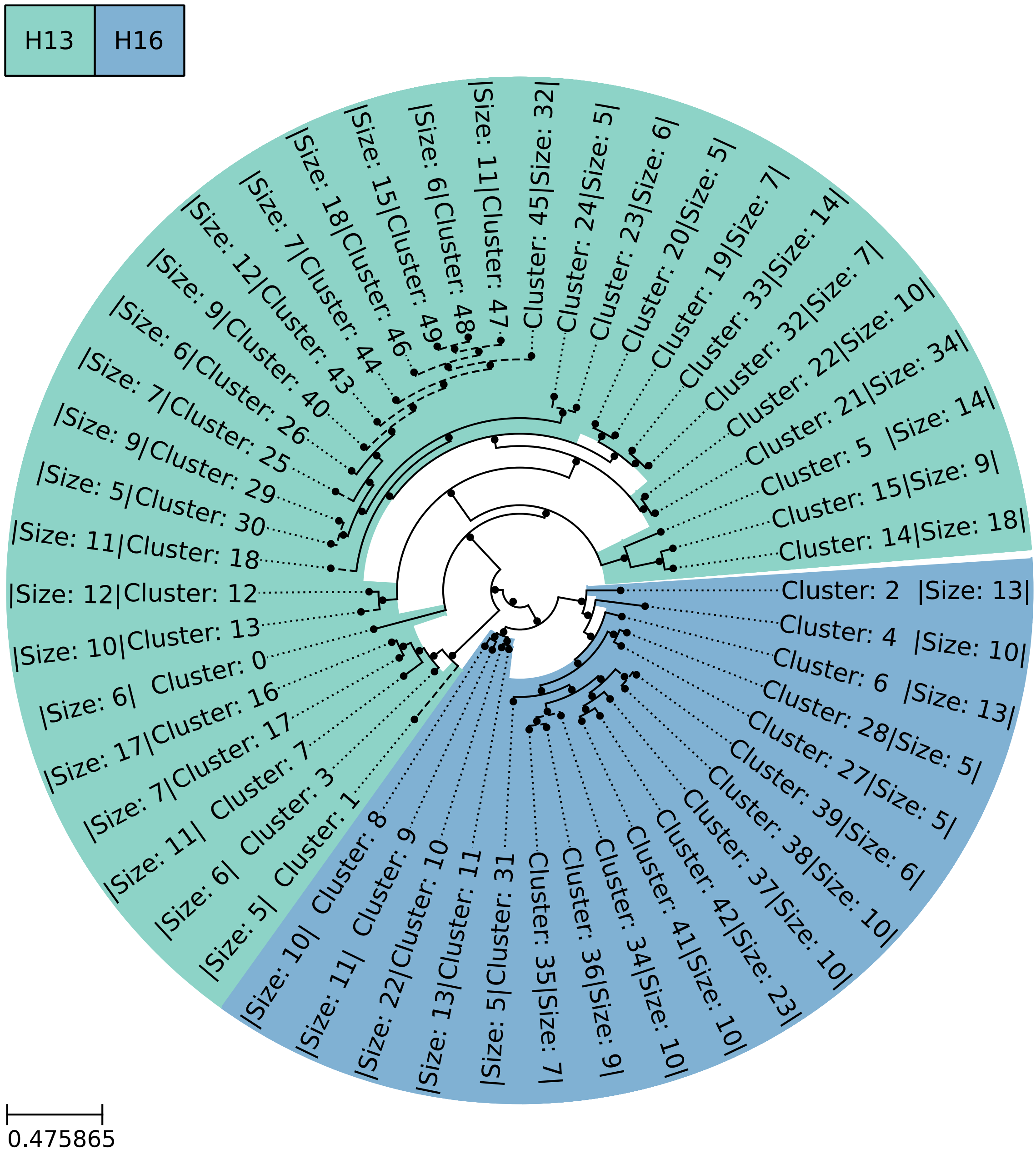


Fig. 3.8 Simple clustering tree of H13/H16 with cosine distance. Clustering tree, based on the clustering by standard HDBSCAN without ε exploration and hybrid clustering. The matrix used as input contained precalculated cosine distances. The distances were calculated from the k-mer frequency vectors related to the sequences, present in the H13 and H16 clusters in Fig. 3.4 without reduction with PCA or UMAP. Therefore, HDBSCAN was used with precalculation input instead of a distance metric.

was performed and both clustering trees indicate full subtype separation. Therefore, these clustering trees (Fig. 3.8 and Fig. 3.9) were the only ground truth for H13/H16 clustering with HDBSCAN available. As already mentioned precalculated clustering with HDBSCAN, is highly computationally expensive, as the matrices of size $n \times n$ have to be calculated

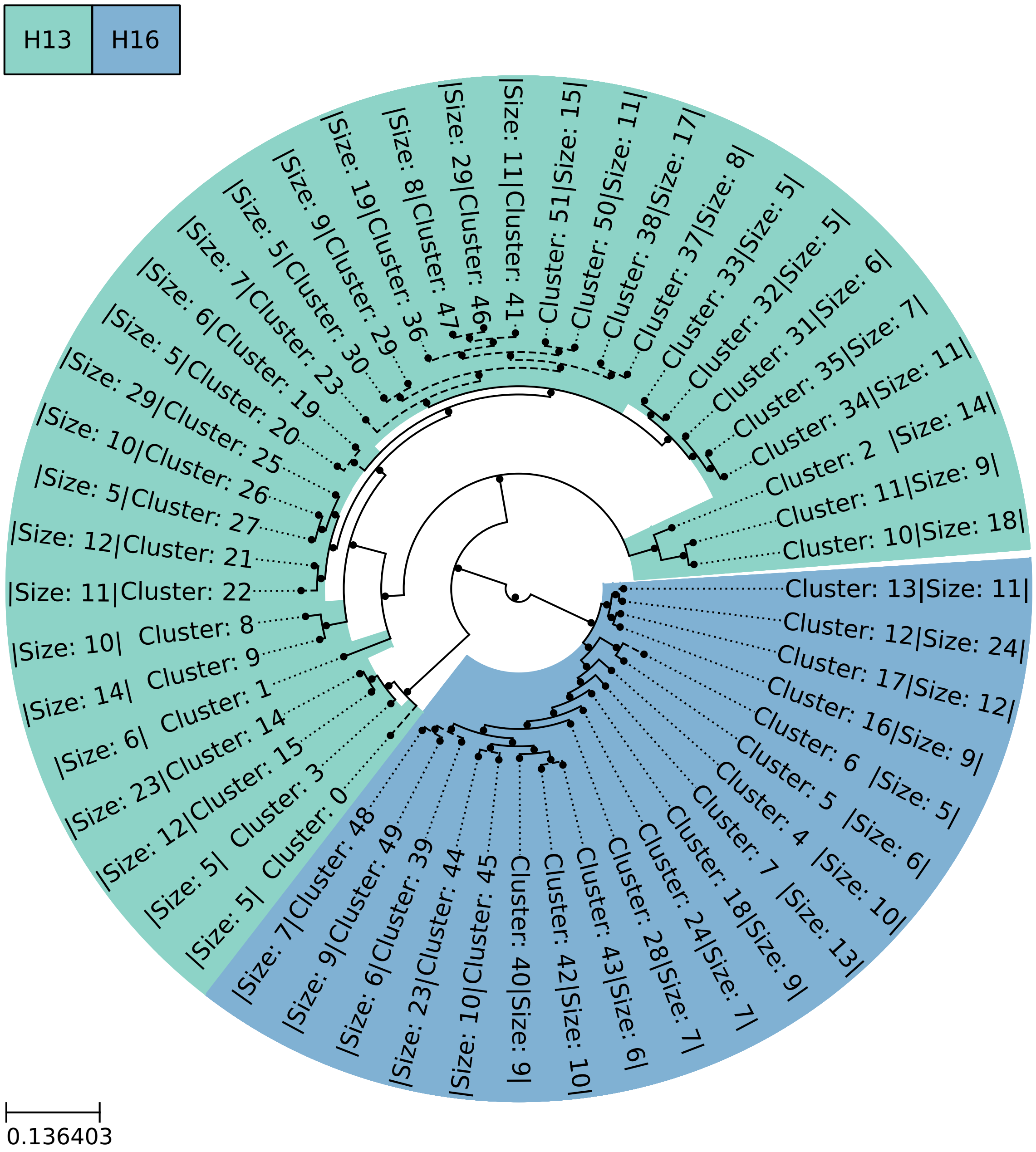


Fig. 3.9 Simple clustering tree of H13/H16 with evolutionary distance. Clustering tree, based on the clustering by standard HDBSCAN without ε exploration and hybrid clustering. The matrix used as input contained precalculated MSA based evolutionary distances. The sequences, present in the H13 and H16 clusters in Fig. 3.4 were used for the MSA. Therefore, HDBSCAN was used with precalculation input instead of a distance metric.

and saved to be used in HDBSCAN. The calculation is, therefore, not possible without the availability of major RAM space. When using HDBSCAN with the k-mer vectors posterior to reduction with PCA to 30 dimensions, only a matrix of size $n \times 30$ has to be saved without

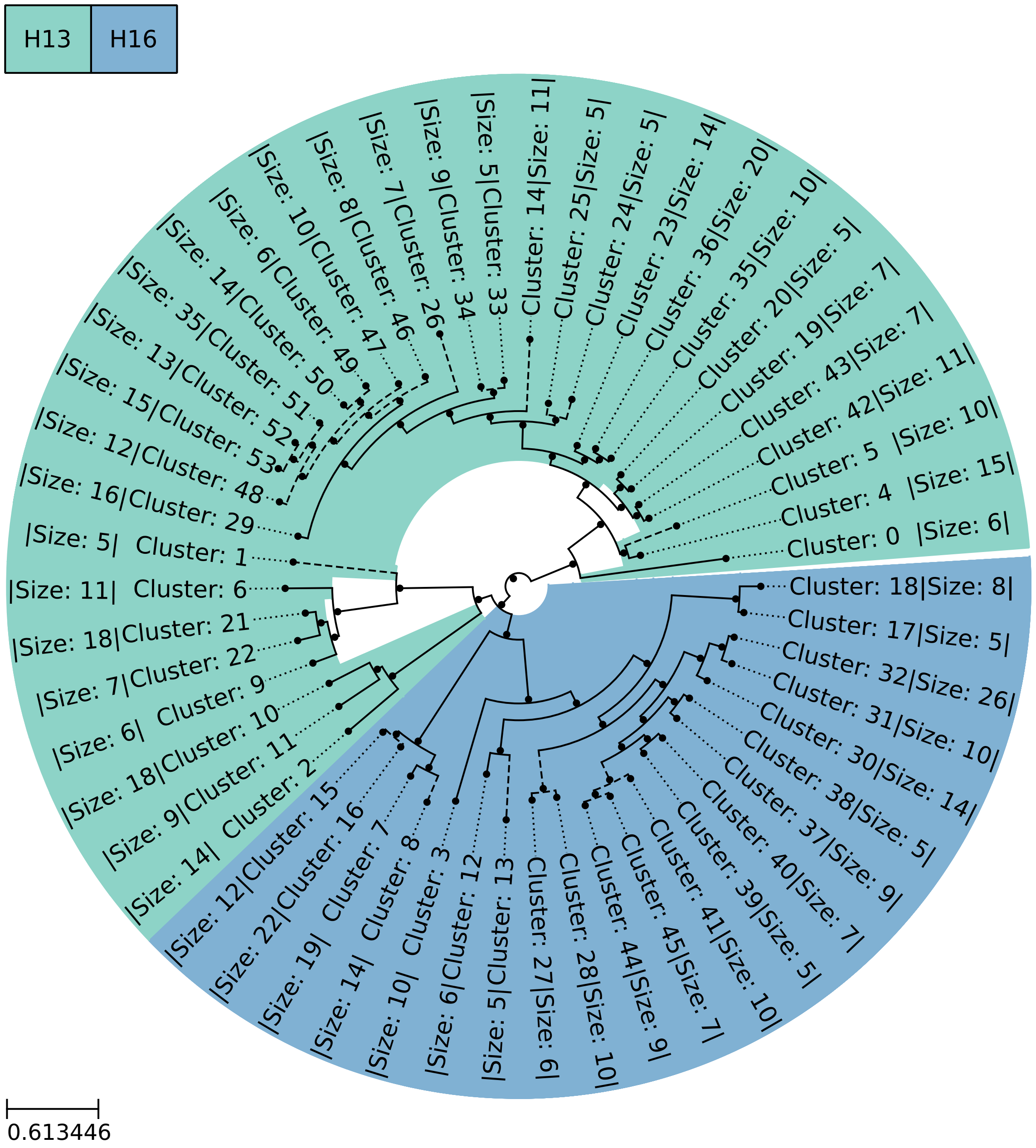


Fig. 3.10 Simple clustering tree of H13/H16 with PCA. Clustering tree, based on the clustering by standard HDBSCAN without ε exploration and hybrid clustering. The used vectors were related to the sequences, present in the H13 and H16 clusters in Fig. 3.4 and reduced by PCA to 30 dimensions.

the necessity of any distance precalculation. This is a **major** reduction of computational power necessary.

