dialectR: Doing Dialectometry in R

Ryan Soh-Eun Shim[*]
Institute for Natural Language Processing
University of Stuttgart
soh-eun.shim@ims.uni-stuttgart.de

John Nerbonne
Linguistics
Groningen, Freiburg & Tübingen
j.nerbonne@rug.nl

Abstract

We present dialectR, an open-source R package for performing quantitative analyses of dialects based on categorical measures of difference and on variants of edit distance. dialectR stands as one of the first programmable toolkits that may freely be combined and extended by users with further statistical procedures. We describe implementational details of the package, and provide two examples of its use: one performing analyses based on multidimensional scaling and hierarchical clustering on a dataset of Dutch dialects, and another showing how an approximation of the acoustic vowel space may be achieved by performing an MFCC (Mel-Frequency Cepstral Coefficients)-based acoustic distance on audio recordings of vowels.

1 Introduction

The quantitative analysis of dialect relatedness has yielded respectable results in the field of dialectometry, where sophisticated methods of measuring linguistic distance have been developed which correlate to a large degree with perceptual measurements of intelligibility (Gooskens and Heeringa, 2004; Beijering et al., 2008). The use of such methods offers an objective basis to the determination of dialect distributions, including boundaries at times, and which overcomes some of the subjective biases inherent in earlier approaches that utilized the notion of isogloss for dialect classification.

However, despite the success of these methods, access to their use has generally relied on GUI-based software such as Visual DialectoMetry (VDM) (Goebl, 2006), DiaTech (Aurrekoetxea et al., 2013), and Gabmap (Nerbonne et al., 2011; Leinonen et al., 2016), which are easy to use, but which accept the trade-off of impeding easy modification for those who wish to extend existing methods. Users who wish to perform statistical analyses outside of what is provided or make changes to the existing pipeline do not have easy access to the internals of such software, and consequently have to start from a higher technical threshold. In fact, these packages have not been modified by others. A notable exception is the L04 software,[1] which operates in the UNIX ecosystem and would allow for some degree of user modification, but few, if any users have taken advantage of this. In addition to providing for more flexibility that exists in current packages, the present effort also facilitates the work of those who like the provisions of the older packages, but who wish to try out contemporary approaches, something the existing packages likewise do not readily support.

In view of this situation, we present dialectR, an open-source software package that allows the construction of dialectometric pipelines in the statistical programming language R (R Core Team, 2020). It is largely inspired by Gabmap, but attempts to overcome some shortcomings of its monolithic presentation. Our vision is to facilitate more wide-ranging dialectological experimentation with the data analysis possibilities in R. For example, dialectologists should be able to experiment more directly with geostatistical analyses, which, with honorable exceptions (Grieve, 2018), have largely been ignored in dialectometry. For a second example, we note that, although dialectometry makes extensive use of multi-dimensional scaling (see below), other dimension-reducing techniques (for non-distance matrices), such as factor analysis or principal component analysis, have received less attention, again with some honorable exceptions (Pickl, 2013; Nerbonne, 2015). The present paper offers a foundation from which much more extensive experimentation may be launched. We offer

[*] The project began in a course taught by JN, where RS-ES suggested the idea of an R-package. The authors went back and forth on design decisions, but RS-ES implemented the software, and wrote most of the first version of the paper. JN wrote some subsections and collaborated on the others.

[1] http://www.let.rug.nl/kleiweg/L04/

Proceedings of the Ninth Workshop on NLP for Similar Languages, Varieties and Dialects, pages 20–27
October 16, 2022. ©2022 Association for Computational Linguistics

further examples below.

