Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *doi: 10.18637/jss.v000.i00*

Modelling Prehistorical Iconographic Compositions. The R package decorr

Thomas Huet
UMR 5140

Abstract

By definition, Prehistorical societies are characterised by the absence of a writing system. Prehistorical times cover more than 99% of the human living. Even if it is being discussed, first symbolic manifestations start around 200,000 BC (d'Errico and Nowell 2000). The duration from first symbolic expressions to start of writing represents 97% of the human living. In illiterate societies, testimonies of symbolic systems mostly come from iconography (ceramic decorations, rock-art, statuary, etc.) and signs are displayed mostlty a discontinuous figures which can have different relationships one with another. An graphical composition can be "read" as a spatial distribution of features having intrinsic values possibily having meaningful relationships one with another depending on their pairwise spatial proximities.

To understand meaningful associations of signs, geometric tools, graph analysis and statistical analysis offer great tools to recognize iconographical patterns and to infer collective conventions. We present the **decorr** R package which ground concepts, methods and tools to analyse ancient graphical systems.

Keywords: Iconography, Prehistory, Graph Theory, Graph Drawing, Spatial Analysis, R.

1. Introduction

For decades, study of ancient iconography was linked to history of religion because closely linked to symbolism, believes and religions. Since the *New Archaeology* developpement during the 60's (Clarke 2014), symbolic expressions start to be studied with the same formal methods (statistics, seriations, distribution maps, etc.) as any another aspect of social organisation: settlement patterns, tools *chaine opÃľratoire*, susbsitence strategies, etc. (Renfrew and Bahn 1991), (Leroi-Gourhan 1992). But unlike many aspects of the material culture – a flint blade for cutting, a pottery for containing, a house for living –, the function of an iconographic composition cannot be drawn directly from itself. Whether study of ancient iconography had undergone significative improvements at the site scale – with GIS, database, paleoclimatic restitutions, etc. – and at the sign scale with the development of archaeological sciences – radiocarbon dating, use-wear analysis, elemental analysis, etc. –, these improvement do not necessarly help to understand the semantic content of the iconography. Semantics or semiotics can be defined as a system of conventional signs organised also in conventional manners. Until our days, formal methods to study ancient iconography Semantics, has been mostlty been grounded (explicitly or not) on the prime principle of Saussurian linguistic: the 'linearity of the signifier' (De Saussure 1989). Writing is one of the most rational semiographical system with a clear distinction between signified and signifier – specially in alphabetic and binary writings – and the development of the signified on a horizontal, vertical or boustrophedon axis. Let us take the example of the word `"art"` which contains three vertices (`a`, `r`, `t`) and two edges (one between `a` and `r`, the other between `r` and `t`). In R, these features, concatenated in this order with a `paste0()`, is `art`, and not `rat`,

Figure 1: concatenate of `a`, `r` and `t` graphical units (GUs) is `art`.

But, as stated, in Prehistorical the writing system does not exists. Spatial relationships between graphical features, or graphical units (GUs) are not necessarly linear and directed but could most probably be more multi-directional and undirected: the direction of the interactions of pairwise GUs can be in any order.

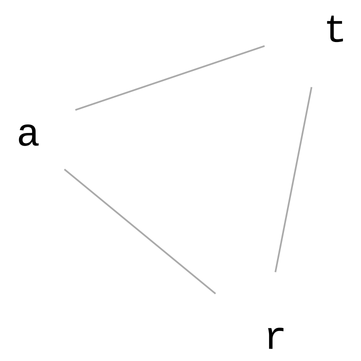


Figure 2: Potential spatial relations between `a`, `r` and `t` GUs.

Applying the Saussurian model to any prehistorical graphical content had led to tedious division of the iconographical content with, for example, graphical as a relationship of figures grouping GUs, patterns grouping figures, motives grouping patterns, etc., until the entire decorated support is described and can be compared to another decoration (XXX). But during this *decomposition* process, many imprecisions occur:

- groups and relationships are often defined empirically
- their level of significance are often implicit
- the iconographical and spatial proximities between GUs and categories of GUs are not quantified

Furthermore, due to the inherent variability of iconography, most of the studies developp proper descriptive vocabularies, singular relationships of categories, idosyncratic methods in a site-dependend or period-dependend scales. This limits drastically the possibility to conduct cross-cultural comparisons and to draw a synthesis of humankind's symbolism at a large scale and over the long-term.

