Bachelor's Thesis

Refining the Multidimensional Scaling and Spatial Interpolation of Language Data from NorthEuraLex

Author
Steinar Grässel
steinar.graessel@student.uni-tuebingen.de

Supervisor
Prof Gerhard Jäger
gerhard.jaeger@uni-tuebingen.de

A thesis submitted in partial fulfilment
of the requirements for the degree of

Bachelor of Arts
in
International Studies in Computational Linguistics

January 2023

Antiplagiatserklärung

Ich erkläre hiermit,

dass ich die vorliegende Arbeit selbständig verfasst habe,

dass ich keine anderen als die angegebenen Hilfsmittel und Quellen benutzt habe,

dass ich alle wörtlich oder sinngemäß aus anderen Werken übernommenen Aussagen als solche gekennzeichnet habe,

dass die Arbeit weder vollständig noch in wesentlichen Teilen Gegenstand eines anderen Prüfungsverfahrens gewesen ist,

dass ich die Arbeit weder vollständig noch in wesentlichen Teilen bereits veröffentlicht habe.

Steinar Grässel (Matrikelnummer: 4100139), Januar 2023

Contents

List of Tables

List of Figures

Acronyms

ASJP Automated Similarity Judgment Program

IPA International Phonetic Alphabet

LD Levenshtein Distance

LDN Levenshtein Distance Normalized

LDND Levenshtein Distance Normalized Divided

MDS Multidimensional Scaling

NJ Neighbor Joining

PMI Point-wise Mutual

S2R Similarity to Reference

SD Symmetric Divergence

WALS World Atlas of Language Structures

1 Introduction

The classification of languages and the understanding of their evolutionary relationships are important aspects of comparative linguistics. The NorthEuraLex database is a valuable resource for this type of research, as it contains a large lexical database of words from various languages in Northern Eurasia [1]. This thesis focuses on the computation of pairwise language distances based on the lexical data from NorthEuraLex, and the subsequent multidimensional scaling and spatial interpolation of the data.

In the first section, the pairwise language distances are computed with different distance measures. To evaluate the measures, phylogenetic trees are constructed on the basis of the resulting distance matrices. These trees are scored according to how well they compare to the expert classification in the Glottolog [2]. The distance matrix from the highest-scoring approach is then used as input for multidimensional scaling, which reduces the dimensionality of the matrix and allows for visualization of the languages on maps. These maps are scanned for surprising or interesting deviations from the expected outcome, to then delve further into the affected languages. Finally, spatial interpolation is applied to expand the visualization from the data points to the surface of the Northern Eurasian region.

The goal is to review the utility of the NorthEuraLex database in inferring the classification of languages and to provide insights into the evolutionary relationships between languages in the Northern Eurasian region. Furthermore, limitations and challenges of using these methods are also discussed, along with possible improvements that could be made.

The theoretical research here is intended to build on the work of a prior term paper [3], thus creating a more comprehensive analysis.

All the code for this thesis was written in R using the RStudio IDE [4, 5]. While the use of some packages and function will be highlighted in the paper, the entirety of the code can be found in a set of Jupyter Notebooks on a public GitHub repository, hopefully easing the process of reproducing the results [6].

1.1 Related Research

Automatically language classification has been an extensive topic of research in the field of comparative linguistics. Brown et. al broke new ground with their ASJP, which classifies languages automatically based on their Swadesh lists [7]. It provides a valuable resource for linguists as it can also be used to search for or confirm new relationships between languages. An even shorter lists was introduced by the ASJP team as well, cutting computing time [8].

Both Jäger and Dellert have been successful in further improving methods, especially with regards to automatic cognate detection, by adopting point-wise mutual information methods [9, 10].

For the calculation of tree distances, Asher et. al just recently provided a new approach, specifically for cases where reconstructed trees are compared to a reference tree [11].

2 Data

As discussed earlier, the NorthEuraLex database will serve as the source of the language data. Its lexical data covers 1,017 concepts across 107 northern Eurasian languages [1].

The data set can be downloaded directly from the NorthEuraLex website, where three TSV files are available. For the purpose of this thesis the first two TSV files - containing the lexical data and the language information respectively - are relevant.

2.1 Lexical Data

The first six rows of the data frame with the lexical data are depicted in Table 1. Each row in the data frame corresponds to exactly one word from one of the 107 languages featured in the data set, with its orthographic form located in the third column, *Word_Form*. While the original data set consists of eleven columns, five of them have been removed as they provide additional information that is not relevant for the following analysis.

Language_ID	Concept_ID	Word_Form	rawIPA	ASJP	Next_Step
fin	Auge::N	silmä	silmæ	silmE	validate
fin	Ohr::N	korva	kɔrʋɑ	korwa	validate
fin	Nase::N	nenä	nɛnæ	nEnE	validate
fin	Mund::N	suu	suː	su	validate
fin	Zahn::N	hammas	hɑmːɑs	hamas	validate
fin	Zunge::N	kieli	kiɇli	kiEli	validate

Table 1: First six rows of the lexical data.

The first column, which is titled *Language_ID*, specifies which language the word belongs to. It does so via the ISO 639-3 code, an internationally standardized code to represent languages [12]. The six rows in Table 1 are all Finnish words, denoted by the code *fin*.

The second column is pivotal as it shows which concept the word represents in its language. The *Concept_ID* consist of the German word for the concept, followed by two colons and a shorthand for the type of concept. For example, the concept in row four, *Mund::N*, is a nominal concept, because in most languages the word for the concept *MOUTH*[1] will be a noun. The *N* after the double colon signifies this. Other options include *V* for verbal concepts and *A* for adjectival concepts.

The fourth column contains the phonetic transcription of the words using the International Phonetic Alphabet (IPA).

The ASJP code in the fifth column is a simplification of these IPA transcriptions. It clusters similar sound segments together and therefore only needs 41 different symbols to represent all phonemes [7]. The third row is a prime example of this process, as both the *ɛ* and *æ* phonemes are represented by an *E* in ASJP code.

Finally, the last column is a rating, which reflects how certain the creators of the data set are that a row is correct. In the current state of the data set, all rows have one of two possible entries, *validate* or *review*. The latter is reserved for words which need further review, a state mostly caused by ambiguous sources [1]. Meanwhile *validate* signifies solid sources for the translation and no evidence pointing to another possible word choice for the concept in question [1]. Due to the uncertainty associated with rows labeled *review*, I have opted to ignore them in the coming analysis and they will therefore be removed from the data set.

[1]To clearly distinguish them, concepts will be in UPPER CASE.

2.2 Language Data

The second data frame, a sample of which is depicted in Figure 2, contains 107 rows - one for each language represented in the data set.

name	glotto_code	iso_code	family	subfamily	latitude	longitude
Chuvash	chuv1255	chv	Turkic	Bolgar	55.5	47.2
Modern Hebrew	hebr1245	heb	Afro-Asiatic	Semitic	31.1	35.0
Livonian	livv1244	liv	Uralic	Finnic	57.6	22.0

Table 2: Three sample rows of the language data.

In addition to the ISO code, the Glottocode of each language is also given here. This is an eight character long alphanumerical code used by the Glottolog project, a language catalogue that aims to provide extensive information about all of the world's languages [2]. Each languoid, i.e. a language, a dialect, or a group of either, is assigned a Glottocode. For example, Modern Hebrew has the code *hebr1245* and is - as can be gleaned from the columns labeled *family* and *subfamily* - a Semitic language (*semi1276*) and part of the Afro-Asiatic language family (*afro1255*).

The last two columns contain the real-world coordinates for each language. These are used for visualization and spatial interpolation on maps in later steps.

3 Pairwise Language Distances

Two languages l_1 and l_2 from a list of languages L can be compared by averaging their concept distances. For each concept c_x from a list of concepts C, the distance between the words for c_x in l_1 and l_2 are calculated with a metric for string similarity. Many languages have more than one word for some of the concepts . To simplify later calculations, I've decided to always use the synonym pair with the lowest distance and drop the others. E.g. in Spanish, *MOUNTAIN* can be *monte* or *montaña*. When Spanish and English are compared, two pairs emerge for *MOUNTAIN*: *monte/mountain* and *montaña/mountain*. The pair which is more similar - according to the chosen distance measure - will be used.

The following sections shed light on the details of the language distance calculation, i.e. how the concept list is chosen, which string similarity metrics are applied, and in which form the words are represented.

3.1 ASJP Code

Neither the orthographic word forms nor the IPA transcription will serve as the basis for the distances. Instead the ASJP codes of the words are used, since certain weights that will be introduced in section *5.3.3* are only applicable on ASJP code. As stated in the *Data* section, the ASJP code is a simplification of the IPA transcription. It is brought forward by a group of researchers in their project called the Automated Similarity Judgment Program (ASJP), in which they seek to tackle comparative linguistics by applying computational methods [7].

3.2 Concept Selection

The ASJP project introduces a short list of 40 concepts for calculating language distances. Based on only these 40 concepts automated language classification can achieve high correlation with

expert classifications presented in the Ethnologue or the World Atlas of Language Structures (WALS) [13]. Additionally, comparing additional concepts beyond 40 does not improve the classification results significantly, if at all. This is a consequence of ranking the concepts by stability and preferring stable concepts over less stable concepts. Stability measures how resistant a concept is to change, i.e. words for a stable concept are less likely to be substituted by another word in the course of a language's history [13]. Included in the 40 most stable concepts are e.g., pronouns (*I*, *YOU*, *WE*), body parts (*EYE*, *EAR*, *NOSE*), and concepts describing nature (*MOUNTAIN*, *STAR*, *FIRE*).

Based on these findings, more than 96% of the 1,017 concepts can be removed from the NorthEuraLex lexical data, leaving the remaining 40, most stable concepts[2], significantly cutting later computing time.

3.3 Distance Measures

3.3.1 Levenshtein Distance Normalized

In the prior paper, the chosen metric to calculate the distance between two words was the Levenshtein Distance (LD). It is equal to the minimum number of edit operations to change one string into another [14]. The following example shows the three edits that are necessary to turn *braid* into *rainy*.

$$braid \xrightarrow[+1]{\text{Delete } b} raid \xrightarrow[+1]{\text{Transform } d \text{ to } n} rain \xrightarrow[+1]{\text{Insert } y} rainy = 3$$

Additionally, the distance was normalized, i.e. divided by the length of the longer input string. This ensures that the result is independent of string length [15]. This *Levenshtein Distance Normalized (LDN)* returns a value of $3/5 = 0.6$ for the above example.

