ShinyGeode User Manual

Michael Otterstatter

2022-02-10

Overview

This manual describes how to access the functionality of the R package **geode** using the friendly graphical interface **ShinyGeode**. This interface provides most of the functions available in **geode** through point-and-click features. No R coding is required.

ShinyGeode was created using the **shiny** package (https://shiny.rstudio.com) and the freely available, open source software R (https://cran.r-project.org/).

Prerequisites

The **ShinyGeode** interface is intended to be used with R version 4.0.0 (April 2020) or later and RStudio version 1.2.1335 or later. Earlier versions of R/RStudio may not support all of the functionality of this tool. The latest versions of R and RStudio can be freely downloaded from https://cran.r-project.org/ and https://www.rstudio.com/products/rstudio/download/.

Downloading and installing

The easiest way to install **ShinyGeode** is directly from your R/RStudio session. Running the following script in R will first install (or update) the **geode** package (which is required to run **ShinyGeode**) and then install (or update) the **ShinyGeode** package:

```
install.packages("remotes")
remotes::install_github("cpacc/geode")
remotes::install_github("cpacc/ShinyGeode")
```

This process may also install other R packages ('dependencies') used by **geode** or by **ShinyGeode** from an online repository. These may take several minutes to install and you may be asked i) if you want to update certain packages that have more recent versions available, and ii) if you wish to install from sources the packages which need compilation. Both of these options are recommended, but probably not essential – and could take a while if many packages need updating.

Note that installation of **geode** and **ShinyGeode** and their dependencies only needs to be done once.

1 Using ShinyGeode

The following is a "quick start" guide to using **ShinyGeode**. Additional details can be found in the later chapters of this manual.

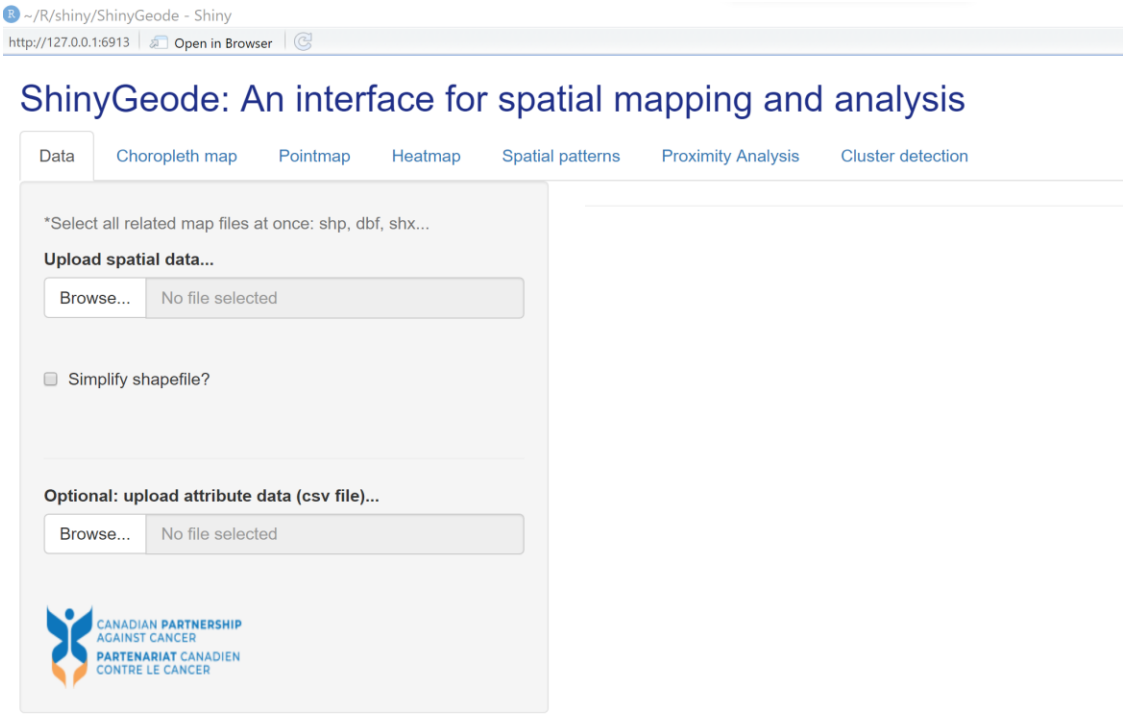
1.1 Structure of the interface

ShinyGeode is organized as a series of tabs across the top of the interface. Each tab is a separate screen with a separate set of geode functions.