In this article we present the R package **decorr**. Its purpose is to formalise a method based on geometric graphs to analyse any graphical content. As any formal system, iconography can be modelled as spatial features related one with the other depending on rules of spatial proximities. The idea is that a graphical system can be represented by vertices connected (or not) to each other with edges. This package has been grounded on the seminal work of C. Alexander (Alexander 2008) and its first IT implementation by T. Huet (Huet 2018).

2. Model

Graph theory offers a conceptual framework and indices (global at the entire graph scale, local at the vertex scale) to deal with notions of networks, relationships and neighbourhoods. The spatial levels of the GUs can be retrieve by a planar graph (Graph Theory) and a spatial

(GIS) analysis. Nodes and edges – repectively for GUs and their connexion – are created on a GIS interface. In the GIS, the decoration figure is open in the first place in a new project with no projection. The decoration image will be considerated as the basemap of the project and will cover the region of interest of the analysis. The decoration image can be binarized where GUs are considerated active and the undecorated parts of the support, or background, are considerated inactive. After what, the decoration image is tiled. A simplier solution will be to create directly centroids over the GUs. The x and y coordinates of the nodes are relative to the decoration and measured in pixels. Exist a link between a couple of GUs when these graphical units share a border. A planar graph is constructed from grahical units (nodes) and their proximity links (edges). This model is a Voronoi diagram of the support where the Voronoi seeds are the GUs. Its geographical equivalent is a Thiessen polygon.

