Project Title

COVID-19 Interactive WebMap

Team Members

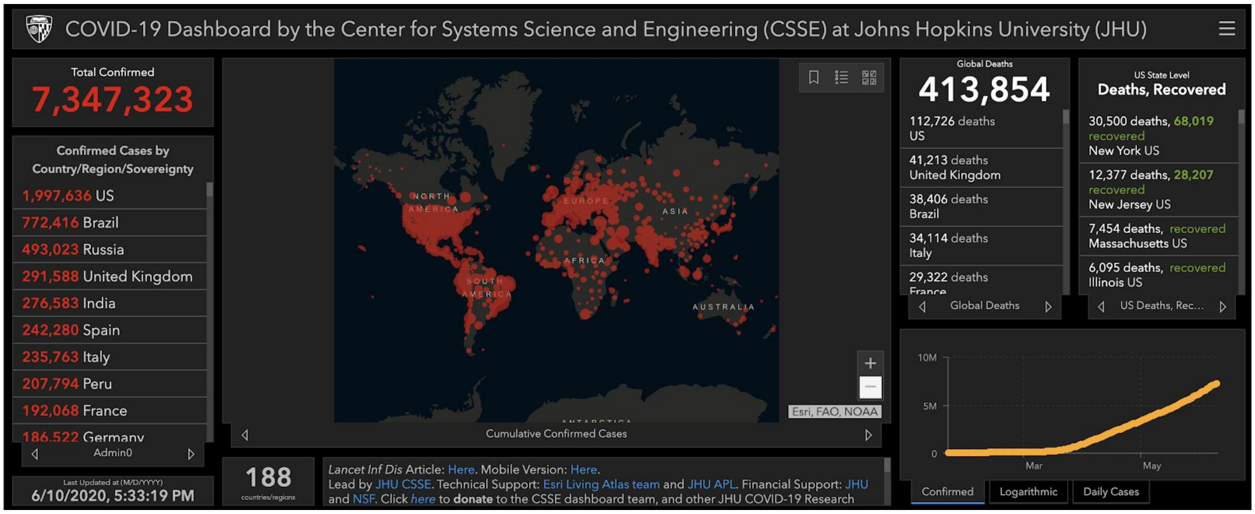
Karanveer Singh
Kleanthis Karavangelas
Liam Shamir
Dan Dunkers
Daniel Drzewicki

Abstract

In this report, the WebMap presented was created to interactively display the data based on the work of Zhao et al. explained in the paper “Characterizing geographical and temporal dynamics of novel coronavirus SARS-CoV-2 using informative subtype markers” [1]. Zhao’s paper introduces a new technique for visualizing the geographic and temporal dynamics of the spread of SARS-CoV-2, responsible for the Covid-19 pandemic [2]. This new framework for genetic subtyping referred to as ISM (Informative Subtype Marker) allows for individual genome subtyping, generating and assigning a signature to each genome. This method makes tracking viral evolution easier over geography and time. The website created displays a colored world map, and countries of the same color have the same most abundant SARS-CoV-2 genome present. The web interface is also interactive, allowing users to click on a country and view relative abundance graphs and pie charts of the ISM data. The web development stack used for this project includes MongoDB for the database creation, React from JavaScript for building the user interface, Node.js for the back-end server and finally GraphQL for middleware. The ultimate goal of this work is to provide a user-friendly platform for research to be better accessible and understood by the scientific community as well as the general population.

Literature Review

Efforts have been made in the past to track the spread of the Sars-CoV 2 virus. Perhaps the most well known web interface for tracking the spread of this virus has been developed by Dong et al. and documented in the paper “An interactive web-based dashboard to track COVID-19 in real time” [3]. This web application is hosted by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University with the goal of visualizing and tracking reported cases of the virus. Data is gathered from various sources including Twitter, local news, and the online community run platform DXY. They display a world map and data from each country and city-level data for the United States. The dashboard includes a clickable map that will give information about that region as well as various lists detailing the total number of deaths and cases in a particular country. Figure 1 shows how the web dashboard looks.



