Bachelor's Thesis

# Visualization Tool for the Geographical Distribution and Genetic Similarity of SARS-CoV-2 in Germany

submitted by

Leon Theis

from San Francisco

Department Algorithmic Bioinformatics
Prof. Dr. Gunnar Klau
Heinrich Heine University Düsseldorf

4. March 2024

| | |
|---|---|
| Primary Reviewer: | Univ.Prof. Dr. G. Klau |
| Secondary Reviewer: | Univ.Prof.Dr. A. Dilthey |
| Supervisor: | Philipp Spohr |

# Acknowledgements

I would like to extend my sincere thanks to Phillip Spohr for supervising and providing assistance in the formulation of this thesis.

I would also like to express my deepest gratitude to Markus Schäfer for providing much needed company during the all nighters I pulled finishing this thesis.

# Abstract

The goal of this paper is to provide a locally runnable Python and Angular based framework that allows for geography based analysis of COVID-19 sequencing data within Germany using the Robert-Koch Institute COVID-19 sequencing dataset. This entails operations that give overview of Pango lineages within a given area as well as the ability to perform alignment operations on individual sequences of the dataset. The application has efficient mechanisms to access sequencing data through the use of memory mapping techniques. The application is built to enable the addition of custom figures without modifying code but adding it. The application's effectiveness was analyzed through two hypothetical use cases: analyzing the impact a policy shift had on lineages in an area and determining if there is a correlation between geographical and genetic distance within given sequences. It was compared to two tools with similar functionality, the Wellcome Sanger Institute's COVID-19 surveillance tool and the Nextstrain ncovid dashboard. The application performed better in the provided use cases than its points of comparison, however during these analyses it was noted that the other applications have features that are possible to implement in the application and would enhance its usability such as heat map visualization on the map and phylogenetic lineage visualization.

# Contents

# Chapter 1

# Introduction

## 1.1 Motivation

From 2020 to 2023 the novel Coronavirus (COVID-19) spread across the world, causing the first global pandemic in nearly a century. Over the course of 4 years there have been over 700 million reported COVID-19 infections, with cases present on every continent [1]. To track changes in the virus that could impact vital metrics like transmissibility or severity of symptoms there was an international effort to sequence the genome of the virus [1]. Initiatives like that of GISAID [2] and the Robert-Koch institute [3]have made this sequencing data public. Over the course of the pandemic tools were developed that provided web based interfaces to analyze and view the sequencing data made public, such as COnVIDa [4], the Nextstrain Ncov Visualization Tool [5], the Düssledorf based SARS-CoV-2 Genomics Dashboard from the Institute of Virology at the University Clinic in Düsseldorf [6], the British Sanger Institute's COVID-19 surveillance tool [7] and the American CDC's variant dashboard [8]. These vary in their exact functionality but all provide interfaces to visualize and analyze the spread of variants of COVID-19 relative to geography. However, none of these tools provide graphical interfaces for comparison of individual sequences, such as sequence alignment.

Starting in mid 2023, many international bodies have considered the COVID-19 pandemic to have passed, and as of `5/5/2023` the World Health Organization no longer considers COVID-19 a public health emergency of international concern [9]. Due to this decreased interest, visualization tools have been discontinued or are no longer maintained. Of the given examples COnVIDa is no longer accessible, with all the links to the tool in the article leading to broken pages, the Düsseldorf based dashboard has not been given new data since 2021, and the Sanger Institute tool only presents data up to February 2023 as the initiative it sourced its data from, the COVID-19 genomics consortium, ceased operations in March 2023 [10]. As many of these tools are web hosted, the utilities that they provided in visualizing data have been lost to the public. The application described in this paper intends to provide a locally runnable platform to enable users to perform various geography based genomic analysis operations on a given COVID-19 dataset.

The goal of this paper is to provide a base framework built using Angular and Python that can be used to analyze genomic data through geographical selection methods with functionality including being able to select and align individual sequences, and assess the Pangolin makeup of a given region at a given time. The application as presented in this paper is configured for Germany, using the Robert-Koch Institute dataset. However, the application is designed to make it easily expandable and work with any given dataset as its secondary purpose is to serve as a base platform for others to build on top of. The application is to be developed and tested on a Lenovo Ideapad Gaming 3-15ARH05 Laptop (16GB Ram, AMD Ryzen 5 4600H).

## 1.2 Structure

The chapter Background will give a brief overview of required knowledge to operate the application. The following chapter, Implementation will provide insight into the structure of the application as well as highlight choices made to improve run time and provide expandability. The chapter Assessment of the Implementation will act as an assessment of the efficacy of the app by summarizing its core functions and providing two use cases where it will be compared against contemporary applications. The last chapter, Conclusion, will provide a summary of the results of the project, drawing conclusions about its efficacy and providing points where it could be improved by future projects.

# Chapter 2

# Background

The background chapter serves to provide a baseline of knowledge to understand the operations performed by the application.

## 2.1 Pango Lineage

The Pango nomenclature is a system used to name different lineages of COVID-19 that was developed in the wake of the pandemic to track the spread of lineages of interest. A lineage is defined by an alphabetical prefix and a numerical suffix, conveying phylogenetic information about the lineage through the naming schema [11]. The system is hierarchical with the numerical suffix allowing up to 3 decimal points to be read as “descendant of previous lineage”. For example, the lineage J.1.2.4 is the fourth named descendant of lineage J.1.2 which in turn is the second named descendant of lineage J.1. [12]

What defines the need for a new lineage is variable. Accepted schemas include transmission to new geographical locations, rapid proportional growth that warrants inspection, being a variant of concern that should be tracked, or simply finding several samples that trend towards the same genetic changes. [12]

## 2.2 Renkonen Similarity Index

The Renkonen Similarity Index is a measure of dissimilarity between populations that is based on proportional presence in a population, defined as:

$$P = \sum min(p_i; p_j)$$

Where $p_i$ and $p_j$ represent the proportional presence of species in both datasets.The resulting $P$ acts as an index of similarity, with 1 representing an identical makeup of species and 0 representing that there are no shared species between the two populations.[13] The index is used in the application to chart a rate of change over a date range by comparing weekly populations.

## 2.3 Sørensen–Dice coefficient

Where $X$ and $Y$ are the count of unique species in each set. The $P$ acts as an index of similarity bound to species present with 1 representing that the two populations have

The Sørensen–Dice coefficient is a measure of similarity between two populations that only considers the difference in present species, defined as:

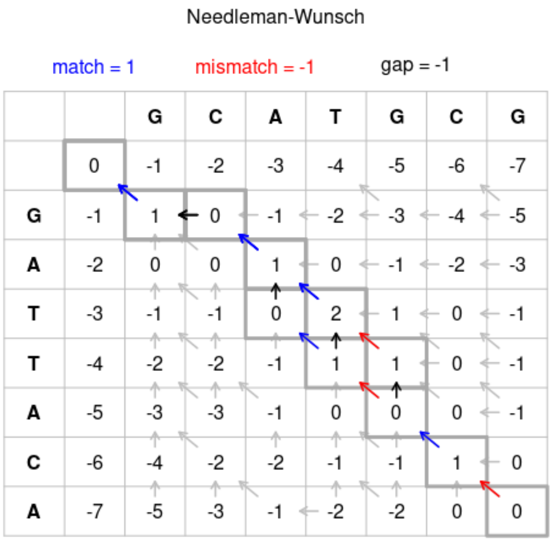
$$P = \frac{2|X \cap Y|}{|X| + |Y|}$$

identical species and 0 representing that there are no shared species between the two populations.[14] This is used in combination with the Renkonen Similarity Index to provide context to the nature of the rate of change.

## 2.4 Sequence Alignment

Sequence alignment is a methodology of arranging strings to identify regions that diverge from one another. In bioinformatics this is used to compare sequences of DNA, RNA and proteins to find areas that may be significant in their function or indicate a shared evolutionary link [15]. There are two generalized methods of sequence alignment, Pairwise Sequence Alignment and Multiple Sequence Alignment. While they differ in methodology the relevant difference for this paper is that pairwise sequence alignment only aligns two strings while multiple sequence alignment aligns several. The application presented in this paper can perform both using the following implementations:

The pairwise algorithm used in this paper is the Needleman-Wunsch algorithm as implemented in the bioPython library [16]. The Needleman-Wunsch algorithm is a dynamic programming approach to finding the global alignment of two sequences. Its methodology involves setting up a matrix wherein the values are dependent on how many elements of the strings match. The values within the matrix are then used to determine an optimal path by backtracking through the generated matrix [17] as seen in 2.1. The path found through this represents the optimal alignment.



| | | G | C | A | T | G | C | G |
|---|---|---|---|---|---|---|---|---|
| | 0 | -1 | -2 | -3 | -4 | -5 | -6 | -7 |
| G | -1 | 1 | 0 | -1 | -2 | -3 | -4 | -5 |
| A | -2 | 0 | 0 | 1 | 0 | -1 | -2 | -3 |
| T | -3 | -1 | -1 | 0 | 2 | 1 | 0 | -1 |
| T | -4 | -2 | -2 | -1 | 1 | 1 | 0 | -1 |
| A | -5 | -3 | -3 | -1 | 0 | 0 | 0 | -1 |
| C | -6 | -4 | -2 | -2 | -1 | -1 | 1 | 0 |
| A | -7 | -5 | -3 | -1 | -2 | -2 | 0 | 0 |

Figure 2.1: A visualization of the score path of the needleman-wunsch algorithm [18]

The bioPython implementation follows the algorithm detailed by Durbin et al. [19]

Multiple Sequence Alignment in this application is handled by the MUSCLE application. The methodology the program uses involves estimating how similar strings are before alignment by comparing how many common substrings they share. This metric is then used to build a tree wherein the leaves are the individual sequences. This tree is then traversed and successively aligns the sequences until a Multiple Sequence Alignment is found at the root. MUSCLE is faster than its contemporaries with relatively little loss in accuracy for its speed. [20]

# Chapter 3

# Implementation

The implementation chapter serves to give an overview of structural decisions made in the application that impact the user experience, performance and expandability.

