Journal of Geovisualization and Spatial Analysis (2020) 4:17
https://doi.org/10.1007/s41651-020-00056-5

When Spatial Analytics Meets Cyberinfrastructure: an Interoperable and Replicable Platform for Online Spatial-Statistical-Visual Analytics

Hu Shao[1] · Wenwen Li[1] · Wei Kang[2] · Sergio J. Rey[2]

Abstract
Developing spatial analytical methods as open source libraries is an important endeavor to enable open and replicable science. However, despite the fact that large geospatial data and geospatial cyberinfrastructure (GeoCI) resources are becoming available, many libraries and toolkits are only initialized and designed for analytics in a desktop environment. Coupling spatial analytical functionality with big data and high-performance computing will result in immediate benefits for multidisciplinary research in terms of addressing challenging socioeconomic and environmental problems, as well as supporting remote collaboration between participants from physically distributed research groups, and assisting informed decision-making. In this article, we present the design and implementation of a general workflow to integrate state-of-the-art open source libraries with GeoCI resources. We also solve various interoperability and replicability issues that arise during the implementation process. The popular open source Python Spatial Analysis Library (PySAL) was selected to build the interoperable Web service, WebPySAL, which was then successfully integrated in GeoCI. With this integration between spatial analytics and cyberinfrastructure, the new GeoCI platform provides easy-to-use, efficient, and interactive exploratory spatial analysis functions to public users. The GeoCI capability is demonstrated through two regional economic case studies of (1) evaluating global spatial autocorrelation and identifying local clusters in the spatial pattern of median household incomes for US counties (with global and local Moran's I statistics) and (2) modeling the space-time dynamics of per capita incomes at the state level (with spatial Markov statistics).

Keywords Spatial analysis · Geoprocessing services · Cyberinfrastructure · Reproducibility · Provenance · PySAL

Introduction

Geographic information science (GIScience) has led to tremendous developments in new spatial analytical methods and new data-driven techniques for supporting geospatial problem solving. It has also contributed to multidisciplinary research by providing modern theories, methodologies, software, and tools to address pressing social and environmental problems, such as understanding human mobility, regional inequality, and global environmental change (Wang 2013; Li et al. 2016c; Li et al. 2019a). The recent open science movement also fosters the development of open source libraries and software toolkits for spatial analysis (Li et al. 2010; Steiniger and Hunter 2013; Swain et al. 2015), such as Python Spatial Analysis Library (PySAL) (Anselin and Rey 2014; Rey 2014; Rey et al. 2015), GeoDa (Anselin et al. 2010), GDAL (Geospatial Data Abstraction Library; Warmerdam 2008), GRASS (Geographic Resources Analysis Support System; Neteler et al. 2012), GIS, spacetime (Pebesma 2012), STARS (Space-Time Analysis of Regional Systems; Rey and Janikas 2010), and spdep (Bivand et al. 2011). These toolkits play a critical role in promoting continuous innovations in GIScience.

There are two main working modes within the geospatial community: the "single-user" mode and the "collaboration" mode. The former is most suited to individual researchers who possess professional domain knowledge. They generally conduct research and experiments from an exploratory perspective and in an "agile" manner. Since this working mode gives researchers absolute control over the data and analytical methods or software, it is very popular among individual researchers.

✉ Wenwen Li
wenwen@asu.edu

1 School of Geographical Sciences and Urban Planning, Arizona State University, Tempe, AZ 85287-5302, USA

2 Center for Geospatial Sciences, University of California Riverside, Riverside, CA 92521, USA

Published online: 30 June 2020

However, the single-user mode is infeasible in certain scenarios, including:

- Very large projects that require the collaboration among participants from different domains and physical locations. The data set, documents, and knowledge must be simultaneously shared among team members in an efficient way (Rinner et al. 2008; Sun and Li 2016).
- Time-critical and data-intensive scenarios (e.g., when a natural disaster occurs), where massive data sets, including basic terrain, hydrology, transportation data, and real-time observation data, need to be gathered for on-the-fly spatial analysis. The results should be delivered to decision-makers in real time to ensure that rapid responses and evacuation plans can be developed (Li et al. 2015b; Wu et al. 2013).
- Mobile working projects or field investigations, where the architecture of the system could be distributed. The server side should be responsible for data storage and computation, while the tasks on the client side in relation to mobile phones and tablets should involve data collection and visualization (Cerón et al. 2018).
- Demonstration and education scenarios (e.g., dashboard system cases), where livestream data is visualized and data patterns are displayed, or where the usage of complicated data sets or newly developed data analysis methods are taught through demonstrations (Harris 2003; Veenendaal 2015).

The rapid development of geospatial technologies has enabled scientists to gather large amounts of high-quality georeferenced data from the physical world, society, the economy, social media, etc. Data deluges such as these provide GIScience researchers with opportunities to obtain deeper insights into the phenomena occurring in nature and society. Consequently, the acceleration in the development and achievement of theories, methods, software, and discoveries is driven by the richness of the data of recent decades.

In addition to the big data deluge, cyberinfrastructure (CI) theories and technologies have undergone extensive development, and numerous commercial and academic products and platforms have been widely adopted, including Amazon Cloud, Microsoft Azure, Google Earth Engine, Hadoop, and Apache Spark. These CI facilities are capable of hosting big data sets and conducting large-scale analyses and simulations that are infeasible on individual desktops. All of these factors combine to render the second collaborative working mode increasingly popular (Rinner et al. 2008). In the geospatial domain, Earth and environment science is a discipline that has witnessed a widespread adoption of cyberinfrastructure to support scalable and online data analysis and decision-making. Exemplar projects include DataOne (Allard 2012), OpenTopography (Krishnan et al. 2011), GEON (GEOscience network; Keller 2003), PolarHub (Li 2018), and Ocean Observatories Initiative (OOI; Rodero Castro and Parashar 2016). They each provide a cyberinfrastructure solution to increase open sharing, effective search, and analytics of environmental, atmospheric, or geological data. In the social sciences, however, such efforts are sparse. But social scientists do have access to data and knowledge that have great practice value, and these data and the sharing of such data will complete the picture of a "knowledge society" that (physical) scientists are exploiting quite lonely in. It was also argued by Unsworth (2008) that the social sciences need more collaborative computational methods and decision-making. What is more, producing replicable research has become an increasing concern across all science disciplines. Replicability ensures that a research can be validated, the data and methods can be easily reused, and science can be more rapidly advanced. The work we present in this paper aims right at addressing the above challenges by building a cyberinfrastructure for social sciences and making spatial-social analytics openly available as well as replicable.

The remainder of the paper is organized as follows: Section 2 introduces related research in developing CI to support interoperable research. Section 3 describes the integration strategy and implementation details of the proposed GeoCI platform. Section 4 uses two use cases to demonstrate the power of the CI in performing spatial social analysis interactively and efficiently. This section also presents a comparison of the performance of spatial analyses in the desktop environment with the proposed cyber-environment. We conclude our work with potential areas of future research in Section 5.

Literature Review

Coupling spatial analysis models with CI resources to support collaborative research under the Web environment could accelerate the solution process of complex spatial problems and support effective decision-making.

A number of related research studies and practices have been conducted recently with different areas of focus (Anselin and Rey 2012). Some of these studies and practices are dedicated to deploying sophisticated spatial analysis models in a high-performance computing (HPC) environment to solve data-intensive research questions related to hydrology (Rajib et al. 2016), ecology (Dubois et al. 2013; Sugumaran et al. 2009), and the environment (Delipetrev et al. 2014; Swain et al. 2015). Others have focused on technical solutions, such as the design and implementation of a CI working environment to handle and manipulate big geospatial data and conduct analyses and simulations (Astsatryan et al. 2015; Mihon et al. 2013; Wang and Liu 2009), or develop a computing capacity to parallelize the geoprocessing in a HPC environment (Laura et al. 2015; Wang et al. 2009; Wang and Armstrong 2009).

However, challenges remain to enable a smart CI that allows online collaborative analysis of spatial social data. Several of the main challenges are:

1. The lack of interoperability between software components: The deployed toolkits should be compatible with popular open standards and be easily exploited by users in a Web-based environment.
2. The lack of provenance and metadata to capture automatically the spatial analytical workflows: This lack could be one of the most critical factors under the collaborative and reproducible working mode. It refers to all of the information involved, ranging from how spatial data is produced and how geoprocessing steps are chained and conducted, to how to obtain results. This is key for maintaining quality control and reproducing a spatial analytical process (Anselin et al. 2014; Wu et al. 2015).
3. The granularity of the functionalities and their exposure as application programming interfaces (APIs): Many open source libraries are designed for the single-user working mode, where the functionalities of each method and class are usually designed to be atomic, thus helping users to combine various methods in a flexible manner in order to conduct the exploratory analysis. However, when deploying the functions on the server side, the communication cost between the client and server needs to be considered. The most intuitive way to reduce the communication cost is to combine atomic APIs with non-atomic APIs. This results in a sophisticated workflow, as several parameter inputs from users can be accepted all at once. This can be exemplified in the inference of local indicators of spatial association (LISAs) (Anselin 1995).
4. The provision of proper documentation and supporting materials: Many open source projects serve as pioneers in implementing and introducing newly developed methodologies of spatial analysis. When deploying these methodologies, the provision of adequate documentation and materials to educate users to appropriately use the APIs should be carefully considered.

Initiated in 2012, the NSF-funded GeoCI platform serves as a one-stop portal for interoperating and visualizing distributed geospatial data resources (Li et al. 2016a, 2019a). A spatial data search engine is integrated into GeoCI, thus enabling it to discover an extensive amount of open geospatial data shared on the internet (Li 2018). Rich data visualization and exploration functions have also been integrated into the GeoCI. Furthermore, this portal employs cutting-edge techniques for the efficient representation, management, and transfer of geospatial big data (Li et al. 2016c; Song et al. 2016; Shao and Li 2018). In this article, we will use GeoCI platform as the test bed to demonstrate its ability to support social science research.

