The Journal of Open Source Software

# bleiglas: An R package for interpolation and visualisation of spatiotemporal data with 3D tessellation

**Clemens Schmid**[1] **and Stephan Schiffels**[1]

**1** Department of Archaeogenetics, Max Planck Institute for the Science of Human History, Kahlaische Strasse 10, 07745 Jena, Germany

**DOI:** 10.21105/joss.03092

**Software**

- Review
- Repository
- Archive

**Editor:** Vissarion Fisikopoulos

**Reviewers:**

- @corybrunson
- @fabian-s

**Submitted:** 12 February 2021
**Published:** 09 April 2021



## Background

The open source software library Voro++ (Rycroft, 2009) allows fast calculation of Voronoi diagrams in three dimensions. Voronoi diagrams are a special form of tessellation (i.e., filling space with geometric shapes without gaps or overlaps) where each polygon is defined as the area closest to one particular seed point. Imagine a volume in three dimensional space and an arbitrary distribution of unique points within this volume. Voro++ creates a polygon around each point so that everything within this polygon is closest to the corresponding point and farther away from every other point.

Voronoi tessellation has useful applications in all kinds of scientific contexts spanning astronomy (e.g., Paranjape & Alam (2020)), material science (e.g., Tsuru et al. (2020)), and geography (e.g., Liu et al. (2019)). In computational and landscape archaeology, Delaunay triangulation and Voronoi diagrams have also been applied (Nakoinz & Knitter, 2016), but to our knowledge, mostly limited to an entirely spatial 2D perspective. 3D tessellation could be employed here to add a third dimension, most intriguingly a temporal one. This could allow for new methods of spatiotemporal data interpolation, analysis, and visualisation.

## Statement of need

The `bleiglas` R package serves as an R interface to the Voro++ command line tool. It adds a number of utility functions for particular data manipulation applications, including but not limited to automatic 2D cutting of the 3D Voro++ output for subsequent mapping and grid sampling for position and value uncertainty mitigation. The relevant workflows are explained below. Although we wrote this package for our own needs in archaeology and archaeogenetics, the code is by no means restricted to data from these fields, just as Voronoi tessellation is a generic, subject-agnostic method with a huge range of use cases.

Voronoi tessellation is implemented in many R packages, perhaps most prominently in the deldir (*Delaunay Triangulation and Dirichlet (Voronoi) Tessellation*) (Turner, 2021) and tripack (*Triangulation of Irregularly Spaced Data*) (Gebhardt et al., 2020) packages, which were specifically designed for this application. ggvoronoi (*Voronoi Diagrams and Heatmaps with 'ggplot2'*) (Garrett et al., 2018) and `dismo::voronoi()` (Hijmans et al., 2020) build on deldir. Other implementations such as `sf::st_voronoi()` (Pebesma, 2018) and `geos::geos_voronoi_polygons()` (Dunnington & Pebesma, 2021) rely on the GEOS library (GEOS Development Team, 2021). All of these packages and functions focus on 2D data, and to our knowledge, none offer a toolset to handle 3D Voronoi diagrams comparable to that introduced with `bleiglas`.

## Core functionality

`bleiglas` provides the `bleiglas::tessellate()` function, which is a command line utility wrapper for Voro++. It requires Voro++ to be installed locally. `tessellate()` takes input points in the form of a `data.frame` with an integer ID and three numeric coordinate columns. Additional Voro++ options can be set with a character argument `options` and only the output format definition (`-c`) is lifted to an extra character argument `output_definition`. `tessellate()` returns a character vector containing the raw output of Voro++ with one vector element corresponding to one row. Depending on the structure of this raw output, different parsing functions are required to transform it to a useful R object. At the moment, `bleiglas` provides one such function: `bleiglas::read_polygon_edges()`. It is configured to read data produced with the Voro++ output format string `%i*%P*%t`, which returns polygon edge coordinates. These are necessary for the default `bleiglas` workflow illustrated in the example below. Future versions of the package may include other parsing functions for different pipelines.