The application has two primary components, a Python based backend and an Angular based frontend. Communication between the two is achieved through http requests that are handled in the backend through the Flask web api [21]. This separation is to ensure that Angular handles the visualization while Python handles the data processing. The application can also be deployed as a web service if desired. A simplified visualization of the structure described can be seen in 3.1

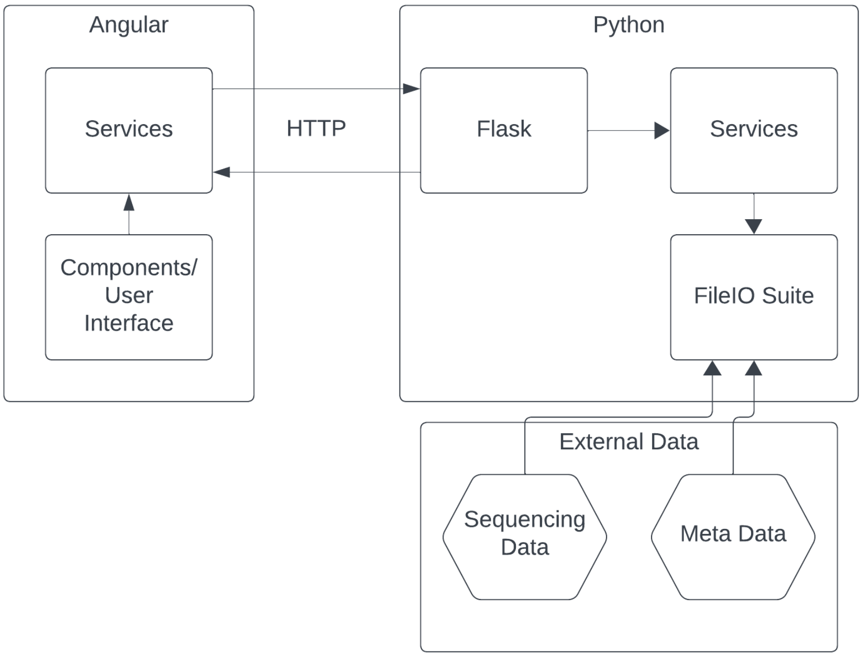


Figure 3.1: Overview of Application Structure

## 3.1 Frontend

The Frontend accepts user input and dynamically displays various figures generated by that input. The framework the Frontend uses to achieve this is Angular. Angular was chosen as it has a module focused design and a two way binding between its template and typescript files, allowing for dynamically rendered templates that influence the component through event bindings without requiring the template to be rebuilt. [22] [23] [24]

The primary feature of the web interface is a map which acts as a selection interface. It takes up the most space on the interface as it is the main context given to the information that is not dependent on user input. Data points do not exclusively have to be selected through the map, there is also a table based interface which contains additional information and is filterable and searchable. The table is hidden behind a button by default and is displayed as a pop out overlay. The table can be displayed in a separate browser tab, a functionality it shares with the figure output window. The figure output window exists to display the figure generated by the operations of the application. The ability to display certain components in separate tabs was implemented to allow the user to arrange the interfaces as they require. Both of these components have their data stored in the local data of the browser with the exception of the table in the sequence mode as that table reaches a size where the JSON parser of javascript takes a not insignificant amount of time, so it is stored in the Backend and is retrieved through a request.

The interface has two distinct modes, separated by the scope of the comparison operations offered. There are comparison operations on a postal code level using Pango lineages and individually between sequences, reflected in two separate data selection mechanisms. The user can change the mode with a toggle button.

## 3.2 Backend

The Backend has two major functions: accepting input from the Frontend and returning a figure as well as providing file access and data processing to the application.

### 3.2.1 File IO

To perform visualization operations the application requires two sets of data: the sequencing data in a text-based format and a metadata file containing an entry for each sequence in TSV format. The sequence data is assumed to be FASTA, a standard defined by a line prefaced with a “>” followed by an ID string and a line break after which follows an alpha numeric representation of a nucleotide or peptide sequence. [25]

The application is developed in a manner that it can be easily modified to accept formats other than FASTA, as the function used to parse the file is being injected in a way that allows it to be adjusted for other file formats. The metadata file is an auxiliary file that contains all information that is not the actual sequence for each sequenced sample, it must contain an ID that matches one found in the sequencing dataset, a geographical location in the form of a postal code, the date the sample was taken in `dd/mm/yy` format and a string used for lineage classification. The column names that the application accepts in the TSV can be changed through environmental variables. The default naming scheme is that found in the metadata file provided by the Robert-Koch institute's corona sequencing data. These two datasets are collectively accessed by the `FileIO` suite of classes which provide an interface to access this information within the application as well as to efficiently parse them into a usable state.

The `FileReader` class acts as an intermediary between the rest of the application and the provided sequencing data file through a memory map storage solution. Memory mapping refers to a methodology of loading files into the virtual memory through demand paging, meaning that pages are loaded into the physical memory based on what the system requests on a moment to moment basis rather than having the entire file loaded into memory. This enables access to specific portions of the file through address based requests, rather than searching through a file for a specific ID [26]. The `FileReader` class contains a dictionary with the sequence ID keys and a tuple containing the start and end addresses of the related sequence in the sequencing data file. These addresses are used when a sequence is requested, the mmap object directly opens the memory of the file at a specific location and retrieves the sequence through these addresses. For scenarios where the size of the file being accessed is larger than the memory the system has, this improves access times as only a single page must be loaded at a time rather than loading the entire file in parts linearly. This allows the application to run on systems that have lower memory and cpu specifications than if the files were used without using Memory Mapping. The built- in Python library `mmap` was used to achieve this.

`FileReader` has two methods of instantiation. The default method has it constructed with a file path, a function pointer and an optional string, it then uses the function in a multithreaded function call to fill the local dictionary with addresses. If the string has a value then the secondary method is used which parses the string as a file path, converting a JSON file into a dictionary to be handed to the object. This overrides the local dictionary without having to parse the file containing the sequences. The secondary instantiation method is meant to be a tool to minimize start up time as even with a multithreaded approach parsing sequence data into the address dictionary can take several minutes. There is a corresponding function to export the current address dictionary. This function is used on every startup where a new address dictionary is generated and can be downloaded from the user interface. Setting the path for this dictionary is determined through the environmental variable `FILEREADER_JSON_PATH_READ`.

`GenomeData` acts as a wrapper for the `FileReader` class, providing the function it uses to populate its address dictionary as well as helper functions such as retrieving the address of a sequence and exporting the address map of `FileReader` as a JSON.

`MetaData` acts as a wrapper to read and interact with the data given in the supplemental metadata TSV

Additionally the structure of the `FileIO` classes ensures that certain data is processed before the application has a usable interface, ensuring that static data sets that don't change over the course of the application are loaded on startup.

### 3.2.2 Data Analysis

**Figure Creation**

Figure creation is handled through the Python backend using the Plotly library. The resulting Plotly figure can be natively displayed by the frontend using the Plotly Javascript library. Figure creation requests are handled through a post request to a single static URL in the backend. The body of this request contains the name of the operation that is to be done and the data that operation will be done with. The name of the operation is parsed as a Python function from the `ComparisonFunction` module and is executed asynchronously. The function with the relevant data as a parameter is used to create the Plotly figure. Once it is done it terminates the thread and pushes the result to an internal variable. The frontend periodically checks with a get request if the figure has been completed. This ensures that the figure creation request does not time out before the figure is finished.

Figure generation was designed to enable expansion. As the controller retrieves a function by name from the `ComparisonService`, implementing an additional figure only requires creating an additional function in the `ComparisonService` module that maintains the standard of returning a Ploty figure or text. The frontend can now use this function with the post request using its name. In this way figure generation requires only adding code, rather than modifying it to expand the app.

To enable the comparison of several postal codes worth of data the application must be able to generate several figures. Generating several figures to display within a single Plotly object necessitated the use of ploty's subplot feature. It should be stated that in the case of attempting to implement an additional figure in `ComparisonService` that Plotly subplot objects do not natively support Plotly express objects. This necessitates extracting the necessary information from the Plotly express object and adding them to the subplot object. The function `translate_express_to_graph_object` can be used as a reference for this process.

**Sequence Alignment**

Pairwise sequence alignment in this application uses the bioPython pairwise aligner module. The pairwise aligner module is capable of using the Needleman-Wunsch, Smith-Waterman, Gotoh (three-state), and Waterman-Smith-Beyer pairwise alignment algorithms, depending on how it is configured [16]. By default the application finds the global alignment using Needleman-Wunsch. This can be changed in the configuration of the aligner module. The user can set the gap, extension, match and mismatch scores for the operation in the Frontend interface. The result is the alignment in text form displayed on the figure interface and is additionally saved to a downloadable text file. More than two sequences can be given as an input to this operation; the result is every sequence aligned with each other. This will not display the alignments, instead displaying the alignment scores of each pair sorted from highest to lowest. To view the full alignments the text file must be downloaded.

Multiple sequence alignment is done through the command line interface MUSCLE and is visualized through the pyMSAviz library. Due to the application not being designed to stop the MUSCLE process the user is prompted to give a timeout when attempting to perform the operation. pyMSAviz gets the resulting alignment and generates a figure as an image [27]. To maximize visual clarity and reduce hardware load the only portions of the resulting alignment that are visualized are those with gaps or mismatches in them, with 15 base pairs before and after to provide context. After the operation is complete the resulting fasta file containing the sequence alignment can be downloaded through a button on the user interface.