We first enable Web-based spatial analytics by deploying PySAL library as WebPySAL based on a widely adopted Web service standard. These Web services are then seamlessly integrated into the GeoCI portal so that advanced spatial analysis functions can interoperate with the various spatial and temporal data for an integrated study. The deployment strategy and architecture of various components make our system highly interoperable from a user's perspective and extensible from a developer's perspective.

Methodologies and System Implementation

The Interoperable Architecture of WebPySAL

Our implementation selected PySAL, a python-based open source library for spatial analysis and statistics (Rey and Anselin 2010). This library is based on modularized design and contains modules for point pattern analysis (*pointpats*), spatial distribution dynamics analysis (*giddy*), map classification (*mapclassify*), and exploratory spatial data analysis (*esda*), among others. All these core modules, and their class and function definition, as well as relevant metadata, configuration file, and documentation, are automatically fetched using the *sphinx* library (Brandl 2009) to support the deployment of the functions into a series of Web services; we named it WebPySAL, to enable geospatial interoperability across different platforms and programming languages (Fig. 1).

In WebPySAL, the OGC (Open Geospatial Consortium) WPS (Web Processing Service) standards are adopted for providing geoprocessing services. A python-based open source library, PyWPS (Čepický 2007), is employed for platform development. PyWPS provides OGC WPS implementation on the server side and supports easy wrapping of spatial analytical functions into WPS. In our implementation, PyWPS acts as the middleware for transforming the functionalities from PySAL into WebPySAL. Each spatial analysis functionality is wrapped in an individual class, with predefined inputs and outputs, according to the rules of PyWPS. PyWPS will then publish these functionalities as OGC-compliant WPS services.

The following WPS operations are enabled for WebPySAL analytics: (1) GetCapabilities: provides xml files that hold metadata about the service, including service provider information and available processes; (2) DescribeProcess: provides a detailed description of the supported processes, their parameter settings, and input/output formats; (3) execute: allows users to invoke operations by providing proper data and input/output variables through a Web request. A WebPySAL execution operation is capable of accepting a

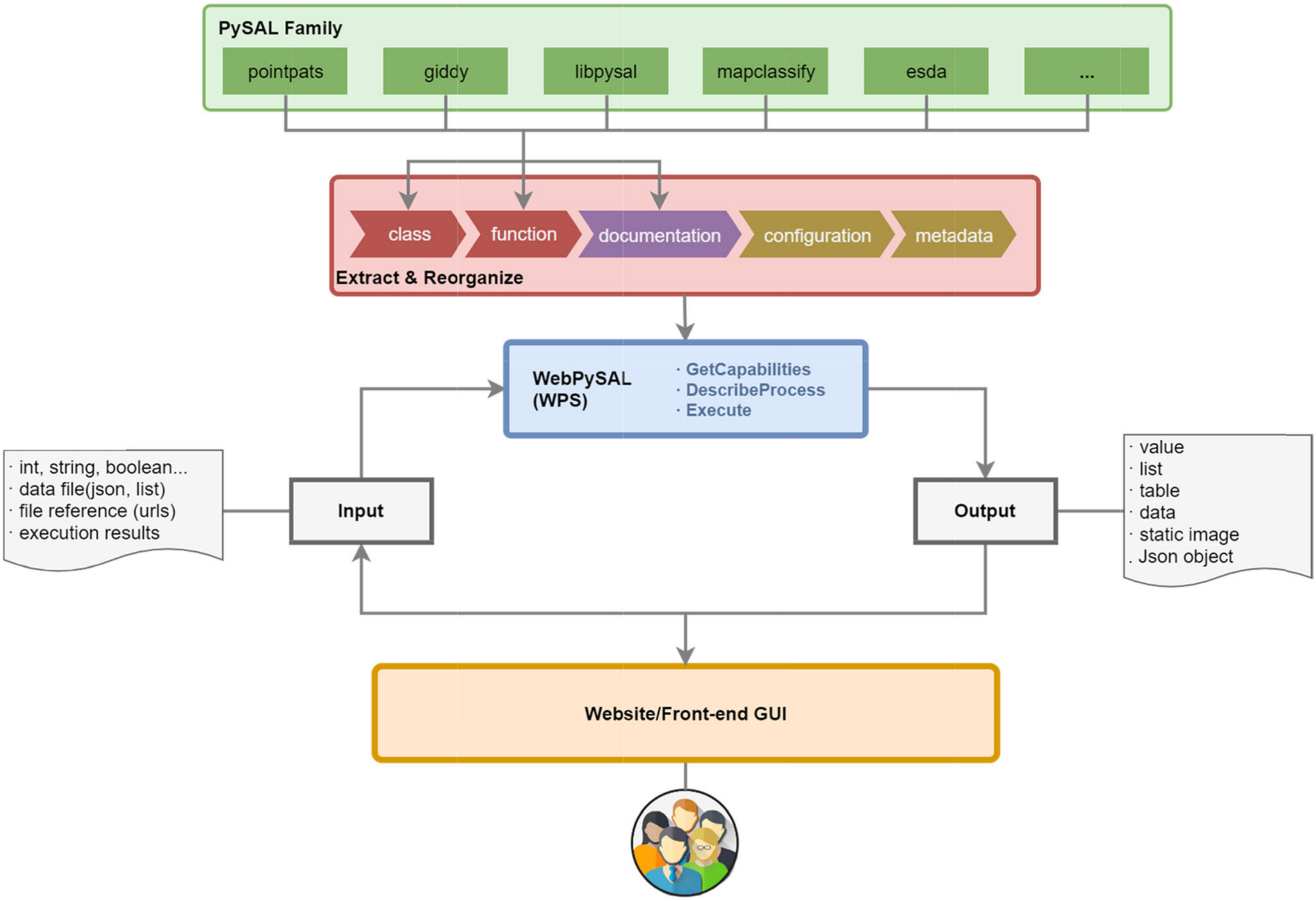


Fig. 1 The architecture of WebPySAL

wide range of inputs, including (1) *literal data*, such as numbers, strings, and Booleans; (2) *complex data*, such as GML, JSON, and text files; (3) *file references*, such as URLs (for geoprocessing, the system will automatically fetch the data set according to the URLs on the server side); and (4) *results/outputs of other operations*. This provides users with the flexibility to chain multiple operations together to generate and execute a complex geoprocessing task.

Provenance and Metadata

Tracking the provenance of operations is a key factor to ensure the full replicability of data analysis as well as the interoperability with other systems. These are critical factors in the current and increasingly popular context of collaboration (Anselin et al. 2014). Two strategies for maintaining provenance are adopted in WebPySAL's implementation: software versioning and metadata for executing geoprocesses.

In WebPySAL, since each geoprocessing API wraps some specific functionalities from the submodules of PySAL, the development version of these submodules will be automatically extracted and used by WebPySAL. In terms of the open source libraries that are developed and upgraded rapidly, this strategy can help users obtain a better sense of which library version they are using and whether or not they can obtain results that are identical to older versions.

The execution metadata of WebPySAL contains all of the essential parameters needed to execute the API. After specific parameters and configurations are provided by an end user, they will be injected into the execution request form and submitted to the server side to initialize the analytic process. These forms are in XML (eXtensible Markup Language) format and designed to be readable by both humans and machines. In this way, the provenance of a workflow can be properly captured so that users can replicate the process to obtain identical results at a later time.

We will take the spatial weight construction, which is essential to many spatial analytical tasks, as an example. Spatial weights, often in the format of a matrix, provide the spatial relationships among spatial entities. The higher the weight value, the stronger the spatial relation. Table 1 shows the metadata description for the WebPySAL Web service API which provides KNN (K-nearest neighbor)-

Table 1 Metadata description for the process of KNN spatial weight

```xml
<wps:ProcessDescriptions … service="WPS" version="1.0.0" xml:lang="en-US">
 <ProcessDescription wps:processVersion="3.0.5" storeSupported="true" statusSupported="true">
  <ows:Identifier>libpysal:KNN</ows:Identifier>
  <ows:Title>K Nearest Neighbor Weights Calculation</ows:Title>
  <ows:Abstract>Calculate the KNN weights object from a collection of geometries. Classes and functions used in
this API include libpysal.weights.Distance.KNN. For more information, see the metadata</ows:Abstract>
  <ows:Metadata xlink:title="KNN"
xlink:href="https://github.com/pysal/libpysal/blob/master/libpysal/weights/Distance.py" xlink:type="class"/>
  <DataInputs>
   <Input minOccurs="1" maxOccurs="1">
    <ows:Identifier>data</ows:Identifier>
    <ows:Title>Data</ows:Title>
    <ComplexData>
     <Default>
      <Format><MimeType>application/vnd.geo+json</MimeType></Format>
     </Default>
    </ComplexData>
   </Input>
   <Input minOccurs="0" maxOccurs="1">
    <ows:Identifier>k</ows:Identifier>
    <ows:Title>Number of nearest neighbors</ows:Title>
    <ows:Abstract>Number of nearest neighbors for querying, default value is 2</ows:Abstract>
    <LiteralData>
     <ows:DataType ows:reference="urn:ogc:def:dataType:OGC:1.1:integer">integer</ows:DataType>
     <ows:AnyValue/>
    </LiteralData>
   </Input>
   <Input minOccurs="0" maxOccurs="1">
    <ows:Identifier>idVariable</ows:Identifier>
    <ows:Title>Id Variable</ows:Title>
    <ows:Abstract>The name of the column to use as IDs. If nothing is provided, the dataframe index is used.
(Note: the ids should be unique and Integer type is preferred.)</ows:Abstract>
    <LiteralData>
     <ows:DataType ows:reference="urn:ogc:def:dataType:OGC:1.1:string">string</ows:DataType>
     <ows:AnyValue/>
    </LiteralData>
   </Input>
  </DataInputs>
  <ProcessOutputs>
   <Output>
    <ows:Identifier>weights</ows:Identifier>
    <ows:Title>Result Bundle</ows:Title>
    <ows:Abstract>The calculated weights by using this method.</ows:Abstract>
    <ComplexOutput>
     <Default>
      <Format><MimeType>application/json</MimeType></Format>
     </Default>
    </ComplexOutput>
   </Output>
  </ProcessOutputs>
 </ProcessDescription>
</wps:ProcessDescriptions>
```

based spatial weight operation. The required input parameters include “Data” as well as 2 optional input parameters: “Number of nearest neighbors” and “Id Variable.” According to this metadata, an “execution” operation can be invoked to create the weight matrix based on user input. The execution metadata will be saved on the server side to ensure that the analysis can be reproduced by the same or different researchers.

