

Article

An Open-Source Workflow for Spatiotemporal Studies with COVID-19 as an Example

Anusha Srirenganathan Malarvizhi ^{1,2}, Qian Liu ^{1,2}, Dexuan Sha ^{1,2}, Hai Lan ^{1,2} and Chaowei Yang ^{1,2,*}

¹ NSF Spatiotemporal Innovation Center, George Mason University, Fairfax, VA 22030, USA; asriring@gmu.edu (A.S.M.); qliu6@gmu.edu (Q.L.); dsha@gmu.edu (D.S.); hlan5@gmu.edu (H.L.)

² Department of Geography and GeoInformation Science, George Mason University, Fairfax, VA 22030, USA

* Correspondence: cyang3@gmu.edu

Abstract: Many previous studies have shown that open-source technologies help democratize information and foster collaborations to enable addressing global physical and societal challenges. The outbreak of the novel coronavirus has imposed unprecedented challenges to human society. It affects every aspect of livelihood, including health, environment, transportation, and economy. Open-source technologies provide a new ray of hope to collaboratively tackle the pandemic. The role of open source is not limited to sharing a source code. Rather open-source projects can be adopted as a software development approach to encourage collaboration among researchers. Open collaboration creates a positive impact in society and helps combat the pandemic effectively. Open-source technology integrated with geospatial information allows decision-makers to make strategic and informed decisions. It also assists them in determining the type of intervention needed based on geospatial information. The novelty of this paper is to standardize the open-source workflow for spatiotemporal research. The highlights of the open-source workflow include sharing data, analytical tools, spatiotemporal applications, and results and formalizing open-source software development. The workflow includes (i) developing open-source spatiotemporal applications, (ii) opening and sharing the spatiotemporal resources, and (iii) replicating the research in a plug and play fashion. Open data, open analytical tools and source code, and publicly accessible results form the foundation for this workflow. This paper also presents a case study with the open-source spatiotemporal application development for air quality analysis in California, USA. In addition to the application development, we shared the spatiotemporal data, source code, and research findings through the GitHub repository.

Keywords: open-source; COVID-19; spatiotemporal analytics; air quality; environmental analysis



Citation: Malarvizhi, A.S.; Liu, Q.; Sha, D.; Lan, H.; Yang, C. An Open-Source Workflow for Spatiotemporal Studies with COVID-19 as an Example. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 13. <https://doi.org/10.3390/ijgi11010013>

Academic Editor: Wolfgang Kainz

Received: 25 October 2021

Accepted: 25 December 2021

Published: 29 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

On 30 January 2020, the Director-General of the World Health Organization (WHO) declared the novel coronavirus outbreak a global health emergency concerning the well-being of the world population [1]. To prevent the spread of the virus and create awareness, timely information about the cause and efficacy of safety guidelines, the total number of confirmed cases, recoveries, and death is critical. The need for COVID-19 datasets drives the demand for opening the data and technologies for fighting the pandemic. The openness of data and technologies offers cross-platform information sharing globally, and users can access the data based on their needs.

Openness refers to transparency, free, inclusive, and unrestricted access to data and information [2]. The adoption of openness has its highest level of application in knowledge [3]. According to the definition of the Open Knowledge Foundation, “Knowledge is open if anyone is free to access, use, modify, and share it—subject, at most, to measures that preserve provenance and openness” [4]. The two major components of knowledge are science and education [3]. Science is the art of building knowledge, and education is the process of transferring knowledge attained from scientific observations and experiments [3].

Educational institutions, governments, and research organizations have started adopting openness into their daily activities.

The advancements in digital technologies such as distributed systems, cloud computing, public repositories, and web services have corroborated the practice of openness in software development. Openness is a virtuous circle with six major components: open-source software, open data, open hardware, open standards, open education, and open science [3]. All the components in the virtuous circle are interdependent:

1. Open-source software describes the free collaborative and interoperability approach of software development.
2. Open data are data that are freely accessible, shareable, and usable.
3. Open hardware includes the machine, physical devices, and build environment.
4. Open standards represent the open process, including the specifications for hardware, software, and data.
5. Open education denotes the transfer of knowledge without restriction.
6. Open science represents the promotion of scientific research and its dissemination of discoveries across the globe.

In the late 90s, people in the software community began using the terms “open” and “free” interchangeably [5]. However, the famous free software activist Richard M. Stallman, argued that both terms differ in their values and perceptions. He stated that “Open source is a development methodology; free software is a social movement.” For the open-source movement, non-free software is a suboptimal solution. For the Free Software movement, non-free software is a social problem, and free software is the solution” [5]. According to him, “free software” grants four freedoms: (i) the freedom to run the program (ii); the freedom to study and adapt the software for own needs; (iii) the freedom to redistribute the software; and (iv) the freedom to improve the software and to release improvements to the public [5]. The notion of free has originated from the idea of freedom and not from the idea of free of cost [6]. The freedom of free software is essential to encourage consensus among the society in sharing and cooperation [7]. The root for free and open source in the geospatial world arose from Open Source Geospatial (OSGeo) Foundation, which started in February 2006. This foundation was the joint efforts of team members focused on free and open-source geospatial projects [8]. In OSGeo, the practice of both open and free geospatial software development coexists. The open-source community in OSGeo encourages access, modification, and redistribution of access to geospatial projects [9]. The foundation of OSGeo highlighted the importance of the collaborative approach to develop open geospatial technologies, data, tools and promote their usage [3]. The open geospatial resources released under Open-Source Initiative (OSI) certified licenses.

Currently, with the proliferation of open-source technologies in the geospatial world, there are sufficiently free and open geospatial software products to satisfy various geospatial research needs, including data collection, crowdsourcing, desktop application, data management systems, and web-based applications [10]. Table 1 shows the widely used free and open-source resources in geospatial applications.

The users of the open geospatial technologies predict that more users and more organizations will adopt open-source geospatial solutions and services in the future [3]. The growing users will add value to the dynamic, diversified, and complex changes in the geospatial domain [3]. Hall et al. [11] emphasized the importance of open standards in the geospatial community. The Open Geospatial Consortium (OGC) was founded in 1994 to provide open specifications to geospatial data access, processing, and visualization. The open standard specifications encourage collaboration, partnerships, and integration among organizations because well-known open standards allow for common tools and practices.

Table 1. Examples of open-source geospatial resources based on various categories.

Free and Open-Source Category	Function	Examples
Desktop GIS	Organizing, analyzing, and visualizing spatial data	GRASS, QGIS, gvGIS, uDig, OpenJUMP
Web GIS services	Web-based GIS services including searching, retrieving, and visualizing	GeoServer, MapServer, QGIS Server, MapCache, deegree
Geospatial libraries	Accessing, analyzing, and spatial data processing	GDAL/OGR, GeoTools, GEOS, PROJ4
Spatial Data Storage	Database management system for spatial data	PostgreSQL/PostGIS, SpatiaLite, SQLite, MySQL Spatial, MongoDB
Geovisualization	Interactive visualization to support spatial analysis	OpenLayers, Leaflet, Cesium, WebGL Earth, OpenWeb Globe
Platform/Language	Provides tools to support handling and analysis of Spatio-temporal data	R (Rspatial) Python (GeoPython, GeoPandas, PySAL, landlab) JavaScript (Leaflet, D3, MapBox, NodeJS)