COVID-19 Dashboard by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University (JHU)

Total Confirmed
7,347,323

Confirmed Cases by Country/Region/Sovereignty
1,997,636 US
772,416 Brazil
493,023 Russia
291,588 United Kingdom
276,583 India
242,280 Spain
235,763 Italy
207,794 Peru
192,068 France
186,522 Germany

Last Updated at (M/D/YYYY)
6/10/2020, 5:33:19 PM

188
countries/regions

Lancet Inf Dis Article: Here. Mobile Version: Here.
Lead by JHU CSSE. Technical Support: Esri Living Atlas team and JHU APL. Financial Support: JHU and NSF. Click *here* to **donate** to the CSSE dashboard team, and other JHU COVID-19 Research

Global Deaths
413,854
112,726 deaths
US
41,213 deaths
United Kingdom
38,406 deaths
Brazil
34,114 deaths
Italy
29,322 deaths
France

US State Level
Deaths, Recovered
30,500 deaths, 68,019 recovered
New York US
12,377 deaths, 28,207 recovered
New Jersey US
7,454 deaths, recovered
Massachusetts US
6,095 deaths, recovered
Illinois US

Figure 1. JHU CSSE COVID-19 dashboard

While this application provides valuable information about the growth and number of cases throughout the world, it is lacking some important information and we wish to fill that gap with the results of our project. This web app lacks information about the different strains or genetic subtypes of the virus. In order to truly understand the spread of the virus, it is necessary to look at more than just the number of cases or deaths. You must look at the virus at the genomic level. For this reason, a similar application that displays the data about informative subtype markers would be important to visualize the geographic and temporal spread of the virus. This will show how different subtypes of the virus developed and spread across the world.

HealthMap is another widely used and available interactive map for tracking the spread of COVID-19 [4]. This web tracker is run by researchers, doctors, and engineers at Boston Children's Hospital. This tracker similarly gathers data from local news media, social media, and organizations such as the WHO. This website updates data almost real-time; however, it has the same lackings as the JHU dashboard in that only the total number of cases and deaths are tracked rather than looking deeper and tracking the virus on the genomic level.

The ISM's and pipeline used in this experiment are from the work of Zhao et al. [1]. Results are displayed throughout the paper as well as in the associated GitHub. The GitHub includes Jupyter notebooks that display various graphs about the ISMs in different regions. However, the display of this data can be improved from a Jupyter notebook to an interactive web application. The web app has several advantages. Firstly, it would be much more presentable and informative as users will be able to click on any country and view statistics for that country rather than being shown static, non-interactive images. Additionally, a web page would be much more accessible. It can be reached by anyone that has the URL to the page.

