

POLITECNICO DI TORINO

Project for the course "Bioinformatics"

# SARS-CoV-2 variants classification and characterization



Students:

Sofia Borgato, s265348  
Marco Bottino, s274110

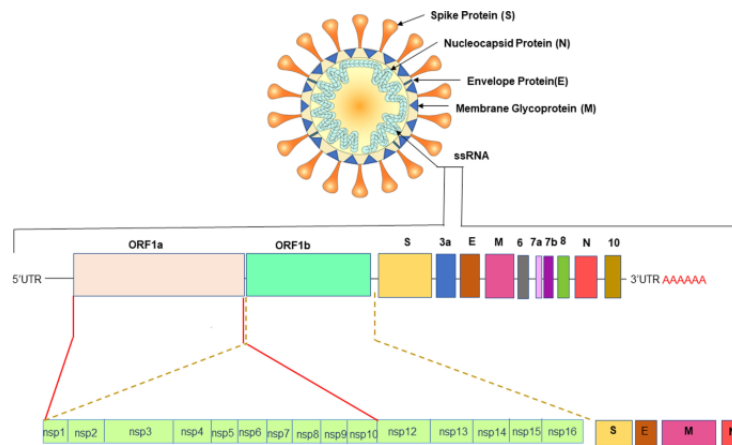
A.Y.  
2020/2021

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Structure of the genome . . . . .	3
1.2	Variants description . . . . .	3
1.3	The dataset . . . . .	4
<b>2</b>	<b>Preprocessing</b>	<b>6</b>
2.1	Alignment . . . . .	6
2.2	Mutation analysis . . . . .	7
<b>3</b>	<b>Supervised classification of the variants</b>	<b>10</b>
3.1	Variants analysis . . . . .	10
3.2	Variants classification . . . . .	17
3.2.1	Random Forest . . . . .	17
3.2.2	Support-vector machine . . . . .	20
3.2.3	Multilayer Perceptron . . . . .	22
3.2.4	Results . . . . .	23
<b>4</b>	<b>Unsupervised classification of the variants</b>	<b>24</b>
4.1	DBSCAN . . . . .	25
<b>5</b>	<b>Discussion and conclusion</b>	<b>27</b>

# Chapter 1

## Introduction



**Figure 1.1:** Structure of SARS-CoV-2 genome

In the end of 2019 a new virus of the species SARS-CoV was spotted in some inhabitants of Chinese region Wuhan. The virus causes a severe respiratory illness later called COVID-19 which led to a global pandemic. A year later, in December 2020, the first vaccine approved by EMA was adopted for EU citizens. At the same time, due to some key mutations in the genome of the virus, new lineages of the viruses, commonly known as variants, began to spread, with the risk of making the vaccines less effective. The idea of this work is to automatize the process of analysis and description of the virus starting from a sample of its genome and to be able to assign a group of samples to the correct variant. By using a clustering algorithm, in the end, it's also possible to distinguish a new variant and obtain a description of its most common mutations.

Gene	Start	End
ORF1ab	266	21555
S (Spike)	21563	25384
ORF3A	25393	26220
E	26245	26472
M	26523	27191
ORF6	27202	27387
ORF7a	27394	27759
ORF7b	27760	27887
ORF8	27894	28259
N	28274	29533
ORF10	29558	29674

**Table 1.1:** Relative position of the genes in reference genome of SARS-CoV-2

## 1.1 Structure of the genome

A sample isolation from pneumonia patients who were some of the workers in the Wuhan seafood market found that strains of SARS-CoV-2 had a length of 29.9 kb. Structurally, SARS-CoV-2 has four main structural proteins including spike (S) glycoprotein, small envelope (E) glycoprotein, membrane (M) glycoprotein, and nucleocapsid (N) protein, and also several accessory proteins. Table 1.1 describes all the different genes and their positions with respect to the reference genome.

## 1.2 Variants description

The know variants we consider in our work are the following ones:

- **Lineage B.1.1.7 (English variant)**, first detected in October 2021, it is correlated with a significant increase in the rate of COVID-19 infection in United Kingdom, associated partly with the N501Y mutation. There is some evidence that this variant has 40–80% increased transmissibility, and early analyses suggest an increase in lethality. More recent work has found no evidence of increased virulence. As of May 2021, Lineage B.1.1.7 has been detected in some 120 countries.
- **Lineage B.1.351 (South-African variant)**, was first detected in South Africa and reported by the country’s health department. Researchers and officials reported that the prevalence of the variant was higher among young people with no underlying health conditions, and by comparison with other variants it is more frequently resulting in serious illness in those cases. The South African health department

also indicated that the variant may have driven the second wave of the COVID-19 epidemic in the country due to the variant spreading at a more rapid pace than other earlier variants of the virus.

- **Lineage P.1 (Brazilian variant)** was detected in Tokyo on January 2021. The new lineage was first identified in four people who arrived in Tokyo having travelled from the Brazilian Amazonas. Later, the Brazil-UK CADDE Centre confirmed 13 local cases of the P.1 new lineage in the Amazon rain forest. A study found that P.1 infections can produce nearly ten times more viral load compared to persons infected by one of the other Brazilian lineages (B.1.1.28 or B.1.195). P.1 also showed 2.2 times higher transmissibility with the same ability to infect both adults and older persons, suggesting P.1 lineages are more successful at infecting younger humans irrespective of sex.
- **Lineage B.1.427/B.1.429 (Californian variant)** was first detected in Fall 2021 in Northern California. CDC has listed B.1.429 and the related B.1.427 as "variants of concern," and cites a preprint for saying that they exhibit a 20% increase in viral transmissibility and moderately reduce neutralization by plasma collected by people who have previously infected by the virus or who have received a vaccine against the virus. After an initial increase, its frequency rapidly dropped from February 2021 as it was being outcompeted by the more transmissible B.1.1.7.
- **Lineage B.1.525 (Nigerian variant)** The first cases were detected in December 2020 in the UK and Nigeria, it had occurred in the highest frequency among samples in the latter country. UK experts are studying it to understand how much of a risk it could be. It is currently regarded as a "variant under investigation", but pending further study, it may become a "variant of concern". B.1.525 appeared to have significant mutations already seen in some of the other newer variants, which is partly reassuring as their likely effect is to some extent more predictable.
- **Lineage B.1.617 (Indian variant)**, was first identified in Maharashtra, India on October 2020, but it reached a global spread in Spring 2021. Emerging research suggests the variant may be more transmissible than previously evolved ones. Whether the effectiveness of currently-deployed vaccines is affected remains under investigation.

