

# SpatialEpiApp: A Shiny Web Application for the Analysis of Spatial and Spatio-Temporal Disease Data

Paula Moraga  
Lancaster University

---

## Abstract

During last years, public health surveillance has been facilitated by the existence of several packages implementing statistical methods for the analysis of spatial and spatio-temporal disease data. However, these methods are still inaccessible for many health researchers lacking the adequate statistical and programming skills to effectively use the required software. In this paper we present **SpatialEpiApp**, a Shiny web application that integrates two of the most common approaches for health surveillance: disease mapping and detection of clusters. **SpatialEpiApp** is easy to use and does not require any programming knowledge. Given information about the cases, population and optionally covariates for each of the areas and dates of study, the application allows to fit Bayesian disease models to obtain disease risk estimates and their uncertainty for each of the areas and dates by using INLA, and to detect spatial and spatio-temporal disease clusters by using the scan statistics implemented in **SaTScan**. The application allows user interaction and the creation of interactive data visualizations and reports showing the analyses performed.

*Keywords:* spatial and spatio-temporal epidemiology, disease mapping, detection of clusters, visualization, R, Shiny, INLA, **SaTScan**.

---

## 1. Introduction

Public health surveillance provides information to identify public health problems and respond appropriately when they occur. This information is crucial to prevent and control a variety of health conditions such as chronic and infectious diseases, injuries, or health-related behaviors (Thacker and Berkelman 1988; Lawson and Kleinman 2005). There is a wide range of spatial and spatio-temporal methods and software that can be applied as a surveillance tool, and these are useful for highlighting areas at high risk (Moraga *et al.* 2015), detecting disease clusters (Moraga and Montes 2011), early detection of epidemics (Stelling *et al.* 2010), assessing disease risk in relation to a putative source (Wakefield and Morris 2001), and identifying disease risk factors (Hagan *et al.* 2016). For example, Bayesian disease mapping models can be used when the interest is to understand the spatial and the temporal disease patterns. Bayesian inference may be performed using the Integrated Nested Laplace Approximation (INLA) approach (Rue *et al.* 2009) which is implemented in the R package **R-INLA** (Lindgren and Rue 2015). Also, the detection of clusters in spatial, temporal, and space-time settings can be performed using the scan statistics methodology (Kulldorff and Nagarwalla 1995; Kulldorff 1997) which is implemented in the statistical package **SaTScan**

(Kulldorff 2006a). Unfortunately, many researchers with training in biomedical sciences lack the adequate programming skills to effectively use the statistical softwares required to conduct the statistical analyses.

In this paper we present **SpatialEpiApp**, a Shiny web application (Chang *et al.* 2016) for the analysis of spatial and spatio-temporal disease data. This application is useful for interactive data visualization and integrates two of the most common approaches in public health surveillance: disease mapping and detection of clusters. **SpatialEpiApp** is mainly addressed to health researchers interested in understanding the spatial and spatio-temporal variations of disease but lacking the appropriate knowledge to carry out the statistical analysis and the processing of the results. The application is also useful for researchers with more advanced statistical skills since it allows to visualize the data using interactive maps and tables which can be useful prior to the statistical analyses.

In spatial epidemiology, available data can be point data containing the locations at which cases of disease occur, or areal data that arise when point data are aggregated over disjoint subregions of the region of study due to ethical concerns over data use and patient confidentiality. **SpatialEpiApp** is designed to work with areal data. Given information about the cases, population and optionally covariates for each of the areas in the study region and periods of time, the application allows to fit Bayesian disease models to obtain risk estimates and their uncertainty for each of the areas and dates by using INLA, and to detect spatial and spatio-temporal clusters by using the scan statistics implemented in **SaTScan**. To carry out these analyses users simply need to click the buttons that create the input files required, execute the software and process the output to generate tables of values and plots needed for the interpretation of the results. The application allows user interaction, creates interactive visualizations of the data and results, and generates reports showing the analyses performed.

The remainder of the paper is organized as follows. First we briefly introduce the statistical methods and software used in spatial and spatio-temporal epidemiology. Then we introduce Shiny and describe the main components we use to write the application. In Section 4 we illustrate the use of the application via a spatio-temporal analysis of lung cancer mortality in Ohio, United States, in years 1981 to 1984. Specifically, we discuss the input files required, show how to perform the statistical analyses and interpret the results, and explain how to generate reports. Instructions for the installation of the application are provided in Section 5. Finally, the conclusions are presented.

## 2. Spatial and spatio-temporal epidemiology

Spatial and spatio-temporal epidemiology is concerned with the description and analysis of spatial and spatio-temporal variations in disease risk with respect to risk factors such as sociodemographic and environmental covariates (Elliott *et al.* 2000). Over the past few decades, this field has been aided by the increased availability of geographically indexed health data, the development of geographic information systems (GIS), as well as the advances in statistical methodology such as disease mapping and cluster detection methods.

Disease mapping is widely used in public health surveillance since it allows to describe the spatial and spatio-temporal variation of the disease, identify areas which exhibit an unusual high risk and formulate etiological hypotheses (Lawson 2009). Cluster analysis has also attracted a great interest, and various techniques have been developed for evaluating whether the

incidence of a disease shows a particular tendency to group together (Kulldorff 2006b). The information resulting from these analyses is crucial to appropriately implement prevention and surveillance programmes.

The available data in these type of studies can be point data containing the locations at which cases of disease occur, or areal data containing aggregated outcomes over a finite number of subregions forming a partition of the region of study. Whenever point data are available, it is recommended to perform the analyses at this level of resolution so detailed information do not get lost. However, the exact location of the individuals is not always available for various reasons, such as confidentiality. In the following subsections we introduce some of the most common statistical methods and software for disease mapping and the detection of clusters using areal data. These are the approaches that are implemented in **SpatialEpiApp**.

## 2.1. Disease mapping

Most disease mapping approaches estimate disease risks within areal units that form a partition of the study region such as zip codes or provinces, and this is mainly done for confidentiality reasons (Waller and Gotway 2004). Often, the disease risk in area  $i = 1, \dots, N$  is estimated by the standardized incidence ratio (SIR) which is obtained as the ratio of observed counts versus the expected counts:  $SIR_i = Y_i/E_i$ . The expected counts in each of the areas are calculated based on their population demographics and usually the internally standardized expected counts are used. Specifically,

$$E_i = \sum_{j=1}^m r_j^{(s)} n_j,$$

where  $r_j^{(s)}$  is the rate in stratum  $j$  in the standard population (usually the relevant national or regional population), and  $n_j$  is the population in stratum  $j$  of the area. The expected counts represent the total number of events one would expect if the observed population behaved the way the standard population behaved.