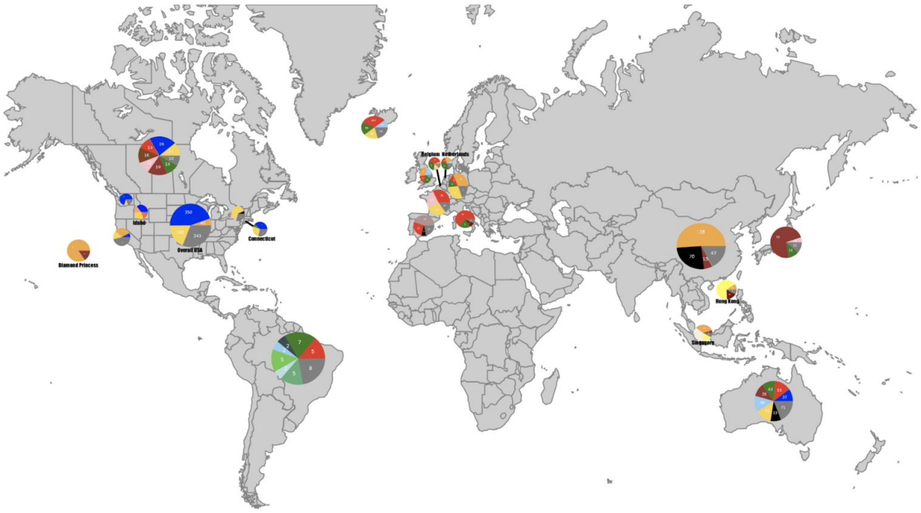


Figure 2. Current map visualization of the ISM data on the EEE/ISM GitHub

Methods

The front-end of the website was developed in Javascript using the React framework. There are several reasons why React was chosen to build the front-end. The React framework is based on components. These components can either be class based for functional components and can be built from other components. These components can have states that can cause the webpage to change accordingly thus leading to responsive and dynamic web pages. For example, states can be what the mouse is currently hovering over or what country on the map has been clicked on. This is a useful feature as we would like to display graphs in a dialog box every time a country is clicked on and display a tooltip over the mouse when the mouse is hovering over a country or part of a graph. To display the interactive map on the main page, the react simple maps package was used. React simple maps is a wrapper around the d3-geo package. This package can be used to generate a world map. The borders within this map are supplied as a GeoJSON file. Once this is done, listeners can be added for DOM events such as onclick or onmouseenter. These listeners will take as input the country that is being clicked or hovered over. Each geographic entity specified by the GeoJSON file can also be colored. This is useful as each country can be colored depending on the most abundant ISM. The data for each country was displayed using chart.js. The chart.js package allows for simple and flexible graphing of data. Data is supplied as an array of values and an array of labels. Each dataset can also include a value for color and other visual properties that can make the graphs more aesthetic as well as informative. Another great feature of this package is the interactivity. The graphs and charts can be hovered over, revealing information about the data. Additionally, data on the graphs can be removed and the graphs will automatically rescale to the remaining data. Components from the Material UI package were used in the front end. Material UI contains various react components such as dialog boxes and tool tips. Another important addition to the front-end was the color code (Figure 3). The top 49 ISMs are all mapped to unique colors so that whenever they are displayed in graphs and charts, they use those colors. As stated before, the color codes are also used to color countries on the map with the corresponding color of the most abundant ISM within that country.

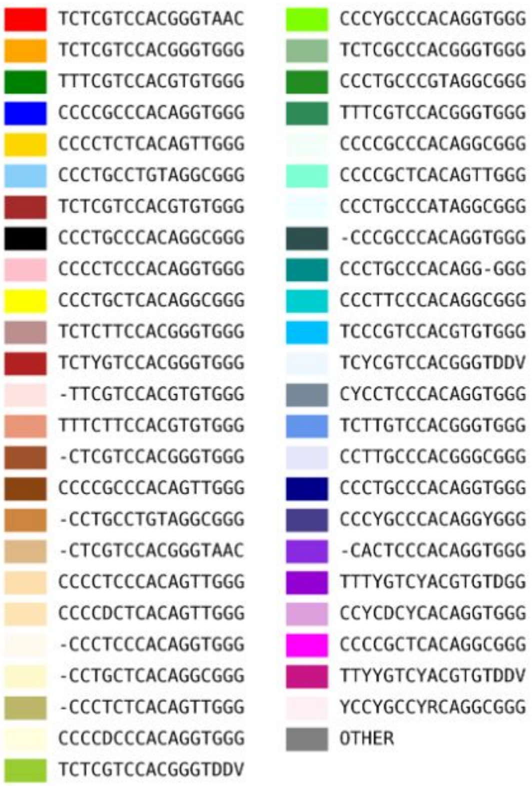


Figure 3. ISM color codes

In addition to the Javascript and React front-end, a back-end was written to handle the processing of the data. The backend consists of a MongoDB database and several Python scripts. Due to the nature of the data, using a noSQL database provided the best functionality for data importing as well as query. Data for this project is provided by GISAID and all of their generous contributors [5]. Once said GISAID data has been downloaded, a pipeline written by Zhengqiao Zhao is initiated to process the data into several csv and json files [6]. Once the pipeline has completed its filtering, MAFFT alignment, and ISM sequencing, the resulting json files are processed through two scripts. Each script is written to handle either inserting the region pie chart data or the region time series data. By passing the path to the respective json file, the python script provides index fields that are searchable in MongoDB and the data is inserted into their respective collection in the mongo db living on the localhost. Since each iteration of the pipeline provides all data, and not just new data, precautions were taken to ensure duplicate data is not imported into the database, thereby keeping the overall size of the database as small and efficient as possible. The database itself runs on the server that was provided by the ECE department on the default port 27017, running as a daemon so upon a server restart the mongod process will launch itself. The structure of the database is as follows: the db is called gisaid, and under gisaid there are two collections, regionPieChart and regionTimeSeries. The python scripts are configured to load into these locations but could be changed in the future. Additionally, the only prerequisite to run the scripts in python is installing pymongo.