### 1.3 The dataset

We downloaded the samples for our work from the global science initiative GISAID<sup>[1]</sup> which provides open access to whole-genome sequences of SARS-

Name	# samples	Submission Period
Original(Wuhan)	1000	01/01/2020 - 24/03/2021
Nigerian	1000	04/01/2021 - 09/04/2021
Californian	1000	04/04/2021 - 09/04/2021
Brazilian	1000	17/03/2021 - 09/04/2021
South African	1000	04/04/2021 - 09/04/2021
English	1000	08/04/2021 - 09/04/2021
Indian	500	01/04/2021 - 22/04/2021

**Table 1.2:** Dataset composition

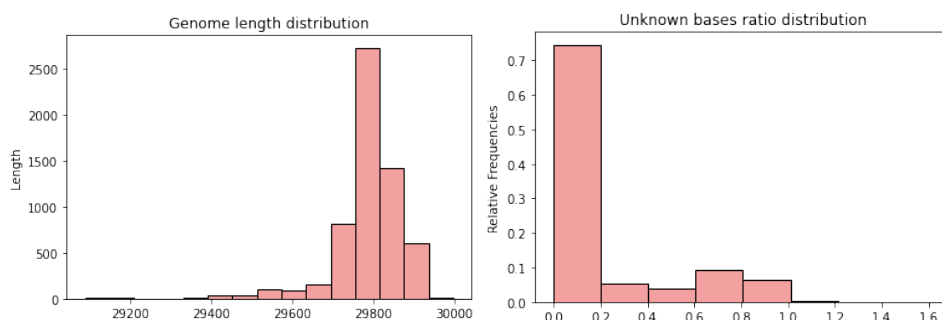
CoV-2. We downloaded 7 FASTA files containing genome from the original Wuhan cases and the 6 variants previously described. At each sample is associated a numeric label for the corresponding variants. If the label is unknown or not provided it will be assigned the label -1. Every sample are *complete* (>29 kb) and *high coverage* (only entries with < 1% undefined bases, < 0.05% of unique amino acid mutations and verified insertions/deletions).

We used the reference genome provided by National Center for Biology Information (NCBI)<sup>[2]</sup>. Table 1.2 describes the composition of the dataset.

## Chapter 2

# Preprocessing

We first read the FASTA files by creating a dataframe with a row for each whole-genome. The genomes are sequences of nucleotides represented by string of letters. When the nucleotide is known the letter can be one between **A** for adenine, **C** for cytosine, **G** for guanine, **T** for thymine. If the nucleotide is unknown different letters can be used, according to the probabilities of being one of the previous bases. We decided to change all of these letters with **Xs** to symbolise the unknown bases in the sequence.



### 2.1 Alignment

The second step of the preprocessing phase consists in the alignment of the sequences to the reference genome by NCBI. This is a required step to be able to highlight and describe the mutations characterising each genome. We tried both global and local alignment approaches: the latter proved to be less sensible to the presence of unknown bases in the samples.

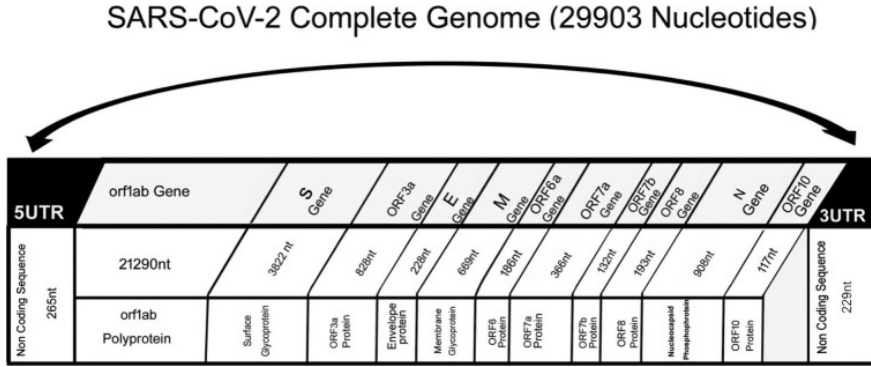
The performance of the aligner varied according to the choice of the different scores.

- **correct match**: score assigned when the basis of the sequence matches the one of the reference. We set 2;

- **mismatch**: score assigned when the basis of the sequence doesn't match the one of the reference. We set -0.1;
- **gap**: score assigned when a first gap is inserted in the sequence or in the reference. We set -2;
- **repeated gap**: score assigned when there's more than one consecutive gap in the sequence or in the reference. We set -0.2;

We empirically found these parameters to be optimal for the alignment of SARS-CoV-2 sequences. This combination, in fact, is stable to the presence of sequences of Xs, which have to be considered as mismatches.

## 2.2 Mutation analysis



**Figure 2.1:** Description of the genome structure in SARS-CoV-2

The third step of the preprocessing phase consists in a comparison between the aligned sequences and the reference genome.

The uptake of working with aligned sequences is that it allows to split them into different genes according to the division of the reference (see Table 1.1) and evaluate the mutations separately for each gene. The results of this evaluation are then summarized by three output dataframes.

### Gene sequences dataset

The first dataset contains in each row the sequence split according to the gene division in figure 2.1 and the label of the variant. This dataset can be adopted in order to perform a classification task in the deep learning field able to discriminate between different variants.









## Mutations statistics dataset

The second dataset describes numerically the kind and the number of mutations divided by region. The mutations can be divided according to the following kinds:

- **Substitution:** single base substitutions, or point mutations, happen whenever a base changes from the reference to the sequence which is being compared. This can lead to a change of amino acid, according to which it can be done a further division:
  - ❖ **Silent:** mutations which code for the same amino acid and don't affect the functioning of the protein
  - ❖ **Nonsense:** mutations that result in a premature termination codon which signals the end of translation. This interruption causes the protein to be abnormally shortened. The number of amino acids lost mediates the impact on the protein's functionality and whether it will function whatsoever.
  - ❖ **Missense:** mutations that occur when base substitution results in the generation of a codon that specifies a different amino acid and hence leads to a different polypeptide sequence. Depending on the type of amino acid substitution the missense mutation is either conservative or nonconservative.
- **Deletion** mutations where one or more bases are lost from the reference genome. According to the number of lost bases the deletion can be in-frame or it can cause a frameshift which can result in a garbled message or a nonfunctional product
- **Insertion** mutations where one or more bases are added with respect to the reference genome. Once again they can be in-frame or frameshift.

This dataset counts the number of mutations for each region divided according to the kind.