Abstraction and Aggregation of PySAL Functions to Provide Synthetic APIs

PySAL was originally designed for a desktop working environment. During its implementation, the object-oriented strategy was adopted, meaning that class objects are widely used for hosting analysis functions and relevant variables. When users are exploring the library within the desktop environment (e.g., in a Jupyter Notebook), intermediate results, such as class instances and variables, can be easily stored and reused for the next-step analysis. Consequently, it is appropriate to make each method atomic so that it is only responsible for performing a single task; this enables end users to flexibly combine different methods in the exploratory analysis. WebPySAL will be mainly used in the Web environment; interacting and transforming data between the server and client will be more time-consuming than in a local environment, and this will also burden the user interface design on the client's side. Thus, for WebPySAL, the data transmitted between the server and client via a network should not be too fragmented, and the interaction should not be too frequent. During the implementation of WebPySAL, we adopt a "synthetical" strategy that enables each WebPySAL API to take combined input parameters, conduct the entire geoprocessing workflow, and return the completed results. Specifically, semantic-based workflow generation (Li et al. 2019b) is enabled in our collaborative spatial-statistical analysis environment. In this semantic framework, ontological components including theme ontology, spatial operation ontology, and spatial data ontology are developed. The spatial operation ontology describes the input, output and the function of a spatial operation. The spatial data ontology provides a hierarchical and semantic classification of spatial data, which serves as the input or the output of a spatial operation. The theme ontology defines workflow templates for different use cases. It also includes a spatial theme ontology that classifies locations depicting municipal boundaries for place name disambiguation in a spatial query. In addition, service chaining rules are defined to plug data and operation services into the workflow template automatically. The chained result is a synthetic workflow metadata which is compliant with OGC WPS standard and executable using a WPS engine, such as GeoServer. This metadata also serves as the provenance of the analytical workflow as well as the required input of the synthetic APIs.

The results returned from the APIs can then be used for visualization and interpretation on the client's side.

Figure 2 illustrates the different ways of interacting between PySAL and WebPySAL when calculating Moran's I statistic, which is an important measure for spatial autocorrelation of a data set. In the desktop environment, the user needs to invoke 3 functions sequentially in order to read the geospatial file, generate a weight matrix, and initialize Moran's I statistic. Different parameters should be separately provided to these functions during the process. Results concerning Moran's I are assigned to the Moran object as attributes. All of the intermediate results are temporarily stored in the local computer's memory for quick access. Under the Web environment, the "synthetical" API takes the inputs of all the parameters needed to produce the final results. After the parameter inputs are submitted through the execution request form to the server's side, they will be assigned to their respective atomic functions in order to execute the processing chain. When the process is finished, the resulting attributes will be extracted and injected into the result bundle and returned to the user or stored on the server side as files for future access. After the results are returned, the memory for preserving the intermediate results will be freed on the server side. From Fig. 2, we can see that the user only needs to interact with WebPySAL once to invoke the API, and they still have the flexibility of providing different parameter values.

Integration of WebPySAL into GeoCI

Currently, a large number of organizations are collecting and sharing geospatial data sets on the internet through OGC's WFS and WMS (Web Map Service) standards for both public and scientific use. In our previous work, we developed a geospatial data discovery engine named PolarHub (Li et al. 2016b), which is capable of collecting metadata information from hundreds of thousands of geospatial data sets. The metadata information is stored in a relational database and integrated into GeoCI's system (Li et al. 2019b). A geospatial data search engine is implemented in GeoCI to help users conveniently find their desired data sets by using keywords and/or spatial extent filtering (Li et al. 2015a). The selected data sets can be easily included in GeoCI under the user's account for visualization and analysis.

WebPySAL's geoprocessing APIs have been fully integrated into GeoCI. Specific exploratory data analysis modules are designed and implemented to help users take advantage of WebPySAL's spatial analysis models and functionalities. The architecture of the integrated systems is presented in Fig. 3.

A typical workflow for a user in GeoCI is as follows: First, a user browses the data sets available in GeoCI and selects the suitable ones. Next, the user can select a spatial analysis module in GeoCI and provide the spatial data set and other input

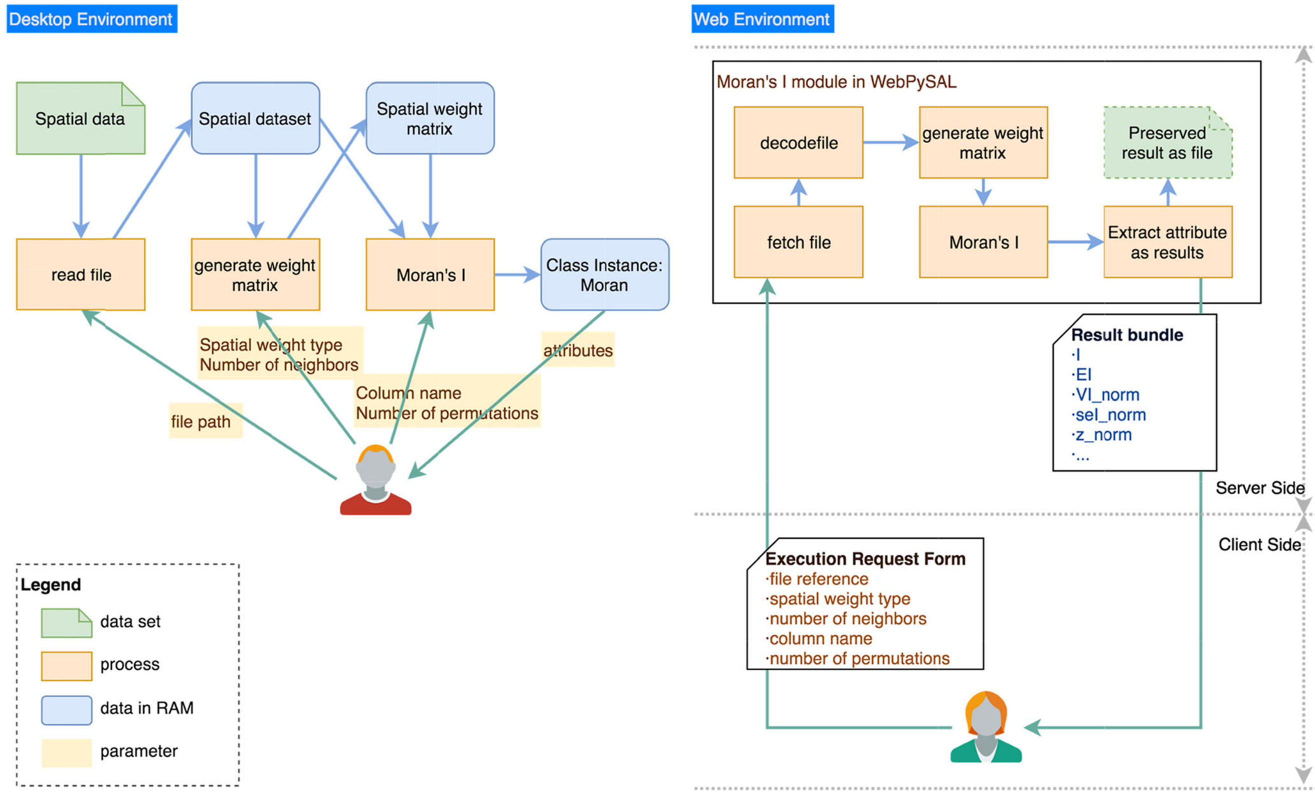


Fig. 2 A comparison of the interaction modes with PySAL and WebPySAL in a desktop environment vs. Web environment

parameters in a graphic user interface. Once the request is submitted, the geoprocessing APIs in WebPySAL will be invoked and the results will be returned. Last, the GeoCI will parse the resultant data, then visualize and demonstrate the results through interactive maps, graphics, and provenance information.