2 Software Design

The component of dialectR which most interests users is probably the edit distance computation of pronunciation differences based on transcriptions, which is written in C++11 and interfaced to R through the R package Rcpp (Eddelbuettel and François, 2011). Once a distance matrix between data-collection sites has been produced, dialectR additionally offers a number of ready-made functions for common analyses, such as an RGB-based multidimensionsal scaling for visualizing dialect continuua (Nerbonne et al., 1999), or a function for visualizing discrete dialect groupings based on hierarchical clustering (Nerbonne et al., 2008). Moreover, in the case where the input data is acoustic, we show how an additional acoustic distance proposed in Bartelds et al. (2020) can be leveraged, something missing in all alternative packages and web applications. We describe specifics of these components in the following subsections.

2.1 Distance Computation

Methods in dialectometry have revolved around aggregating linguistic differences between data collection sites since the inception of the field, in large part to overcome the noisy geographic distribution of sites and sample material (Goebl, 2018). A pioneering attempt in this direction can be seen in Séguy (1971), who worked with questionnaire data, where the number of possible answers are relatively limited (e.g. “what do you call a serving-size, unsweetened pastry?”). Séguy’s method is essentially to count the number of different responses to the same survey questions at two dialect sites, and his paper marked the first important breakthrough in the establishment of the subfield.

To give a practical example of how such categorical could be used to quantify linguistic differences, suppose we have lexical data for two related dialect sites as shown in Table 1. To quantify how different these two sites are, a difference of 1 can be counted for every mismatch between vocabulary items, ignoring the pairs where data is unavailable. The total count is then normalized by taking the mean, resulting in a lexical distance of 0.25, meaning that there is a 75% lexical similarity between the two sites (Nerbonne and Kleiweg, 2003).

Site	Vocabulary Items				
Brownsville	*dog*	*hat*	*horse*	*bathroom*	*pinkie*
White Plain	*dog*	*cap*	*horse*	*bathroom*	-

Table 1: Sample data as taken from LAMSAS for the illustration of lexical distance.

Such an approach provides a simple notion of lexical distance that can be used to aggregate over items, but a number of issues remain. For one, it would be desirable for morphologically related words to carry a smaller distance than words that are completely unrelated. Thus if in response to the question “if the sun comes out after a rain, you say the weather is doing what?", elicitations such as *fair off*, *fairs off*, and *faired off* come up, these variants of the same lexical item should count as less distant when compared with terms such as *clearing up* and *breaking away* (Nerbonne and Kleiweg, 2003). Similarly, it would also be insightful for there to be a metric that can quantify the degree of difference between phonetic transcriptions of related dialects. The solution to both issues may be found in edit distance, which forms the basis for methods developed in in the 1990s in Groningen.

Edit distance was first applied to dialect data in Kessler (1995), where it was applied on phonetic transcriptions of Irish Gaelic dialects and assigned to groups with hierarchical clustering, which proved to yield sensible results that correlate well with provincial boundaries. This in turn inspired further work at the University of Groningen that refines upon various aspects of the edit distance algorithm and the clustering algorithms (Nerbonne et al., 2008; Wieling et al., 2012), among other procedures. The original edit distance algorithm is a measure of distance between two strings, where the distance is derived from how many insertions, deletions, and substitutions it would take for one string to transform into the other. As an example, consider how in the table below, the string “koguma", the word for sweet potato in Korean, may be transformed into “kokoimo", a possible origin of the Korean term from the Tsushima dialect in Japan, with one insertion followed by three substitutions:

koguma	insert k	1
kokguma	replace g/o	1
kokouma	replace u/i	1
kokoima	replace a/o	1
kokoimo	Sum distance	4

However, in comparing two sequences with edit distance, longer sequences possess a much higher chance of containing more differences than shorter sequences. If used directly, this would bias the results by causing varieties with longer sequences to appear more different. Thus for a fair comparison of string distance across multiple samples, we follow Heeringa et al. (2006) by providing the option to normalize the distance by dividing the length of the alignment between the two strings. We furthermore also provide the option to use a variant of edit distance that forbids the alignment of vowels and consonants, which results in more plausible alignments, and thus also results in an improvement in the computed distance.

