CSIS Discussion Paper No. **127**

`seg`: Implementing recent developments in the measurement of segregation

Seong-Yun Hong *, David O'Sullivan **, & Yukio Sadahiro *

June 2014

***** Center for Spatial Information Science, The University of Tokyo
** Department of Geography, The University of California, Berkeley

Center for Spatial Information Science, The University of Tokyo
5-1-5, Kashiwanoha, Kashiwa-shi, Chiba 277-8568, Japan
E-mail: yun.hong@csis.u-tokyo.ac.jp

seg: Implementing recent developments in the measurement of segregation

Abstract

Despite the advances in the measurement of segregation over the last few decades, recently developed segregation indices have been rarely used in the literature, primarily due to the computational difficulties. The calculation procedure of these measures tends to be more sophisticated than the traditional counterparts, and it often involves spatial data processing using Geographic Information Systems techniques. Although considerable efforts have been made in recent years to implement some of the newly proposed approaches, either it does not incorporate important improvements in the field, or it requires commercial software to run. In this paper, we describe our contribution to the implementation of segregation measures in R, open-source software environment for statistical computing and graphics. Our implementation has several merits over the existing tools. First, we provide coercion methods that enable the transformation of output from the implemented functions into more general R classes. This feature allows using thousands of standard and modern statistical techniques, as well as facilities for data manipulation and visualisation, for the post-processing of the results. Second, the implemented functions work with a wide range of input parameters, and most of them have carefully chosen defaults, which will perform fine in many situations. This provides greater flexibility and control over the calculation procedure for advanced users, while ensuring that less experienced R users can still use the functions without too much difficulty. Third, our implementation does not require commercial software to operate, so it is accessible to a wider group of people.

Keywords: segregation; segregation measures; R;

Introduction

The measurement of segregation has been a topic of debate and discussion among sociologists and geographers for decades (Johnston, Poulsen, & Forrest, 2010; Kramer, Cooper, Drews-Botsch, Waller, & Hogue, 2010; Massey & Denton, 1988; Peach, 2009, 2010; Wong, Reibel, & Dawkins, 2007). Many measures have been proposed over the last half century, to capture various dimensions of this complex social phenomenon, but only a few of them have been regularly used in the segregation literature. Some of the indices have not been adopted in practice because they overlap with the existing ones to a large extent, providing little new insight into the patterns of segregation (Massey & Denton, 1988), and some have not been chosen due to their methodological flaws and ambiguity in interpretation (Johnston, et al., 2010; Peach, 2009).

There are, however, several methods that are generally acknowledged to have theoretical advantages over the conventional ones but have been rarely used, primarily due to the computational difficulties. Recently developed spatial indices might be cases in point: the calculation procedure of these measures tends to be more sophisticated than the traditional counterparts, and it often involves spatial data processing using Geographic Information Systems (GIS) techniques (Wong, 2003). Although considerable efforts have been made in recent years to implement these spatial indices (Apparicio, Martori, Pearson, Fournier, & Apparicio, 2013; Reardon, et al., 2009; Wong, 2003), either it does not incorporate important improvements in the field, or it requires commercial software to run, which is not available to the public.

To address this problem, we have developed an R package **seg** that provides facilities for a wider variety of segregation measures. R is a multi-platform, open-source software environment for statistical computing and graphics (R Core Team, 2014), so it is accessible to almost all members of the academic and research communities. Furthermore, since R offers numerous powerful statistical and graphical tools, the manipulation and visualisation of the spatial data, as well as the post-processing of the results can be readily performed without exporting it to another data format.

This paper is mainly concerned with describing the structure and functions of the **seg** package. In the next section, we present the definitions of the segregation measures currently included in this extension package and explain briefly how these are implemented. The subsequent section evaluates the reliability and computational efficiency of the implemented functions with a set of hypothetical segregation patterns: the idealised landscapes are adopted from Morrill (1991) and Wong (1993), as they are intended to test the accuracy of the associated functions through regression testing. This paper concludes with a discussion about the limitations of the current work and future directions for development.

Implementation

The measures of segregation can be classified based on a number of different criteria. Massey and Denton, for instance, examined 20 indices available at that time and grouped them into five categories, based on their correlations to each other (Massey & Denton, 1988). The indices may also be distinguished into *spatial* and *non-spatial* indices, depending on whether the calculation is sensitive to the spatial arrangement of the population. One well-known example of the latter is the index of dissimilarity developed by Duncan and Duncan (1955), while a considerable number of more recent methods, such as the set of measures proposed by Reardon and O'Sullivan (2004), belong to the former.

In this paper, we classify the segregation measures under two headings, namely, *zone-based* and *surface-based* measures, based on the types of input data required. Zone-based measures use aggregated population counts for their calculation, and surface-based measures utilise a continuous population density surface to minimise the so-called modifiable areal unit problem (MAUP) (Openshaw, 1984). We distinguish segregation measures in this manner because the amount of information required for the calculation significantly differs between the two, and hence the computational steps are also very different. Table 1 presents the zone-based and surface-based segregation measures implemented in the **seg** package; each of these will be discussed in the following subsections.

Zone-based measures

The calculation of the zone-based measures is relatively straightforward: most can be calculated by hand, or using a simple spreadsheet program. There are, however, several more complex methods that demand extensive data preparation. The **seg** package provides tools for some of these methods, including the index of dissimilarity (Duncan & Duncan, 1955) and its spatially-modified forms (Morrill, 1991; Wong, 1993), the index of spatial proximity (White, 1983), and the concentration profile (Poulsen, Johnston, & Forrest, 2002). In this section, we present a brief introduction to these indices and their implementation in R. More detailed descriptions of the methods are given in the corresponding original papers.

The index of dissimilarity, *D*, is one of the most widely used measures in the segregation literature. For the study region consisting of *n* census tracts, *D* is defined as:

$$D = \frac{1}{2}\sum\nolimits_{i=1}^{n}\left|\frac{x_i}{X} - \frac{y_i}{Y}\right| \tag{1}$$

where *X* and *Y* denote the total population counts of two population groups, and x_i and y_i are the local populations in the census tract *i*. Although *D* itself is non-spatial, this can be adjusted to reflect the spatial distribution of the population. For example, Morrill (1991) suggested adding a spatial term to the equation (1), so it becomes:

$$DM = D - \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\left|z_i - z_j\right|c_{ij}}{\sum_{i=1}\sum_{j=1}c_{ij}} \tag{2}$$

where z_i and z_j are the proportions of the minority population in the census tracts *i* and *j* (i.e., $z_i = x_i / (x_i + y_i)$ and $z_j = x_j / (x_j + y_j)$), respectively, and c_{ij} denotes an element at (*i*, *j*) in a contiguity matrix **C**, which becomes one only if *i* and *j* are adjacent.

This spatially-adjusted version of *D* can be further supplemented by taking into account additional geometric features of the spatial units, which might influence individuals' accessibility to neighbouring areas. Wong (1993) proposed replacing the binary contiguity matrix in (2) with a distance-weighted matrix, **W**, whose (*i*, *j*) element is denoted by w_{ij}:

$$DW = D - \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\left|z_i - z_j\right|w_{ij}}{\sum_{i=1}\sum_{j=1}w_{ij}} \tag{3}$$

where

$$w_{ij} = \frac{d_{ij}}{\sum_j d_{ij}}$$

and d_{ij} is the shared boundary length between the census tracts *i* and *j*. He further argues that if the area and perimeter of the census tracts are known, they can be incorporated into *D* (Wong, 1993):

$$DS = D - \frac{1}{\sum_i\sum_j w_{ij}}\sum\nolimits_i\sum\nolimits_j\left(\left|z_i - z_j\right|w_{ij}\frac{(P_i/A_i + P_j/A_j)}{2\max\limits_{1\le i\le n}(P_i/A_i)}\right) \tag{4}$$

where P_i and A_i are the perimeter[1] and area of the census tract *i*, and $\max_{1\le i\le n}(P_i/A_i)$ is the maximum possible ratio in the entire region.