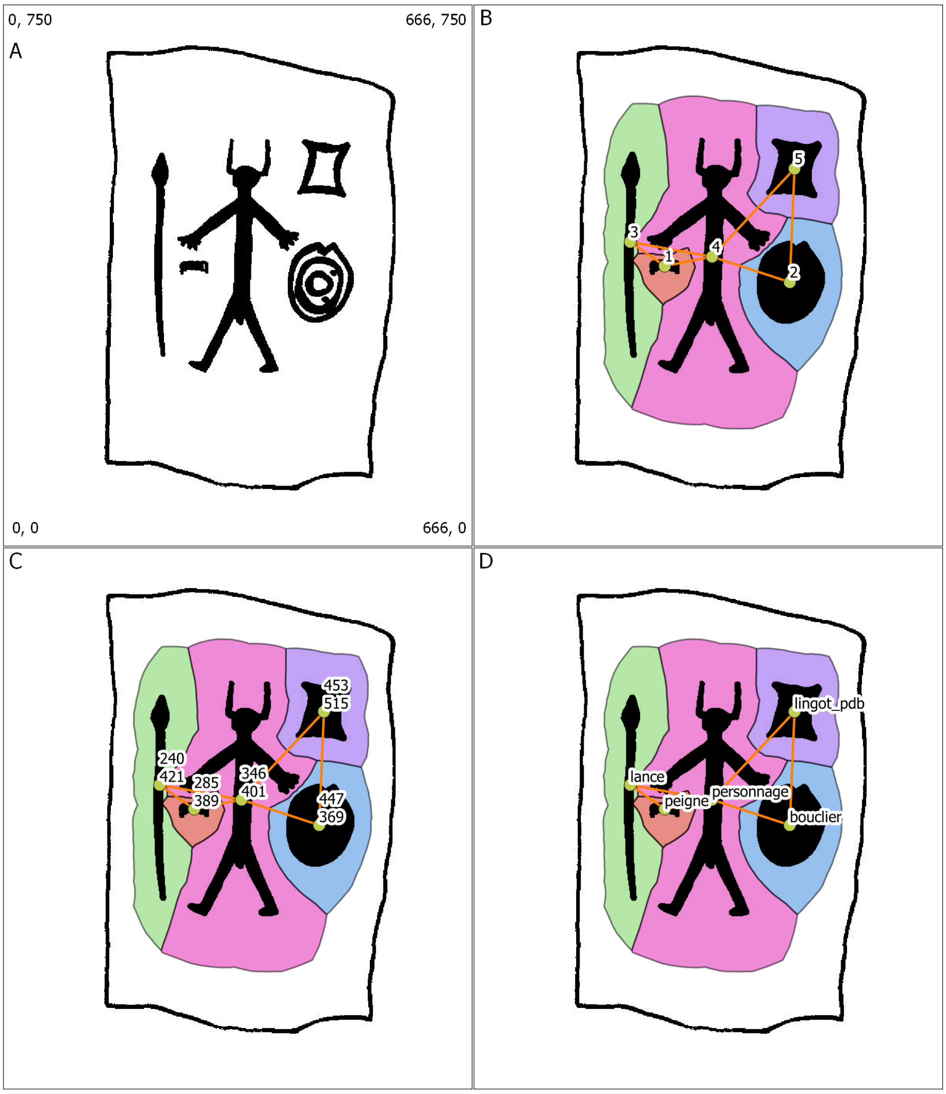


Figure 3: GIS interface. A) Original decoration of the Late Bronze Age Cerro Muriano 1 stelae (drawing: Díaz-Guardamino Uribe (2010)) with its extent (`xmin, xmax, ymin, ymax`); B) After the polygonisation of the GUs, including the border of the stelae, the Voronoi cells, the centroid of GUs and the links between GUs having adjacent cells (ie, sharing a border) are calculated; C) For each GUs, x and y are calculated; D) At least one variable, like the `type` of the GUs is defined in order to compute composition analysis.

This model has a minimal of *a priori* definitions. Those only concern the GUs (type, technology, color, orientation, size, etc.). Between two GUs the links are conventionaly represented with a plain line. But sometimes a graphical unit can be divided into a main unit (eg, a man) and attribute units (eg, a helmet, a sword). So, the links between the main unit and the attribute units are directed and displayed with a dashed line.

