On the Computation of Large Spatial Datasets With M

Eric Marcon

Florence Puech

October 1, 2024

Abstract

Increasing access to large individual and spatial datasets, coupled with the development of computing power, has encouraged the search for suitable statistical tools to best analyse such data. In a recent article published in this journal, Tidu et al. (2024) highlight the qualities of the *M* function (Marcon and Puech, 2010), a measure of spatial concentration in continuous space. They also express reservations about the computation times required. Our methodological work seeks to specify the processing of large spatialized data sets with *M* using R software. Two avenues are being explored to determine the computational performance of *M*. Firstly, a precise evaluation of the computational time and memory requirements for geo-localised data is carried out using the dbmss package in R by means of performance tests. Then, as suggested by Tidu et al. (2024), we also consider the possibility of approximating the geographical positions of the entities analysed. The extent of the deterioration in the estimate of *M* that this approach creates can thus be estimated, as can the gains in computation time made possible by the spatial approximation of locations. The complete R code is given for the reproducibility of the results.

1 Introduction

Increasing access to large individual and spatial datasets and greater computing power have encouraged the development of statistical analysis tools for processing such data in the best possible way (Baddeley et al., 2016). Empirical studies at very fine geographical levels have thus been proposed in recent years for large datasets. Particular attention has been paid to detecting the spatial structures (attraction, repulsion, independence) of individual spatialised data using analyses that are no longer based on zoned data but on geolocalised data. This type of approach has the advantage of preserving the exact positions of the entities analysed and therefore does not erase individual specificities. Various studies have shown how important it is to use this type of methodology in a wide variety of fields, including geography (Sweeney and Feser, 1998; Deurloo and De Vos, 2008), economics (Arbia, 1989; Marcon and Puech, 2003), ecology (Cressie, 1993; Lentz et al., 2011), biology (Dray et al., 2021), and so on. In a recent article, Tidu et al. (2024) highlights the interest of a particular statistical measure, the M function proposed by Marcon and Puech (2010). This measure, which we will refer to as M in the remainder of the article, makes it possible to highlight spatial structures within a spatialised distribution (attraction, repulsion, independence) from a study based on the distances separating the entities analysed. However, while this measure preserves all the richness of individual geolocated data, it requires a longer calculation time than other distance-based measures, since it is both a cumulative and relative measure (see Marcon and Puech (2017) for a literature review on the advantages and limitations of a dozen existing distance-based measures). Tidu et al. (2024) propose to limit M calculation times by introducing a voluntary positioning error for the entities analysed. For example, in their study, industrial establishments in Sardinia (Italy) are no longer located at their exact postal address but at the centroid of their municipality. This repositioning reduces calculation times, as the number of possible distances between establishments is in fact limited to the distances separating the centroids of the municipalities. This approach is similar to that of Scholl and Brenner (2015) who proposed, for the K_d function (Duranton and Overman, 2005) which characterises spatial structures using another method, to approximate the distances between pairs of entities by grouping them into classes. The method of Scholl and Brenner (2015), implemented in the *dbmss* package (Marcon et al., 2015) for R (R Core Team, 2024) provides a considerable gain in computational performance with little loss of accuracy.

In our paper, we propose to test the effectiveness of Tidu et al. (2024)'s method. First, we show the advantages of using the *dbmss* package to estimate the M function on datasets with an order of magnitude of 100,000 points or less, and we show that the computation times become excessive beyond that, on a personal computer. We then study the effect of the geographical approximation of the locations of the entities analysed. This methodological work, based on a deliberately limited number of entity locations, enables us to quantify the extent of the deterioration in information that this approach creates. These performance tests provide a precise answer to the computational advantages and limitations of the M function as a function of the size of the datasets.

The plan of the article is as follows. The first section generates the necessary data. Large point sets (of the order of several tens of thousands of points) that are either completely random or (geographically) concentrated are drawn. The second section details the use of the *dbmss* package to calculate the M function and its confidence interval from a table giving the position and characteristics of the points or a matrix of distances between them. The third section measures the performance of *dbmss* as a function of the size of the set of points, in terms of computing time and memory requirements. Finally, the last section tests the approximation which consists of grouping them together at the centre of the cells of a grid, following the approach of Tidu et al. (2024) which positions them at the centre of the administrative units in which they are located.