	Point mutations				
	No mutation	Silent	Nonsense	Missense	
				conservative	non-conservative
DNA level	TTC	TTT	ATC	TCC	TGC
mRNA level	AAG	AAA	UAG	AGG	ACG
protein level	Lys	Lys	STOP	Arg	Thr
					

 basic  
 polar

**Figure 2.2:** Examples of substitutions

## **Mutations description dataset**

The third dataset describes the most frequent mutations in the input sample. Every time a mutation is encountered when comparing the sequences and the reference genome, different features are saved to describe it:

- The nucleotide position
- The amino acid position
- The gene/region
- The kind of mutation

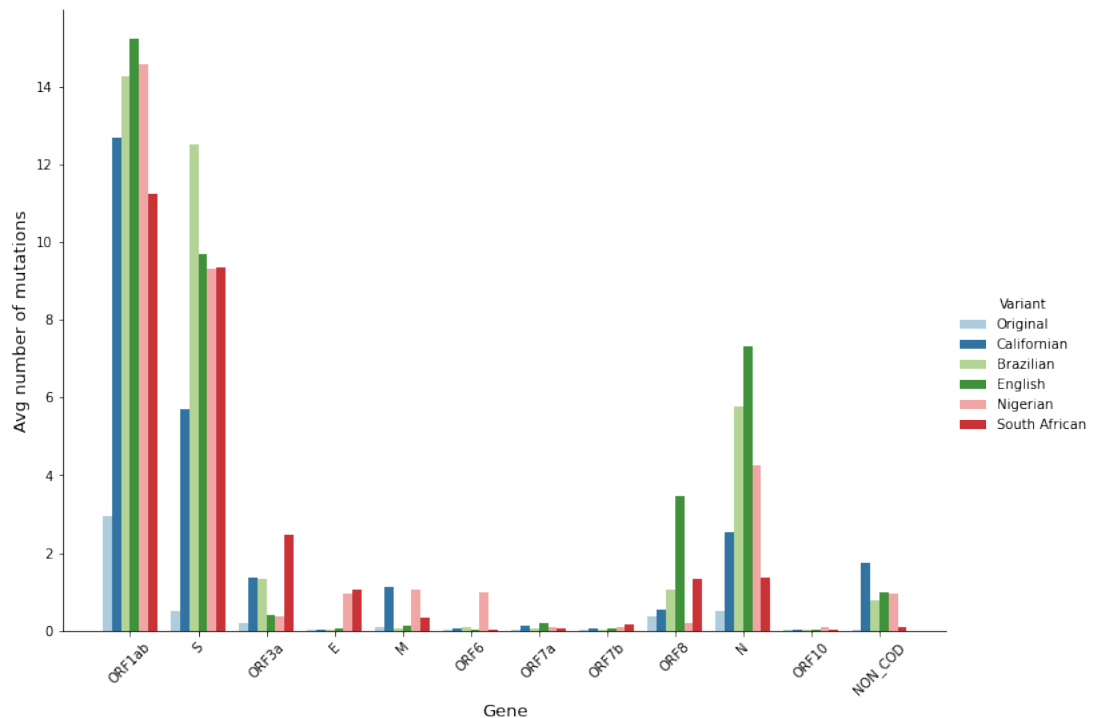
Afterwards, the most frequent mutations are saved in the dataset in descending order according to the frequency.

## Chapter 3

# Supervised classification of the variants

We first applied the preprocessing previously described on our labelled dataset to get a thorough description of the known variants.

### 3.1 Variants analysis



**Figure 3.1:** Distribution of the mutations among the genes for each variant

The plot in Fig. [3.1] shows the average number of mutations in each gene divided according to the variants in the statistics dataset. We can see that most of the mutations occur in the genes N and S, which are structural genes, and ORF1ab, which represents more than 70% of the genome. The variants with most mutations are the English and the Brazilian. We also plotted the average number of mutations in the samples obtained in January 2020 from Wuhan to be used as a control group. They contain some isolate mutation with respect to the reference genome, but the number is significantly lower than the other variants.

We proceed with an example of analysis of the individual variants, using exclusively the results of our preprocessing. The variants are characterized by their most common mutations: we know that the most dangerous ones are the missense non conservative, since they lead to significant changes in the structure of the protein, and the ones which cause frameshift. We also know from the literature<sup>[4]</sup> that E484K/E484Q and L452R are mutations of therapeutic concern since they can make the virus more resistant to the body's immune response, whereas mutation N501Y can increase the virus transmissibility. All of this dangerous mutations happen in the Spike glycoprotein S.

### Lineage B.1.1.7 (English variant)

	Silent	Nonsense	Missense Conservative	Missense Non Conservative	Deletion in Frame	Insertion in Frame	Frameshift
ORF1ab	5.57	0.09	0.66	7.90	1.01	0.00	0.00
S	0.29	0.00	0.11	7.29	1.98	0.00	0.02
ORF3a	0.17	0.03	0.02	0.17	0.00	0.00	0.00
E	0.03	0.00	0.00	0.01	0.00	0.00	0.00
M	0.01	0.04	0.02	0.07	0.00	0.00	0.00
ORF6	0.00	0.00	0.00	0.03	0.00	0.00	0.00
ORF7a	0.06	0.02	0.04	0.06	0.00	0.00	0.02
ORF7b	0.00	0.00	0.00	0.05	0.00	0.00	0.01
ORF8	1.03	0.00	0.02	2.30	0.00	0.05	0.07
N	0.14	0.00	2.02	5.12	0.00	0.00	0.05
ORF10	0.01	0.01	0.00	0.01	0.00	0.00	0.00
NON_COD	0.01	0.01	0.00	0.01	0.00	0.00	0.95

**Figure 3.2:** Average number of mutations in each gene divided by kind - English variant

Mutation (Nucleotide)	Mutation (Aminoacid)	Type	Gene	Percentage
del 11288:11296	-	deletion in frame	ORF1ab	100.0
T24506G	S982A	missense non conservative	S	100.0
T16176C	L5304P	missense non conservative	ORF1ab	100.0
G28882A	R203K	missense conservative	N	100.0
G28881A	R203K	missense conservative	N	100.0
G24914C	D1118H	missense non conservative	S	100.0
C913T	S216S	silent	ORF1ab	100.0
C5986T	F1907F	silent	ORF1ab	100.0
C5388A	A1708D	missense non conservative	ORF1ab	100.0
C3267T	T1001I	missense non conservative	ORF1ab	100.0
C3037T	F924F	silent	ORF1ab	100.0
C23604A	P681H	missense non conservative	S	100.0
C23271A	A570D	missense non conservative	S	100.0
C15279T	T5005I	missense non conservative	ORF1ab	100.0
C14676T	P4804L	missense non conservative	ORF1ab	100.0
C14408T	L4715L	silent	ORF1ab	100.0
A23403G	D614G	missense non conservative	S	100.0
G28048T	STOP52Y	missense non conservative	ORF8	99.9
C23709T	T716I	missense non conservative	S	99.9
A23063T	N501Y	missense non conservative	S	99.9

**Figure 3.3:** 20 most frequent mutations in samples from English variant

The samples from lineage B.1.1.7 contain on average 40.7 (std 2.5) mutations. They are mainly located in ORF1ab, S, ORF8 and N and they are mostly silent and missense non conservative mutations. Among the most common mutations we can see the deletion of 9 bases in position 11288 and the mutation N501Y, which makes this variant more infective.