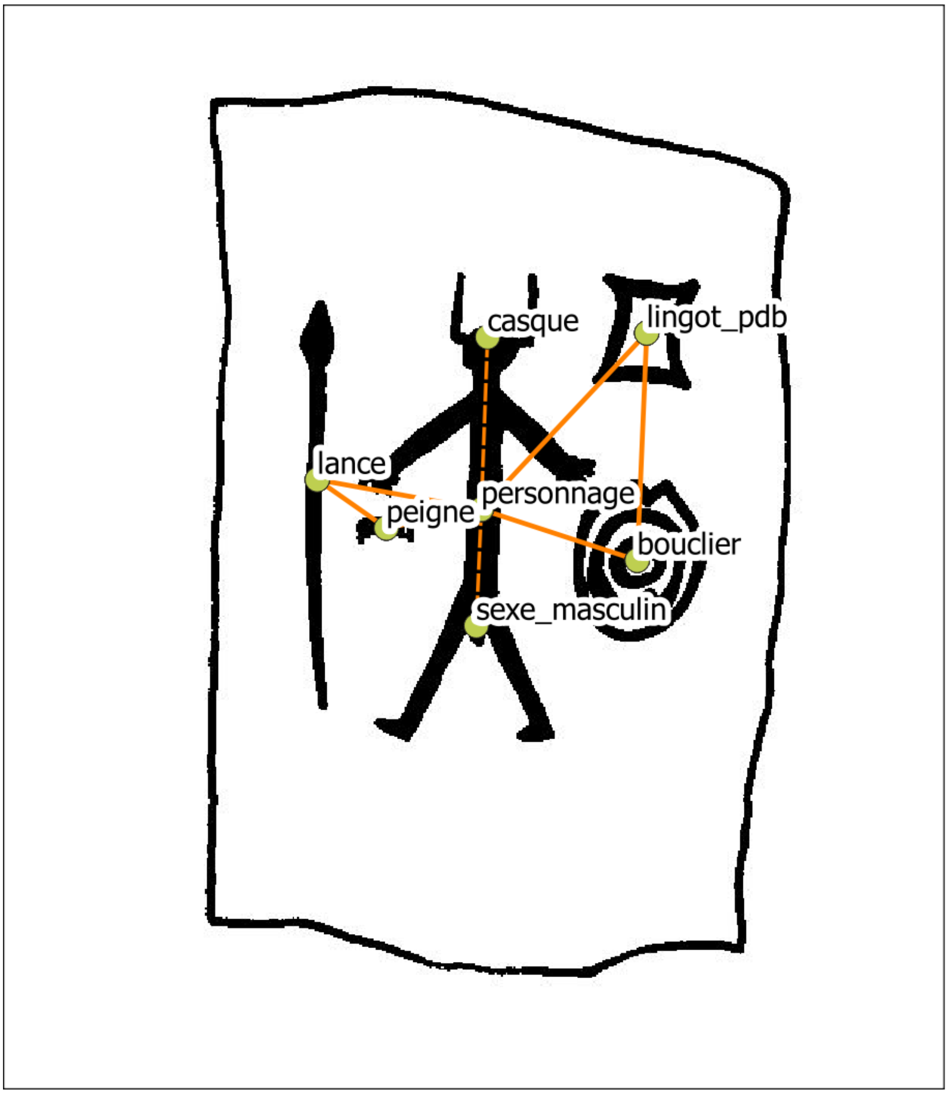


Figure 4: GIS interface. The graphical unit 'casque' (helmet) and the 'sexe_masculin' (male sex) are two attributes of the graphical unit 'personnage' (character).

3. The R package decorr

The **decorr** package can be downloaded from GitHub

```
R>   devtools::install_github("zoometh/iconr")
```

3.1. External package

The **decorr** package imports the following packages:

- **magick** for image manipulation (Ooms 2018)

- **igraph** for graph and network analysis (Csardi and Nepusz 2006)

- **rgdal** to read shapefiles of nodes and/or edges (Bivand, Keitt, and Rowlingson 2019)

- **grDevices** for colors and font plotting, **graphics** for graphics, **utils** and **methods** for formally defined methods and *varia* methods (all combinations, etc.) (R Core Team 2019)

3.2. External data

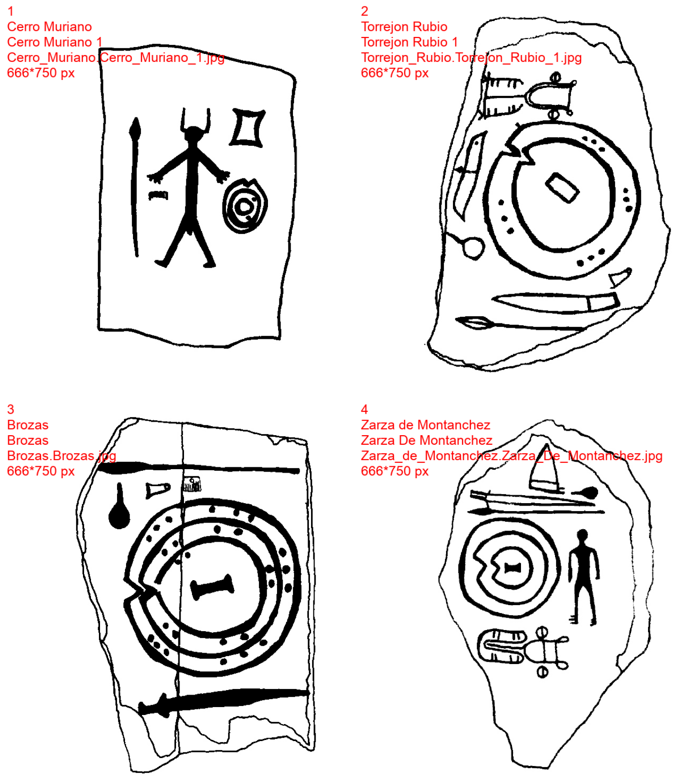
The training dataset is a selection of four drawing of stelae belonging to the Late Bronze age of the SW Iberian peninsula.

idf	site	decor	img
1	Cerro Muriano	Cerro Muriano 1	Cerro_Muriano.Cerro_Muriano_1.jpg
2	Torrejon Rubio	Torrejon Rubio 1	Torrejon_Rubio.Torrejon_Rubio_1.jpg
3	Brozas	Brozas	Brozas.Brozas.jpg
4	Zarza de Montanchez	Zarza De Montanchez	Zarza_de_Montanchez.Zarza_De_Montanchez.jpg

Table 1: The studied corpus, the imgs.tsv dataframe

The default dataframe storing the inventory of decorations is `imgs`. The field `imgs$idf` is the short name of the decoration, useful during statistical analysis. The concatened of `imgs$site` and `imgs$decor` is the primary key of each decoration. All decoration drawings belongs to (Díaz-Guardamino Uribe 2010).