A third clustering was performed using PCA reduced vectors of the same H13/H16 sequences (Fig. 2.10). To validate the accuracy of the dimension reduction by PCA, used in this project, the clustering tree in Fig. 3.10 should have represented the ground truth of the previous trees as best as possible. As described in Sec. 3.1, the method using PCA and the Kneedle Algorithm was declared as best method for IAV clustering (PK). In this comparison by standard HDBSCAN clustering without ε exploration, the sole reduction with PCA, representing simplified PK method was, therefore, used (Fig. 3.10). Unfortunately, some major differences between the clustering trees using PCA and the two trees using precalculated cosine distance and MSA evolutionary distance stood out. In Fig. 3.10, the whole tree of H16 is first joined to a subtree of H13, before joining to the other H13 subtree. Therefore, no clear separation is present. The same clustering behavior, without a clear separation was observed in the complete clustering tree in the previous section (Fig. 3.4 **B**). The euclidean distance calculation included in the mutual reachability distance of HDBSCAN is related to the cosine distance as proven in Chap. 2. Furthermore, the Kneedle Algorithm was not used as no ε exploration was performed. Thus, excluding the distance calculation and the Kneedle Algorithm from the error sources. Thereby, the PCA dimension reduction step seemed to be the origin of the clustering error in Fig. 3.3 **B** and **D**. The behavior of the dimension reduction will be fully examined in the following section. Similar clustering with standard HDBSCAN was also performed with the same subset of H13 and H16 sequences reduced with UMAP and PCA, with results inferior to the sole use of PCA, thus, proving again the unsuitability of UMAP for IAV clustering (Fig. A.5).

3.5 Differences in dimension reduction

To investigate the dimension reduction behavior prior to the clustering and, thereby, find explanations for the mentioned errors, the small H13 and H16 subset of segment 4 k-mer frequencies, was reduced by PCA and UMAP to two components for visualization. Comparison to UMAP was done although the method was already declared as not appropriate, to validate this statement again and see the impact of different neighbor values mentioned in Chap. 2.

The target of the dimension reduction prior to the HDBSCAN clustering, was to find a representation of the data with lower complexity, that is suitable to be used for the clustering, while preserve as much information as possible. As explained in Sec. 3.3 and Sec. 3.4, the optimal representation of the vectors should make a clear difference between H13 and H16. This finding will also be used as the ground truth in the following. Since the vectors were visualized in two dimensions, the term point instead of vector will be used.

The visualization of the reduction by PCA is denoted as neighbors value -1 (Fig. 3.11). It shows five different accumulations of points. Labeling of these points is based on the

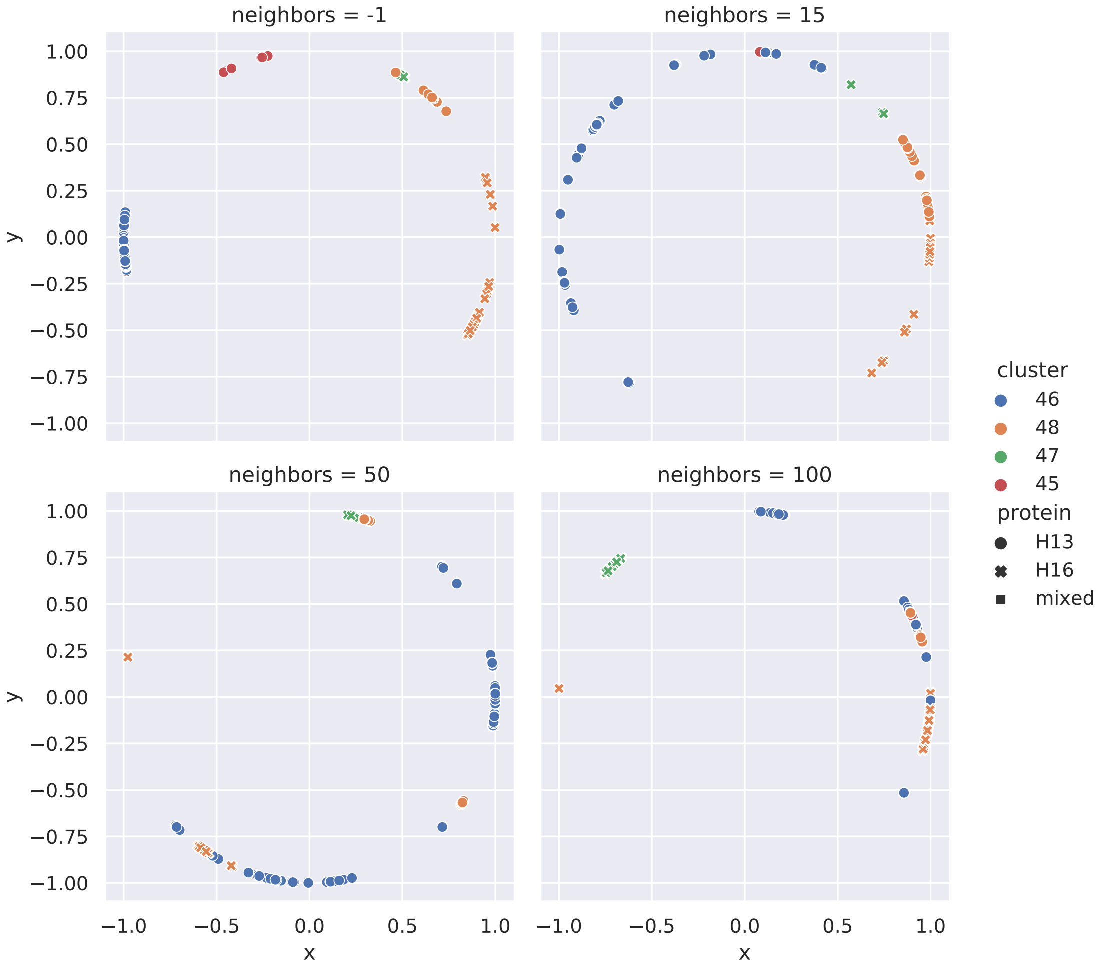


Fig. 3.11 Comparison of H13/H16 component reductions. The subset of sequences from the H13 and H16 clusters in Fig. 3.4 were reduced down to two dimensions enabling simple visualization. Cluster labeling was performed according to Fig. 3.4. Sole use of `PCA` (top left picture) as well as the combination with `UMAP` (other three pictures) was performed as described in Sec. 3.5 with the reduction to two components for visualization. For the combination of `PCA` and `UMAP` different values for the neighbors setting were used, the `UMAP` standard value 15, a average value 50 and the standard value of this project 100. The subtypes of the sequences are labeled by different types of points.

original clustering example in Fig. 3.3. This is becoming apparent when focusing on the cluster 48 points containing H13 and H16 sequences. That way a fundamental distribution on the points of H13 and H16 could be reviewed as well.

The reduction with `PCA` on the subset resulted in easy separable accumulations of the cluster 46 and 48 points in Fig. 3.11 (neighbors = -1). The distribution of these points is basically in line with the result shown in Fig. 3.3, as their accumulations are well separated, building the two clusters with the same sequences in both figures. The major difference, however, is the distance between the accumulations of cluster 48 points to each other as well as to the ones of cluster 47. This would probably result in a imaginary clustering of unchanged cluster 46 and 45 and two or three clusters consisting of the cluster 48 points of which one also contains the points of cluster 47. It seems as if the distance of the cluster 47 points and the H13 cluster 48 points is largely affected by the reduction. The difference between cluster 46 and 45 in Fig. 3.11 (neighbors = -1) is on the other hand preserved and would result in clustering similar to Fig. 3.3. In Fig. 3.3 cluster 47 and 48 are also relatively close related as they would be linked on the next higher tree-node.

In Fig. 3.11 (neighbors = -1) it appears as if the points of cluster 47 and 48 are possibly quite similar, which is not the case as the Fig. 3.6 subtrees clearly show the wanted separation of H13 and H16 in cluster 48, as well as the wanted distance to 47. Keeping the lower complexity in mind, the consequence of lowering the dimension by PCA to two dimension seemed to preserve most of the information related to the difference of cluster 45 and 46. The difference of the subtype separation inside 48 as well as the overall difference to 47 on the other hand, seemed to be lost completely and caused the unwanted effects. Since the ground truth separation of Fig. 3.6 seems to be partially present in Fig. 3.4, by at least separating 47 completely from 48, the higher number of dimensions might be in direct connection to the correct separation of some part of H13 and H16. Therefore, even when raising the computationally effort, the number of components should be increased to the maximum of 50, that still preserves all functions of HDBSCAN for spanning tree building.

Comparing these results to the use of UMAP with different settings of the neighbors value, the impact of this parameter becomes clear. The higher the value, the more crowded the points. This also explains the crowded behavior in Fig. 3.2b. Since a neighbors value of 100 was used as standard in this project, the values were overall crowded in groups of at least 100 points. The random subset for Fig. 3.2b was reduced by the same setting with UMAP, despite the small sample size of 100 used there. The small random sample in addition to a high neighbors value resulted in a low number of overall distribution to clarify the behavior. Aside from the example in Fig. 3.2b, the usage of a high neighbors value through the project was well reasoned and based on the huge size of the dataset used as described in Sec. 3.5. The same value of 100, as well as, 15 and 50 was used on the subset of H13 and H16 segment 4 sequences to visualize the difference in Fig. 3.11.

None of the settings resulted in a separation as good as with the sole use of PCA. With the UMAP standard neighbors value of 15, all the points are placed next to each other and there is no reasonable cluster building possible Fig. 3.11 (neighbors = 15). Furthermore, H13 points would be merged with H16 points before merging with others from H13, thereby breaking the subtype division similar to the PCA use. Aside from the fact that the other points, when using PCA, are well separated. Setting the neighbors value to 50 results in a spreading of the cluster 46 points and mixing with little islands of cluster 48 points Fig. 3.11 (neighbors = 50). With a neighbors value of 100, a separation into imaginary clusters is possible, when ignoring the cluster labeling and only taking the subtypes labeling into consideration. This is, therefore, the only setting with use of `UMAP` that would provide a more of less reasonable separation of the subtypes in imaginary clusters. However, clusters of different subtypes are closer than to similar subtypes, resulting also in no real subtype separation, even when ignoring the cluster 47 points that might be very sensible to the magnitude of preserved information.

In conclusion, the use of PCA generated better results compared the ones with additional use of `UMAP`. Still, there were challenges to overcome as could be seen with the position of the cluster 47 points. Increasing the information preserved by the `PCA` would possibly give clearer results. This project aimed to find high-quality representations of IAV genomes for the purpose or clustering in a extend that was never reached before. Therefore, the usability of hybrid `HDBSCAN` with parameters as good as possible was of higher importance than the use of `UMAP` at all costs. In the results of the project, `PCA` performed better than `UMAP` but only with all the tested parameters. Thus, it might be possible to find parameters for `UMAP` not explored in this project to represent the genomes even better in a equal low-dimension in the future. Also, change of `UMAP` in favor of `t-SNE` could be tested in terms of vector representation quality.