### Lineage B.1.351 (South-African variant)

	Silent	Nonsense	Missense Conservative	Missense Non Conservative	Deletion in Frame	Insertion in Frame	Frameshift
ORF1ab	4.44	0.13	1.53	4.15	1.01	0.0	0.00
S	0.30	0.00	1.11	6.93	1.00	0.0	0.00
ORF3a	0.20	1.11	0.01	1.13	0.00	0.0	0.00
E	1.00	0.00	0.00	0.05	0.00	0.0	0.00
M	0.00	0.25	0.02	0.05	0.00	0.0	0.00
ORF6	0.00	0.00	0.00	0.00	0.00	0.0	0.00
ORF7a	0.02	0.00	0.01	0.03	0.00	0.0	0.00
ORF7b	0.00	0.00	0.00	0.14	0.00	0.0	0.00
ORF8	0.02	0.02	0.01	1.24	0.00	0.0	0.04
N	0.08	0.00	0.01	1.28	0.00	0.0	0.00
ORF10	0.00	0.00	0.00	0.00	0.01	0.0	0.02
NON_COD	0.07	0.00	0.00	0.01	0.00	0.0	0.00

**Figure 3.4:** Average number of mutations in each gene divided by kind - South-African variant

Mutation (Nucleotide)	Mutation (Aminoacid)	Type	Gene	Percentage
C28887T	T205I	missense non conservative	N	99.8
C26456T	L71L	silent	E	99.7
C23664T	A701V	missense conservative	S	99.7
A23403G	D614G	missense non conservative	S	99.7
C3037T	F924F	silent	ORF1ab	99.6
A23063T	N501Y	missense non conservative	S	99.6
G25563T	R57I	missense non conservative	ORF3a	99.5
A22206G	D215G	missense non conservative	S	99.5
G5230T	K1655N	missense non conservative	ORF1ab	99.4
A10323G	K3353R	missense conservative	ORF1ab	99.4
C1059T	T265I	missense non conservative	ORF1ab	99.3
del 11288:11296	-	deletion in frame	ORF1ab	99.0
G23012A	E484K	missense non conservative	S	99.0
A21801C	D80A	missense non conservative	S	99.0
C25904T	Q171STOP	non sense	ORF3a	98.7
del 22281:22289	-	deletion in frame	S	97.8
G22813T	K417N	missense non conservative	S	97.8
C14408T	L4715L	silent	ORF1ab	88.7
A2692T	T809T	silent	ORF1ab	85.0
C28253T	H120Y	missense non conservative	ORF8	76.2

**Figure 3.5:** 20 most frequent mutations in samples from South-African variant

The samples from lineage B.1.351 contain on average 32.5 (std 2.3) mutations. This variant contain more mutations than the others in ORF3a and they are mostly missense non conservative. In protein S we can find mutations E484K and N501Y.

### Lineage P.1 (Brazilian variant)

	Silent	Nonsense	Missense Conservative	Missense Non Conservative	Deletion in Frame	Insertion in Frame	Frameshift
ORF1ab	7.61	0.02	0.45	5.33	0.86	0.0	0.01
S	0.45	0.00	0.06	12.01	0.00	0.0	0.00
ORF3a	1.17	0.01	0.01	0.14	0.00	0.0	0.00
E	0.02	0.00	0.00	0.01	0.00	0.0	0.00
M	0.01	0.01	0.00	0.03	0.00	0.0	0.00
ORF6	0.00	0.05	0.00	0.02	0.00	0.0	0.00
ORF7a	0.02	0.01	0.00	0.03	0.00	0.0	0.00
ORF7b	0.00	0.00	0.00	0.01	0.00	0.0	0.00
ORF8	0.02	0.00	0.00	1.00	0.00	0.0	0.00
N	2.64	0.00	1.81	1.30	0.00	0.0	0.00
ORF10	0.00	0.00	0.00	0.00	0.01	0.0	0.01
NON_COD	0.00	0.00	0.00	0.01	0.00	0.0	0.75

**Figure 3.6:** Average number of mutations in each gene divided by kind - Brazilian variant

Mutation (Nucleotide)	Mutation (Aminoacid)	Type	Gene	Percentage
C23525T	H655Y	missense non conservative	S	100.0
A23403G	D614G	missense non conservative	S	100.0
C21614T	L18F	missense non conservative	S	99.9
A23063T	N501Y	missense non conservative	S	99.9
G23012A	E484K	missense non conservative	S	99.8
C3037T	F924F	silent	ORF1ab	99.8
C21638T	P26S	missense non conservative	S	99.8
C14408T	L4715L	silent	ORF1ab	99.8
T733C	D156D	silent	ORF1ab	99.7
T26149C	H253H	silent	ORF3a	99.7
G25088T	V1176F	missense non conservative	S	99.7
A5648C	K1795Q	missense non conservative	ORF1ab	99.6
G17259T	S566I	missense non conservative	ORF1ab	99.5
C24642T	T1027I	missense non conservative	S	99.5
C13860T	T453I	missense non conservative	ORF1ab	99.5
C12778T	Y4171Y	silent	ORF1ab	99.5
C21621A	T20N	missense non conservative	S	99.4
C3828T	S1188L	missense non conservative	ORF1ab	99.3
A6613G	V2116V	silent	ORF1ab	99.2
C2749T	D828D	silent	ORF1ab	99.0

**Figure 3.7:** 20 most frequent mutations in samples from Brazilian variant

The samples from lineage P.1 contain on average 37.9 (std 2.5) mutations. This variant has the highest number of mutation in Spike glycoprotein and they are mostly missense non conservative. Once again we can find both E484K and N501Y.

### Lineage B.1.427/B.1.429 (Californian variant)

	Silent	Nonsense	Missense Conservative	Missense Non Conservative	Deletion in Frame	Insertion in Frame	Frameshift
ORF1ab	6.37	0.37	1.92	3.99	0.03	0.0	0.0
S	1.28	0.01	0.08	4.30	0.03	0.0	0.0
ORF3a	0.11	0.03	0.02	1.21	0.00	0.0	0.0
E	0.01	0.00	0.00	0.01	0.00	0.0	0.0
M	0.01	0.02	0.01	1.10	0.00	0.0	0.0
ORF6	0.01	0.00	0.00	0.04	0.00	0.0	0.0
ORF7a	0.03	0.00	0.02	0.06	0.00	0.0	0.0
ORF7b	0.00	0.00	0.02	0.05	0.00	0.0	0.0
ORF8	0.13	0.00	0.05	0.35	0.00	0.0	0.0
N	0.97	0.00	0.03	1.54	0.00	0.0	0.0
ORF10	0.01	0.01	0.00	0.01	0.00	0.0	0.0
NON_COD	0.04	0.69	0.00	1.01	0.00	0.0	0.0