# Chapter 4

# Assessment of the Implementation

This section acts as an assessment of how effective the app is. The assessment is achieved through comparison to other applications that provide similar functionality.

It is important to preface these comparisons with a disclaimer that these applications are similar in function but may not be using the same data set. This may lead to unexpected deviation In results, thus these comparisons are being made on a functional level rather than comparing results.

For these comparisons the application was using the Robert-Koch Institute dataset retrieved on 16/11/2023.

## 4.1 Building and Deployment

If the user is performing analysis on a country other than Germany they must exchange the files in the folder `/Data/GeoData` with appropriate shape files containing the vectors for visualization on the map.

To build the application the user must run the Angular command
`ng build -configuration production` at the top level of the Angular project. Afterwards the Frontend and Backend must be bound together by copying the contents of the `/dist/rkidata-viz-frontend/browser` folder into the `templates` folder of the Backend.

Before deploying the application the user must set three environmental variables in the Backend through `enviroment.py`:
`OS_PATH`, which consists of the path up to and including `/RKIDataViz_Backend`, `FASTA_FILE`, which is the path of the file containing the sequencing data to be analyzed and `METADATA_FILE` the path of the metadata file associated with the aforementioned sequencing data.

The application then can be deployed with the provided `run.bat` script. The script checks if the Conda environment the application requires is present and builds if it is not. It then activates the environment and starts the program, automatically opening a browser window once it has finished preprocessing tasks.
For best performance it is recommended the application be run in a Chromium browser.

## 4.2 Functionality

The application can generate 6 distinct figures. These figures can be differentiated into the two different modes of the application as defined in Frontend.

Postal Code mode operations are comparisons of present Pango lineages in a postal code within a given time frame. The figure types that can be generated are further subdivided into relative and absolute figures.
Relative figures are those that the summary interface can generate, they present data in the context of how present a Pango lineage is within a population. The specific figures that can be generated are a pie chart displaying the ratios of each lineage within the given data set or a line graph that uses Renkonen Similarity Index and Sørensen–Dice coefficient to display a rate of change within the present lineages.
Absolute figures are those generated by the other two interfaces, and display data in the context of how many samples are in the dataset without any weighting or proportionality. The figure that the lineage per postal code interface generates is a line plot of how many samples of each Pango lineage were found within the postal code over the given time frame. The figure that the postal code per lineage interface generates is a line plot of every selected lineage displaying how many postal codes of the selected set it is present in the given date range. Both of these functions allow the user to isolate which lineages to show and determine a minimum sample count to remove insignificant data.
Additonally, all postal code mode operations display a map with the geographical distances and a table with the Renkonen-Indexes between selected postal codes.

Sequence mode operations refer to alignment operations performed on individual sequences within the data set. The possible operations are pairwise sequence alignments and multiple sequence alignment and are implemented as detailed in Sequence Alignment.

To assess the efficacy of the application two hypothetical use cases will be outlined, followed by the introduction of two similar applications with their feature sets and performance in the same use cases.

### 4.2.1 Use Cases

**Analysis of Policy Shifts**

A context in which the application could find use is surveying the impact of a shift in lockdown policy on the distribution of Pango lineages in a specific area. For example, this use case will examine the policy shift of the German Coronavirus-Schutzverordnung loosening of mask mandates and quarantine measures on `3/4/2022` [28].

A way to determine this would be to analyze the rate of change of Pango lineages around the time of this event, with focus on shared trends between postal codes. Followed by verification of the rate of change through checking the proportions of each Pangolin in the population before and after the event.
Assessing the raw number of samples for trends in the appearance of lineages could be

a valid approach. However, this methodology is susceptible to noise from inconsistent sample sizes and is only suggested when a curated dataset is present.

The application has been loaded with the Robert-Koch Institute sequencing dataset. The Postal codes that will be inspected are the 7 with the highest number of samples. And as the event being assessed is on the 3/4/2022 the date range to be assessed is approximately 8 weeks before and after the event. This range is 30/01/2022 - 20/06/2022.

As stated above the first step is to retrieve a metric that can be used for rate of change in a date range around the event, this is done using the summary function where a date is entered and a line graph for each postal code is posted with the change of Renkonen Index per week providing a visible representation of the rate of change. The results of the summary function are condensed in the figure 4.1.

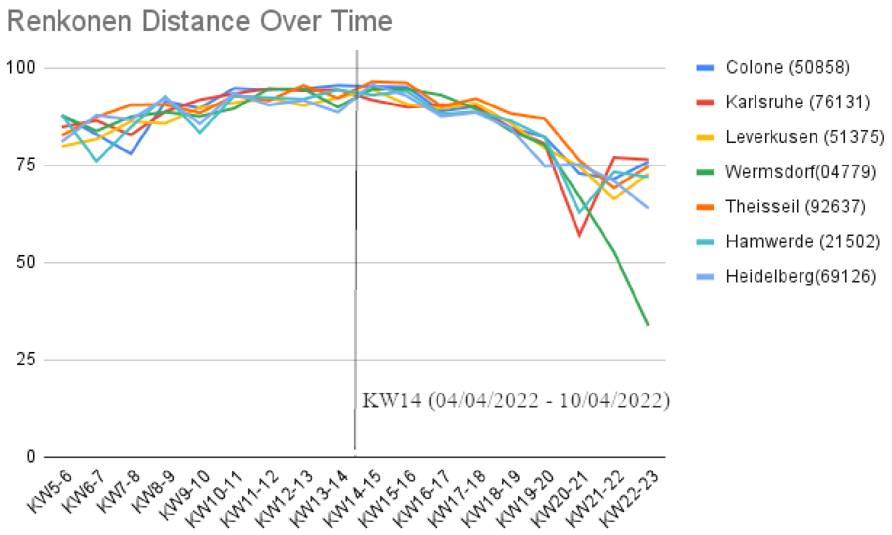


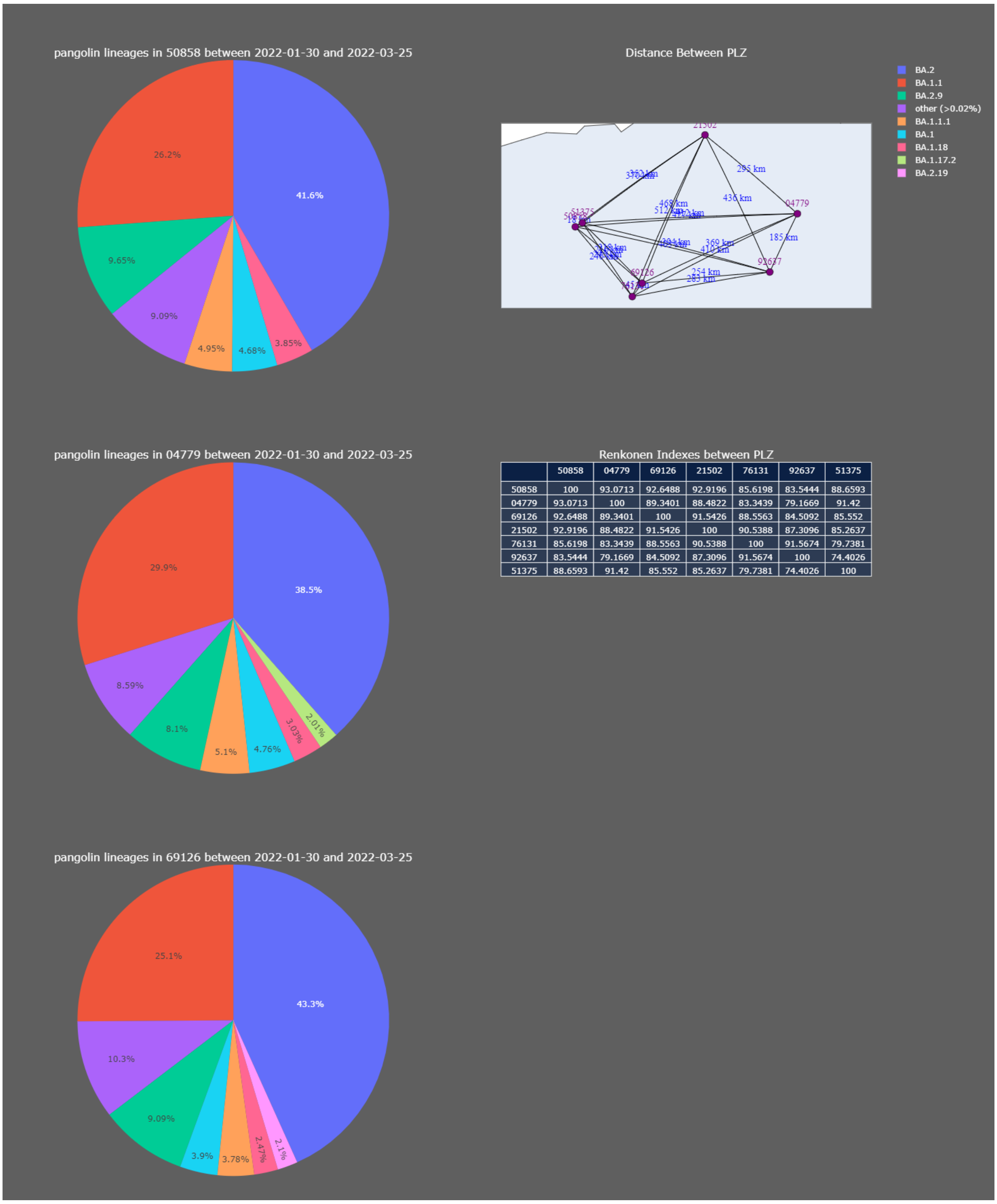
Figure 4.1: Renkonen Distance Over Time As Generated From First Use Case

The metric the application uses by default is Renkonen Index and Sørensen-Dice Coefficient as defined in Renkonen Similarity Index and Sørensen–Dice coefficient. It should be noted however, that the implementation of the summary function enables the replacement of this method with any function that accepts the same inputs and produces a numerical output.