The output of `read_polygon_edges()` is (for performance reasons) a `data.table` (Dowle & Srinivasan, 2019) object that can be used with `bleiglas::cut_polygons()`. This function now shoulders the core task of cutting the 3D, polgyon-filled Voro++ output box into 2D slices. 3D data is notoriously difficult to plot and read. Extracting and visualizing slices is therefore indispensable. The necessary algorithm for each 3D polygon can be summarised as finding the cutting point of each polygon edge line with the requested cutting surface and then defining the convex hull of the cutting points as a result 2D polygon. We implemented the line-segment-plane-intersection operation via Rcpp (Eddelbuettel & Balamuta, 2017) in C++ for better performance and used `grDevices::chull()` for the convex hull search. The output of `cut_polygons()` is a list (for each cut surface) of lists (for each polygon) of `data.tables` (3D coordinates for each 2D cutting point). Optionally, and in case of horizontal (z-axis) cuts with spatial coordinates on the x- and y-axis, this output can be transformed to an `sf` (Pebesma, 2018) object via `bleiglas::cut_polygons_to_sf()`. This significantly simplifies subsequent map plotting.

The final core function of `bleiglas` is `bleiglas::predict_grid()`, which paves the way for more complex applications and data subject to a higher degree of positional uncertainty. It employs the tessellation output to predict values at arbitrary positions by determining in which 3D polygons they are located. Therefore `bleiglas::predict_grid()` should theoretically mimic the outcome of a nearest neighbor search, but is fully implemented with the above mentioned tessellation workflow. The core algorithm `bleiglas::attribute_grid_points_to_polygons()` uses `cut_polygons()` to cut the tessellation volume at the z-axis level for each prediction point. It then checks in which 2D polygon the point is located to attribute it accordingly. This is done with custom C++ code initially developed for our recexcavAAR package (Schmid & Serbe, 2017). `bleiglas::predict_grid()` can be used to automatically rerun tessellation multiple times for data with uncertain position in one or multiple of the three dimensions. In this case the resulting wiggling of input points and therefore output polygons is recorded with the static prediction grid. Different per-prediction-point observations in the grid can eventually be summarised to calculate mean outcomes and deviation. One concrete archaeological application of this feature is temporal resampling from post-calibration radiocarbon age probability distributions.

A prerequisite for performing tessellation in three dimensions is the normalisation or mapping of length units across three dimensions. If all three dimensions have the same units (as is the case for 3D spatial data), this is not an issue, and tessellation works as expected. However, if dimensions have different units, the outcome and meaning of the tessellation depends crucially on how these units are mapped to each other. This is the case for spatiotemporal data, in which one axis denotes time and the other two axes denote a 2D spatial position. In such cases it is critical to use external information to inform on an appropriate scaling. For example, one might set 1 km to correspond to 1 year, in which case two contemporaneous points 100

km apart are considered "as close as" two points 100 years apart. What scaling to use clearly depends on the dataset and how to query it.

## Example: Burial rite distributions in Bronze Age Europe

One strength of `bleiglas` is visualisation of spatiotemporal data. Here we show an example of Bronze Age burial rites as measured on radiocarbon dates from burials in Central, Northern, and Northwestern Europe between 2200 and 800 calBC. Information about source data (taken from the RADON-B database (Kneisel et al., 2013)), data preparation, and meaning are presented in Schmid (2019). A vignette in `bleiglas` (`vignette("bleiglas_case_study")`) contains the complete code to reproduce the following figures.

Bronze Age burials can be classified by two main aspects: inhumation vs. cremation (*burial type*) and flat grave vs. burial mound (*burial construction*). Figure 1 is a map of burials through time for which we have some information about these variables. Each grave has a position in space (coordinates) and in time (median calibrated radiocarbon age). For Figure 2 and Figure 3, we only look at the *burial type* aspect. The burials are distributed in a three dimensional, spatiotemporal space and therefore can be subjected to Voronoi tessellation with Voro++. As detailed above, the outcome depends on the relative scaling of the input dimensions - for this example we choose $1\text{kilometer} = 1\text{year}$, informed by some intuition about the range of human movement through time.

For Figure 3 we cut these polygons into 2D time slices that can be visualized in a map matrix (*bleiglas plot*). We believe this matrix is a visually appealing and highly informative way to communicate processes derived from 3D point patterns. It conveys both the main trends (here, the general switch from inhumation to cremation from the Middle Bronze Age onwards) as well as how much data is available in certain areas and periods. The latter is especially relevant regarding the derived question which resolution could be expected from a model based on this input data.