Assuming you have already installed the geode and ShinyGeode packages (as described at the beginning of this manual) you need only load ShinyGeode by running the following lines in your R/RStudio session:

```
library(ShinyGeode)
ShinyGeode()
```

ShinyGeode will then open, displaying the 'Data' page:



The other pages (functions) can be access simply by clicking the links at the top of the interface: 'Choropleth map,' 'Pointmap,' 'Heatmap,' 'Spatial patterns,' 'Proximity analysis,' and 'Cluster detection.'

Each page in ShinyGeode is divided into two halves: on the left, there is a set of user options in a grey shaded area; on the right, there is an open space where imported data tables and generated maps will appear.

Most functions are accessed using drop-down menus, sliders and checkboxes. Most functions will not appear until spatial data (and, optionally, attribute data) have been imported.

The following section provides a simple example of importing and mapping data.

1.2 Getting started

For illustration purposes, we make use of the example data provided with the geode package at:

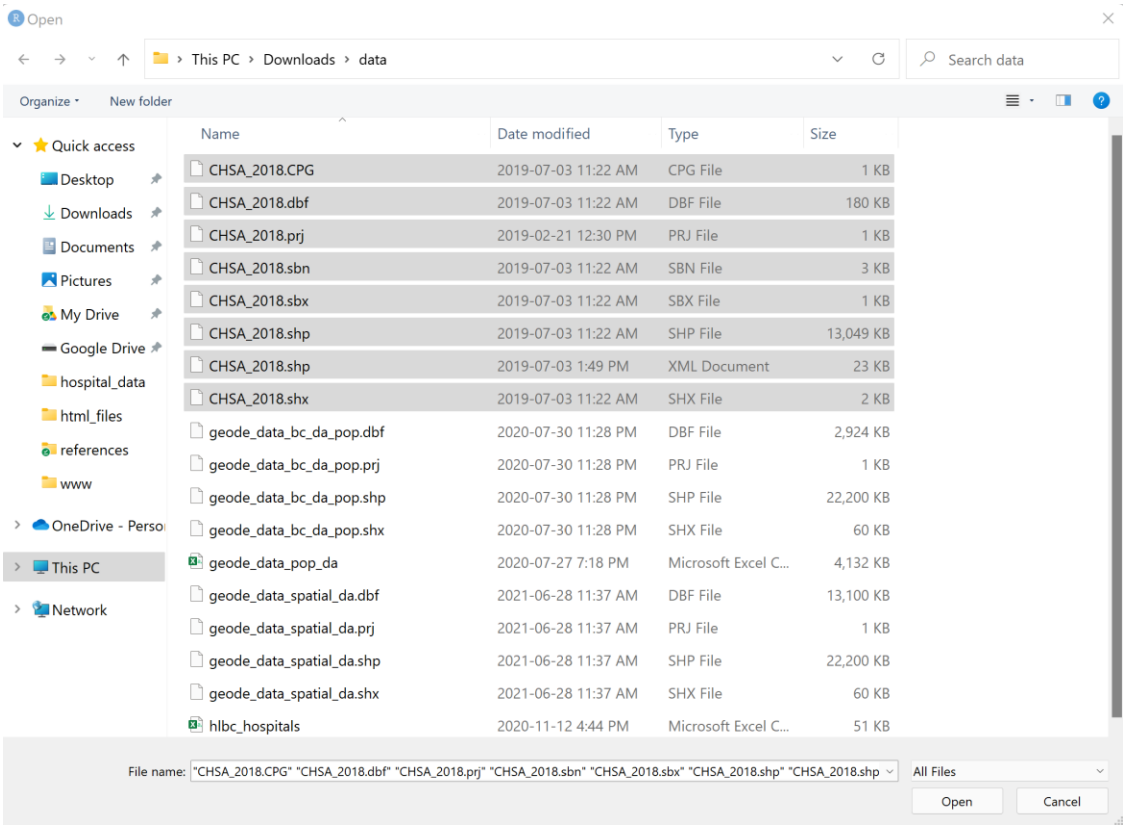
https://github.com/cpacc/geode/blob/main/zip/example_data.7z

These example datasets include the spatial boundary files for the Community Health Service Areas of BC (CHSA.shp), the Dissemination Areas of BC (geode_data_bc_da_pop.shp) and the locations of hospitals throughout BC (hlbc_hospitals.csv).