To process data between and the front end, a middleware was created. The middleware server consists of a javascript based framework called ExpressJs. To streamline application programming calls (API) GraphQL was used as well. GraphQL allows the batching of multiple calls from the front end to the middleware. Another advantage of GraphQL is that the front-end dictates what information it needs from the middleware, reducing the amount of logic that is needed on the front end, and speeding up the interface. The middleware retrieves the country and graph information from the database and sends it to the front end.

Results

Figure 4 below shows the front page of the website created, which displays the interactive world map, or WebMap. As it can be seen, countries filled with the same color share a common most abundant genome of SARS-CoV-2. For example, countries colored in green, like the USA and Sweden, have the same most reported subtype sequence, or ISM. The next tab included in this website is the “Acknowledgments” tab, in which we acknowledge all the contributors to the GISAID and Nextstrain organizations, which share their sequence data and metadata, and have made this work possible. The last tab present in this webpage is an “Info” tab, which describes the purpose of the website, lists its creators and contributors, a brief introduction to ISMs, as well as provides links to the website GitHub and the ISM research GitHub.

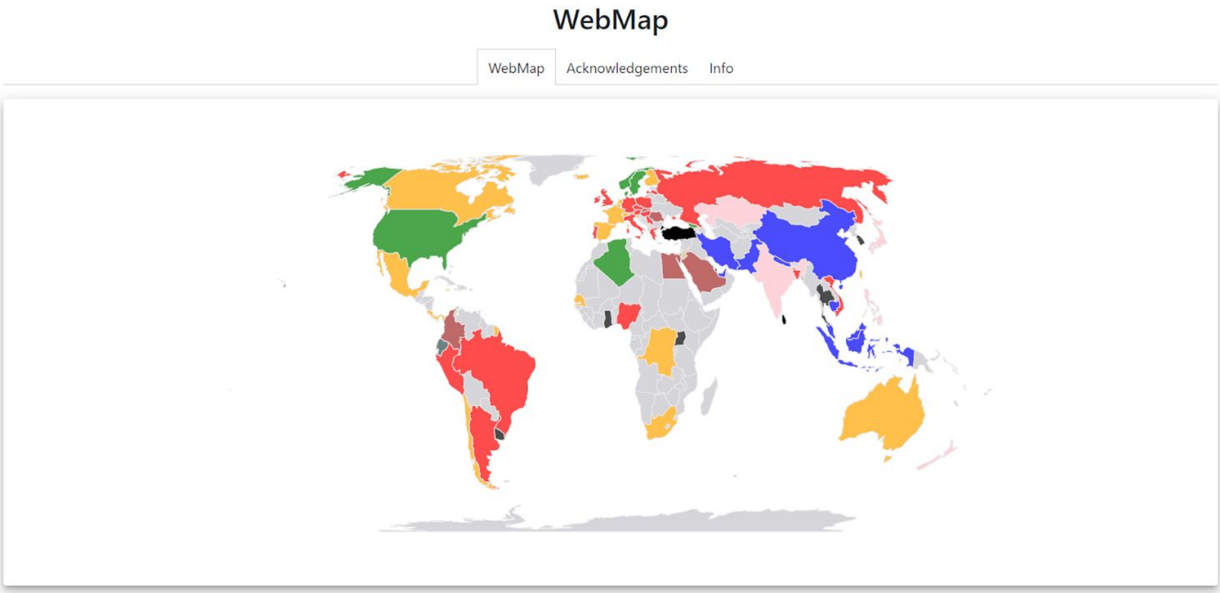


Figure 4. Interactive world map, WebMap

When clicking on a country, a new window pops up. Figure 5 shows the first half of the pop up window, which displays a relative abundance graph regarding the different genomes present in that specific country. The country chosen in this specific case for display was Italy. The different ISMs are color coded. The vertical axis represents a percentage, and the horizontal axis represents time.