A spatiotemporal dataset describes snapshots of a natural event that are utilized to understand natural phenomenon. Remote sensors, for example, continuously scan the Earth's features and produce voluminous spatiotemporal data and thus results in tremendous expansion in size of datasets. The mining of knowledge from spatiotemporal data is complex due to various reasons. Rao et al. [12] discussed the complexities associated with the mining and representation of spatiotemporal datasets. These datasets (i) possess continuous and discrete changes of spatial and non-spatial properties, and (ii) are influenced by collocated neighboring spatiotemporal objects of one another. It is crucial to critically address the above-mentioned complexities to produce analyses ready data. Brusdon et al. [13] suggested that openness and reproducible research is important for critical consideration and review of the interpretation of the spatiotemporal data. Researchers widely used sophisticated and commercially available software analytical tools to process spatiotemporal datasets. But these tools have been criticized for offering “black boxes” in which internal analytical processing is not revealed to the users [13]. Additionally, there was less requirement to think about the underlying process that would have resulted in erroneous results. In recent years, researchers have been moving towards free and open software and tools. For example, they develop open-source packages to process spatiotemporal datasets using high-level scripting languages such as R and Python. These languages offer advantages such as free of cost, understanding the internal processes, flexibility to easily replicate, and quick availabilities of new methods. In addition, open source packages, software, and tools enable anyone to access, modify or reuse. Hence, it removes the barriers and offers “data democracy” [3]. Among multiple areas in which open-source technologies have been used, epidemiology is one of the most prominent. For example, the launch of the EpiFlu database widens access to viral genomes, enabling the virologists to examine the phylogenetics of the virus as it spread across the globe [14]. The EpiFlu open database has recently been widely used during the COVID-19 pandemic to study the disease-causing virus and its attributes [15]. The global information sharing enabled through open-source technologies provides solutions to address issues such as sharing heterogeneous, inconsistent, and incompatible epidemiology data [3]. In addition to the open database, researchers across multiple disciplines have come together to collect daily counts of country-level case data, develop dashboards, prediction models, and analytical tools to monitor the spread of the virus. The scientific community shared their findings with the public to create awareness and make better informed decisions [16]. Until now, open-source research on COVID-19 is focused on sharing COVID-19 related datasets and images with the community. But there is no defined open-source software development process that provides a collective knowledge and sharing platform integrating multiple domains such as policy, environment, public place reopening, social media, and publication.

To fill this gap, we proposed and demonstrated the open-source workflow process followed in the NSF Spatiotemporal Innovation Center for spatiotemporal research. The main objective of using the workflow is to standardize spatiotemporal research using open data and derive useful patterns to understand a phenomenon. This paper used the open-source workflow proposed and KNIME to generate and visualize daily mean concentrations of spatiotemporal air pollutants such as nitrogen dioxide (NO₂), PM_{2.5}, PM₁₀, ozone, and carbon monoxide (CO).

Additionally, the workflow emphasizes a collaborative and collective approach by sharing free and open geospatial resources, including data, source code, analytical tools, and ancillary information. The openness in sharing resources encourages other researchers to access, download, modify, reuse, and replicate them for their needs and recontribute to the community.

The organization of this paper is as follows: Section 2 discusses the previous studies on open-source technologies, usage, and challenges. Section 3 describes the open-source workflow, including (1) developing a geospatial application, (2) sharing and maintenance, and (3) reproducible research. Section 4 provides a use case illustrating the development of an open-source geospatial application for air quality analysis in California, the USA, using the open-source workflow. Section 5 discusses the traffic and commit statistics of our COVID-19 data repository, which shows the popularity of the open-source geospatial resources published by the team. The paper's conclusion discusses the center achievements enabled by open-source efforts to support the COVID-19 research and proposes the scope of improvements for future work.

2. Literature Review

2.1. Available Open-Source COVID-19 Digital Resources and Their Issues

The role of an open-source platform is essential during COVID-19 [17]. Researchers and organizations publish COVID-19 related case data, risk factors, government policies, and economical data through open-source technologies. The decision-makers can easily extract the data and use it for their own needs. The primary open resources are daily infection cases, policy data, social media data, health risk factors, mobility data, and publication data. The John Hopkins University Center for Systems Science and Engineering (JHU CSSE) provides global daily counts with an interactive dashboard [18]. It is one of the most widely used datasets in research studies. The Oxford COVID-19 Government Response Tracker (OxCRT) collects information on common policy measures governments have undertaken across the world to fight the pandemic, records the stringency scores of the policy measures, and aggregates those scores into policy indices [19].

Several previous studies have reviewed available open-source COVID-19 resources and identified the underlying issues. For example, Alamo et al. [17] have reviewed the available open data resources and data-driven technologies to better understand the worldwide spread of the pandemic. The researchers listed variables such as COVID-19 time series data, health risk associated with age and secondary medical conditions, and government measures to facilitate the communities. Additionally, they have identified repositories relevant to COVID-19 cases both at global and regional scales. The team identified issues such as inconsistent data, changing criteria, heterogeneous sources, and no standard metrics to compare data between countries with the current open-source data sources with the open data resources. Hu et al. [16] built an open digital resource repository, based on Harvard Dataverse [20], for data management and sharing platforms to publish the data relevant to COVID-19 research. The team collected daily counts of COVID-19 confirmed cases, global news, social media, population mobility, health facilities, and scholarly articles for building the repository. The collected data were archived into the Harvard Dataverse and performed preliminary data analysis. The team mentioned issues, including missing values and outdated basemaps in the current dataset, and progress towards providing continuous support and maintenance for data archiving. Shuja et al. [21] surveyed the existing COVID-19 open-source datasets and illustrated the future challenges. The researchers organized

the review based on the type of data and application. Medical images (CT scans and X-ray) and textual data (COVID-19 case reports, tweets, and scholarly articles) formed the main data types. The research found that more work is required on cough-based COVID-19 diagnosis by analyzing the applications of open-source datasets. The additional efforts would improve the deep learning techniques by expanding open access to CT scan and X-ray dataset and ensure privacy for patient and mobility data and contact tracing. Table 2 presents the currently available open-source COVID-19 dataset examples based on different data types and applications.

Table 2. COVID-19 open-source datasets.

Study	COVID-19 Datasets	Data Type	Application	Link
[22]	CT scans and X-ray	Image	COVID-19 Diagnosis and infected area segmentation	https://github.com/ieee8023/COVID-chestxray-dataset (accessed on 14 December 2021)
[23]	Daily cases	Textual	Reporting and visualizing	https://github.com/CSSEGISandData/COVID-19 (accessed on 14 December 2021)
[24]	Tweets	Textual	Conversation dynamics	https://github.com/eichen102/COVID-19-TweetIDs (accessed on 14 December 2021)
[25]	Scholarly articles	Textual	Reporting	https://www.semanticscholar.org/cord19/download (accessed on 14 December 2021)
[26]	Environmental factors	Textual	Reporting	https://github.com/stccenter/COVID-19-Data/tree/master/Environmental%20factors (accessed on 14 December 2021)

2.2. Significance of Open and Collaborative Approach

Marivate et al. [27] argued the importance of open and collaborative data during the COVID-19 pandemic. Focused on the African continent, the team collected COVID-19 related data and published it for public use. The research team stated that data collection and management should be a proactive approach that is not implemented only during a pandemic. Wang et al. [28] studied the COVID-19 themed GitHub repositories by classifying them into six categories. The six categories of COVID-19 themed repositories are (i) data repositories, include COVID-19 case statistics, image datasets, data visualization, and analysis; (ii) contact tracing repositories include applications and frameworks developed for exposure notifications; (iii) toolkit repositories include tracking toolkits for COVID-19 implemented using mobile applications, APIs, and Python packages; (iv) forecast and simulation repositories include predicting and simulation models to study the spreading of disease; (v) detection and diagnosis repositories include COVID-19 diagnosis models from chest CT or X-rays; (vi) other repositories include data that are not directly relevant to COVID-19 but significantly impact people's lives during the pandemic. The researchers showed a promising way to respond to the COVID-19 pandemic through open-source technologies and resources. By investigating the patterns of contribution, development, and maintenance of these repositories, the data-themed repositories are found to have more commits than other similar repositories because they require regular commits, and communities use them to keep track of COVID-19 data.

Brunsdon et al. [13] discussed the three essential factors for critical data science: openness (open data, open code, and open disclosure of methodology), collective working (sharing, collaboration, peer review), and reproducibility (methodological and inferential transparency). The researchers asserted that a critical approach to spatial data analysis is to be aware of issues with the crucial misunderstanding while using geospatial data

that leads to misinformed decision making. By adopting openness and reproducibility, the scrutinizing geospatial data analysis enhances by review, comment, and suggestions. Sharing code, tools, and open code libraries using repositories provide a platform for transparent and reproducible research. Singleton et al. [29] argued that the validity of the scientific work depends on how the methodologies employed are reproducible. The reproducible work can be achieved by workflows utilizing open-source software and data. The methodology describing the process of converting raw geospatial data into information formed the basis of reproducible research.