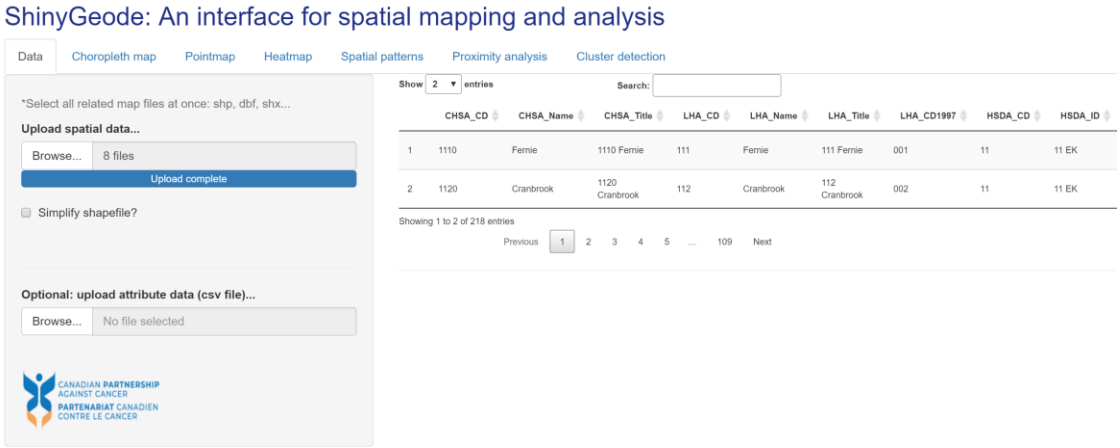
Importing data

To begin, click on the *Browse…* button underneath 'Upload spatial data….' This will open a file explorer window and allow you to navigate to the location where you have saved the example datasets.

In order to upload spatial data, we must **import all of the multiple files (including .shp, .prj, .dbf, and so on) that correspond to our spatial boundaries**. In the case of the CHSA files, we select all of the eight files beginning with 'CHSA_2018' and click Open, or hit Enter.

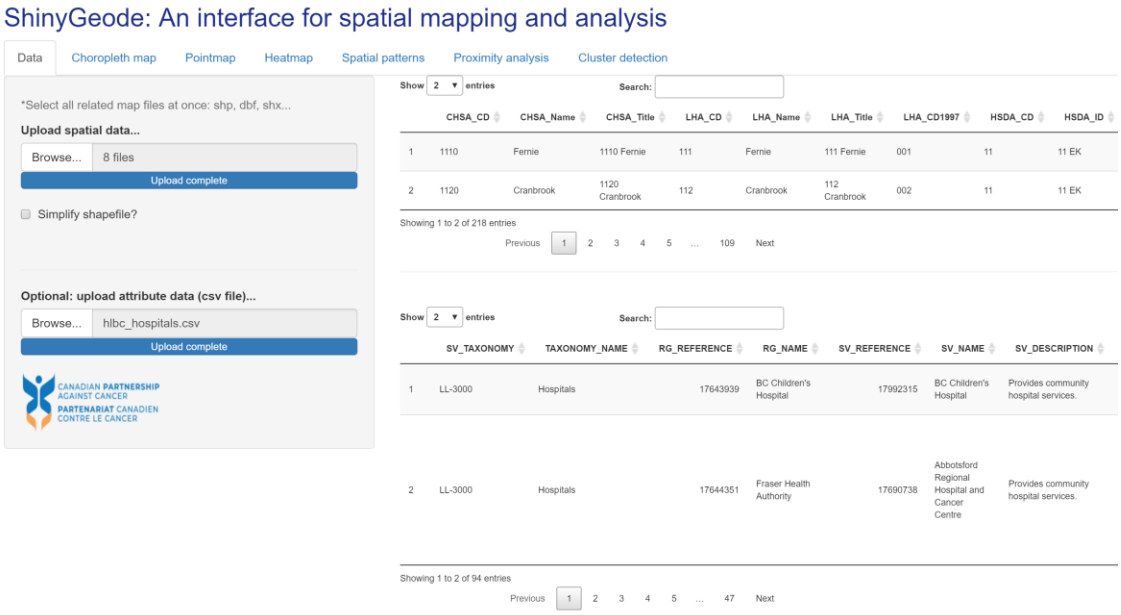


ShinyGeode will import these files and automatically display a data table in our 'Data' screen.



