Space-constrained hierarchical clustering for the iAtlantic workshop

Pierre Legendre, Université de Montréal, May 2021

Hierarchical clustering is a well-known method of multivariate data analysis. It is used to identify groups of objects (e.g. sites) that are more similar within than among the groups. In R, at least three functions propose the Lance & Williams (1966, 1967) general agglomerative clustering method: hclust.R in {stats}, agnes.R in {cluster} and constr.hclust.R in {adespatial}. This method implements most commonly-used methods of agglomerative clustering through a unified algorithm. Before hierarchical clustering, one must compute a dissimilarity matrix **D** among the objects from the response data matrix **Y** of interest. The dissimilarity measure is chosen with respect to the mathematical nature of the data, e.g. {quantitative, presence-absence, qualitative}, and the type of information one wishes the clustering method to represent.

For some problems involving sites observed in geographic space, one may want to divide the study area (map) into groups of sites that are geographically connected to one another in addition to having similar vectors of response variables. The geographic information must be incorporated in the cluster analysis, in addition to the information from the dissimilarity matrix **D** computed from **Y**. Several authors have independently proposed to use constraints of contiguity in spatial clustering, including Lefkovitch (1978, 1980), Monestiez (1978), Lebart (1978), Roche (1978), Perruchet (1981) and Legendre & Legendre (1984). Reviews of constrained classification methods are found in Gordon (1980, 1996), Murtagh (1985) and Legendre (1987).

A constrained clustering analysis comprises four steps:

1. Compute a dissimilarity matrix **D** among the sites, which are the rows of the response data matrix **Y** of interest. For community ecologists, matrix **Y** often contains community composition data.

2. Choose a connecting scheme among the sites and produce a list of link edges forming matrix **E**.

3. Carry out constrained clustering using **D** as the response dissimilarity matrix and **E** as the spatial or temporal contiguity constraint.

4. Select a number of groups, *k*. Represent the clustering results on a map. Obtain a list of the objects members of the groups and perhaps also a dendrogram.

The classification of the objects (sites or time points) into groups may be interpreted using explanatory variables like geographic, geomorphological, or environmental variables, using RDA, multivariate regression tree analysis, or discriminant analysis. The procedure and functioning of constrained hierarchical clustering is summarized in Figure 1.

1. Compute a dissimilarity matrix D among the sites

The dissimilarity function must be chosen as a function of the mathematical nature of the data, of their ecological nature (e.g. frequencies of organisms or physical variables), and of the objective of the study. See the course segment on “Dissimilarity and transformations”.

2. Prepare the matrix of edges E

In graph theory, an *edge* is a line connecting two points in a graph. Points are also called vertices or nodes. An edge may also be called a link or a line, although the word *link* may be restricted to designate an edge that has distinct end vertices. Traditionally, a set of edges is called *E*. In undirected graphs, an edge is described by an unordered pair of points; edges don't have directions in this type of graph.

The simplest way to build a matrix of edges **E**, for small data sets (say, up to 50 points), is to write it by hand in a text editor or a spreadsheet.

• Write a data frame with two columns in which each row represents an edge of the connexion diagram (see example below).

• In the list of edges, the points are represented by numerals corresponding to the positions of these points in data matrix **Y**. Examples: the first site in matrix **Y** is called *1* in data frame **E**, whatever its real identifying label, and the 25th site in the list is called *25*. Do not use site labels, e.g. {"SiteA", "SiteB", "SiteC"}, to describe edges unless the site labels are the same as the site positions in **Y**. The site numerals used in edge matrix **E** must follow the order of the objects in matrix **Y**.

• An edge is represented by two numbers, for example 3 and 15. The order of these numerals in the first or second column does not matter. It is also useless to repeat the edges, placing the numerals in the inverse order, for example [3 8] followed by [8 3].