At first the drawing dataset can be checked by using the `imgs` dataframe and the **magick**



3.3. Functions

The first step is to load `nodes`, `edges` and `images` dataframes in order to construct a graph for each decoration. At the first, the training dataset is in the `extdata` folder of the **decorr**.

```
R> imgs <- read.table(system.file("extdata", "imgs.csv", package = "decorr"),
+                      sep="\t",stringsAsFactors = FALSE)
R> nodes <- read.table(system.file("extdata", "nodes.csv", package = "decorr"),
+                       sep="\t",stringsAsFactors = FALSE)
R> edges <- read.table(system.file("extdata", "edges.csv", package = "decorr"),
+                       sep="\t",stringsAsFactors = FALSE)
```

Once done, the list of graphs can be stored with the `list_dec()` function.

3.4. list$_{d}ec$()

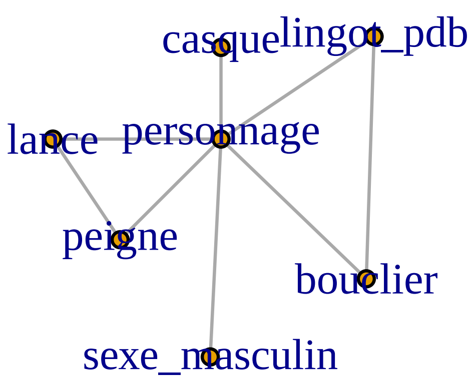
```
list_dec(imgs,
         nodes,
         edges,
         var="type")
```

This function allows to store undirected graphs for each decorations stored into `nodes`, `edges` and `images` dataframes and store the graphs in a list. The join between these dataframes is done on the two fields `site` and `decor`.

```
R> library(decorr)
R> lgrph <- list_dec(imgs,nodes,edges,var="type")
```

The first graph of can be plotted

```
R> plot(lgrph[[1]],
+       vertex.size = 10,
+       vertex.label.cex =.8)
```



The `labels_shadow()` function is a re-use of the `shadowtext()` function from the **TeachingDemos** package (Snow 2020).

The others **decorr** package functions can be divided into:

1. functions for a single decoration

2. **functions for comparisons between different decorations**

3.5. Single decoration functions

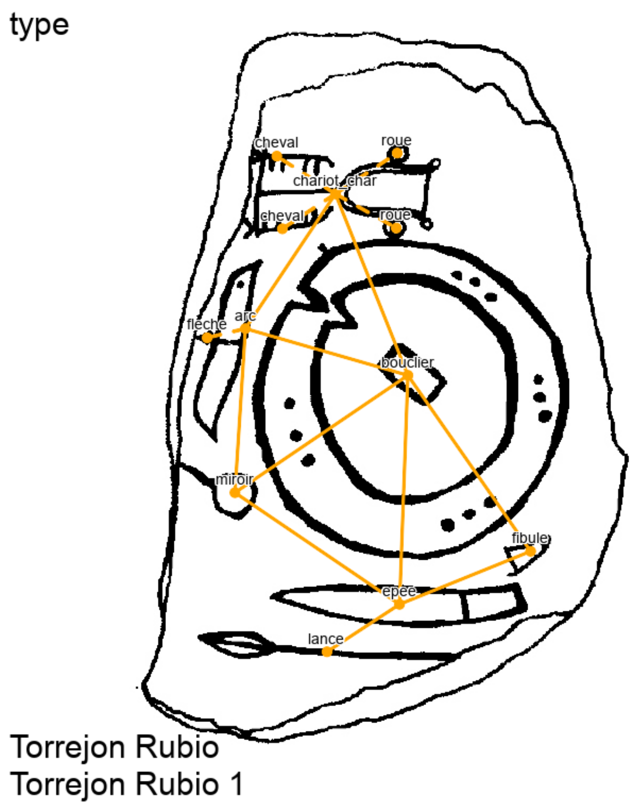
Functions allowing to model a single decoration with a geometric graph are:

- `read_nds()` and `read_eds()` allow to read respectively a file of nodes and a file of edges (`.tsv` or `.shp` files)