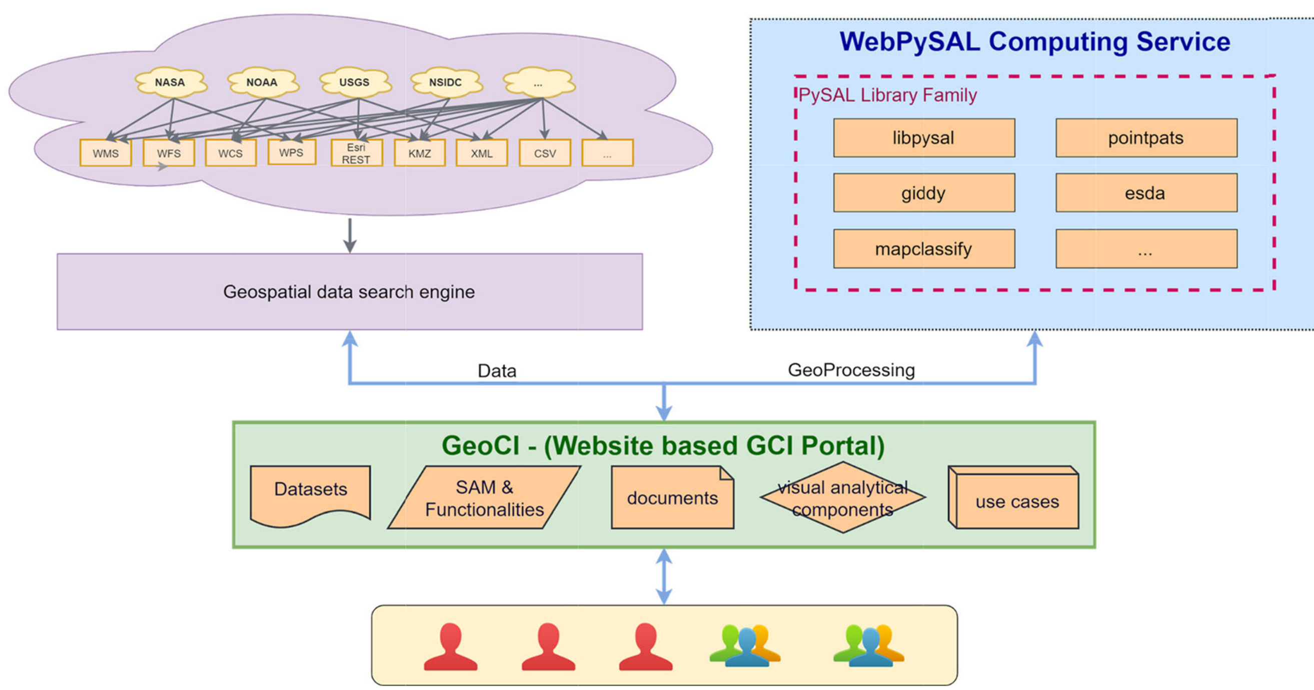


Fig. 3 The architecture of GeoCI

Illustration and Experiments on Spatial and Spatiotemporal Statistics

In this section, we use two use cases to illustrate how GeoCI can help users fulfill spatial analytical tasks with visual aids in a convenient, efficient, and replicable fashion. The case in section "Global and Local Indicators of Spatial Association" represents an important way in exploring global and local spatial patterns of lattice data at a given time point. The spatial statistics methods—global and local Moran's I statistics—are integrated into GeoCI to investigate the global pattern and the heterogeneity of spatial dependence. The case in section "Spatial Markov Tests" explores the role of space in shaping the evolution of a variable (i.e., per capita income) over time. We demonstrate the ability of GeoCI in performing spatiotemporal statistical analysis by integrating spatial Markov method to reveal how the transition of states (i.e., the status of being poor or rich) is dependent on location and status of spatial neighbors. Both methods are widely used in regional economic studies and social science applications.

Global and Local Indicators of Spatial Association

Global and local indicators of spatial association are the most important tools for exploring the spatial distribution of a given variable at a given time point. Both pertain to the question of spatial randomness by examining whether or to what degree location similarity and attribute similarity coincide. While global indicators operate at the global level, meaning that a single summary statistic is produced, local indicators operate at the local level by decomposing the global indicators to provide insights into local patterns, such as hot and cold spots, as well as the instability of spatial associations (Anselin 1995).

The PySAL submodule *esda* implements a wide array of global indicators including Moran's I, Geary's C, Getis-Ord G, and joint count statistics. All of these global indicators have also been integrated in WebPySAL and GeoCI. Here, we detail the usage of Moran's I and local Moran's I as an illustration: These are the most widely used in empirical settings. Given n spatial observations with attribute Y, the global indicator of spatial association, Moran's I (Cliff and Ord 1981), can be defined in Eq. (1):

$$I = \frac{n}{S_0}\frac{\sum_{i=1}^{n}\sum_{j=1}^{n} Z_i W_{i,j} Z_j}{\sum_{i=1}^{n} Z_i Z_i} \tag{1}$$

where $z_i = y_i - \overline{y}$ is the deviation from the global mean, and $\boldsymbol{W}$ is the (n,n) spatial weight matrix formalizing the spatial relationship between any pair of spatial units:

$$S_0 = \sum_{i=1}^{n}\sum_{j=1}^{n} W_{i,j}$$

Inference could be made under the normality assumption or based on random permutations. For the proper estimation and inference of this statistic, the user must supply the attribute, spatial weight matrix, and number of permutations if randomization-based inference is desired. We shall see that WebPySAL provides options for setting these parameters in a convenient fashion.

Local Moran's I is a spatial decomposition of Moran's I, which has a value for each spatial unit. This is shown in Eq. (2). As suggested by Anselin (1995), a pseudo p value could be obtained for I_i, based on conditional randomization. The required parameters are similar to the global indicator.

$$I_i = \frac{(n-1) Z_i \sum_{j=1}^{n} W_{i,j} Z_j}{\sum_{j=1}^{n} Z_j^2} \tag{2}$$

Data

We applied the global and local indicators of spatial association to the median household incomes of 3219 US counties in 2016. The county boundaries were acquired from the US Census Bureau's TIGER (Topologically Integrated Geographic Encoding and Referencing) geographic database and the "Unemployment and Median Household Income for the U.S., States, and Counties, 2007–17." The county-level median household incomes in 2016 were downloaded from the website of the US Department of Agriculture. These 2 data sets are spatially joined and hosted on our test bed as a standard WFS data service for public use. The spatial distribution of the median household incomes can be conveniently visualized in GeoCI, as shown in Fig. 4. It can be seen that similar values tend to be geographically close to each other.

Empirical Results and Visualization

Global and local Moran's I statistics are applied to the US county-level median household incomes of 2016 to explore the spatial distribution, or more specifically, whether the observed incomes are randomly distributed spatially and whether there are hot spots of high incomes or cold spots of low incomes. We begin with global Moran's I. As displayed in Fig. 5a, GeoCI provides a GUI for selecting values for all of the relevant parameters. There are 2 ways to specify the spatial weight matrix $\boldsymbol{W}$: Choose a weight type (e.g., queen/rook contiguity, KNN) so that a spatial weight matrix is constructed for the geometries using functions in *libpysal*, or supply a spatial weight file. Users also have the option to leave the weight type parameter blank so that the default value is used. This creates a row-normalized rook contiguity weight matrix, where spatial units that share an edge are considered neighbors. Here, we use the default value for the spatial weight; 999

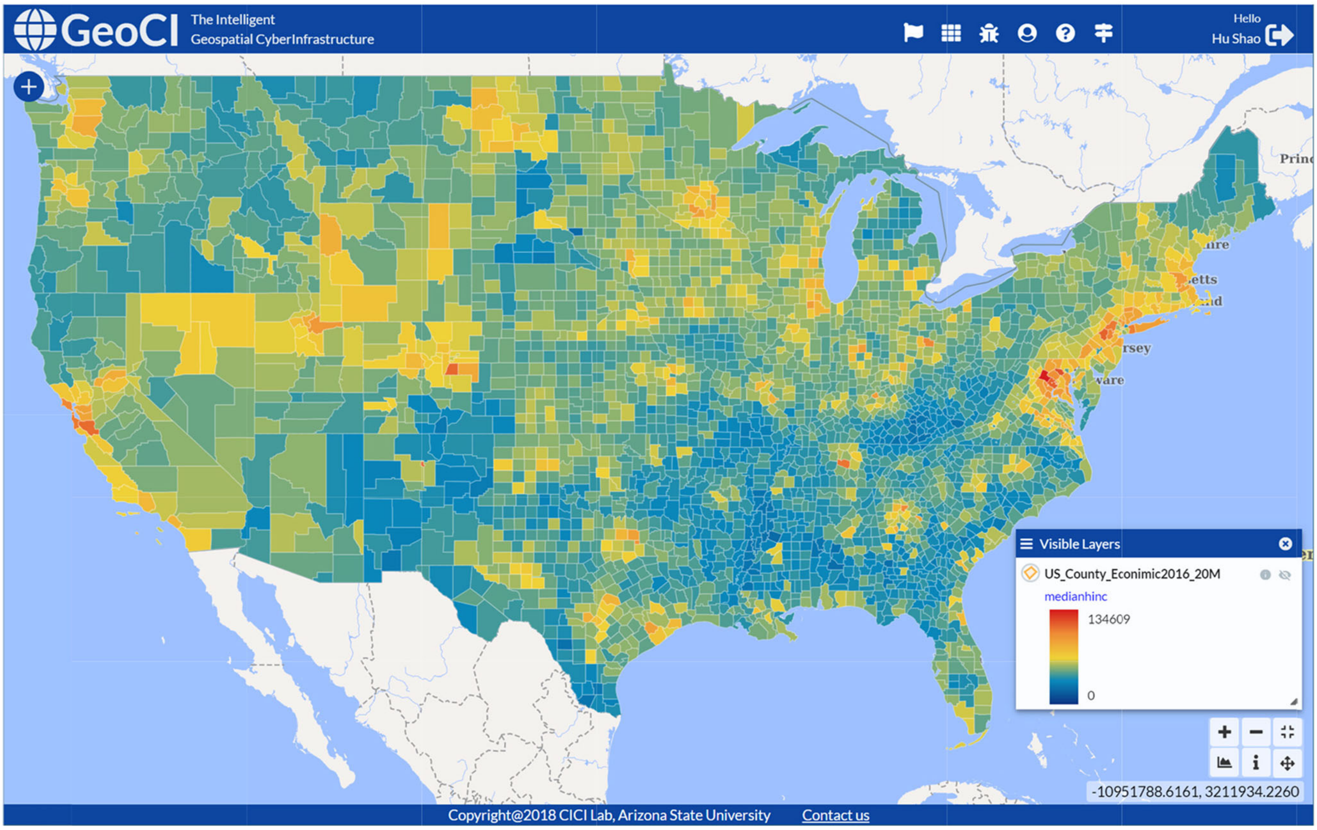
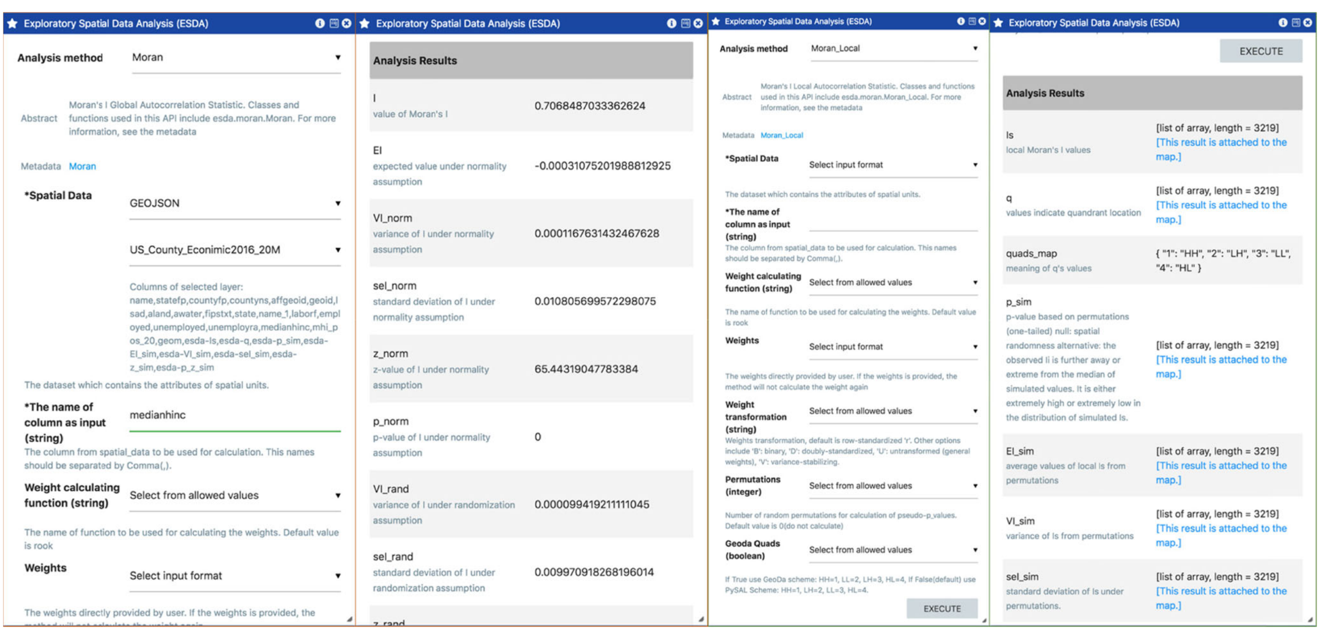