Moreover, due to the possibility of informants giving multiple responses in a single site, we provide the option to normalize for multiple responses with Bilbao distance (Aurrekoetxea et al., 2020), which is as follows:

$$D_B(A,B) = \frac{\sum_{i=1}^{|A|} \min_{b_j \in |B|} d(a_i b_j) + \sum_{j=1}^{|B|} \min_{a_j \in |A|} d(a_i b_j)}{|A| + |B|}$$

Where, in plain words, for every element in a given set A, we compute its minimal distance to all the elements of set B, using only that in the sum, and where we proceed the same way with respect to set B, seeking for each b in B, the closest element in A. The mean of the distances is then taken for normalization. We illustrate this with an example: suppose we have elicited responses to the question “what do you call the place where people are buried?" [2] from two sites, A and B. Site A has obtained the responses of {*graveyard*, *boneyard*}, and Site B has obtained the responses of {*cemetery*, *kirkyard*, *graveyard*}. Using a length-normalized edit distance as metric, the distance for every response in Site A as compared against the responses in Site B is shown in Table 2. We choose the combination for each response that minimizes the distance, add them up, and divide the sum by the total number of elements, which yields:

$$D_B(A,B) = \frac{0 + 0.44 + 0.75 + 0.5 + 0}{2+3} = 0.338$$

Finally, after the above computations have been applied to all pairs of words between sites, we discount the pairs where there is no data and take the average. This results in a distance matrix of normalized dialect distances, which is amenable to further statistical treatment.

B A	*cemetery*	*kirkyard*	*graveyard*
graveyard	0.78	0.56	0
boneyard	0.75	0.5	0.44

Table 2: Example data for illustration of Bilbao distance, where the cells indicate the length-normalized edit distance between responses.

2.2 Visualization

dialectR provides two visualization methods common in dialectometry: one based on multidimensional scaling, and another based on hierarchical clustering. We discuss their implementation in dialectR below.

2.2.1 Multidimensional Scaling

Multidimensional scaling refers to a family of dimensionality reduction techniques, where complex data is reduced to a smaller number of dimensions that can be more easily interpreted. Multidimensional scaling has been applied to distance tables extensively in dialectometry for the purpose of showing dialect continuum phenomena (Nerbonne et al., 1999; Embleton et al., 2013), and usually provides more robust results than those of clustering. The `mds_map` function in dialectR uses a refinement of Torgerson’s multidimensional scaling (Torgerson, 1952), where provided a matrix of dissimilarities, the algorithm projects each data point into a lower dimensional space with the goal of preserving the distance between them as best possible.

The distance matrix of edit distance between varieties as described in section 2.1 can therefore in the aforementioned manner be given as input; reduced to three dimensions, where each dimension is rescaled to a range of [0, 1] with min-max scaling, and transformed proportionately to RGB values respectively. The three colors are then mixed, and at last projected onto the geographic locations of each variety. Figure 4 shows the results of applying this method on Dutch dialect data provided in the Goeman-Taeldeman-Van Reenen-project (Taeldeman and Goeman, 1996).

2.2.2 Hierarchical Clustering

Complementing the possibility of showing dialect continuua, in dialectology it is often also desirable to pursue a notion of distinct dialect groups.

[2] Question and responses sampled from Linguistic Atlas Project, item number 78.8.

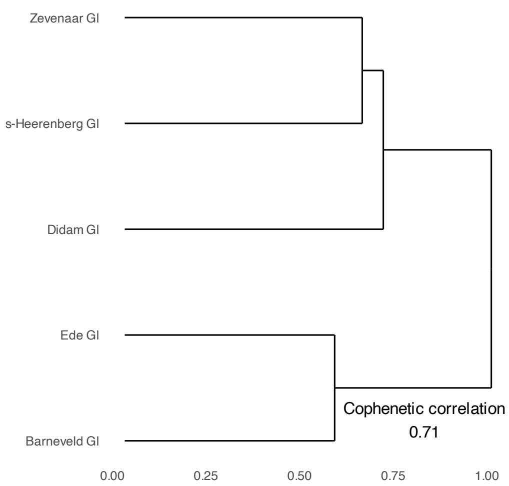

Figure 1: A partial dendrogram of the Goeman-Taeldeman-Van Reenen-project.