Note that this table has a variety of functions for exploring the data, including left-to-right scrolling, a keyword search box, and sorting buttons (ascending or descending) next to each column.

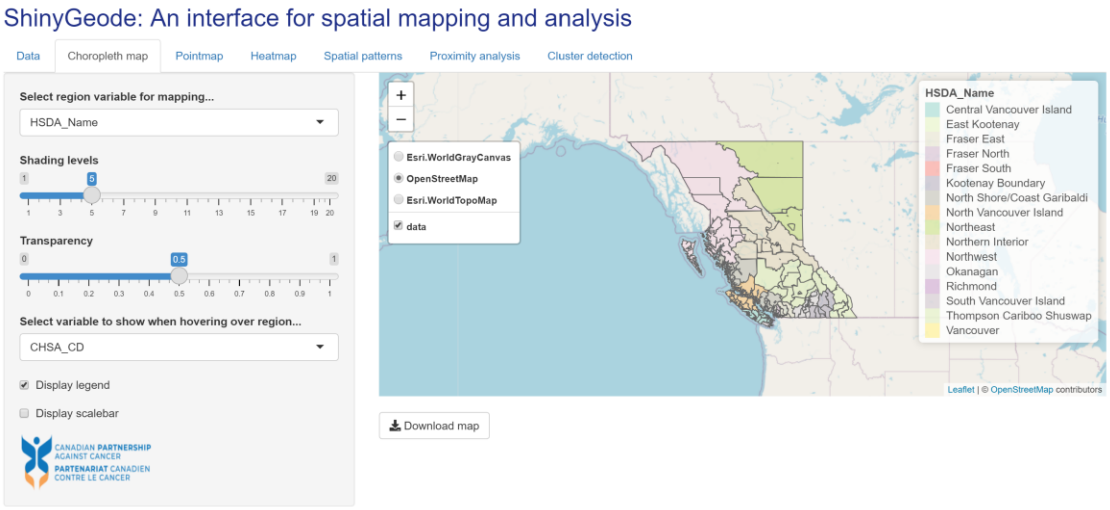
Similarly, we can import attribute data by clicking on the *Browse…* button underneath 'Optional: upload attribute data (csv file)….' Attribute data are assumed to be in .csv files. Selecting and importing the hospital location data file (hlbc_hospitals.csv) will generate a second dynamic data table in our Data screen. Again, this table is searchable and sortable for the purposes of quickly exploring the file.



Note that we can simplify imported shape files by clicking the checkbox 'Simplify shapefile?' underneath the *Browse…* button. Simplification removes vertices along complex geographic boundaries, thereby resulting in a much smaller, but less detailed, data object. This is useful for reducing processing time and for experimenting with large shapefiles; however, simplified files are not recommended when conducting spatial analysis.

Creating a map

After loading data, click on the 'Choropleth map' button at the top of the interface. A map will be generated and displayed once a variable has been selected from the input dataset for mapping. Below, a map is shown based on the 'HSDA_Name' variable that was selected from the dropdown menu at the top of the screen.