As seen in 4.1 the Renkonen Indexes of the observed postal codes increase consistently towards their peak around week 14 after which they rapidly descend. Week 14 is the calendar week that contains the date of interest.

Thus the observation can be made that within a month period around the Coronavirus-Schutzverordnung being loosened there was a drastic decrease in the rate of change up until approximately a week before and after it was loosened at which point the rate of change rapidly increased, peaking around 8 weeks later.

With a visible trend in the data found, it should be determined what change this trend indicates. This is achieved using the second functionality of the summary interface, “Pango Makeup” which provides a pie chart of the relative proportion of each lineage in a given group of postal codes within a given date range. As the date range being assessed is the 30/1/2022 to the 20/6/2022 and the date of interest that acts as the middle point for our comparisons is 3/4/2022, the date ranges being compared will be the date range split by the middle point with a buffer of approximately 2 weeks on both ends. These ranges are 30/1/2022 to 25/3/2022 and 28/4/2022 to 21/6/2022. The resulting output of the date range 30/1/2022 - 25/3/2022 is shown in 4.3.



Renkonen Indexes between PLZ

| | 50858 | 04779 | 69126 | 21502 | 76131 | 92637 | 51375 |
|---|---|---|---|---|---|---|---|
| 50858 | 100 | 93.0713 | 92.6488 | 92.9196 | 85.6198 | 83.5444 | 88.6593 |
| 04779 | 93.0713 | 100 | 89.3401 | 88.4822 | 83.3439 | 79.1669 | 91.42 |
| 69126 | 92.6488 | 89.3401 | 100 | 91.5426 | 88.5563 | 84.5092 | 85.552 |
| 21502 | 92.9196 | 88.4822 | 91.5426 | 100 | 90.5388 | 87.3096 | 85.2637 |
| 76131 | 85.6198 | 83.3439 | 88.5563 | 90.5388 | 100 | 91.5674 | 79.7381 |
| 92637 | 83.5444 | 79.1669 | 84.5092 | 87.3096 | 91.5674 | 100 | 74.4026 |
| 51375 | 88.6593 | 91.42 | 85.552 | 85.2637 | 79.7381 | 74.4026 | 100 |

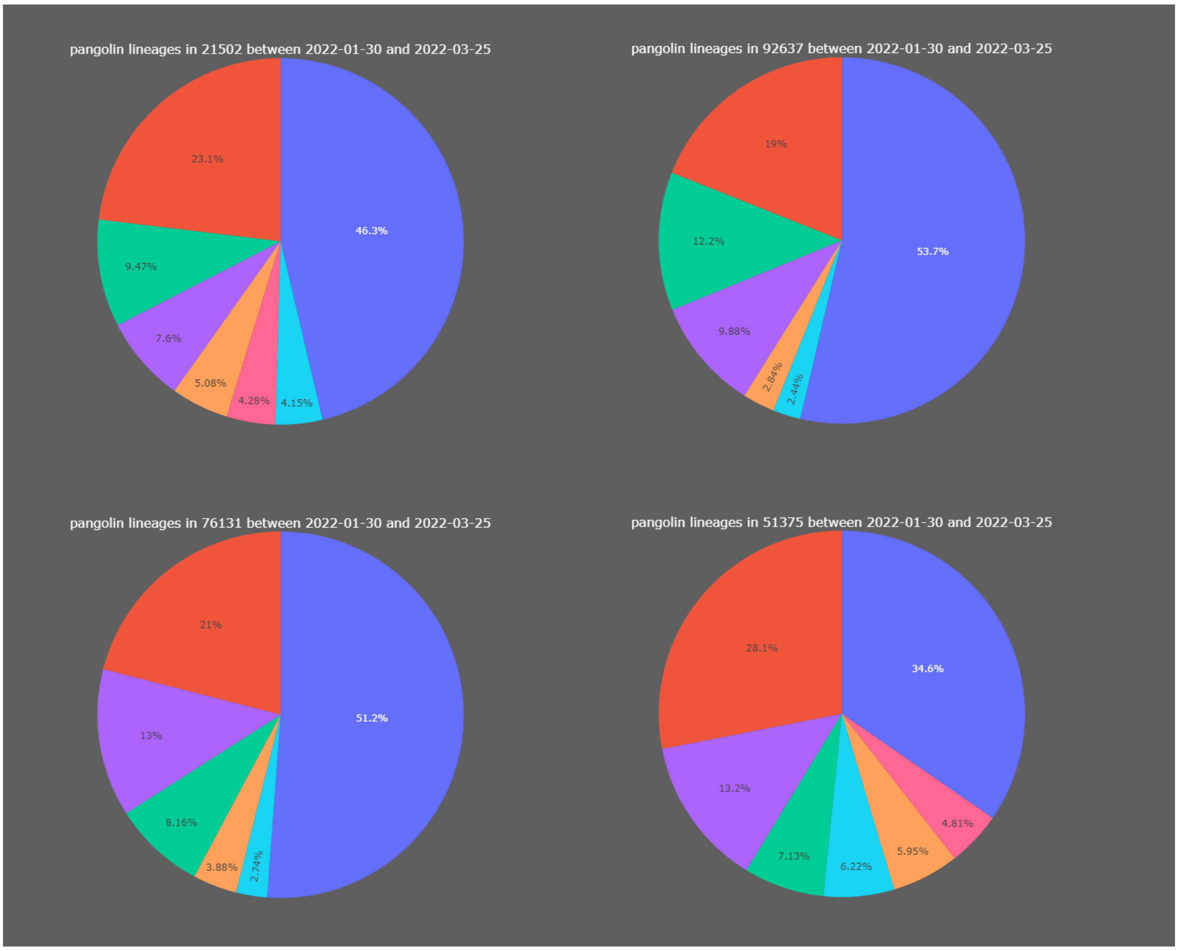
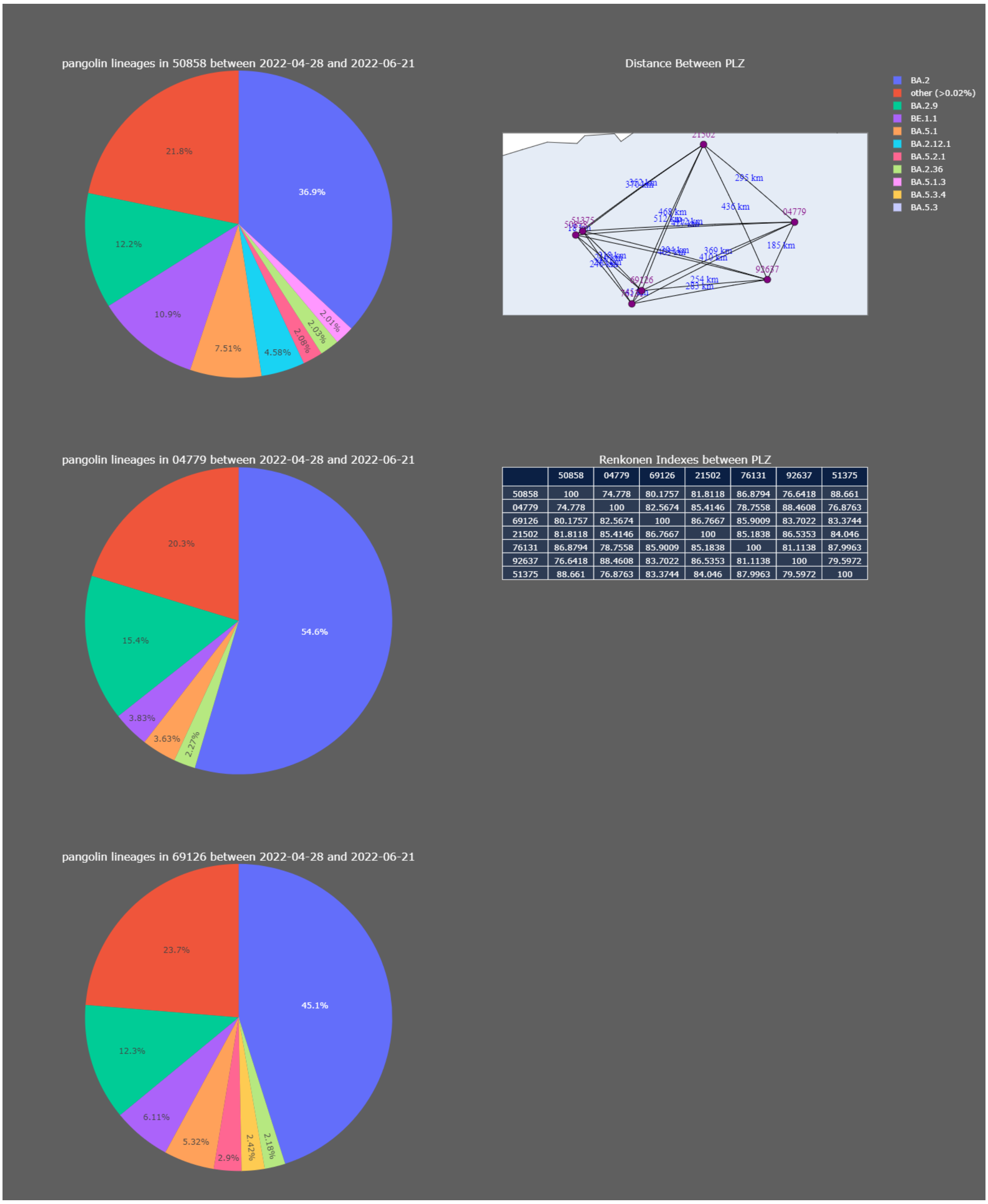


Figure 4.3: Pango Lineage Makeup of selected postal codes between 30/1/2022 - 25/3/2022

It can be observed that the most present lineage in every postal code was BA.2 with the second most present consistently being BA 1.1.