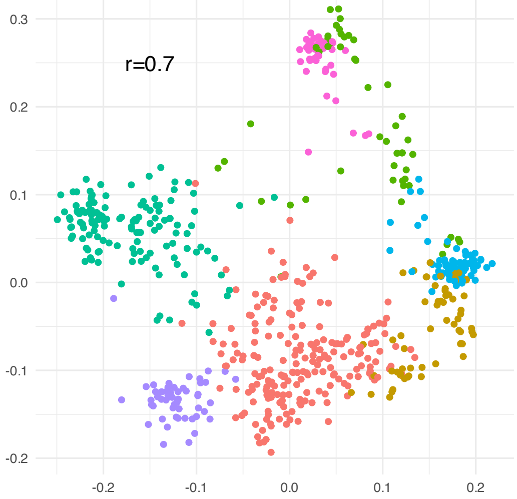

Figure 2: Dutch data from the Goeman-Taeldeman-Van Reenen-project reduced to two dimensions with multidimensional scaling, where the colors are groupings as obtained by hierarchical clustering.

This is achieved in dialectometry through hierarchical clustering, which dialectR implements by building upon the `hclust` function built natively in R. This allows for a number of agglomeration methods to be specified, including the weighted average method (alternatively known as the WPGMA method) and Ward's method, which differ in how proximity between clusters is defined, and can lead to somewhat different results. The result of applying hierarchical clustering on our distance matrix is a dendrogram, an example of which is shown in Figure 1, where the cophenetic distance between nodes can be seen. A cophenetic correlation coefficient between the original distance matrix and the results of clustering can also be calculated, which indicates how well the dendrogram has preserved the original distances in the data, and comes down to 0.71 for the Dutch dialect dataset using Ward's method.

However, due to the instability of hierarchical clustering, steps of validation and bootstrapping may be necessary to confirm the validity of the clusters. One possible method of validation is to plot the cluster groupings against the results of the a multidimensional scaling. This would result in Figure 2, where the difference in spread of the seven clusters would point to the possibility that certain edge cases remain ambiguous between clusters due to the continuous nature of the dialect data. The implementation of further bootstrapping and validation procedures such as described in Nerbonne et al. (2008) is also possible with the help of numerous related packages such as Suzuki and Shimodaira (2006) and Hennig (2020), the ready availability of which is a strength of dialectR over comparable closed systems.

2.3 Acoustic Distance

As an example of the benefit of the framework presented here, we turn to an open-source implementation of recent work that is not yet available in other comparable closed systems such as Gabmap and DiaTech. In order to demonstrate the advantage of an open system, we re-implemented the acoustic distance in Bartelds et al. (2020) in Python,[3] and include it here in R through the reticulate package (Ushey et al., 2020).

The method transforms audio samples into numerical feature representations based on 39-dimensional Mel-frequency cepstral coefficients (MFCCs), which include the first 12 cepstral coefficients and energy in each frame; the first and second derivatives from each of the cepstral coefficients and energy features; and one first and one second derivative related to the energy feature. These coefficients are computed with a window size of 25 ms and a stride of 10 ms. Cepstral means and variance normalization are used to reduce the effect of noise. After obtaining MFCCs for the two audio samples under consideration, dynamic time warping is then performed upon them to derive a measure of their distance. Bartelds et al. apply