3.6 A new classification

Reannotation of the most likely false annotated sequence in Fig. 3.4 **C**, as well as increasing of the components in `PCA`, successfully raised the accuracy of the workflow. Thus, the clustering was performed according to the PK method, that proved to give the most stable results, with 50 components reduction instead of 30. Clustering errors found in the previous section were resolved successfully as H13 and H16 are now completely divided in Fig. 3.12, with the mentioned small but still present difference between these subtypes. Also, all clusters of H3 are now present in direct connection to each other and no cluster not homogeneous for one subtype existed anymore. Comparison of the associated clustering information graphics in Fig. A.6 to the previous ones in Fig. 3.3 also indicated a small improvement in stability of the Kneedle Algorithm ε exploration. Little changes in the distribution of the cluster sizes were also noticeable, as a cluster came to existence, containing 20000 of the H3 sequences and pointing in direction of a merge of two big clusters by the availability of a higher amount of information.

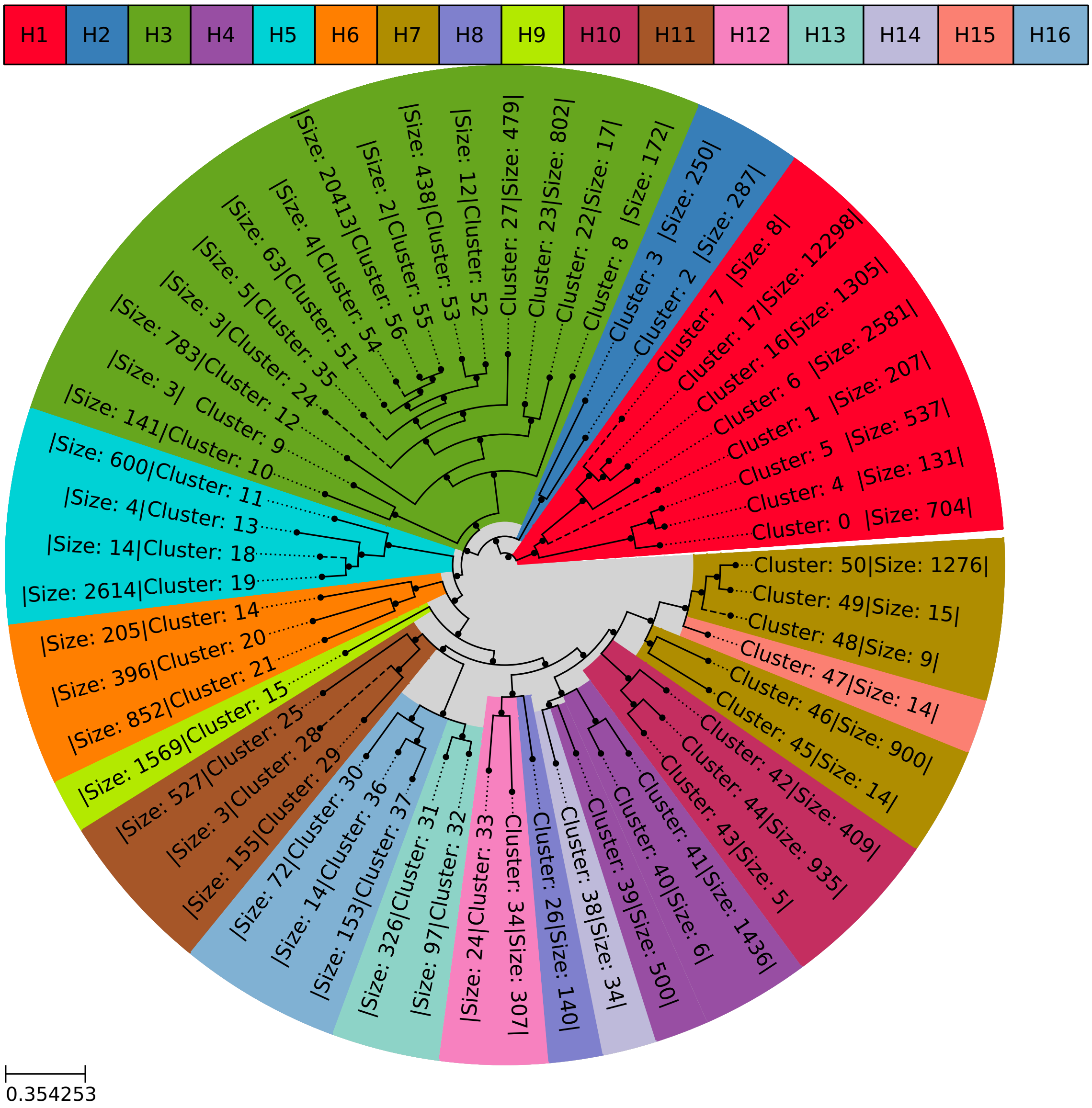


Fig. 3.12 Clustering tree of segment 4. The clustering tree of segment 4 clustering, using the combination of PCA and the Kneedle Algorithm (PK) with a reduction to 50 instead of 30 dimensions (Fig. 3.3). The labeling of the clusters in the tree is based on the subtype of the contained sequences. Unclassified sequences of a cluster are reclassified as a given subtype if sequences of only this subtype are present in the cluster in addition to the unclassified ones. Unlabeled clusters contain sequences from at least two subtypes and zero or more unclassified sequences. Dotted lines in the tree indicate the same host.

Resulting from this improved clustering of segment 4 a new classification involving 57 clusters instead of 18 subtypes is proposed (Fig. 3.12). The big differences in the cluster sizes related to H1 and H3 indicated a high amount of more similar sequences. This could be caused by the higher abundance of more present-day sequences and continued evolution, thus, higher differences in comparison to less sequenced strains of the past decades. Less present mutations lowering the infectiousness and therefore, not established in a higher amount of strains or sequencing errors that produced undesired niches are also possible. These possible error sources have to be examined in the next steps following this project.

Validation of the clusters was performed by pairwise comparison of random 10 sequence samples of the clusters in Fig. 3.12. Thereby, the degree of similarity inside these clusters and in comparison to other clusters is described in Fig. 3.13. All clusters were compared to all other clusters and the clusters themselves with a second sample. To avoid creating a bias, the result for accidental comparisons of sequences with the same accession were ignored and not considered in calculations of respective mean values. Samples of 10 sequences were used due to the high amount of computational power that was necessary for this calculation. For segment 4 calculation involved 10^2 pairwise alignments, for every of the 57 clusters to each other. Making $57^2 \cdot 10^2$ calculations.

All clusters in Fig. 3.13 have the highest similarities with themselves and in addition high similarities with other clusters of the same subtype. For instance the sample of cluster 0, shared some degree of sequence identity with other samples of clusters also stemming from the same subtype H1 (cluster 1 to 7, 16 and 17). However, the highest similarity of around 90% is only shared with the other sequence sample of the cluster 0 itself, which makes these clusters highly self-contained.

Exceptions are the subtype H7 and H15 clusters 45 to 50, as well as subtype H4 and H14 clusters 35 to 38, that share a high degree of similarity despite of the subtype difference in Fig. 3.13. In the clustering tree no clear separation is visible involving these subtypes, as the clusters of H7 merge with H15 in a higher tree nodes, although other clusters of H7 are still available (Fig. 3.12). Similar behavior can be observed for H4 with H14. These same merges were also present in the previous clustering with reduction to 30 dimensions (Fig. 3.4). Due to the high similarities inside these two groups of clusters, a relation neglected by the current subtype classification could be possible. The almost uniform H7 subtree of clusters 45, 46, 48, 49, and 50 is, therefore, divided by the one and only cluster of subtype H15 cluster 47. When comparing the sequence similarities in Fig. 3.13, the highest similarity persist inside the clusters themselves, but the cross similarities of cluster 45 and 46 to 47, 48, 49, and 50 are around 65% without big difference between subtype H7 and H15. Further pointing to more subtle, but present differences between and inside of the current subtypes. Aside from that, the amount of information that was preserved by the PCA could still be to small to fully separate the subtypes H4 and H14 and the subtypes H7 and H15, that possibly involve even more subtle differences than the previously discussed separation of H13 and H16. The inclusion of evolution into the vectors, as described in

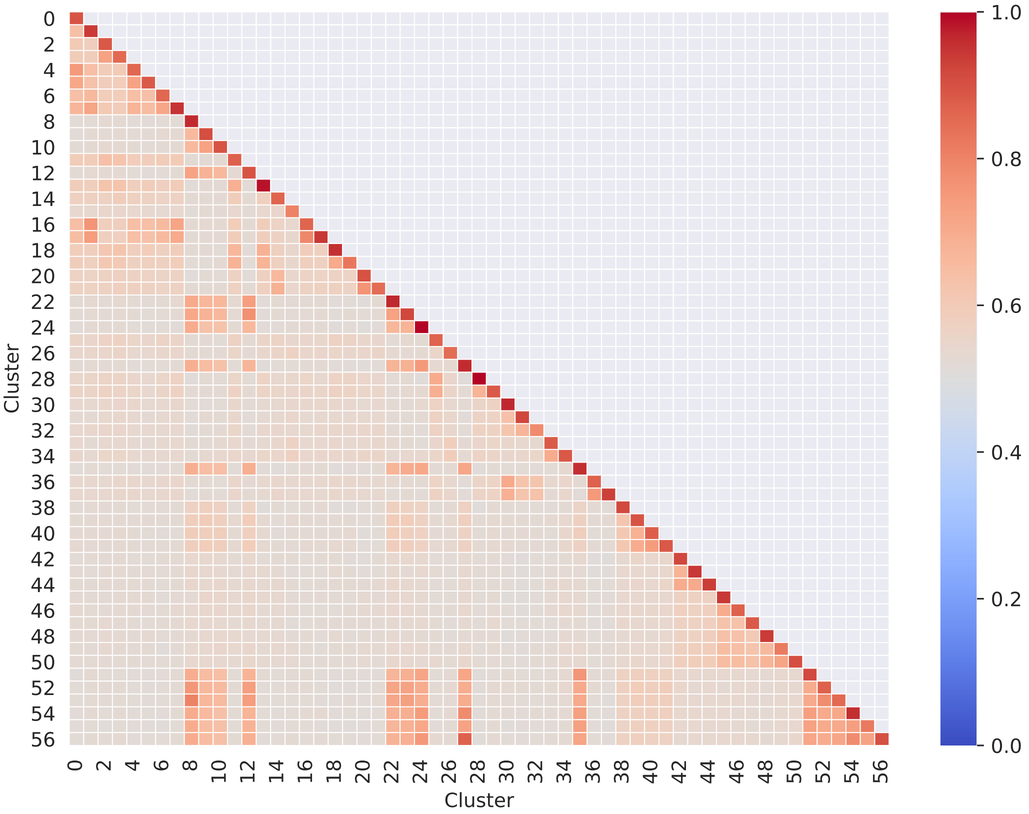


Fig. 3.13 Similarity matrix of segment 4 clusters.. Random samples of up to 10 sequences of every cluster in Fig. 3.12 were compared by pairwise alignments with the samples of all other clusters. Less than 10 sequences were only used in clusters containing less than 10 sequences. The percentage of similarity was calculated for very alignment making a matrix of up to 10 × 10 holding the similarity values of the samples comparisons. The mean of the matrix was then calculated and written as the given cluster interactions mean similarity. This was repeated for every possible cluster interaction resulting in the presented figure. Interactions of the same clusters were reduced to only different sequences, results of alignments with sequences having the same accession were removed, thereby, preventing bias creation. The mean of these up to 100 comparisons for every cluster interaction is colored according to their similarity from 1.0 or 100% similarity in red to 0.0 or 0% similarity in blue.

the following can be the necessary step to increase the amount of information, without further raising the components by PCA, to successfully separate these subtypes.

While the clusters of segment 4 in Fig. 3.12 seem to be highly self-contained and gave a good representation on possible subdivisions inside the subtypes, the trees based on evolutionary distances increase the subtypes distance even more (Fig. 3.5 and Fig. 3.9). Present day research propose a phylogenic tree of IAV that is split in four subtrees [84]. The subdivisions are mostly present in the clustering tree in Fig. 3.12. Still, there are some difference primarily the higher-ranking structure of the subtypes. In Wei et al. [84] subtype H3 is contained in a subtree alongside H4 and H14 and H9 in a subtree with H8 and H12. This relations are not present in Fig. 3.12. While other subtrees contain subtypes similar to the proposed phylogenic tree in Wei et al. [84], the order of the subtrees in Fig. 3.12 also do not match the one from the phylogenic tree. However, all the clusters in Fig. 3.12 are highly self-contained and homogeneous for one subtype, only the position in the tree might be inaccurate and, therefore, should be subject of improvements.