As an example, the following edge matrix **E** represents the Delaunay triangulation in Figure 1c:

from	to
1	2
1	3
1	4
2	3
2	5
2	9
3	4
3	5
3	6
4	6
4	7
5	6
5	8
5	9
6	7
6	8
7	8
7	9
8	9

Note that the order of the objects in data file **Y**, shown in Figure 1a, has to correspond to the object numbers in **E**, for the software to interpret correctly the relationships between the list of objects in **Y** and the edges in **E**. However, edges may be listed in any order in matrix **E**. In this example, the object labels (1 to 9) are also the object input order. The object labels could be different, e.g. "Site.z", "Site.p", "Site.x", etc.

Another example of a data frame representing a matrix **E** is provided in folder "Chesapeake-Data" of the Chesapeake Bay exercises of this course. The file is called E50.

It is not necessary to include *rownames* in that file; although they were added for elegance in matrix E50. For column names, "from" and "to" are traditionally used to make the file understandable by other researchers, but any column names may be used, or none at all; they are not read or used by the function.

For larger data sets, containing sites at random positions or forming regular grids, Dray (2020) wrote an extensive tutorial, which clearly explains how to proceed to generate matrix **E** for regular grids of points, or for irregularly placed sites on geographic surfaces. See section 2 of the tutorial, "Building spatial neighbourhood". To access the tutorial, type

```
library(adespatial)
vignette("tutorial", "adespatial")
```

and export the tutorial to a pdf file. In the tutorial, the neighbourhood data frames **E** are generated by specialized functions of the {spdep} package.

In the case of a spatial transect or a time series, one does not have to create matrix **E** explicitly. If the software allows it (this is the case with function constr.hclust.R), this task will be done by the constrained clustering software, which will assume that the sites along the transect, or the observations along the time series, are listed in **Y** according to their positions in the sequence.

3. Carry out constrained clustering analysis

The constrained clustering algorithm described in Figure 1 is implemented in the R function constr.hclust.R of {adepsatial}. This algorithm was originally developed by computer scientist Alain Vaudor at Université de Montréal for the fish postglacial dispersal study published by Legendre & Legendre (1984). The original algorithm (called BioGeo) only implemented hierarchical agglomerative proportional-link linkage clustering with constraint of spatial contiguity. On output, the agglomerative clustering steps were represented in the form of maps. The algorithm was generalized to the Lance & Williams algorithm in Legendre & Legendre (2012, section 13.3.2). From 2011 to 2020, this algorithm was implemented in successive function versions, called const.clust.R (in R language) by P. Legendre and constr.hclust.R (with functions for intensive calculation coded in C) by G. Guénard. The procedure is described in Guénard & Legendre (2021). The algorithm is described in the extended caption of Figure 1 of the present paper.

The constr.hclust.R function produces an output object with classes "constr.hclust" and "hclust". This output object can readily be used by function plot.hclust of {stats} to produce a dendrogram. The resulting dendrogram is likely to present reversals, i.e. instances where the fusion distance at a clustering step is lower than that of the previous step. Reversals produce dendrograms that are not *monotonic*, or fully nested. Reversals may also be produced in ordinary hierarchical cluster analysis. Their generation is explained in Legendre & Legendre (2012, section 8.6). – For constrained clustering, Ferligoj & Batagelj (1982) showed that the introduction of relational constraints, e.g. spatial contiguity, in the clustering algorithm may produce reversals with any of the hierarchical clustering methods included in the Lance & Williams algorithm, except complete linkage.

Constrained clustering can also be applied to time series of univariate or multivariate data. An example is shown in the "Practical in R, Chesapeake Bay data", section 4.2.2.

4. An example of spatially-constrained clustering

A simple artificial example presented in the Guénard & Legendre (2012) paper will be used to illustrate the calculation of spatially constrained clustering. The data, illustrated in Figure 2, are the following:

```
# For simplicity of the example, matrix Y only contains one response variable
var = c(1.5, 0.2, 5.1, 3.0, 2.1, 1.4)
ex.Y = data.frame(var)

# Site coordinates, matrix xy
x.coo = c(-1, -2, -0.5, 0.5, 2, 1)
y.coo = c(-2, -1, 0, 0, 1, 2)
ex.xy = data.frame(x.coo, y.coo)

# Matrix of connecting edges E
from = c(1,1,2,3,4,3,4)
to = c(2,3,3,4,5,6,6)
ex.E = data.frame(from, to)
```

Note how some possible adjacency edges, i.e. those between sites (1, 4), (1, 5), (2, 6) and (5, 6), were not incorporated in the matrix of edges **E** because these edges cross land masses. The list of edges can be customized in order to adapt it to the analysis to particularities of the landscape or to particular ecological hypotheses to be tested. Here the objective was to group aquatic sites on the basis of some underwater characteristic, like sediment characteristics or faunal composition.

```
# Carry out constrained clustering analysis –

library(adespatial)

cclust.out <-
constr.hclust(
      dist(ex.Y),                    # Response dissimilarity matrix
      method="ward.D2",              # Clustering method[1]
      links=ex.E,                    # File of link edges (constraint) E
      coords=ex.xy)                  # File of geographic coordinates

# Plot the results for 2 to 5 spatially-constrained clusters (Figure 3)

par(mfrow=c(1,2))
plot(cclust.out,k=2,links=TRUE,xlab="X (km)",ylab="Y (km)",cex=1.5)
plot(cclust.out,k=3,links=TRUE,xlab="X (km)",ylab="Y (km)",cex=1.5)

par(mfrow=c(1,2))
plot(cclust.out,k=4,links=TRUE,xlab="X (km)",ylab="Y (km)",cex=1.5)
plot(cclust.out,k=5,links=TRUE,xlab="X (km)",ylab="Y (km)",cex=1.5)
```

[1] Ward's minimum variance method selected here (method="ward.D2") is a general-purpose clustering method optimizing Ward's least-squares criterion. Method "ward.D2" (default value in constr.hclust.R) implements the Ward clustering criterion, whereas method "ward.D" does not (Murtagh and Legendre, 2014). See also the documentation file of {stats} function hclust.R.

Produce numeric vectors describing the partition of the sites into k groups

```
( k2.groups = as.numeric(cutree(cclust.out, k=2)) )
# [1] 1 1 2 2 2 2
( k3.groups = as.numeric(cutree(cclust.out, k=3)) )
# [1] 1 1 2 3 3 3
( k4.groups = as.numeric(cutree(cclust.out, k=4)) )
# [1] 1 1 2 3 3 4
( k5.groups = as.numeric(cutree(cclust.out, k=5)) )
# [1] 1 2 3 4 4 5
```

Unconstrained clustering solution

```
hclust.out = hclust(dist(ex.Y), method="ward.D2")
```

Produce dendrograms for the constrained (left) and unconstrained (right) solutions (Figure 4)

```
par(mfrow=c(1,2))
stats:::plot.hclust(cclust.out, hang=-1)
plot(hclust.out, hang=-1)
```

The two dendrograms are very different yet they are both mathematically correct. The constrained clustering solution (left) describes a geographic partition of the map into groups of contiguous points that are fairly similar in the values of the variable in file ex.Y. The unconstrained solution (right) simply partitioned the points according to the values of the variable, without attempt to take geography into account.