These three spatial associates of *D* (i.e., equations (2) – (4)) might be more realistic representation of residential segregation, but they—particularly, *DW* and *DS*—are rather complicated to calculate. To facilitate the use of these extended indices, we have implemented them in the **seg** package, as a single function called `dissim()`. Once the function is called, it first estimates the traditional index of dissimilarity, and then the spatial term (i.e., the second component of (2), (3) and (4)) is deducted from it. Which index to calculate is controlled by input arguments, because the calculation procedures of *DM*, *DW* and *DS* are essentially identical: the difference lies in the amount of spatial information required, not in the way in which the spatial term is calculated.

The first two arguments to this function, `x` and `data`, specify the data for which the index is calculated: `x` should be of class `SpatialPolygons` (or one that extends it) representing the study region, and `data` should be a *n*-by-2 table, where *n* is the number of census tracts and the two columns contain the population counts of mutually exclusive groups. If `data` is given, `x` becomes optional. If `x` is present, and if it includes a data frame, `data` may be omitted. However, at least one of these arguments must be supplied.

The third argument to `dissim()` is `nb`. It should be a *n*-by-*n* matrix, sorted in the same order as `data`, whose elements describe the social and physical distances between the census tracts. If this is a simple binary matrix indicating the adjacency between the units, the function calculates *DM*. If it is a numeric matrix representing the standardised lengths of common boundaries, or the perimeter-to-area ratio, the function returns *DW* or *DS*, respectively. When `nb` is not specified, the spatial component is assumed to be zero, so the output is not adjusted (i.e., equation (1)).

Although the **seg** package does not provide facilities for creating an object for `nb`, there are a number of user-contributed packages that can model spatial relationships between the census tracts. For example, a binary contiguity matrix can be constructed using `poly2nb()` and `nb2mat()`in the **spdep** package. The area of spatial polygons can be obtained by invoking `slot()` if it is stored as class `SpatialPolygons` (or its inherit class). The lengths of shared boundaries are a little more difficult to get, because topological information is not available in most of the common spatial classes in R (Gómez-Rubio & López-Quílez, 2005). One way to work around this problem is to use the **spgrass6** package that provides an interface between R and GRASS (GRASS Development Team, 2012), open-source GIS software. It has a function called `vect2neigh()`, which computes the lengths of common boundaries in an R object through GRASS.

When the argument `adjust` is set to `TRUE`, `dissim()` checks if these functions are available on the user's system and attempts to calculate the spatially-adjusted indices. If the calculation fails, or if this spatial adjustment is disabled (by setting `adjust = FALSE`), `NA` is assigned to the corresponding elements of a list, which will eventually be returned by the function.

The returned list has five elements: `d` for *D*, `dm` for *DM*, `dw` for *DW*, `ds` for *DS*, and `user` for the index adjusted by the optional argument `nb`. Each element ranges between 0 and 1, where a value of zero represents no segregation and a value of one indicates complete segregation. Theoretically, the spatially-adjusted indices are similar to the traditional version when the census tracts with a high proportion of the minority population are clustered together in the study region (i.e., positive spatial autocorrelation). If areas with similar population composition are dispersed, however, output from the equations (2) – (4) should be considerably lower than that from (1), as the additional spatial component becomes large.

It is noteworthy that *D* and its spatial associates consider only two population groups at a time. Considering that many societies are increasingly diverse in terms of race, ethnicity, culture, and religion, this limitation is not desirable. One of the classical, zone-based measures that can work with multiple groups is the index of spatial proximity *SP* (White, 1983), which is defined as:

$$SP = \frac{X \sum_i \sum_j p_{xi} p_{xj} f(d_{ij}) + Y \sum_i \sum_j p_{yi} p_{yj} f(d_{ij})}{(X+Y) \sum_i \sum_j p_i p_j f(d_{ij})} \tag{4}$$

where

$$p_{xi} = \frac{x_i}{X}, \qquad p_{yi} = \frac{y_i}{Y}, \qquad p_i = \frac{x_i + y_i}{X+Y} \tag{5}$$

and $f\,(d_{ij})$ is a function of distance between two census tracts *i* and *j*. In the equations (4) and (5), *X*, *Y*, x_i, and y_i are defined as in (1).

SP evaluates the average distance between all individuals regardless of their population group and then compares that with the sum of the within-group proximities weighted by their respective size. Unlike the index of dissimilarity, this formula can be easily extended to three or more groups by adding them to the numerator and denominator in the same manner as the groups *X* and *Y*.

In the **seg** package, the function to compute this measure of clustering is called `isp()`. As with `dissim()`, the arguments `data` and `nb` specify the population counts and distances between the census tracts, respectively. However, `data` does not have to be a *n*-by-2 matrix; it accepts a numeric matrix with more than two columns, as *SP* can handle multiple population groups. These arguments become optional, if an R object containing points or polygons is provided via the argument `x`, and if it has the population counts as attributes. In this case, the Euclidean distances between the spatial features are estimated using `dist()` when `nb` is not given. The argument `fun` defines $f(d_{ij})$ and controls how the distance affects the social interactions between people. It defaults to a simple negative exponential function (i.e., $e^{(-d_{ij})}$), but it is recommend to try several different distance-decay models and choose the most plausible one.

The function returns a single numeric value indicating the degree of segregation: a value of one means absence of segregation, and values greater than one indicate clustering. If the index value is less than one, it indicates an unusual form of segregation (i.e., people live closer to other population groups).

It is important to note that although *SP* is a useful method for evaluating the degree of residential clustering in the study region, it tends to neglect geographic patterns of small minorities by definition. If one's interest lies in identifying residential clustering of an individual population group, the concentration profile approach proposed by Johnston, Poulsen and Forrest (2002) might be more suitable.

A concentration profile is a cumulative distribution curve, which displays the proportion of the subject group along the *y*-axis and their share in census tracts along the *x*-axis. This is conceptually similar to the Lorenz curve, but in the segregation literature, the Lorenz curve is often constructed by plotting the cumulative proportion of one population group against that of the other group. The concentration profile is different from the Lorenz curve in the sense that it plots the cumulative proportion of the population group against their relative demographic share in geographic units.

Figure 1 is an example of a concentration profile for Pacific peoples in Auckland, the largest city in New Zealand. From this graph, it is apparent that the majority of Pacific peoples reside in the areas where they are relatively overrepresented (i.e., > 30%), and that almost 30% of them live in the areas where they comprise over a half of the local demographic composition. Compared to the single-value indices above, this visual inspection of the data enables a more detailed interpretation of the population distribution.

Concentration profiles can be produced using a function called `conprof()` in the **seg** package. Similar to `dissim()` and `isp()`, the function uses the argument `data` for the population counts, but it does not require any spatial information as this method is non-spatial. Additional arguments `grpID` and `n` should be a single integer value, specifying the population group (i.e., column of the input data matrix) to be examined and the number of *x* values to be used for the construction of the concentration profile. If, for example, `n = 5`, the function takes five equally-spaced values between 0% and 100% as *x* values. For each *x*, it examines how many of the selected group members live in the census tracts where they comprise at least *x* % of the local population.

Unless the argument `graph` is set to `FALSE`, it draws a concentration profile on the current graphic device and returns a numeric value ranging between 0 and 1. The return value is a summary statistic for the concentration profile, *R*, which is derived as described in Hong and Sadahiro (2014). This output can be interpreted in a similar manner to the index of dissimilarity: a small value indicates that the group comprises similar proportions of the local population in all census tracts, and a large value implies a high degree of residential concentration (Hong & Sadahiro, 2014).

Surface-based measures

The segregation measures in the previous section work on the data in which the population counts are agglomerated into arbitrarily defined geographic areas, such as census tracts, electorates, and school zones. Therefore, the resulting degree of segregation depends not only on the actual distribution of the population but also on the choice of spatial units (Openshaw, 1984).