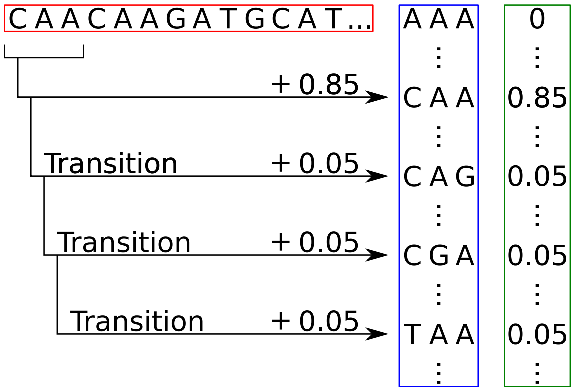


Fig. 3.14 7-mer vector calculation involving transitions. By changing Fig. 2.4 to include e. g. , all transitions possible with one mutation, evolutionary distance can be included in the vectors themselves. Thereby, vectors involving sequences different by a small amount of mutations with higher evolutionary significance move closer to each other. All weightings used here are only exemplary and useful values have to be selected in the future based on novel publications involving mutation probabilities.

As already mentioned the proposed method for IAV clustering did not acknowledge evolutionary distances. Transitions and transversion for example were handled as nucleotide differences without a higher or lower chance of change. When including mutation chances in the k-mer frequency clustering, the relation of whole subtypes subtrees might improve and give a result more similar to the one proposed in Wei et al. [84]. Because HDBSCAN only uses existing distance metrics, when not using the precalculated option, which should be avoided at any case, the k-mer frequency vectors themselves have to be change in some way. Therefore, Fig. 3.14 illustrate a possible option for inclusion of evolution inside k-mer frequency vectors, by considering all mutations of a given k-mer with low representation value. Vectors containing k-mers only apart with single mutations are, thus, closer to each other in the high-dimensional vector space. The magnitude of these values have to be considered wisely and should be subject of future research optimizing the proposed clustering even more. For choice of appropriate values representing the transversions or transitions, consideration of PAM or BLOSUM substitution matrices could be beneficial [48]. Since k-mers of a sequence are generated by shifting a window of size k by one, all aminoacid (AA) constellations are automatically included, independent of the k value itself. Thereby, weighting of the mutation possibility could be also implemented by considering the actual k-mer with a fixed value e. g. 0.85 and the mutations with fractures of 0.15 given by the priority in the BLOSUM or the PAM matrices. Summing up the values would, thus, still result in one but the weightings are distributed based on the score of the AA change. Expanding the minimal example Fig. 3.14 in that way, involving BLOSUM could result in something similar to CAA → 0.85, CAG → $\frac{0.15}{(5+1+0)} \cdot 5$, CGA → $\frac{0.15}{(5+1+0)} \cdot 1$, and TAA → $\frac{0.15}{(5+1+0)} \cdot 0$. The portion of 0.15 is divided by the sum of all considered AAs change values and subsequently multiplied by the value of the given one.

With this workflow blueprint created by adjusting the clustering for segment 4 in the previous sections, all IAV segments could be clustered the same way, since no subtype information was necessary for the clustering. Subtype labeling of the tree was performed as guideline and was the only part involving actual subtype evaluation of the sequences. Thereby, the clustering trees for other segments, except segment 6, are unlabeled. All other clustering trees and similarity matrix graphics, as well as tables containing sequence cluster assignment and the tables containing the values used to create the graphics are presented in the Appendix A. Hereby, new classifications for all segments based on k-mer frequency vectors are proposed, containing 28 clusters for segment 1, 28 for segment 2, 29 for segment 3, the shown 57 for segment 4, 26 for segment 5, 40 for segment 6, 30 for segment 7 and 24 clusters for segment 8 Table A.1.

The clusters of the segments 1 to 3, 5, 7, and 8 shared by far more overall similarity, therefore, less clusters were created (Appendix A). Still, the similarity inside the clusters themselves were higher than the cross similarity and, thereby, solid clustering by the proposed clustering method was possible despite the overall higher similarity. Higher similarity of the segments not encoding the surface protein is most likely reasoned by the lower evolutionary pressure, since the higher pressure on the surface proteins is necessary to ensure continued infectiousness of IAV.

4 Conclusions and Outlook

The existing classification of the *Influenza A Virus* (IAV) is solely based on the surface proteins and gives, therefore, no real insight on the differences of the other segments. While many publications exist, involving evolutionary research on these, as well as, the other segments, that propose subgroups or clusters, most are based on alignments for comparison [52, 71, 85]. Present day alignment methods offer accurate insight of sequence relation. Nevertheless, downsides exist, as the use of multiple sequence alignments (MSAs) as distance measurement for clustering of IAV is bound to the necessity of high nearly unfeasible computational power, when involving all the existing sequences. Even with the hardware available, a threshold for usability at a given number of sequences still exist or a drop of alignment quality is unavoidable. The usage of distance precalculation on the non-reduced k-mer frequency vectors, mentioned in this project, is another method aside from MSA involving also no dimension reduction methods. The exhibited results are, indeed, of mostly equal quality to the MSA, as shown in Sec. 3.4 and drawing a similar clear line between subtypes. Still, similar downsides exist also, making scaling to the enormous number of existing IAV sequences nearly impossible and render both methods mostly usable on smaller subsets.

This project proposed a method, scale-able to a much higher degree for clustering all segments of IAV, that is usable on multi-core computers with around 32Gb of RAM available in less than two hours. The fast execution time of the method makes it usable to even cluster novel sequenced IAV genomes with the existing ones to find associated well-known strains in short time. The dimension reduction with `PCA` proved to preserve the necessary amount of information for robust clustering and with the Kneedle Algorithm a solid threshold was defined. The present day version of the method already produced self contained clusters, with a higher order mostly in line with the current subtypes classification. Furthermore, increasing the amount of informations by reasonable subdivision in smaller subordinated groups. Still, there are options to fathom in the future to possibly produce even better results.

Including measurement possibilities for evolutionary distances by slight changes in the vector creation process as described in Fig. 3.14 could possibly improve the accuracy in comparison to MSAs. In addition to that, PCA as a tool for dimension reduction performed sufficient enough, still, the usage of a methods offering better low dimension representations and higher information preservation of the vectors could be beneficial. Therefore, repeating the comparisons to PCA with t-SNE instead of UMAP might be rewarding. Also all clusters and especially the high difference in the cluster sizes also have to be examined for possible clustering errors or rare mutations with possibly yet unknown purpose.

However, the results point in the direction of a bioinformatical IAV classification with more subdivisions as the known subtypes classification can offer. Renewing the classification in a similar way and, thereby, including more subtle differences might considerably improve future large scale *in silico* secondary structure analyses. With better and more self contained subgroups of IAV, searching for conserved structures would probably make a step ahead and could offer new insights into the IAV.

Bibliography

[1] "A revised system of nomenclature for influenza viruses". In: *Bulletin of the World Health Organization* 45.1 (1971), pp. 119–124. URL: https://pubmed.ncbi.nlm.nih.gov/5316848/.

[2] "A revision of the system of nomenclature for influenza viruses: a WHO memorandum". In: *Bulletin of the World Health Organization* 58.4 (1980), pp. 585–591. URL: https://pubmed.ncbi.nlm.nih.gov/6969132/.

[3] A Ali, RT Avalos, E Ponimaskin, and DP Nayak. "Influenza Virus Assembly: Effect of Influenza Virus Glycoproteins on the Membrane Association of M1 Protein". In: *Journal of Virology* 74.18 (2000), pp. 8709–8719. DOI: 10.1128/JVI.74.18.8709-8719.2000.

[4] Anaconda Software Distribution. *Anaconda*. 2020. URL: https://anaconda.com.

[5] E Area, J Martin-Benito, P Gastaminza, E Torreira, JM Valpuesta, JL Carrascosa, and J Ortin. "3D structure of the influenza virus polymerase complex: Localization of subunit domains". In: *Proceedings of the National Academy of Sciences* 101.1 (2004), pp. 308–313. DOI: 10.1073/pnas.0307127101.

[6] I Assent. "Clustering high dimensional data". In: *WIREs Data Mining and Knowledge Discovery* 2.4 (2012), pp. 340–350. DOI: 10.1002/widm.1062.

[7] IG Barr, J McCauley, N Cox, R Daniels, OG Engelhardt, K Fukuda, G Grohmann, A Hay, A Kelso, A Klimov, T Odagiri, D Smith, C Russell, M Tashiro, R Webby, J Wood, Z Ye, and W Zhang. "Epidemiological, antigenic and genetic characteristics of seasonal influenza A(H1N1), A(H3N2) and B influenza viruses: Basis for the WHO recommendation on the composition of influenza vaccines for use in the 2009–2010 Northern Hemisphere season". In: *Vaccine* 28.5 (2010), pp. 1156–1167. DOI: 10.1016/j.vaccine.2009.11.043.

[8] R Belshaw, A Gardner, A Rambaut, and OG Pybus. "Pacing a small cage: mutation and RNA viruses". In: *Trends in Ecology & Evolution* 23.4 (2008), pp. 188–193. DOI: 10.1016/j.tree.2007.11.010.

[9] SH Bernhart, IL Hofacker, S Will, AR Gruber, and PF Stadler. "RNAalifold: improved consensus structure prediction for RNA alignments". In: *BMC Bioinformatics* 9.1 (2008), p. 474. DOI: 10.1186/1471-2105-9-474.

[10] RJGB Campello, D Moulavi, and J Sander. "Density-Based Clustering Based on Hierarchical Density Estimates". In: *Advances in Knowledge Discovery and Data Mining*. Ed. by J Pei, VS Tseng, L Cao, H Motoda, and G Xu. Red. by D Hutchison, T Kanade, J Kittler, JM Kleinberg, F Mattern, JC Mitchell, M Naor, O Nierstrasz, C Pandu Rangan, B Steffen, M Sudan, D Terzopoulos, D Tygar, MY Vardi, and G Weikum. Vol. 7819. Springer Berlin Heidelberg, 2013, pp. 160–172. ISBN: 978-3-642-37455-5.

[11] RJGB Campello, D Moulavi, A Zimek, and J Sander. "Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection". In: *ACM Transactions on Knowledge Discovery from Data* 10.1 (2015), pp. 1–51. DOI: 10.1145/2733381.

[12] A Cann. "Chapter 2 Particles". In: *Principles of molecular virology*. Sixth edition. Amsterdam: Elsevier/AP, Academic Press is an imprint of Elsevier, 2016. ISBN: 978-0-12-801946-7.

[13] PJ Cock, T Antao, JT Chang, BA Chapman, CJ Cox, A Dalke, I Friedberg, T Hamelryck, F Kauff, B Wilczynski, et al. "Biopython: freely available Python tools for computational molecular biology and bioinformatics". In: *Bioinformatics* 25.11 (2009). Publisher: Oxford University Press, pp. 1422–1423.

[14] B Dadonaite, B Gilbertson, ML Knight, S Trifkovic, S Rockman, A Laederach, LE Brown, E Fodor, and DLV Bauer. "The structure of the influenza A virus genome". In: *Nature Microbiology* 4.11 (2019), pp. 1781–1789. DOI: 10.1038/s41564-019-0513-7.

[15] YM Deng, N Spirason, P Iannello, L Jelley, H Lau, and IG Barr. "A simplified Sanger sequencing method for full genome sequencing of multiple subtypes of human influenza A viruses". In: *Journal of Clinical Virology* 68 (2015), pp. 43–48. DOI: 10.1016/j.jcv.2015.04.019.

[16] S Duffy. "Why are RNA virus mutation rates so damn high?" In: *PLOS Biology* 16.8 (2018), e3000003. DOI: 10.1371/journal.pbio.3000003.

[17] R Edgar. "Usearch". In: (2010). URL: https://www.osti.gov/biblio/1137186.

[18] RC Edgar. "MUSCLE: multiple sequence alignment with high accuracy and high throughput". In: *Nucleic Acids Research* 32.5 (2004), pp. 1792–1797. DOI: 10.1093/nar/gkh340.