For the final Figure 4, we applied temporal resampling with `bleiglas::predict_grid()` to record observational error caused by radiocarbon dating uncertainty. For reasons of computational performance we kept the number of resampling runs and the spatial resolution in this example low. Nevertheless the advantages of the resampling approach are easily seen: areas and periods with uncertain attribution to one burial type or the other are indicated as such.

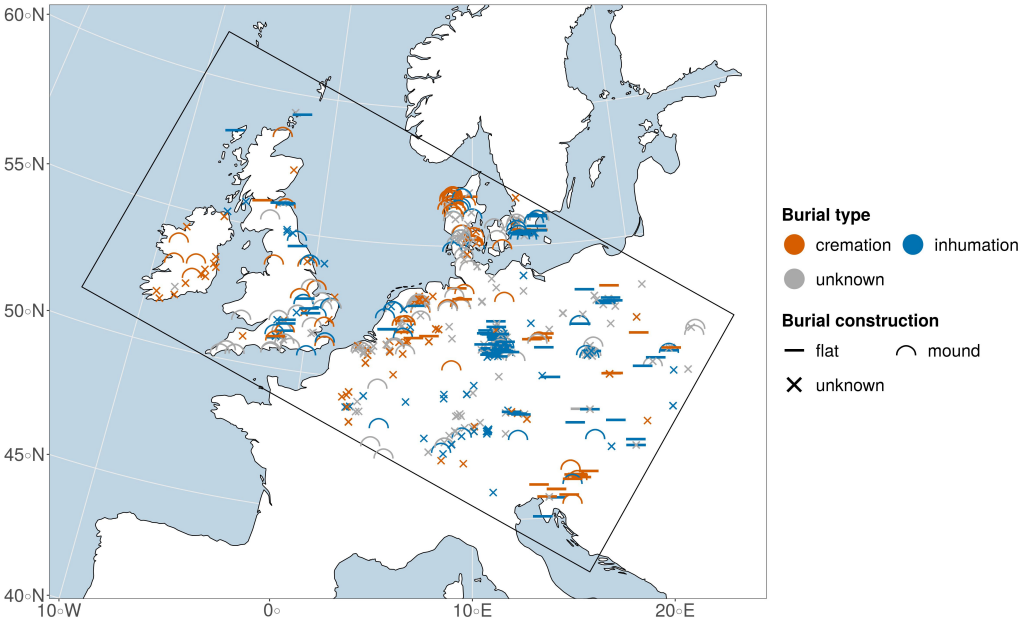


**Figure 1:** Graves in the research area (rectangular frame) dating between 2200 and 800 calBC as extracted from the Radon-B database. The classes of the variable burial type are distinguished by colour; the ones of burial construction by shape. The map projection is EPSG:102013 (Europe Albers Equal Area Conic) and the base layer data is taken from the Natural Earth project.