[3] `https://github.com/b05102139/acoustic_distance`

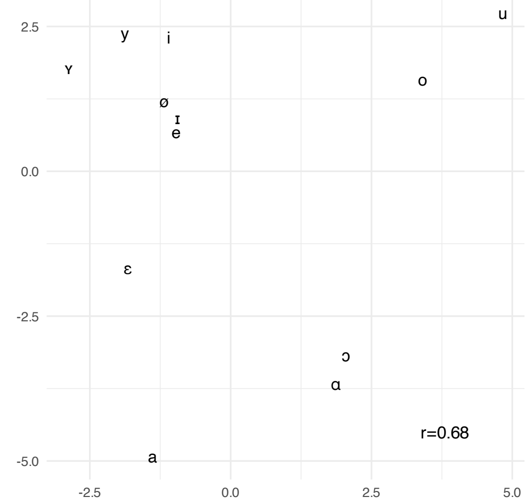

Figure 3: Acoustic vowel space as approximated with acoustic distance.

this method to audio samples in the Speech Accent Archive (Weinberger and Kunath, 2011), where a correlation of $r = -0.71 (p < 0.0001)$ was found between human judgments of native-likeness and the distance derived from their method. An approximate acoustic vowel space was also derived by applying their method to vowels, which we replicate in Figure 3 by using the recordings of vowels in the international phonetic alphabet as recorded by Peter Ladefoged[4] and plotting the two first dimensions of a multidimensional scaling.

This method enables the reduction of time and effort needed for transcription-based methods, where the human resources needed to transcribe the dialect audio into IPA may not be available. The implementation of this method relies heavily on speech processing packages in the Python ecosystem, and serves to illustrate the broader potential of doing dialectometry with open-source software, where the ability to utilize external resources in Python through the reticulate package constitutes a further advantage (Ushey et al., 2020).

Concepts / Sites	*aarde*	*adem*	*appels*
AalsmeerNH	ʔɒrde	ʔɒdəm	ʔapəls
AalstBeLb	ɛət	osəm	ɑpəls
AalstBeOv	eɛrdə	osəm	ɑpələn

Table 3: Excerpt of the transcriptions in the Goeman-Taeldeman-Van Reenen-project, where the cells are phonetic transcriptions of concepts collected at multiple sites.

3 Example Session

We show in this section an example session, by analyzing Dutch dialect data in the Goeman-Taeldeman-Van Reenen-project with dialectR. The IPA transcription dataset comes installed with the package, along with a sample Keyhole Markup Language (KML) file that is required in order to provide geographic data of the collection sites. The Keyhole Markup Language is an XML-based markup language for geographic data, and is principally associated with Google Earth,[5] which users may utilize to create KML files for their own data. An excerpt of the phonetic transcriptions is shown in Table 3. An excerpt of the KML file is shown below:

```
<Placemark>
    <name>Zwolle Ov</name>
    <Point>
    <extrude>1</extrude>
    <coordinates>6.10418,52.5146,0</coordinates>
    </Point>
</Placemark>
```

The transcription data can be called with the `data` function built natively in R, and the geographic data can be loaded with `get_points` and `get_polygons`, which respectively extract the points and polygon data from the KML file into dataframes:

```
library(dialectR)
data(Dutch)
pathToKML <- system.file("extdata",
                         "DutchKML.kml",
                          package="dialectR")
dutchPoints <- get_points(pathToKML)
dutchPolygons <- get_polygons(pathToKML)
```

With the transcription data and geographic information ready, we can call `distance_matrix` and set the option of `alignment_normalization` to true, which computes the edit distance between the pronunciations of all corresponding words in all pairs and normalizes the score by length; we also set `funname` to `leven`, which uses the plain edit distance for its computation, as opposed to `vc_leven`, which implements the vowel-consonant constraint. The details of both of these options are discussed in Section 2.1.