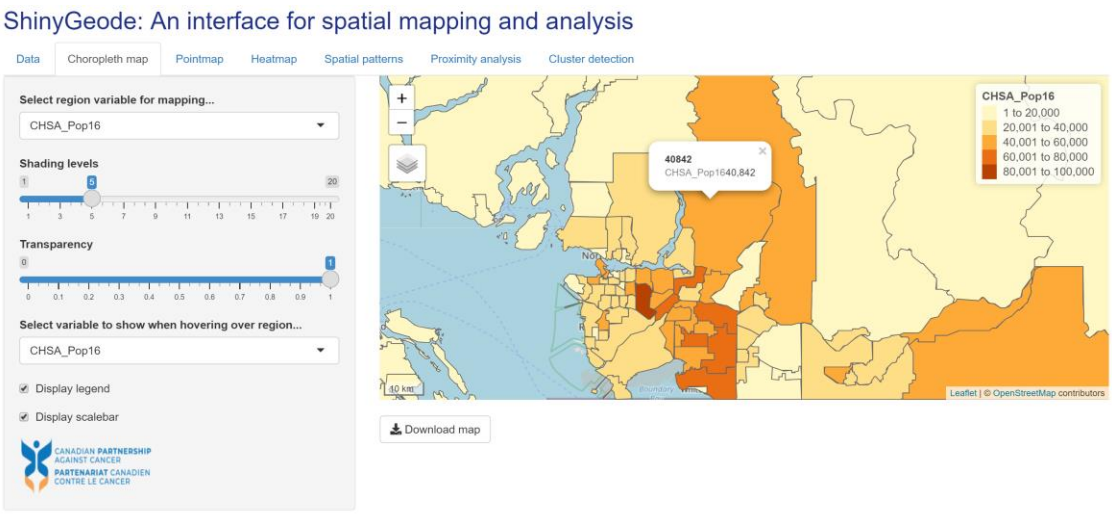


The time required to generate and display a map depends on the size of the shapefile and the processing power of the computer. Most maps will appear a few seconds after a region variable has been selected from mapping. Maps based on simplified shapefiles will typically appear faster than those based on unsimplified files.

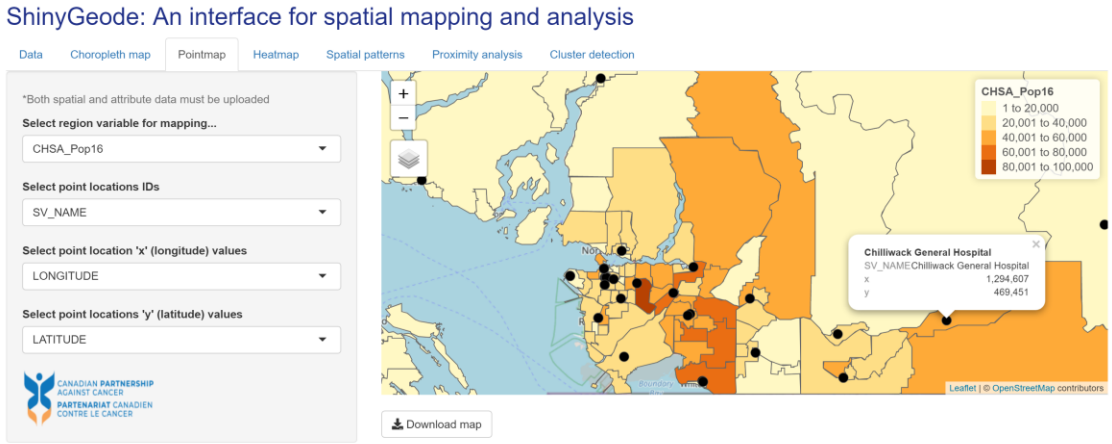
Note in the above example that a menu is visible near the centre of the screen, with options for selecting the background layer (e.g., OpenStreetMap, Esri.WorldTopoMap, etc.) and a checkbox for whether or not the imported data should be displayed.

In ShinyGeode, maps are fully interactive allowing zooming (+ or - buttons), scrolling (any direction) and hover-over features. Sliders are shown along the left of the screen for controlling the number of shading levels shown in the choropleth map and the transparaency of the shading. Further, the information shown while hovering over a region can be selected as a variable from the imported dataset using the dropdown menu. Finally, two checkboxes allow you to control whether or not a legend and/or a scalebar are shown on the map.

In the map shown below, the CHSA population size variable has been selected for mapping, the shading levels have been decreased to 5, the transparency has been changed to 1 (opaque), and a scalebar has been added to the map. The selected hover-over variable is the population size, which in the example below is displayed for a region in the centre of the map that was clicked on.



Given that attribute data (hospital locations) were also uploaded in the 'Data' screen, the above map could also be generated in the 'Pointmap' screen but with hospital locations shown as black circles. For pointmaps, there are additional dropdown menus for the variables identifying the point locations (here the SV_NAME variable contains the hospital names), and their x (longitude) and y (latitude) coordinates. By default, the point location ID variable provides the information that is shown when you hover over or click on a point location. In the example below, the Chilliwack General Hospital location was clicked on, revealing the hospital name its x-y coordinates.



Exporting a map

In ShinyGeode, maps can be downloaded with a simple button click. Once a map is generated, a 'Download map' button will appear – clicking on this button opens a file explorer window, allowing you to name the downloaded file and navigate to the location where the file should be saved. If the file explorer window does not open immediately, it is likely that the underlying file is large and takes time for the computer to process.

ShinyGeode is principally a tool for interacting with and understanding spatial data. Maps are saved in their interactive form, as an .html file (i.e., webpage), retaining all of the information and interactivity of the original version. High quality static images can also be saved using simple screen capture or snipping tools (e.g., like the images in this User Manual).

Note that interactive maps may result in large file sizes. The pointmap above, for example, results in a 30MB download. Most of this size is due to the underlying shapefile - simplifying this shapefile during data import reduces the download size to only 2MB.