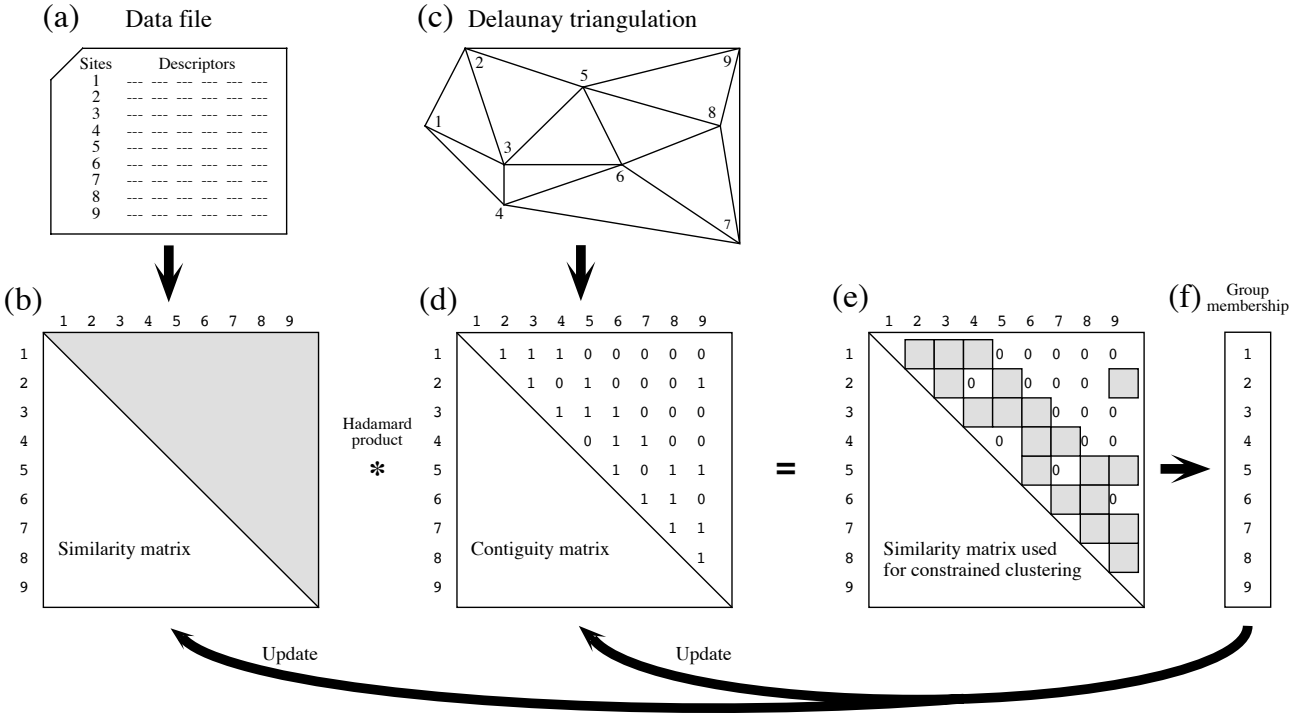

Figure 1 – Summary of the spatially-constrained clustering procedure. It is applicable to any clustering method compatible with the Lance & Williams agglomerative clustering algorithm.

• Arrow from (a) to (b) – The data matrix **Y** (data file in panel a) is used to compute a dissimilarity matrix. Here the site-by-site resemblance is expressed by a similarity matrix **S**, where $s_{ij} = 1 - d_{ij}$; the reason is explained below.

• Arrow from (c) to (d) – The graph edges of the Delaunay triangulation (c) are written to matrix **E** (shown in section 2 of the main text), which is used to generate the contiguity matrix (d).

• Matrix(e) = Matrix(b) * Matrix(d), where " * " designates the Hadamard product, or cell-by-cell product, of two matrices. Similarities are used in (b) instead of dissimilarities in order to make the Hadamard product create zeros in (e) for pairs of unconnected sites; these sites are not connected by an edge in (c) and have contiguity of 0 in (d). Zeros in (e) mean "not connected" and "do not cluster". The use of similarities in (b) and (e) avoids the confusion with a value d_{ij} of zero, which could be a legitimate value. Grey squares in (e) designate the connected pairs of points in graph (c).

• Vector (f) – Objects are labelled 1 to 9 in this example. At the start of the clustering process, each object is labelled as a separate group in vector (f).

• Constrained clustering – Among the grey squares in (e), the one showing the highest similarity determines the next fusion of objects or groups. The group membership vector (f) is then updated by attributing the same identifier to the two objects or groups that have clustered.

• Matrices (b) and (d) are also updated to account for that fusion. In (b), cell fusions are computed following the Lance & Williams general agglomerative formula. In (d), the contiguities are updated by the rule that any point that was connected to one of the two points that have clustered is now connected to the new row and column representing the new cluster of points.

• Matrix (e) is recomputed, ready for the next fusion of objects or groups.