2.3. Platforms That Support Open-Source Collaboration

GitHub is one of the most widely used code hosting platforms for version control and collaborative software development. GitHub offers a distributed version control system where each collaborator has a separate branch required in the software development process. Dabbish et al. [30] examined the visible cues of a user's behavior on the GitHub page. The research study found that the community's collaboration, learning, and reputation management enhances using GitHub. The GitHub users make social inferences from the activity information of other users. For example, recency and volume of activity indicate the liveliness and level of indication of the developers. The inferences made on visible cues could be built into effective strategies to enhance coordination, transparency, and learning in the complex software development process. Zagalsky et al. [31] examined the social and collaborative features in GitHub and found that GitHub can be an ideal tool for knowledge sharing and education based on various themes such as transparency of activity, encourage participation, reusability and sharing, and ease to use. Though GitHub offers competitive advantages for open-source projects, it is not ideal for remote use due to its complexity [32]. It requires SSH keys for both local and remote machines to access the Git server. This burden makes it an obstacle for new users to adopt and learn it.

Apache Subversion, often abbreviated as SVN, is used as a software versioning and revision control system. SVN is used to maintain current and historical versions of open-source code and other digital resources in software development. Unlike Git, SVN is built upon a centralized version control system. In a centralized system, all collaborators work on the same repository. During each commit, the collaborator must check out the latest version to avoid conflicts [32]. The major disadvantages of Subversion are scalability and version policies that are not well-defined [33].

2.4. Challenges in Open-Source Software Development and Collaboration

Stol et al. [34] identified the potential challenges in adopting open-source software development. They conducted a literature review and identified 21 challenges categorized into six aspects: product selection, documentation, community support, maintenance, integration and architecture, migration and usage, and legal and business. One of the significant challenges represented in the research paper was difficulty in identifying quality products among the large pool of products. It is a major concern due to ambiguity about the quality, usability, and reliability of open-source products. Another challenge was the lack of quality in the documentation. Overall, the researchers pointed out challenges in open-source software development were mainly related to uncertainty in ensuring quality and proper maintenance. Ankolekar et al. [35] examined the challenges to collaborate in open-source software development. The research team pointed out that the main hindrance to collaboration is the lack of communication between team members, resulting in the lack of awareness and blocking information crucial for development. Finding and seeking the right help is difficult when developers are not physically located at the exact location. The team also found that informal communication is nearly absent in open-source collaboration, but it is essential to coordinate the activities. To overcome these challenges, the team suggested encouraging social interaction and social structure within the open-source community.

2.5. Open-Source Software Development Approach

We searched the CiteSeer for published articles related to open-source software development. CiteSeer is a public digital library that focuses mainly on computer and information science. The search terms used are “open-source software development” OR “free and open-source software development”. There has been very little research that discusses the open-source software development approach except the one reviewed hereinafter.

Scacchi et al. [36] identified the requirements for open software development and specified the artifacts that characterize how the requirements for developing open software systems. The paper compared the classic software requirements engineering process and open-source processes for developing requirements. Specifically, the paper highlighted the software informalism approach in open-source software development. Mockus et al. [37] illustrated the case study of open-source software (OSS) development used in Apache server. Two sets of key properties of OSS development were discussed by researchers. The first set analyzed the basic parameters of Apache development process. The parameters include process used to develop Apache, the number of people involved, the nature of the role carried out by people, and code ownership. The second set was related to the outcomes of the Apache development process, such as defect density of the code and time taken to resolve problems. Kon et al. [38] discussed the development process of free and open-source software development. The process starts with the code development of a software product. Then, it undergoes multiple revisions that propose any changes to fix defects. Lastly, the final version is made available to the end-users.

Software developed by the open-source approach are listed in Table 3 and compared with their license, formal development approach, testing, maintenance, code availability, and documentation. One of the major setbacks in open-source development is that there is no universally agreed upon formal project design or process. As a result, there are issues in ownership, people contribution, and defect detection rate. For example, in the Apache and GNOME development community, only the top few core developers actively participated in a significant portion of software development. Thus, the lack of many active contributors results in enormous defects in the software that are either unidentified or not rectified. Additionally, the top core developers become too overburdened to handle both the development and defect fixing activities.

Table 3. A selected list of collaborative and open-source research using a collaborative platform.

Name	License	Formal Process	Testing	Maintenance	Code Availability	Documentation
GNOME [39]	GNU	Yes	Yes	Yes	Yes	Yes
Apache [37]	Apache	No	Yes	Yes	Yes	Yes
FreeBSD [40]	FreeBSD	No	Yes	Yes	Yes	Yes
GRASS [41]	GNU GPL	No	Yes	Yes	Yes	Yes

While the above-discussed open source projects successfully delivered the products or services to the end-users, challenges exist due to the lack of standard workflow. The proposed paper introduces a novel, standard, and easily replicable workflow suitable for spatiotemporal research.

The major challenges identified in the literature review are users’ concerns about the quality of the open resources, lack of documentation, maintenance, and community support. These challenges are addressed by proposing the open-source workflow discussed in Section 3. To ensure the quality of the shared resources, we validate the usability of open-source geospatial resources by peer members. Additionally, we provide detailed documentation and tutorial videos that offer comprehensive understanding for any individual with or without a technical background to replicate the research efficiently. The open resources published are regularly maintained and supervised by the team members,

evident from the GitHub commit history. The results are free to download; users can access, modify, and run the open-source tool for their research needs and redistribute it.

3. Methods

An event is termed as something that happens at a specific time and place [42]. Yu et al. [43] proposed spatiotemporal event definitions based on three stages. On the first stage, time and space dimensions are involved; in the second stage, participants are involved in addition to space and time. In the final stage, along with other information, change of state in the object is involved. Extracting useful information from an event consists of a sequence of steps from data collection through sensors, information extraction, and identifying attributes of interest, and applying the knowledge for decision making. Thus, spatiotemporal events serve as a triggering point for spatiotemporal application development. Spatiotemporal application development is applicable in various domains such as environmental sustainability and management, health management, urban mapping and development, business intelligence, crisis, crime, and social unrest monitoring [43]. The users from these domains demand improved support for systems involving spatiotemporal events. To better understand the user requirements and spatiotemporal concepts, we propose this open-source workflow for spatiotemporal research and application development. To advance the research and encourage collaboration, we initiated the development as an open source. The workflow has three stages, starting with spatiotemporal application development, sharing and maintenance, and reproducible research.

3.1. Spatiotemporal Application Development

Figure 1 illustrates the open source workflow for spatiotemporal research. Firstly, we included the three traditional concepts of spatiotemporal applications in the development phase as its core. Tryfona et al. [44] proposed the spatiotemporal application based on the type of spatiotemporal data. They are: (i) applications may involve objects with continuous motion, (ii) applications dealing with discrete changes of and among objects, and (iii) applications may manage objects integrating continuous motion as well as changes of shape. To better understand the spatiotemporal applications, the initial step is to understand the modeling and user requirements of a spatiotemporal application. Pfoser et al. [45] drew the set of spatiotemporal requirements: The various needs are (i) representation of objects with the position in space and time; (ii) capture the position change in space over time whereby continuous and discrete changes are considered depending on the type of application of interest; (iii) definition of spatial attributes over time; (iv) capture the change of spatial attributes over time; (v) connection of spatial attributes to objects; (vi) representation of spatial relationships among objects in time; (vii) representation of relationships among spatial attributes in time; and (viii) specify spatiotemporal integrity constraints.

After identifying the requirements of spatiotemporal research, we specify the spatiotemporal modeling environment. Based on the above-discussed modeling requirements, we designed the environment for spatiotemporal data representation. The modeling environment determines the infrastructure and database design for spatiotemporal application development. The database design should support spatial, temporal, and spatiotemporal concepts. According to Pfoser et al. [45], spatial and temporal concepts would exist independently and then be combined as spatiotemporal concepts. The spatial concepts included space, objects, attributes, relationships, and layers, and the temporal concepts included time snapshots, intervals, and periods. The spatiotemporal concepts included spatial objects in time points, spatial objects in time intervals, spatial objects in time periods, layers in time points, layers in time intervals, and layers in time periods. To advance the current needs of event detection, we adopt other essential concepts such as theme (represents what), participants (represents who), and thematic attributes such as pollution, socio-economic factors, crime, and vegetation [43].