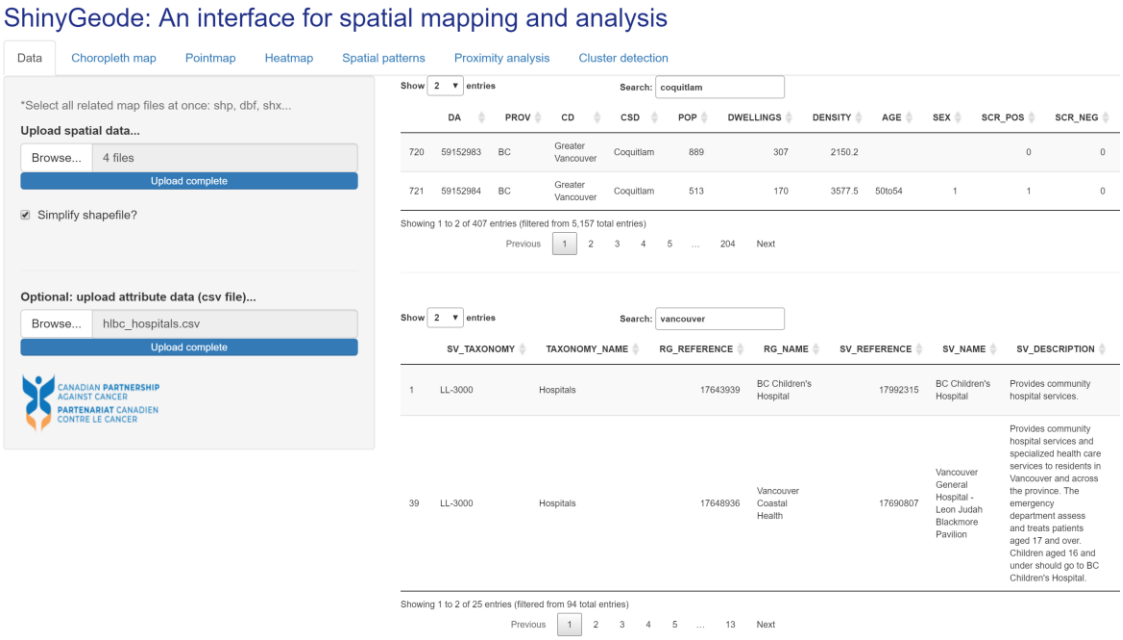
2 Interface pages

This section briefly describes each of the ShinyGeode pages (tabs) and their associated functionality from the underlying geode package.

2.1 Data

The Data screen allows users to efficiently import both spatial and attribute files for exploration, mapping and analysis.

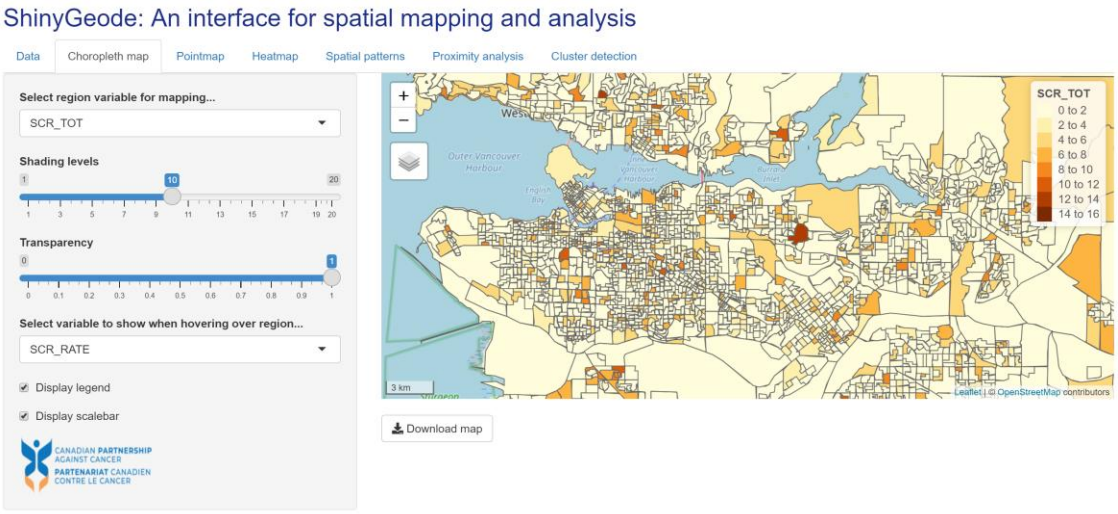
This screen utilizes the `geo_import()` function from the geode package. This screen allows importation of both geographic boundary data, utilizing the `filetype = 'spatial'` option, and attribute data, utilizing the `filetype = 'attribute'` option. Both simplification (`simplify =`) and validity checking and fixing (`validity_check =`) options are retained in the ShinyGeode interface.