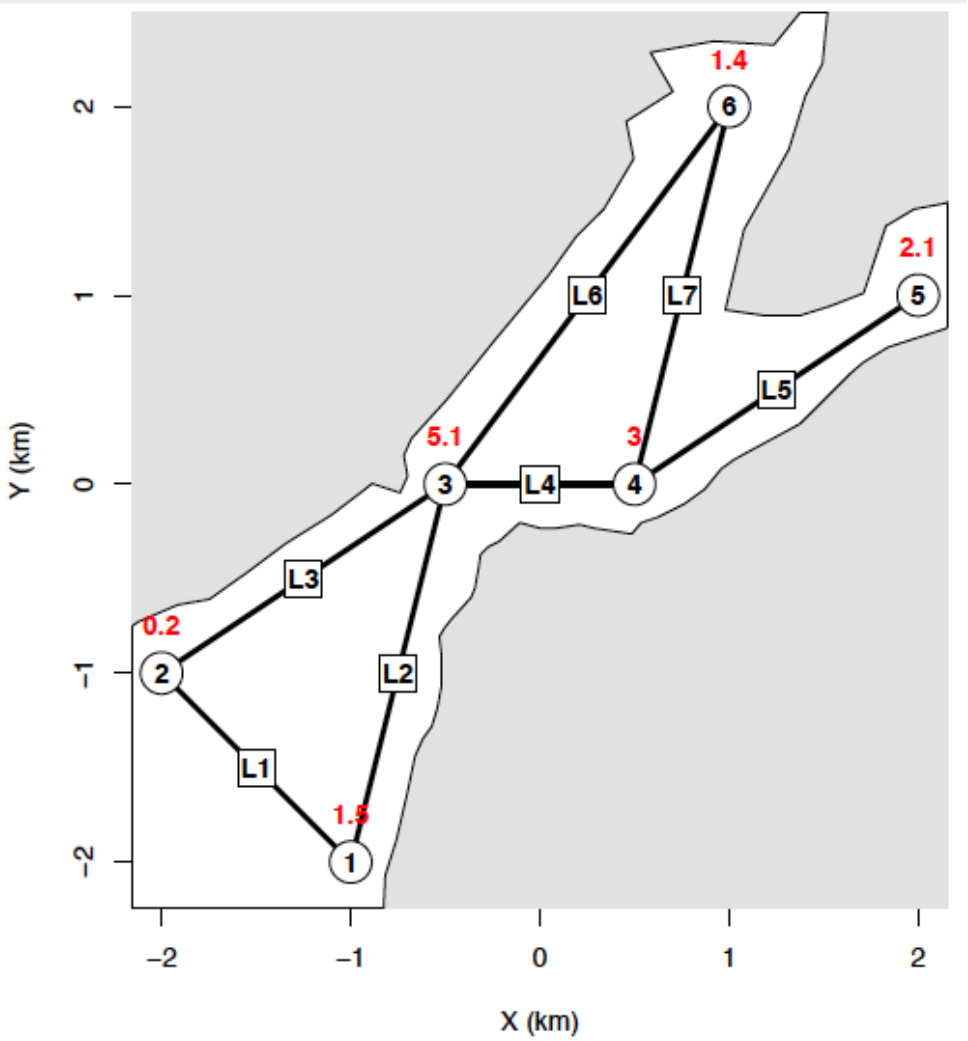

Figure 2 – Artificial example of Guénard & Legendre (2021, Figure 2) illustrating spatially-constrained clustering. The grey areas represent unsuitable habitats, like land masses in a study of aquatic ecosystems, or the converse, water masses in a study of the dispersion of terrestrial ground-dwelling (i.e. non-flying) organisms.