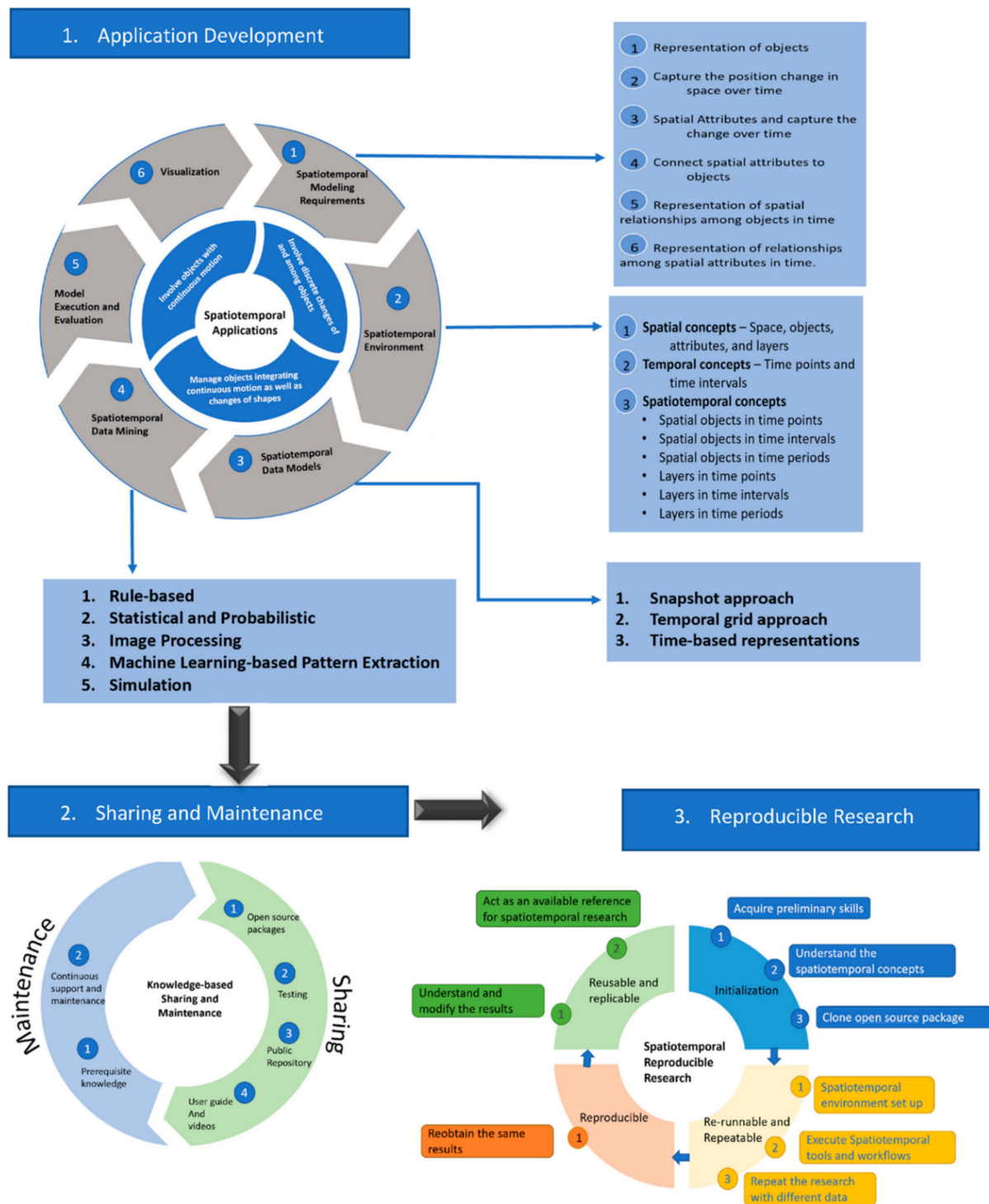


Figure 1. Open source workflow for spatiotemporal research.

In the next step, we included the spatiotemporal data models. Peuquet et al. [46] proposed the different approaches to represent spatiotemporal data models. They are (i) snapshot-based approach; (ii) temporal grid approach; and (iii) time-based approach. Selecting the right approach for data modeling is a challenging task. Particularly, the emergence of big spatiotemporal data requires specialized approaches due to its tremendous volume and complex data structures [11].

After spatiotemporal data modeling, the next step is to extract useful patterns from spatiotemporal data. Then, the patterns are interpreted by the scientists to obtain new

insights about the object of interest. Yu et al. [43], summarized the extraction methods as rule-based, statistical and probabilistic, image processing (pixel-based and object-based detection), machine learning-based pattern extraction (hotspots, outliers, and change detection), and simulation (numerical, statistical and agent-based simulation).

The next step in spatiotemporal application development is model execution and evaluation. The model execution requires (i) computational platforms [47] such as Google Earth Engine, Amazon Web Services (AWS), and Microsoft Planetary Computer Hub, (ii) GIS softwares including commercial software (ESRI) and open-source software (QGIS), (iii) spatial statistical tools (GDAL, R packages such as spatstat, gstat, geoR, and spdep, KNIME Spatial Processing Nodes) [48], (iv) spatial Database Management system (Oracle Spatial, PostGIS, DB2 Spatial Extender), and (v) Spatial Big Data Platform (Hadoop GIS, Spatial Hadoop). After model execution, we examined the residues in the model. Using visualizations, we analyzed the temporal and spatial distribution of the residuals in a model. If the distribution appears as a random noise, we may choose to revise the model. The final stage is the model interpretation and utilization of the results. Depending on the purpose, the model results can be utilized. For example, a model can predict an unknown pattern or feature or provide insight of an outlier.

3.2. Sharing and Maintenance

We shared the developed application with the community during the second phase of the open-source workflow and maintained it. The sharing stage includes (i) developing open source packages, (ii) testing, (iii) creating a public repository, and (iv) creating tutorial videos and user guides. The open source package comprises spatiotemporal datasets, source code, models, tools, and results. The open resources available for public use should be free of errors and ready to use with the required information. Peer review testing is a cost-effective way to test software product quality and detect defects [49]. In peer review testing, members who are not involved in the software development process contribute to testing the code written by their peers. We carried out peer review testing to verify the quality of the open resources before publishing them to the community.

We leveraged the GitHub public repositories to publish these resources and share them with the community. It allows other researchers to freely download, reuse, modify, and redistribute the available resources for reproducible research. Finally, we made user guides and tutorial videos that demonstrated detailed step-by-step instructions on replicating the research process. The user guide and tutorial video specified the software requirements, environmental set up to run the spatiotemporal analytical tool, required libraries, input parameters, and a way to execute the analytical tool.

The ultimate purpose of continuous support and maintenance is to ensure the value of open-source software over time; we constantly enhance their capabilities, delete obsolete data, and optimize the code when necessary. The open source packages require continuous maintenance to keep the software packages up to date. However, people involved in maintenance tasks require pre-requisite technical skills and spatiotemporal knowledge. Technical skills include keeping the python packages up to date and troubleshooting any compatibility issues in the python environment and packages.

3.3. Reproducible Research

To transform the spatiotemporal applications into scientific contributions, we shared the applications in the open source community through GitHub. When a researcher executes the application, it should yield the same result, termed reproducible research [50]. A researcher should possess certain preliminary skills to reproduce spatiotemporal research. The preliminary skillsets include using spatiotemporal analytical tools, GIS software, and spatial DBMS. In addition to that, the researcher should know certain concepts in the spatiotemporal world. For example, Tobler's first law of geography [51] says, "Everything is related to everything else, but near things are more related than distant things". If a

researcher ignores this law in spatiotemporal analysis, it will result in an inaccurate and inconsistent interpretation of results.

Spatiotemporal modeling environment encompasses various dependencies such as systems, software, libraries, packages, and tools. Even the most minor change in one of the dependencies or version conflict would be a major hindrance to re-run the spatiotemporal application. Therefore, we describe the spatiotemporal execution environment with details to effectively reproduce the research. When a researcher can re-run the spatiotemporal application with the modeling environment, they should also be able to repeat the process by successively running the application more than once. A deterministic spatiotemporal model will produce the same result on the ‘n’ number of runs.

Reproducing the same result as the published results is the major challenge in reproducible research. To avoid issues in reproducibility, we provide as much as information that is necessary. For example, while specifying the python version that supports the spatiotemporal application, we mention the exact version that produces the results because it is common in programming languages to show different behavior in different versions [52]. The complexities in spatiotemporal research are scrutinized if we ensure reproducibility. Reusability is the next important aspect of reproducible research. It enables the researchers to use, modify, and distribute spatiotemporal applications.