[4] http://www.phonetics.ucla.edu/course/chapter1/vowels.html

[5] https://earth.google.com/web/

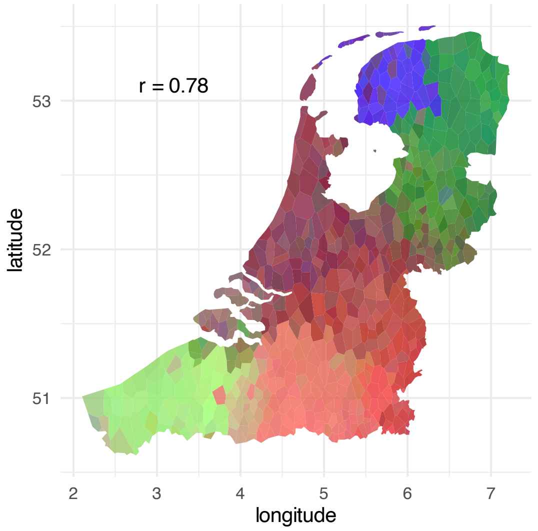

Figure 4: Three dimensions of the multidimensional scaling plotted respectively as RGB values, mixed together, and projected onto their respective locations.

We call `mds_map` upon the resulting distance matrix along with the required geographic information, which results in Figure 4:

```
distDutch <- distance_matrix(Dutch,
                funname="leven",
                alignment_normalization=TRUE)
mds_map(distDutch, dutchPoints, dutchPolygons)
```

We briefly remark that Friesland (the area in blue) clearly stands out as a variety most distinctly separate from its surroundings, which is consistent with its status as an independent language. The low Saxon area (the green area on the top right) and the west of Flanders (lower left) also show a notable similarity, which Wieling and Nerbonne (2011) also noted.

For purposes of illustration, we also show here how the edit distance and its variants as implemented in `distance_matrix` can be called independently of the function:

```
leven("graveyard/boneyard",
       "cemetery/kirkyard/graveyard",
                   alignment_normalization = T,
                   delim = "/")
```

Where the `alignment_normalization` parameter normalizes the distance by dividing the length of the alignment between two strings, and the `delim` parameter allows for comparing multiple responses in one or both of the sites with Bilbao distance.

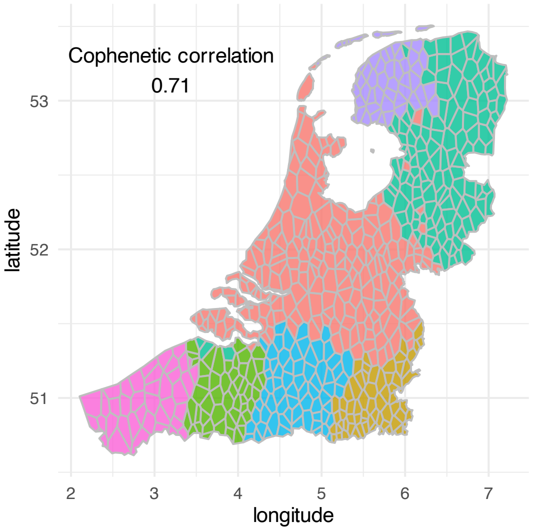

Figure 5: The groupings of hierarchical clustering as projected onto their respective locations.

To gain more specific insights into how one might classify significant similarities in a given area, we are now in a place to complement the multidimensional scaling analysis as performed above with hierarchical clustering. In dialectR this can be called via `cluster_map`, which results in Figure 5:

```
cluster_map(distDutch,
              kml_points = dutchPoints,
              kml_polygon = dutchPolygons,
              cluster_num = 7,
              method = "ward.D2")
```

We observe that the projection of our hierarchical clusters onto the geographic locations of the collection sites results in sensible aggregate isoglosses that largely correspond with the classification of dialectologists.

4 Conclusion and Future Work

We presented dialectR, an open-source package that attempts to facilitate community-based extensions to dialectometric methods by situating itself in the statistical environment of R. In doing so, we echo the sentiment in Nerbonne et al. (2011) regarding the future of Gabmap, a web application for dialectometry that served as the primary reference for the present package: “[t]here are also opportunities for further development. Probably the most important of these would involve making it easier for others to contribute modules, i.e. adopting an open-source development mode. Once it becomes easier for others to contribute, then scientific imagination is the limiting factor”.