Assessing the time frame of 28/4/2022 - 21/6/2022 provides the figures 4.5.



pangolin lineages in 50858 between 2022-04-28 and 2022-06-21

Distance Between PLZ

pangolin lineages in 04779 between 2022-04-28 and 2022-06-21

Renkonen Indexes between PLZ

| | 50858 | 04779 | 69126 | 21502 | 76131 | 92637 | 51375 |
|---|---|---|---|---|---|---|---|
| 50858 | 100 | 74.778 | 80.1757 | 81.8118 | 86.8794 | 76.6418 | 88.661 |
| 04779 | 74.778 | 100 | 82.5674 | 85.4146 | 78.7558 | 88.4608 | 76.8763 |
| 69126 | 80.1757 | 82.5674 | 100 | 86.7667 | 85.9009 | 83.7022 | 83.3744 |
| 21502 | 81.8118 | 85.4146 | 86.7667 | 100 | 85.1838 | 86.5353 | 84.046 |
| 76131 | 86.8794 | 78.7558 | 85.9009 | 85.1838 | 100 | 81.1138 | 87.9963 |
| 92637 | 76.6418 | 88.4608 | 83.7022 | 86.5353 | 81.1138 | 100 | 79.5972 |
| 51375 | 88.661 | 76.8763 | 83.3744 | 84.046 | 87.9963 | 79.5972 | 100 |

pangolin lineages in 69126 between 2022-04-28 and 2022-06-21

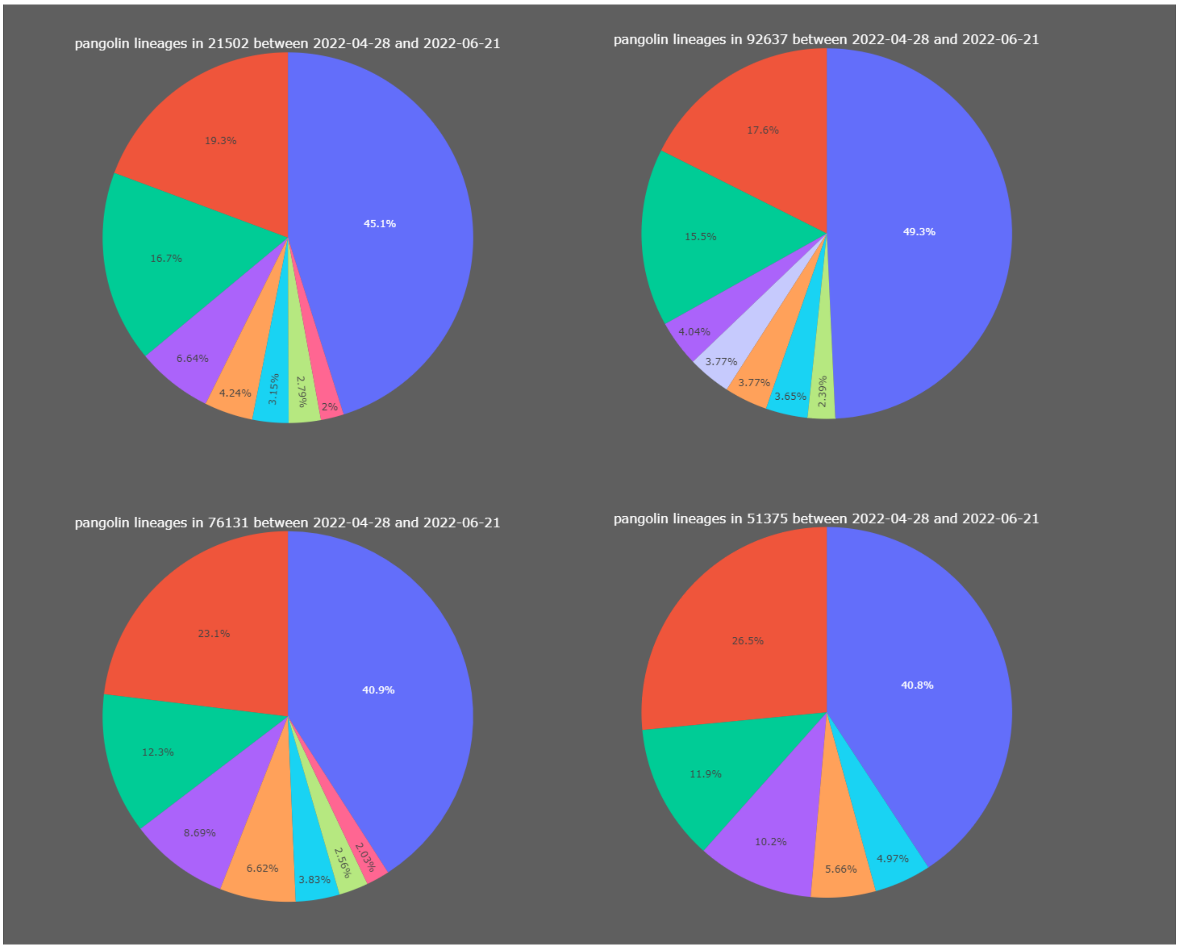


Figure 4.5: Pango Lineage Makeup of selected postal codes between 28/4/2022 - 21/6/2022

In these it is visible that BA 2 is still the most dominant lineage. However, BA 1.1 has been almost entirely removed from the population either not being present at all or making up less than 2% of the data set.

With this analysis complete results have been produced that can be further used and interpreted to answer if a shift in COVID policy can have an influence on the distribution of Pango lineages in a specific area.

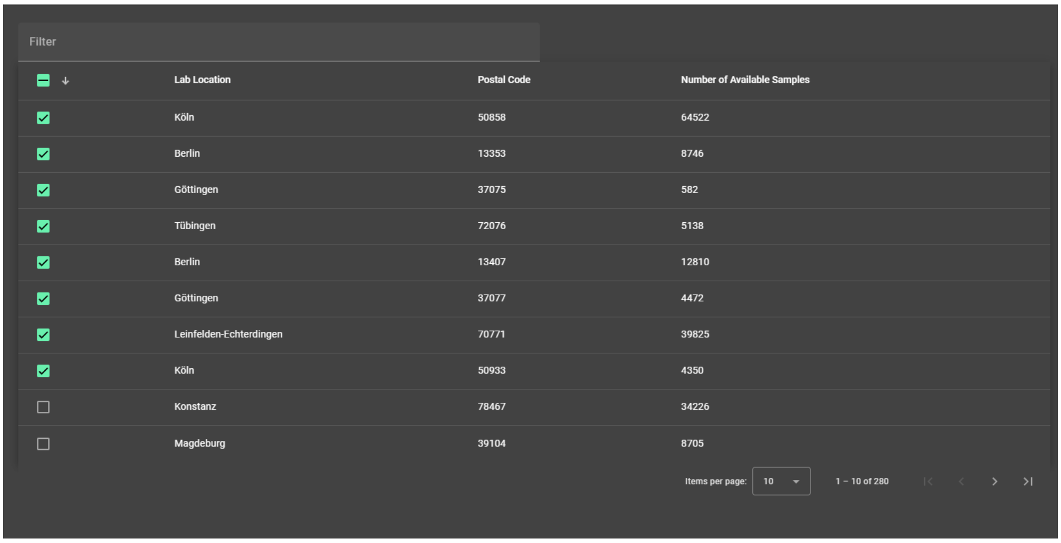
### Correlating Geographic and Genetic Distance

A second use the application could have is to determine if there is a correlation between geographical distance and genetic distance within sequences in the same lineage. To achieve this the application has the ability to show physical distance between selected postal codes as well as the ability to perform alignment operations on sequences.

The steps involved in performing this analysis are as follows:

1. Determine which geographic locations are suitable candidates to be used in this comparison. The geographic distance between the candidates as well as the number of samples each candidate has are deciding factors.
2. Determine the most present lineage within the selected postal codes and if a time frame exists where they all have high incidences of the same lineage. If an acceptable date range is found, select an equal amount of sequences from each postal code.
3. Perform sequencing operations between all of the selected sequences. Use the mean alignment score to determine if there is a significant correlation between the geographic and genetic distances.

As stated above the first step is to select candidate postal codes. The map selection interface displays how many available samples there are in a given postal code and provides a visual frame of reference for how close two postal codes are. The methodology in this example is to select the largest present sample size from the table, causing the map to jump to that location and selecting nearby postal codes, as shown in 4.6.



| ▬ ↓ | Lab Location | Postal Code | Number of Available Samples |
|---|---|---|---|
| ☑ | Köln | 50858 | 64522 |
| ☑ | Berlin | 13353 | 8746 |
| ☑ | Göttingen | 37075 | 582 |
| ☑ | Tübingen | 72076 | 5138 |
| ☑ | Berlin | 13407 | 12810 |
| ☑ | Göttingen | 37077 | 4472 |
| ☑ | Leinfelden-Echterdingen | 70771 | 39825 |
| ☑ | Köln | 50933 | 4350 |
| □ | Konstanz | 78467 | 34226 |
| □ | Magdeburg | 39104 | 8705 |

Items per page: 10 1 – 10 of 280

Figure 4.6: Selected Postal Codes As Displayed In Application

In the second step the Pangolin by postal code operation is used to get 4.7.