4. Use Case—Assess the Impact of Air Quality during COVID-19 Based on the Proposed Open-Source Spatiotemporal Workflow

Figure 2 illustrates the development of a spatiotemporal application to assess the impact of control measures of COVID-19 on air quality in California, USA. The analytical tool is built using open air pollutants data.

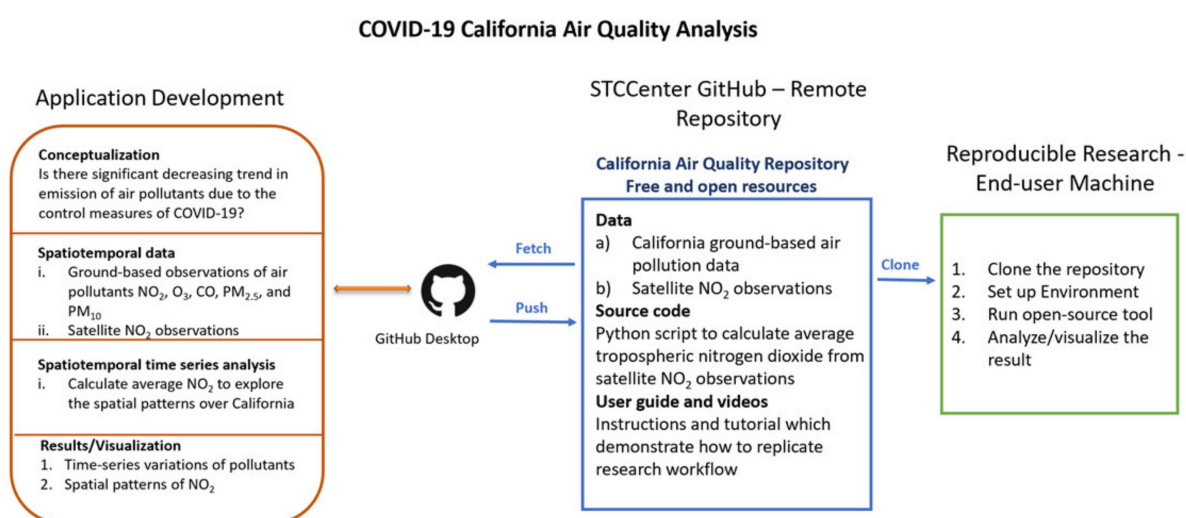


Figure 2. Spatiotemporal open-source workflow for COVID-19 California Air Quality Analysis.

4.1. Hypothesis

The societal efforts to mitigate the spread of the 2019 COVID-19 pandemic caused significant impacts on the environment. This study investigated “if the interventional policies due to COVID-19 have had a similar impact in the US state of California”. Specifically, we aimed to confirm whether the decreasing trends in the emission of air pollutants in California, USA, were related to control measures of COVID-19. Decision-makers can then use the analytical results to assess the economic impact caused by the mitigation policies.

4.2. Open Data Sources

1. Ground-based air pollutants data required for this study are obtained from the US Environmental Protection Agency (EPA). The data are downloaded from this link: <https://www.epa.gov/outdoor-air-quality-data/download-daily-data> (accessed on 14 December 2021).
2. Tropospheric nitrogen dioxide needed for this study is obtained from GES DISC: https://disc.gsfc.nasa.gov/datasets/OMNO2d_003/summary (accessed on 14 December 2021).
3. The locations of major power plants in CA are obtained from Wikipedia: https://en.wikipedia.org/wiki/List_of_power_stations_in_California (accessed on 14 December 2021).
4. The locations of major wildfires are downloaded from the California Department of Forestry and Fire Protection (CAL FIRE): <https://www.fire.ca.gov/incidents/2020/> (accessed on 14 December 2021).
5. The locations of national highways in California are obtained from the official Website of US Census Bureau, Department of Commerce: <https://catalog.data.gov/dataset/tiger-line-shapefile-2016-nation-u-s-primary-roads-national-shapefile> (accessed on 14 December 2021).

4.3. Spatiotemporal Data

The study was conducted through the comparison among three periods, before (January 26–March 18), during (March 19–May 8), and after (May 9–June 14) the lockdown of California in 2020 and compared the emission patterns of air pollution with annual means of 2015–2019. California was selected as the study area of this research due to its severe condition of the spread of COVID-19 and high air pollution rates. We collected ground-based observations of air pollutants such as NO₂, O₃, CO, PM_{2.5}, and PM₁₀ obtained from the Environmental Protection Agency (EPA) and satellite NO₂ observations acquired from Ozone Monitoring Instrument (OMI) aboard NASA's Earth Observing System's (EOS) Aura satellite.

4.4. Spatiotemporal Analytical Tool

KNIME is an analytical software that supports integration of complex software tools and libraries. It reduces the time and errors faced by researchers to set up the programming environment. The KNIME workflows can be easily exported and shared with others who may not have an expert programming skill. The researchers can simply execute the workflow to reproduce the results or reuse it for their own analysis.

Figure 3 shows the KNIME workflow, an open-source spatiotemporal analytical tool, developed using python packages such as numpy, netCDF4, h5py, math, and scipy. The tool is used to generate the daily mean concentration of each pollutant in California for both 2020 and 2015–2019 data.

The spatial patterns of atmospheric NO₂ were also investigated over California by comparing 2020 data to the historical means. We used (i) Python geospatial libraries such as GDAL, Fiona, Geopandas to extract spatial information from vector and raster files; (ii) netCDF package to read and write netCDF files; (iii) matplotlib and basemap packages to visualize the spatial patterns. We formalized the spatiotemporal concepts for this research with spatial object as California, temporal concepts as three periods before (January 26–March 18), during (March 19–May 8), and after (May 9–June 14) the lockdown of California in 2020 and compared the emission patterns of air pollution with annual means of 2015–2019 and thematic attribute as air pollution.

The air pollutants' emissions were analyzed for the three periods above separately. Then the percentage changes between different periods were calculated to confirm if the COVID-19 pandemic has influenced the air pollutants' emissions.

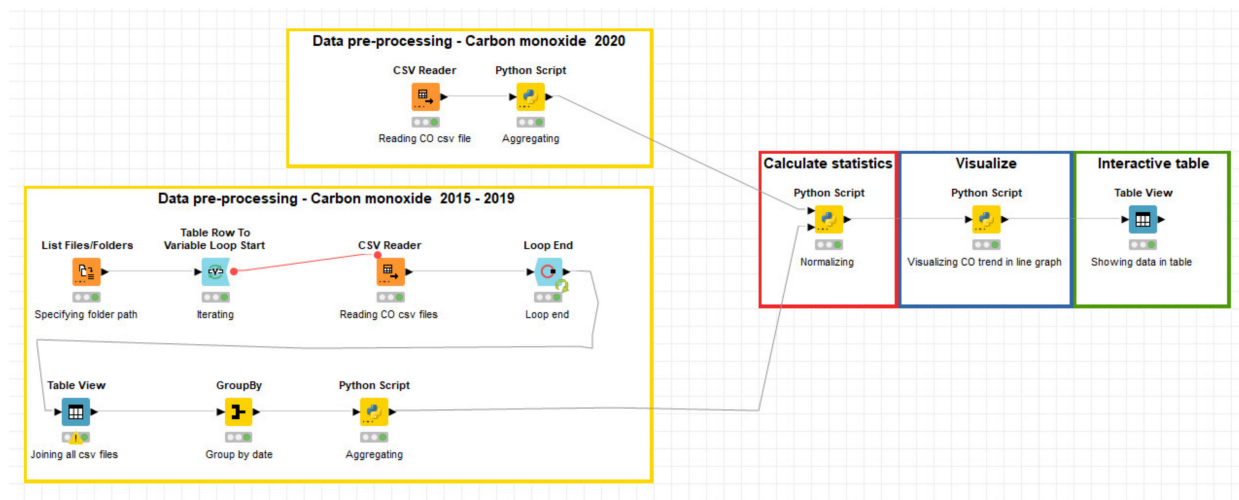


Figure 3. KNIME workflow to estimate the daily mean concentration of pollutants in California for both 2020 and 2015–2019.

4.5. Results/Visualization

With the air pollutants, spatiotemporal analytical tool, and guidance shared in the repository, the programming environment was set up, and the analytical tool was executed for research replication. Figure 4 shows the replicated result of daily variations in 7-day moving averages of the CO concentration over CA. Figure 5 shows the replicated result of spatial patterns of tropospheric NO₂ TVCD over CA post-period in 2020.