In areas with small populations SIRs can be very extreme and are insufficiently reliable for reporting. In contrast, Bayesian disease models are preferred to obtain risks estimates since they enable to borrow information from neighboring areas and may also incorporate covariate information to improve local estimates (Gelfand *et al.* 2010).

A common practice is to model the observed counts  $Y_i$  in area  $i$ , using a Poisson distribution with mean  $E_i \times \theta_i$ , where  $E_i$  is the expected counts and  $\theta_i$  is the relative risk in area  $i$ . To model the relative risk  $\theta_i$ , a log linear model is used which includes an intercept to model the overall disease level, and random effects that account for extra-Poisson variability in the observed data (Lawson 2009). The standard model in disease mapping is expressed as

$$Y_i \sim Po(E_i \times \theta_i), \quad i = 1, \dots, N;$$

$$\log(\theta_i) = \alpha + u_i + v_i.$$

Here,  $\alpha$  represents the overall risk in the region of study,  $u_i$  is a correlated heterogeneity (CH) component that models the spatial dependence between the relative risks, and  $v_i$  is an unstructured exchangeable component that models uncorrelated noise (UH). Sometimes other terms such as covariates and random effects that can deal with other sources of variability can also be included in the model.

The most popular model to spatial disease mapping is the Besag-York-Mollie (BYM) model (Besag *et al.* 1991). Here, the clustering component  $u_i$  is modelled with the conditional autoregressive (CAR) distribution. The CAR distribution smoothes the data according to a certain neighborhood structure given by a neighborhood matrix that is specified such as two areas are neighbours if they share a common boundary. The CAR distribution is expressed as

$$u_i | \mathbf{u}_{-i} \sim N \left( \bar{u}_{\delta_i}, \frac{\sigma_u^2}{n_{\delta_i}} \right),$$

where  $\bar{u}_{\delta_i} = n_{\delta_i}^{-1} \sum_{j \in \delta_i} u_j$ ,  $\delta_i$  represents the set of neighbours of area  $i$ , and  $n_{\delta_i}$  is the number of neighbours of area  $i$ . The uncorrelated heterogeneity  $v_i$  is modeled as independent and identically distributed normal variables with mean zero and variance  $\sigma_v^2$

Spatio-temporal disease mapping models are used when the interest is to understand the spatial and the temporal disease patterns. In the space-time setting, the disease count  $Y_{ij}$  observed in the area  $i$  and time period  $j$ , may be modeled as

$$Y_{ij} \sim Po(E_{ij} \times \theta_{ij}),$$

where  $\theta_{ij}$  is the risk and  $E_{ij}$  is the expected number of cases in the given area and period of time. Then, three groups of components for  $\log(\theta_{ij})$  can be considered:

$$\log(\theta_{ij}) = \alpha_0 + A_i + B_j + C_{ij},$$

where  $A_i$  is the spatial group,  $B_j$  is the temporal group, and  $C_{ij}$  is the space-time interaction group (Lawson 2009). For example, in Bernardinelli *et al.* (1995) these groups are defined as follows:  $A_i = u_i + v_i$ ,  $B_j = \beta t_j$  and  $C_{ij} = \delta_i t_j$  where  $u_i + v_i$  is an area random effect,  $\beta t_j$  is a linear trend term in time  $t_j$ , and  $\delta_i t_j$  is an interaction random effect between area and time.

Bayesian inference may be performed using the Integrated Nested Laplace Approximation (INLA) (Rue *et al.* 2009). INLA uses a combination of analytical approximation and numerical integration to do approximate Bayesian inference in latent Gaussian models which includes a large class of models ranging from generalized linear mixed to spatial and spatio-temporal models. The INLA approach is implemented in the R package **R-INLA** (Lindgren and Rue 2015). This package can be downloaded from the <http://www.r-inla.org> website which also includes documentation about the package, examples and a discussion forum.

## 2.2. Detection of clusters

Methods for the detection of clusters are designed for detecting and localizing specific clusters and evaluating their statistical significance. The scan statistics methodology (Kulldorff and Nagarwalla 1995; Kulldorff 1997) has been implemented as a major analytical tool for cluster detection in a spatial, temporal, and space-time setting. A statistical package, **SaTScan**, facilitates its use and can be downloaded from the <http://www.satscan.org> website (Kulldorff 2006a).

Scan statistics gradually scan the study region with a huge number of overlapping windows and determine the windows which group together an unusual number of cases (Kulldorff 1997). Typically, the spatial version uses circular windows with radius varying continuously from zero to some upper limit such as that it does not pass beyond 50% of the at risk population. In the space-time setting windows are defined as cylinders. The circular base defines a geographical

area and the height the time period. For each choice of base all choices of the temporal height are considered and vice versa.

Conditioning on the observed total number of cases  $C$ , the scan test statistic  $S$  is defined as the maximum likelihood ratio over all possible windows  $Z$ . The likelihood ratio  $S$  is expressed as

$$S = \max_Z \frac{L(Z)}{L_0},$$

where  $L(Z)$  is the likelihood for window  $Z$ , and  $L_0$  is the likelihood function under the null hypothesis which states that the probability of being a case inside  $Z$  is equal to the probability of being a case outside  $Z$ . When the number of cases in each window is Poisson distributed a Poisson model is used. Under this model, the ratio  $L(Z)/L_0$  for a specific window is

$$\left(\frac{c}{E(Z)}\right)^c \left(\frac{C-c}{C-E(Z)}\right)^{C-c},$$

if  $c > E(Z)$ , and 1 otherwise. Here,  $c$  is the observed number of cases within  $Z$ , and  $E(Z)$  denotes the covariate adjusted expected number of cases under the null hypothesis.