Unlike the zone-based measures outlined above, the surface-based measures do not require the use of aggregate spatial units, so they are theoretically free from this problem. Although, in practice, almost all data available are provided in aggregate form, a plausible population density surface can be obtained using a variety of interpolation techniques by making certain assumptions regarding the distribution of the population. This sort of approach does not necessarily eliminate all possible errors, but previous studies argued that it could reveal important patterns that would not be found using the conventional index of dissimilarity (Kramer, et al., 2010).

Nonetheless, the surface-based measures have not been as widely used in the literature as they might deserve to be. There has been hesitation among scholars to employ these indices, partly because their calculation is complicated and constructing an appropriate interpolation map requires significant computing skills and knowledge in statistics. In order to lower such computational barriers and facilitate the use of these potentially useful methods, we have implemented two sets of surface-based measures in the **seg** package. One is the spatial segregation indices developed by Reardon and O'Sullivan (2004), which consists of the general spatial exposure/isolation index (*P**), the spatial information theory index (*H*), the spatial relative diversity index (*R*), and the spatial dissimilarity index (*D*). The other is the decomposable segregation measure proposed by Sadahiro and Hong (2013).

The core function for the former method is called `spseg()`.The first argument to this function (`x`) should be a spatial object of class `Spatial` or `matrix` of *x* and *y* coordinates, and the second argument (`data`) should be an object of class `matrix` or `data.frame` containing the population data to be analysed. As with `isp()`, the second argument may be omitted if `x` has the data attached to the spatial features. The third argument (`method`) determines which of the four indices should be calculated. By default, it is set to `all`, but a character vector indicating one or more of the following pre-defined strings can be used to specify a subset to calculate: `exposure` for *P**, `information` for *H*, `diversity` for *R*, and `dissimilarity` for *D*.

The rest of the arguments are either optional or have a default value. Experienced R users can control, for example, how the population density surface is estimated and the scale at which segregation is measured through these arguments. Some of the useful arguments include:

`nrow`, `ncol`: Define a regularly-spaced grid over the study region with the specified number of rows and columns; when `smoothing = "kernel"`, the kernel estimate is computed for each node of the grid, and segregation is measured at the same location. In theory, `nrow` and `ncol` should be infinite, because segregation is a continuous phenomenon (i.e., segregation can be measured at any location). In practice, this is not computationally feasible, so a reasonably large number is often used instead.

`power`, `useExp`, `maxdist`: Define a distance decay function. This function is applied to estimate the population composition of the local environment at each point of measurement. When `useExp` is set to `FALSE`, it has the form of $w(d) = e^{(-d \times \alpha)}$, where *d* is the distance between two spatial units, and α is `power`. Otherwise, it is defined as $w(d) = d^{-\alpha}$. If `maxdist` is given, any spatial units that are further than the specified distance will not be considered while evaluating the demographic mix. As will be demonstrated in the next section, the use of this option can help enhance the computation speed, with little or no practical impact on the output.

`smoothing`, `window`, `sigma`: Determine whether or not to interpolate (or simply redistribute) the input spatial data (`x`). If `x` represents exact locations of individuals, the spatial redistribution of the population may not be necessary. When `smoothing = "equal"`, it assumes that the population is uniformly distributed within each census tract. If `smoothing = "kernel"`, spatial interpolation is performed through `kernel2d()` in the **splancs** package, with optional arguments `window` and `sigma` that define the spatial extent and the kernel bandwidth.

Once the function is called with appropriate arguments, it invokes a series of subroutines to accomplish the calculation (Figure 2). Although these subroutines are designed to work in sequence within the main function, they can also be run on their own. This *modularisation* is particularly advantageous when one wants to repeat only part of the calculation procedure. For instance, suppose that the user is interested in how the level of segregation changes with scale. One way to test this is multiple calls to `spseg()` with different scale arguments while holding other arguments constant. This is, however, computationally redundant, because it leads to the construction of the same population density surface each time it runs. It would be more efficient if we execute the subroutines separately, as it allows repeating the necessary components only.

In addition, it offers more flexibility in terms of data preparation. As mentioned above, `spseg()` employs a negative exponential distance decay function to model the effect of the distance on social interactions. While this simple function has been commonly adopted in the literature and is considered appropriate for general use (White, 1983), more realistic representation of neighbourhood may yield a more reliable estimate of segregation. If the user has irregularly-shaped neighbourhood boundaries generated from other R extension packages, or from other software, the local demographic composition could be manually calculated and passed to `spatseg()` directly, instead of going through all the steps in Figure 2.

Regardless of whether the final subroutine `spatseg()` is invoked from the main function `spseg()` or by a direct call, it always returns an object of class `SegSpatial`. It comprises four slots, `p`, `h`, `r`, and `d`, to hold results for *P**, *H*, *R*, and *D*, respectively, along with another four slots, `coords`, `data`, `env`, and `proj4string`, to store information about the data. In order to access and retrieve the values in these slots, the **seg** package provides methods for some standard generic functions, including `show()`, `print()`, and `plot()`. Figure 3 presents a list of the generic functions that have a method for `SegSpatial`.

Another surface-based approach implemented in the **seg** package is the decomposable segregation measure, *S* (Sadahiro & Hong, 2013). One advantage of this method is that it allows decomposing the estimated level of segregation into three independent components, namely, locational segregation, compositional segregation, and qualitative segregation. By evaluating each of these components separately, one can identify whether the observed segregation is mainly due to the demographic structure in the study region, such as the number of ethnic groups and their sizes, or it is caused by geographic clustering/isolation of certain groups.

In the **seg** package, there exists a function called `deseg()` to calculate this decomposable measure of segregation. It works in much the same way as `spseg()`: most of the arguments are identical to those for `spseg()`, except that it does not have the arguments for defining a distance decay function. As with `spseg()`, once the function is initiated, it calls a series of subroutines as illustrated in Figure 4, and each of the subroutines can be invoked separately to avoid unnecessary duplication of processes.

Upon successful run, the function returns an object of class `SegDecomp`, which has four slots, `d`, `coords`, `data`, and `proj4string`. Of these slots, `d` contains a numeric vector of length three, giving the level of locational, compositional, and qualitative segregation, respectively, and the remaining three slots are the same as in `SegSpatial` (i.e., describing the input data). Objects of this class can be manipulated and plotted using the methods implemented in the package (Figure 5).

Results

Zone-based measures

The **seg** package has a sample data set of eight different distributions of the population for demonstration and maintenance purposes. The data set itself is a simple data frame but can be displayed on a 10-by-10 grid, as shown in Figure 6. Example code in the package documentation applies the implemented functions to these patterns, with various combinations of user input, not only to illustrate their use but also to ensure that they produce the *expected* output. Since the same spatial configurations have been used elsewhere, one can easily determine whether the results from `dissim()` are correct or not by comparing them with the relevant figures in the previous studies.

Table 2 presents the output from the `dissim()` function for the idealised patterns above, as well as five additional landscapes portrayed in Figure 7. Although some of the results seem to be slightly different from the measurements of Morrill (1991) (i.e., *D* for the pattern *D* and *DM* for the patterns *C* and *D*), these figures are consistent with the ones in Wong (1993) for the pattern *C*, and also with those obtained from other implementations (e.g., Apparicio, et al., 2013). Considering that the differences between `dissim()` and Morrill (1991) are relatively minor, this might be due to rounding errors.

In comparison, the differences in *DM* for the last two patterns are fairly large, and it is perhaps because `dissim()` uses a different way of counting the neighbouring pairs from Wong (1993). In Wong (1993), the pattern *L* has eight pairs of neighbours: [1-2], [1-3], [1-5], [2-4], [2-5 through the edge A], [2-5 through the edge B], [3-4], and [4-5]. In four of these neighbouring pairs, the spatial units have the same population composition (i.e., $z_i - z_j = 0$). The other four pairs (i.e., [1-5], [2-5 through the edge A], [2-5 through the edge B], and [4-5]), consist of one unit with, say, Asians only, and the other unit with only non-Asians. In this context, the additional spatial component in the equation (2) becomes 4/8 = 0.5, and therefore, *DM* = 1 – 0.5 = 0.5. By contrast, the `dissim()` function does not distinguish the neighbouring pairs by the edge, so there are only seven pairs of neighbours, not eight: [1-2], [1-3], [1-5], [2-4], [2-5], [3-4], and [4-5]. As a result, the spatial component changes to 3/7 = 0.4286, and *DM* = 1 – 0.4286 = 0.5714.