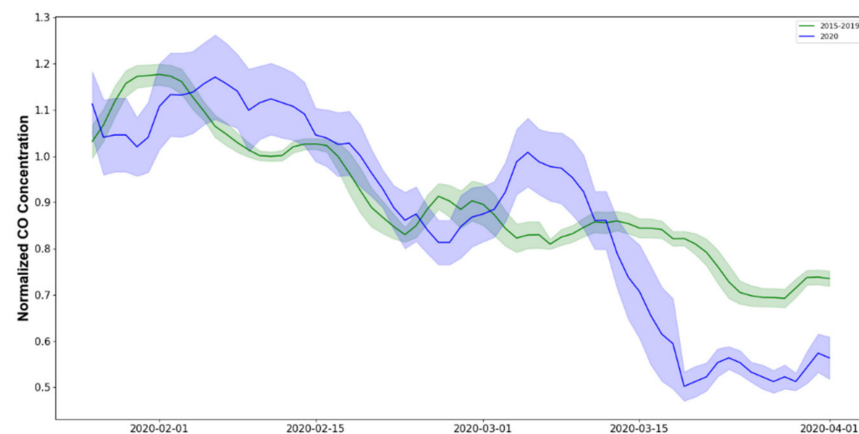


Figure 4. Daily variations in 7-day moving averages of the CO concentration over CA (green represents average CO of past years, purple blue represent CO of 2020, Replicated from [53]).

4.6. Sharing and Maintenance

We shared the air pollutants data, source code to calculate the mean concentration of air pollutants, and results from the research work with the public through the GitHub repository of STC. The GitHub repository for COVID-19 analysis on air quality was named “COVID-19”. Inside this repository, under the “analysis/CA-Air Pollution” folder, we stored the resources relevant for this air quality analysis. Figure 6 shows the air pollutants data, analytical tool, and results shared in the STC COVID-19 GitHub repository.

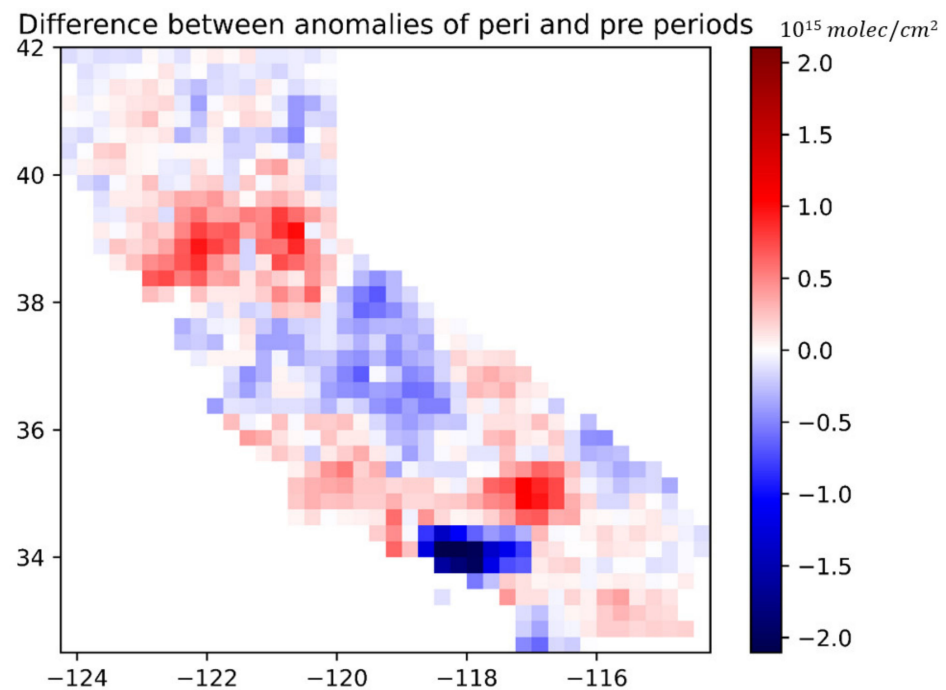


Figure 5. Spatial patterns of changed tropospheric NO₂ TVCD over CA (post-period of 2020 in comparison to peri-period of 2020). (Replicated from [53]).

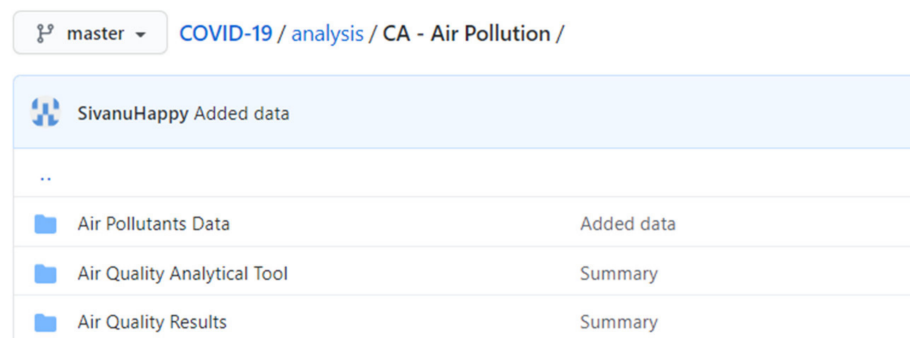


Figure 6. Air pollutants' data, analytical tool, and results shared in the STC repository.

The CA-Air Pollution folder has three sub-folders, namely:

- “Air Pollutants Data” folder contains “CA ground-based air pollution data” of carbon monoxide, ozone, nitrogen dioxide, PM₁₀, PM_{2.5}, and sulfur dioxide and “Satellite-based NO₂ data” for the study period 2020.
- “Air Quality Analytical Tool” folder contains the python script (OMI_static_ca.py) to calculate the daily mean of nitrogen dioxide and calculate the difference between the two study periods.
- “Air Quality Results” folder contains a spreadsheet named california_counties_COVID_env_data.xlsx. It has a daily average concentration for each pollutant.

4.7. User Guide and Video

We created a user guide that provided step-by-step instructions to replicate the California air quality analysis. The user guide has the following information to assist researchers for replication:

1. Specify the required python packages.
2. Provide instructions on how to set up the virtual environment.
3. Install the python packages using pip.
4. Show where to obtain the input datasets.
5. Execute the analytical tool and obtain the results.

Additionally, we also made a tutorial video that followed the above sequence of steps to replicate the air quality analysis research. We used YouTube to share the video for public use.

5. Traffic and Commit Statistics

We used unique clones and visitor counts for 14 days in May 2021 (from 16 May 2021 to 29 May 2021) to visualize the traffic in our repositories. Figures 7–9 show the number of unique clones, the number of unique visitors, and popular content for the 14 days of the COVID-19 data repository. These figures provide insight into the interest of the community in our collected open datasets. During the 14 day period, we have a total of 67 unique clones and 77 unique visitors. The national COVID-19 daily case data collection is the most popular content in the repository, with a count of 171 views. We used commit statistics to show our activeness and commitment to maintaining up-to-date information in the repository. Figure 10 shows the number of commits for the past year. We have been updating country-level COVID-19 daily case data using GitHub Actions.



Figure 7. Number of unique clones of COVID-19 Data repository for past 14 days.

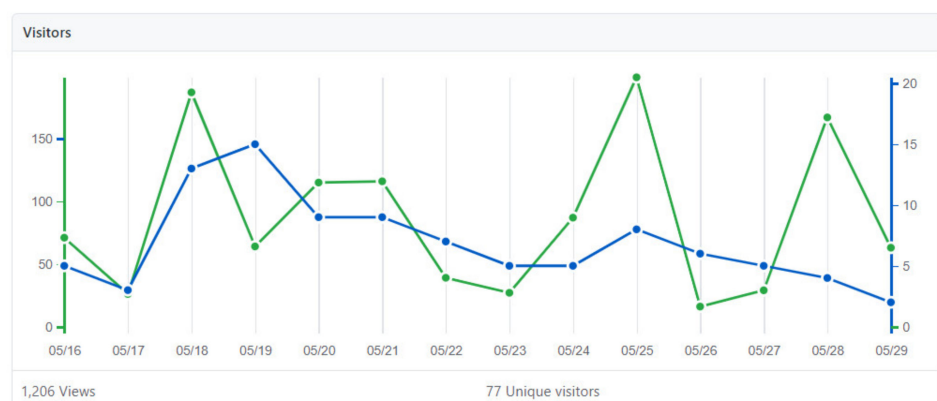


Figure 8. Number of unique visitors of COVID-19 Data repository for past 14 days.