**Figure 3.8:** Average number of mutations in each gene divided by kind - Californian variant

Mutation (Nucleotide)	Mutation (Aminoacid)	Type	Gene	Percentage
C14408T	L4715L	silent	ORF1ab	99.0
A23403G	D614G	missense non conservative	S	98.9
C3037T	F924F	silent	ORF1ab	98.8
C28887T	T205I	missense non conservative	N	98.8
C26681T	P53S	missense non conservative	M	98.8
C1059T	T265I	missense non conservative	ORF1ab	98.7
G25563T	R57I	missense non conservative	ORF3a	98.6
A28272T	-	missense non conservative	non-coding region	98.6
T22917G	L452R	missense non conservative	S	98.2
G17014T	Q5583H	missense non conservative	ORF1ab	98.2
G22018T	W152C	missense non conservative	S	97.3
G21600T	S13I	missense non conservative	S	97.3
C29362T	F363F	silent	N	79.3
A12878G	I4205V	missense conservative	ORF1ab	68.7
T2597C	L778L	silent	ORF1ab	68.5
T24349C	S929S	silent	S	68.3
G27890T	-	non sense	non-coding region	68.3
C2395T	V710V	silent	ORF1ab	68.3
C8947T	N2894N	silent	ORF1ab	63.2
C12100T	A3945A	silent	ORF1ab	63.2

**Figure 3.9:** 20 most frequent mutations in samples from Californian variant

The samples from lineage B.1.427/B.1.429 contain on average 27.0 (std 3.8) mutations. This variant contains a low number of mutations in the S protein, and most of the mutations are silent. The most dangerous mutation is L452R.



### Lineage B.1.525 (Nigerian variant)

	Silent	Nonsense	Missense Conservative	Missense Non Conservative	Deletion in Frame	Insertion in Frame	Frameshift
ORF1ab	8.94	0.02	1.42	3.19	0.99	0.0	0.00
S	1.28	0.00	1.06	5.05	1.91	0.0	0.01
ORF3a	0.05	0.03	0.04	0.23	0.01	0.0	0.00
E	0.81	0.00	0.00	0.02	0.00	0.0	0.12
M	1.00	0.01	0.00	0.05	0.00	0.0	0.00
ORF6	0.01	0.00	0.00	0.02	0.96	0.0	0.00
ORF7a	0.02	0.01	0.00	0.07	0.00	0.0	0.00
ORF7b	0.06	0.00	0.01	0.00	0.00	0.0	0.01
ORF8	0.02	0.00	0.05	0.10	0.00	0.0	0.00
N	1.11	0.00	1.01	1.17	0.97	0.0	0.00
ORF10	0.00	0.00	0.00	0.01	0.03	0.0	0.04
NON_COD	0.00	0.00	0.00	0.95	0.00	0.0	0.00

**Figure 3.10:** Average number of mutations in each gene divided by kind - Nigerian variant

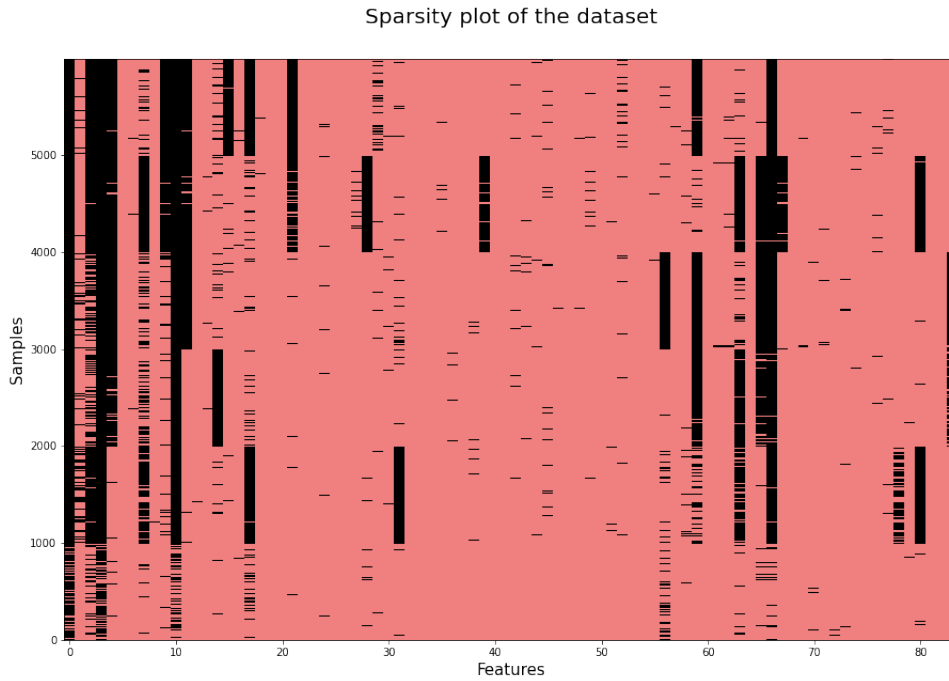
Mutation (Nucleotide)	Mutation (Aminoacid)	Type	Gene	Percentage
C24748T	F1062F	silent	S	99.4
T26767C	Y82Y	silent	M	99.3
C14408T	L4715L	silent	ORF1ab	99.2
A23403G	D614G	missense non conservative	S	99.2
G23012A	E484K	missense non conservative	S	99.1
C3037T	F924F	silent	ORF1ab	99.0
A20724G	STOP6820W	missense non conservative	ORF1ab	99.0
T24224C	F888L	missense non conservative	S	98.9
G23593C	Q677H	missense non conservative	S	98.9
C14407T	H4714H	silent	ORF1ab	98.9
C6285T	T2007I	missense non conservative	ORF1ab	98.8
T8593C	V2776V	silent	ORF1ab	98.7
G2659A	K798K	silent	ORF1ab	98.6
C28887T	T205I	missense non conservative	N	98.6
C18171T	A5969V	missense conservative	ORF1ab	98.6
C1498T	F411F	silent	ORF1ab	98.6
A28699G	P142P	silent	N	98.4
C28308G	A12G	missense conservative	N	97.5
C21762T	A67V	missense conservative	S	97.2
A1807G	G514G	silent	ORF1ab	97.1

**Figure 3.11:** 20 most frequent mutations in samples from Nigerian variant

The samples from lineage B.1.525 contain on average 36.9 (std 2.7) mutations. This is the only variant that contains a mutation in the gene ORF6, a deletion of three bases in position 27205 (not reported in Fig. [3.11]). Moreover it contains a high number of mutations in the ORF1ab region, which are mostly silent, and the mutation E484K.