We suggested several lines of research above which dialectR might be used to support, including the use of geostatistical analysis or a wider range of dimension-reducing techniques. We further demonstrated how dialectR could be used to incorporate acoustics-based aggregate analyses in Sec. 2.3 above. So it is fitting that we close with yet another suggestion for work that dialectR might be used to support.

Edit distance measures for phonetic transcriptions have been shown to improve in sensitivity when used with sensitive segment weights (Wieling et al., 2012). Work in this direction has sought to take into account that frequent sound substitutions should be taken as more similar than infrequent ones (e.g., a substitution of [ɛ] should count as more similar to [e] than to [o]). Such a procedure has been used for the measurement of foreign accent strength (Wieling et al., 2014) and for the rectification of "field worker isoglosses", which refers to a systematic difference in transcription that occurs due to the field workers preferences, as opposed to any real linguistic differences between the dialect sites (Wieling and Nerbonne, 2011). These applications together point towards its usefulness as a future module, either to be incorporated into the current package, or, alternatively, to be made available alongside it.

As increasingly sophisticated statistical methods come to be used to examine dialect data (Wieling and Nerbonne, 2015; Wieling et al., 2018), the possibility of interfacing with dedicated packages in R facilitates the community-based effort to keep the latest methods within the reach of the general user.

Acknowledgements

We would like to thank the anonymous reviewers for providing valuable feedback in improving this work.

References

Gotzon Aurrekoetxea, Karmele Fernandez-Aguirre, Jesus Rubio, Borja Ruiz, and Jon Sanchez. 2013. 'DiaTech': A new tool for dialectology. *Literary and Linguistic Computing*, 28(1):23–30.

Gotzon Aurrekoetxea, John Nerbonne, and Jesus Rubio. 2020. Unifying analyses of multiple responses. *Dialectologia*, 25:59–86.

Martijn Bartelds, Caitlin Richter, Mark Liberman, and Martijn Wieling. 2020. A new acoustic-based pronunciation distance measure. *Frontiers in Artificial Intelligence*, 3.

Karin Beijering, Charlotte Gooskens, and Wilbert Heeringa. 2008. Predicting intelligibility and perceived linguistic distance by means of the Levenshtein algorithm. *Linguistics in the Netherlands*, 25:13–24.

Dirk Eddelbuettel and Romain François. 2011. Rcpp: Seamless R and C++ integration. *Journal of Statistical Software*, 40(8):1–18.

Sheila Embleton, Dorin Uritescu, and Eric S. Wheeler. 2013. Defining dialect regions with interpretations: Advancing the multidimensional scaling approach. *Literary and Linguistic Computing*, 28(1):13–22.

Hans Goebl. 2006. Recent advances in Salzburg dialectometry. *Literary and Linguistic Computing*, 21(4):411–435.

Hans Goebl. 2018. Dialectometry. In Charles Boberg, John Nerbonne, and Dominic Watt, editors, *The handbook of dialectology*, pages 123–142. John Wiley & Sons, Ltd.

Charlotte Gooskens and Wilbert Heeringa. 2004. Perceptive evaluation of Levenshtein dialect distance measurements using Norwegian dialect data. *Language Variation and Change*, 16(3):189–207.

Jack Grieve. 2018. Spatial statistics for dialectology. In Charles Boberg, John Nerbonne, and Dominic Watt, editors, *The handbook of dialectology*, pages 415–433. Wiley Online Library.

Wilbert Heeringa, Peter Kleiweg, Charlotte Gooskens, and John Nerbonne. 2006. Evaluation of string distance algorithms for dialectology. In *Proceedings of the workshop on linguistic distances*, pages 51–62, Sydney, Australia. Association for Computational Linguistics.

Christian Hennig. 2020. *fpc: Flexible procedures for clustering*. R package version 2.2-9.