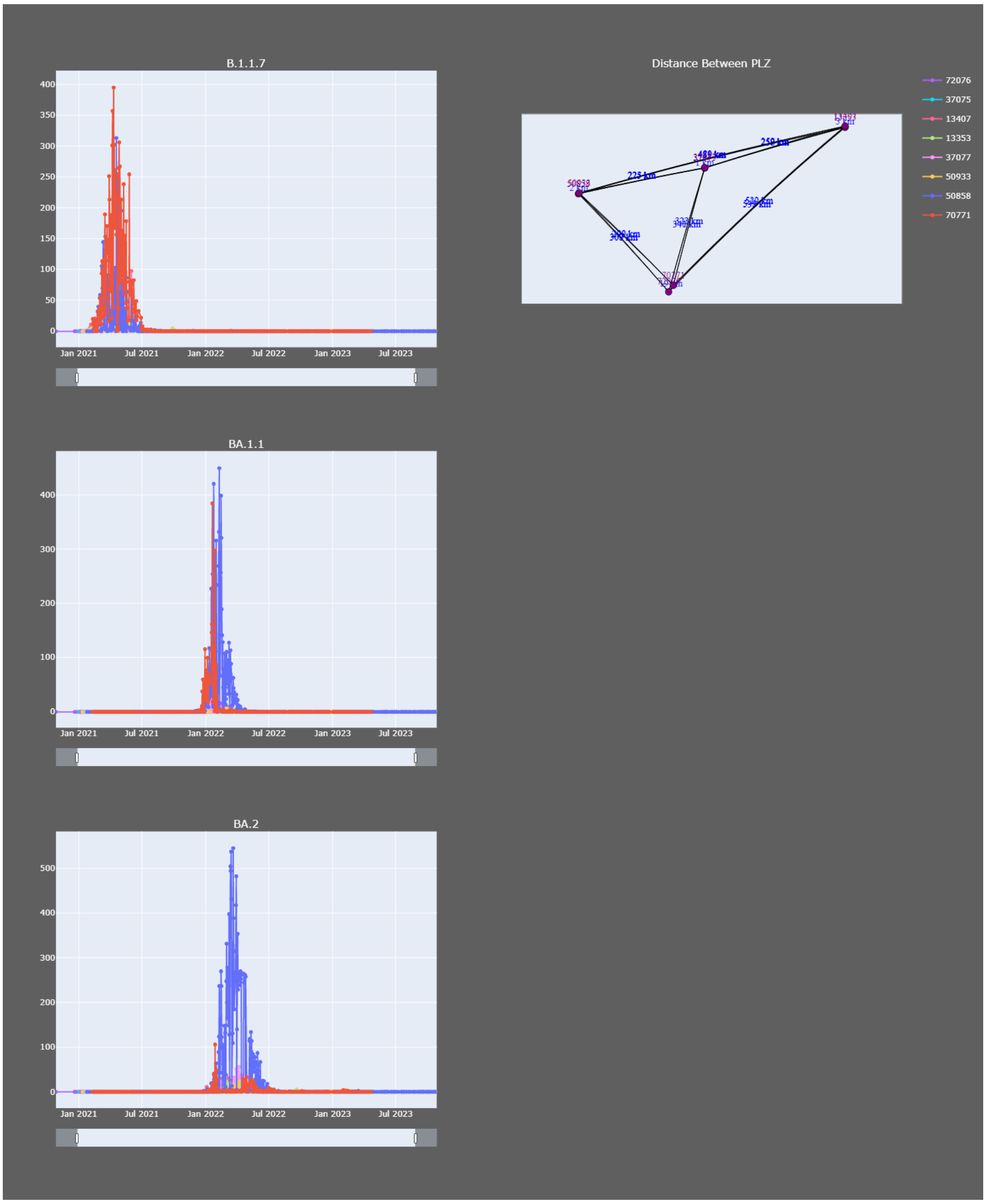


Figure 4.7: result of Pangolin by postal code operation on selected data

It can be observed that the pago lineage B.1.1.7 has at least five samples between the dates `17/3/2021` and `23/3/2021`. This sample count was chosen to keep the time it takes to perform the operation within acceptable bounds on the hardware used during this project. To minimize the factors that could influence genetic distance these samples were kept as closely aligned in time as possible. Furthermore, the results of Pangolin by postal code operation also provides the geographical distances between the selected postal codes.

In the last step the sequence alignment is performed on the selected samples, using a pairwise schema every sequence is aligned with each other. The alignment is performed using a Match Score of 1 and a gap score of $-1$. As seen in 4.1 these results are compiled into a mean alignment score.

Table 4.1: Mean Alignment Scores Between Selected Samples and Postal Codes

| | **13353** | **50858** | **37075** | **72076** | **13407** | **37077** | **70771** | **50933** |
|---|---|---|---|---|---|---|---|---|
| **13353** | 29784.50 | 29636.48 | 28042.89 | 29538.45 | 29707.80 | 29631.52 | 29356.00 | 28370.24 |
| **50858** | X | 29841.00 | 28234.53 | 29660.55 | 29574.36 | 29825.8 | 29545.12 | 28290.28 |
| **37075** | X | X | 28969.69 | 28012.31 | 27918.80 | 28291.24 | 28074.32 | 26890.80 |
| **72076** | X | X | X | 29454.25 | 29453.78 | 29566.16 | 29302.48 | 28150.92 |
| **13407** | X | X | X | X | 29664.30 | 29567.32 | 29292.08 | 28375.56 |
| **37077** | X | X | X | X | X | 29838.40 | 29557.40 | 28276.72 |
| **70771** | X | X | X | X | X | X | 29272.50 | 28022.40 |
| **50933** | X | X | X | X | X | X | X | 28604.70 |

Determining the correlation between the mean alignment and geographical distance requires the geographical distance between the postal codes. These are extracted into 4.7 from the results of step 2:

Table 4.2: Geographical Distance Between Postal Codes(km)

| | **13353** | **50858** | **37075** | **72076** | **13407** | **37077** | **70771** | **50933** |
|---|---|---|---|---|---|---|---|---|
| **13353** | 0 | 481.74 | 258.29 | 538.73 | 3.46 | 257.88 | 519.48 | 479.63 |
| **50858** | X | 0 | 225.20 | 308.85 | 483.00 | 225.44 | 298.32 | 2.472 |
| **37075** | X | X | 0 | 340.96 | 259.82 | 1.06 | 322.30 | 223.22 |
| **72076** | X | X | X | 0 | 541.60 | 341.99 | 19.40 | 309.88 |
| **13407** | X | X | X | X | 0 | 259.39 | 522.34 | 480.89 |
| **37077** | X | X | X | X | X | 0 | 323.33 | 223.46 |
| **70771** | X | X | X | X | X | X | 0 | 299.22 |
| **50933** | X | X | X | X | X | X | X | 0 |

To determine the potential correlation between geographical distance and alignment score the Pearson Correlation Coefficient and Spearman Rank Correlation Coefficient are calculated. The Pearson Correlation Coefficient is a value that spans from $-1$ to 1 that indicates the strength of a potential linear correlation between two continuous variables [29]. The Spearman Rank Correlation Coefficient is a similar value that indicates how likely two sets of values are related to each other monotonically [30]. Both of these values are calculated in addition to the null-hypothesis of the samples being uncorrelated and normally distributed [31] [32]. These values were calculated externally through Scipy using data downloaded from the application.

Using the values in 4.1 and 4.2 results in a Pearson Correlation Coefficient of $-0.0636$ and a null-hypothesis value of 0.7122. The resulting Spearman Rank Correlation is $-0.1466$ and a null hypothesis value of 0.3935. The resulting null hypothesis values indicate that there is no correlation between the values. Furthermore even if the null hypothesis values were in a range that indicates causation, the correlation coefficients are small enough that the correlation would be minimal. The application has been used to collect data to show that there is no correlation between the genetic and geographical distances of the selected sequences.

### 4.2.2 Comparison With Contemporaries

**Wellcome Sanger Tool**

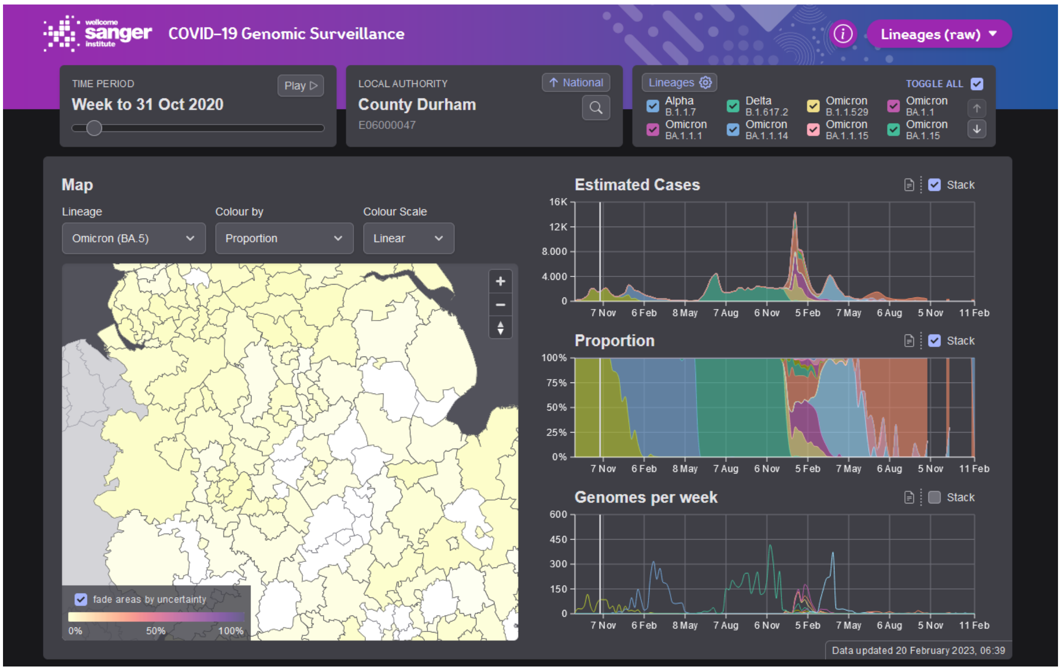
The first tool that will be compared to the application is the visualization tool provided by the Wellcome Sanger Institute. They are a nonprofit British genetic research Institute. During the pandemic they provided a tool to give an overview of Coronavirus data sequenced in Britain. It provides a geographical overview of Pango linage based visualisations.This tool sourced its data from the COVID-19 Genomics UK Consortium [7].