The window with the maximum likelihood constitutes the most likely cluster. The cluster statistical significance is obtained through Monte Carlo hypothesis testing (Dwass 1957). Specifically, the previous procedure is repeated for a large number of replicas, say  $R$ , of data generated under the null hypothesis, and their respective test statistics are calculated. Next, the test statistic of the real data is combined with these, and the set of the  $R + 1$  values are ordered. The p-value is obtained as  $M/(R + 1)$ , where  $M$  is the rank of the test statistic from the real data. Apart from the most likely cluster, secondary clusters can also be identified, ordered according to the value of  $S$ .

### 3. SpatialEpiApp

**SpatialEpiApp** is a Shiny web application (Chang *et al.* 2016) that allows to visualize spatial and spatio-temporal disease data, estimate disease risk and detect clusters. It is addressed to health researchers interested in analyzing disease data but lacking the appropriate programming skills to use the statistical software required. The application allows to fit Bayesian disease models to obtain risk estimates and their uncertainty by using INLA, and to detect clusters by using the scan statistics implemented in **SaTScan**. To carry out these analyses users simply need to click the buttons that create the input files required, execute the software and process the output to generate tables of values and plots with the results. The application allows user interaction and creates interactive visualizations such as maps supporting panning and zooming and tables that allow for filtering. It also enables the generation of reports containing the analyses performed.

The application consists of three pages: 1) an 'Inputs' page where the user can upload the input files and select the type of analysis; 2) an 'Analysis' page where statistical analyses are carried out, results can be visualized, and reports can be generated; and 3) a 'Help' page containing information about the application use and references. In the following subsections, we give a brief summary of Shiny and explain the basic elements for implementing the pages in our application including control widgets, HTML widgets and R Markdown (Allaire *et al.* 2016).

### 3.1. Shiny

Shiny is a web application framework for R that enables to build interactive web applications. Two R scripts are needed to create the application: an user-interface script called `ui.R` and a server script `server.R`. The user-interface script controls the layout and appearance of the application. The server script contains the R objects and the instructions about how they are displayed. Shiny applications use a functionality called reactivity to support interactivity. In this way, users can introduce texts, select dates or modify other inputs and automatically the R objects displayed will change.

Reactive objects can be created with the following steps. First, R objects are added to the user-interface. This is done by placing `*Output` functions that turn R objects into output in the `ui.R` script. Next, the R code that builds the objects is provided in `server.R`. This script contains an unnamed function with two list-like objects named `output` and `input`. `Output` contains all the instructions for building the R objects, and `input` stores the current values of the objects in the application. The objects are built by using a `render*` function and then saved in the `output` list. Reactivity is created by including an input value in a `render*` expression. For example, to create a reactive plot we need to add a `plotOutput` function in the `ui.R`. Then we use a `renderPlot` function to build the plot and add it to the output object in the `server.R` script.

To create Shiny applications there is not web development experience required, although it is possible to use HTML, CSS, or JavaScript to achieve greater flexibility and customization. Shiny applications can be run locally by users that have the application files and R installed in their computer. Applications can also be hosted as a web page at its own URL and can be navigated through the internet with a web browser. This facilitates its use to people without R knowledge. Examples of Shiny applications are provided in the gallery: <http://shiny.rstudio.com/gallery>.

### 3.2. Control widgets

*SpatialEpiApp* includes several control widgets users can interact with to send messages to the application. For example, in the 'Inputs' page we can upload the map and the data files by using a file upload control which is created by the `fileInput` function. In this page we can also specify the variables names by selecting the appropriate names from boxes containing the possible choices. These boxes are created with the `selectInput` function. The 'Inputs' page also includes a pair of calendars for selecting the analysis minimum and maximum dates that are created with the `dateRangeInput` function, and radio buttons that enable to specify the temporal unit in the data (year, month or day), and the type of analysis (spatial or spatio-temporal) that are created with the `radioButtons` function.

In the 'Analysis' page there is a slider bar created with the `sliderInput` function that allows to select a range of values by which we wish to filter the data. We also have the option to download tables with values and reports by clicking the corresponding download buttons that are created using the `downloadButton` function in `ui.R` and the `downloadHandler` function in `server.R`.

Finally, *SpatialEpiApp* includes some buttons that permit to switch between the 'Inputs', 'Analysis' and 'Help' pages, create plots, and carry out the statistical analyses. These buttons are created with the `actionButton` function.

### 3.3. HTML widgets

**SpatialEpiApp** includes three HTML widgets for interactive web data visualization, namely, **Leaflet** for rendering maps (Cheng and Xie 2016), **dygraphs** for plotting time series (Vanderkam *et al.* 2016), and **DataTables** for displaying data objects (Xie 2016). HTML widgets are created with JavaScript libraries and embedded in Shiny by using the **htmlwidgets** package (Vaidyanathan *et al.* 2016).

**Leaflet** is used to render maps depicting the values of the variables of interest in each of the areas. The created maps support interactive panning and zooming which is very convenient when the map contains small areas. **Dygraphs** is used for showing the variables of interest over time. These plots include support for interactive features such as panning, zooming and series highlighting. **DataTables** is used to display interactive tables containing the information of the variables of interest. These tables support filtering, pagination, and sorting which is very helpful in situations where we wish to locate the information of one particular area or show the areas with the higher or lower values.

These three HTML widgets are included into the application by calling an output for the widget in the user-interface, and assigning a render call to the output on the server side in the same way as `plotOutput` and `renderPlot` functions work. **Leaflet** uses `leafletOutput` and `renderLeaflet`, **dygraphs** `dygraphOutput` and `renderDygraph`, and **DataTable** `dataTableOutput` and `renderDataTable`.

### 3.4. R Markdown

The application allows the generation of reports in PDF format that include the results of the analyses performed. The reports are R Markdown documents, that is, they are written in a plain text formatting syntax called markdown and include chunks of executable R code. They also include chunks of LaTeX code, <http://www.latex-project.org>. The reports are generated using the **knitr** package (Xie 2015) which runs the R code and add the results of the code to the final document.

### 3.5. Dependencies

**SpatialEpiApp** is dependent on several softwares and R packages including **SpatialEpi** (Kim and Wakefield 2016) for computing the internally indirect standardized expected disease cases; **R-INLA** (Lindgren and Rue 2015) for performing Bayesian inference; **SaTScan** (Kulldorff 2006a) for the detection of clusters; **spdep** (Bivand and Piras 2015) for neighbourhood matrix manipulation; **xts** (Ryan and Ulrich 2014) for creating time-series objects; and **ggplot2** (Wickham 2009) and **maptools** (Bivand and Lewin-Koh 2016) for data visualisations. Information about all the dependencies is shown in Table 1.