2 Data simulation

The datasets we will consider in this article are obtained by simulation. The R code is given in the appendix, which allows perfect reproducibility of the examples treated or the development of others.

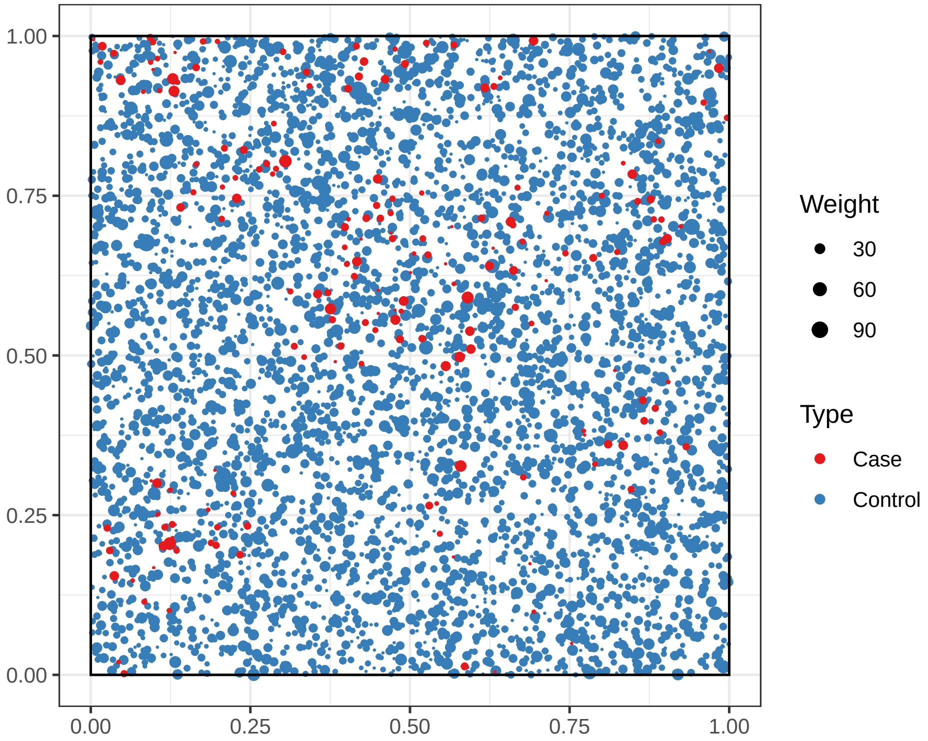

Figure 1: Drawing of a set of points where the Cases (in red) are aggregated and the Controls (in blue) are distributed completely randomly. The size of the points is proportional to their weight.

2.1 Drawing the points

A set of points is drawn by a Poisson process (whose expectation of the number of points is 5000) in a square window of side 1. Each point is assigned a qualitative mark: ‘Case’ or ‘Control’. The majority of points are ‘Controls’ and a proportion (5%) are ‘Cases’, whose spatial structure is studied. The weight of the points is drawn from a gamma distribution with free shape and scale parameters.

In this example, the drawing of points is completely random (*complete spatial randomness*: CSR), i.e. there is no simulation of the attraction or dispersion of points which could generate spatial concentrations of points (aggregates) or, on the contrary, spatial regularities (dispersions).

Sets of aggregated points can be drawn in a Matérn (1960) process.

The Cases are shown in figure 1: the aggregates are clearly visible. The controls are distributed