There is a lot of feature overlap between the Sanger Institute visualization tool and the application. However the Sanger Institute tool is targeted at providing an overall perspective of the data rather than targeted analysis. The data can only be observed on a week to week basis and by default it only shows highly present lineages, requiring interaction with a drop down menu to be able to see more than the highest tier of Pango lineage. The tool does give the option to group geographical data, allowing selection on a local district (the equivalent of a Landkreis in germany) and country wide level. The visualizations that the tool provides are line charts containing estimated case numbers, proportionality of lineage in the population and genomes per week. Each of these statistics can be displayed as a heatmap on the map portion of the application. And can be individually downloaded as a CSV file.

Unlike the Sanger tool, the application in this paper cannot provide estimated case numbers, as it is not configured to display them, nor has the means to calculate them with the provided data. Furthermore, the application can not display map based heatmaps.

The Sanger Institute tool offers no option to analyze individual samples of the dataset. And can only display up to 20 lineages at a time. Furthermore, the Sanger Institute tool has a predetermined curated data set that cannot be changed by the end user. Additionally outside of the heatmap the Sanger tool exclusively displays its data using line charts, which can lead to difficult to parse graphs compared to the pie charts the application generates. It cannot perform sequence alignment operations.

The tool provided by the Sanger Institute can be used for the use case Analysis of Policy Shifts with some limitations. While it can show the proportionality of lineages It can only show these proportionalities for a maximum of 20 lineages simultaneously and provides no way to compare the entire population of lineages at a single specific location. Despite this limitation it can still display the information required for the analysis in the use case, just with a higher hierarchical level of Pango linage. Another issue is that in this application all data is communicated through line graphs, as shown in 4.8.



TIME PERIOD Week to 31 Oct 2020

LOCAL AUTHORITY County Durham E06000047

Lineage: Omicron (BA.5)

Colour by: Proportion

Colour Scale: Linear

☑ fade areas by uncertainty

Data updated 20 February 2023, 06:39

Figure 4.8: The Sanger Institute Visualisation Tool [7]

This leads to charts that may be more difficult to read than the pie charts the application would generate for the same use. Furthermore, the Sanger Institute tool does not accept manual date inputs, only being iterable on a weekly basis through a slider. This makes finding a specific date range unintuitive.

The second use case, Correlating Geographic and Genetic Distance is not possible with this application as it offers no sequencing capability. It also offers no way to download the sequencing data that it uses for its visualizations, as the source of its data has been decommissioned since march of 2023 [10]

### Nextstrain Tool

Nextstrain is a visualization framework for genomic data.The Nextstrain Ncov Visualisation tool (titled Genomic epidemiology of SARS-CoV-2 with subsampling focused globally over the past 6 months) is a visualization tool built by the developers of Nextstrain using their ncov framework. It provides the phylogenetic trees, geographical distribution and lineage diversity of COVID-19 on a global scale from various data sources. [5]

The Nextstrain tool is more suited for phylogenetic tree analysis on a wide scale than for localized data analysis like the tool presented in this paper. It visualizes the phylogenetic tree of recorded samples, the nucleotide diversity of the genome as well as the frequencies of lineages over time. However it has no selection mechanism for individual data outside of phylogenetic tree analysis meaning every operation is performed on the entirety of the dataset that is selected. The lowest level of filtration for the dataset provided is on a continental level.

The Nextstrain application also contains a map based visualization of the lineage makeup of an area, shown on the map as a pie chart.

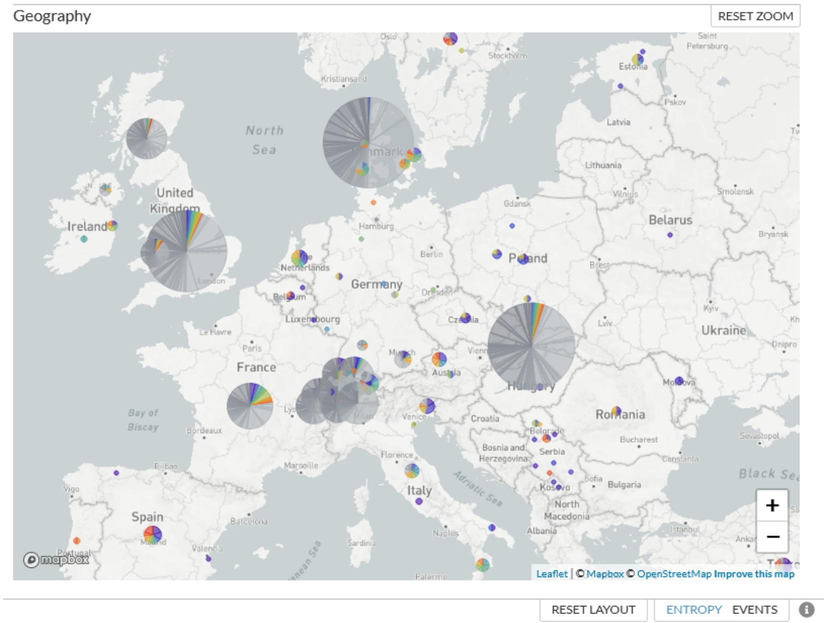


Figure 4.9: The Geography Based Lineage Analysis provided by Nextstrain ncov [5]

The area this visualization covers is limited, as the smallest unit of area it covers is “district” which appears to be a provincial level. The data points on the map are non intractable and cannot be modified to group a minimum sample size, leading to illegible graphs. Furthermore, they do not increase with size as the user zooms into the map, as displayed in the figure below where the map is zoomed in to its maximum possible zoom level in the attempt to make a chart more visible.

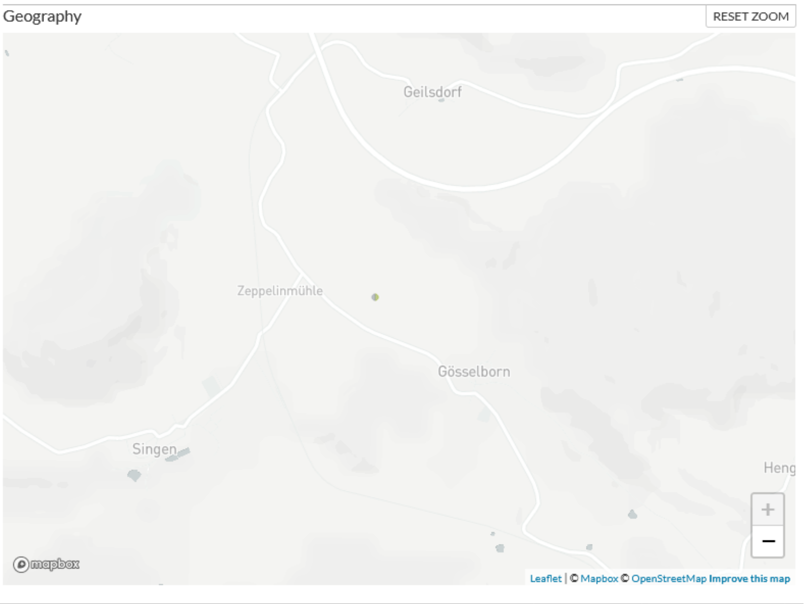


Figure 4.10: The Geography Based Lineage Analysis provided by Nextstrain ncov, zoomed in as far as the map will allow [5]

Nextstrain does have a featureset that the application does not. The phylogenetic visualization tools presented by the interface as well as the ability to group samples by several metrics for visualization, not just Pango lineage. Are features that the application does not have in any capacity.

The data that nexstrain sources for germany is from the Robert-Koch Institute so if it was possible to filter it down to Germany this would be the exact same data set being used in the application presented in this paper [5].

As a matter of disclosure it should be mentioned that late into the creation of this paper it was found that Nextstrain is open source and feasibly could be modified to include features that would allow it to fulfill the use case. This was not analyzed in any capacity as it was discovered too late to be integrated into the paper.

In the case of the Nextstrain application the use case Analysis of Policy Shifts is not possible primarily because of how large the smallest unit of filtering is in the dataset, analyzing Pango lineages across such a large geographical distance does not yield data that can be used for this analysis.

The Correlating Geographic and Genetic Distance is also not possible due to lack of sequencing functionality. However the tool provides the sequencing dataset it uses to determine lineages, in this way it can still be used to fulfill the second use case indirectly as it provides the relevant data required but no way to align it directly.

# Chapter 5

# Conclusion

The goal of this paper was to provide a locally runnable framework that can be used to analyze genomic data through geographical selection methods.