Another notable difference appears in *DS* for the pattern *M*. The cause for this discrepancy is not certain, but one possible explanation is that the standardised shared boundary length, w_{ij}, in the equation (4) was rounded to the first decimal place during the calculation of *DS* in Wong (1993). The R code in Appendix A shows that, in this way, `dissim()` generates the same result as Wong (1993), up to the second decimal place.

Table 3 shows *SP* and *R* for the same data set (i.e., Figure 6 and 7). Unlike `dissim()`, there is no control output (i.e., results from a known set of the data) available for these indices, so the quality of the results cannot be assessed in the same manner as the index of dissimilarity. Nonetheless, the positive correlations between *SP* and *DS* and between *R* and *D* suggest that the functions `isp()` and `conprof()` also produce plausible results.

In terms of the computation speed, all the zone-based tools appear to perform quickly[2]: when applied to a 10-by-10 grid, the `dissim()` function with the `nb` argument completed the calculation in less than 0.03 seconds from 20 iterations, and `conprof()` took only around 0.01 seconds on the average (Figure 8). As the size of the data increased, the amount of time required to obtain results also increases, but not to a large extent; the computation speed does not seem to be too much of an issue here.

In the case of `isp()`, on the other hand, an increase in the number of spatial units tends to slow down the process significantly: it ran in less than 0.03 seconds for a simple 10-by-10 grid, but this figure grew up to about 92.35 seconds for a larger, 100-by-100 grid, as it involves the construction and manipulation of a 10,000-by-10,000 matrix (Figure 8). This is not very slow, but a caution is probably needed when applied to a larger data set, because the current implementation always uses spaces proportional to N^2, where N is the number of spatial units (i.e., n^2).

Surface-based measures

One important consideration that must be taken into account before using `spseg()` or `deseg()` is the choice of arguments, such as the number of measurement points (i.e., `nrow` and `ncol`), the kernel bandwidth (i.e., `sigma`), and distance decay parameters (i.e., `power` and `useExp`). Ideally, the number of measurement points should be as many as possible for an accurate estimation of segregation, because the level of segregation an individual experiences changes continuously over space. The kernel bandwidth should be chosen to ensure that the estimated density surface provides a plausible representation of the actual distribution of the population, and the distance decay parameters should reflect the intensity of social interactions between locations.

In real word applications, however, a large value of `nrow` and `ncol` compared to the spatial resolution of input data often slows down the calculation significantly, while making little difference in the output. Figure 9 shows that as the dimensions of the grid, *n*, increase, the computation time also increases at an exponential rate for the same data set. Nonetheless, the changes in the spatial information theory index, *H*, from the `spseg()` function seem to be negligible: when a 10-by-10 grid (i.e., 100 measurement points in total) was superimposed on the pattern *A*, it took only around 0.02 seconds to complete the task, and *H* was 0.697. This value remained quite similar (i.e., H = 0.685), even when a much larger, 200-by-200 grid was employed, but the computation time increased up to 74.4 seconds on the average.

This result is of course data-dependent, and sometimes a fine grid (i.e., a large number of *n*) is desired to produce more accurate estimates of segregation. In this case, the optional argument `maxdist` can be used to improve the running speed: for example, when `spseg()` was run on the pattern *A* with default values, it spent more than 5 seconds to return *H* = 0.6846. However, when `maxdist` was given, the computation time was reduced by more than one third (i.e., 3.501 seconds), and almost the same figure (i.e., 0.6853) was obtained[3]. In general, the smaller this value, the faster the calculation, but it should be large enough to make sure that *f* (`maxdist`) is practically zero, where *f* (*x*) is the distance decay function, minimising its impact on the output.

In the case of the kernel bandwidth, there is lots of literature on a data-driven choice of this value, but it is often useful to examine several candidates first, as it could shed some light on the scale of segregation (Figure 10). In a similar vein, although a simple negative exponential function with a decay factor of 1 or 2 has been conventionally used as the distance decay parameters since White (1983), the use of varying distance decay rates can help reveal the scale of segregation present in the study region. Reardon and his colleagues (2009), for example, demonstrated how changes in the distance decay parameters affect measured segregation using segregation profiles. This implies that unlike the number of measurement points controlled by `nrow` and `ncol`, the arguments, `sigma`, `power`, and `useExp`, should be chosen more carefully, not on the basis of computational considerations.

Discussion

In this paper, we have described our contribution to the implementation of segregation measures in R. The **seg** package contains various zone-based and surface-based measures of segregation, and among these, the concentration profile approach and the decomposable segregation measure are not available elsewhere. Although there are a few recently developed standalone applications and add-on packages that provide access to *D* and its spatial associates, and the spatial segregation measures, *P**, *H*, *R*, and *D*, the present implementation has a number of merits over the existing tools.

First, since the **seg** package works within the R environment, thousands of standard and modern statistical techniques, as well as facilities for data manipulation and visualisation, can be used to analyse and map the results. This is an important advantage, especially over the standalone applications, because the measurement of segregation is often the beginning of research, not the end. Once the presence of segregation is identified, the next step is to investigate its cause and potential consequences, and a variety of exploratory and confirmatory methods in R can be very useful in this phase. To help the use of other extension packages, the **seg** package provides coercion methods that enable the transformation of the output from `spseg()` or `deseg()` into more general classes, such as `list`, `SpatialPoints`, and `SpatialPolygons`.

Second, the implemented segregation measures are invoked by typing the name of the function, followed by a list of arguments in parentheses. This command line interface probably makes it difficult to use for those who have little experience of R, so we have made the parameter names consistent across the functions and have set default values for most of them. As a result, less skilled users can execute the functions without too much difficulty, while more advanced users can benefit from greater flexibility and control over the calculation procedure through the options.

Third, the **seg** package does not require commercial software to operate, so it is accessible to a wider group of people. R is an open-source software program, and the **seg** package is downloadable from the Comprehensive R Archive Network (CRAN) without charge. Considering the high cost of commercial statistical and GIS software, our implementation in R might be a reasonable alternative for individual researchers and students.

At present, our implementation is limited to place-based segregation measures that assess the demographic diversity of certain geographic areas. These methods are usually applied to residential areas, based on the assumption that where people live determines other aspects of their lives. The recent advances in computing power and the increasing availability of detailed data on daily travel patterns have, however, encouraged the development of various activity space-based segregation measures (Farber, Páez, & Morency, 2012; Wong & Shaw, 2011). In future work, we will try to incorporate some of these indices to the **seg** package.

Note

1. In general, the boundaries of the study region do not account for the calculation of the area-to-perimeter ratio, as no interactions with the outside are presumed.

2. All tests in this section were performed on a computer running Windows 7 and R 3.0.2 with a 3.40 GHz Intel Core i7 processor and 8 GB of RAM.

3. This is because the use of `maxdist` considerably reduces the computational complexity of the implemented algorithm. When `maxdist` is not given, the function considers all points in the data set to estimate the local population composition for each location, so the calculation takes $O(N^2)$ time. By specifying `maxdist`, we can approximate it by evaluating only nearby points, and this can decrease the computational complexity to $O(N)$.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Number 24-02309.

Appendix A

Provided that the GRASS interface is already loaded into R, and that an object `x` is a `SpatialPolygonsDataFrame` object describing the pattern *M*:

```
> writeVECT6(x, vname = "tmp", v.in.ogr_flags = "o")
...
> nl <- vect2neigh(vname = "tmp", units = "me")
...
> a <- unlist(lapply(slot(x, "polygons"), function(z) slot(z, "area")))
> p <- attr(nl, "total") - attr(nl, "external")
> par <- p/a
> mat <- matrix(NA, nrow = length(par), ncol = length(par))
> for (i in 1:length(par)) {
+   for (j in 1:length(par))
+     mat[i,j] <- (par[i] + par[j]) / (max(par) * 4)
+ }
> lm <- listw2mat(sn2listw(nl))
> lm <- (2 * lm) / sum(lm)
> dissim(data = data.frame(x), nb = lm * mat)
[1] 0.6071429
> dissim(data = data.frame(x), nb = round(lm, 1) * mat)
[1] 0.5725
```

The code above shows that if we round w_{ij} to the first decimal place, `dissim()` generates the same result as Wong (1993). Note that the lengthy (and unnecessary) output has been replaced with an ellipsis ("…").

References

Apparicio, P., Martori, J. C., Pearson, A. L., Fournier, É., & Apparicio, D. (2013). An Open-Source Software for Calculating Indices of Urban Residential Segregation. *Social Science Computer Review*.

Duncan, O. D., & Duncan, B. (1955). A methodological analysis of segregation indexes. *American Sociological Review, 20*, 210-217.

Farber, S., Páez, A., & Morency, C. (2012). Activity spaces and the measurement of clustering and exposure: a case study of linguistic groups in Montreal. *Environment and Planning A, 44*, 315-332.

Gómez-Rubio, V., & López-Quílez, A. (2005). RArcInfo: Using GIS data with R. *Computers & Geosciences, 31*, 1000-1006.

GRASS Development Team (2012). Geographic Resources Analysis Support System (GRASS) Software. *Open Source Geospatial Foundation Project*.

Hong, S.-Y., & Sadahiro, Y. (2014). Measuring geographic segregation: a graph-based approach. *Journal of Geographical Systems, 16*, 211-231.

Johnston, R., Poulsen, M., & Forrest, J. (2010). Moving on from indices, refocusing on mix: On measuring and understanding ethnic patterns of residential segregation. *Journal of Ethnic and Migration Studies, 36*, 697-706.

Kramer, M., Cooper, H., Drews-Botsch, C., Waller, L., & Hogue, C. (2010). Do measures matter? Comparing surface-density-derived and census-tract-derived measures of racial residential segregation. *International Journal of Health Geographics, 9*, 29.

Massey, D. S., & Denton, N. A. (1988). The dimensions of residential segregation. *Social Forces, 67*, 281-315.

Morrill, R. L. (1991). On the measure of geographic segregation. *Geography Research Forum, 11*, 25-36.

Openshaw, S. (1984). *The modifiable areal unit problem*. Norwich: Geo.

Peach, C. (2009). Slippery segregation: Discovering or manufacturing ghettos? *Journal of Ethnic and Migration Studies, 35*, 1381-1395.

Peach, C. (2010). ‘Ghetto-Lite’ or Missing the G-Spot? A Reply to Johnston, Poulsen and Forrest. *Journal of Ethnic and Migration Studies, 36*, 1519-1526.

Poulsen, M., Johnston, R., & Forrest, J. (2002). Plural cities and ethnic enclaves: Introducing a measurement procedure for comparative study. *International Journal of Urban and Regional Research, 26*, 229-243.

R Core Team (2014). R: A Language and Environment for Statistical Computing. In R Foundation for Statistical Computing (Ed.). Vienna, Austria.

Reardon, S. F., Farrell, C. R., Matthews, S. A., O'Sullivan, D., Bischoff, K., & Firebaugh, G. (2009). Race and space in the 1990s: Changes in the geographic scale of racial residential segregation, 1990-2000. *Social Science Research, 38,* 55-70.

Reardon, S. F., & O'Sullivan, D. (2004). Measures of spatial segregation. *Sociological Methodology, 34,* 121-162.

Sadahiro, Y., & Hong, S.-Y. (2013). Decomposition approach to the measurement of spatial segregation. *CSIS Discussion Paper*. Center for Spatial Information Science, University of Tokyo.

White, M. J. (1983). The measurement of spatial segregation. *The American Journal of Sociology, 88,* 1008-1018.

Wong, D. W. S. (1993). Spatial indices of segregation. *Urban Studies, 30,* 559-572.

Wong, D. W. S. (2003). Implementing spatial segregation measures in GIS. *Computers, Environment and Urban Systems, 27,* 53-70.

Wong, D. W. S., Reibel, M., & Dawkins, C. (2007). Introduction—segregation and neighborhood change: where are we after more than a half-century of formal analysis. *Urban Geography, 28,* 305-311.

Wong, D. W. S., & Shaw, S.-L. (2011). Measuring segregation: an activity space approach. *Journal of Geographical Systems, 13,* 127-145.

Figures and Tables

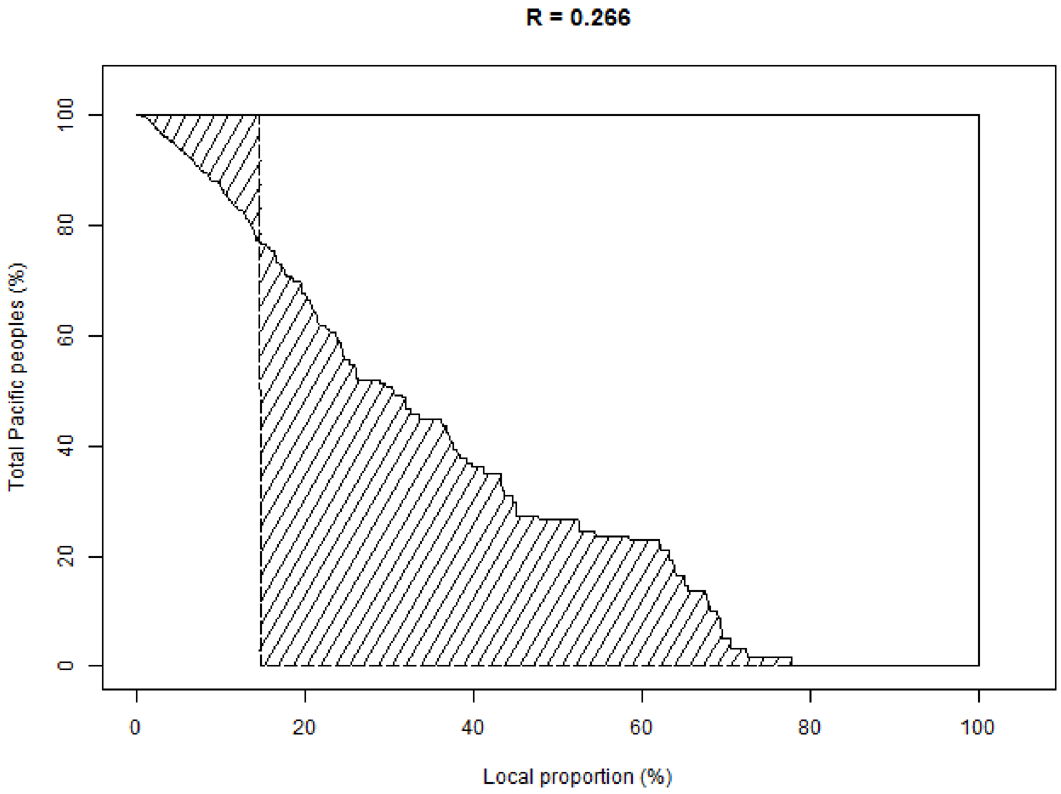

Figure 1 Concentration profile for Pacific peoples in Auckland, New Zealand

Source: New Zealand census of population and dwellings, 2006

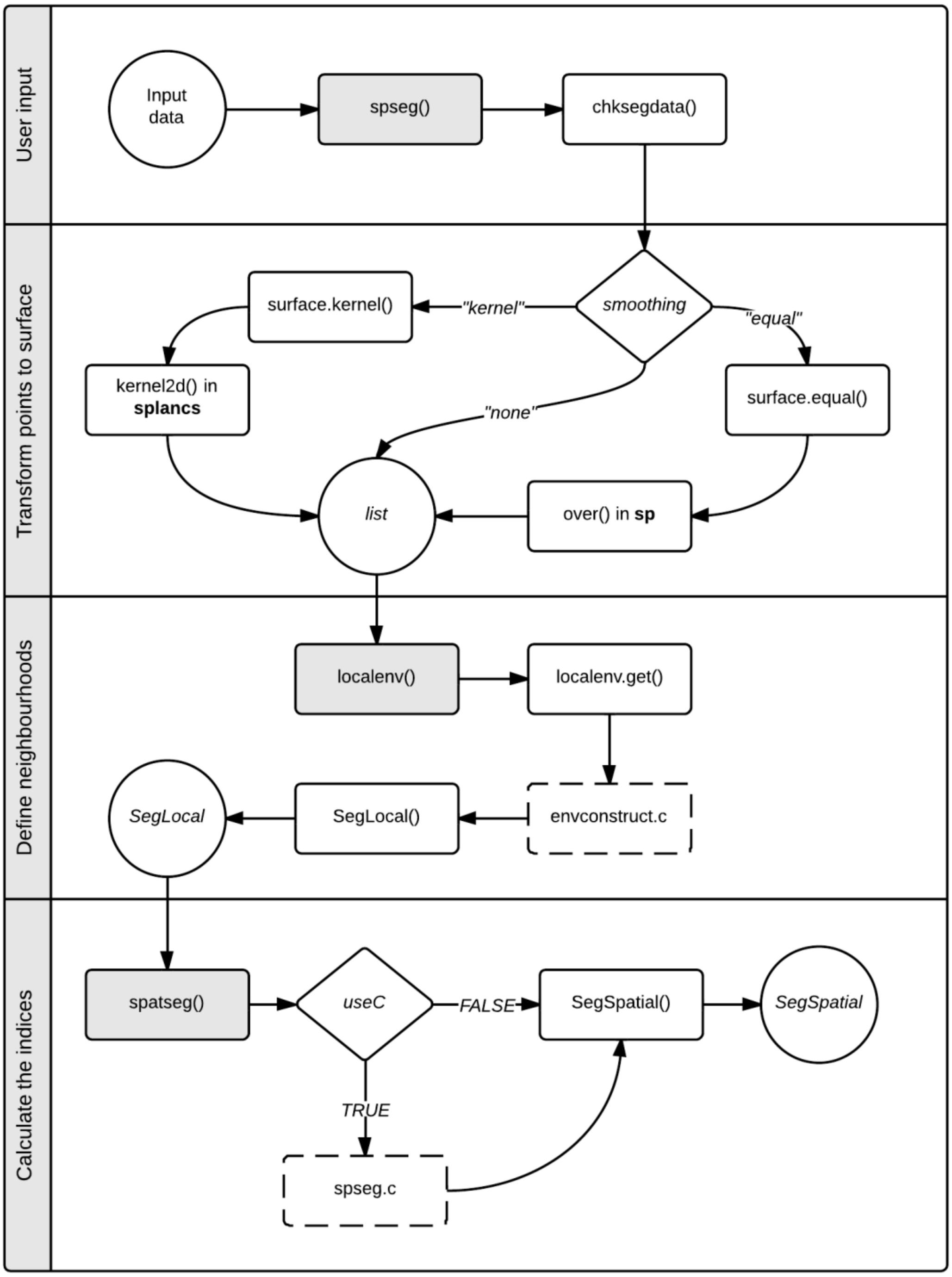

Figure 2 Computational flow of the function `spseg()`. In the diagram, circles represent an R object, rhombuses represent user arguments, and curved-rectangles refer to R functions. Among the rectangles, only the shaded ones are user-level functions.

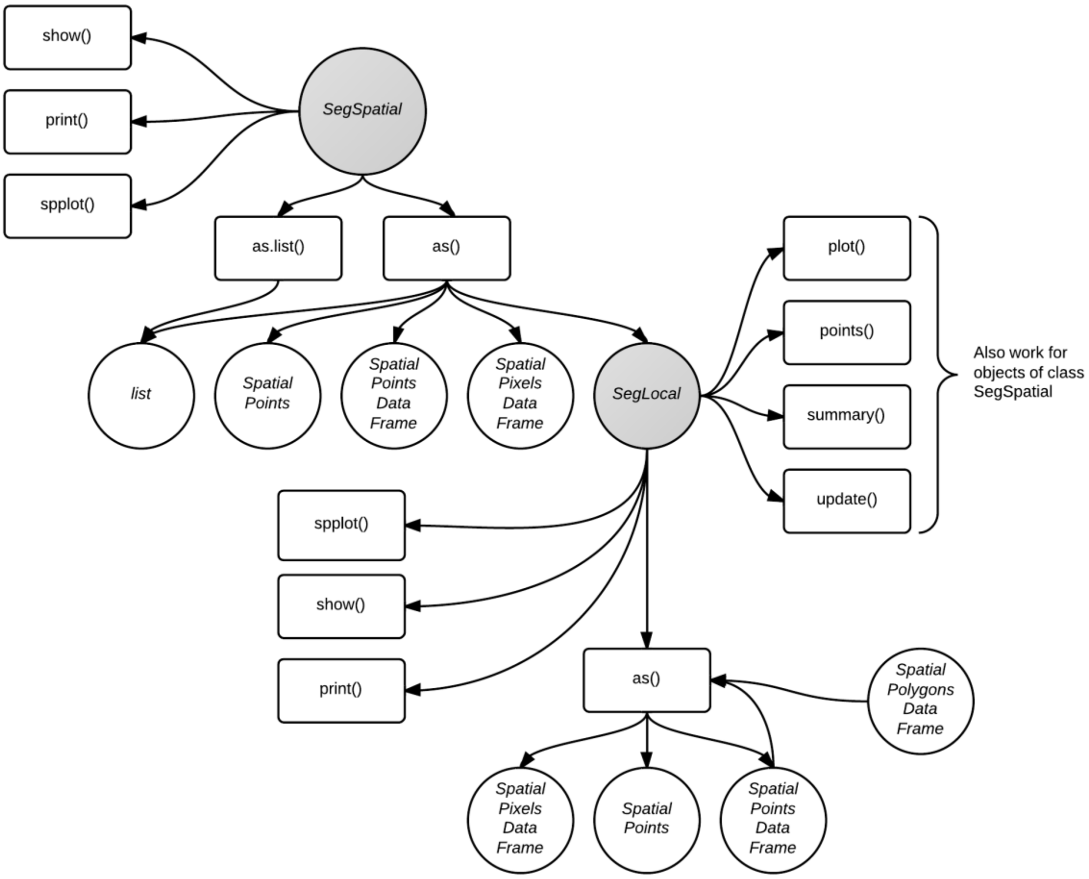
show()
print()
spplot()
SegSpatial
as.list()
as()
list
Spatial Points
Spatial Points Data Frame
Spatial Pixels Data Frame
SegLocal
plot()
points()
summary()
update()
Also work for objects of class SegSpatial
spplot()
show()
print()
as()
Spatial Polygons Data Frame
Spatial Pixels Data Frame
Spatial Points
Spatial Points Data Frame

Figure 3 Class `SegSpatial` and the associated methods

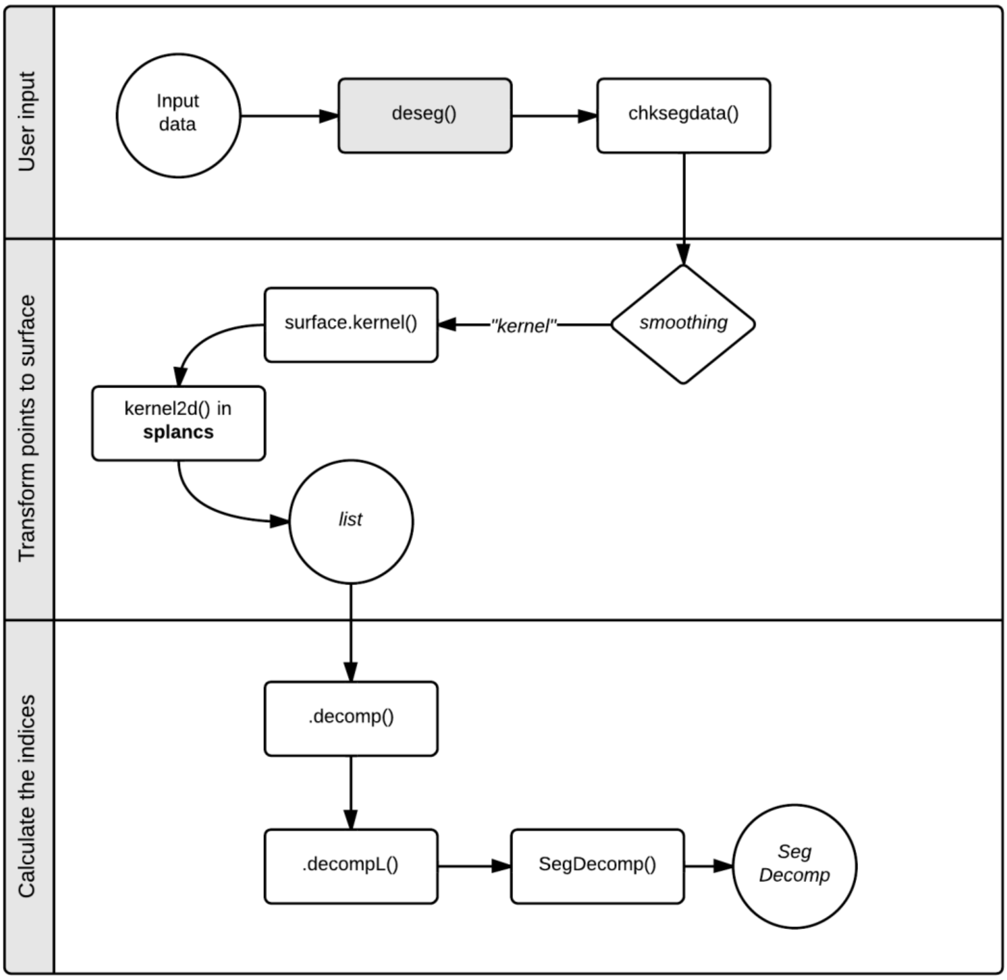

Figure 4 Computational flow of the function `deseg()`

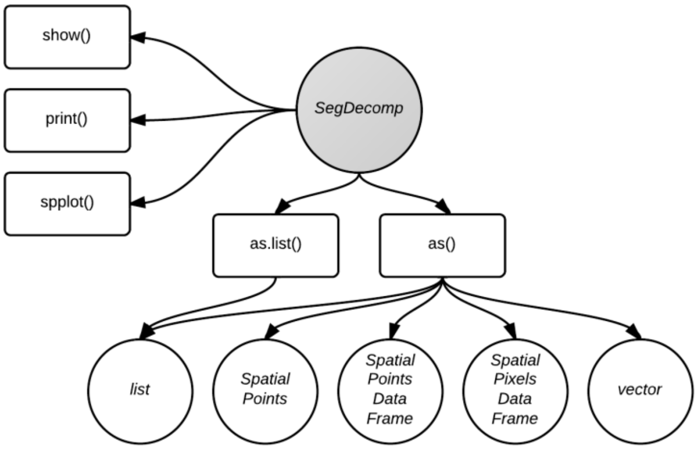

Figure 5 Class `SegDecomp` and the associated methods

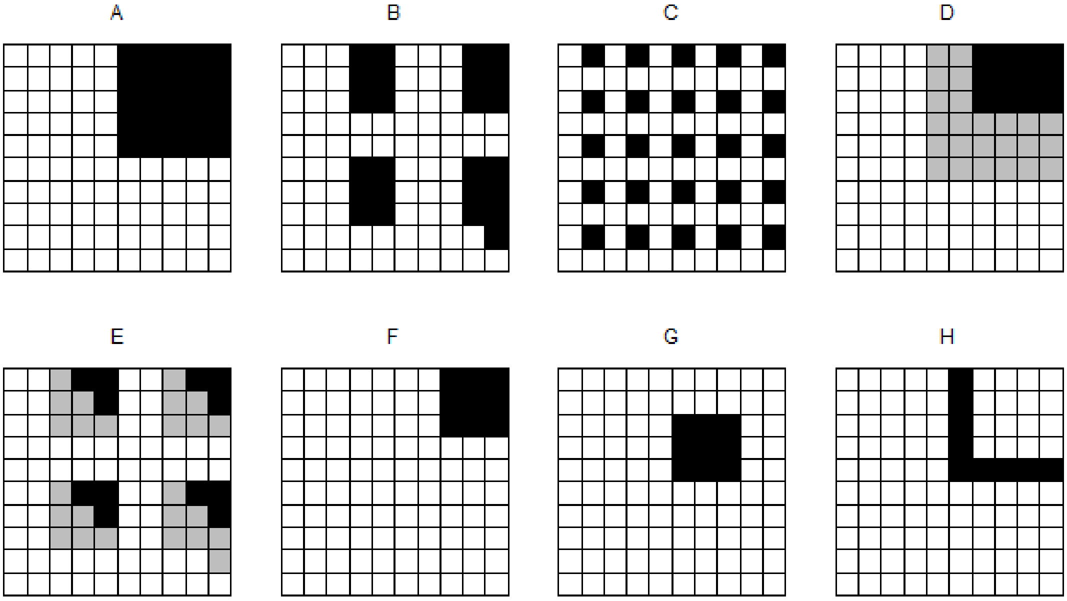

Figure 6 Spatial patterns of segregation on a 10-by-10 grid. The black cells are where the minority population comprises 100% of the local population, the grey cells 50%, and the white cells 0%.

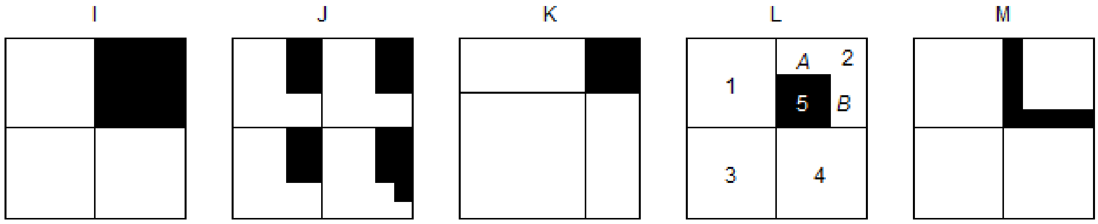

Figure 7 Same spatial patterns of segregation as the patterns *A*, *B*, *F*, *G*, and *H*, except that the cells with the same colour have been aggregated. The numbers inside of the cells in the pattern *L* indicate the cell ID, and the letters denote the edges.

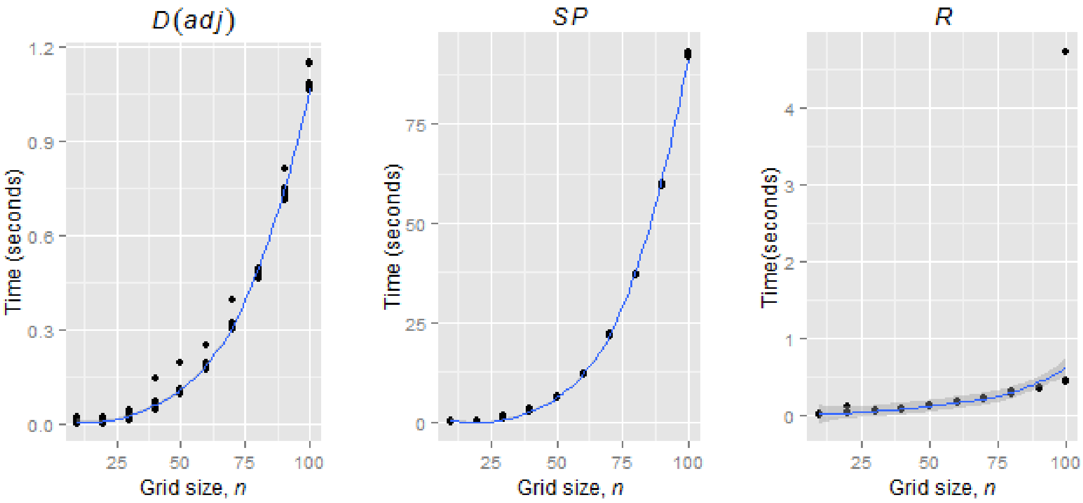

Figure 8 Computation time (in seconds) of $D(adj)$, SP, and R on a n-by-n grid. For each n, the calculations were repeated 20 times. The blue lines are locally weighted scatter-plot smoothing (LOESS) curves.

