

## Section 3. COST-BASED GEOSTATISTICS: METHODOLOGICAL OVERVIEW

### Computing the cost/surface

The first operation is to encode the spatial heterogeneity of the working area into one *cost-surface*.

This imply some modeling decisions and assumptions which are not technical but scientific in nature. In our case, we assume that the soil is homogeneous except for the areas with solid sunken structures. These structures will completely interrupt the continuity of the area, blocking the dispersion of substances.

This conceptual model yields a cost-surface with a constant value of 1 everywhere, except over the structures where it takes an infinite value. In practice, any value larger than the diameter of the region will suffice. Alternatively, can be more practical to work in the inverted scale of a *conductivity* surface. In this case, the values would be simply 1 for regular conductivity and 0 for no conductivity, or infinite cost.

Any of these alternative surfaces can be easily produced from a digital representation of the region with any GIS software, or with other spatially-capable software like R (R Core Team, 2015).

In our case, we imported the ESRI shapefiles describing the geometry of the structures into spatial classes defined in the Rpackage `sp` (Pebesma and Bivand, 2005). Then we used functions from the `raster` package (Hijmans, 2015) to produce a discretized surface with constant value 1 over the region of interest, and the function `sp::point.in.poly()` to determine which raster cells fall within the boundaries of the solid structures. These raster cells are then assigned a value of 0. We used a resolution of about 5 pixels/m.

### Computing the cost-based distances

The cost/conductivity surface is the object representing our model of the region, and from where the distances between locations can be computed. Specifically, two matrices of cost-based distances are required: one  $n \times n$  matrix with distances among the  $n$  observations, and one  $m \times n$  matrix with the distances between each observation to each one of the  $m$  prediction locations.

We used the centroids of the conductivity raster cells as prediction locations to simplify mapping, although any set of prediction locations can be used.

The computation of the distance matrices can be also be performed using a GIS software or right within R.

López-Quílez and Muñoz (2009) use the first approach with the help of a specific script `v.costdist.mat` (Muñoz 2015b) for GRASS GIS (GRASS Development

Team, 2010). For this study we used the R package **gdistance** (van Etten, 2015) instead.

This package provides the function **transition** which computes the *transition value* from every raster cell to every other *adjacent* cell. This value is computed after a user-specified function of the corresponding conductivity values. Here we used the function **min**, so that the transition value between two cells any of which with conductivity of 0 remains 0. We considered *adjacent* all the 16 cells that can be reached by one move of the king or the knight pieces in the game of Chess. The transition values are *corrected* for diagonal connections. These settings provide the maximum accuracy for the subsequent calculations of distances based on a discretized surface. Finally, the function **costDistance** uses the transition values to compute least-cost distances between two sets of points. It is based on the well known Dijkstra's algorithm for finding shortest paths between nodes in a network.

### Using cost-based geostatistical algorithms

The R package **geoRcb** (Muñoz, 2015a) is an extension of the **geoR** package (Ribeiro and Diggle, 2015) that provide alternative version of some functions, in order to make them capable of working with cost-based distances. Specifically, the functions **variog** and **likfit** feature an additional argument **dists.mat** which takes a symmetric matrix of distances between observation locations. These functions are respectively used to compute empirical variograms and fit variogram models. Finally, the alternative function **krige.conv** performs the cost-based spatial prediction through conventional kriging by taking the required distance matrices as the additional arguments **dd.dists.mat** and **dl.dists.mat**.

## Bibliography

GRASS Development Team (2010). Geographic Resources Analysis Support System (GRASS GIS) Software. USA: Open Source Geospatial Foundation. <http://grass.osgeo.org/>

Hijmans R.J. (2015). raster: Geographic Data Analysis and Modeling. R package version 2.4-15. <http://CRAN.R-project.org/package=raster>

López-Quílez A. and F. Muñoz (2009). Geostatistical computing of acoustic maps in the presence of barriers. *Mathematical and Computer Modelling* 50(5-6):929-938

Muñoz F. (2015a). geoRcb: An Extension of Package geoR that Works with Cost-Based Distances. R package version 1.7-5. <https://github.com/famuvie/geoRcb>. DOI:10.5281/zenodo.23568

Muñoz F. (2015b). v.costdist.mat: A GRASS script for computing cost-based distance matrices. Zenodo. DOI:10.5281/zenodo.23546

Pebesma E.J., R.S. Bivand, 2005. Classes and methods for spatial data in R. R News 5 (2), <http://cran.r-project.org/doc/Rnews/>

R Core Team (2015). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.R-project.org/>.

Ribeiro Jr P.J. and P.J. Diggle (2015). geoR: Analysis of Geostatistical Data. R package version 1.7-5.1. <http://CRAN.R-project.org/package=geoR>

van Etten J. (2015). gdistance: Distances and Routes on Geographical Grids. R package version 1.1-7. <http://CRAN.R-project.org/package=gdistance>