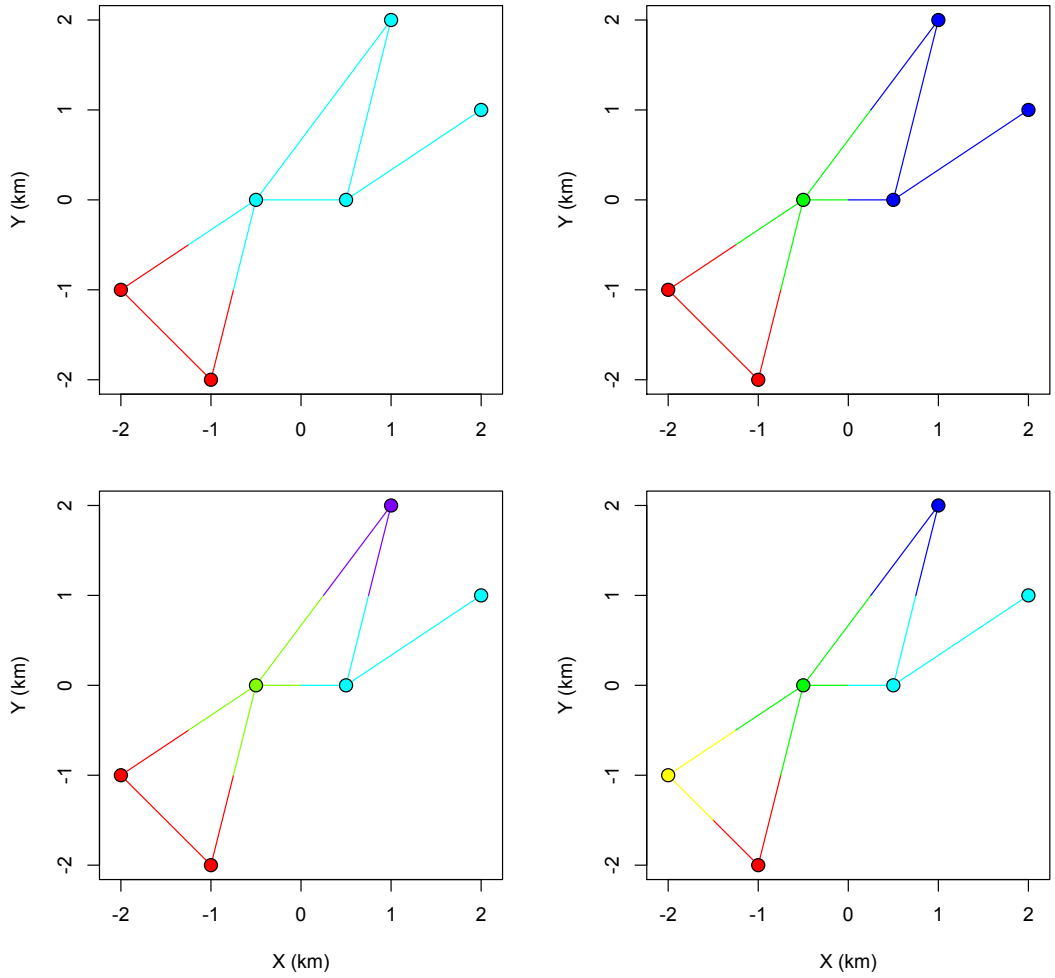

Figure 3 –Constrained clustering maps for the artificial example of Guénard & Legendre (2021). Top: k=2 and k=3 groups. Bottom: k=4 and k=5 groups. Colours may sometimes look fairly similar.