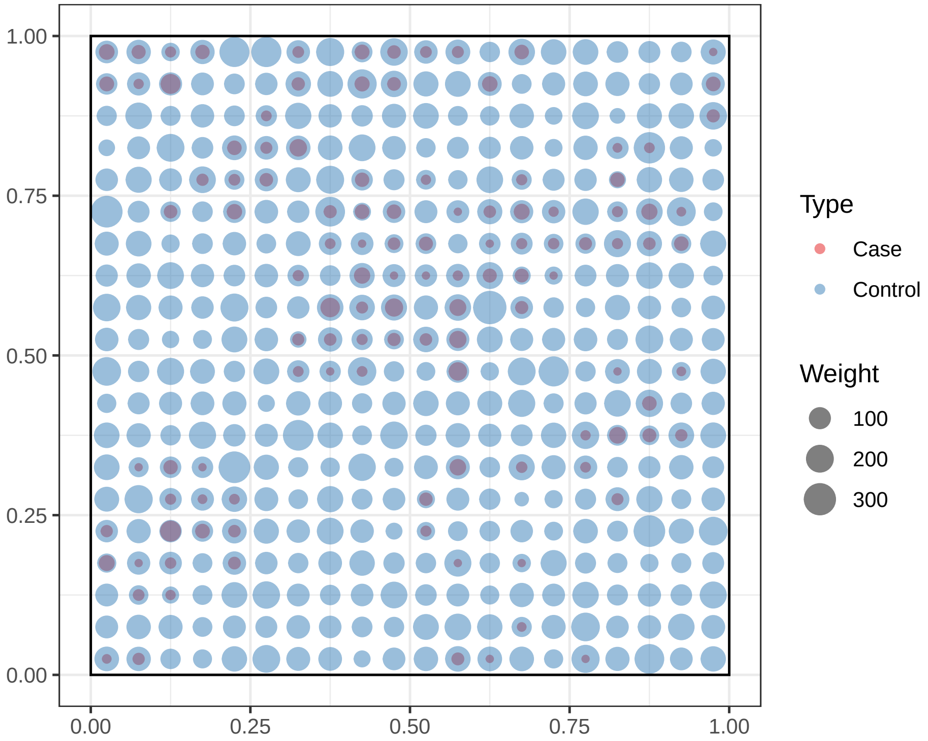

Figure 2: Repositioning of points in an arbitrary grid. The absence of Cas in a cell is easily detected (single-colour blue dot), as is the strong presence of Cas in a cell (two-colour dot, but predominantly red).

completely randomly.

2.2 Gridding the space

Let's consider the simulation of the Cases obtained by the Matérn process and cut the window into a grid. An east mesh of the space previously considered is obtained, which simulates the usual approximation of the position of the points of an administrative unit to the position of its centre.

The refocused position is shown on the map in figure 2. Each cell now contains only one point of each type, whose weight is the sum of the weights of the individual points.

The values of the M function can now be calculated from the original point set or its approximation after recentring.

3 Computing M with the *dbmss* package

3.1 Necessary data

In the *dbmss* package, the function is applied to a set of points or a distance matrix. The set of points in figure 1 is used. The distance matrix between all the pairs of its points is calculated to form the data on which the performance tests will be carried out.

3.2 Point pattern

The `Mhat()` function in the *dbmss* package is used to estimate the M function. The theoretical reference value for M is 1, as this function relates the proportion of Cases up to a distance r to that observed over the entire window. The aggregation of Cases will be highlighted by values of M greater than 1 (the relative presence of Cases is greater locally than over the whole window) and the dispersion of Cases by values less than 1. We observe (figure 3) that M detects an agglomeration of Cases, which is in agreement with the simulation of this type of point (the controls having a completely random location on the window). The advantage of a function based on distances is clearly visible: it allows us to detect exactly at which distance(s) the attraction phenomena occur and are the most important (for functions whose values can be compared at different radii, such as M).

In addition to estimating the M function, the `Menvelope()` function can be used to calculate its global confidence interval (Duranton and Overman, 2005) under the null hypothesis of random point location. The result is shown in figure 3.

3.3 Distance matrix

Matrices can be used to process non-Euclidean distances (transport time, road distance, etc.) which cannot be represented by a set of points.

The `Mhat()` and `MEnvelope()` functions are the same, and provide the same results whatever the form of the data used here (point set or distance matrix).