[19] AJ Eisfeld, G Neumann, and Y Kawaoka. “At the centre: influenza A virus ribonucleoproteins”. In: *Nature Reviews Microbiology* 13.1 (2015), pp. 28–41. DOI: 10.1038/nrmicro3367.

[20] M Ester, HP Kriegel, J Sander, and X Xiaowei. “A density-based algorithm for discovering clusters in large spatial databases with noise”. In: (1996). URL: https://www.osti.gov/biblio/421283.

[21] DF Feng and RF Doolittle. “Progressive sequence alignment as a prerequisitetto correct phylogenetic trees”. In: *Journal of Molecular Evolution* 25.4 (1987), pp. 351–360. DOI: 10.1007/BF02603120.

[22] J Gea-Banacloche, RT Johnson, A Bagic, JA Butman, PR Murray, and AG Agrawal. “West Nile Virus: Pathogenesis and Therapeutic Options”. In: *Annals of Internal Medicine* 140.7 (2004), p. 545. DOI: 10.7326/0003-4819-140-7-200404060-00015.

[23] M Gerber, C Isel, V Moules, and R Marquet. “Selective packaging of the influenza A genome and consequences for genetic reassortment”. In: *Trends in Microbiology* 22.8 (2014), pp. 446–455. DOI: 10.1016/j.tim.2014.04.001.

[24] J Handl, J Knowles, and DB Kell. “Computational cluster validation in post-genomic data analysis”. In: *Bioinformatics* 21.15 (2005), pp. 3201–3212. DOI: 10.1093/bioinformatics/bti517.

[25] CR Harris, KJ Millman, SJvd Walt, R Gommers, P Virtanen, D Cournapeau, E Wieser, J Taylor, S Berg, NJ Smith, R Kern, M Picus, S Hoyer, MHv Kerkwijk, M Brett, A Haldane, JFd Río, M Wiebe, P Peterson, P Gérard-Marchant, K Sheppard, T Reddy, W Weckesser, H Abbasi, C Gohlke, and TE Oliphant. “Array programming with NumPy”. In: *Nature* 585.7825 (2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2.

[26] PG Higgs. “RNA secondary structure: physical and computational aspects”. In: *Quarterly Reviews of Biophysics* 33.3 (2000), pp. 199–253. DOI: 10.1017/S0033583500003620.

[27] J Huerta-Cepas, F Serra, and P Bork. “ETE 3: Reconstruction, Analysis, and Visualization of Phylogenomic Data”. In: *Molecular Biology and Evolution* 33.6 (2016), pp. 1635–1638. DOI: 10.1093/molbev/msw046.

[28] “Chapter 17 Pandemics: Risks, Impacts, and Mitigation”. In: *Disease Control Priorities, Third Edition (Volume 9): Improving Health and Reducing Poverty*. Ed. by DT Jamison, H Gelband, S Horton, P Jha, R Laxminarayan, CN Mock, and R Nugent. The World Bank, 2017. ISBN: 978-1-4648-0527-1.

[29] KE Jones, NG Patel, MA Levy, A Storeygard, D Balk, JL Gittleman, and P Daszak. "Global trends in emerging infectious diseases". In: *Nature* 451.7181 (2008), pp. 990–993. DOI: 10.1038/nature06536.

[30] E Jourdain, G Gunnarsson, J Wahlgren, N Latorre-Margalef, C Bröjer, S Sahlin, L Svensson, J Waldenström, Å Lundkvist, and B Olsen. "Influenza Virus in a Natural Host, the Mallard: Experimental Infection Data". In: *PLoS ONE* 5.1 (2010). Ed. by J Chave, e8935. DOI: 10.1371/journal.pone.0008935.

[31] I Julkunen, K Melén, M Nyqvist, J Pirhonen, T Sareneva, and S Matikainen. "Inflammatory responses in influenza A virus infection". In: *Vaccine* 19 (2000), S32–S37. DOI: 10.1016/S0264-410X(00)00275-9.

[32] K Katoh. "MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform". In: *Nucleic Acids Research* 30.14 (2002), pp. 3059–3066. DOI: 10.1093/nar/gkf436.

[33] K Katoh and DM Standley. "MAFFT Multiple Sequence Alignment Software Version 7: Improvements in Performance and Usability". In: *Molecular Biology and Evolution* 30.4 (2013), pp. 772–780. DOI: 10.1093/molbev/mst010.

[34] H Kida, R Yanagawa, and Y Matsuoka. "Duck influenza lacking evidence of disease signs and immune response". In: *Infection and Immunity* 30.2 (1980), pp. 547–553. DOI: 10.1128/iai.30.2.547-553.1980.

[35] JS Kieft. "Viral IRES RNA structures and ribosome interactions". In: *Trends in Biochemical Sciences* 33.6 (2008), pp. 274–283. DOI: 10.1016/j.tibs.2008.04.007.

[36] T Korenius, J Laurikkala, and M Juhola. "On principal component analysis, cosine and Euclidean measures in information retrieval". In: *Information Sciences* 177.22 (2007), pp. 4893–4905. DOI: 10.1016/j.ins.2007.05.027.

[37] W Li and A Godzik. "Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences". In: *Bioinformatics* 22.13 (2006), pp. 1658–1659. DOI: 10.1093/bioinformatics/btl158.

[38] R Lorenz, SH Bernhart, C Höner zu Siederdissen, H Tafer, C Flamm, PF Stadler, and IL Hofacker. "ViennaRNA Package 2.0". In: *Algorithms for Molecular Biology* 6.1 (2011), p. 26. DOI: 10.1186/1748-7188-6-26.

[39] LVD Maaten and GE Hinton. "Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605.

[40] TS Madhulatha. "An Overview on Clustering Methods". In: (2012). URL: http://arxiv.org/abs/1205.1117.

[41] C Malzer and M Baum. “A Hybrid Approach To Hierarchical Density-based Cluster Selection”. In: *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. IEEE, 2020, pp. 223–228. DOI: 10.1109/MFI49285.2020.9235263.

[42] M Mann, PR Wright, and R Backofen. “IntaRNA 2.0: enhanced and customizable prediction of RNA–RNA interactions”. In: *Nucleic Acids Research* 45 (W1 2017), W435–W439. DOI: 10.1093/nar/gkx279.

[43] L McInnes, J Healy, and S Astels. “hdbscan: Hierarchical density based clustering”. In: *The Journal of Open Source Software* 2.11 (2017), p. 205. DOI: 10.21105/joss.00205.

[44] L McInnes, J Healy, and J Melville. “UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction”. In: (2020). URL: http://arxiv.org/abs/1802.03426.

[45] W McKinney. “Data Structures for Statistical Computing in Python”. In: *Proceedings of the 9th Python in Science Conference*. Ed. by Svd Walt and J Millman. 2010, pp. 56–61. DOI: 10.25080/Majora-92bf1922-00a.

[46] WN Moss, SF Priore, and DH Turner. “Identification of potential conserved RNA secondary structure throughout influenza A coding regions”. In: *RNA* 17.6 (2011), pp. 991–1011. DOI: 10.1261/rna.2619511.

[47] D Moulavi, PA Jaskowiak, RJGB Campello, A Zimek, and J Sander. “Density-Based Clustering Validation”. In: *Proceedings of the 2014 SIAM International Conference on Data Mining* (2014), pp. 839–847. DOI: 10.1137/1.9781611973440.96.

[48] DW Mount. “Comparison of the PAM and BLOSUM Amino Acid Substitution Matrices”. In: *Cold Spring Harbor Protocols* 2008.6 (2008), pdb.ip59–pdb.ip59. DOI: 10.1101/pdb.ip59.

[49] D Mudhakir and H Harashima. “Learning from the Viral Journey: How to Enter Cells and How to Overcome Intracellular Barriers to Reach the Nucleus”. In: *The AAPS Journal* 11.1 (2009), p. 65. DOI: 10.1208/s12248-009-9080-9.

[50] F Murtagh and P Contreras. “Algorithms for hierarchical clustering: an overview”. In: *WIREs Data Mining and Knowledge Discovery* 2.1 (2012), pp. 86–97. DOI: 10.1002/widm.53.

[51] SB Needleman and CD Wunsch. “A general method applicable to the search for similarities in the amino acid sequence of two proteins”. In: *Journal of Molecular Biology* 48.3 (1970), pp. 443–453. DOI: 10.1016/0022-2836(70)90057-4.

[52] MI Nelson, C Viboud, L Simonsen, RT Bennett, SB Griesemer, K St. George, J Taylor, DJ Spiro, NA Sengamalay, E Ghedin, JK Taubenberger, and EC Holmes. “Multiple Reassortment Events in the Evolutionary History of H1N1 Influenza A Virus Since 1918”. In: *PLoS Pathogens* 4.2 (2008). Ed. by Y Kawaoka, e1000012. DOI: 10.1371/journal.ppat.1000012.

[53] C Notredame, DG Higgins, and J Heringa. “T-coffee: a novel method for fast and accurate multiple sequence alignment 1 1Edited by J. Thornton”. In: *Journal of Molecular Biology* 302.1 (2000), pp. 205–217. DOI: 10.1006/jmbi.2000.4042.

[54] J Oxford and D Hockley. “Chapter 15 Orthomyxoviridae”. In: *Perspectives in Medical Virology*. Vol. 3. Elsevier, 1987, pp. 213–232. ISBN: 978-0-444-80879-0.

[55] J Parker. “Errors and alternatives in reading the universal genetic code”. In: *Microbiological Reviews* 53.3 (1989), pp. 273–298. DOI: 10.1128/mr.53.3.273-298.1989.

[56] K Pearson. “LIII. *On lines and planes of closest fit to systems of points in space*”. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), pp. 559–572. DOI: 10.1080/14786440109462720.

[57] F Pedregosa, G Varoquaux, A Gramfort, V Michel, B Thirion, O Grisel, M Blondel, P Prettenhofer, R Weiss, V Dubourg, J Vanderplas, A Passos, D Cournapeau, M Brucher, M Perrot, and E Duchesnay. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[58] JK Pfeiffer and K Kirkegaard. “Increased Fidelity Reduces Poliovirus Fitness and Virulence under Selective Pressure in Mice”. In: *PLoS Pathogens* 1.2 (2005). Ed. by M Manchester, e11. DOI: 10.1371/journal.ppat.0010011.

[59] A Phillips, D Janies, and W Wheeler. “Multiple Sequence Alignment in Phylogenetic Analysis”. In: *Molecular Phylogenetics and Evolution* 16.3 (2000), pp. 317–330. DOI: 10.1006/mpev.2000.0785.

[60] RM Pielak and JJ Chou. “Influenza M2 proton channels”. In: *Biochimica et Biophysica Acta (BBA) - Biomembranes* 1808.2 (2011), pp. 522–529. DOI: 10.1016/j.bbamem.2010.04.015.

[61] “Reconsideration of influenza A virus nomenclature: a WHO memorandum”. In: *Bulletin of the World Health Organization* 57.2 (1979), pp. 227–233. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2395768/.

[62] V Satopaa, J Albrecht, D Irwin, and B Raghavan. “Finding a "Kneedle" in a Haystack: Detecting Knee Points in System Behavior”. In: *2011 31st International Conference on Distributed Computing Systems Workshops (ICDCS Workshops)* (2011), pp. 166–171. DOI: 10.1109/ICDCSW.2011.20.

[63] IA Schaap, F Eghiaian, A des Georges, and C Veigel. “Effect of Envelope Proteins on the Mechanical Properties of Influenza Virus”. In: *Journal of Biological Chemistry* 287.49 (2012), pp. 41078–41088. DOI: 10.1074/jbc.M112.412726.

[64] DS Schmeller, F Courchamp, and G Killeen. “Biodiversity loss, emerging pathogens and human health risks”. In: *Biodiversity and Conservation* 29.11 (2020), pp. 3095–3102. DOI: 10.1007/s10531-020-02021-6.

[65] E Schubert, J Sander, M Ester, HP Kriegel, and X Xu. “DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN”. In: *ACM Transactions on Database Systems* 42.3 (2017), pp. 1–21. DOI: 10.1145/3068335.