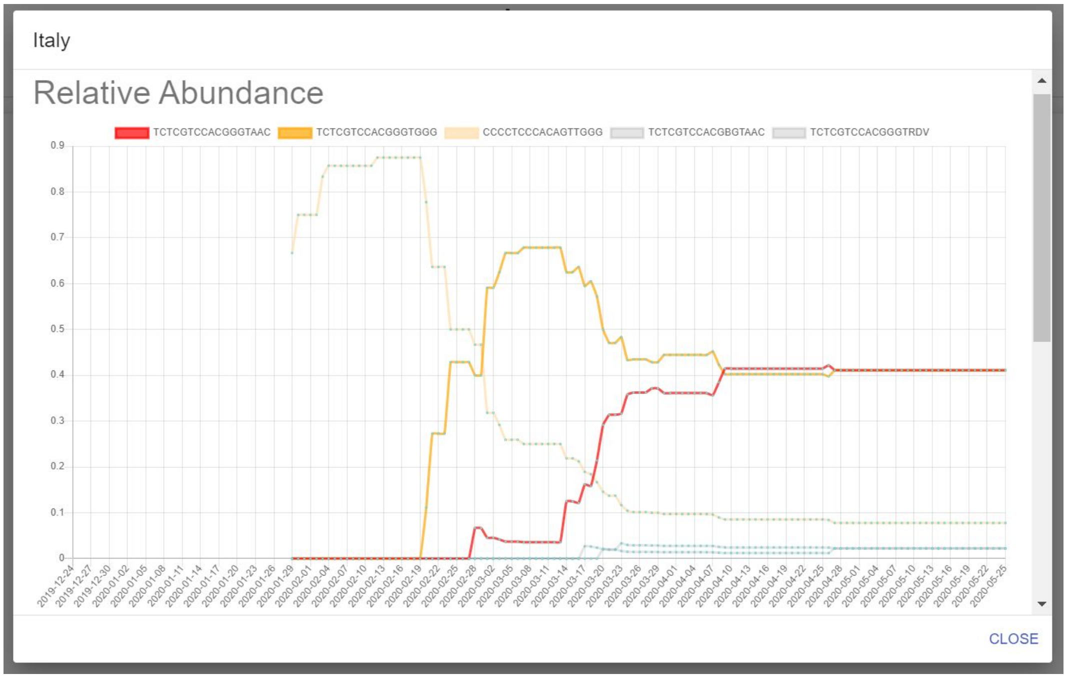


Figure 5. Relative abundance graph

The second half of the pop up window displays the most popular ISMs in a given country, in the form of a pie chart. The country chosen once again is Italy, and Figure 6 displays the country's ISM pie chart.