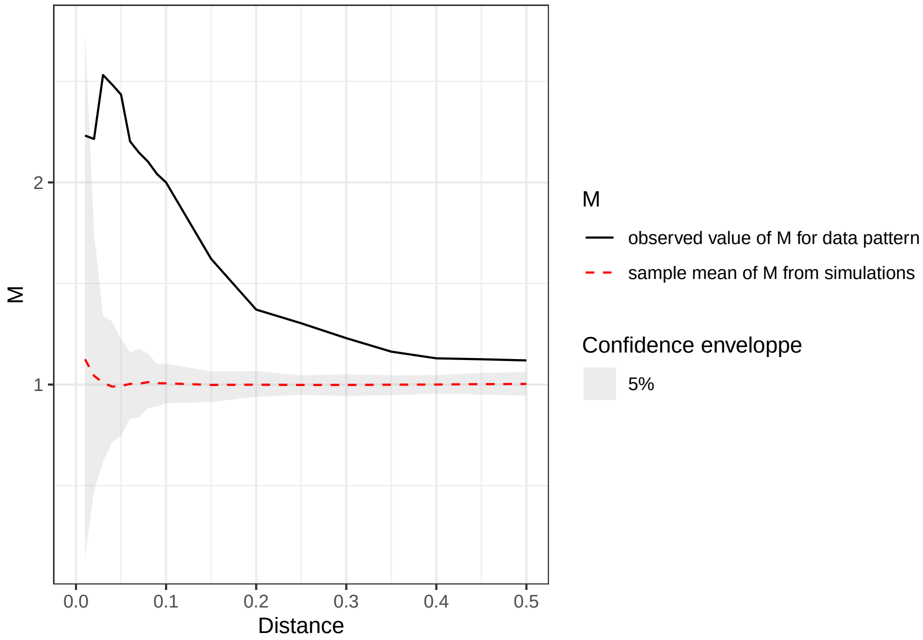

Figure 3: Value of M as a function of distance from the reference point. The confidence interval, simulated at 95%, appears in grey and is centred on the value 1.

4 Computational performance

The use of the M function to characterise the spatial structure of large sets of points may be limited by the computing time or memory required.

4.1 Computing time

Calculating the distances between all pairs of points is necessary to estimate M. The calculation time is therefore expected to increase as the square of the number of points.

The calculation time required for the exact calculation is evaluated for a range of numbers of points (figure 4).

The calculation time is related to the size of the set of points by a power law.

It increases less quickly than the square of the number of points. It can be estimated very precisely ($R^2 = 0.97$) by the relation $t = t_0(n/n_o)^p$ where t is the estimated time for n points knowing the time t_0 (ex. r format(as.numeric(M_time[5, 2]), digits=3)‘ seconds) for n_0 points

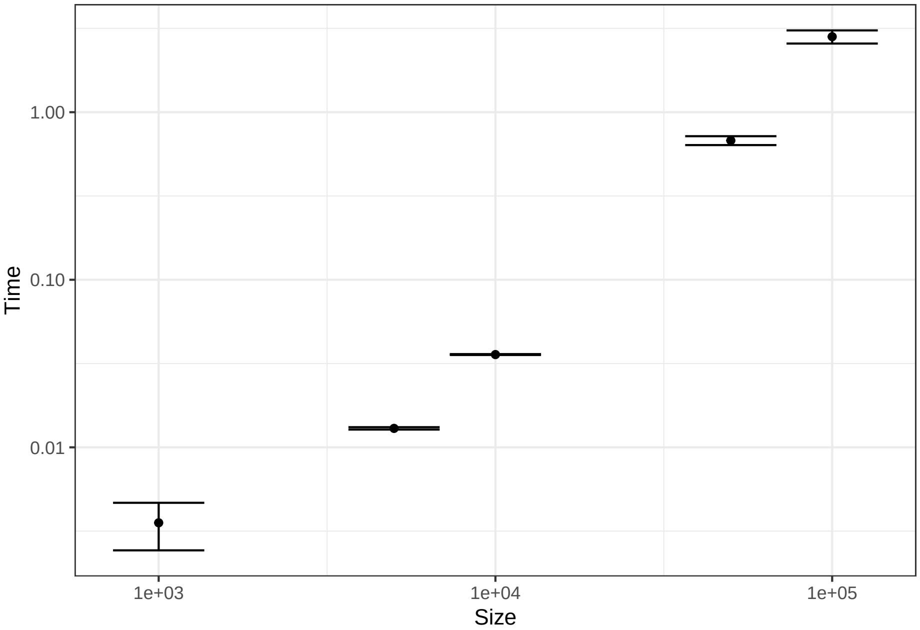

Figure 4: Calculation time (seconds) of the estimate of the M function as a function of the size of the set of points. The bars represent the ±1 standard deviation interval.

(e.g.: 100000) and p the power relation (1.49).

Using a distance matrix may seem an efficient way of saving computation time, but in reality calculating distances is extremely fast and the whole process from a matrix is ultimately more time-consuming (array **??**).

4.2 Memory

The memory used is evaluated for the same data sizes (figure 5).