2.2 Choropleth map

The Choropleth map screen allows users to generate interactive, shaded areal maps showing spatial patterns of interest by health or administrative regions.

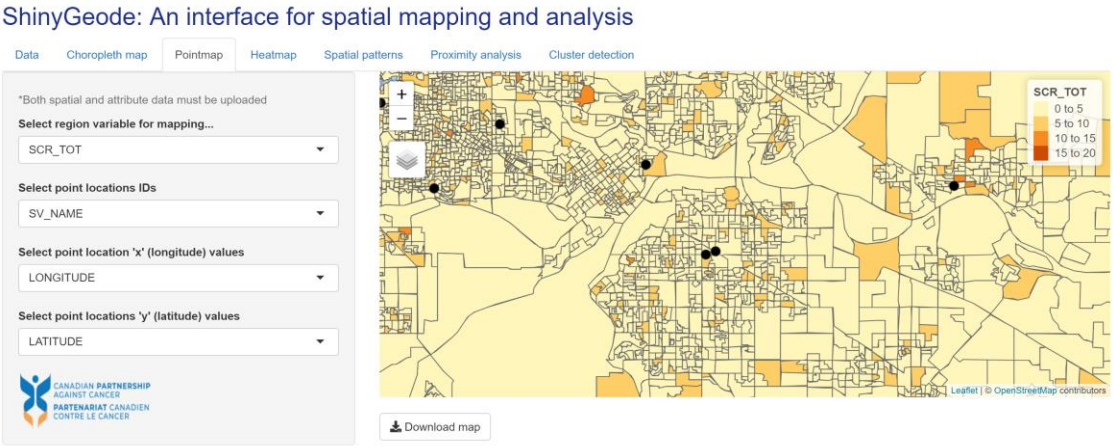
This screen utilizes the `geo_plot()` function from the geode package. Specifically, this screen makes use of the `plot_type = 'choropleth` option from this package, while providing several of the key user options including `levels =`, `transparency =`, `hover_id =` and `scale_bar =`.



2.3 Pointmap

The Pointmap screen allows users to add point locations of any type choropleth (or blank) maps in order to visualize patterns in relation to, for example, service locations of interest.

This screen utilizes the `geo_plot()` function from the geode package. Specifically, this screen makes use of the `plot_type = 'pointmap` option and allows users to define point locations via the underlying `points_col =` argument.



2.4 Heatmap

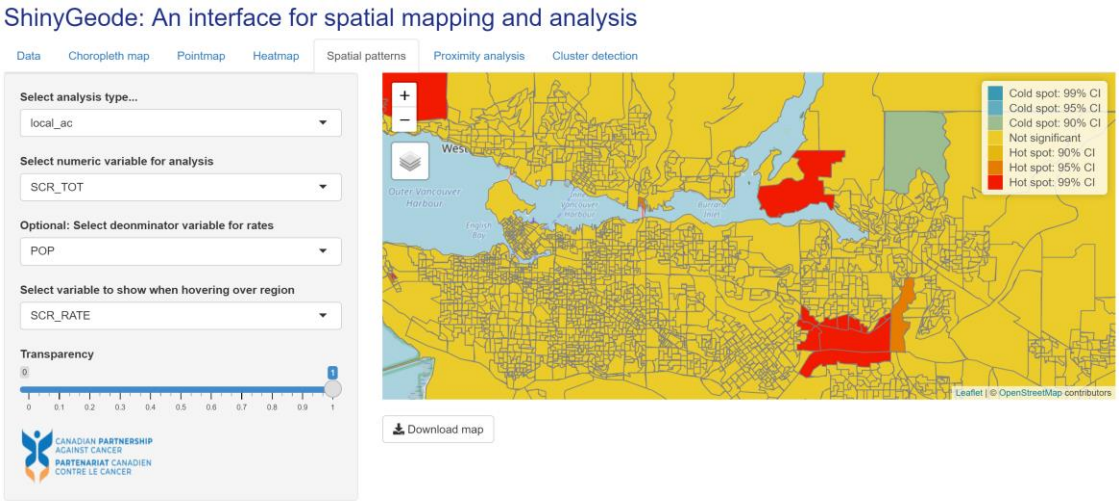
The Heatmap screen allows users to visualize the density of measures of interest in a mapping context. This may be relevant, for example, in the assessment of exposure levels near regions or point location of interest.