Fig. 4 Map of county-level median household incomes of the USA in 2016

permutations are selected for randomization-based inference. The same values are selected for the inference of local Moran's *I*, as shown in Fig. 5b.

The results of Moran's *I* will be appended to the analysis method window once the calculation is complete (Fig. 5a). The visual impression of spatial clustering of similar values



(a) Moran's I window (b) Local Moran's I window

Fig. 5 Moran's *I* and local Moran's *I* in WebPySAL and GeoCI. **a** Moran's *I* window. **b** Local Moran's *I* window is confirmed by the positive and significant Moran's *I* estimate of 0.707, with a *p* value of 0 (rounded) under the normality assumption, and pseudo *p* value of 0 (rounded) based on the 999 spatial permutations.

Having established that county-level median household incomes are spatially dependent from a global perspective, we now turn to local Moran's *I* statistics, the local decomposition of global Moran's *I*, to investigate potential heterogeneity of spatial dependence. Compared with global Moran's *I* which produces a series of single values for the whole spatial distribution, local Moran's *I* produces a series of single values for each spatial unit (US county here). Therefore, for each output statistic from applying local Moran's *I*, we will have a list of *n* values, each corresponding to a county. As shown in Fig. 5b, the output attributes, such as "Is" (local Moran's *I* estimates), "q" (quadrant location in the Moran's *I* scatterplot), "p_sim" (*p* values based on permutations), and "z_sim" (standardized local Moran's *I* based on permutations), comprise of *n* values each. We could easily join any of these outputs with the county boundaries to facilitate geovisualization in GeoCI. Figure 6 visualizes spatial distributions of two output variables: "q" on the left and "z_sim" on the right. The map on the left uses four colors to indicate whether a county's and the average of its neighbors' median household incomes are higher or lower than national average. For instance, red color represents "HH" (High-High), indicating that a high-income county is joint with its neighbor(s) with high incomes. Comparatively, dark blue represents "LL" (Low-Low), indicating that both the county and its neighbor counties have low income values. It is visually obvious that hotspots ("HH") are clustered in the southwest coast and northeast coast while the coldspots ("LL") are clustered in the southeast. The map on the right displays standardized local Moran's *I* estimates, which could be used for determining whether the estimate is statistically significant. Usually, adopting 5% significance level means that estimates out of the range [– 1.96, 1.96] are statistically significant. Aside from these two output variables, we could conveniently visualize other output variables of interests in GeoCI.

Spatial Markov Tests

The first-order discrete Markov chains model is a widely used stochastic model in which the current status is only dependent on its status in the immediately preceding time period. It has been widely applied to provide insights into the underlying temporal dynamics of land use, land cover change, crime patterns, and income distribution (McMillen and McDonald 1991; Quah 1993; Rey et al. 2014). By further assuming time homogeneity, the transitional dynamics for the time span of the entire study could be summarized in a (k, k) stochastic matrix $\boldsymbol{P}$, in which each element $p_{i,j}$ presents the probability of transitioning from state i to j over 2 consecutive time periods. The maximum likelihood estimator $\hat{p}_{ij}$ is displayed in Eq. (3):

$$\hat{p}_{ij} = \frac{n_{ij}}{\sum_{j=1}^{k} n_{ij}}, \tag{3}$$

where $n_{i,j}$ is the number of transitions from state i to j across two consecutive time periods. The conventional application of the Markov chains model to a spatial setting assumes that the dynamics are identical across all spatial units. Thus, $\boldsymbol{P}$ is estimated from the pooled data. However, the ignorance of space in shaping the dynamics could lead to false conclusions. Spatial Markov tests, which test for spatial dependence in the discrete Markov chains framework, have been developed, and their properties have been evaluated for the study of regional income distribution dynamics (Rey et al. 2016; Kang and

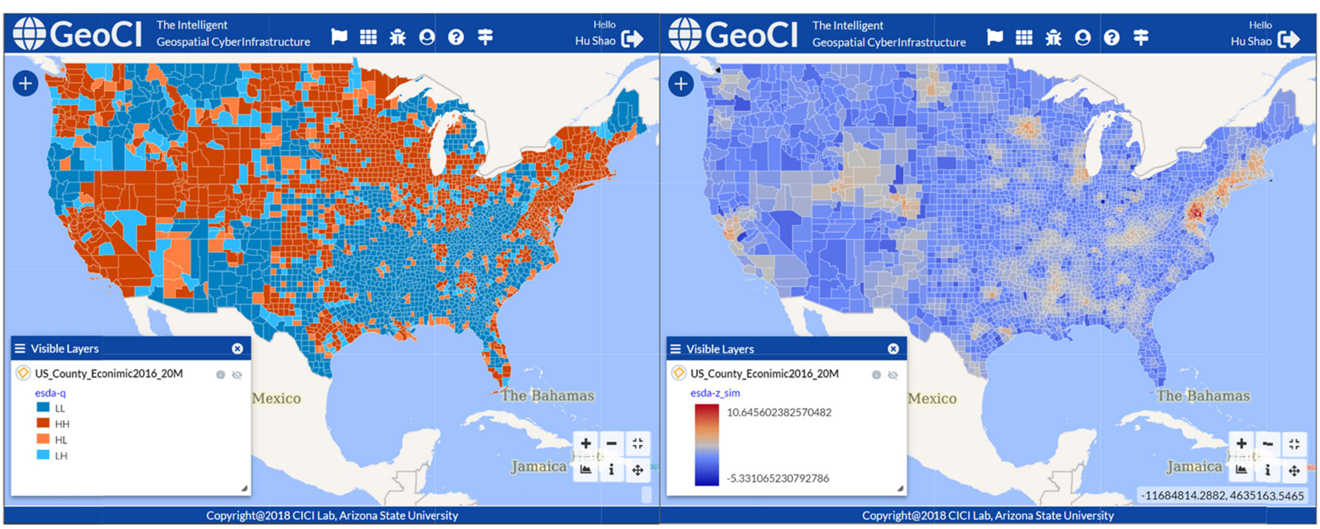


Fig. 6 Visualization of output variables for local Moran's *I* in GeoCI

Rey 2018). The alternative to spatial Markov tests contends that the underlying dynamics are too complex to be summarized in a single transition probability matrix. Rather, the transition probability is context sensitive in that it is also dependent on the current status of its neighbors. The so-called spatial lag, which is the weighted average of neighbors' values (e.g., income) shown in Eq. (4), is usually used to quantify the status of neighbors:

$$\boldsymbol{Z}_t = \boldsymbol{W}\boldsymbol{y}_t, \tag{4}$$

where z_t is the n-dimensional spatial lag at t. Following the similar discretization strategy to the original time series, the time series of spatial lags could also be discretized into k categories, on which transition probabilities are conditional, resulting in k spatially dependent transition probability matrices. The likelihood ratio (LR), x^2, and Kullback information-based (Kullback et al. 1962) tests can be calculated by comparing them with the single matrix estimated from the pooled data.