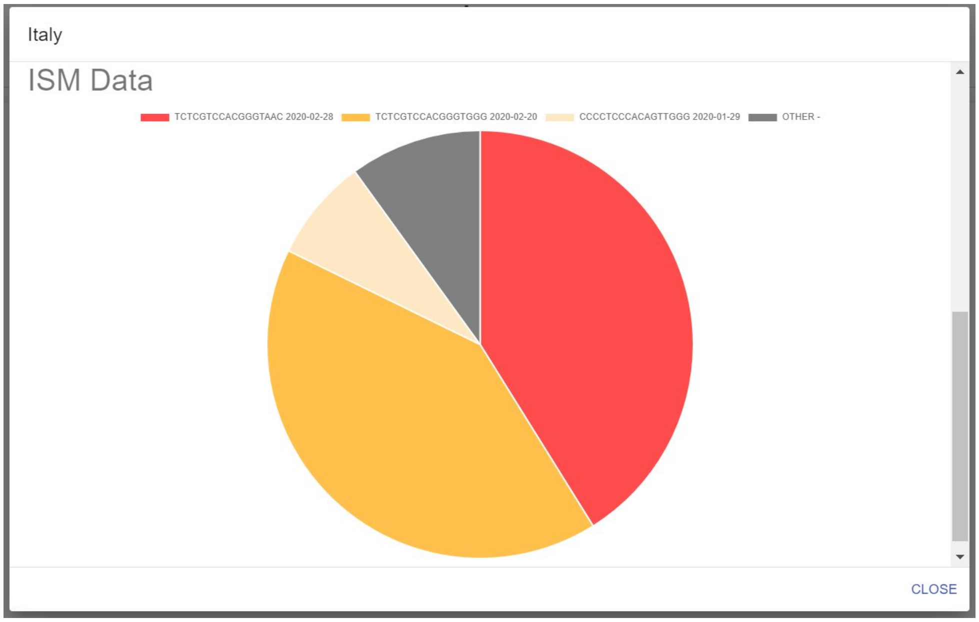


Figure 6. ISM pie chart

Discussion

Overall, the website created allows users to visualize a world map that indicates which countries share the same most common ISM. In addition, they are able to click on a country and view the relative abundance of ISMs in a graph or a pie chart. The primary goal was to make the ISM research as easy as possible to view and understand. Creating awareness about how the virus spreads geographically and through time is especially important during a pandemic.

Regarding future work, the first step that has to be completed is to connect the middleware to the front-end portion of the website. This will allow for processing data to be done on the server side instead of on the web browser, making the process faster. This is an important step in terms of scalability, since adding more data will significantly slow down the process if only done on the web browser. Presently, deployment of the entire website is rather cumbersome, requiring specific knowledge of Node.js in order to initialize the website. Another future improvement would be to have a script or a Docker image created in order to ease the deployment and scalability of the website. Once a location for deployment on the server is finalized and daemonized to run continuously in production, a DNS entry for the server could be requested for easy access to the website by users on the network. Additionally, the data for the website is provided by GISAID. Due to the nature of the data and GISAID's end user license agreement, automation of data retrieval from their system is not possible, requiring a user to manually go and retrieve the data from GISAID. Scripting could be provided to wrap the existing pipeline and json processing scripting to allow for said user to simply move the GISAID data to designated location and the pipeline will initiate as well as upload to the database, allowing for a more efficient processing of data and a more accurate representation of the data on the website.

Some concern could be made about the backend data itself scaling with the data over time. Since the region data is time series and will obviously grow over time, the three concerns are: server memory, insertion time, and the size of the database. The server memory at the time is unknown and loading in a large json script needs to be done when processing the data and loading into mongo. Therefore, this should be monitored in case more RAM needs to be added to the server. The data insertion time at the moment is less than a second, therefore this process at the moment should be able to handle a lot more data before a noticeable slowdown will occur. However, even if the script starts to take a longer to run, nothing else runs on the server to compete for memory, so letting the script run should not be an issue. Finally, the database size itself will increase over time. However, since the scripts are upserting the data, only new data will be inserted, rather than adding duplicate data from previous inserts for time periods of time that are already in the database.

References

1. Zhao, Zhengqiao; Sokhansanj, Bahrad A. & Rosen, Gail L. 2020. "Characterizing geographical and temporal dynamics of novel coronavirus SARS-CoV-2 using informative subtype markers". *Electrical and Computer Engineering, College of Engineering, Drexel University, Philadelphia, PA, USA*.

2. Li, Qun et al. 2020. "Early Transmission Dynamics in Wuhan, China, of Novel Coronavirus–Infected Pneumonia". *The New England Journal of Medicine*.

3. Dong et al. "An interactive web-based dashboard to track COVID-19 in real time"; The Lancet, Volume 20, Issue 5, May 01, 2020.

4. Kamel Boulos, M.N., Geraghty, E.M. Geographical tracking and mapping of coronavirus disease COVID-19/severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) epidemic and associated events around the world: how 21st century GIS technologies are supporting the global fight against outbreaks and epidemics. Int J Health Geogr 19, 8 (2020).

5. Ezez. 2020. "GISAID - Initiative". *Gisaid.org*. https://www.gisaid.org/.

6. Zhengqiao, Zhao & Rosen, Gail. 2020. "EESI/ISM". *GitHub*. https://github.com/EESI/ISM.