## 3.2 Variants classification

After the analysis of the variants we proceeded to organize the statistics datasets to be able to build a supervised classifier. We started by merging the datasets into a "global" new dataset, composed by 6000 rows, 84 features and a label column as target. This new dataset is very sparse (see Figure 3.12, therefore we chose to filter the features by dropping the ones where  $> 99\%$  of the elements were null.



**Figure 3.12:** Sparsity plot of the merged dataset

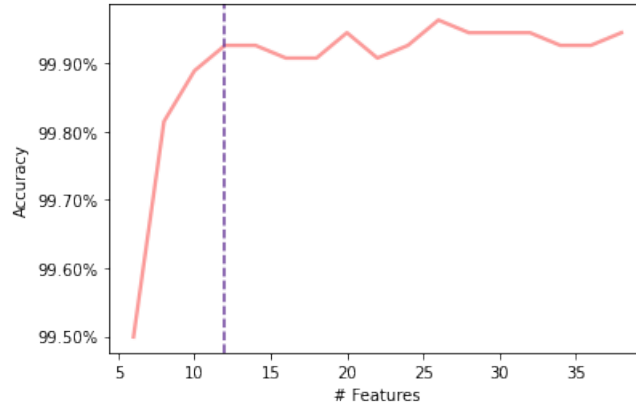
The filtered dataset contains now 39 features. According to the classifier used, we continued the processing in different ways.

### 3.2.1 Random Forest

We chose two different methods of feature selection to optimize the results of a random forest classifier (RF) on our dataset. Both the strategies are supervised and obtain perfect accuracy, but it's interesting to compare the features selected with different methods.

#### Univariate feature selection

This method is based on assigning a score based on a  $\chi^2$  test to every feature and selecting the first  $n$  best according to this score.

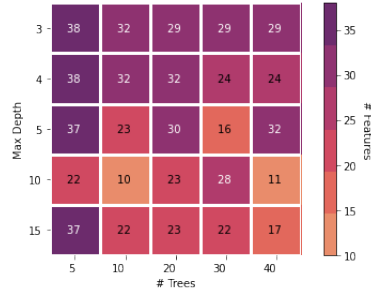


**Figure 3.13:** Value of accuracy with cross-validation according to the number of features. We chose 12 as most suitable number.

We performed a stratified k-fold cross-validation to choose the best value of  $n$ .

In Fig. [3.13] we can see how the mean accuracy in the folds varies according to the number of features selected: we chose  $k = 12$  as a good trade-off between number of features and mean accuracy. The features chosen by the algorithm are the following:

- Number of deletions in ORF1ab
- Number of deletions in S
- Number of missense non conservative substitutions in S
- Number of nonsense substitutions in ORF3a
- Number of silent substitutions in E
- Number of silent substitutions in M
- Number of deletions in ORF6
- Number of missense conservative substitutions in N
- Number of missense non conservative substitutions in N
- Number of deletions in N
- Number of missense non conservative substitutions in non coding regions
- Number of frameshifts in non coding regions



**Figure 3.14:** Optimal number of features according to number of trees and their max depth.

### Recursive feature elimination

This method searches for a subset of features by starting with all the ones in the training dataset and successfully removing them until a minimum number is reached. This is achieved by fitting the given machine learning algorithm used in the core of the model, ranking features by importance, discarding the least important ones, and re-fitting the model. The algorithm returns the best performing subset of features according to the mean accuracy scored with cross-validation. We performed a grid search between the number of estimators (trees) in the model and their maximum depth to obtain high accuracy with a low number of features. All of the models scored  $> 99\%$  accuracy on cross-validation, therefore we only looked for the minimum number of features, as shown in Figure [3.14]. The best model found uses 10 estimators with max depth 10 and performs best with the following features:

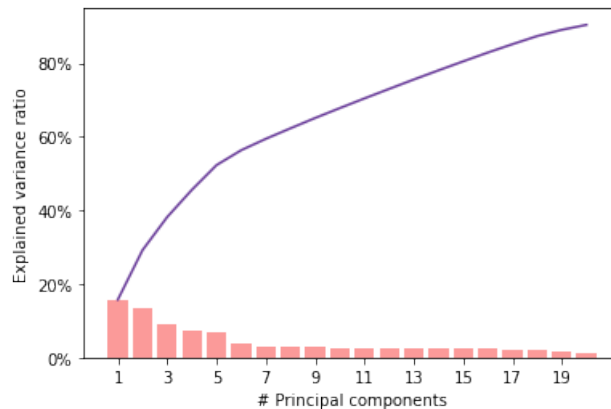
- Number of silent substitutions in ORF1ab
- Number of deletions in ORF1ab
- Number of deletions in S
- Number of missense non conservative substitutions in S
- Number of silent substitutions in E
- Number of silent substitutions in M
- Number of missense conservative substitutions in N
- Number of missense non conservative substitutions in N
- Number of deletions in N
- Number of missense non conservative substitutions in non coding regions

### 3.2.2 Support-vector machine

We also built a support-vector machine (SVM) classifier to try a distance-based approach. To do so we decided to use another approach of dimensionality reduction. Principal component analysis (PCA) is a useful technique which helps to visualize the dataset on a lower dimensional space according to the directions of maximum variance and to improve the performance of classifier based on separating data like SVM.

#### PCA

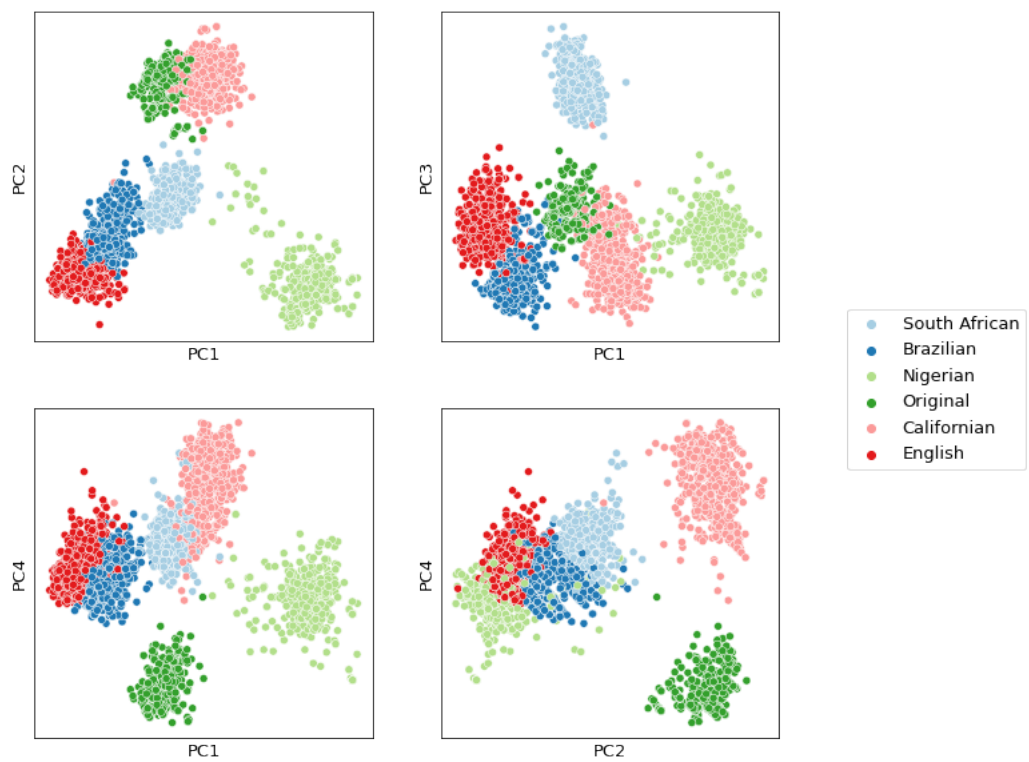
The data exploration phase showed that features have different means and standard deviations. This can affect the quality of PCA, therefore we first performed standard scalarization so that all of the features would have mean 0 and std 1. The variance explained by the principal components is shown in Fig [3.15].