- `plot_dec_grph ()` allows to plot a geometric graph over a decoration image

The Torrejon Rubio 1 stele decoration graph, with nodes, type of nodes, and edges, can be plotted

```
R> # library(decorr)
R> nds.df <- read_nds(site = "Torrejon Rubio", decor = "Torrejon Rubio 1", dev = "
+                     doss = system.file("extdata", package = "decorr"))
R> eds.df <- read_eds(site = "Torrejon Rubio", decor = "Torrejon Rubio 1", dev = "
+                     doss = system.file("extdata", package = "decorr"))
R> img.graph <- plot_dec_grph(nds.df = nds.df,
+                             eds.df = eds.df,
+                             site = "Torrejon Rubio",
+                             decor = "Torrejon Rubio 1",
+                             doss = system.file("extdata", package = "decorr"),
+                             lbl.txt = "type",
+                             shw = c("nodes","edges"))
R> plot(img.graph)
```


type
cheval
roue
chariot_char
cheval
roue
fleche
arc
bouclier
miroir
fibule
epee
lance
Torrejon Rubio
Torrejon Rubio 1

3.6. Decoration comparisons function

Functions allowing to compare different decorations with geometric graphs are:

- `list_nds_compar()` **and** `list_eds_compar()` **allow to compare respectively the common nodes and the common edges between two decorations**

- `plot_nds_compar()` **and** `plot_eds_compar()` **allow to plot and save two figures side-by-side for a decorations pairwise with, respectively, common nodes and common edges identified**

- `same_nds()` **and** `same_eds()` **allow to repectively count matching nodes and matching edges between decoration pairwises**

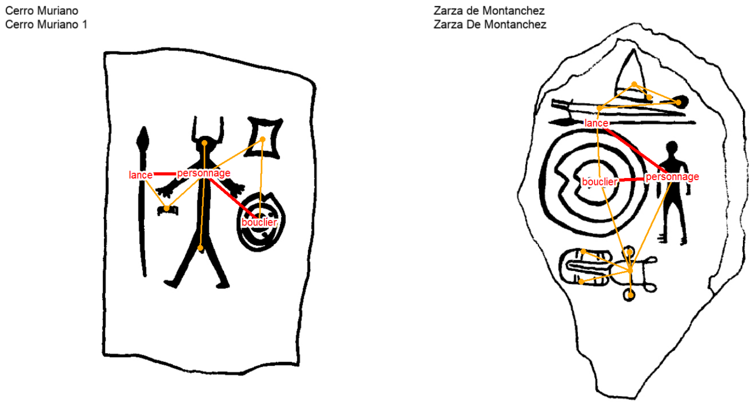
Comparisons between pairwise of decorations are first stored into list. These comparisons are performed for nodes and/or edges. There are four (4) decorations in the default dataset, so there is $\frac{4!}{(4-2)!2!} = 6$ **pairwise comparisons**

	decorA	decorB
1	Cerro Muriano 1	Torrejon Rubio 1
2	Cerro Muriano 1	Brozas
3	Cerro Muriano 1	Zarza De Montanchez
4	Torrejon Rubio 1	Brozas
5	Torrejon Rubio 1	Zarza De Montanchez
6	Brozas	Zarza De Montanchez

Table 2: comparison dataframe

The `plot_nds_compar()`**and** `plot_eds_compar()` **functions create a** `.png` **image of two decorations plotted side-by-side with common nodes or edges identified. Functions returns also the name of the image**

```
R> library(magick)
R> eds_compar <- plot_eds_compar(g.compar, c(1,4), doss = system.file("extdata", p
R> plot(image_read(eds_compar))
```



compare decorations '1' and '4' on 'type'

same_nds() and same_eds() allow to repectively count matching nodes and matching edges between decoration pairwises. The result is a square matrix with all pairwise comparisons and the number of common nodes or edges in the cells.

```
R> df.same_edges <- same_eds(lgrph,"type")
R> print(xtable::xtable(df.same_edges,
+                        caption="same edges",
+                        label="Test_table_2",
+                        size=7,
+                        digits=c(0)),
+         include.rownames=TRUE)
```

For these two last exemples, the edges comparisons between the decoration 1 (Cerro Muriano 1) and the decoration 4 (Zarza de Montsanchez) show that they have two common edges.

4. Illustrations

	1	2	3	4
1	0	0	1	2
2	0	0	3	7
3	1	3	0	1
4	2	7	1	0

Table 3: same edges

The example will be to classify decoration by their decoration

5. Summary and discussion

As usual . . .