The memory required increases linearly with the number of points and is never critical for point set sizes that can be processed in reasonable times. This puts Tidu et al. (2024)'s conclusion about the power and computation time required when using M on large datasets into perspective.

The memory used by `Dtable` objects to calculate M from a distance matrix is much greater: it is that of a numerical matrix, of the order of 8 bytes times the number of points squared, i.e. 800 MB for 10000 points alone. As the calculation time is not reduced by this approach, its use should

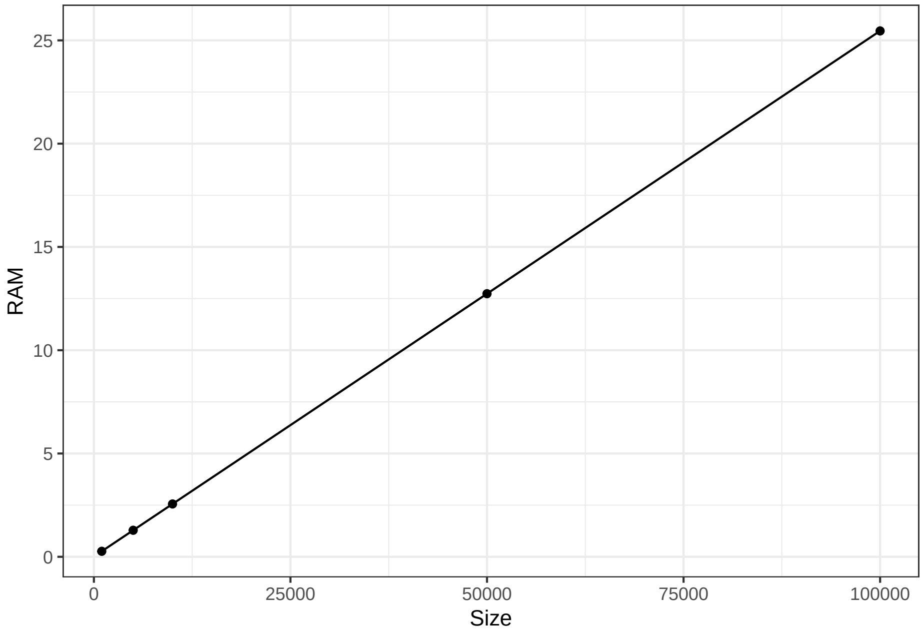

Figure 5: Memory required (MB) for estimating the M function as a function of the size of the set of points.

be reserved for non-Euclidean distances.

5 Effects of approximating the position of points

Clearly, approximating the position of the points results in a loss of information: in each grid cell, the distance between all the points is set to 0, and the distance between two points in different cells is approximated by the distance between the centroids of the two cells. We therefore expect a severe error in the estimation of M on a small scale (of the order of magnitude of the size of the cells) and an error that decreases with distance, when the relative size of the cells decreases.

The effect of the location approximation is first tested on a set of aggregated points, similar to the real Tidu et al. (2024) data. Secondly, the case of an unstructured set of points is considered.