Brett Kessler. 1995. Computational dialectology in Irish Gaelic. In *Seventh Conference of the European Chapter of the Association for Computational Linguistics*, Dublin, Ireland. Association for Computational Linguistics.

Therese Leinonen, Çağrı Çöltekin, and John Nerbonne. 2016. Using Gabmap. *Lingua*, 178:71–83.

John Nerbonne. 2015. Various variation aggregates in the LAMSAS South. In Michael D. Picone and Catherine Evans Davies, editors, *Language Variety in the South III*. University of Alabama Press, Tuscaloosa, Alabama.

John Nerbonne, Rinke Colen, Charlotte Gooskens, Therese Leinonen, and Peter Kleiweg. 2011. Gabmap – A web application for dialectology. *Dialectologia*, SI II:65–89.

John Nerbonne, Wilbert Heeringa, and Peter Kleiweg. 1999. Comparison and classification of dialects. In *Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics*, EACL '99, page 281–282, USA. Association for Computational Linguistics.

John Nerbonne and Peter Kleiweg. 2003. Lexical distance in LAMSAS. *Computers and the Humanities*, 37(3):339–357.

John Nerbonne, Peter Kleiweg, Wilbert Heeringa, and Franz Manni. 2008. Projecting dialect distances to geography: Bootstrap clustering vs. noisy clustering. In Christine Preisach, Hans Burkhardt, Lars Schmidt-Thieme, and Reinhold Decker, editors, *Data analysis, machine learning and applications*, pages 647–654. Springer Berlin Heidelberg, Berlin, Heidelberg.

Simon Pickl. 2013. *Probabilistische Geolinguistik*. Franz Steiner Verlag, Stuttgart.

R Core Team. 2020. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria.

Jean Séguy. 1971. La relation entre la distance spatiale et la distance lexicale. *Revue de Linguistique Romane*, 35:335–357.

Ryota Suzuki and Hidetoshi Shimodaira. 2006. Pvclust: An R package for assessing the uncertainty in hierarchical clustering. *Bioinformatics*, 22(12):1540–1542.

Johan Taeldeman and Ton Goeman. 1996. Fonologie en morfologie van de Nederlandse dialecten: een nieuwe materiaalverzameling en twee nieuwe atlasprojecten. *Taal en Tongval*, 48:38–59.

Warren S. Torgerson. 1952. Multidimensional scaling: I. Theory and method. *Psychometrika*, 17(4):401–419.

Kevin Ushey, JJ Allaire, and Yuan Tang. 2020. *reticulate: Interface to 'Python'*. R package version 1.18.

Steven H. Weinberger and Stephen A. Kunath. 2011. The speech accent archive: Towards a typology of English accents. In *Corpus-based studies in language use, language learning, and language documentation*, pages 265–281. Brill, Leiden, The Netherlands.

Martijn Wieling, Jelke Bloem, Kaitlin Mignella, Mona Timmermeister, and John Nerbonne. 2014. Measuring foreign accent strength in English: Validating Levenshtein distance as a measure. *Language Dynamics and Change*, 4(2):253–269.

Martijn Wieling, Eliza Margaretha, and John Nerbonne. 2012. Inducing a measure of phonetic similarity from pronunciation variation. *Journal of Phonetics*, 40(2):307–314.

Martijn Wieling and John Nerbonne. 2011. Measuring linguistic variation commensurably. *Dialectologia*, SI II:141–162.

Martijn Wieling and John Nerbonne. 2015. Advances in Dialectometry. *Annual Review of Linguistics*, 1(1):243–264.

Martijn Wieling, Esteve Valls, Rolf H. Baayen, and John Nerbonne. 2018. Border effects among Catalan dialects. In Dirk Speelman, Kris Heylen, and Dirk Geeraerts, editors, *Mixed-effects regression models in Linguistics*, pages 71–97. Springer International Publishing, Cham.