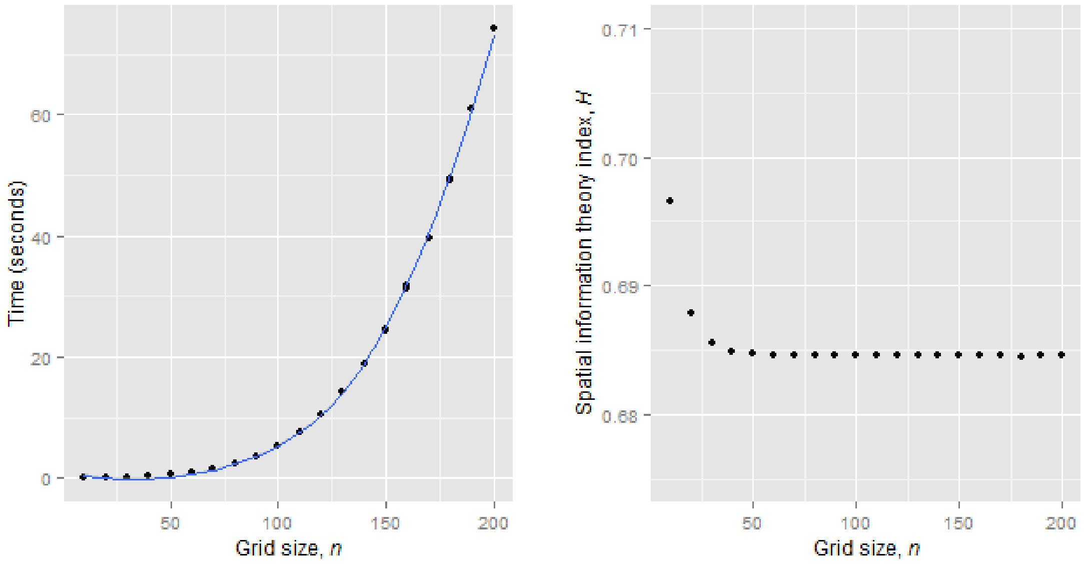

Figure 9 Computation time (in seconds) of the spatial information theory index, *H*, on a *n*-by-*n* grid (left) and changes in the output (right). For each *n*, the calculations were repeated 10 times. The blue line on the left represents a LOESS curve.

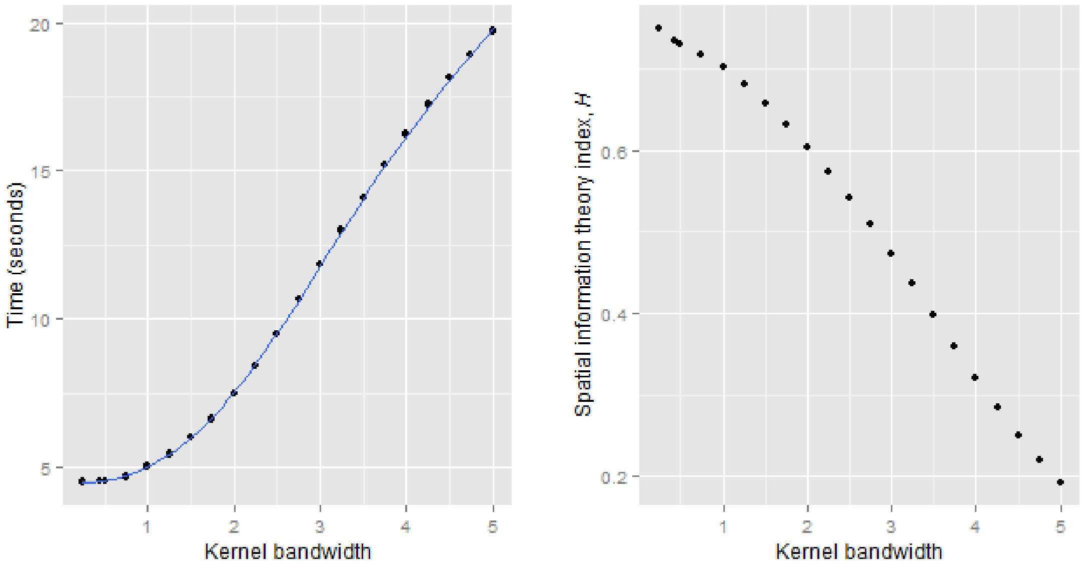

Figure 10 Computation time (in seconds) of the spatial information theory index, *H*, for the pattern *A*, with different kernel bandwidth values (left) and changes in the output (right). For each *n*, the calculations were repeated 10 times. The blue line on the left represents a LOESS curve.

Table 1 Zone- and surface-based segregation measures implemented in the **seg** package

	Measure	Original paper	Function
Zone-based	Index of dissimilarity, *D*	Duncan & Duncan (1955)	`dissim()`
	Spatially-adjusted *D* (contiguity), *DM*	Morrill (1991)	`dissim()`
	Spatially-adjusted *D* (boundary length), *DW*	Wong (1993)	`dissim()`
	Spatially-adjusted *D* (perimeter/area ratio), *DS*	Wong (1993)	`dissim()`
	Index of spatial proximity, *SP*	White (1983)	`isp()`
	Concentration profile	Poulsen, Johnston & Forrest (2002)	`conprof()`
Surface-based	Spatial exposure/isolation index, $\tilde{P}*$	Reardon & O'Sullivan (2004)	`spseg()`
	Spatial information theory index, $\tilde{H}$	Reardon & O'Sullivan (2004)	`spseg()`
	Spatial relative diversity index, $\tilde{R}$	Reardon & O'Sullivan (2004)	`spseg()`
	Spatial dissimilarity index, $\tilde{D}$	Reardon & O'Sullivan (2004)	`spseg()`
	Decomposable measure of segregation	Sadahiro & Hong (2013)	`deseg()`

Table 2 *D*, *DM*, *DW* and *DS* from the function `dissim()`

	D	*DM*	*DW*	*DS*	*D*	*DM*	*D*	*DM*	*DW*	*DS*
A	1.00	0.94	0.94	0.95	1.00	0.94	1.00	0.94	0.94	0.95
B	1.00	0.83	0.83	0.84	1.00	0.83	1.00	0.83	0.83	0.84
C	1.00	0.50	0.50	0.54	1.00	0.48	1.00	0.50	0.50	0.54
D	0.84	0.79	0.79	0.79	0.83	0.76	-	-	-	-
E	0.83	0.66	0.66	0.68	0.83	0.66	-	-	-	-
F	1.00	0.97	0.97	0.97	-	-	1.00	0.97	0.97	0.97
G	1.00	0.93	0.93	0.93	-	-	1.00	0.93	0.93	0.93
H	1.00	0.90	0.90	0.91	-	-	1.00	0.90	0.90	0.91
I	1.00	0.50	0.50	0.50	-	-	1.00	0.50	0.50	0.50
J	1.00	0.33	0.24	0.54	-	-	1.00	0.33	0.24	0.54
K	1.00	0.50	0.70	0.74	-	-	1.00	0.50	0.70	0.74
L	1.00	0.57	0.54	0.68			1.00	0.50	0.54	0.68
M	1.00	0.40	0.36	0.61	-	-	1.00	0.33	0.36	0.57

Table 3 *SP* and *R* from the functions `isp()` and `conprof()`

	A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*	*L*	*M*
SP	1.62	1.16	0.89	1.44	1.12	1.42	1.34	1.16	0.67	0.26	0.91	0.37	0.34
R	1.00	1.00	1.00	0.67	0.66	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00