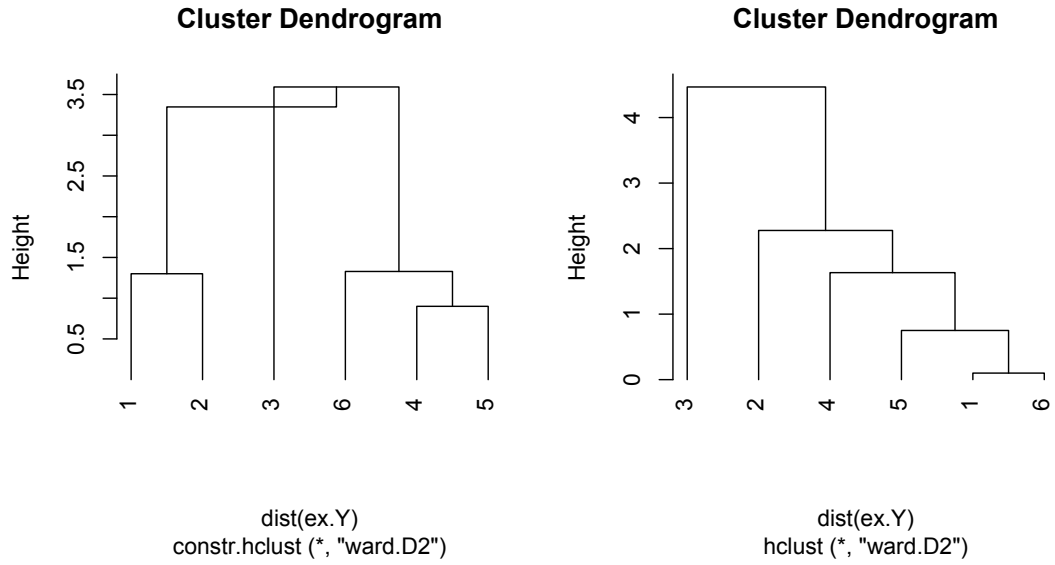

Figure 4 – Dendrograms for the constrained (left) and unconstrained (right) solutions.

References

Borcard, D., F. Gillet & P. Legendre. 2011. *Numerical ecology with R.* Use R! series, Springer Science, New York. xi + 306 pp.

Borcard, D., F. Gillet & P. Legendre. 2018. *Numerical ecology with R, 2nd edition.* Use R! series, Springer International Publishing AG. xv + 435 pp.

De'ath, G. 2002. Multivariate regression trees: a new technique for modeling species-environment relationships. *Ecology* 83: 1105-1117.

Ferligoj, A. & V. Batagelj. 1982. Clustering with relational constraint. *Psychometrika* 47: 413–426.

Gordon, A.D. 1980. Methods of constrained classification. Pp. 149–160 in: R. Tomassone [ed.], Analyse de données et informatique. INRIA, Le Chesnay .

Gordon, A. D. 1996. A survey of constrained classification. *Computational Statistics & Data Analysis* 21: 17–29.

Guénard, G. & P. Legendre. 2021. Hierarchical clustering with contiguity constraint in R. *Journal of Statistical Software* (under review).

Lance, G. N. & W. T. Williams. 1966. A generalized sorting strategy for computer classifications. *Nature (London)* 212: 218.

Lance, G. N. & W. T. Williams. 1967. A general theory of classificatory sorting strategies. I. Hierarchical systems. *Computer Journal* 9: 373–380.

Lebart, L. 1978. Programme d'agrégation avec containtes (C. A. H. contiguïté). Cahiers *d'Analyse des Données* 3: 275–287.

Lefkovitch, L. P. 1978. Cluster generation and grouping using mathematical programming. *Mathematical Biosciences* 41: 91-110.

Lefkovitch, L. P. 1980. Conditional clustering. *Biometrics* 36: 43–58.

Legendre, P. 1987. Constrained clustering. Pp. 289–307 in: P. Legendre & L. Legendre [eds.] *Developments in numerical ecology*. NATO ASI series, Vol. G-14. Springer-Verlag, Berlin.

Legendre, P. & L. Legendre. 2012. *Numerical ecology, 3rd English edition.* Elsevier Science BV, Amsterdam.

Legendre, P. & V. Legendre. 1984. Postglacial dispersal of freshwater fishes in the Québec peninsula. *Canadian Journal of Fisheries and Aquatic Sciences* 41: 1781–1802.

Monestiez, P. 1978. Méthodes de classification automatique sous contraintes spatiales. Pp. 367–379 in: J. M. Legay & R. Tomassone [eds.] *Biométrie et écologie*. Institut national de la Recherche agronomique, Jouy-en-Josas.

Murtagh, F. 1985. A survey of algorithms for contiguity-constrained clustering and related problems. *Computer Journal* 28: 82–88.

Murtagh, F. and P. Legendre. 2014. Ward's hierarchical agglomerative clustering method: which algorithms implement Ward's criterion? *Journal of Classification* 31: 274-295.

Perruchet, C. 1981. Classification sous contrainte de contiguïté continue. Pp. 71-92 in: *Classification automatique et perception par ordinateur.* Séminaires de l'Institut national de Recherche en Informatique et en Automatique (C 118), Rocquencourt.

Roche, C. 1978. Exemple de classification hiérarchique avec contrainte de contiguïté. Le partage d'Aix-en-Provence en quartiers homogènes. *Cahiers d'Analyse des Données* 3: 289–305.