Data

The average per capita incomes for the lower 48 US states from 1929 to 2009 are used for demonstration. The data set was acquired from Bureau of Economic Analysis in the US Department of Commerce. The state-level cartographic boundary data were downloaded from the US Census Bureau's geographic database. These 2 data sets are bound together and hosted on our test bed as a standard WFS data service. Note that the GeoCI portal is not limited to data service on a local server; any remote server which provides such data through standardized service can be integrated into the portal in the same way. The map of per capita incomes of each US state in 2009 can be easily visualized in GeoCI. We can also interactively explore the time dimension with the help of the time series plot, as shown in Fig. 7 (left figure). As the user moves the vertical dotted line in the time series plot, the map on the right will be updated to the chosen year (e.g., 1973), and the colors of the time series plot will be updated to match the color scheme of the map. In addition, we can select a US

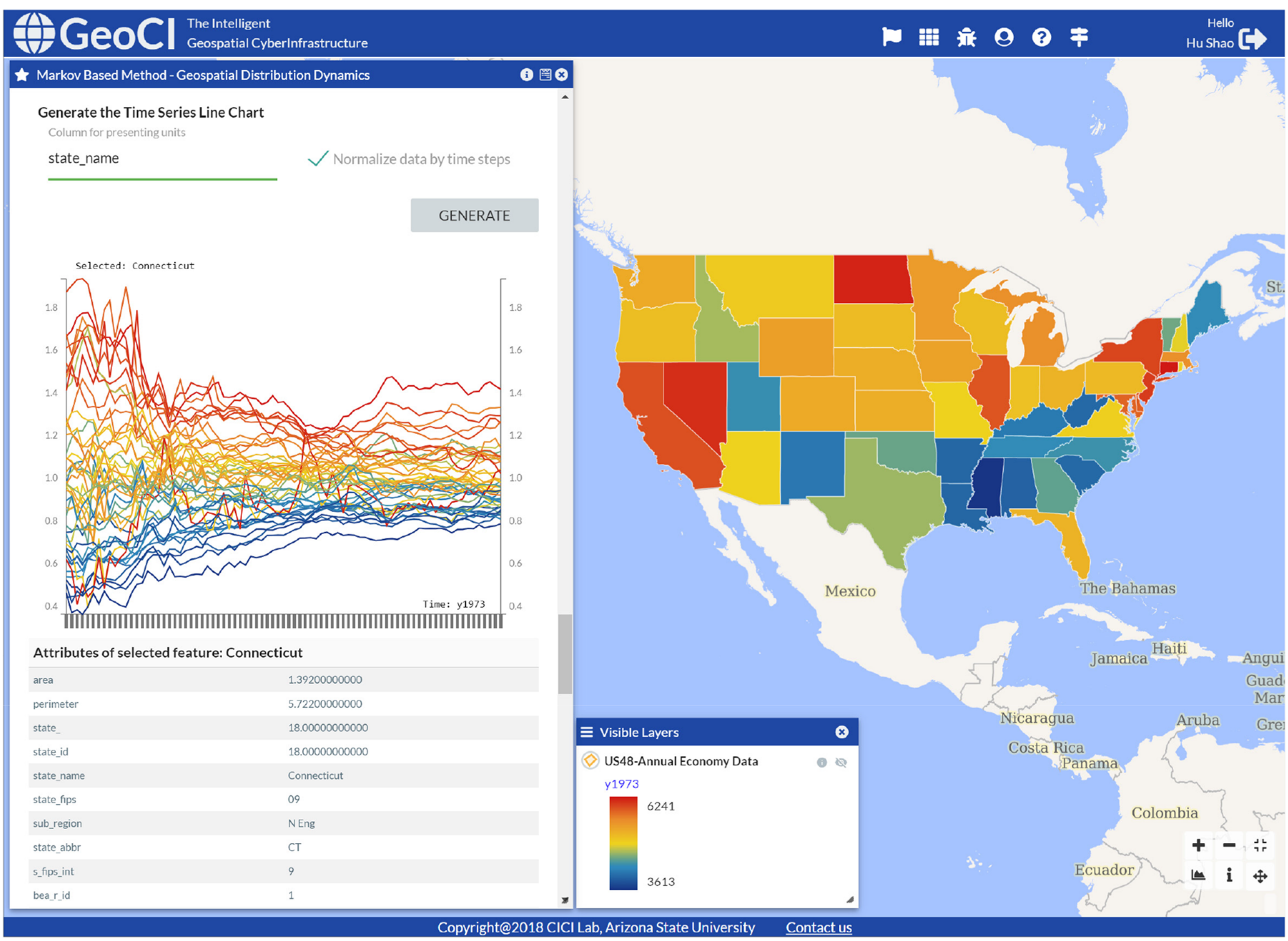


Fig. 7 Interactive visualization of average per capita incomes for the lower 48 US states, 1929–2009. On the time series plot chart, x axis refers to the year, and y axis refers to the per capita income. Each line shows the change of (normalized) per capita incomes of a state over time

state on the map, the attribute of which will be displayed under the time series plot.

Empirical Results and Visualization

The default values for the discretization and year-specific quintiles are used as the cutoffs to discretize the continuous per capita incomes and their spatial lags, giving rise to 5 income classes “Poor” (1), “Lower” (2), “Middle” (3), “Upper” (4), and “Rich” (5)), as well as a 5 × 5 transition probability matrix under the null of spatial randomness of dynamics and 5 5 × 5 transition probability matrices under the alternative of spatially dependent dynamics.

The results of the spatial Markov tests are appended to the interface of the analysis method once the calculation is completed. Figure 8a shows the results of 3 spatial Markov tests: all 3 test statistics are strongly significant: LR test statistic, 93.96 (p value 0.003); x^2 test statistic, 96.07 (p value 0.002); Kullback test statistic, 127.01 (p value 0.0006). These statistics together will very small p values thus reject the null hypothesis of spatial randomness of transition dynamics and confirm the role of space in shaping the dynamics of US state per capita incomes. We turn to the estimated transition probabilities to further explore how spatial contexts impact the dynamics. Figure 8b visualizes the estimated transition probability matrix P under the null of spatial random transition dynamics (“Pooled”) and 5 spatially conditioned transition probability matrices (“Spatial Lag 1/2/3/4/5”) in 6 heatmaps with red representing a high probability and purple a low probability. Since the null hypothesis is rejected based on all 3 spatial Markov tests, we only focus on the last 5 heatmaps. Taking the one with title “Spatial Lag 1” as an example, this heatmap visualizes the estimated probabilities of transitioning from one income class to another for any US state with poor neighbors. Comparatively, the heatmap with the title “Spatial Lag 5” visualizes the transition probabilities for any US state that has rich neighbors. For a US state which is poor and has poor neighboring states, the probability for it to stay poor is 0.934 (first row, first column in the matrix named Spatial Lag 1). In comparison, when a poor state has rich neighbors (Spatial Lag 5), the probability for it to stay poor is 0.857 (first row, first column in the matrix named Spatial Lag 5), lower

a

Analysis Results	
s matrix; (k, 1), ergodic distribution for a-spatial Markov.	0.179573762996596, 0.21631443192445254, 0.21499941873019354, 0.2113466241384415, 0.1777657622103163
Q Chi-square test of homogeneity across lag classes based on Bickenbach and Bode (2003) [Bickenbach2003]_.	96.06880345073806
Q_p_value p-value for Q.	0.0021468038924211674
LR Likelihood ratio statistic for homogeneity across lag classes based on Bickenback and Bode (2003) [Bickenbach2003]_.	93.96308889871956
LR_p_value p-value for LR.	0.0033281833802590866
dof_hom degrees of freedom for LR and Q, corrected for 0 cells.	60
kullback Kullback information based test of Markov Homogeneity.	{ "Conditional homogeneity": 127.02364377858612, "Conditional homogeneity dof": 80, "Conditional homogeneity pvalue": 0.0006443452550778384 }

Fig. 8 Output of spatial Markov tests. **a** Results of spatial Markov tests. **b** Heatmaps of the estimated transition probability matrices P under the null and alternative hypotheses

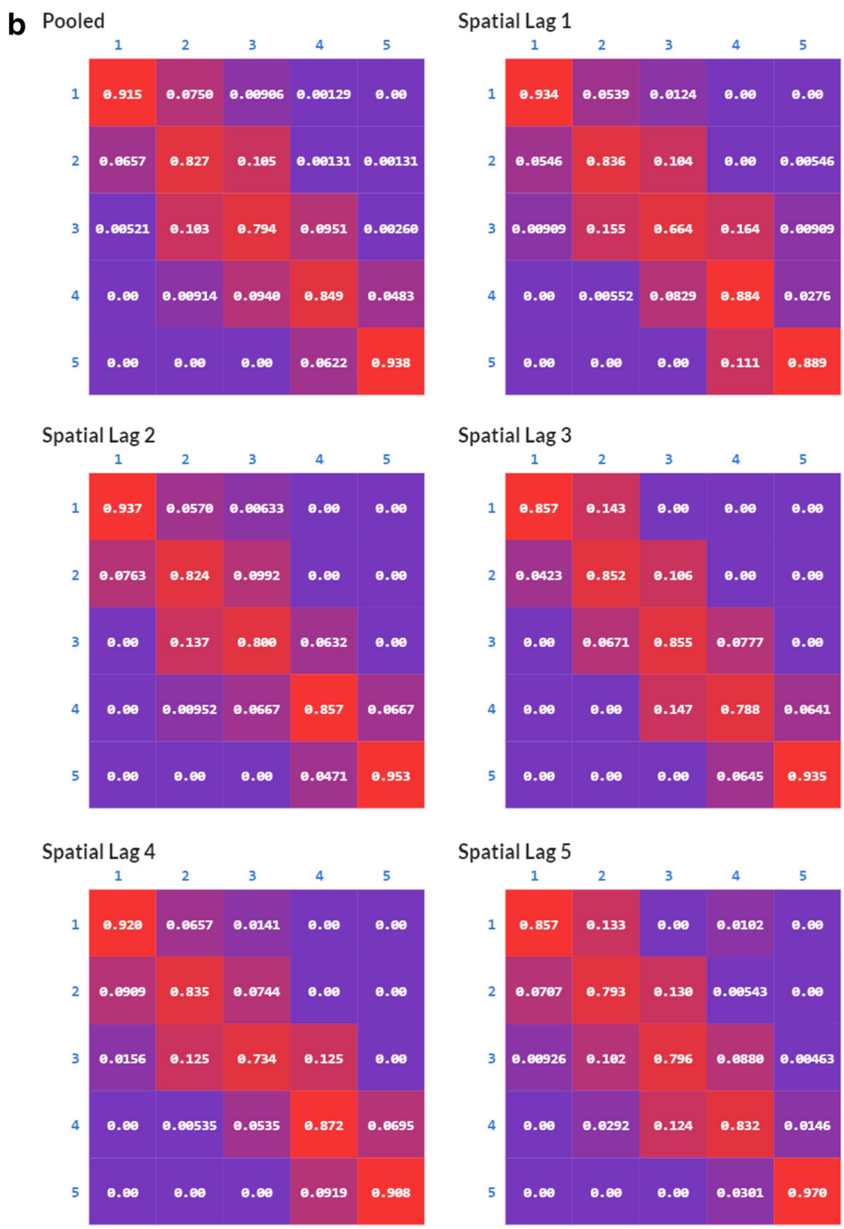


Fig. 8 (continued)

than 0.934 when a poor state has also poor neighbors. These findings point to spatial poverty trap where states' poverty status reinforces one another spatially and could also have important regional policy implications.