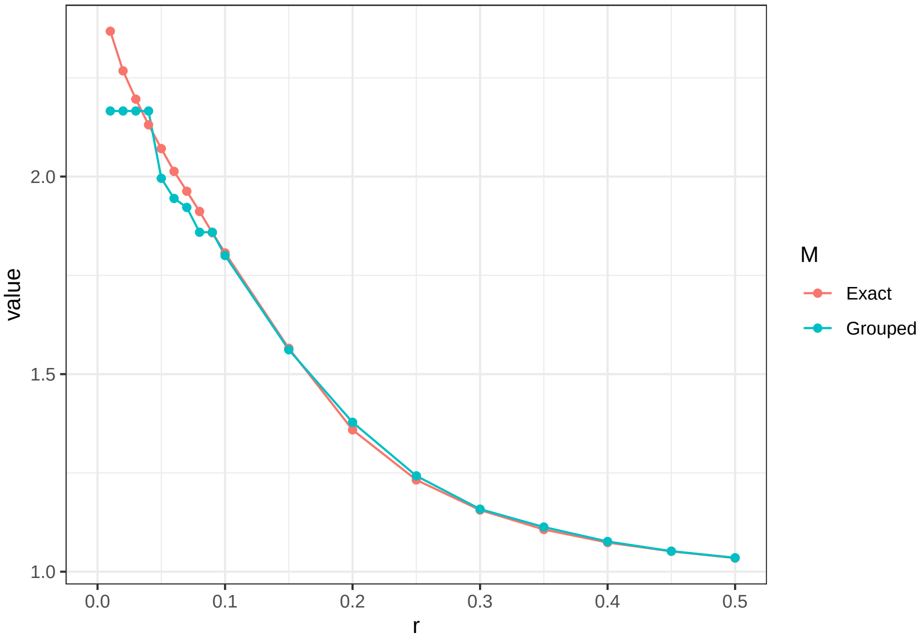

Figure 6: Average estimate of the M function from the exact position of the points compared with the values obtained by grouping the points. Cases form aggregates of radius 0.1.

5.1 Case of an aggregated distribution (Matérn)

100 sets of aggregated points (5000 points with 5% of Cases) are simulated. To evaluate the effect of the position approximation, the exact calculation and the calculation on the grid points are performed on each set of points.

The mean values of the estimates of M are presented in figure 6.

The size of the grid cells is equal to 0.05. All neighbours at distances less than this threshold are placed at zero distance: the estimate of the function is constant up to this threshold.

The correlation between the M values estimated by each method is calculated at each distance (figure 7).

The correlation is very close to 1, and the estimated values very similar, as soon as the distance taken into account exceeds the grid cell: the approximation is not a problem if the interactions between the points are studied beyond this distance. The information on interactions at short distances, i.e. within each grid cell, is lost, or, more precisely, approximated by its value at the grid

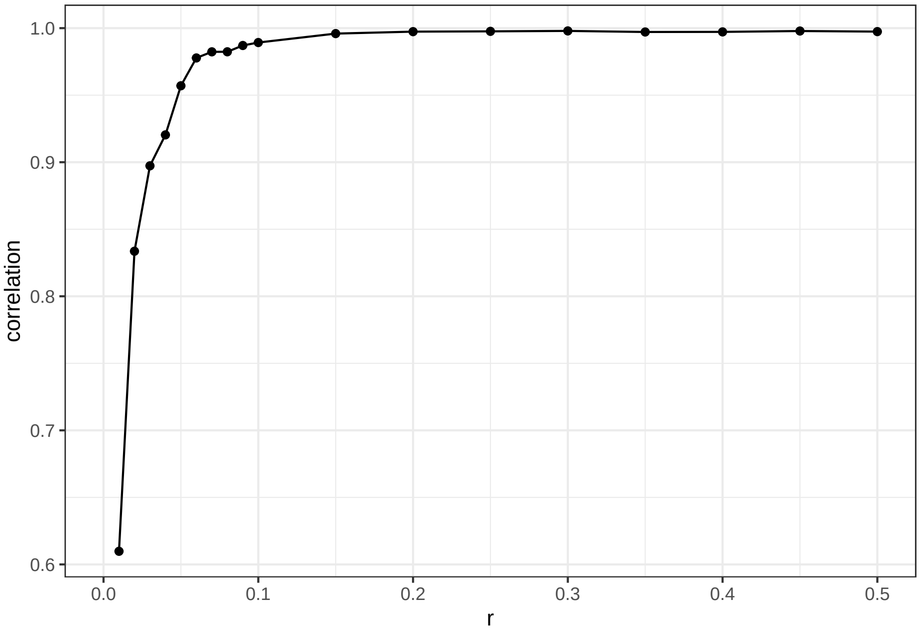

Figure 7: Correlation between M values estimated from the exact position of the points and by grouping the points. Cases form aggregates of radius 0.1.

scale. grid scale.

5.2 Case of a completely random distribution (CSR)

The same simulations are run with a completely random set of points. The exact calculation and the calculation on the grid points are carried out on each set of points.

The average values are shown in figure 8.

The mean value of M is equal to 1 at all distances by construction: Cases and Controls are distributed completely randomly. The approximations are relatively small in value (a few percent) but as the real value of M varies little around 1, the correlations are much weaker (figure **?**(fig:CorCSRFig)) in the absence of spatial structure.