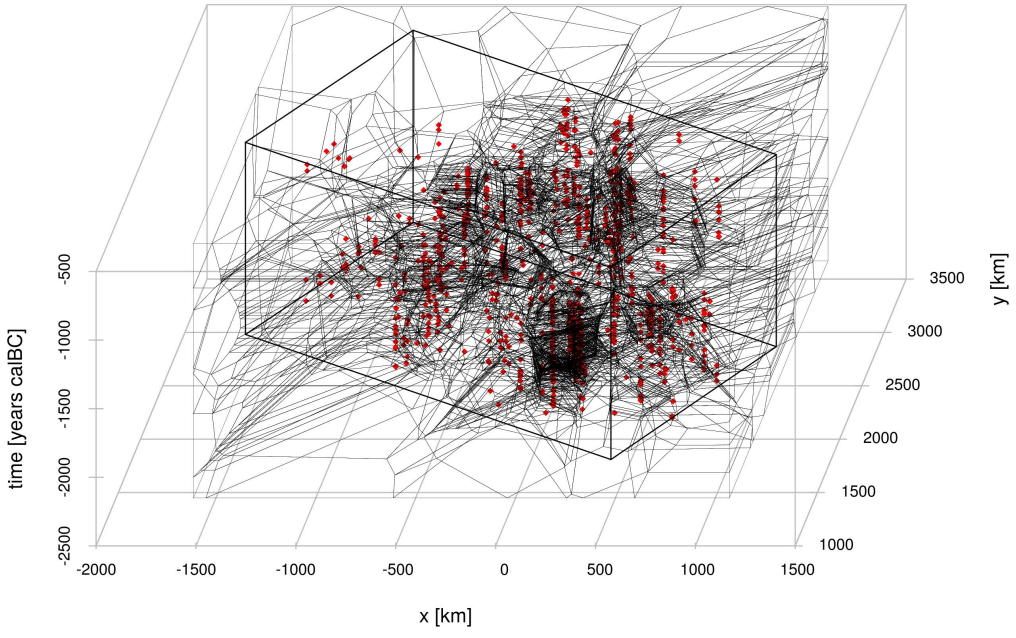


**Figure 2:** Graves in 3D space defined by two spatial (x and y in km) and one temporal (z in years calBC) dimensions with Voronoi polygons constructed by Voro++. Each red dot represents one grave with known burial type, the fine black lines the edges of the result polygons, and the rectangular wireframe box the research area now in space and time.

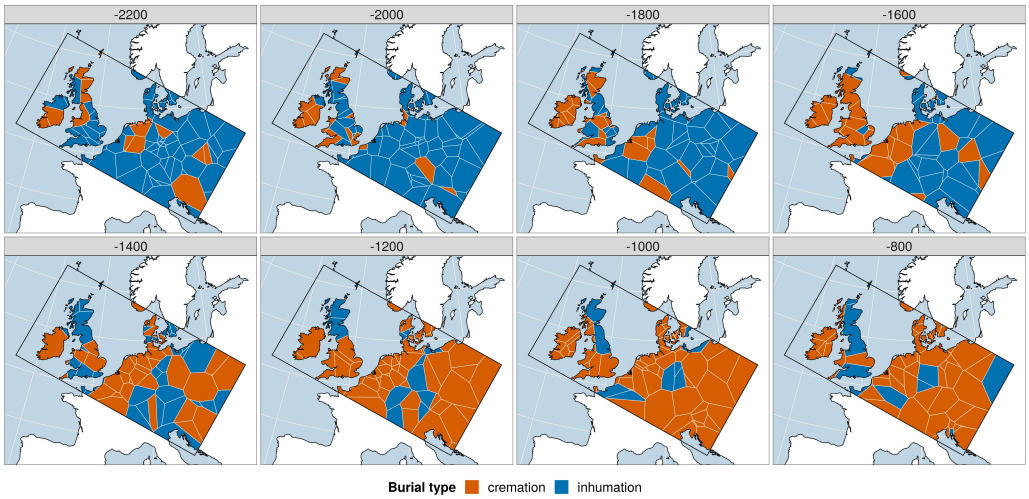


**Figure 3:** *bleiglas plot*. Map matrix of 2D cuts through 3D Voronoi polygons as presented in Figure 2. Each subplot shows one timeslice between 2200 and 800 calBC. As each 2D polygon belongs to one input grave and data density in some areas and time periods is very low, some graves are represented in multiple subplots. Color coding and map background is as in Figure 1.

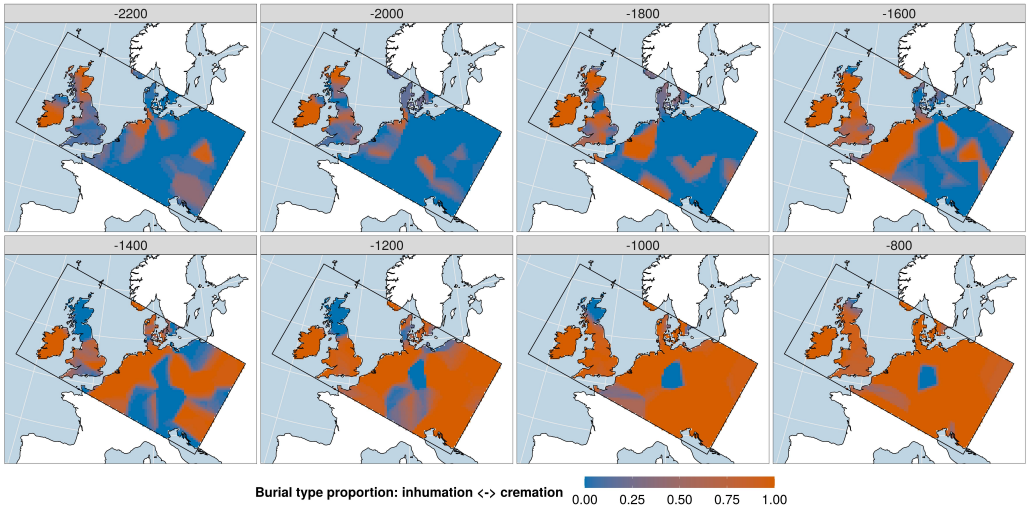


**Figure 4:** Temporal resampling version of the bleiglas map matrix. 30 resampling runs, and a spatial resolution of 100*100 cells in the research area shape bounding box. Color coding and map background again as in Figure 1.