[66] SciPy 1.0 Contributors, P Virtanen, R Gommers, TE Oliphant, M Haberland, T Reddy, D Cournapeau, E Burovski, P Peterson, W Weckesser, J Bright, SJ van der Walt, M Brett, J Wilson, KJ Millman, N Mayorov, ARJ Nelson, E Jones, R Kern, E Larson, CJ Carey, İ Polat, Y Feng, EW Moore, J VanderPlas, D Laxalde, J Perktold, R Cimrman, I Henriksen, EA Quintero, CR Harris, AM Archibald, AH Ribeiro, F Pedregosa, and P van Mulbregt. “SciPy 1.0: fundamental algorithms for scientific computing in Python”. In: *Nature Methods* 17.3 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[67] PM Sharp and BH Hahn. “Origins of HIV and the AIDS Pandemic”. In: *Cold Spring Harbor Perspectives in Medicine* 1.1 (2011), a006841. DOI: 10.1101/cshperspect.a006841.

[68] LL Shu, YP Lin, SM Wright, KF Shortridge, and RG Webster. “Evidence for Interspecies Transmission and Reassortment of Influenza A Viruses in Pigs in Southern China”. In: *Virology* 202.2 (1994), pp. 825–833. DOI: 10.1006/viro.1994.1404.

[69] MJ Smola, GM Rice, S Busan, NA Siegfried, and KM Weeks. “Selective 2’-hydroxyl acylation analyzed by primer extension and mutational profiling (SHAPE-MaP) for direct, versatile and accurate RNA structure analysis”. In: *Nature Protocols* 10.11 (2015), pp. 1643–1669. DOI: 10.1038/nprot.2015.103.

[70] J Steel and AC Lowen. “Influenza A Virus Reassortment”. In: *Influenza Pathogenesis and Control - Volume I*. Ed. by RW Compans and MBA Oldstone. Vol. 385. Cham: Springer International Publishing, 2014, pp. 377–401. ISBN: 978-3-319-11154-4.

[71] DL Suarez. “Evolution of avian influenza viruses”. In: *Veterinary Microbiology* 74.1 (May 2000), pp. 15–27. DOI: 10.1016/S0378-1135(00)00161-9.

[72] N Suwantarat and A Apisarnthanarak. “Risks to healthcare workers with emerging diseases: lessons from MERS-CoV, Ebola, SARS, and avian flu”. In: *Current Opinion in Infectious Diseases* 28.4 (2015), pp. 349–361. DOI: 10.1097/QCO.0000000000000183.

[73] LH Taylor, SM Latham, and ME woolhouse. “Risk factors for human disease emergence”. In: *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences* 356.1411 (2001). Ed. by MEJ Woolhouse and C Dye, pp. 983–989. DOI: 10.1098/rstb.2001.0888.

[74] JD Thompson, DG Higgins, and TJ Gibson. “CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice”. In: *Nucleic Acids Research* 22.22 (1994), pp. 4673–4680. DOI: 10.1093/nar/22.22.4673.

[75] K Van Reeth. “Avian and swine influenza viruses: our current understanding of the zoonotic risk”. In: *Veterinary Research* 38.2 (2007), pp. 243–260. DOI: 10.1051/vetres:2006062.

[76] JN Varghese, WG Laver, and PM Colman. “Structure of the influenza virus glycoprotein antigen neuraminidase at 2.9 Å resolution”. In: *Nature* 303.5912 (1983), pp. 35–40. DOI: 10.1038/303035a0.

[77] R Verity, LC Okell, I Dorigatti, P Winskill, C Whittaker, N Imai, G Cuomo-Dannenburg, H Thompson, PGT Walker, H Fu, A Dighe, JT Griffin, M Baguelin, S Bhatia, A Boonyasiri, A Cori, Z Cucunubá, R FitzJohn, K Gaythorpe, W Green, A Hamlet, W Hinsley, D Laydon, G Nedjati-Gilani, S Riley, S van Elsland, E Volz, H Wang, Y Wang, X Xi, CA Donnelly, AC Ghani, and NM Ferguson. “Estimates of the severity of coronavirus disease 2019: a model-based analysis”. In: *The Lancet Infectious Diseases* 20.6 (2020), pp. 669–677. DOI: 10.1016/S1473-3099(20)30243-7.

[78] A Viehweger, M Hoelzer, and C Brandt. “Addressing dereplication crisis: Taxonomy-free reduction of massive genome collections using embeddings of protein content”. In: (2019). DOI: 10.1101/855262.

[79] A Viehweger, S Krautwurst, DH Parks, B König, and M Marz. “An encoding of genome content for machine learning”. In: (2019). DOI: 10.1101/524280.

[80] M Vignuzzi and R Andino. “Closing the gap: the challenges in converging theoretical, computational, experimental and real-life studies in virus evolution”. In: *Current Opinion in Virology* 2.5 (2012), pp. 515–518. DOI: 10.1016/j.coviro.2012.09.008.

[81] J Wahlgren. “Influenza A viruses: an ecology review”. In: *Infection Ecology & Epidemiology* 1.1 (2011), p. 6004. DOI: 10.3402/iee.v1i0.6004.

[82] ML Waskom. “seaborn: statistical data visualization”. In: *Journal of Open Source Software* 6.60 (2021), p. 3021. DOI: 10.21105/joss.03021.

[83] RG Webster. “Chapter 14 Antigenic Variation in Influenza Viruses”. In: *Origin and Evolution of Viruses*. Elsevier, 1999, pp. 377–390. ISBN: 978-0-12-220360-2.

[84] CJ Wei, MC Crank, J Shiver, BS Graham, JR Mascola, and GJ Nabel. “Next-generation influenza vaccines: opportunities and challenges”. In: *Nature Reviews Drug Discovery* 19.4 (2020), pp. 239–252. DOI: 10.1038/s41573-019-0056-x.

[85] WHO/OIE/FAO H5N1 Evolution Working Group. “Continued evolution of highly pathogenic avian influenza A (H5N1): updated nomenclature: H5N1 clade nomenclature update”. In: *Influenza and Other Respiratory Viruses* 6.1 (2012), pp. 1–5. DOI: 10.1111/j.1750-2659.2011.00298.x.

[86] IA Wilson, JJ Skehel, and DC Wiley. “Structure of the haemagglutinin membrane glycoprotein of influenza virus at 3 Å resolution”. In: *Nature* 289.5796 (1981), pp. 366–373. DOI: 10.1038/289366a0.

[87] SS Wong and RJ Webby. “Traditional and New Influenza Vaccines”. In: *Clinical Microbiology Reviews* 26.3 (2013), pp. 476–492. DOI: 10.1128/CMR.00097-12.

List of Figures

List of Tables

A Appendix

All the supplementary material can be found on the attached USB-Stick and with exception of the FASTA file in the GitHub repository[1]. The FASTA can be found FSU-Cloud[2].

`Graphics/`	Folder containing all graphics created with `Inkscape`
`PCA/`	Folder containing all PK and PD method comparison results
`Results/`	Folder containing the result of every segments final clustering
`Thesis/`	Folder containing the components of written the thesis
`UMAP/`	Folder containing all UK and UD method comparison results
`A.fasta`	The FASTA file used in the project
`Clustering.py`	The *Influenza A Virus* clustering tool
`Environment.yml`	The configuration file for recreation of the used environment
`PCA.ipynb`	The pipeline used for the PK and PD method comparison
`README.md`	The instructions for usage of the clustering tool
`Thesis.pdf`	The written thesis
`UMAP.ipynb`	The pipeline used for the UK and UD method comparison

Supplementary tables and graphics directly linked in the thesis can be found on the following pages.

[1] https://github.com/ahenoch/Masterthesis.git

[2] https://cloud.uni-jena.de/s/fYkQ2NAwjND8oEM

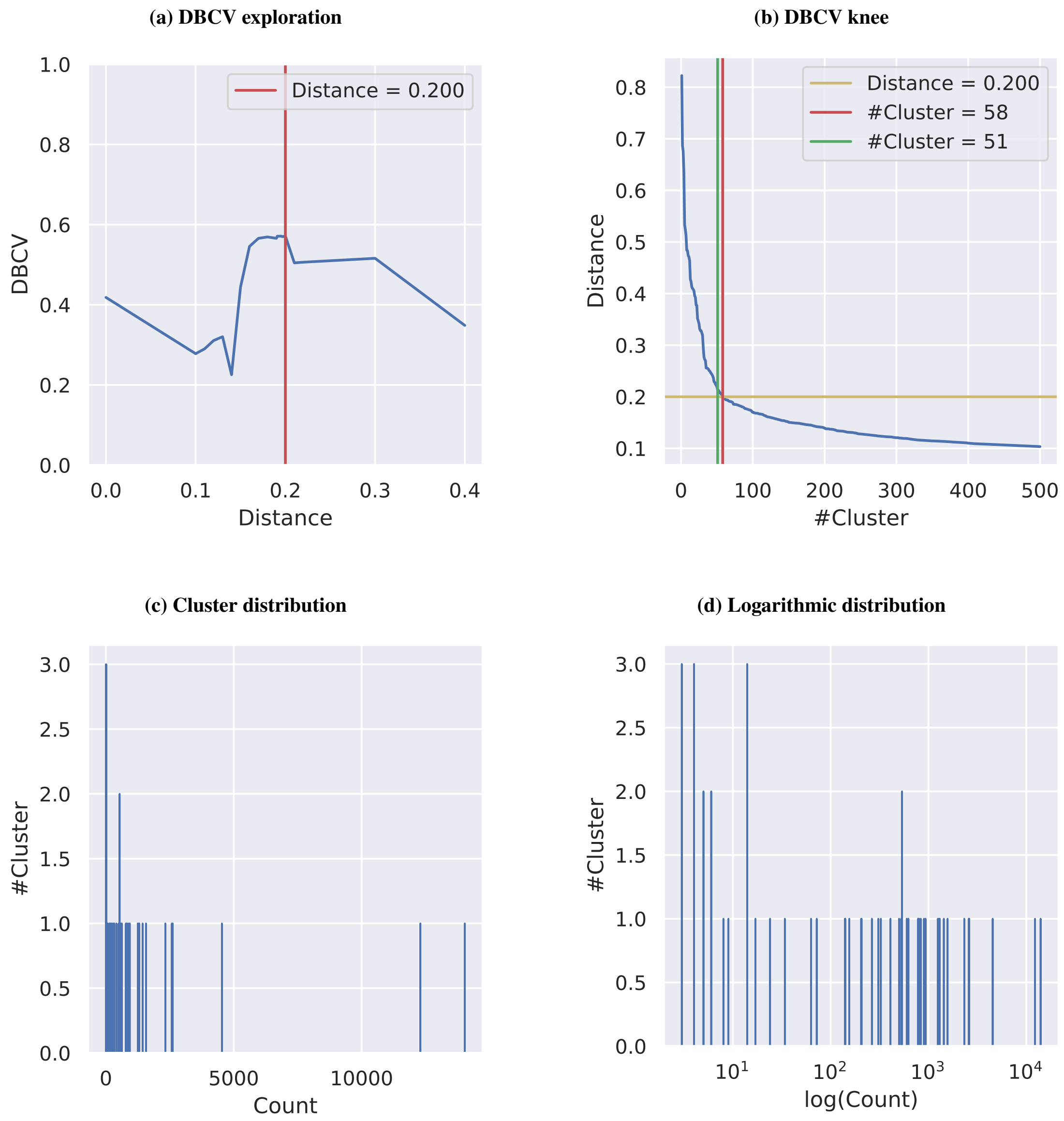


Fig. A.1 Clustering of segment 4 with PD. Segment 4 clustering, using the combination of PCA and the density based cluster validity (DBCV) exploration (PD) results in the given figure. The blue line is the DBCV value resulting from hybrid HDBSCAN clustering with a given distance value ε. The highest DBCV value and, therefore,resulting ε value is described with the red line. The top right subfigure shows the absolute relation of the distance in the single linkage tree to the total number of clusters as the blue line. With the red line, the number of raw clusters, prior to the HDBSCAN part of the hybrid clustering is marked and the final cluster number after it in green. The yellow line describes the threshold, extracted from the maximum DBCVs distance threshold ε, used to perform the hybrid clustering and to get the final cluster number. The red line in the top left subfigure denotes, thus, the same value as the yellow line in the top right subfigure, the distance threshold ε located by the DBCV exploration. The bottom subfigures give information about the distribution of the clusters sizes in continuous and logarithmic scale.