The application was produced with a Python backend and an Angular frontend. The web interface provides a geographical based selection interface through a map and provides an alternative selection mechanism through a table based interface. The table interface and figure interface both facilitate custom positioning through the ability to pop them into separate browser tabs. This was achieved by using the local storage of the browser, except the selection table for individual sequences which has to use the backend due to the limitations of the JSON parser of Javascript.

The backend of the application handles various file processing and visualization tasks. The application can run on systems with lower memory and CPU specifications due to the usage of memory mapping in retrieving the sequencing data. The application can be easily changed to accept formats other than FASTA for sequencing data as the function to parse the sequencing data is user defined. Generating figures was designed for easy expandability, with adding a new figure type only requiring adding code rather than modifying it. The application can generate six unique figures, four of which were required in the example use cases.

Those use cases included if a specific event changed the Pango lineage make up of an area as well as determining if genetic and geographical distance of sequences within a Pango lineage are correlated. To provide this analysis the application can use the renkonen-index over time function, the proportional pangolin lineage function and the sequence alignment function. The application was compared to tools with similar functionality in the context of these use cases with mixed results. While the other tools did not have alignment functionality they had different toolsets in the realm of phylogenetic tree analysis and data visualization directly on the map rather than just having it act as a selection pane.

The result of the paper is an application that can take the dataset of the Robert-Koch Institute and provide the user with fairly accurate visualization and analysis tools. The paper achieved its goal with some minor caveats. While the application has been designed to be used with a genomic data set it still requires external files to visualize selection surfaces on the map and these data sources can vary wildly in quality, the map that the application ships with is missing 6 postal codes that do exist but are simply not in the shapefile data. Due to an inherent limit on the size of the files that the javascript JSON parser can process within a reasonable timeframe, sequence mode selection tables had to be moved into the backend and handled through http requests. This shift caused several performance issues on the device the application was developed on. Sequence tables do not properly synchronize, lag and cause crashes when more than 30 samples are selected.

While the application achieved its stated goal, the comparison of it to its contemporaries highlighted that there are many potentially useful features that could be implemented due to its modular nature. Leaflet, the map tool used in the application has native support for heatmaps which could be a useful visualization mechanism. Furthermore, phylogenetic analysis is an interesting field that the tool has no capabilities in but would be relevant in tandem with the visualizations it can already provide. Implementing these features as well as optimizing and fixing the aforementioned caveats may be possible topics of research papers using this one as a basis.

# Bibliography

[1] "World health organization 2023 data.who.int, who coronavirus (covid-19) dashboard > cases [dashboard]". (2024), [Online]. Available: `https://data.who.int/dashboards/covid19/cases` (visited on 01/19/2024).

[2] "Gisaid.org". (2024), [Online]. Available: `https://gisaid.org/` (visited on 01/19/2024).

[3] "Sars-cov-2-sequenzdaten aus deutschland - robert-koch institut". (2024), [Online]. Available: `https://github.com/robert-koch-institut/SARS-CoV-2-Sequenzdaten_aus_Deutschland` (visited on 01/19/2024).

[4] E. T. Martínez Beltrán, M. Quiles Pérez, J. Pastor-Galindo, P. Nespoli, F. J. García Clemente, and F. Gómez Mármol, "Convida: Covid-19 multidisciplinary data collection and dashboard", *Journal of Biomedical Informatics*, vol. 117, p. 103 760, 2021, ISSN: 1532-0464. DOI: `https://doi.org/10.1016/j.jbi.2021.103760`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S1532046421000897`.

[5] "Genomic epidemiology of sars-cov-2 with subsampling focused globally over the past 6 months, nextstrain-ncov". (2024), [Online]. Available: `https://nextstrain.org/ncov/open/global/6m` (visited on 02/29/2024).

[6] A. Walker, T. Houwaart, P. Finzer, *et al.*, "Characterization of Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2) Infection Clusters Based on Integrated Genomic Surveillance, Outbreak Analysis and Contact Tracing in an Urban Setting", *Clinical Infectious Diseases*, vol. 74, no. 6, pp. 1039–1046, Jun. 2021, ISSN: 1058-4838. DOI: `10.1093/cid/ciab588`. eprint: `https://academic.oup.com/cid/article-pdf/74/6/1039/42992507/ciab588.pdf`. [Online]. Available: `https://doi.org/10.1093/cid/ciab588`.

[7] "Sanger institute covid–19 genomic surveillance". (2023), [Online]. Available: `https://covid19.sanger.ac.uk/lineages/` (visited on 01/19/2024).

[8] "Cdc covid data tracker: Variant proportions". (2024), [Online]. Available: `https://covid.cdc.gov/covid-data-tracker/#variant-proportions` (visited on 01/19/2024).

[9] "Statement on the fifteenth meeting of the ihr (2005) emergency committee on the covid-19 pandemic". (2023), [Online]. Available: `https://www.who.int/news/item/05-05-2023-statement-on-the-fifteenth-meeting-of-the-international-health-regulations-(2005)-emergency-committee-regarding-the-coronavirus-disease-(covid-19)-pandemic` (visited on 02/23/2024).

[10] "Covid-19 uk genetics consortium". (2023), [Online]. Available: https://www.cogconsortium.uk/ (visited on 02/19/2024).

[11] "Pango network - what are pango lineages?" (2023), [Online]. Available: https://www.pango.network/how-does-the-system-work/what-are-pango-lineages/ (visited on 02/19/2024).

[12] "Pango network - rules for the designation and naming of pango lineages". (2023), [Online]. Available: https://www.pango.network/the-pango-nomenclature-system/statement-of-nomenclature-rules/ (visited on 02/19/2024).

[13] O. Renkonen, "Statistisch-ökologische untersuchungen über die terrestrische käferwelt der finnischen bruchmoore", *Societas zoologica-botanica Fennica Vanamo*, 1938.

[14] T. Sørensen, *A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and Its Application to Analyses of the Vegetation on Danish Commons* (Biologiske skrifter). Munksgaard in Komm., 1948. [Online]. Available: https://books.google.de/books?id=rpS8GAAACAAJ.

[15] P. M. Altschul SF, "Handbook of discrete and combinatorial mathematics. 2nd edition.", in CRC Press, 2017, ch. Chapter 20.1, Available at https://www.ncbi.nlm.nih.gov/books/NBK464187/.

[16] "Pairwise sequence alignment - biopython 1.84". (), [Online]. Available: https://biopython.org/docs/dev/Tutorial/chapter_pairwise.html.

[17] M. K. et al., *Computational Biology - Genomes, Networks, and Evolution.* Massachusetts Institute of Technology via MIT OpenCourseWare, 2021, [Online; accessed 2024-03-03. [Online]. Available: https://bio.libretexts.org/Bookshelves/Computational_Biology/Book%3A_Computational_Biology_-_Genomes_Networks_and_Evolution_(Kellis_et_al.)/02%3A_Sequence_Alignment_and_Dynamic_Programming/2.05%3A_The_Needleman-Wunsch_Algorithm.

[18] Wikipedia contributors, *Needleman–wunsch algorithm — Wikipedia, the free encyclopedia*, [Online; accessed 3-March-2024], 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Needleman%E2%80%93Wunsch_algorithm&oldid=1207872635.

[19] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids.* Cambridge University Press, 1998.

[20] R. C. Edgar, "MUSCLE: multiple sequence alignment with high accuracy and high throughput", *Nucleic Acids Research*, vol. 32, no. 5, pp. 1792–1797, Mar. 2004, ISSN: 0305-1048. DOI: 10.1093/nar/gkh340. eprint: https://academic.oup.com/nar/article-pdf/32/5/1792/7055030/gkh340.pdf. [Online]. Available: https://doi.org/10.1093/nar/gkh340.

[21] "Welcome to flask - flask documentation (3.0x)". (2024), [Online]. Available: https://flask.palletsprojects.com/en/3.0.x/ (visited on 02/19/2024).

[22] "Angular -what is angular?" (2023), [Online]. Available: https://angular.io/guide/what-is-angular (visited on 02/15/2024).

[23] "Angular -two-way binding". (2023), [Online]. Available: https://angular.io/guide/two-way-binding (visited on 02/15/2024).

[24] "Angular - how event binding works". (2023), [Online]. Available: https://angular.io/guide/event-binding-concepts (visited on 02/15/2024).

[25] 2021. [Online]. Available: https://blast.ncbi.nlm.nih.gov/doc/blast-topics/#query-input-and-database-selection.

[26] M. Kerrisk, *Mmap(2) — linux manual page*, 2023. [Online]. Available: https://man7.org/linux/man-pages/man2/mmap.2.html#NOTES.

[27] "Pymsaviz". (2023), [Online]. Available: https://moshi4.github.io/pyMSAviz/ (visited on 02/15/2024).

[28] "Viele corona-schutzmaßnahmen fallen weg", *DW - Deutsche Welle*, Apr. 3, 2022. [Online]. Available: https://www.dw.com/de/viele-corona-schutzma%C3%9Fnahmen-fallen-weg/a-61343763 (visited on 02/12/2024).

[29] K. S. U. Libraries. "Spss tutorials: Pearson correlation". (Dec. 18, 2023), [Online]. Available: https://libguides.library.kent.edu/SPSS/PearsonCorr.

[30] C. Spearman, "The proof and measurement of association between two things", *The American Journal of Psychology*, vol. 15, no. 1, pp. 72–101, 1904, ISSN: 00029556. [Online]. Available: http://www.jstor.org/stable/1412159 (visited on 03/03/2024).

[31] 2024. [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.pearsonr.html.

[32] 2024. [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.spearmanr.html.

# List of Figures

# List of Tables