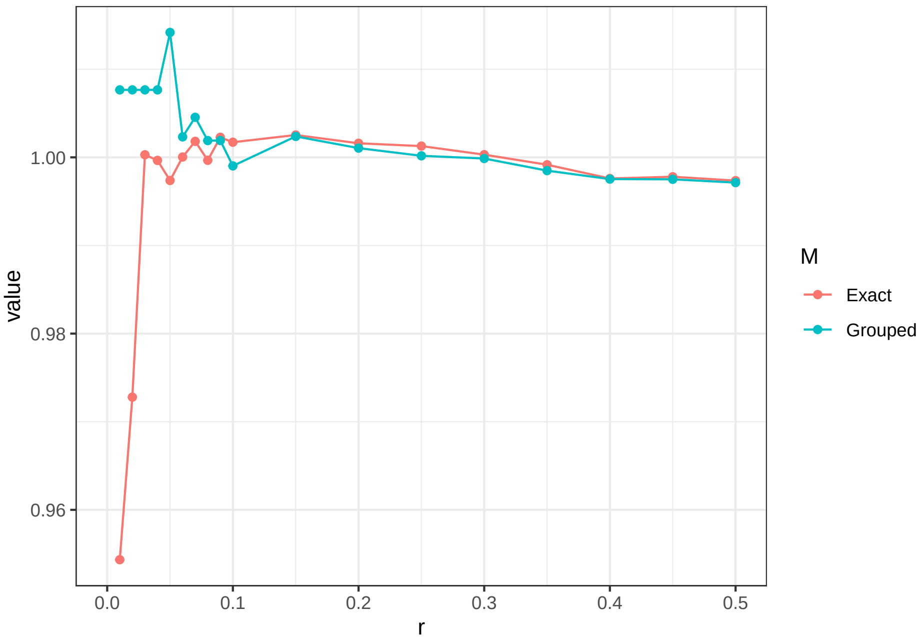

Figure 8: Average estimate of the M function from the exact position of the points compared with the values obtained by grouping the points. Both Cases and Controls are drawn in a Poisson process.

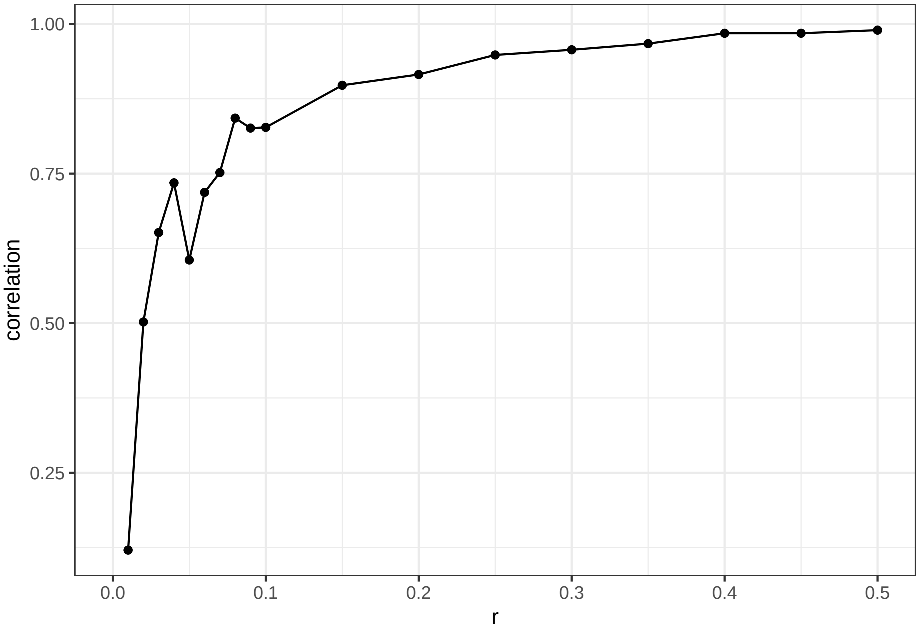

Figure 9: Correlation between M values estimated from the exact position of the points and by grouping the points. Both Cases and Controls are drawn in a Poisson process.

6 Conclusion

To summarise, and based on the cases proposed in this article, it seems that approximation on location can be considered to save computation time, given the strong correlation observed between the values of M on exact and approximated data, but keeping a very fine mesh. Our result is therefore in line with Tidu et al. (2024)'s article, which mentions strong correlations on Italian company location data. Since the spatial structure of their data is probably an intermediate case between the two cases dealt with in our article (aggregated and random theoretical distributions), the results provided by our two contributions are complementary. On the other hand, if the aim of the study is to look at spatial structures at very small distances, then approximating geographical positions is not desirable because it is for these distances that the discrepancies between the M results are greatest.