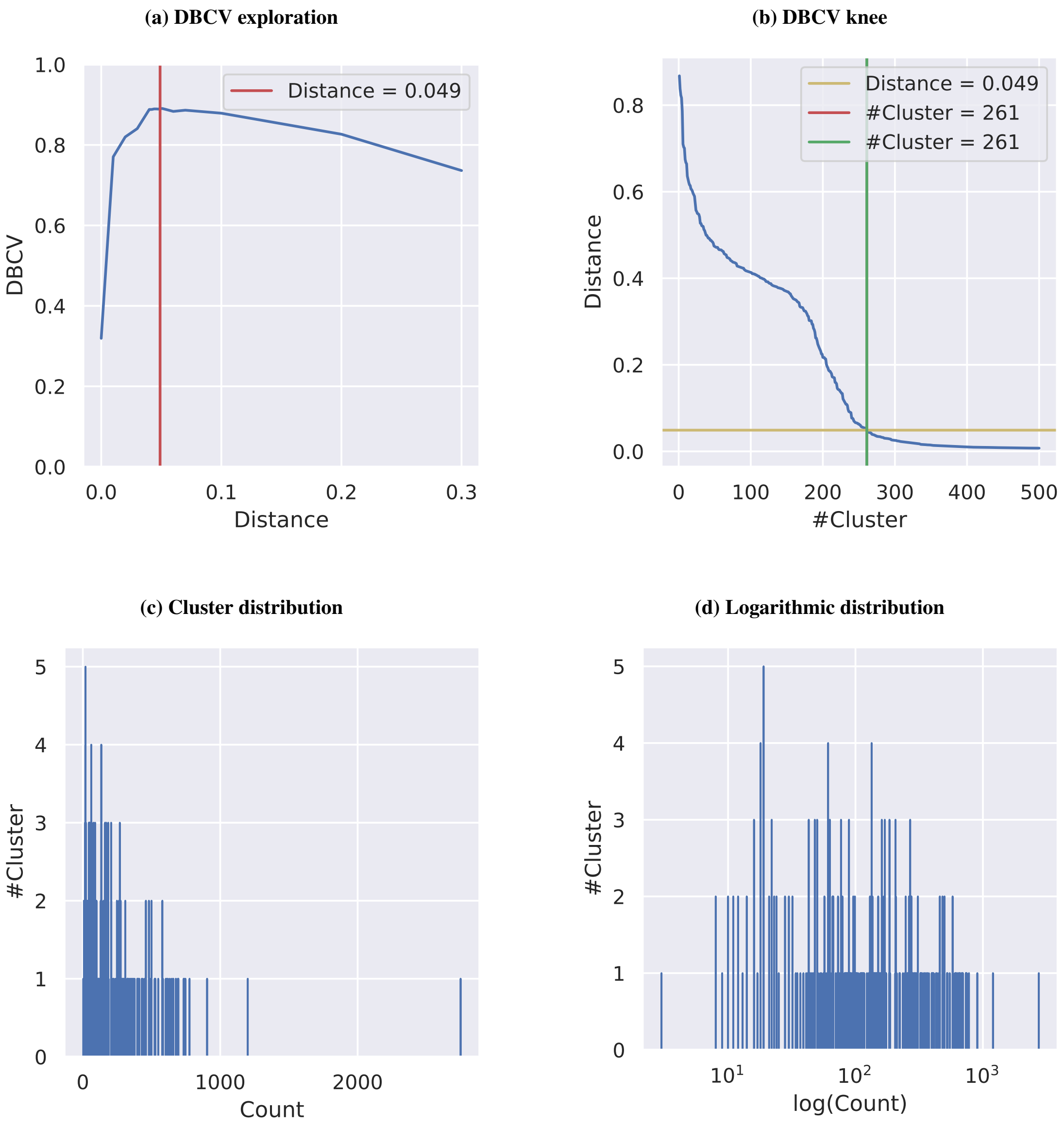


Fig. A.2 Clustering of segment 4 with UD. Segment 4 clustering, using the combination of PCA, UMAP and the DBCV exploration (UD) results in the given figure. The blue line is the DBCV value resulting from hybrid HDBSCAN clustering with a given distance value ε. The highest DBCV value and, therefore, resulting ε value is described with the red line. The top right subfigure shows the absolute relation of the distance in the single linkage tree to the total number of clusters as the blue line. With the red line, the number of raw clusters, prior to the HDBSCAN part of the hybrid clustering is marked and the final cluster number after it in green. The yellow line describes the threshold, extracted from the maximum DBCVs distance threshold ε, used to perform the hybrid clustering and to get the final cluster number. The red line in the top left subfigure denotes, thus, the same value as the yellow line in the top right subfigure,the distance threshold ε located by the DBCV exploration. The bottom subfigures give information about the distribution of the clusters sizes in continuous and logarithmic scale.

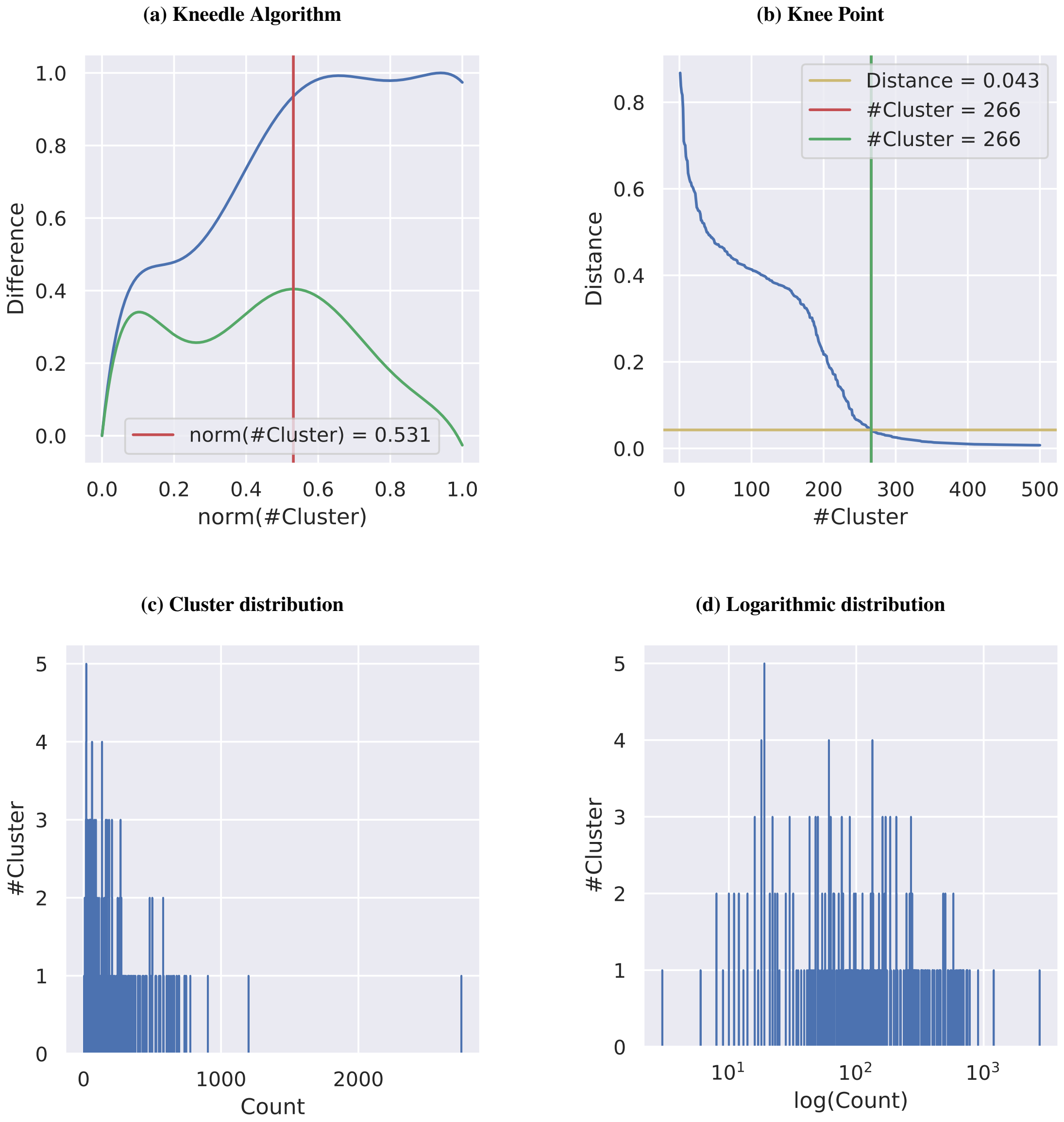


Fig. A.3 Clustering of segment 4 with UK. Segment 4 clustering, using the combination of PCA, UMAP and the Kneedle Algorithm (UK) results in the given figure. The green curve in the top left subfigure describes the change of the distance in the single linkage tree with increasing normalized cluster number and, therefore, the location of the knee, as normalized cluster size at the maximum, highlighted by the red line. The blue line represents the inverse polynomial representation of the blue line in top right subfigure. The top right subfigure shows the absolute relation of the distance in the single linkage tree to the total number of clusters as the blue line. With the red line, the number of raw clusters, prior to the HDBSCAN part of the hybrid clustering is marked and the final cluster number after it in green. The yellow line describes the threshold, extracted from the knees raw cluster number and, therefore, the ε value used to perform the hybrid clustering and to get the final cluster number. The normalized cluster number in the red line in the top left subfigure is the raw cluster number in the top right subfigure calculated on the range of one to 500 clusters, thus, directly derived from it. The bottom subfigures give information about the distribution of the clusters sizes in continuous and logarithmic scale.