Comparison of Computational Time Between WebPySAL and PySAL

Compared with the desktop-based data analysis working mode, there is an overhead of communication time between the server side and client side when conducting the analysis on WebPySAL. In this section, we conduct a series of experiments to reveal whether the overhead of communication time will significantly affect WebPySAL's performance in terms of computational time. Experiment 1 will compare the performance variation of two spatial statistical methods: (1) local Moran's *I* statistics, (2) inter- and intra-regional indicators of exchange mobility (inter- and intra-regional Tau statistics) (Rey 2016). Experiment 2 will compare different data sets: a data set of 48 US states and a data set of 3141 US counties. Experiment 3 will compare different numbers (99, 499, 999) of permutations to use for the statistical inference.

The performance of the experiments is evaluated under the computing environment configured in the following way: WebPySAL is hosted on a server machine with 2 12-core 2.1-GHz 64-bit Xeon CPUs and 64-GB RAM running Ubuntu 16.04.4. The client side is tested on a laptop with a 4-core 2.50-GHz 64-bit Intel i-7 CPU and 8-GB RAM running Windows 10. The internet speed for the experiment is relatively high (50 Mbps). The geospatial data sets used for PySAL are stored locally on the same laptop, while the data sets for WebPySAL are provided as a WFS service hosted on the same server.

Series of tests are conducted with combinations of different methods, data sets, and permutation times. For the PySAL calculation, since the data loading time is very short, we only record the total time in each calculation. For WebPySAL, we record (1) the total calculation time and (2) the time used for communication and data transmission. Figure 9 presents the comparison results. The solid orange lines represent time consumption in PySAL; the solid blue lines and dashed lines represent the total calculation time and communication time, respectively, in WebPySAL. From the graphs, we find that:

- For the small state-based data set (results presented in the first column), all of the calculations can be completed within 1 s. Hence, the differences in time consumption between PySAL and WebPySAL will not be noticed by users.
- When calculating local Moran's *I* with the large county-based data set (results presented in the top right), the total time cost of WebPySAL is slightly longer. This is as a result of the communication between the server side and client side.
- When calculating the more complex regional Tau with county-based data, since the time spent on simulation is extensive, the communication time is negligible in such cases. It is worthy of note that since the Numpy library is adopted to conduct the matrix calculations, which is parallelized, the calculation time will be much shorter on a powerful machine. Hence, enabled by our GeoCI-based strategy, the computer's intensive computations can be moved to a cloud server from a local computer, yielding a much better performance in the WebPySAL environment as a result.

To summarize, the experimental results show that the communication overhead is consistently low compared with the actual computation time in WebPySAL. Hence, it is a feasible and efficient solution for handling complex computing tasks in the proposed service-oriented architecture.

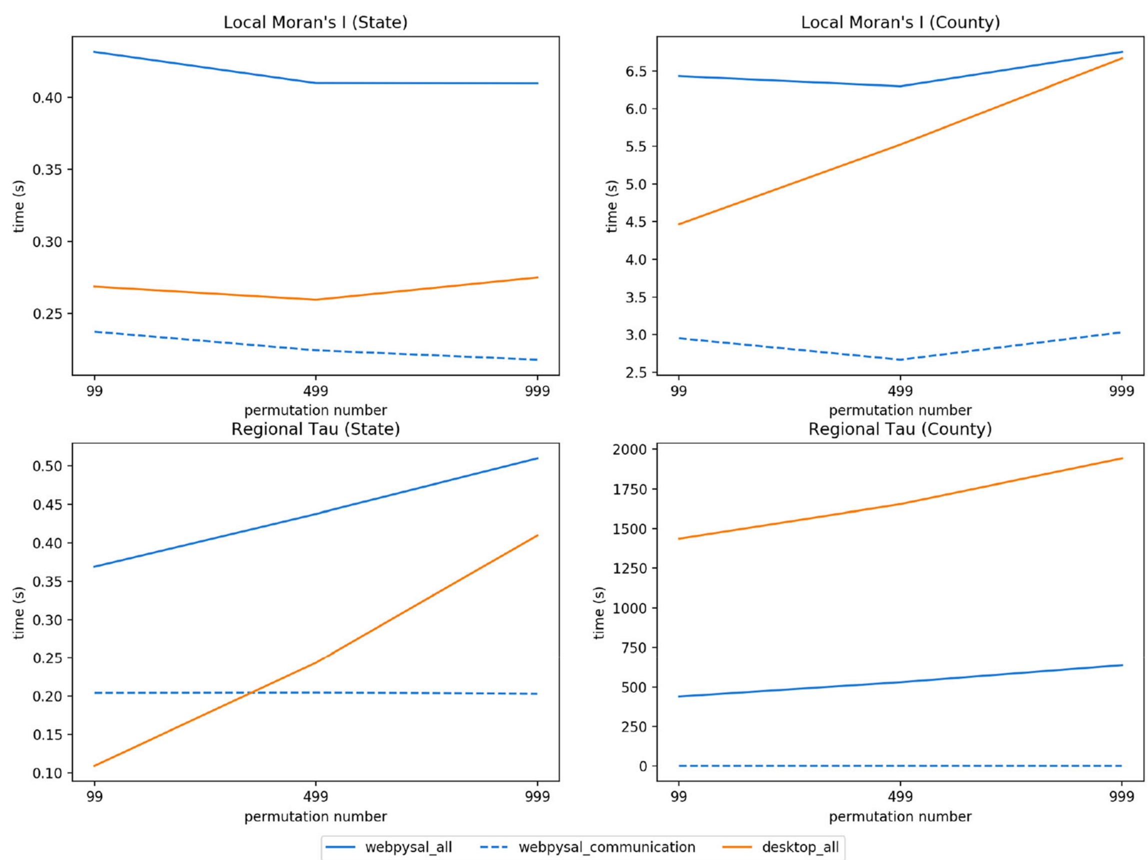


Fig. 9 Comparison of time taken on PySAL and WebPySAL in different experiments

Discussion and Conclusion

This article introduces our research in designing and implementing an interoperable and replicable CI—GeoCI for online spatial-statistical-visual analytics. In particular, we introduce the sharing of spatial analytical functions within a popular open source library, PySAL, through standard WPS interface. This new Web package—WebPySAL—is integrated seamlessly into GeoCI and works collaboratively with GeoCI's existing modules for data integration, visualization, and analytical workflow generation. The powerful GeoCI platform harnesses scalable spatial analysis modules to address real-world research challenges.

Our research contributes to the field in several important ways. We have established the GeoCI as a working model to benefit the GIScience and social science communities. We have presented strategies and methodologies concerning how to guarantee interoperability and replicability of complex spatial analysis. We have implemented an interactive and user-friendly GUI in the Web portal of GeoCI to assist users in conducting exploratory spatial/spatiotemporal data analysis with massive open-access geospatial data sets. In addition to the potential benefits of this work as a result of its bridging spatial analysis toolkits with CI, the design and implementation of this system also enables online replicability of analyses. Researchers can easily reproduce others' results in the GeoCI portal without the need of setting up system environment, implementing the algorithm or duplicating the workflow. Thus, this solution could also help users who lack a GIScience background, knowledge, or programming skills to better understand and adopt advanced spatial analytical methodologies.

The integration work of PySAL's advanced spatial analysis functionalities into GeoCI will continue. An active instance of GeoCI is currently available.[1] In the future, we will continue to exploit cutting-edge computing technology, such as edge-computing and microservices to enable the ubiquitous use of GeoCI in both traditional and edge devices (i.e., mobile phone). Parallelization strategies which incorporate space-time patterns in the data and analytics (Wang et al. 2020) will be developed and integrated into GeoCI to further improve system performance.

Funding Information This research was funded in part by National Science Foundation, grant number 1455349, 1831615, 1937908, 1936677.

Compliance with Ethical Standards

This paper is compliant with ethical standards.

Conflicts of Interest The authors declare that they have no conflict of interest.