The calculation time for M is around 6 seconds for a set of 100,000 points on a laptop (Intel i7-1360P 2.20 GHz processor), and requires 25 MB of RAM. Calculating a confidence interval from

1,000 simulations therefore takes less than two hours.

For a set of five million points, the expected calculation time is $6 times 50^{1.8} = $6860 seconds, nearly two hours. 1000 simulations would then take around three months. The calculation of distances is parallelized: a calculation server would drastically increase performance, but at the cost of a complexity of implementation that limits its use.

If we limit ourselves to the computing power of a personal computer, exact calculation is fully justified for data of the order of $10^5 points: a few hours are enough to calculate confidence intervals.

Beyond that, approximating the location reduces the size of the set of points to the number of locations selected. The price to pay is the absence of information at the scale of elementary geographical units (the grid cells in this case). Depending on the issues addressed, this limitation may or may not be acceptable: the overall description of the spatial structure is not significantly degraded, but the study of externalities, which is particularly interesting at short distances, is very limited.

References

Arbia, G. (1989). *Spatial Data Configuration in Statistical Analysis of Regional Economic and Related Problems.* Dordrecht: Kluwer.

Baddeley, A., E. Rubak, and R. Turner (2016). *Spatial Point Patterns: Methodology and Applications with R.* Chapman & Hall/CRC Interdisciplinary Statistics Series. Boca Raton London New York: CRC Press.

Cressie, N. A. (1993). *Statistics for Spatial Data.* New York: John Wiley & Sons.

Deurloo, M. C. and S. De Vos (2008). Measuring segregation at the micro level: An application of the M measure to multi-ethnic residential neighbourhoods in Amsterdam. *Tijdschrift voor economische en sociale geografie 99*(3), 329–347.

Dray, N., L. Mancini, U. Binshtok, F. Cheysson, W. Supatto, P. Mahou, S. Bedu, S. Ortica,

E. Than-Trong, M. Krecsmarik, S. Herbert, J.-B. Masson, J.-Y. Tinevez, G. Lang, E. Beaurepaire, D. Sprinzak, and L. Bally-Cuif (2021). Dynamic spatiotemporal coordination of neural stem cell fate decisions occurs through local feedback in the adult vertebrate brain. *Cell Stem Cell 28*(8), 1457–1472.e12.

Duranton, G. and H. G. Overman (2005). Testing for localisation using micro-geographic data. *Review of Economic Studies 72*(4), 1077–1106.

Lentz, J. A., J. K. Blackburn, and A. J. Curtis (2011). Evaluating Patterns of a White-Band Disease (WBD) Outbreak in Acropora palmata Using Spatial Analysis: A Comparison of Transect and Colony Clustering. *PLoS ONE 6*(7), e21830.

Marcon, E. and F. Puech (2003). Evaluating the geographic concentration of industries using distance-based methods. *Journal of Economic Geography 3*(4), 409–428.

Marcon, E. and F. Puech (2010). Measures of the geographic concentration of industries: Improving distance-based methods. *Journal of Economic Geography 10*(5), 745–762.

Marcon, E. and F. Puech (2017). A typology of distance-based measures of spatial concentration. *Regional Science and Urban Economics 62*, 56–67.

Marcon, E., S. Traissac, F. Puech, and G. Lang (2015). Tools to characterize point patterns: dbmss for R. *Journal of Statistical Software 67*(3), 1–15.

Matérn, B. (1960). Spatial variation. *Meddelanden från Statens Skogsforskningsinstitut 49*(5), 1–144.

R Core Team (2024). *R: A Language and Environment for Statistical Computing.* Vienna, Austria: R Foundation for Statistical Computing.

Scholl, T. and T. Brenner (2015). Optimizing distance-based methods for large data sets. *Journal of Geographical Systems 17*(4), 333–351.

Sweeney, S. H. and E. J. Feser (1998). Plant size and clustering of manufacturing activity. *Geographical Analysis 30*(1), 45–64.

Tidu, A., F. Guy, and S. Usai (2024). Measuring Spatial Dispersion: An Experimental Test on the M -Index. *Geographical Analysis 56*, 384–403.