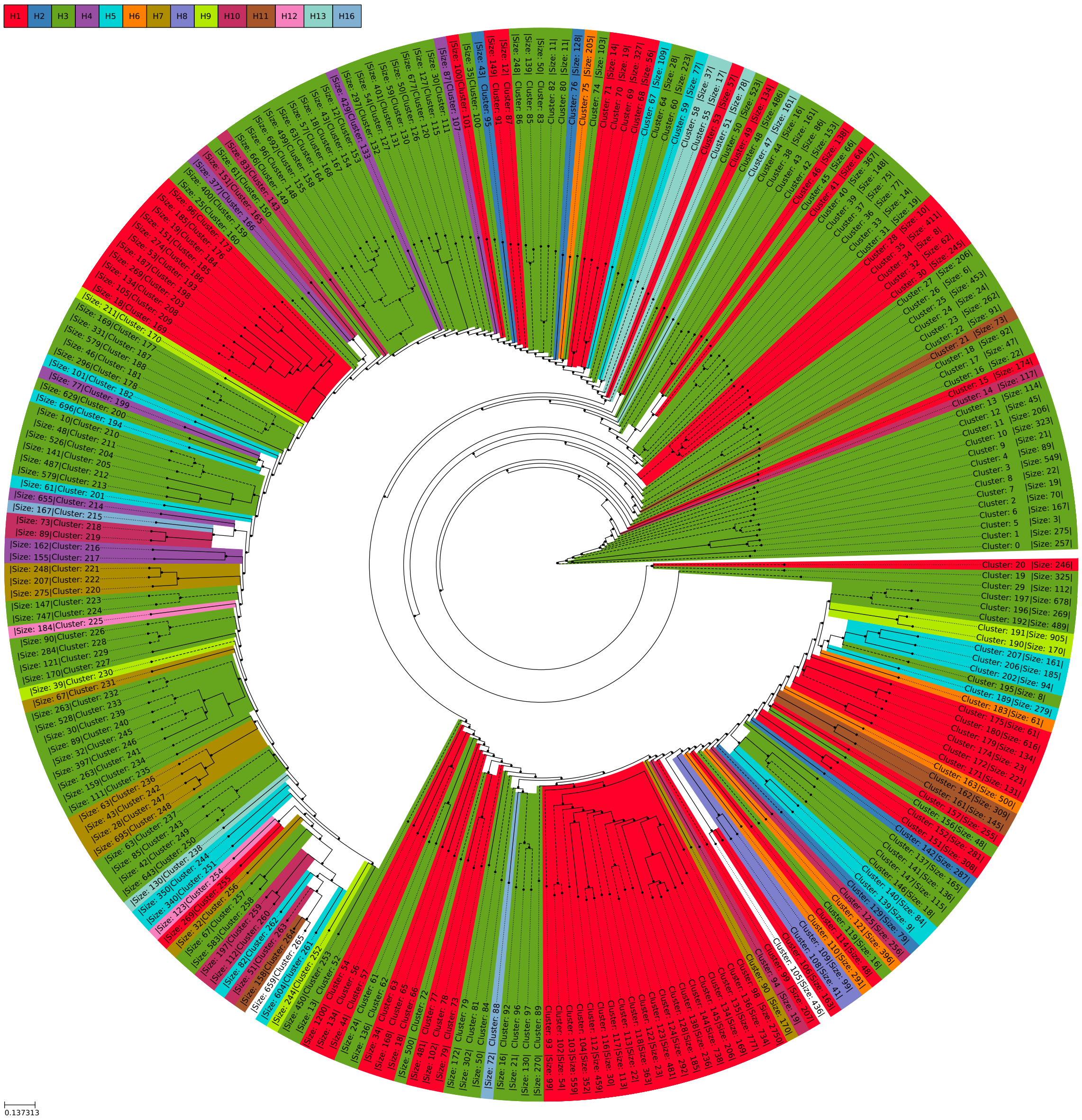


Fig. A.4 Clustering tree of segment 4 with UK. The cluster tree of segment 4 clustering, using the combination of PCA, UMAP and the Kneedle Algorithm (UK) (Fig. A.3). The labeling of the clusters in the tree is based on the subtype of the contained sequences. Unclassified sequences of a cluster were reclassified as a given subtype if sequences of only this subtype are present in the cluster in addition to the unclassified ones. Unlabeled clusters contain sequences from at least two subtypes and zero or more unclassified sequences. Two clusters are mixed since containing sequences of more than one subtype (Table 3.2).

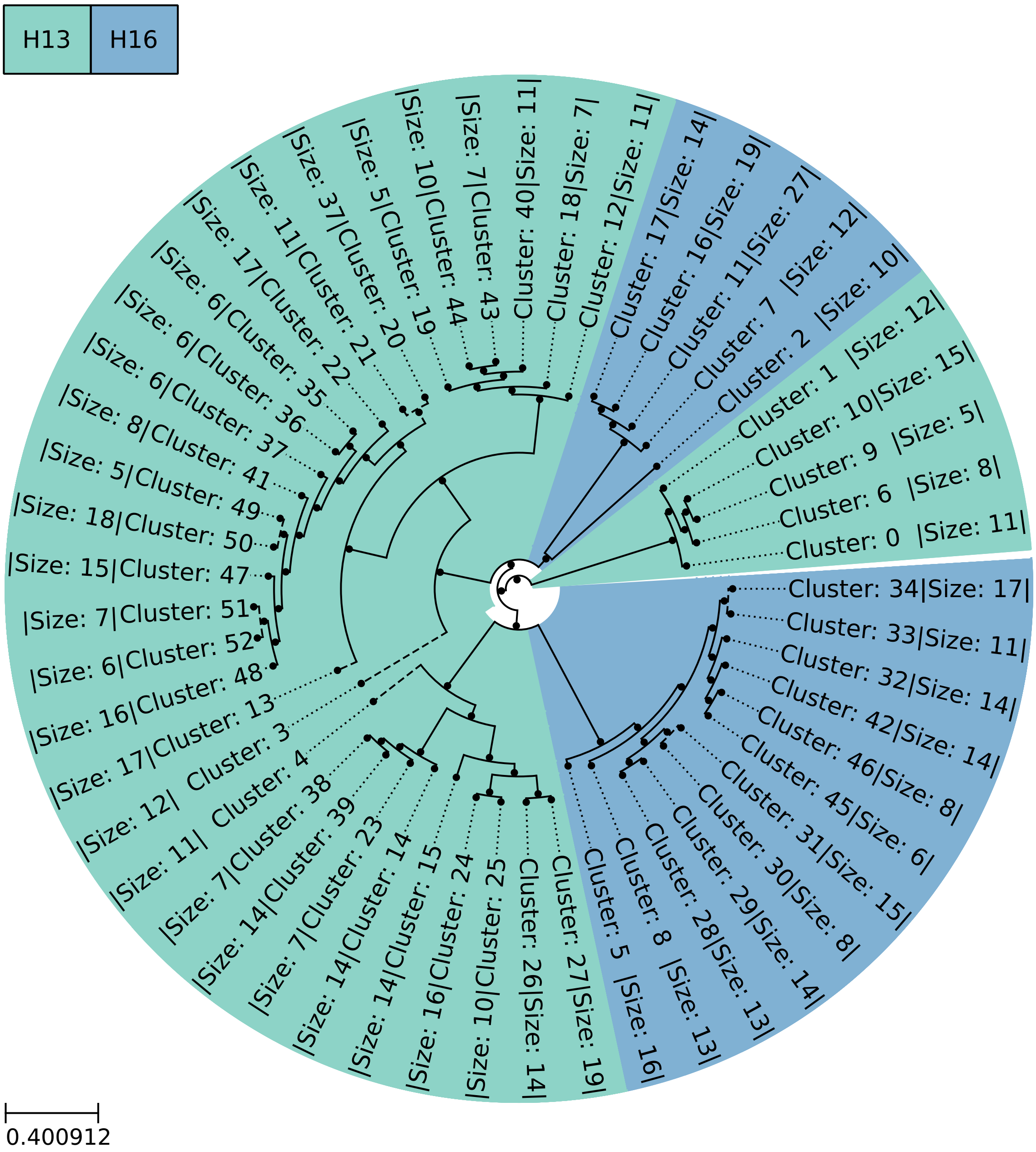


Fig. A.5 Simple clustering tree of H13/H16 with UMAP. Clustering tree, based on the clustering by standard HDBSCAN without ε exploration and hybrid clustering. The used vectors were related to the sequences, present in the H13 and H16 clusters in Fig. 3.4 and reduced by PCA to 100 and afterwards by UMAP to 30 dimensions.

Table A.1 Clustering results. The results of the final clustering using the combination of the `PCA` reduction and the Kneedle Algorithm . Listed is every used segment with the number of raw clusters and the final cluster number after hybrid clustering with the given value of ε. The numbers of mixed cluster numbers of H and N denotes number of clusters that contained vectors related to more than one subtype. The variance is calculated as the sum of the explained variance by the `PCA`.

	#Cluster			**#Mixed**				
Segment	**Final**	**Raw**	**Normalized**	**H**	**N**	**#Unclustered**	ε	**Var**(X)
1	28	71	0.140	19	20	29	0.345	0.811
2	28	66	0.130	17	17	23	0.375	0.798
3	29	70	0.138	18	19	28	0.402	0.816
4	57	78	0.154	1	44	11	0.251	0.793
5	26	74	0.146	22	22	19	0.362	0.837
6	40	58	0.114	28	3	17	0.293	0.803
7	30	75	0.148	16	17	28	0.454	0.858
8	24	67	0.132	16	16	23	0.409	0.860

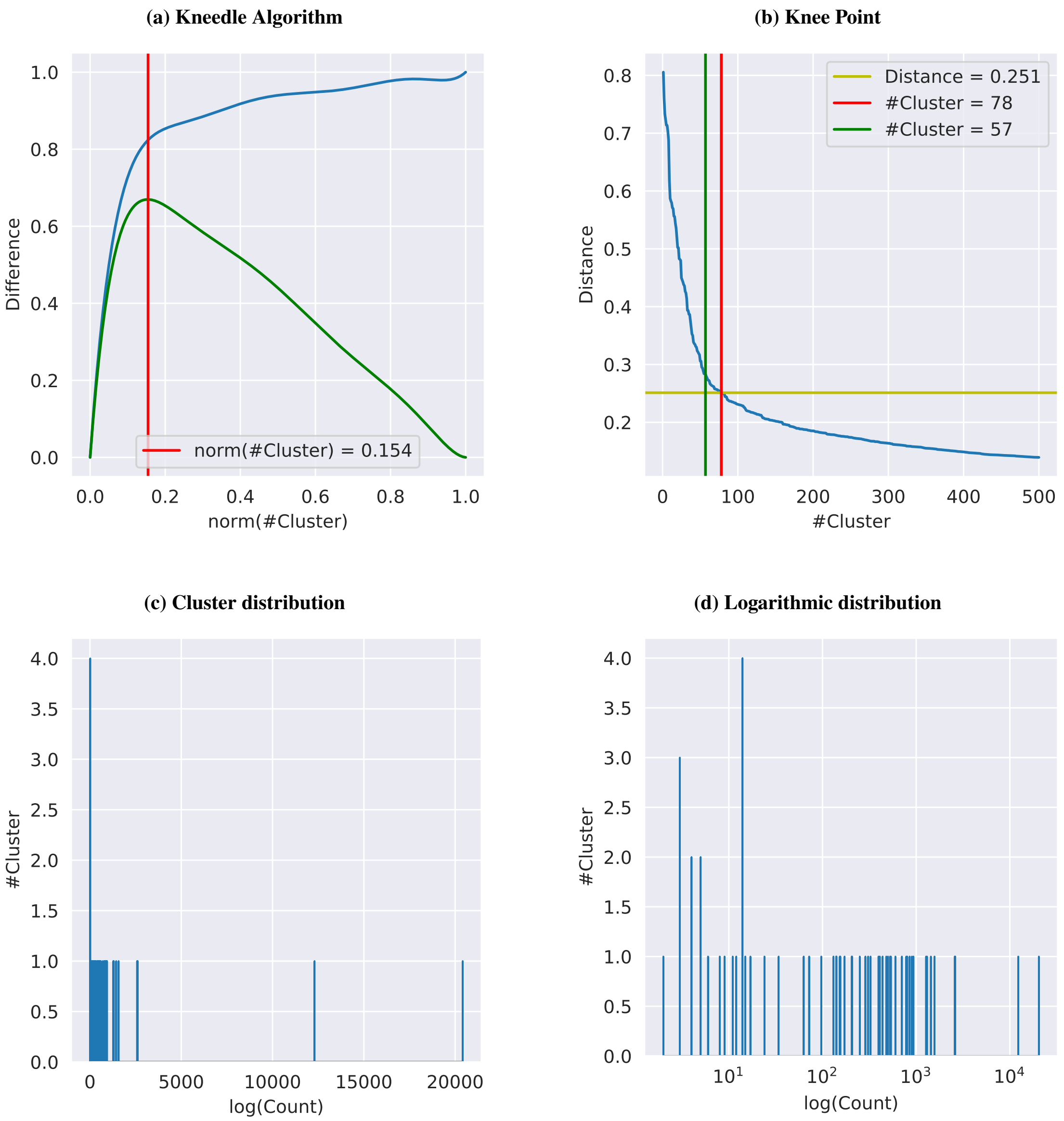


Fig. A.6 Clustering of segment 4. Segment 4 clustering, using the combination of PCA with 50 extracted components and the Kneedle Algorithm results in the given figure. The green curve in the top left subfigure describes the change of the distance in the single linkage tree with increasing normalized cluster number and, therefore, the location of the knee, at the maximum, highlighted by the red line. The blue line represents the inverse polynomial representation of the blue line in top right subfigure. The top right subfigure shows the absolute relation of the distance in the single linkage tree to the total number of clusters as the blue line. The red line, indicates the number of raw clusters, by the DBSCAN part of the hybrid HDBSCAN clustering and the final cluster number in green. The yellow line describes the threshold, extracted from the knee and, therefore, the ε value used to perform the hybrid clustering. The normalized cluster number in the red line in the top left subfigure is equivalent to the raw cluster number in the top right subfigure. The bottom subfigures give information about the distribution of the clusters sizes, by plotting the number of clusters containing a given counted number of sequences in continuous and logarithmic scale.

Selbstständigkeitserklärung

Ich erkläre, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Seitens des Verfassers bestehen (keine) Einwände die vorliegende Masterarbeit für die öffentliche Benutzung im Universitätsarchiv zur Verfügung zu stellen.

Jena, den 14.07.2021

Alexander henoch