[1] Please see <http://cici.lab.asu.edu/gci2>

References

Allard S (2012) DataONE: facilitating eScience through collaboration. J eSci Librariansh 1(1):3

Anselin L (1995) Local indicators of spatial association-LISA. Geogr Anal 27(2):93–115

Anselin L, Rey SJ (2012) Spatial econometrics in an age of CyberGIScience. Int J Geograph Inform Sci IJGIS 26(12):2211–2226

Anselin L & Rey SJ (2014) Modern spatial econometrics in practice: a guide to GeoDa, GeoDaSpace and PySAL. GeoDa Press

Anselin L, Rey SJ, Li W (2014) Metadata and provenance for spatial analysis: the case of spatial weights. Int J Geogr Information Sci 28(11):2261–2280

Anselin L, Syabri I, Kho Y (2010) GeoDa: an introduction to spatial data analysis. Handbook of applied spatial analysis. pp 73–89

Astsatryan H, Hayrapetyan A, Narsisian W, Saribekyan A, Asmaryan S, Saghatelyan A, Muradyan V, Guigoz Y, Giuliani G, Ray N (2015) An interoperable Web portal for parallel geoprocessing of satellite image vegetation indices. Earth Sci Inf 8(2):453–460

Bivand R, Anselin L, Berke O al Bernat A, Carvalho M, Chun Y et al (2011) spdep: spatial dependence: weighting schemes, statistics and models. R package version 0.5-31, URL http://CRAN.R-project.org/package=spdep. Retrieved from http://ftp.auckland.ac.nz/software/CRAN/src/contrib/Descriptions/spdep.html

Brandl G (2009) Sphinx: python documentation generator. URL: https://pypi.org/project/Sphinx/10.(last accessed on April 10, 2020)

Čepický J (2007) PyWPS 2.0.0: the presence and the future. Geoinformatics FCE CTU 2:61–64

Cerón M, Fernández-Carmona M, Urdiales C, Sandoval F (2018) Smartphone-based vehicle emission estimation. In: Proceedings of the International Conference on Information Technology & Systems (ICITS 2018). Springer International Publishing, pp 284–293

Cliff AD, Ord JK (1981) Spatial processes, models & applications. Pion, London

Delipetrev B, Jonoski A, Solomatine DP (2014) Development of a web application for water resources based on open source software. Comput Geosci 62:35–42

Dubois G, Schulz M, Skøien J, Bastin L, Peedell S (2013) eHabitat, a multi-purpose Web processing service for ecological modeling. Environ Model Softw 41:123–133

Harris R (2003) Building a GIScience community in cyberspace: reflections on GIScOnline. J Geogr High Educ 27(3):279–295

Kang W, Rey SJ (2018) Conditional and joint tests for spatial effects in discrete Markov chain models of regional income distribution dynamics. Ann Reg Sci 61(1):73–93

Keller GR (2003) GEON (GEOscience network): a first step in creating cyberinfrastructure for the geosciences. Seismol Res Lett 74(4): 441–444

Krishnan S, Crosby C, Nandigam V, Phan M, Cowart C, Baru C, Arrowsmith R (2011) OpenTopography: a services oriented architecture for community access to LIDAR topography. In: Proceedings of the 2nd International Conference on Computing for Geospatial Research & Applications. ACM, p 7

Kullback S, Kupperman M, Ku HH (1962) Tests for contingency tables and Markov chains. Technometrics J Stat Phys Chem Eng Sci 4(4): 573–608

Laura J, Li W, Rey SJ, Anselin L (2015) Parallelization of a regionalization heuristic in distributed computing platforms – a case study of parallel-p-compact-regions problem. Int J Geograph Inform Sci IJGIS 29(4):536–555

Li W (2018) Lowering the barriers for accessing distributed geospatial big data to advance spatial data science: the PolarHub solution. Ann Am Assoc Geogr 108(3):773–793

Li W, Bhatia V, Cao K (2015a) Intelligent polar cyberinfrastructure: enabling semantic search in geospatial metadata catalogue to support polar data discovery. Earth Sci Inf 8(1):111–123

Li W, Song M, Zhou B, Cao K, Gao S (2015b) Performance improvement techniques for geospatial Web services in a cyberinfrastructure environment–a case study with a disaster management portal. Comput Environ Urban Syst 54:314–325

Li W, Shao H, Wang S, Zhou X, Wu S (2016a) A2CI: a cloud-based, service-oriented geospatial cyberinfrastructure to support atmospheric research. In: Cloud computing in ocean and atmospheric sciences. Academic Press, pp 137–161

Li W, Wang S, Bhatia V (2016b) PolarHub: a large-scale Web crawling engine for OGC service discovery in cyberinfrastructure. Comput Environ Urban Syst 59:195–207

Li W, Wu S, Song M, Zhou X (2016c) A scalable cyberinfrastructure solution to support big data management and multivariate visualization of time-series sensor observation data. Earth Sci Inf 9(4):449–464

Li W, Goodchild MF, Anselin L, Weber KT (2019a) A smart service-oriented CyberGIS framework for solving data-intensive geospatial problems. In: CyberGIS for geospatial discovery and innovation. Springer, Dordrecht, pp 189–211

Li W, Song M, Tian Y (2019b) An ontology-driven cyberinfrastructure for intelligent spatiotemporal question answering and open knowledge discovery. ISPRS Int J Geo Inf 8(11):496

Li X, Di L, Han W, Zhao P, Dadi U (2010) Sharing geoscience algorithms in a Web service-oriented environment (GRASS GIS example). Comput Geosci 36(8):1060–1068

McMillen DP, McDonald JF (1991) A Markov chain model of zoning change. J Urban Econ 30(2):257–270

Mihon D, Colceriu V, Bacu V, Gorgan D (2013) Grid based processing of satellite images in GreenLand Platform. Int J Adv Comput Sci Appl 3:41–49

Neteler M, Bowman MH, Landa M, Metz M (2012) GRASS GIS: a multi-purpose open source GIS. Environ Model Softw 31:124–130

Pebesma E (2012) spacetime: spatio-temporal data in R. J Stat Softw 51(7). https://doi.org/10.18637/jss.v051.i07

Quah DT (1993) Empirical cross-section dynamics in economic growth. Eur Econ Rev 37(2-3):426–434

Rajib MA, Merwade V, Kim IL, Zhao L, Song C, Zhe S (2016) SWATShare – a Web platform for collaborative research and education through online sharing, simulation and visualization of SWAT models. Environ Model Softw 75:498–512

Rey SJ (2014) Python Spatial Analysis Library (PySAL): an update and illustration. In: Brunsdon C, SIngleton A (eds) Geocomputation. Sage

Rey SJ (2016) Space–time patterns of rank concordance: local indicators of mobility association with application to spatial income inequality dynamics. Ann Am Assoc Geographers 106(4):788–803

Rey SJ, Anselin L (2010) PySAL: a python library of spatial analytical methods. In: Fischer MM, Getis A (eds) Handbook of applied spatial analysis: software tools, methods and applications. Springer, Berlin, Heidelberg, pp 175–193

Rey SJ, Anselin L, Li X, Pahle R, Laura J, Li W, Koschinsky J (2015) Open geospatial analytics with PySAL. ISPRS Int J Geo Inf 4(2): 815–836

Rey SJ, Janikas MV (2010) STARS: space-time analysis of regional systems, In Handbook of applied spatial analysis (pp. 91-112). Springer, Berlin, Heidelberg

Rey SJ, Kang W, Wolf L (2016) The properties of tests for spatial effects in discrete Markov chain models of regional income distribution dynamics. J Geogr Syst 18(4):377–398

Rey SJ, Murray AT, Grubesic TH, Mack E, Wei R, Anselin L, Griffin M (2014) Sex offender residential movement patterns: a Markov chain analysis. Prof Geogr J Assoc Am Geogr 66(1):102–111

Rodero Castro I, Parashar M (2016) Architecting the cyberinfrastructure for National Science Foundation Ocean Observatories Initiative (OOI). In: Instrumentation viewpoint, vol 19. SARTI, pp 99–101

Rinner C, Keßler C, Andrulis S (2008) The use of Web 2.0 concepts to support deliberation in spatial decision-making. Comput Environ Urban Syst 32(5):386–395

Steiniger S, Hunter AJS (2013) The 2012 free and open source GIS software map - a guide to facilitate research, development, and adoption. Comput Environ Urban Syst 39:136–150

Sugumaran R, Meyer JC, Davis J (2009) A Web-based environmental decision support system for environmental planning and watershed management. In: Handbook of applied spatial analysis, pp 703–718

Sun Y, Li S (2016) Real-time collaborative GIS: a technological review. ISPRS J Photogramm Remote Sens Off Publ Int Soc Photogramm Remote Sens 115:143–152

Swain NR, Latu K, Christensen SD, Jones NL, Nelson EJ, Ames DP, Williams GP (2015) A review of open source software solutions for developing water resources Web applications. Environ Model Softw 67:108–117

Song M, Li W, Zhou B, Lei T (2016) Spatiotemporal data representation and its effect on the performance of spatial analysis in a cyberinfrastructure environment–a case study with raster zonal analysis. Comput Geosci 87:11–21

Shao H, Li W (2018) A comprehensive optimization strategy for real-time spatial feature sharing and visual analytics in cyberinfrastructure. Int J Digit Earth:1–20

Unsworth J (2008) Cyber infrastructure for the humanities and social sciences. Sydney University Press

Veenendaal B (2015) Using the geospatial Web to deliver and teach GIScience education programs. ISPRS - Int Arch Photogramm Remote Sens Spat Inform Sci XL-6/W1:17–21

Wang FZ, Helian N, Wu S, Guo Y, Deng DY, Meng L et al (2009) Eight times acceleration of geospatial data archiving and distribution on the grids. IEEE Trans Geosci Remote Sens Publ IEEE Geosci Remote Sens Soc 47(5):1444–1453

Wang S (2013) CyberGIS: blueprint for integrated and scalable geospatial software ecosystems. Int J Geograph Inform Sci IJGIS 27(11):2119–2121

Wang S, Armstrong MP (2009) A theoretical approach to the use of cyberinfrastructure in geographical analysis. Int J Geograph Inform Sci IJGIS 23(2):169–193

Wang S, Liu Y (2009) TeraGrid GIScience Gateway: bridging cyberinfrastructure and GIScience. Int J Geograph Inform Sci IJGIS 23(5):631–656

Wang Y, Gui Z, Wu H, Peng D, Wu J, Cui Z (2020) Optimizing and accelerating space–time Ripley's K function based on Apache Spark for distributed spatiotemporal point pattern analysis. Futur Gener Comput Syst 105:96–118

Warmerdam F (2008) The geospatial data abstraction library. In: Open source approaches in spatial data handling. Springer, Berlin, Heidelberg, pp 87–104

Wu A, Convertino G, Ganoe C, Carroll JM, Zhang X (luke) (2013) Supporting collaborative sense-making in emergency management through geo-visualization. Int J Hum Comput Stud 71(1):4–23

Wu H, You L, Gui Z, Hu K, Shen P (2015) GeoSquare: collaborative geoprocessing models' building, execution and sharing on Azure Cloud. Ann GIS 21(4):287–300