**Figure 3.15:** Explained variance ratio and cumulative variance explained by the first 20 principal components

Since the first 5 principal components already explained more than 50% of the variance, we visualized our dataset projected on the first components with 2D plots (see Fig.[3.16]). Most of this plots show a good division of the variants in clusters, which explains the optimal results reached by all the classifiers.

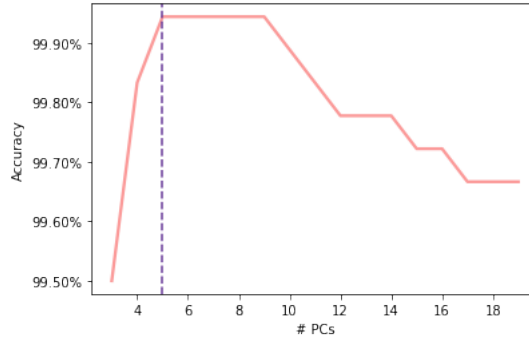
We also investigated the loadings of the features in the first components to try to interpret their meaning, but we didn't find any significant pattern.



**Figure 3.16:** 2D scatter plot of the dataset projected on the first principal components

## SVM

Finally we built a Support Vector classifier on the principal components. We tried different numbers of components, and therefore different dimensions for the subspace where to project the data, and we obtained the best results with 5 PCs (see Figure 3.17). We used a SVM with RBF kernel and the default values for  $C$  and  $\gamma$ .



**Figure 3.17:** Accuracy of SVM after cross-validation on test set with different numbers of PCs

### 3.2.3 Multilayer Perceptron

As last supervised model we chose to use a deep learning method with the Gene sequences dataset described in 2.2. For this task it's necessary to adopt a strategy in order to transform each sequence in numerical array. We had also to keep in mind that each gene has different length. In order to deal with this task we decided to adopt an embedding strategy able to transform each gene sequence in to a well defined numerical array. It's important to notice that in order to score good performance it's necessary to tune the embedding dimension for each gene. We define a specific embedding accordingly with each gene length by dividing that measure by a common factor. The neural network adopted is a Multilayer Perceptron (MLP) : a MLP is a feed forward artificial neural network that is characterized by several layers of input nodes connected as a directed graph between the input nodes and output layers. In particular a 3 dense layer structure is adopted. The number of nodes of each layer might also be sized accordingly to the embedding dimension. The loss used is a Categorical Cross Entropy, the batch size is set to 32 and the number of epoch is set to 60. The different class are represented by integer. With a stratified k-fold cross validation the best parameters found are:

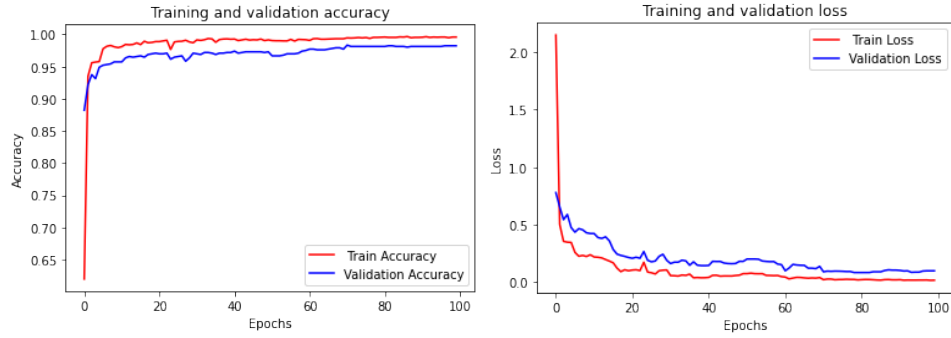
- learning rate:  $10^{-4}$ ;
- embedding dimension: 2921, it means that each gene is encoded and embedded by dividing its length by a factor of 10;

Random Forest with RFE	RandomForest with select k-best	SVM	MLP
1.0	0.98	0.99	0.98

**Table 3.1:** Summary of the different classifiers' accuracies.

- number of nodes in the first dense layer: 256
- number of nodes in the second dense layer: 256
- number of epochs: 100

The result of the best setting are also well described in the plots below.



**Figure 3.18:** Values of accuracy and loss among the epochs with training set and validation set

### 3.2.4 Results

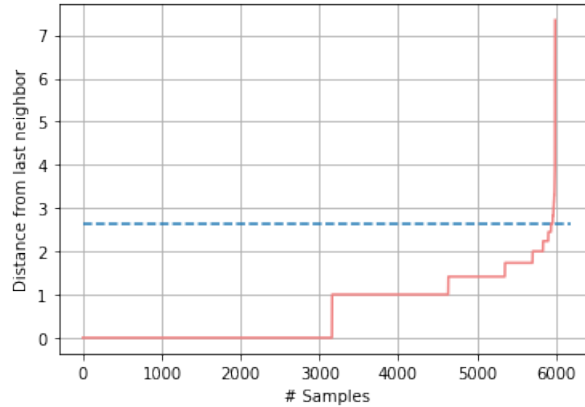
All the models scored almost perfect accuracy as shown in Table 3.1, therefore we chose to use as default in our pipeline the Random Forest with RFE as it proved to be the more stable and fast among the other methods.



## Chapter 4

# Unsupervised classification of the variants

Up until now we considered different types of supervised classification of known variants. Now we want to focus on the task of dividing the samples of a dataset into clusters which can include never seen variants. This can be performed with a density-based clustering algorithm like DBSCAN applied on the dataset of statistics received from the preprocessing. First we worked on the labelled dataset in order to automatize the tuning of the optimal parameters, then we tested the model by concatenating a new dataset containing a new variant to a "training set" containing labelled samples from the known variants in order to see if it were able to create a new cluster.