## Acknowledgements

The package benefitted from valuable comments by Joscha Gretzinger, who also suggested the name *bleiglas* (German *Bleiglasfenster* for English *Leadlight*) inspired by the appearance of the cut surface plots.

## References


Dowle, M., & Srinivasan, A. (2019). *data.table: Extension of data.frame*. https://CRAN.R-project.org/package=data.table

Dunnington, D., & Pebesma, E. (2021). *geos: Open source geometry engine ('GEOS') R API*. https://CRAN.R-project.org/package=geos

Eddelbuettel, D., & Balamuta, J. J. (2017). Extending R with C++: A brief introduction to Rcpp. *PeerJ Preprints*, *5*, e3188v1. https://doi.org/10.7287/peerj.preprints.3188v1

Garrett, R. C., Nar, A., Fisher, T. J., & Maurer, K. (2018). ggvoronoi: Voronoi diagrams and heatmaps with ggplot2. *Journal of Open Source Software*, *3*(32), 1096. https://doi.org/10.21105/joss.01096

Gebhardt, A., Renka, R. J., Eglen, S., Zuyev, S., & White, D. (2020). *tripack: Triangulation of irregularly spaced data*. https://CRAN.R-project.org/package=tripack

GEOS Development Team. (2021). *GEOS - geometry engine, open source*. https://trac.osgeo.org/geos.

Hijmans, R. J., Phillips, S., Leathwick, J., & Elith, J. (2020). *dismo: Species distribution modeling*. https://CRAN.R-project.org/package=dismo

Kneisel, J., Hinz, M., & Rinne, C. (2013). *Radon-B*. http://radon-b.ufg.uni-kiel.de.

Liu, G., Yao, X., Luo, Z., Kang, S., Long, W., Fan, Q., & Gao, P. (2019). Agglomeration centrality to examine spatial scaling law in cities. *Computers, Environment and Urban Systems*, *77*, 101357. https://doi.org/10.1016/j.compenvurbsys.2019.101357

Nakoinz, O., & Knitter, D. (2016). *Modelling human behaviour in landscapes: Basic concepts and modelling elements*. Springer International Publishing. https://doi.org/10.1007/978-3-319-29538-1

Paranjape, A., & Alam, S. (2020). Voronoi volume function: A new probe of cosmology and galaxy evolution. *Monthly Notices of the Royal Astronomical Society*, *495*(3), 3233–3251. https://doi.org/10.1093/mnras/staa1379

Pebesma, E. (2018). Simple Features for R: Standardized Support for Spatial Vector Data. *The R Journal*, *10*(1), 439–446. https://doi.org/10.32614/RJ-2018-009

Rycroft, C. H. (2009). Voro++: A three-dimensional Voronoi cell library in C++. *Chaos*, *19*(4), 041111. https://doi.org/10.1063/1.3215722

Schmid, C. (2019). Evaluating cultural transmission in bronze age burial rites of Central, Northern and Northwestern Europe using radiocarbon data. *Adaptive Behavior*, 1059712319860842. https://doi.org/10.1177/1059712319860842

Schmid, C., & Serbe, B. (2017). *recexcavAAR: 3D reconstruction of archaeological excavations*. https://CRAN.R-project.org/package=recexcavAAR

Tsuru, T., Shimizu, K., Yamaguchi, M., Itakura, M., Ebihara, K., Bendo, A., Matsuda, K., & Toda, H. (2020). Hydrogen-accelerated spontaneous microcracking in high-strength aluminium alloys. *Scientific Reports*, *10*(1), 1998. https://doi.org/10.1038/s41598-020-58834-6

Turner, R. (2021). *Deldir: Delaunay triangulation and Dirichlet (Voronoi) tessellation*. https://CRAN.R-project.org/package=deldir