## 4. Use of SpatialEpiApp

In this section we demonstrate the use of the application through a spatio-temporal analysis of lung cancer mortality in the Ohio counties in years 1981 to 1984. These data appear

Name		Description
Softwares		
R	(R Core Team 2017)	Language and environment for statistical computing and graphics
<b>SaTScan</b>	(Kulldorff 2006a)	Software that analyzes spatial, temporal and space-time data using scan statistics
R packages		
<b>dplyr</b>	(Wickham and Francois 2016)	A fast, consistent tool for working with data frame like objects, both in memory and out of memory
<b>dygraphs</b>	(Vanderkam <i>et al.</i> 2016)	Interface to 'Dygraphs' Interactive Time Series Charting Library
<b>ggplot2</b>	(Wickham 2009)	Creates elegant data visualisations using the grammar of graphics
<b>htmlwidgets</b>	(Vaidyanathan <i>et al.</i> 2016)	Provides a framework for easily creating R bindings to JavaScript libraries
<b>knitr</b>	(Xie 2015)	Tool for dynamic report generation in R
<b>leaflet</b>	(Cheng and Xie 2016)	Create Interactive Web Maps with the JavaScript 'Leaflet' Library
<b>mapproj</b>	(McIlroy <i>et al.</i> 2015)	Converts latitude/longitude into projected coordinates
<b>maptools</b>	(Bivand and Lewin-Koh 2016)	Set of tools for manipulating and reading geographic data, in particular ESRI shapefiles
<b>RColorBrewer</b>	(Neuwirth 2014)	Provides color schemes for maps and other graphics
<b>rgeos</b>	(Bivand and Rundel 2016)	Interface to Geometry Engine - Open Source (GEOS) using the C API for topology operations on geometries
<b>rmarkdown</b>	(Allaire <i>et al.</i> 2016)	Convert R Markdown documents into a variety of formats including HTML, MS Word, PDF, and Beamer
<b>R-INLA</b>	(Lindgren and Rue 2015)	Performs full Bayesian analysis on generalised additive mixed models using Integrated Nested Laplace Approximations
<b>shiny</b>	(Chang <i>et al.</i> 2016)	Web Application Framework for R
<b>shinyjs</b>	(Attali 2016)	Perform common useful JavaScript operations in Shiny apps that will greatly improve the apps without having to know any JavaScript
<b>SpatialEpi</b>	(Kim and Wakefield 2016)	Contains methods for cluster detection, disease mapping and plotting methods
<b>spdep</b>	(Bivand and Piras 2015)	Contains a collection of functions for spatial dependence: weighting schemes, statistics and models
<b>xts</b>	(Ryan and Ulrich 2014)	Extensible time series class and methods

Table 1: Softwares and R packages used for developing *SpatialEpiApp*.



Figure 1: Inputs page. Selected options correspond to the Ohio example.

in Lawson *et al.* (2003) and are available at <http://Paula-Moraga.github.io/software>. The Ohio map has been obtained from the United States Census Bureau website, <http://www.census.gov>.

#### 4.1. Inputs page

The 'Inputs' page is the first page we see when we launch the application. In this page we can upload the required files and select the type of analysis to be performed. It is composed of four components: 1) Upload map; 2) Upload data; 3) Select analysis; and 4) Contents (see Figure 1).

##### *Upload map*

First, we need to upload the map file containing the areas of the region of study. The map file needs to be in shapefile format and it can be uploaded by clicking the 'Browse' button and selecting all the files corresponding to the map. Then we need to specify which are the names of the columns id and name of the areas, and optionally the name of the column name of the region. This is done by selecting the correct options from the dropdown menus which are populated with the names of the columns of the map.

The id area is the unique identifier of the area and should coincide with the id area we specify for the data. If we specify a region name, the areas shown in the maps and tables in the 'Analysis' page can be filtered by region. This is particularly important when the number of areas in the map is large because in these situations the map may render very slowly. When we filter by region, the map shows fewer areas, only the ones corresponding to a given region, and this permits a faster rendering.

##### *Upload data*

In the second component of the 'Input tab' we upload the data file. The file should be an .csv

file with the following columns:

- area id: an unique identifier of the area. It should coincide with the id area we specify for the map;
- date: date that can be written in year (yyyy), month (yyyy-mm) or day (yyyy-mm-dd) format. Dates should be consecutive;
- population: population in each area and date; and
- cases: number of cases in each area and date.

Optionally, the data can also include columns corresponding up to four covariates such as gender and age group. Data should contain the population and the number cases for all combinations of area id, date and covariates. Here we also need to select the name of the data columns in the corresponding dropdown menus.

### *Select analysis*

After uploading the map and the data files we need to specify several options that will be used in the analyses. First, we specify whether the temporal unit in the data is year, month or day.

Second, we select the date range that we want to consider. Data with dates outside this range are excluded in the analysis. The date range is specified by entering a minimum and maximum dates in a calendar. This calendar only permits the selection of dates expressed in day format (yyyy-mm-dd). If the temporal unit is year or month, the application only considers the year or the year and month of the selected date, respectively.

Finally, the type of analysis is selected. If the analysis selected is spatial, the data is aggregated for all dates and the analyses are performed as there is only one period of time. In the spatio-temporal case all the selected dates are considered.

### *Contents*

When we upload the map and data files, their contents are shown at the end of the 'Inputs' page. This information is useful to know which are the variables of the data so we can correctly specify the inputs required such as the name of the columns, the temporal unit and the date range.

### *Start analysis button*

Once we have uploaded all the files and selected the options required we click the 'Start analysis' button. When we do this, the application checks the files and options entered are correct and calculates the expected counts for each area and date. It also computes the ratio of the observed divided the expected. This ratio is called SIR irrespectively if data refers to mortality or incidence. Then the application redirects to the 'Analysis' page.

### *Example*

In our example we first upload the Ohio map which consist of 88 counties. We need to upload all the files (.dbf, .prj, .shp, .xml and .shx) at the same time. Then we select NAME as area id and also as area name. We do not wish to filter by region so we select "-" as name region.

Then we upload the data. The data consist of the number of deaths and population for each race and gender for each of the counties in Ohio in years 1968 to 1988. The columns of the data file are the following:

- NAME: county name;
- gender: gender (1=male, 2=female);
- race: race (1=white, 2=nonwhite);
- year: year (1968 to 1988);
- y: lung cancer death counts;
- n: population counts.

We select area id as NAME, cases as y, population as n, covariate 1 as gender and covariate 2 as race. In covariates 3 and 4 we select "-".

After that, we specify the required options in the 'Select analysis' component. The temporal unit is year. Since we wish to analyze the data corresponding to years 1981 to 1984, the minimum date is entered as any day in 1981 and the maximum date as any day in 1984. Finally, we select type of analysis as spatio-temporal. The selected options are shown in Figure 1.

## 4.2. Analysis page

In the 'Analysis' page we can visualize the data, perform the statistical analyses, and generate reports. On the top of the page there are four buttons. The first one is 'Edit Inputs' and is used when we wish to return to the 'Inputs' page to modify the analysis options or upload new data. The second button is 'Maps Pop O E SIR'. This button creates plots of the population, observed, expected and SIR variables. The third and fourth buttons are used for estimating the disease risk and their uncertainty, and for the detection of disease clusters, respectively. The 'Analysis' page also contains four tabs called 'Interactive', 'Maps', 'Clusters' and 'Report' that include tables and plots with the results. Details about the use of the buttons and the content of each of the tabs are provided in the following sections.

### *Interactive tab*

The 'Interactive tab' includes three interactive visualizations that show the data and the results: a map, a temporal trend plot and a table. The map displays the values corresponding to the variable and date selected in the dropdown menus located on the left of the page. The variable can be the population, the observed cases, the expected cases or the SIR. If the estimation of risk and the detection of clusters analyses have been carried out, the choices also include the risk and the clusters. The map can be zoomed enabling a better examination of the areas, and when the mouse hovers on any area, that area is highlighted and its information is shown on the upper right corner.

The temporal trend plot shows the values of the variable selected over time. The map and the temporal trend plot are linked and when the mouse hovers on any of the areas in the map, the trend corresponding to the same area is highlighted in the temporal trend plot.

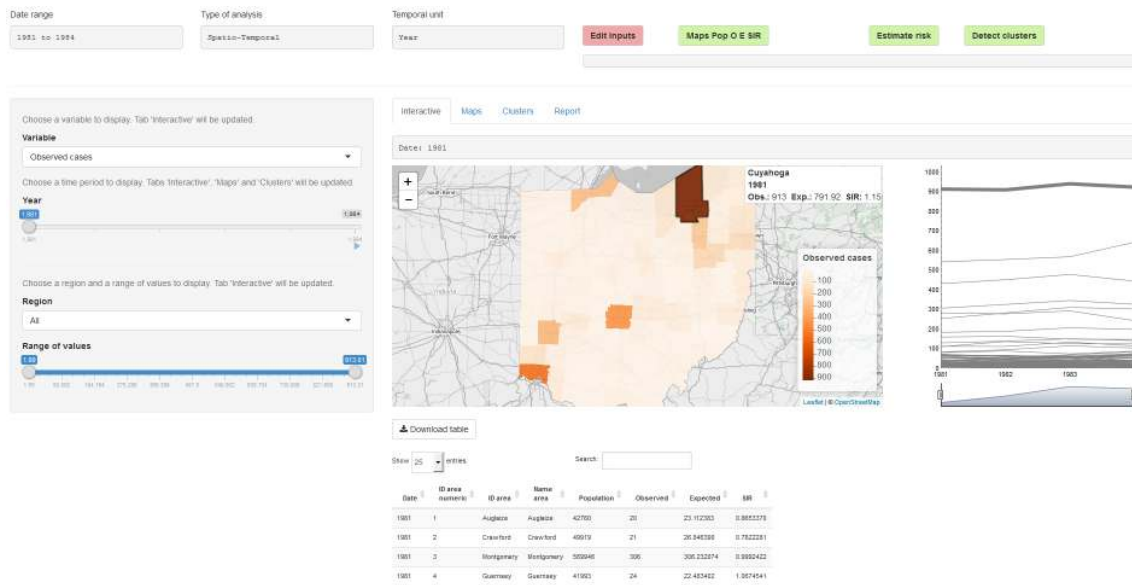


Figure 2: Interactive tab. Results correspond to the Ohio example.

The table displays the information of all areas in the map in the selected date. The user can filter the information by searching a word in any of the columns or in the entire table. It is also possible to sort the rows into ascending or descending order of the values of any of the columns.

In addition, the information shown in the map, the temporal trend plot and the table can be filtered by region and by range of values by selecting an option from the dropdown menu 'Region' and an interval of values from the slider 'Range of values', respectively. In this tab there is also a button called 'Download table' that permits to download the data contained in the table. This option is useful for users that wish to do further analysis or create their own tables and plots.

### Maps tab

The button 'Maps Pop O E SIR' creates maps and temporal trend plots for the variables population, observed, expected and SIR for each of the dates in the study time period. When button 'Estimate risk' is clicked, the application estimates the risk and obtains lower and upper limits of 95% credible intervals showing the uncertainty of the estimates. To obtain these values the application fits a Bayesian model by using **R-INLA**. If the type of analysis selected is spatial or the data pertains to just one date, the BYM model presented by [Besag et al. \(1991\)](#) is used. If the type of analysis selected is spatio-temporal and the data consist of more than one date, the model fitted is the one presented by [Bernardinelli et al. \(1995\)](#).

The 'Maps' tab shows a summary table, maps and temporal trend plots of the population, observed cases, expected cases, SIR, risk and lower and upper limits of 95% credible intervals that were obtained by clicking the 'Map Pop O E SIR' and the 'Estimate risk' buttons (see Figure 3).

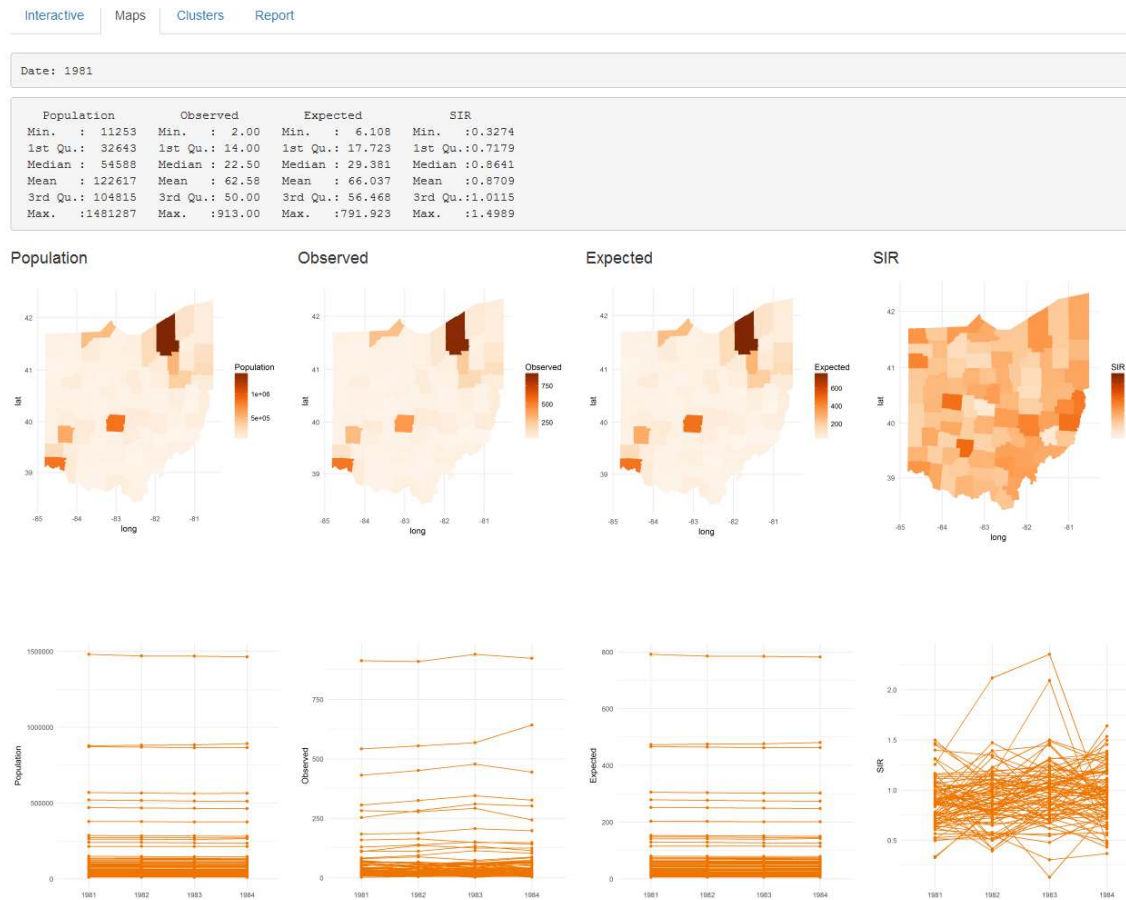


Figure 3: Maps tab. Results correspond to the Ohio example.

### Clusters tab

When clicking the button 'Detect clusters' the program executes the **SaTScan** software to detect spatial or spatio-temporal clusters. First the program creates the input files required by **SaTScan**, then **SaTScan** is executed, and finally the results are processed to generate the plots and tables with the detected clusters. This analysis is performed using 999 Monte Carlo simulations and a level of significance equal to 0.05. The 'Clusters tab' shows, for each of the dates of the period of study, a map with the clusters detected and a plot with all clusters over time. This tab also includes a table with the information relative to each of the clusters such as the central area, the areas included, the ratio of the risk inside over the risk outside, the log likelihood ratio and the significance.

Figure 4 shows the 'Clusters tab' corresponding to the Ohio data. There are 3 clusters detected. The most likely cluster consists of one area located in south west and encompasses years 1983 to 1984. The second cluster is one area located in the north in years 1983 to 1984. The last cluster is in the east and is composed of 5 areas in years 1981 to 1982.

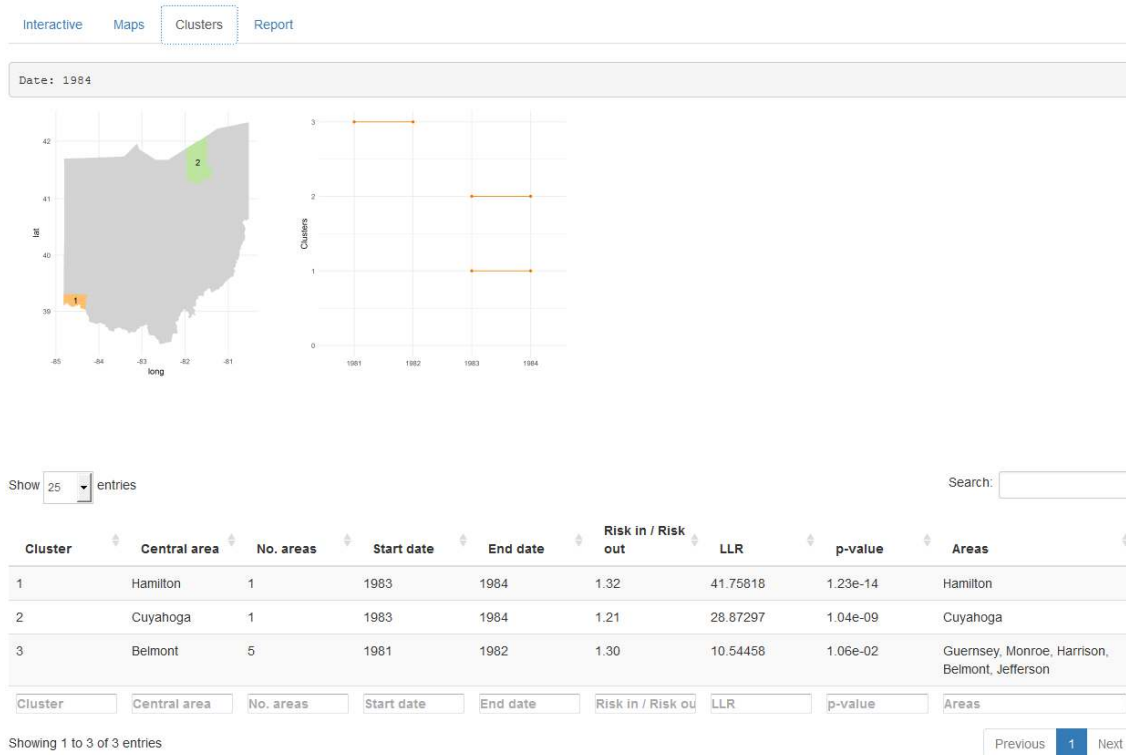


Figure 4: Clusters tab. Results correspond to the Ohio example.

### Report tab

In the 'Report tab' we can download a PDF document showing the results of our analysis. The report includes maps and tables summarizing the variables population, observed, expected, SIR, risk, lower and upper limits of the 95% credible intervals and clusters for each of the periods of time. In this tab there are several radio buttons that permit to choose the variables we want to include in the report. If the variables corresponding to risk or clusters are selected but the risk estimation or the detection of clusters analyses were not carried out, the report will be generated assuming these variables were not selected. Figure 5 shows the report tab where all the variables were selected. This generates a PDF document with all the analyses performed.

### 4.3. Help page

Both the 'Inputs' and the 'Analysis' pages include a 'Help' button that redirects to the 'Help' page. This page shows information about the use of the application, the statistical methodology and the developing tools employed.

## 5. Installation

Interactive Maps Clusters **Report**

Download report

Choose the variables to include in the report. Variables that have not been calculated will not be included.

**Maps**

Population  Observed  Expected  SIR  Risk  2.5 percentile  97.5 percentile  Clusters

**Tables summary**

Population  Observed  Expected  SIR  Risk  2.5 percentile  97.5 percentile

**Table clusters**

Clusters

Figure 5: Report tab.

**SpatialEpiApp** has been implemented in the R package **SpatialEpiApp**. Users can launch the application in R by installing the package and executing the following code:

```
library(SpatialEpiApp)
run_app()
```

The application uses the **R-INLA** package which can be downloaded from <http://www.r-inla.org>. Users wishing to perform clusters analyses need to download **SaTScan** from <http://www.satscan.org> and install it in their computer. They also need to place the **SaTScanBatch64** executable in the `SpatialEpiApp/SpatialEpiApp/ss` folder which is located in the R library path.

Reports use the LaTeX typesetting system. Therefore, users need to have a TeX distribution such as **MiKTeX**, <http://miktex.org>, or **TeX Live**, <http://www.tug.org/texlive>, and the **pandoc** document converter, <http://pandoc.org>, installed in their computer.

## 6. Conclusion

In this paper we presented **SpatialEpiApp**, a Shiny web application for the analysis of spatial and spatio-temporal disease data. The application is easy to use and allows health researchers to perform sophisticated surveillance analyses without the need of having advanced statistical or programming skills. Specifically, it allows to obtain disease risk estimates and their uncertainty by fitting Bayesian models with **R-INLA**, and to detect clusters by using **SaTScan**. It also serves as an exploratory tool for spatial and spatio-temporal disease data since it enables interactive data visualization by using the R packages **Leaflet** for rendering maps, **dygraphs** for plotting time series, and **DataTables** for displaying data objects. **SpatialEpiApp** is an open-source tool which is implemented using R, Shiny and incorporating the functions from several R packages and statistical softwares. Users wishing to publish the results obtained with the Bayesian disease mapping and detection of clusters analyses should provide the proper references to **R-INLA** and **SaTScan**, respectively.

Although the statistical methods **SpatialEpiApp** can perform are very common in surveillance, they are also restrictive. For example, the disease mapping models used are the model introduced by Besag *et al.* (1991) in the spatial case, and the model presented by Bernardinelli *et al.* (1995) in the spatio-temporal setting. In the detection of clusters analysis, **SaTScan** is used with determined options such as Poisson model and clusters with circular shape. Therefore, researchers that need to perform more complex analyses would need to resort to different statistical packages.

Despite these limitations, the methods implemented are very common in health surveillance and we think **SpatialEpiApp** will be very useful for many researchers. Moreover, the application can be easily extended and in future versions we will increase its flexibility enabling more options for disease mapping and the detection of clusters, as well as custom data visualizations. Specifically, we will expand the type of analyses the application can perform so the user can choose among a wider range of models, incorporate covariates, include different types of spatial and temporal random effects, and choose among different shapes of clusters. Another extension will be the inclusion of statistical methods and visualizations for analysing point data. This will make the application useful for people wishing to analyze geostatistical or point process data.

## References

- Allaire J, Cheng J, Xie Y, McPherson J, Chang W, Allen J, Wickham H, Atkins A, Hyndman R (2016). *rmarkdown: Dynamic Documents for R*. R package version 1.2, URL <https://CRAN.R-project.org/package=rmarkdown>.
- Attali D (2016). *shinyjs: Easily Improve the User Experience of Your Shiny Apps in Seconds*. R package version 0.8, URL <https://CRAN.R-project.org/package=shinyjs>.
- Bernardinelli L, Clayton DG, Pascutto C, Montomoli C, Ghislandi M, Songini M (1995). “Bayesian analysis of space-time variation in disease risk.” *Statistics in Medicine*, **14**, 2433–2443.
- Besag J, York J, Mollié A (1991). “Bayesian image restoration with applications in spatial statistics (with discussion).” *Annals of the Institute of Statistical Mathematics*, **43**, 1–59.
- Bivand R, Lewin-Koh N (2016). *maptools: Tools for Reading and Handling Spatial Objects*. R package version 0.8-39, URL <https://CRAN.R-project.org/package=maptools>.
- Bivand R, Piras G (2015). “Comparing Implementations of Estimation Methods for Spatial Econometrics.” *Journal of Statistical Software*, **63**(18), 1–36.
- Bivand R, Rundel C (2016). *rgeos: Interface to Geometry Engine - Open Source (GEOS)*. R package version 0.3-21, URL <https://CRAN.R-project.org/package=rgeos>.
- Chang W, Cheng J, Allaire J, Xie Y, McPherson J (2016). *shiny: Web Application Framework for R*. R package version 0.14.1, URL <https://CRAN.R-project.org/package=shiny>.
- Cheng J, Xie Y (2016). *leaflet: Create Interactive Web Maps with the JavaScript 'Leaflet' Library*. R package version 1.0.1, URL <https://CRAN.R-project.org/package=leaflet>.



- Dwass M (1957). “Modified randomization tests for nonparametric hypotheses.” *Annals of Mathematical Statistics*, **28**, 181–187.
- Elliott P, Wakefield J, Best N, David Briggs e (2000). *Spatial Epidemiology: Methods and Applications*. Oxford University Press.
- Gelfand AE, Diggle PJ, Fuentes M, Guttorp P (2010). *Handbook of Spatial Statistics*. Chapman & Hall/CRC, Boca Raton, FL.
- Hagan JE, Moraga P, Costa F, Capian N, Ribeiro GS, Jr EAW, Felzemburgh RDM, Reis RB, Nery N, Santana FS, Fraga D, dos Santos BL, Santos AC, Queiroz A, Tassinari W, Carvalho MS, Reis MG, Diggle PJ, Ko AI (2016). “Spatiotemporal Determinants of Urban Leptospirosis Transmission: Four-Year Prospective Cohort Study of Slum Residents in Brazil.” *Public Library of Science: Neglected Tropical Diseases*, **10**(1), e0004275.
- Kim AY, Wakefield J (2016). *SpatialEpi: Methods and Data for Spatial Epidemiology*. R package version 1.2.2, URL <https://CRAN.R-project.org/package=SpatialEpi>.
- Kulldorff M (1997). “A spatial scan statistic.” *Communications in Statistics - Theory and Methods*, **26**(1), 1481–1496.
- Kulldorff M (2006a). *SaTScan(TM) v. 7.0. Software for the spatial and space-time scan statistics*. URL <http://www.satscan.org/>.
- Kulldorff M (2006b). “Tests of spatial randomness adjusted for an inhomogeneity: A general framework.” *Journal of the American Statistical Association*, **101**(475), 1289–1305.
- Kulldorff M, Nagarwalla N (1995). “Spatial disease clusters: detection and inference.” *Statistics in Medicine*, **14**, 799–810.
- Lawson AB (2009). *Bayesian Disease Mapping: Hierarchical Modeling In Spatial Epidemiology*. Chapman and Hall/CRC.
- Lawson AB, Browne WJ, Rodeiro CLV (2003). *Disease Mapping with WinBUGS and MLWin*. Wiley.
- Lawson AB, Kleinman K (2005). *Spatial and Syndromic Surveillance for Public Health*. Wiley.
- Lindgren F, Rue H (2015). “Bayesian Spatial Modelling with **R-INLA**.” *Journal of Statistical Software*, **63**(19), 1–25.
- McIlroy D, Brownrigg R, Minka TP, Bivand R (2015). *mapproj: Map Projections*. R package version 1.2-4, URL <https://CRAN.R-project.org/package=mapproj>.
- Moraga P, Cano J, Baggaley RF, Gyapong JO, Njengaf SM, Nikolay B, Davies E, Rebollo MP, Pullan RL, Bockarie MJ, Hollingsworth TD, Gambhir M, Brooker SJ (2015). “Modelling the distribution and transmission intensity of lymphatic filariasis in sub-Saharan Africa prior to scaling up interventions: integrated use of geostatistical and mathematical modelling.” *Parasites & Vectors*, **8**, 560.
- Moraga P, Montes F (2011). “Detection of spatial disease clusters with LISA functions.” *Statistics in Medicine*, **30**, 1057–1071.

- Neuwirth E (2014). *RColorBrewer: ColorBrewer Palettes*. R package version 1.1-2, URL <https://CRAN.R-project.org/package=RColorBrewer>.
- R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org>.
- Rue H, Martino S, Chopin N (2009). “Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations.” *Journal of the Royal Statistical Society, Series B*, **71**(2), 319–392.
- Ryan JA, Ulrich JM (2014). *xts: eXtensible Time Series*. R package version 0.9-7, URL <https://CRAN.R-project.org/package=xts>.
- Stelling J, Yih WK, Galas M, Kulldorff M, Pichel M, Terragno R, Tuduri E, Espetxe S, Binsztein N, O’Brien TF, Platt R, Group WAC (2010). “Automated use of WHONET and SaTScan to detect outbreaks of Shigella spp. using antimicrobial resistance phenotypes.” *Epidemiology and Infection*, **138**(6), 873–883.
- Thacker SB, Berkelman RL (1988). “Public health surveillance in the United States.” *Epidemiologic Reviews*, **10**, 164–90.
- Vaidyanathan R, Xie Y, Allaire J, Cheng J, Russell K (2016). *htmlwidgets: HTML Widgets for R*. R package version 0.7, URL <https://CRAN.R-project.org/package=htmlwidgets>.
- Vanderkam D, Allaire J, Owen J, Gromer D, Shevtsov P, Thieurmel B (2016). *dygraphs: Interface to ‘Dygraphs’ Interactive Time Series Charting Library*. R package version 1.1.1.3, URL <https://CRAN.R-project.org/package=dygraphs>.
- Wakefield JC, Morris SE (2001). “The Bayesian Modeling of Disease Risk in Relation to a Point Source.” *Journal of the American Statistical Association*, **96**(453), 77–91.
- Waller LA, Gotway CA (2004). *Applied Spatial Statistics for Public Health Data*. Wiley, New York.
- Wickham H (2009). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. ISBN 978-0-387-98140-6. URL <http://ggplot2.org>.
- Wickham H, Francois R (2016). *dplyr: A Grammar of Data Manipulation*. R package version 0.5.0, URL <https://CRAN.R-project.org/package=dplyr>.
- Xie Y (2015). *Dynamic Documents with R and knitr. 2nd edition*. Chapman and Hall/CRC.
- Xie Y (2016). *DT: A Wrapper of the JavaScript Library ‘DataTables’*. R package version 0.2, URL <https://CRAN.R-project.org/package=DT>.

**Affiliation:**

Paula Moraga  
Lancaster Medical School  
Lancaster University

Lancaster, LA1 4YG, United Kingdom

E-mail: [p.e.moraga-serrano@lancaster.ac.uk](mailto:p.e.moraga-serrano@lancaster.ac.uk)

URL: <http://Paula-Moraga.github.io>