Popular content		
Content	Views	Unique visitors
 stccenter/COVID-19-Data: COVID-...	171	50
 COVID-19-Data/scripts at master	73	1
 stccenter/COVID-19-Data: COVID-...	55	1
 COVID-19-Data/admin_validation...	46	1
 Actions	44	1
 COVID-19-Data/Socioeconomic D...	35	12
 COVID-19-Data/India at master	31	4
 COVID-19-Data/admin_0_1.py at ...	30	1
 COVID-19-Data/US at master	29	11
 COVID-19-Data/report_generation...	27	1

Figure 9. Popular content of COVID-19 data repository for past 14 days.

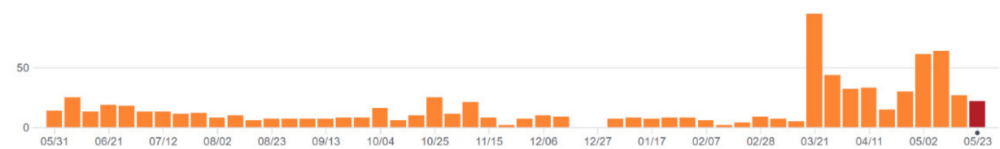


Figure 10. Number of commits of COVID-19 data repository.

6. Conclusions

In this paper, we designed an open-source workflow to develop open-source geospatial applications for environmental analysis. We also discussed a use case with the development of an open-source geospatial project to confirm the decreasing trend in air pollutants in California, USA. We shared the data, analytical tools, and results with the community and made them available for free access. We have successfully deployed COVID-19 related environmental and policy analysis research and country-level daily case data to the public GitHub repository. The shared geospatial resources have garnered considerable attention in the community. Specifically, the COVID-19 daily case data repository has 77 unique visitors from 16 May 2021 to 29 May 2021. The open geospatial resources provide a comprehensive understanding of the pandemic situation. For example, the environmental analysis helps the decision makers to assess the control measures of COVID-19 by tracking the emissions of air pollutants. The emission of air pollutants implies anthropogenic activities such as transportation, operations of industries and power plants, and fossil fuel consumption. The research findings of policy analysis help to understand the impact of policy on mobility, case, and mortality rate. Among many practical advantages of adopting open source in a research environment, fostering a collective and collaborative approach among researchers is critical for COVID-19 rapid response to work towards a common objective. We used GitHub as our open-source collaborative platform. It serves as a single-point solution to publish the geospatial resources and offers knowledge-based sharing. Some of the challenges shown in the previous studies with open-source technology were lack of proper documentation and quality in the open resources. We addressed the quality issue by employing peer-review testing; team members who were not directly involved in the development process performed peer-review testing. The peer-reviewers validate the data, tools, and results before publishing to the community. Additionally, we provided step-by-step instructions and tutorial videos to replicate the research.

The open-source workflow has benefited spatiotemporal research to be easily reproducible and recreate the same results. Further, opening the source code, tools, data, and results to the public promoted the research work and enhanced the credibility of our contribution to the community. The open-source workflow of COVID-19 research followed

a rigid and defined set of activities in each stage. However, with the rapid changes in research scope and requirements, an open-source workflow is needed to support flexibility and rapid adoption. Agile methodologies support iterative and evolutionary development, rapid and flexible response to changing requirements, and promote adaptive planning. In the future, we will improve the open-source workflow to be more agile and rapid prototyping and standardize the workflow for community adoption to support spatiotemporal studies. The rapidness in the open-source software development improves the value of the software by delivering working software by an incremental approach.

Author Contributions: Conceptualization, Chaowei Yang and Anusha Srenganathan Malarvizhi; methodology, Anusha Srenganathan Malarvizhi; software, Qian Liu; validation, Anusha Srenganathan Malarvizhi; formal analysis, Chaowei Yang; investigation, Hai Lan and Dexuan Sha; data curation, Qian Liu and Anusha Srenganathan Malarvizhi; writing—original draft preparation, Anusha Srenganathan Malarvizhi; writing—review and editing, Chaowei Yang and Qian Liu; visualization, Anusha Srenganathan Malarvizhi; supervision, Chaowei Yang; project administration, Anusha Srenganathan Malarvizhi; funding acquisition, Chaowei Yang. All authors have read and agreed to the published version of the manuscript.

Funding: Research is supported by NSF (2027521, 1835507 and 1841520).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in <https://github.com/STC/COVID-19/tree/master/analysis/CA%20-%20Air%20Pollution> (accessed on 14 December 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. World Health Organization—Interactive Timeline. Available online: <https://www.who.int/emergencies/diseases/novel-coronavirus-2019/interactive-timeline> (accessed on 10 May 2021).
2. Brovelli, M.; Ilie, C.M.; Coetzee, S. Openness and community geospatial science for monitoring SDGs—An example from Tanzania. In *Sustainable Development Goals Connectivity Dilemma: Land and Geospatial Information for Urban and Rural Resilience*; CRC Press: Boca Raton, FL, USA, 2019; pp. 313–324.
3. Coetzee, S.; Ivánová, I.; Mitsova, H.; Brovelli, M.A. Open geospatial software and data: A review of the current state and a perspective into the future. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 90. [CrossRef]
4. Open Definition. Open Knowledge Foundation. Available online: <http://opendefinition.org/> (accessed on 19 April 2021).
5. Stallman, R. *Free Software, Free Society: Selected Essays of Richard M. Stallman*; Lulu. Com: Morrisville, NC, USA, 2002.
6. Steiniger, S.; Hunter, A.J.S. The 2012 free and open source GIS software map—A guide to facilitate research, development, and adoption. *Comput. Environ. Urban Syst.* **2013**, *39*, 136–150. [CrossRef]
7. Stallman, R. Viewpoint Why open source misses the point of free software. *Commun. ACM* **2009**, *52*, 31–33. [CrossRef]
8. The Open Source Geospatial Foundation. Available online: <https://www.osgeo.org/> (accessed on 19 April 2021).
9. What Is Open Source? Available online: <https://www.osgeo.org/about/what-is-open-source/> (accessed on 19 April 2021).
10. Brovelli, M.A.; Minghini, M.; Moreno-Sanchez, R.; Oliveira, R. Free and open source software for geospatial applications (FOSS4G) to support Future Earth. *Int. J. Digit. Earth* **2017**, *10*, 386–404. [CrossRef]
11. Hall, G.B.; Leahy, M.G. *Open Source Approaches in Spatial Data Handling*, 2nd ed.; Springer: Berlin, Germany, 2008.
12. Rao, K.V.; Govardhan, A.; Rao, K.C. Spatiotemporal data mining: Issues, tasks and applications. *Int. J. Comput. Sci. Eng. Surv.* **2012**, *3*, 39.
13. Brunsdon, C.; Comber, A. Opening practice: Supporting reproducibility and critical spatial data science. *J. Geogr. Syst.* **2020**, *23*, 477–496. [CrossRef]
14. Shu, Y.; McCauley, J. GISAID: Global initiative on sharing all influenza data—from vision to reality. *Eurosurveillance* **2017**, *22*, 30494. [CrossRef] [PubMed]
15. Frazer, J.S.; Shard, A.; Herdman, J. Involvement of the open-source community in combating the worldwide COVID-19 pandemic: A review. *J. Med. Eng. Technol.* **2020**, *44*, 169–176. [CrossRef] [PubMed]
16. Hu, T.; Guan, W.W.; Zhu, X.; Shao, Y.; Liu, L.; Du, J.; Liu, H.; Zhou, H.; Wang, J.; She, B.; et al. Building an Open Resources Repositories for COVID-19 Research. *Data Inf. Manag.* **2020**, *3*, 130–147.
17. Alamo, T.; Reina, D.G.; Mammarella, M.; Abella, A. Open data resources for fighting covid-19. *arXiv preprint* **2020**, arXiv:2004.06111.