This screen utilizes the `geo_plot()` function from the geode package. Specifically, this screen makes use of the `plot_type = 'heatmap` option and allows users to define point locations via the underlying `points_col =` argument. In this prototype version of ShinyGeode, only static heatmaps are available (interactive versions will be released in subsequent updates).

2.5 Spatial patterns

The Spatial patterns screen employs local and global spatial autocorrelation analysis to identify patterns of clustering, such as significant hotspots or coldspots, in spatial data.

This screen utilizes the `geo_calculate()` function from the geode package. An analysis variable and optional denominator are selected utilizing the `var =` and `denom =` options, respectively. The `hover_id =` and `transparency =` options are retained in the ShinyGeode interface.

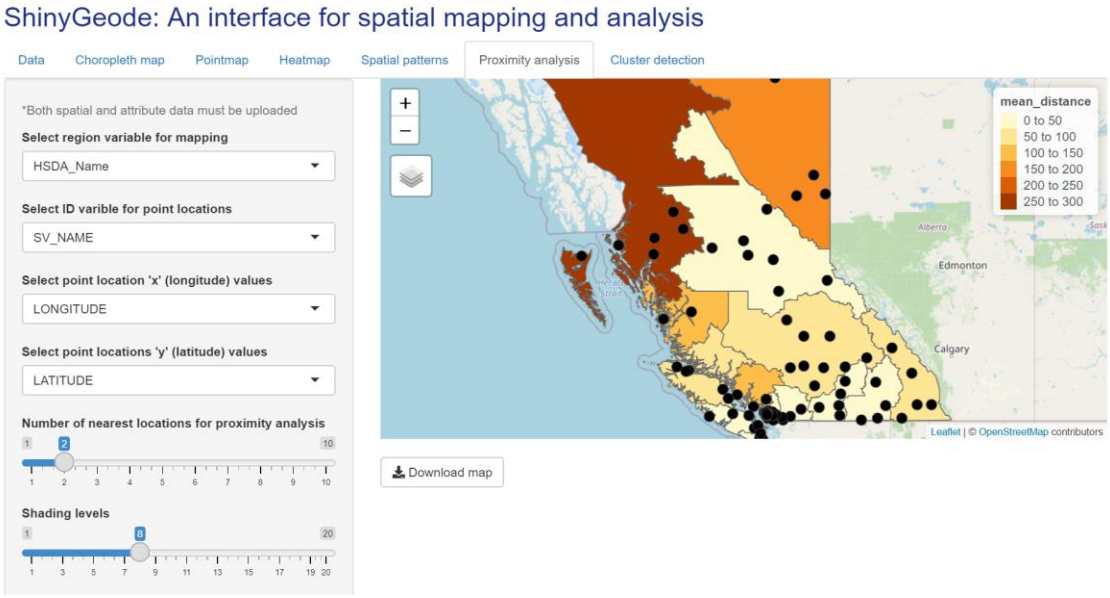


2.6 Proximity analysis

The Proximity analysis screen allows users to conduct a descriptive spatial analysis of the proximity of regional populations to point locations of interest. This analysis may be relevant, for example, in the assessment of geographic variation in access to service locations.

This screen utilizes the `geo_distance()` function from the geode package. The point location data utilize the `points_col =` argument. The number of nearest locations for analysis is based on the `n_nearest =` option and the shading levels utilize the `levels =`

option. The ShinyGeode interface allows the additional functionality to select the x and y coordinate columns from the data, rather than requiring these to be fixed as in the geode package.



2.7 Cluster detection

The Cluster detection screen allows users to conduct a formal cluster analysis using the Kulldorff methods. Statistically significant clusters are identified and shown in the output maps for an geographic attribute data of interest.

This screen utilizes the `geo_detect()` function from the geode package. The analysis counts and population sizes are based on the `counts =` and `pop =` arguments, respectively. The interface also retains the `transparency =` and `hover_id =` options. Both the Kulldorff Binomial and Kulldorff Poisson detection methods are available, utilizing the underlying `method =` option.