3.3.2 Levenshtein Distance Normalized Divided

Wichmann et al. [8] recommend another modification to the distance calculation. To avoid similarity by chance, e.g. because two unrelated languages happen to share a similar phoneme inventory, the LDNs of all the synonym pairs is divided by a factor Γ. To calculate Γ, all the LDNs between words not referring to the same concept are averaged. The final distance is then called *Levenshtein Distance Normalized Divided (LDND)*. While this approach is computationally more intensive, its reported results are slightly better than their LDN counterparts, which is why it is tested as an alternative to LDN here [16, 8].

The concept *FIRE* and its translations in three languages will serve as an example as to why a third distance measure besides LDN and LDND is considered as well.

3.3.3 PMI-based distances

FIRE translates to *Feuer* in German, (obviously) *fire* in English, and *fogo* in Portuguese. The corresponding ASJP codes are *foia*, *fai3*, and *foxu*. To get from *foia* to either *fai3* or *foxu*, two edits are needed (o → a & a → 3 / i → x & a → u), meaning the LDs for both pairs are equal.

[2]To be more precise, they are actually 40 of the 43 most stable concepts. Holman et al. replace *RAIN*, *KILL*, and *BARK*, which are in the top 40, with the concepts in positions 41-43 (*NEW*, *DOG*, *SUN*). This is done because the words for *RAIN*, *KILL*, and *BARK* often share morphemes with the words for *WATER*, *DIE*, and *SKIN* - which are also in the top 40 - turning them into pseudo-duplicate entries [13].

Intuitively though, *Feuer [fɔɤ̯ɐ]* and *Fire [ˈfaɪə]* seem more similar than *Feuer* and *fogo [ˈfoɣu]*. Both *Feuer* and *fire* feature a diphthong - *ɔɤ̯* and *aɪ* - while *fogo* has no diphthong and instead even features a consonant instead of a vowel as the third phoneme of the word.

But the LD treats all edits as equally weighted, meaning a change from a vowel to another vowel has the same effect as a vowel-to-consonant change. To remedy this, a different method is applied, which incorporates weights that rate each phoneme pair individually.

The method is adapted from a paper by Jäger in which the point-wise mutual (PMI) scores for all the ASJP sound classes are presented [9]. These PMI scores can be both positive or negative. Positive PMI scores point to relatedness and are assigned to pairs of similar or equal phonemes. For example, the ASJP sound classes *b* and *p* both represent bilabial stops and fricatives, where *b* is used for voiced and *p* for voiceless phonemes [7]. Since they only differ in voicing, they are similar enough to have a low positive PMI score of 0.44, meaning a change from *b* to *p* is slightly indicative of two compared words being related [9]. *Slightly* is a key word here, as there are significantly stronger indicators both for and against relatedness. Naturally, identical phoneme pairs have high PMI scores, so while the *p/b* pair is a positive indicator, *p/p* is almost 8 times as indicative, with a PMI score of roughly 3.41. Similarly, pairs of very different phonemes like most vowel-consonant pairs, have large negative PMI values, e.g. -10.44 for *p/a*. These PMI scores between sound classes can not be used with the Levenshtein distance, so a different algorithm is used to calculate the PMI scores of word pairs.

The Needleman-Wunsch algorithm, that originates in the field of bioinformatics, takes two strings and finds the alignment of them with the highest score [17]. An alignment can be best visualized by writing one word above the other, where characters are matched pairwise. Table 3 shows two possible ways to align the Latin and Portuguese words for *SLEEP*.

S O M N U S	S O M N U S
S O — N U —	— — S O N U

Table 3: Aligning the ASJP codes of *somnus* and *sono*.

In this example the first alignment would result in a better score, as all the sound classes that occur in both words are paired with each other, marked in blue. Since the strings differ in length, gaps (—) need to be inserted. Gaps contribute a negative value to the PMI score. Here, affine gap penalties are used, i.e. different score penalties for opening (-2.49) and extending (-1.70) a gap.[3] These penalties are also adapted from Jäger [18].

While the LD was easily computed by calling the R function *adist* [4], none of the R implementations of the Needleman-Wunsch algorithm suit the method at hand, as affine gap penalties and the option to use different weights for each sound class pair are necessary. Therefore, to keep all the code in one place, the *affine* function of the Python package *py_stringmatching* is used as a template to write a functionally identical function in R [19]. Testing it with the *hEnd/hant/mano* example[4] from Jäger's paper returns the same results, verifying that the port is successful [9].

The PMI scores are then used to compute the calibrated similarity for each of the concepts. This is done by comparing the word pair of each concept with the PMI scores of all non-synonymous word pairs of the language pair. The calibrated similarity negatively correlates

[3]The gap below the *O* in the second alignment in Table3 is an example for an extending gap, as it is preceded by another gap.

[4]The concept *HAND* in German / English / Spanish and the inspiration for the *FIRE* example.

with the number of non-synonymous pairs that have a higher PMI score than the synonymous pair. This sets a higher bar to jump for languages that are similar to each other, and is therefore a comparable measure to the LDND.

Applying this scoring scheme on the *FIRE* example, results in a calibrated similarity of 6.66 for the word pair *Feuer/fire* and 3.77 for *Feuer/fugo*, which is in line with prior intuitive analysis of *Feuer/fire* being more similar.

Finally, the mean of the calibrated concept similarities is subtracted from the maximum possible calibrated string similarity, transforming the similarities into a dissimilarity measure.[5]

The final language distances for all three methods - LDN, LDND, and PMI-based - can be inspected as CSV files in this folder on the Github repository. They will be analyzed in the next section, as the scores by themselves are hard to interpret.

4 Evaluation of Distance Measures

4.1 Phylogenetic Trees

For the upcoming spatial interpolation, one distance matrix suffices. Therefore a metric is needed to compare the three matrices to then select the best one for use in later steps. To that end, each matrix will first be converted to a phylogenetic tree, showing the supposed relationships between the languages. One of the methods to build these trees is neighbor joining (NJ), which uses the distance matrix to cluster the tree from the leaves upwards, looking to minimize the total length of the tree branches [20].

The R package *ape* provides an implementation of the NJ algorithm in form of the *nj* function [21]. Applying it on each of the three distance matrices returns three trees. Additionally, three trees are created with the similar FastME algorithm, which also minimizes the total branch length, but uses different weights [22].

The resulting trees have 107 leaves, one for each language in the data set. In addition to the full language tree, I also create 6 trees for each of the three language families that are represented with the most languages in NorthEuraLex - Uralic, Indo-European and Turkic. Together they almost make up two thirds of the languages. Figure 1 depicts one of these smaller trees. It shows the Uralic languages in the data set and is the result of NJ with a subset of the distance matrix that was created with the PMI scores and the Needleman-Wunsch algorithm. Languages that are part of the same subfamily are marked with the same colour. Hungarian, Northern Mansi and Northern Khanty are the only languages in their respective subfamilies and are therefore not marked.[6] The dendrogram is a first tentative confirmation that the PMI-based method produces sensible results, as all the subfamilies are clustered together, e.g. the Finnic (purple) or the Sami languages (green).

4.2 Scoring Metric

The Glottolog will serve as a reference point and gold standard for the 24 trees built in the last step. With the R package *glottoTrees* a tree containing all the languoids in the Glottolog can easily be loaded in [23]. The tree is then trimmed, removing all but the languages of the clustered tree it is compared to.

[5] The formulas for the calibrated similarity and the dissimilarity can be found on the final page of [9].

[6] This only pertains to the NorthEuraLex data. For example, there are other languages in the Mansi subfamily - Central and Southern Mansi [2] - but they are not part of the data set.

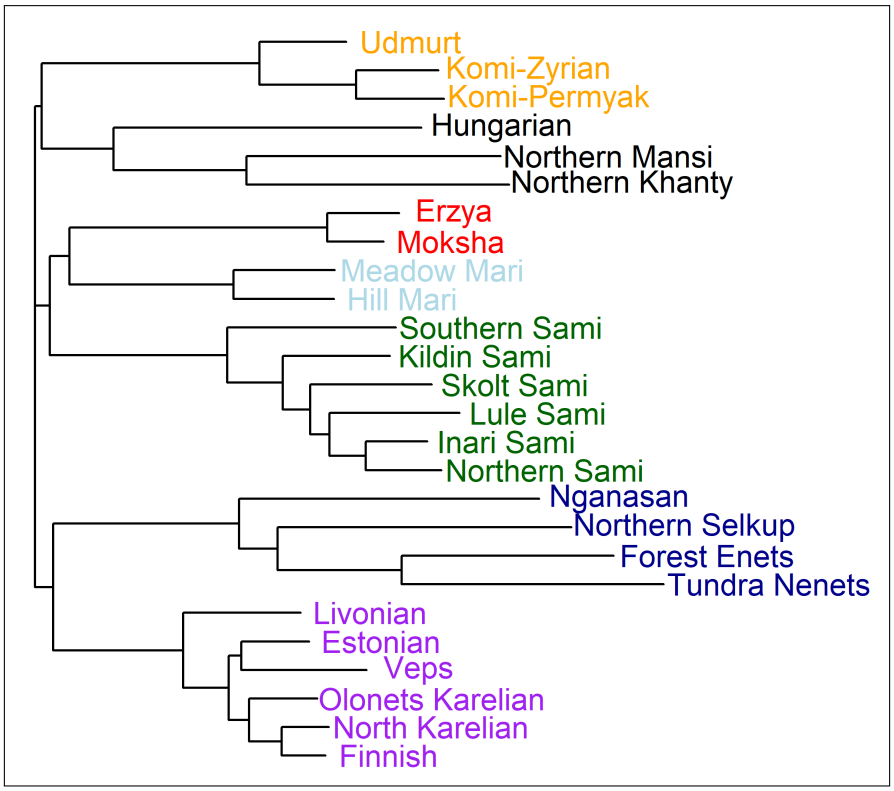


Figure 1: The neighbor joining tree of the Uralic language family.

The Robinson-Foulds distance is a metric commonly used across different scientific fields to calculate distances between trees [24, 25]. However, it is most effective when used on binary trees [16]. While the NJ and FastME trees are binary, the Glottolog tree is not completely resolved and has multifurcating, i.e. non-binary nodes. Therefore, a different metric is chosen, namely the *quartet distance*.

A quartet is defined as a group of four leaves [26]. To use them as a distance measure, each quartet is compared across trees. For example, in Figure 2[7] the first two trees both have the quartet *[ABCD]* resolved as *[AB]/[CD]*, while the third tree has the quartet resolved as *[AD]/[BC]*.

In non-binary trees, not all quartets are resolved. For example, if the tree in Figure 2 was non-binary, it would be possible for the leaves of the quartet *[ABCD]* to be connected directly to the same node creating a group of four leaves that can not be resolved into two distinct groups.

Two modifications of the quartet distance are used here, the *Symmetric Divergence (SD)* and the *Similarity to Reference (S2R)*. The SD normalizes the quartet distance by dividing it by the maximum number of quartets that could have been resolved [27]. Meanwhile, S2R is specifically useful for cases where trees are compared to a reference tree, because it does not "penaliz[e] quartets that are resolved in the reconstructed tree but unresolved in the reference tree" [11]. Therefore, it is a fitting metric here, because the Glottolog tree is not fully resolved and the focus lies on the ability of the reconstructed trees to reproduce its expert classification, while the opposite relation is not relevant.

[7] The code for this figure is adapted from an article by Smith [27]

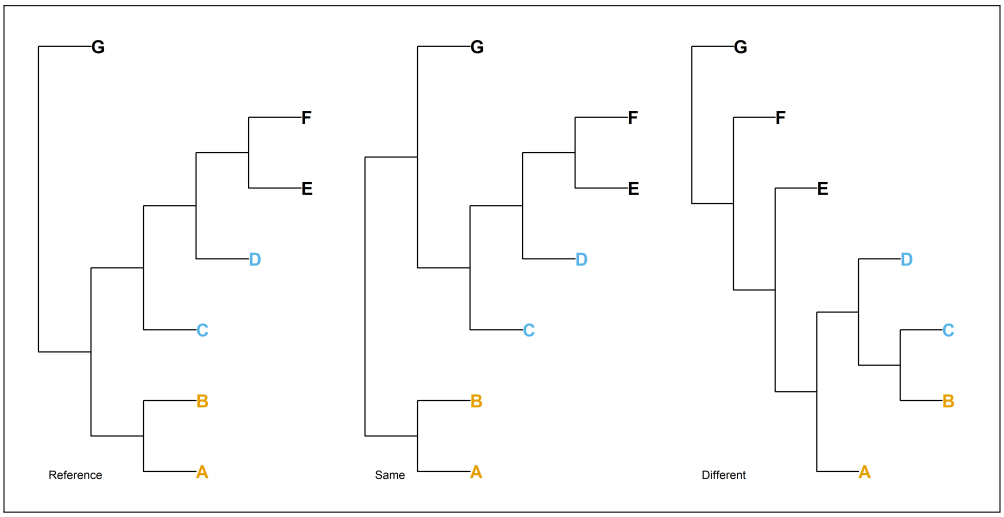


Figure 2: The quartet *[ABCD]* in three binary trees.

The R package *Quartet* provides both of these methods, which can be used to score the 24 trees [28] .

4.3 Results

Table 4 summarizes the results, whereas the lowest distances are always marked **bold**. The column *Δ Resolved Quartets* shows the count of quartets that are resolved differently in the two trees. Meanwhile, columns *SD* and *S2R* contain the metrics discussed in the last section. The scores in the rows labeled *Random* are calculated by creating 100 random trees and averaging their distances and scores when compared with the Glottolog tree. As to be expected, they have much higher distances than the non-random trees, and thus affirm that the reconstructed, non-random trees are in fact able to approximate the Glottolog tree.

Regarding the clustering methods, it is striking that FastME produces identical or better result for all but one tree, the tree with all languages built with the PMI-based distance matrix. The trend here suggest, that FastME should be the preferred clustering method, but a broader pool of examples is needed to consolidate that claim. In any case, the two clustering methods do not significantly shift the results, i.e. the method with the lowest distance never changes between the NJ and FastME trees.

Of the three methods, the LDN method stands out first, as it has no **bold** entries, which means that its trees always have the biggest distance or are at least tied for it. Closer inspection reveals that it doesn't have a single score that is better than either the corresponding LDND or PMI-based scores. While the results are not far off the other two methods - it manages to tie the other methods for a few trees - it still clearly produces the worst results out of the three methods, and will therefore not be considered further.

The remaining choices are the LDND and PMI-based trees. They have identical results for the Turkic family, but PMI trees outperform LDND trees in the Indo-European family, and the Uralic language family. On the other hand, LDND trees outclass the PMI-based trees, when all of the languages in NorthEuraLex are part of the tree.

Therefore, the aim is to use the LDND distance matrix for calculations that involve all the languages, while the dissimilarities in the PMI-based matrix serve as as basis for analysis that

Metric	Δ Resolved Quartets		SD		S2R	
Method	NJ	FastME	NJ	FastME	NJ	FastME
OVERALL						
LDN	27411	26308	0.208	0.207	0.013	0.013
LDND	**12168**	**10054**	**0.205**	**0.204**	**0.006**	**0.005**
PMI-based	18711	19857	0.206	0.206	0.009	0.010
Random	2049003		0.599		1.000	
INDO-EUROPEAN						
LDN	1986	1396	0.176	0.167	0.064	0.045
LDND	1406	1396	0.167	0.167	0.045	0.045
PMI-based	**1290**	**1290**	**0.166**	**0.166**	**0.041**	**0.041**
Random	31235		0.619		1.002	
URALIC						
LDN	422	399	0.237	0.236	0.073	0.069
LDND	422	399	0.237	0.236	0.073	0.069
PMI-based	**359**	**355**	**0.233**	**0.233**	**0.062**	**0.061**
Random	5891		0.603		1.015	
TURKIC						
LDN	14	14	0.364	0.364	0.447	0.447
LDND	**12**	**12**	**0.336**	**0.336**	**0.383**	**0.383**
PMI-based	**12**	**12**	**0.336**	**0.336**	**0.383**	**0.383**
Random	31		0.611		0.998	

Table 4: Quartet distances between clustered trees and Glottolog reference tree.

is restricted to certain language families.

5 Multidimensional Scaling

To visualize the language data on a map, MDS is applied on the distance matrix acquired from the LDND method. This reduces the number of dimensions, as it is very hard to visualize the 107 dimensions of the distance matrix.

A n-dimensional MDS assigns every row in a given distance matrix coordinates in a n-dimensional space, in such a way that the resulting points best resemble the distances from the matrix [29]. For the purposes here, three-dimensional MDS is applied. This results in each of the 107 languages having three coordinates. These can then be converted into a RGB code - one coordinate for the red, green, and blue value - giving each language a unique colour which can be used for visualization.

5.1 Transformation Selection

While different MDS approaches exist, the widely used and recommended SMACOF algorithm, is chosen here [30, 31]. The R package *smacof* implements this method [32]. *Smacof* provides four different options for the transformation of the distance matrix. The ordinal transformation is ignored, as the distance matrix at hand contains metric, not ordinal data. To compare the results of the three remaining transformations, Shepard diagrams are drawn, as pictured in Figure 3. These scatterplots chart the distances of the fitted MDS against the dissimilarities of

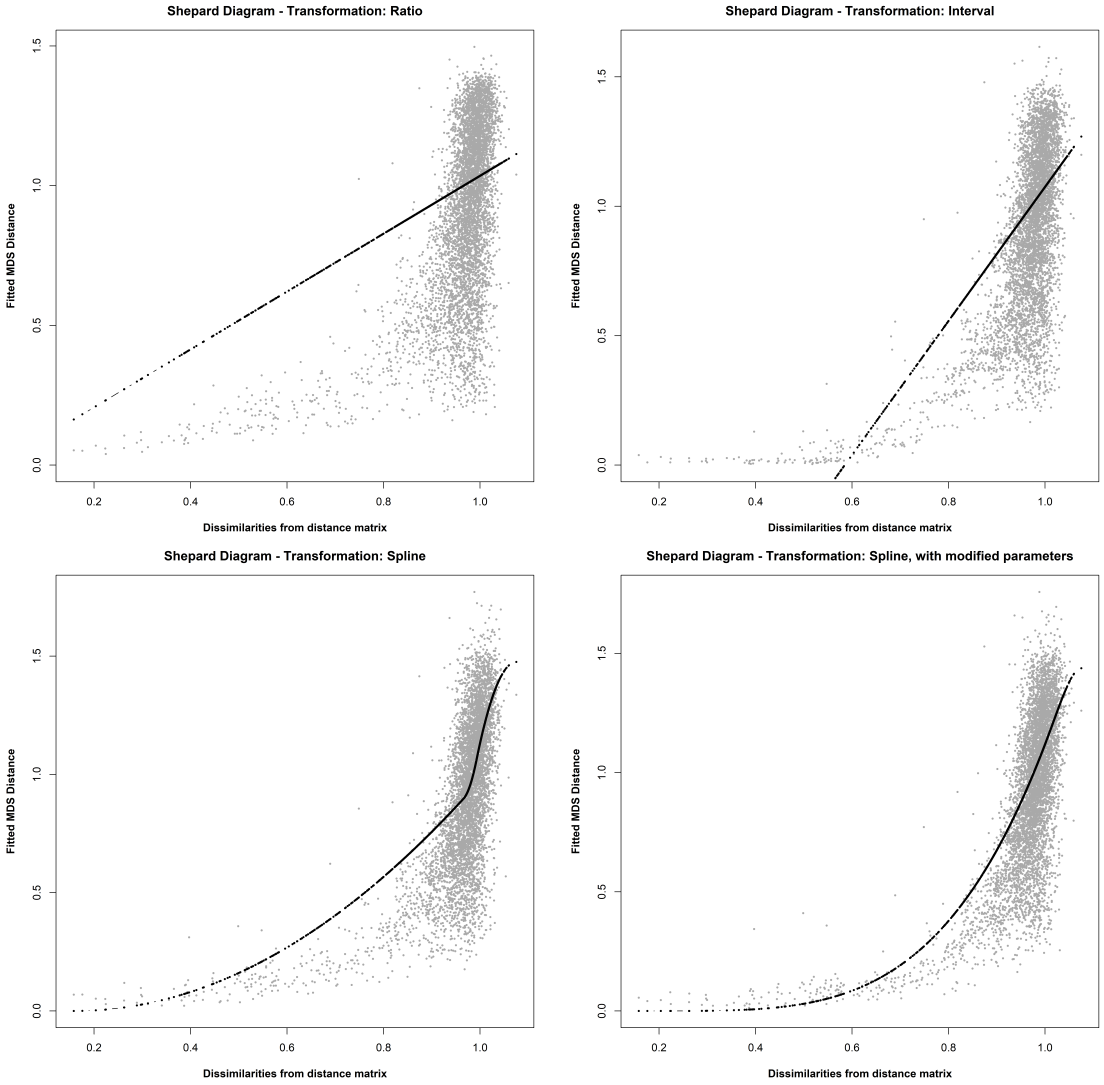


Figure 3: Four Shepard diagrams for different matrix transformation methods.

the input distance matrix [31]. They also include the transformation function. Of the first three diagrams, the third one captures the structure of the data best, as its curved transformation function - *monotone spline* - fits the curve of the data points. The parameter settings of the method are then slightly altered via trial and error, to match the data even better, resulting in the fourth Shepard diagram.

5.2 Stress Test

The better the MDS is fitted, the lower the stress value of the fit should be.[8] Applying SMACOF MDS with the monotone spline transformation results in a stress of 0.254 before, and 0.242 after adapting the parameters. Kruskal provides rules of thumb, which state that a stress of 0.2 or higher signifies a poor fit [33]. As most rules of thumb, this is a simplification and does not necessarily apply to the MDS at hand, since bigger distance matrices - the language distance matrix has 107 rows and columns - result in higher stress values [31].

[8]Stress is the result of the loss function of MDS, i.e. the function that MDS tries to minimize.

To verify the quality of the fit a permutation test is utilized instead. For this, the stress for 1000 permutations of the original distance matrix is computed [34]. If these values are similar to the stress of the original data, it can be assumed that it does not exhibit systematic structure. The stress values for the permutated matrices lie between 0.305 and 0.309 and are thus significantly higher than the original stress, resulting in a p-value smaller than 0.001. Thus, the null-hypothesis - no systematic structure in the original distance matrix - can be rejected. With this the MDS is concluded, and the mapping can commence.

6 Mapping the Languages

As mentioned earlier, the three MDS coordinates of each language are simply converted to RGB values, creating a unique colour for each language. Combining them with the real world coordinates from NorthEuraLex - introduced in section 2.2 - allows for a map of the languages to be drawn. Depicted in Figure 4, the map uses the Equal Earth projection with the central meridian moved to 125° E [35]. This brings the two languages that can now be seen on the far right of the map - Central Siberian Yupik and Aleut - closer to the other languages, giving a clearer, more zoomed-in view on the languages [3].

The maps for the two biggest language families of the data set - Indo-European and Uralic - are collected at the end of the thesis to avoid cluttering the pages here. The colours on those maps are re-scaled and not do use the colours from the first map with all languages. This makes it easier to spot the differences within the languages families.

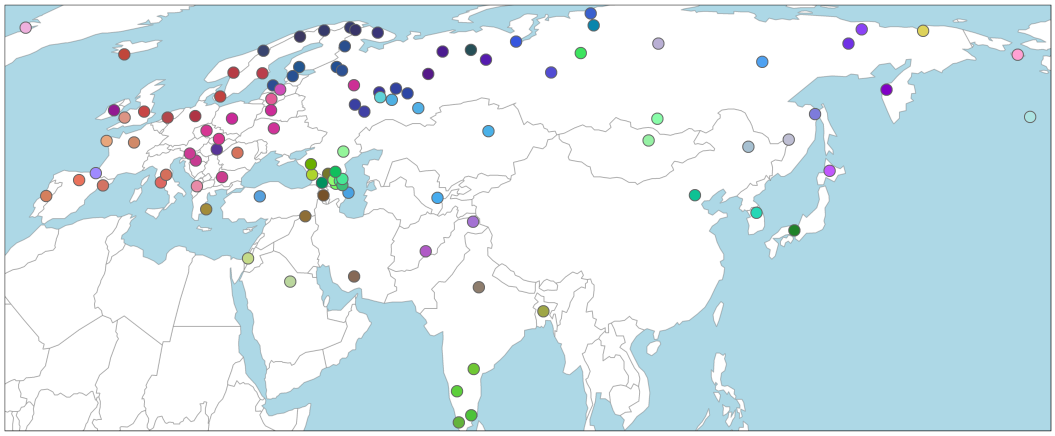


Figure 4: The 107 languages of NorthEuraLex.

6.1 Uralic

Figure 5a shows all the Uralic languages. Some of the language families are clearly identifiable as distinct groups , e.g. Sami and Finnic in brown and green(-grey) respectively. The eastern-most languages on the map, which are the four representatives of the Samoyedic subfamily in the data set, do not match up quite as well. Four different models compared by Blažek all connect the Enets and Nenets families on the lowest level [9], while the map at hand suggests a closer relationship of (Tundra) Nenets and Nganasan [36]. Interestingly, both the FastME and the NJ tree based on the LDND data do actually have a clade which only combines Forest Enets and Tundra Nenets. Thus, the multi-dimensional scaling might be the cause of this distortion.

[9] These include Forest Enets and Tundra Nenets respectively, which are on the map

Considering Figure 3 again, it is clear that most of the language distances in the LDND matrix are very high. In fact, more than 78% of the pairwise distances in the LDND matrix have a value higher than 0.95. The amount of actual information these scores for unrelated languages give is disputable. For example, Basque - a language isolate that is understood to not be related to any other language - has distances of 1.034 and 0.950 to Dutch and Nivkh respectively [37]. As it can not be reasonably argued that Basque is more related to another isolate language in the Russian Far East than to Dutch, it is assumed that these difference are by-chance effects, which even the additional correction from LDN to LDND cannot control [38]. After all the difference is only 0.084. But nearly 78% of the distances are in this range between 0.950 and 1.034 and they are all considered by the MDS. The small by-chance effects can therefore add up, to muddy the results.

Therefore, MDS is recalculated for Uralic with the PMI-based distance matrix. As discussed in section 4.3, it performed better than LDND on family trees in the evaluation. Inspecting the Shepard diagram for the MDS on the PMI-based distance matrix is already slightly reassuring, as the distribution of the scores isn't quite as top-heavy.

The updated map can be inspected as Figure 5b. The Samoyedic language dots match the cited research better, as Nenets and Enets are clearly more similar than in Figure 5a. Moreover, Hungarian is more aligned with Mansi and Khanty than Permian now, which agrees with the sometimes proposed Ugric branch that combines Hungarian, Mansi and Khanty [39, 40]. While connections between Permian and Hungarian have also been described, they have not been deemed as strong as the Ugric link [40].

6.2 Indo-European

For the second big family, Indo-European, four maps are collected in Figure 7 and 8. The maps 7a and 8a are the Indo-European equivalent of the Uralic maps in Figure 5. They confirm again that the grouping of major branches with many representatives in the data set works quite well, as the Balto-Slavic, Germanic, and Italic languages are fairly clear to see.

6.2.1 Irish & Three Eastern Mediterranean Languages

Irish is noticeable in both 7 and 8, as it seems completely detached from the other members of the Celtic branch, Welsh and Breton. This is not to say Irish is as similar to Welsh and Breton as they are to each other. Its language branch supposedly split from the branch that led to Welsh and Breton as early as 1,100 BC [41]. But in the current map state there's not even a hint of the common ancestors of the Celtic branch.

There are three other languages that are interesting, (Standard) Albanian, Armenian and Greek. They are typically considered to be on independent branches of the Indo-European tree, not closely related to any other languages [42]. In 7a, Armenian and Greek are seemingly connected, while Albanian is relatively isolated. Meanwhile, in 8a Albanian and Armenian are closer, with Greek being isolated. When reviewing different sources, the clade connecting Armenian and Greek comes up consistently, while the Albanian branch is generally thought to split earlier, indicating that the LDND map is more accurate here [42].

6.2.2 Rogue Taxa

In addition to the split between LDND and PMI-based maps, two more maps are given in Figure 7b and 8b. The maps are created by removing rogue taxa from the distance matrix and

then applying MDS. Rogue taxa have no clear position in a phylogenetic tree and will cause scoring metrics that examine multiple trees to have worse results, as they can be in vastly different positions for each tree [9]. While the current section is concerned with maps and not trees, it is nevertheless worth it to examine the impact of these taxa - if any - on the MDS and mapping. The list of rogue taxa has been adapted from [9] and can be inspected here. It contains languages from the significantly bigger ASJP database, so not all of them pertain to the NorthEuraLex analysis here.

Figure 7b doesn't yield a lot of new information, as removing the rogue taxa does not seem to affect the MDS much. The biggest difference is probably Breton being slightly more aligned with Welsh. The measure is more effective on the PMI-based distance matrix, as the pair of maps in Figure 8 clearly differs more than their LDND counterpart. Irish is now identifiable as a part of the Celtic branch, with the three languages sharing green as their primary colour. Additionally, Greek morphs from pink to orange, edging closer to Albanian and Armenian. As mentioned earlier, Armenian and Greek are considered to be related more closely, but there is a model by Starostin that proposes that Armenia, Albanian and Greek all split at the same time. This would at least be a somewhat more fitting phylogengy for 8b [41].

7 Spatial Interpolation

Spatial interpolation estimates values at unknown locations based on the known values at nearby locations. It takes points and generates a continuous surface that describes the relationships between the point values and their locations [43]. It is used here to colourize a regional map, where the MDS results of the languages are the values. The spatial interpolation here focuses on a subset of 62, mostly European languages. This is to avoid interpolating over massive distances and regions where none of the languages in NorthEuraLex are represented. The 62 languages can be seen in Figure 9.

The interpolation method of choice is ordinary kriging - implemented in the R package *gstat* [44] - which models the spatial data by fitting variograms. They measure the similarity between values of a variable at different locations, plotting the squared differences of the points against the distance between them. Figure 10 depicts three variograms. 10a contains all points, while the sample variogram in Figure 10b plots the averages of the points at certain distance intervals. It is less cluttered and easier to interpret, especially when deciding on a model for the variogram. Many different models are available, e.g. exponential, linear models or - as applied in Figure 10c - the hole model [44]. The parameters of each model can be tuned, to acquire a optimal fit. For example, it's important to consider the cutoff distance, which controls up to which distance pairs of points are put in relation with each other [45]. When a fitting model is found, the Kriging process can begin.

Kriging uses the fitted variogram to predict values anywhere on the map, which needs to be partitioned into a grid. Since the MDS data has three dimensions, the process is repeated three times, producing a map for each MDS dimension, see Figure 11. The three dimensions are then once again combined, to produce the final map seen in Figure 12a. While 12a is very smooth, choosing different variogram fits, can change the final map drastically. When short cutoff distances are combined with an exponential model fit, a map like Figure 12b is the result, where the original language points are more prominent. It's up to the map-maker to find a balance between smoothing over all local outliers and just recreating the original points with some color in between.

8 Conclusion

Within this thesis, the NorthEuraLex data set has shown itself to be a fruitful base for computational analysis of language relationships. Throughout, many different methods were examined and measured.

Due to the many distinct segments of the pipeline that connects lexical data to a spatially interpolated map, there are a lot of possible improvements to delve into.

Some language families are represented with very languages in NorthEuraLex, and it might be helpful to either consult other data sets for additional data or ignore some families in the analysis. It is probably more effective to compare the heavily represented Indo-European or Uralic families within itself, than against sole representatives of language families on the other side of the globe, e.g. Mandarin Chinese or isolates like Nivkh.

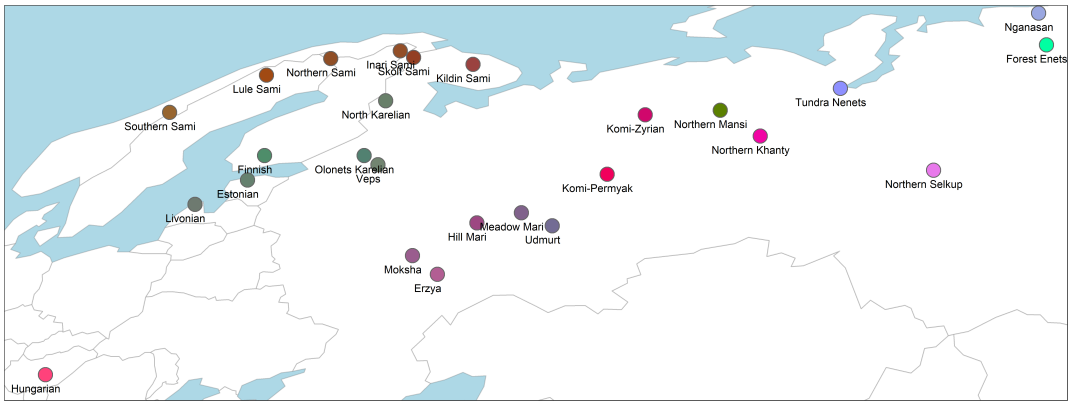
Another improvement would be to removing rogue taxa when the distance matrices are evaluated, to gauge their impact more directly and instead of adapting rogue taxa, the data set can be analyzed to identify them directly [9]. Additionally, the tree analysis after the computation of the distance matrices could be improved upon, e.g. by analysing permutations of the distance matrix via bootstrapping [9, 46].

Furthermore, while reviewing the maps on a visual level already leads to interesting questions, it would be important for further work to focus on analysing the results of the multidimensional scaling directly and not rely purely on eyesight.

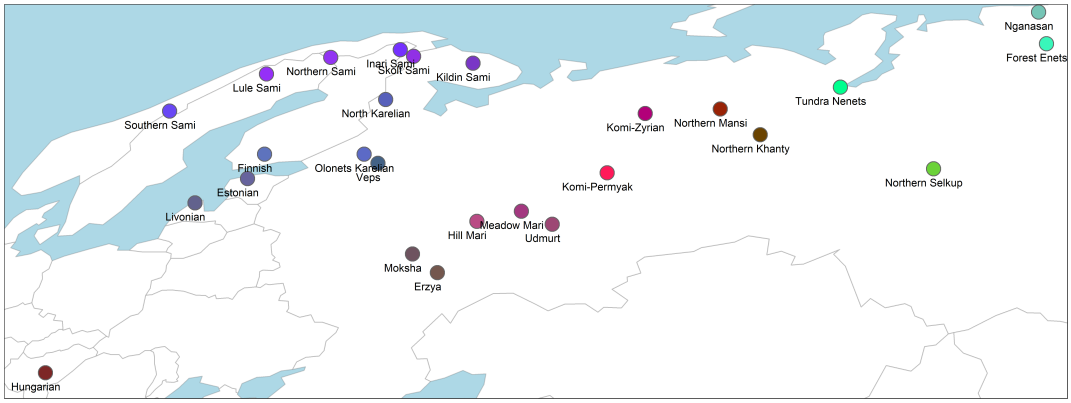
9 Acknowledgments

I want to thank the developers of the R language, R Studio, and all the R packages that were invaluable in processing, analysing, and visualizing the language data [4, 5, 21, 23, 28, 47, 48, 32, 49, 50, 51]. Moreover, I want to express my gratitude to Professor Jäger for acting as supervisor and sparking my interest for the topics discussed in this thesis during his *Languages in Space* seminar.

Additional Figures



(a) MDS on the LDND matrix.



(b) MDS on the PMI-based matrix.

Figure 5: The Uralic languages.

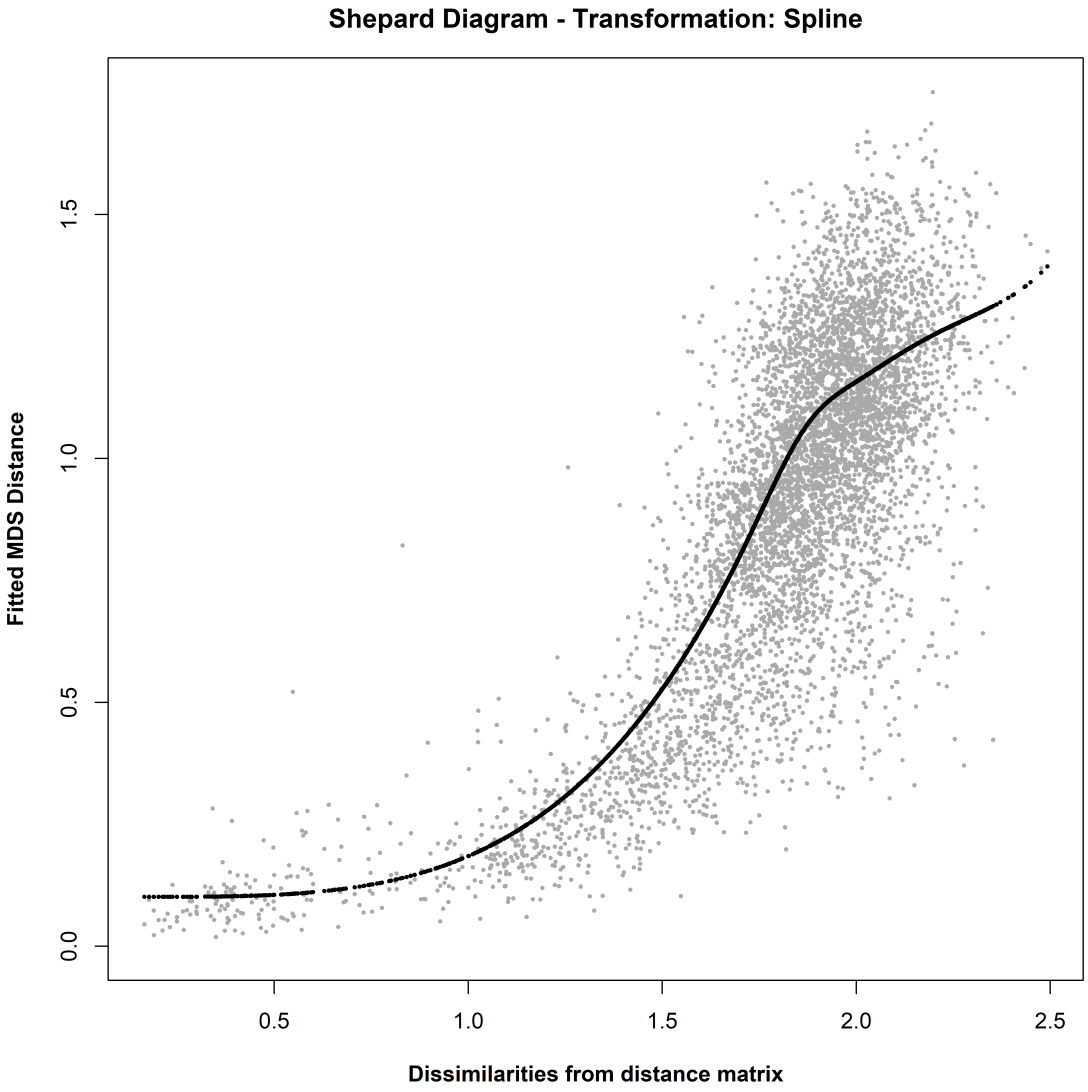
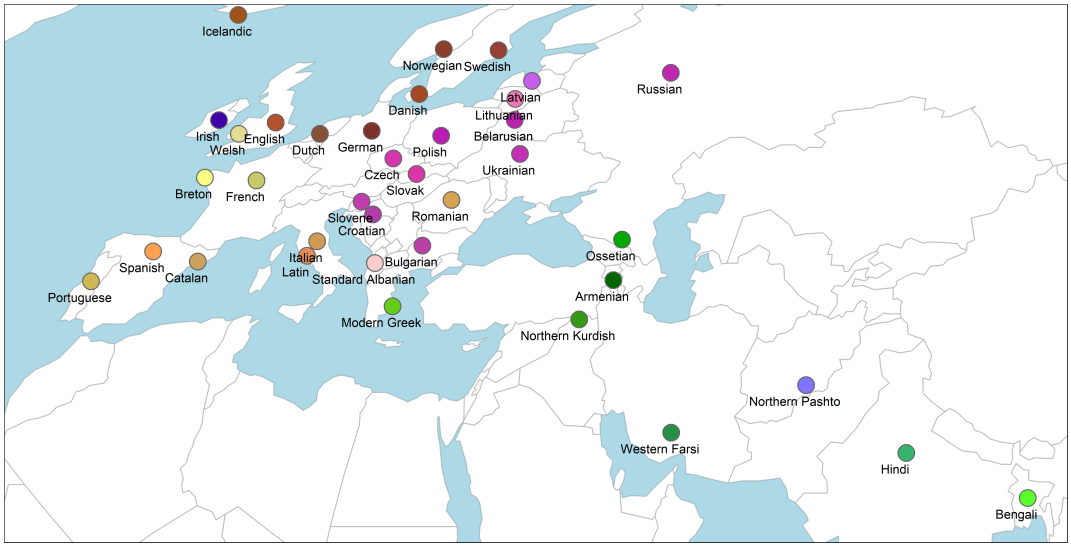
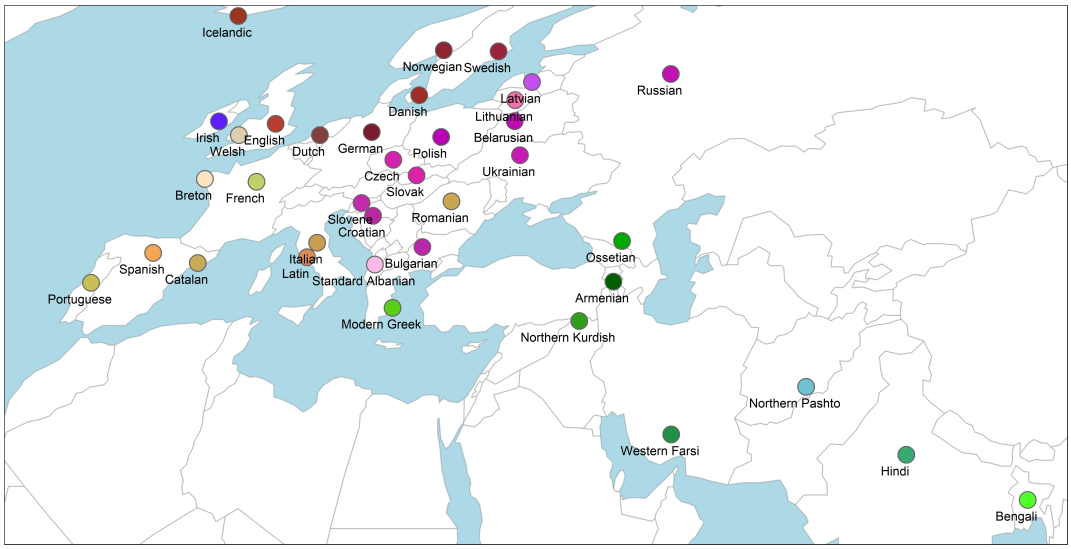


Figure 6: Shepard diagram for MDS on PMI-based distance matrix.

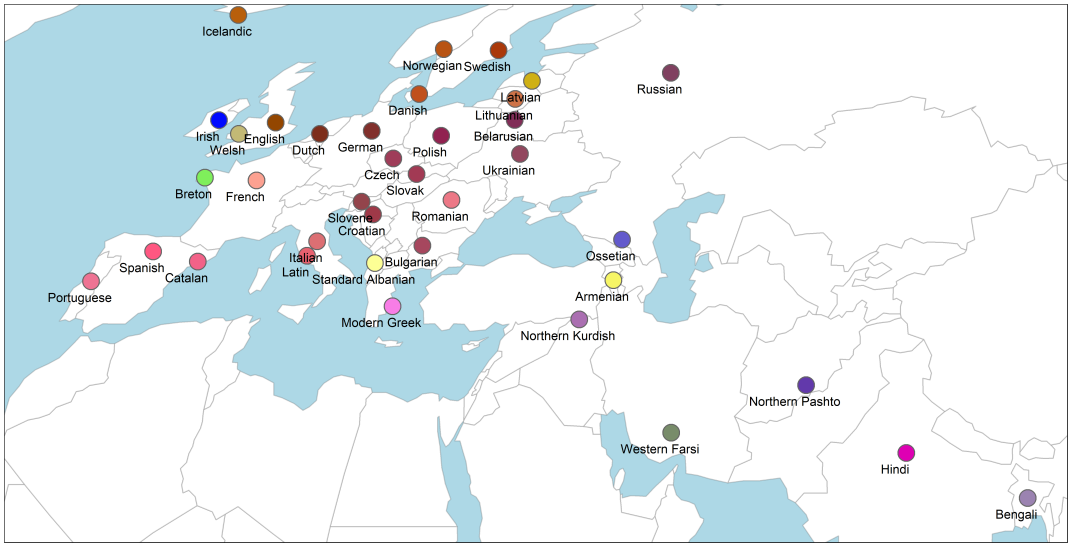


(a) MDS on the complete distance matrix

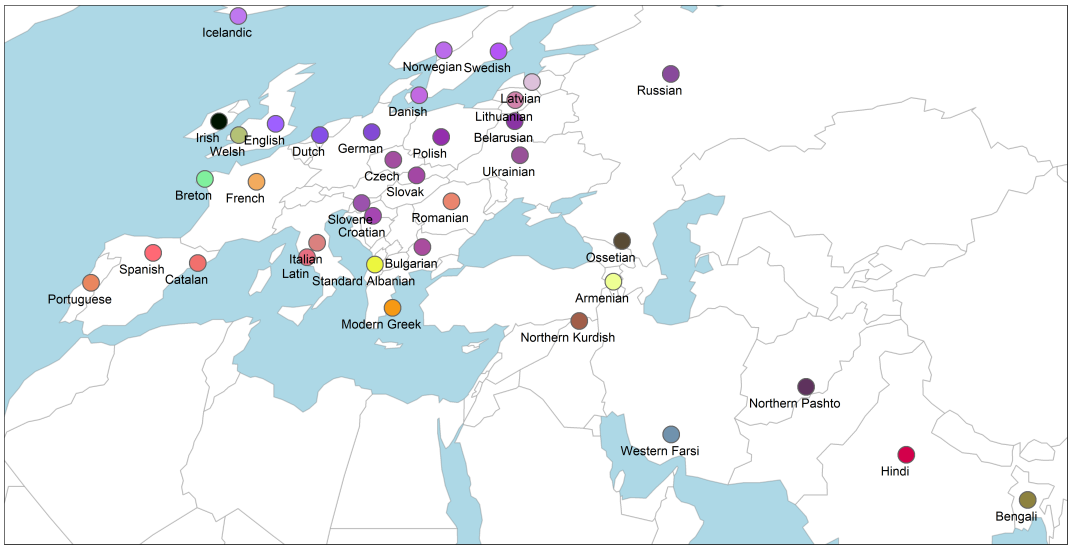


(b) MDS on matrix without potential rogue taxa.

Figure 7: The Indo-European languages - LDND distance matrix.



(a) MDS on the complete distance matrix.



(b) MDS on matrix without potential rogue taxa.

Figure 8: The Indo-European languages - PMI-based distance matrix.

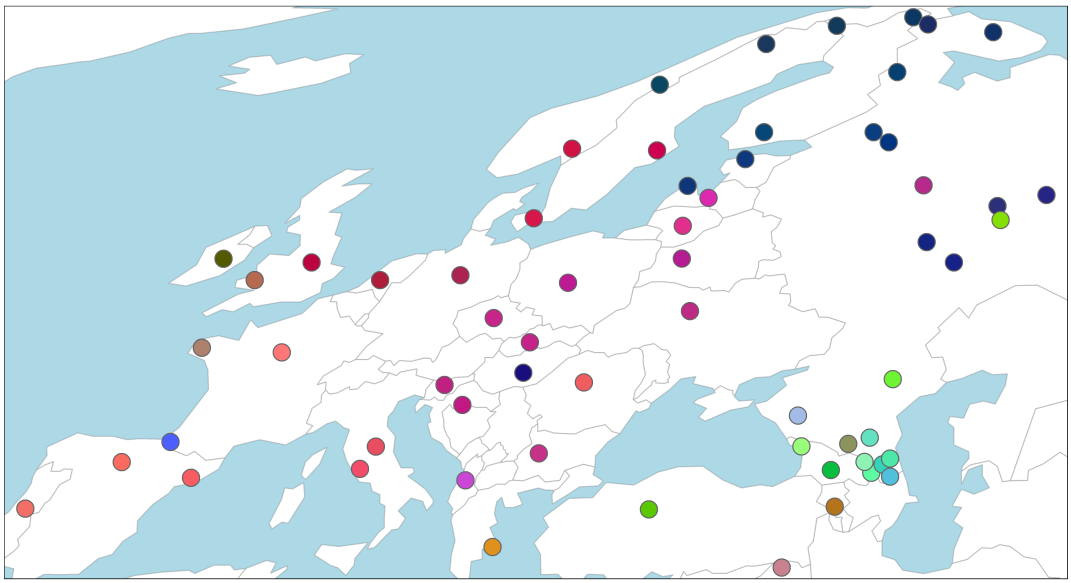
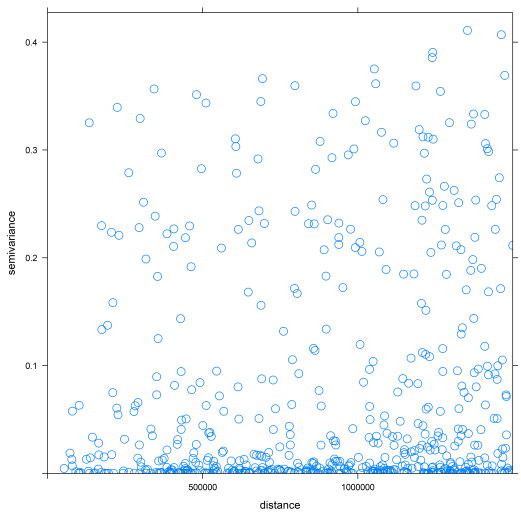
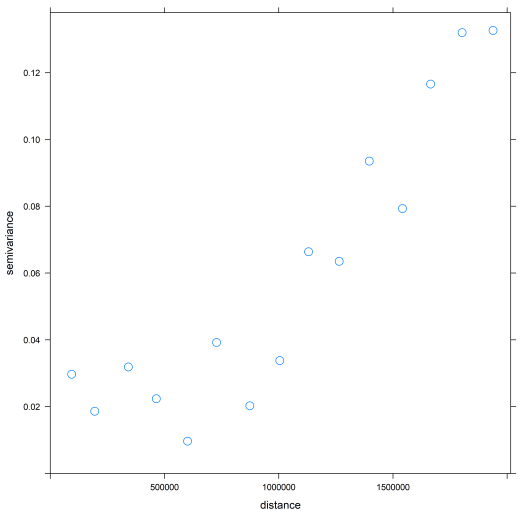


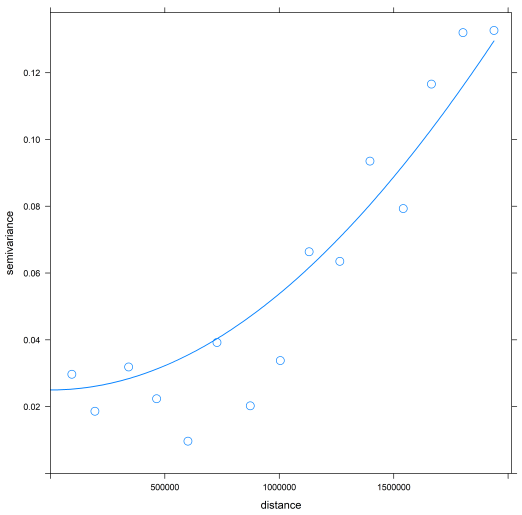
Figure 9: The languages that are used for ordinary kriging.



(a) Cloud variogram.

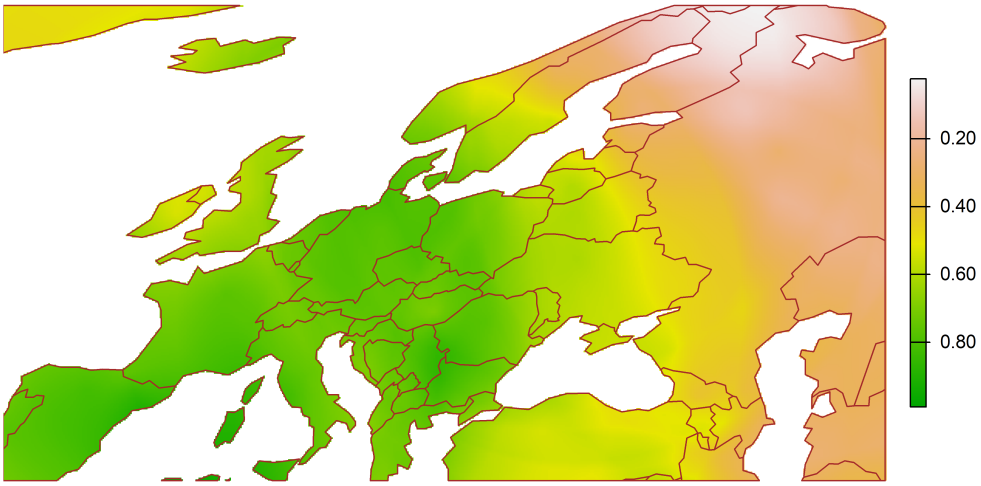


(b) Sample variogram.

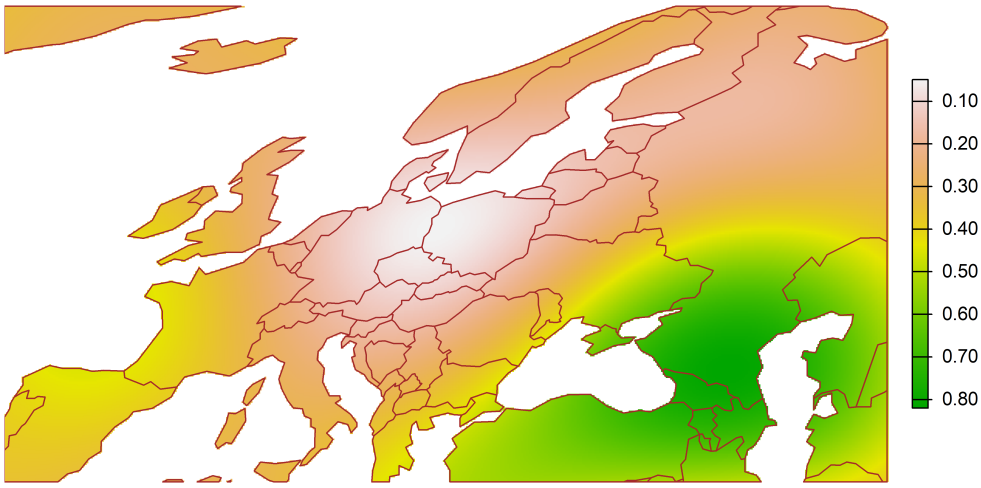


(c) Fitted variogram.

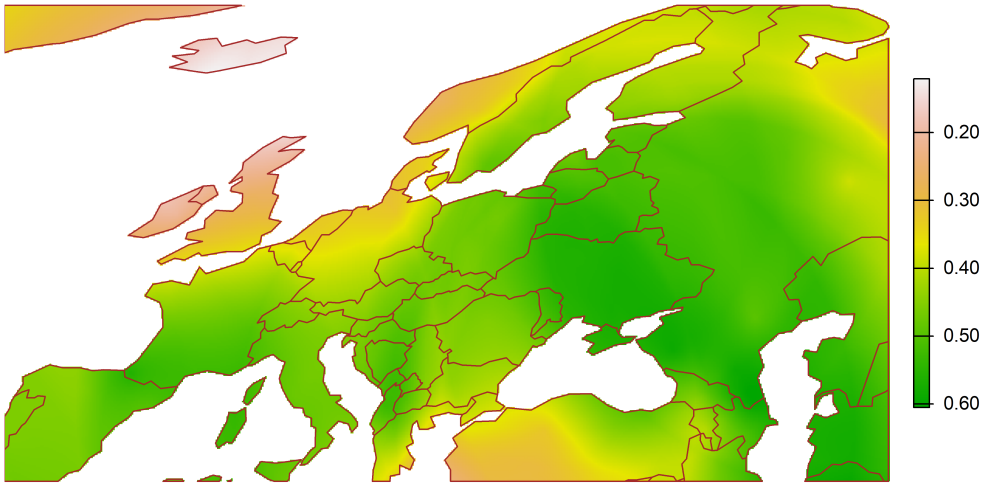
Figure 10: Three Variograms.



(a) First Dimension (red).

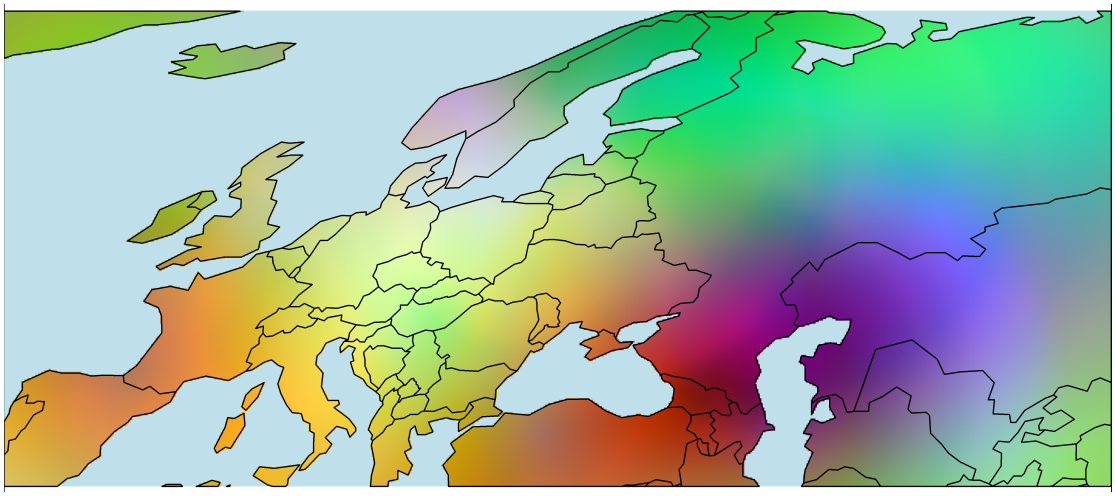


(b) Second Dimension (green).

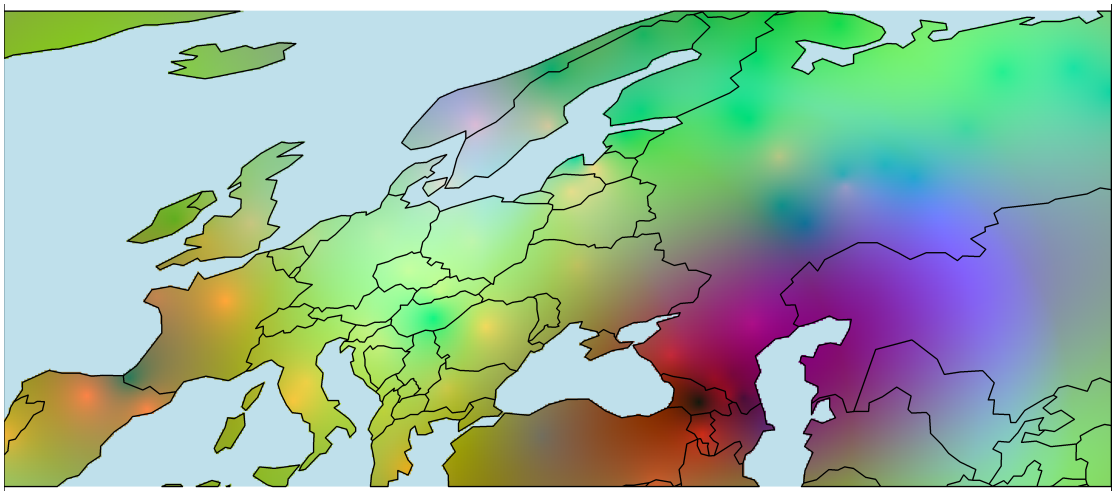


(c) Third Dimension (blue).

Figure 11: The three dimensions after kriging.



(a) A smooth kriging result.



(b) A kriging result with more prominent original points.

Figure 12: Two different kriging results.

References

[1] Johannes Dellert et al. "NorthEuraLex: A wide-coverage lexical database of Northern Eurasia". In: *Language resources and evaluation* 54.1 (2020), pp. 273–301.

[2] Sebastian Nordhoff and Harald Hammarström. "Glottolog/Langdoc: Defining dialects, languages, and language families as collections of resources". In: *First International Workshop on Linked Science 2011-In conjunction with the International Semantic Web Conference (ISWC 2011)*. 2011.

[3] Steinar Grässel. *MDS-for-NorthEuraLex*. May 2022. URL: https://github.com/Arcumenn/MDS-for-NorthEuraLex/blob/main/MDS_for_NorthEuraLex.ipynb (visited on 01/17/2023).

[4] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2022. URL: https://www.R-project.org/.

[5] RStudio Team. *RStudio: Integrated Development Environment for R*. RStudio, PBC. Boston, MA, 2021. URL: http://www.rstudio.com/.

[6] Thomas Kluyver et al. "Jupyter Notebooks – a publishing format for reproducible computational workflows". In: *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. Ed. by F. Loizides and B. Schmidt. IOS Press. 2016, pp. 87–90.

[7] Cecil H. Brown et al. In: *Language Typology and Universals* 61.4 (2008), pp. 285–308. DOI: doi:10.1524/stuf.2008.0026. URL: https://doi.org/10.1524/stuf.2008.0026.

[8] Søren Wichmann et al. "Evaluating linguistic distance measures". In: *Physica A: Statistical Mechanics and its Applications* 389.17 (2010), pp. 3632–3639. ISSN: 0378-4371. DOI: https://doi.org/10.1016/j.physa.2010.05.011. URL: https://www.sciencedirect.com/science/article/pii/S0378437110003997.

[9] Gerhard Jäger. "Support for linguistic macrofamilies from weighted sequence alignment". In: *Proceedings of the National Academy of Sciences* 112.41 (2015), pp. 12752–12757. DOI: 10.1073/pnas.1500331112. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.1500331112. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1500331112.

[10] Johannes Dellert. "Combining Information-Weighted Sequence Alignment and Sound Correspondence Models for Improved Cognate Detection". In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 3123–3133. URL: https://aclanthology.org/C18-1264.

[11] Robert J Asher and Martin R Smith. "Phylogenetic Signal and Bias in Paleontology". In: *Systematic Biology* 71.4 (Sept. 2021), pp. 986–1008. ISSN: 1063-5157. DOI: 10.1093/sysbio/syab072. eprint: https://academic.oup.com/sysbio/article-pdf/71/4/986/44114576/syab072.pdf. URL: https://doi.org/10.1093/sysbio/syab072.

[12] *ISO - ISO 639 — Language codes*. ISO. URL: https://www.iso.org/iso-639-language-codes.html (visited on 01/17/2023).

[13] Eric W Holman et al. "Explorations in automated language classification". In: *Folia Linguistica* 42.2 (2008), pp. 331–354.

[14] Vladimir I Levenshtein et al. "Binary codes capable of correcting deletions, insertions, and reversals". In: *Soviet physics doklady*. Vol. 10. 8. Soviet Union. 1966, pp. 707–710.

[15] M. Serva and F. Petroni. "Indo-European languages tree by Levenshtein distance". In: *Europhysics Letters* 81.6 (Feb. 2008), p. 68005. DOI: 10.1209/0295-5075/81/68005. URL: https://dx.doi.org/10.1209/0295-5075/81/68005.

[16] Simone Pompei, Vittorio Loreto, and Francesca Tria. "On the Accuracy of Language Trees". In: *PLOS ONE* 6.6 (June 2011), pp. 1–11. DOI: 10.1371/journal.pone.0020109. URL: https://doi.org/10.1371/journal.pone.0020109.

[17] Saul B Needleman and Christian D Wunsch. "A general method applicable to the search for similarities in the amino acid sequence of two proteins". In: *Journal of molecular biology* 48.3 (1970), pp. 443–453.

[18] Gerhard Jäger. "Phylogenetic Inference from Word Lists Using Weighted Alignment with Empirically Determined Weights". In: *Language Dynamics and Change* 3 (Jan. 2013), pp. 245–291. DOI: 10.1163/22105832-13030204.

[19] The Magellan Development Team. *Developer Manual for py-stringmatching 0.4.x*. 2017. URL: https://pages.cs.wisc.edu/~anhai/py_stringmatching/v0.4.0/dev-manual-v0.4.0.pdf (visited on 01/17/2023).

[20] N Saitou and M Nei. "The neighbor-joining method: a new method for reconstructing phylogenetic trees." In: *Molecular Biology and Evolution* 4.4 (July 1987), pp. 406–425. ISSN: 0737-4038. DOI: 10.1093/oxfordjournals.molbev.a040454. eprint: https://academic.oup.com/mbe/article-pdf/4/4/406/11167444/7sait.pdf. URL: https://doi.org/10.1093/oxfordjournals.molbev.a040454.

[21] E. Paradis and K. Schliep. "ape 5.0: an environment for modern phylogenetics and evolutionary analyses in R". In: *Bioinformatics* 35 (2019), pp. 526–528.

[22] Richard Desper and Olivier Gascuel. "Fast and accurate phylogeny reconstruction algorithms based on the minimum-evolution principle". In: *International Workshop on Algorithms in Bioinformatics*. Springer. 2002, pp. 357–374.

[23] Erich R. Round. *glottoTrees: Phylogenetic trees in Linguistics*. R package version 0.1. 2021. URL: https://github.com/erichround/glottoTrees.

[24] D.F. Robinson and L.R. Foulds. "Comparison of phylogenetic trees". In: *Mathematical Biosciences* 53.1 (1981), pp. 131–147. ISSN: 0025-5564. DOI: https://doi.org/10.1016/0025-5564(81)90043-2. URL: https://www.sciencedirect.com/science/article/pii/0025556481900432.

[25] Martin R Smith. "Information theoretic generalized Robinson–Foulds metrics for comparing phylogenetic trees". In: *Bioinformatics* 36.20 (2020), pp. 5007–5013.

[26] George F. Estabrook, F. R. McMorris, and Christopher A. Meacham. "Comparison of Undirected Phylogenetic Trees Based on Subtrees of Four Evolutionary Units". In: *Systematic Zoology* 34.2 (1985), pp. 193–200. ISSN: 00397989. URL: http://www.jstor.org/stable/2413326 (visited on 01/25/2023).

[27] Martin R Smith. *About the quartet distance*. en. URL: https://ms609.github.io/Quartet/articles/Quartet-Distance.html (visited on 01/26/2023).

[28] Martin R. Smith. *Quartet: comparison of phylogenetic trees using quartet and split measures*. R package version 1.2.5. 2019. DOI: 10.5281/zenodo.2536318.

[29] Leland Wilkinson et al. “Multidimensional scaling”. In: *Systat* 10.2 (2002), pp. 119–145.

[30] Sean Eom. “Multidimensional scaling”. In: *Author Cocitation Analysis: Quantitative Methods for Mapping the Intellectual Structure of an Academic Discipline*. IGI Global, 2009, pp. 225–254.

[31] Patrick Mair, Ingwer Borg, and Thomas Rusch. “Goodness-of-Fit Assessment in Multidimensional Scaling and Unfolding”. In: *Multivariate Behavioral Research* 51.6 (2016). PMID: 27802073, pp. 772–789. DOI: 10.1080/00273171.2016.1235966.

[32] Jan de Leeuw and Patrick Mair. “Multidimensional Scaling Using Majorization: SMACOF in R”. In: *Journal of Statistical Software* 31.3 (2009), pp. 1–30. URL: https://doi.org/10.18637/jss.v031.i03.

[33] J. B. Kruskal. “Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis”. In: *Psychometrika* 29.1 (Mar. 1964), pp. 1–27. ISSN: 1860-0980. DOI: 10.1007/BF02289565. URL: https://doi.org/10.1007/BF02289565.

[34] Eric Dexter, Gretchen Rollwagen-Bollens, and Stephen M Bollens. “The trouble with stress: A flexible method for the evaluation of nonmetric multidimensional scaling”. In: *Limnology and Oceanography: Methods* 16.7 (2018), pp. 434–443.

[35] Bojan Šavrič, Tom Patterson, and Bernhard Jenny. “The Equal Earth map projection”. In: *International Journal of Geographical Information Science* 33.3 (2019), pp. 454–465. DOI: 10.1080/13658816.2018.1504949. URL: https://doi.org/10.1080/13658816.2018.1504949.

[36] Václav Blažek. “On the classification of the Samoyedic languages”. In: *Finnisch-Ugrische Forschungen* 63 (2016), pp. 79–125.

[37] Robert L Trask. “Origins and relatives of the Basque language: Review of the evidence”. In: *AMSTERDAM STUDIES IN THE THEORY AND HISTORY OF LINGUISTIC SCIENCE SERIES 4* (1995), pp. 65–100.

[38] Michael Fortescue. “The relationship of Nivkh to Chukotko-Kamchatkan revisited”. In: *Lingua* 121.8 (2011), pp. 1359–1376.

[39] Matthew S. Dryer and Martin Haspelmath, eds. *WALS Online*. Leipzig: Max Planck Institute for Evolutionary Anthropology, 2013. URL: https://wals.info/.

[40] Jyri Lehtinen et al. “Behind family trees: secondary connections in Uralic language networks”. In: *Language Dynamics and Change* 4.2 (2014), pp. 189–221.

[41] Václav Blažek et al. “On the internal classification of Indo-European languages: survey”. In: *Linguistica ONLINE. http://www. phil. muni. cz/linguistica/art/blazek/bla-003. pdf* (2005).

[42] James Clackson. *Indo-European linguistics: an introduction*. Cambridge University Press, 2007.

[43] Hans Wackernagel. *Multivariate geostatistics: an introduction with applications*. Springer Science & Business Media, 2003.

[44] Benedikt Gräler, Edzer Pebesma, and Gerard Heuvelink. “Spatio-Temporal Interpolation using gstat”. In: *The R Journal* 8 (1 2016), pp. 204–218. URL: https://journal.r-project.org/archive/2016/RJ-2016-014/index.html.

[45] Humboldt-Universität zu Berlin | Geography Department. *Spatial interpolation in R*. URL: https://pages.cms.hu-berlin.de/EOL/gcg_quantitative-methods/Lab14_Kriging.html.

[46] Pamela S Soltis and Douglas E Soltis. “Applying the bootstrap in phylogeny reconstruction”. In: *Statistical Science* (2003), pp. 256–267.

[47] Martin R. Smith. *TreeDist: Distances between Phylogenetic Trees. R package version 2.5.0.* 2020. DOI: 10.5281/zenodo.3528124.

[48] Hadley Wickham et al. “Welcome to the tidyverse”. In: *Journal of Open Source Software* 4.43 (2019), p. 1686. DOI: 10.21105/joss.01686.

[49] Martijn Tennekes. “tmap: Thematic Maps in R”. In: *Journal of Statistical Software* 84.6 (2018), pp. 1–39. DOI: 10.18637/jss.v084.i06.

[50] Roger Bivand, Jakub Nowosad, and Robin Lovelace. *spData: Datasets for Spatial Analysis.* R package version 2.2.1. 2022. URL: https://CRAN.R-project.org/package=spData.

[51] Edzer Pebesma. “Simple Features for R: Standardized Support for Spatial Vector Data”. In: *The R Journal* 10.1 (2018), pp. 439–446. DOI: 10.32614/RJ-2018-009. URL: https://doi.org/10.32614/RJ-2018-009.