18. Coronavirus Resource Center. John Hopkins University of Medicine. Available online: <https://coronavirus.jhu.edu/map.html> (accessed on 19 April 2021).
19. COVID-19 Government Response Tracker. University of Oxford. Available online: <https://www.bsg.ox.ac.uk/research/research-projects/covid-19-government-response-tracker> (accessed on 19 April 2021).
20. Harvard Dataverse. Available online: <https://dataverse.harvard.edu/dataverse/2019ncov> (accessed on 19 April 2021).
21. Shuja, J.; Alanazi, E.; Alasmay, W.; Alashaikh, A. COVID-19 Datasets: A survey and Future Challenges. *Development* **2020**, *11*, 12.
22. Cohen, J.P.; Morrison, P.; Dao, L.; Roth, K.; Duong, T.Q.; Ghassemi, M. COVID-19 image data collection: Prospective predictions are the future. *arXiv preprint* **2020**, arXiv:2006.11988.
23. Dong, E.; Du, H.; Gardner, L. An interactive web-based dashboard to track COVID-19 in real time. *Lancet Infect. Dis.* **2020**, *20*, 533–534. [[CrossRef](#)]
24. Chen, E.; Lerman, K.; Ferrara, E. Tracking social media discourse about the covid-19 pandemic: Development of a public coronavirus twitter data set. *JMIR Public Health Surveill.* **2020**, *6*, e19273. [[CrossRef](#)]
25. Wang, L.L.; Lo, K.; Chandrasekhar, Y.; Reas, R.; Yang, J.; Eide, D.; Funk, K.; Kinney, R.M.; Liu, Z.; Merrill, W.; et al. COVID-19: The Covid-19 Open Research Dataset. *arXiv* **2020**, arXiv:2004.10706v2.
26. Liu, Q.; Liu, W.; Sha, D.; Kumar, S.; Chang, E.; Arora, V.; Lan, H.; Li, Y.; Wang, Z.; Zhang, Y.; et al. An environmental data collection for COVID-19 pandemic research. *Data* **2020**, *5*, 68. [[CrossRef](#)]
27. Marivate, V.; Combrink, H.M. Use of available data to inform the COVID-19 outbreak in South Africa: A case study. *arXiv preprint* **2020**, arXiv:2004.04813. [[CrossRef](#)]
28. Wang, L.; Li, R.; Zhu, J.; Bai, G.; Wang, H. When the Open Source Community Meets COVID-19: Characterizing COVID-19 themed GitHub Repositories. *arXiv preprint* **2020**, arXiv:2010.12218.
29. Singleton, A.D.; Spielman, S.; Brunson, C. Establishing a framework for Open Geographic Information science. *Int. J. Geogr. Inf. Sci.* **2016**, *30*, 1507–1521. [[CrossRef](#)]
30. Dabbish, L.; Stuart, C.; Tsay, J.; Herbsleb, J. Social Coding in GitHub: Transparency and Collaboration in an Open Software Repository. In Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work, New York, NY, USA, 11–15 February 2012.
31. Zagalsky, A.; Feliciano, J.; Storey, M.-A.; Zhao, Y.; Wang, W. The emergence of github as a collaborative platform for education. In Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing, Vancouver, BC, Canada, 14–18 March 2015.
32. Kemper, C.; Oxley, I. *Foundation Version Control for Web Developers*; APress: New York, NY, USA, 2012.
33. Erenkrantz, J.R. Release management within open source projects. In Proceedings of the 3rd. Workshop on Open Source Software Engineering, IEEE Computer Society, Portland, OR, USA, 3–10 May 2003.
34. Stol, K.-J.; Babar, M.A. Challenges in using open source software in product development: A review of the literature. In Proceedings of the 3rd International Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development, Cape Town, South Africa, 8 May 2010.
35. Ankolekar, A.; Herbsleb, J.D.; Sycara, K. Addressing challenges to open source collaboration with the semantic web. In Proceedings of the 3rd Workshop on Open Source Software Engineering, the 25th International Conference on Software Engineering (ICSE), IEEE Computer Society, Portland, OR, USA, 3–10 May 2003.
36. Scacchi, W. Understanding the requirements for developing open source software systems. *IEEE Proc.-Softw.* **2002**, *149*, 24–39. [[CrossRef](#)]
37. Mockus, A.; RFielding, T.; Herbsleb, J. A case study of open source software development: The Apache server. In Proceedings of the 22nd International Conference on Software Engineering, Limerick, Ireland, 4–11 June 2000.
38. Kon, F.; Meirelles, P.; Lago, N.; Terceiro, A.; Chavez, C.; Mendonça, M. Free and open source software development and research: Opportunities for software engineering. In Proceedings of the 2011 25th Brazilian Symposium on Software Engineering, Sao Paulo, Brazil, 28–30 September 2011.
39. German, D.M. The GNOME project: A case study of open source, global software development. *Softw. Process Improv. Pract.* **2003**, *8*, 201–215. [[CrossRef](#)]
40. Dinh-Trong, T.; Bieman, J.M. Open source software development: A case study of FreeBSD. In Proceedings of the 10th International Symposium on Software Metrics, Chicago, IL, USA, 11–17 September 2004.
41. Mitsova, H.; Neteler, M. Freedom in geoinformation science and software development: A GRASS GIS contribution. In Proceedings of the Open Source Free Software GIS-GRASS Users Conference, Trento, Italy, 11–13 September 2002.
42. Allan, J.; Carbonell, J.G.; Doddington, G.; Yamron, J.; Yang, Y. *Topic Detection and Tracking Pilot Study Final Report*; Carnegie Mellon University: Pittsburgh, PA, USA, 1998.
43. Yu, M.; Bambacus, M.; Cervone, G.; Clarke, K.; Duffy, D.; Huang, Q.; Li, J.; Li, W.; Li, Z.; Liu, Q.; et al. Spatiotemporal event detection: A review. *Int. J. Digit. Earth* **2020**, *13*, 1339–1365. [[CrossRef](#)]
44. Tryfona, N.; Price, R.; Jensen, C.S. Chapter 3: Conceptual Models for Spatio-temporal Applications. In *Spatio-Temporal Databases: The CHOROSCHRONOS Approach*; Sellis, T.K., Koubarakis, M., Frank, A., Grumbach, S., Güting, R.H., Jensen, C., Lorentzos, N.A., Manolopoulos, Y., Nardelli, E., Pernici, B., et al., Eds.; Springer: Berlin/Heidelberg, Germany, 2003; pp. 79–116.

45. Pfoser, D.; Tryfona, N. Requirements, Definitions, and Notations for Spatiotemporal Application Environments. In Proceedings of the 6th ACM International Symposium on Advances in Geographic Information Systems, New York, NY, USA, 2–7 November 1998.
46. Peuquet, D. Time in GIS and Geographical Databases. *Geogr. Inf. Syst.* **2005**, *1*, 91–103.
47. Yang, C.; Clarke, K.; Shekhar, S.; Tao, C.V. Big Spatiotemporal Data Analytics: A research and innovation frontier. *Int. J. Geogr. Inf. Sci.* **2020**, *34*, 1075–1088. [\[CrossRef\]](#)
48. Shekhar, S.; Jiang, Z.; Ali, R.Y.; Eftelioglu, E.; Tang, X.; Gunturi, V.M.V.; Zhou, X. Spatiotemporal Data Mining: A Computational Perspective. *ISPRS Int. J. Geo-Inf.* **2015**, *4*, 2306–2338. [\[CrossRef\]](#)
49. Fagan, M.E. Design and code inspections to reduce errors in program development. *IBM Syst. J.* **1999**, *38*, 258–287. [\[CrossRef\]](#)
50. Peng, R.D. Reproducible research in computational science. *Science* **2011**, *334*, 1226–1227. [\[CrossRef\]](#)
51. Tobler, W.R. A Computer Movie Simulating Urban Growth in the Detroit Region. *Econ. Geogr.* **1970**, *46*, 234–240. [\[CrossRef\]](#)
52. Benureau, F.C.Y.; Rougier, N.P. Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions. *Front. Neuroinform.* **2018**, *11*, 69. [\[CrossRef\]](#)
53. Liu, Q.; Harris, J.T.; Chiu, L.S.; Sun, D.; Houser, P.R.; Yu, M.; Duffy, D.Q.; Little, M.M.; Yang, C. Spatiotemporal impacts of COVID-19 on air pollution in California, USA. *Sci. Total Environ.* **2021**, *750*, 141592. [\[CrossRef\]](#)