**Figure 4.1:** Samples sorted by distance to the  $\mu^{th}$  neighbor

/

## 4.1 DBSCAN

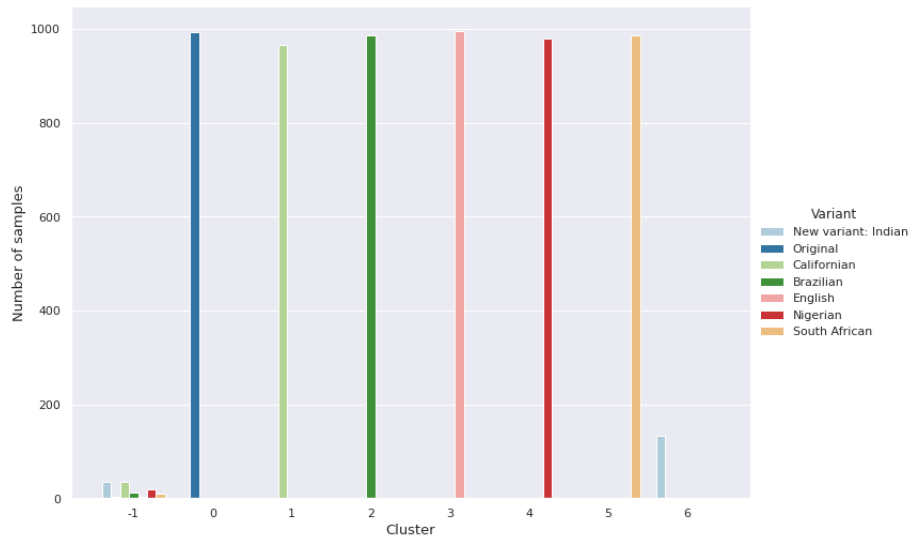
DBSCAN is a spatial clustering algorithm based on grouping together samples if there are at least a minimum number  $\mu$  of points within a distance  $\epsilon$ . We found the best performing distance to be the euclidean distance. According to the values of  $\mu$  and  $\epsilon$  chosen the number of clusters created can vary. An optimal number for  $\mu$  is usually  $2 \cdot \# \text{ features}$ , whereas there's no general rule for the best choice of  $\epsilon$ . According to the paper[3], a suitable value for  $\epsilon$  can be found by calculating the distance to the nearest  $\mu$  points for each point, sorting and plotting the results. Then the optimal value for  $\epsilon$  corresponds to the elbow of this plot. Since the plot 4.1 is a step function, we automatized the choice of the elbow by selecting the index where the distance between two consecutive steps is under a certain threshold. We set empirically the threshold 30 to be the most stable for the optimal choice of  $\epsilon$ . After having set the values of  $\mu$  and  $\epsilon$ , we obtained a division in cluster corresponding to the variants. The number of clusters created corresponds to the number of known variants, a cluster of the outliers and, if data from new variants are present, one cluster for each new variant.

First we tried to make the model capable of discriminating between the six known kind of genomes and optimize  $\epsilon$  and  $\mu$  on a dataset composed by 6000, 1000 for each variants.

We then trained the model on the dataset containing 1000 samples for each of the 5 known variants, 1000 samples from the original Wuhan genome and 168 samples of the Indian variant. The DBSCAN divided the dataset into 8 clusters, corresponding to the 6 original variants, a cluster for the outliers and a cluster for the new variant. The division scored rand index = 0.97 and silhouette = 0.45.

The result are fully described by the plot in 4.2.

If one or more clusters for new variants are created, a new analysis of the key mutations for each variant is performed, so that a detailed description for each new variant can be given as output with a dataset like the third one described in section 2.2.



**Figure 4.2:** Distribution of the original labels among the different clusters. We can see that the clusters from 0 to 5 correspond to the known variants, whereas most of the unknown samples are assigned to a brand new cluster. The cluster -1 collects the outliers.

## Chapter 5

# Discussion and conclusion

Summing up, our pipeline receives as input a FASTA file containing genome sequences from SARS-CoV-2 virus, aligns them to a reference genome with an algorithm of local alignment and compares them to the reference to obtain information about the mutations. This information can be exploited to train supervised classifiers which can assign the samples to the correct variant or to train a clustering algorithm to highlight new clusters of genomes, which could represent new unseen variants.

The main limit of our work is the time used to align the genomes. This process has a very high computational cost since for every genome two strings of 29 kb need to be aligned. We tried to optimize this process, but every sequence needs at least 15/20 seconds to be aligned, which means that the alignment of 1000 samples requires 5/6 hours. This is the true bottleneck of our work, since the rest of the analyses requires few seconds. We also tried to perform classification on the strings without alignment by building, but the results were very disappointing. This is due to the fact that, as we were able to learn from the analysis of the mutations, the differences between the different variants are exaggeratedly small compared to the total length of the string; we are talking about a few characters within a string of almost 30k bases. Moreover, since it is not computationally viable to maintain a sequence of this size, it was necessary to resort to an embedding strategy that is certainly more effective when applied to individual genes than to the total sequence. In the end we decided to use the numeric information about the mutations which can only be obtained by first aligning the strings.

Another limit is in the ability of DBSCAN of finding new clusters with unseen variants: since this algorithm is density-based, it fails in finding a new variant if the number of samples is too small. We managed to obtain good results if the number of samples from the new variant is  $>100$ . On the other hand, it wouldn't make much sense to identify a group of anomalous

samples as "variant" if the number of exemplars is too small.

A further development for this work could be making the training dataset updatable whenever one or more new variants are met. By adding the samples from the new variant to the training dataset and labelling them, the supervised classifier would become able to predict new classes for the following observations. It would also be interesting, by having an objective measure of the effects of the single mutations on the disease, to build a score for how dangerous can be a new variant according to the mutations that are highlighted by our pipeline. This work can be useful in clinical applications to automatize the analyses of the isolated genomes, which would make much faster the identification of new dangerous mutations in clusters of new cases of COVID-19.

# Bibliography

- [1] GISAID Dataset, *gisaid.org*
- [2] Severe acute respiratory syndrome coronavirus 2 isolate Wuhan-Hu-1, complete genome,  
*https : //www.ncbi.nlm.nih.gov/nuccore/NC\_045512*
- [3] N. Rahmah and I. S. Sitanggang, *Determination of Optimal Epsilon (Eps) Value on DBSCAN Algorithm to Clustering Data on Peatland Hotspots in Sumatra*, IOP Conference Series: Earth and Environmental Science, 2016
- [4] Public Health Engalnd, *Investigation of SARS-CoV-2 variants of concern in England*, February 2021
- [5] *https://www.tensorflow.org/*
- [6] *https://scikit-learn.org/stable/*