Computational details

If necessary or useful, information about certain computational details such as version numbers, operating systems, or compilers could be included in an unnumbered section. Also, auxiliary packages (say, for visualizations, maps, tables, . . .) that are not cited in the main text can be credited here.

The results in this paper were obtained using R 3.4.1 with the MASS 7.3.47 package. R itself and all packages used are available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/`.

Acknowledgments

All acknowledgments (note the AE spelling) should be collected in this unnumbered section before the references. It may contain the usual information about funding and feedback from colleagues/reviewers/etc. Furthermore, information such as relative contributions of the authors may be added here (if any).

References

Alexander C (2008). "The Bedolina map – an exploratory network analysis." In A Posluschny, K Lambers, I Herzog (eds.), *Layers of Perception. Proceedings of the 35th International Conference on Computer Applications and Quantitative Methods in Archaeology (CAA),* *Berlin, 2.-6. April 2007*, pp. 366–371. Koll. Vor- u. FrÃijhgesch. doi:https://doi.org/10.11588/propylaeumdok.00000512.

Bivand R, Keitt T, Rowlingson B (2019). *rgdal: Bindings for the 'Geospatial' Data Abstraction Library*. R package version 1.4-7, URL https://CRAN.R-project.org/package=rgdal.

Clarke DL (2014). *Analytical archaeology*. Routledge.

Csardi G, Nepusz T (2006). "The igraph software package for complex network research." *InterJournal*, Complex Systems, 1695. URL http://igraph.org.

De Saussure F (1989). *Cours de linguistique générale*, volume 1. Otto Harrassowitz Verlag.

d'Errico F, Nowell A (2000). "A new look at the Berekhat Ram figurine: implications for the origins of symbolism." *Cambridge Archaeological Journal*, 10(1), 123–167.

Díaz-Guardamino Uribe M (2010). *Las estelas decoradas en la Prehistoria de la Península Ibérica*. Ph.D. thesis, Universidad Complutense de Madrid, Servicio de Publicaciones.

Huet T (2018). "Geometric graphs to study ceramic decoration." In M Matsumoto, E Uleberg (eds.), *Exploring Oceans of Data, proceedings of the 44th Conference on Computer Applications and Quantitative Methods in Archaeology, CAA 2016*, pp. 311–324. Archaeopress.

Leroi-Gourhan A (1992). *L'art pariétal: langage de la préhistoire*. Editions Jérôme Millon.

Ooms J (2018). "Magick: advanced graphics and image-processing in R." *CRAN. R package version*, 1.

R Core Team (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Renfrew C, Bahn PG (1991). *Archaeology: theories, methods and practice*, volume 2. Thames and Hudson London.

Snow G (2020). *TeachingDemos: Demonstrations for Teaching and Learning*. R package version 2.12, URL https://CRAN.R-project.org/package=TeachingDemos.

A. More technical details

Appendices can be included after the bibliography (with a page break). Each section within the appendix should have a proper section title (rather than just *Appendix*).

For more technical style details, please check out JSS's style FAQ at https://www.jstatsoft.org/pages/view/style#frequently-asked-questions which includes the following topics:

- Title vs. sentence case.
- Graphics formatting.
- Naming conventions.
- Turning JSS manuscripts into **R** package vignettes.
- Trouble shooting.
- Many other potentially helpful details...

B. Using BibTeX

References need to be provided in a BibTeX file (`.bib`). All references should be made with `\cite`, `\citet`, `\citep`, `\citealp` etc. (and never hard-coded). This commands yield different formats of author-year citations and allow to include additional details (e.g., pages, chapters, ...) in brackets. In case you are not familiar with these commands see the JSS style FAQ for details.

Cleaning up BibTeX files is a somewhat tedious task – especially when acquiring the entries automatically from mixed online sources. However, it is important that informations are complete and presented in a consistent style to avoid confusions. JSS requires the following format.

- JSS-specific markup (`\proglang`, `\pkg`, `\code`) should be used in the references.
- Titles should be in title case.
- Journal titles should not be abbreviated and in title case.
- DOIs should be included where available.
- Software should be properly cited as well. For **R** packages `citation("pkgname")` typically provides a good starting point.

Affiliation:

Thomas Huet
CNRS-UMR 5140
Archeologie des Societes Mediterraneennes
Universite Paul Valery
route de Mende
Montpellier 34199, France
E-mail: thomashuet7@gmail.com

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd