Language classifications as standardized Newick phylogenetic trees with branch length

Dan Dediu (Dan.Dediu@mpi.nl), Max Planck Institute for Psycholinguistics, Nijmegen, The Netherlands

23 October 2015

Abstract: One of the best-known types of non-independence between languages is represented by genetic relationships due to descent from a common ancestor. While there are several classifications of languages into language families, each with its own advantages and disadvantages, they are relatively difficult to use by computational methods due to a lack of standardization. Moreover, certain advanced methods (such as phylogenetics) require not only the topology of the language family tree but also information concerning the amount of evolution that has happened on the tree represented as the branch lengths, and this information is usually missing. This paper presents a method that converts the language classifications provided by four widely-used databases (Ethnologue, WALS, AUTOTYP and Glottolog) into the *de facto* Newick standard, aligns the four most used conventions of unique identifiers for linguistic entities (ISO 639-3, WALS, AUTOTYP and Glottocode), and adds branch length information form a variety of sources (the tree's own topology, an externally given numeric constant or a distance matrix). The `R` scripts, input data and resulting Newick trees are provided in the GitHub repository https://github.com/ddediu/lgfam-newick in the hope that this will promote the use of advanced quantitative methods in answering questions concerning linguistic diversity and its temporal dynamics.

---

1 Introduction

Languages are not independent entities and the proper treatment of the various types of non-independence is crucial to drawing valid inferences (e.g., Ladd, Roberts, and Dediu 2015; S. Roberts and Winters 2013). One of the best-known types of non-independence is due to shared ancestry (Campbell and Poser 2008): the daughter languages tend to be more similar than expected due to the inheritance of characteristics from the mother language, similarity that tends to decrease with increasing temporal separation (this is also knwon as "Galton's problem" and applies more generally than linguistics; Mace and Pagel 1994). Such related languages descending from a common *proto-language* form a *language family*, the internal structure of which is usually represented as a tree. In such a tree, the attested, present-day or recent, languages form the *leaves* (or *terminal nodes*) of the tree and the *internal nodes* represent extinct, mostly unattested, languages[1].

Reliably identifying such *genetic relationships* is a complex problem (Campbell and Poser 2008; Bowern and Evans 2014) and many controversies exist, not only in what concerns the so-called "macro-families" but also in the composition and internal structure of more accepted language families. For example, disagreements might exist in the actual set of languages belonging to the same family, in the internal relationships between these languages (the tree *topology*) and the amount of change (the *branch lengths*); see the Figure 1 below.

There are three major difficulties facing modern quantitative methods that need to use such language classifications:

- the existence of several such classifications,
- the often non-standardized format these classifications are available in, and,

[1] Of course there are exceptions, such the inclusion of Latin – a well-attested extinct language – at the base of the Romance subfamily (e.g., W. Chang et al. 2015).

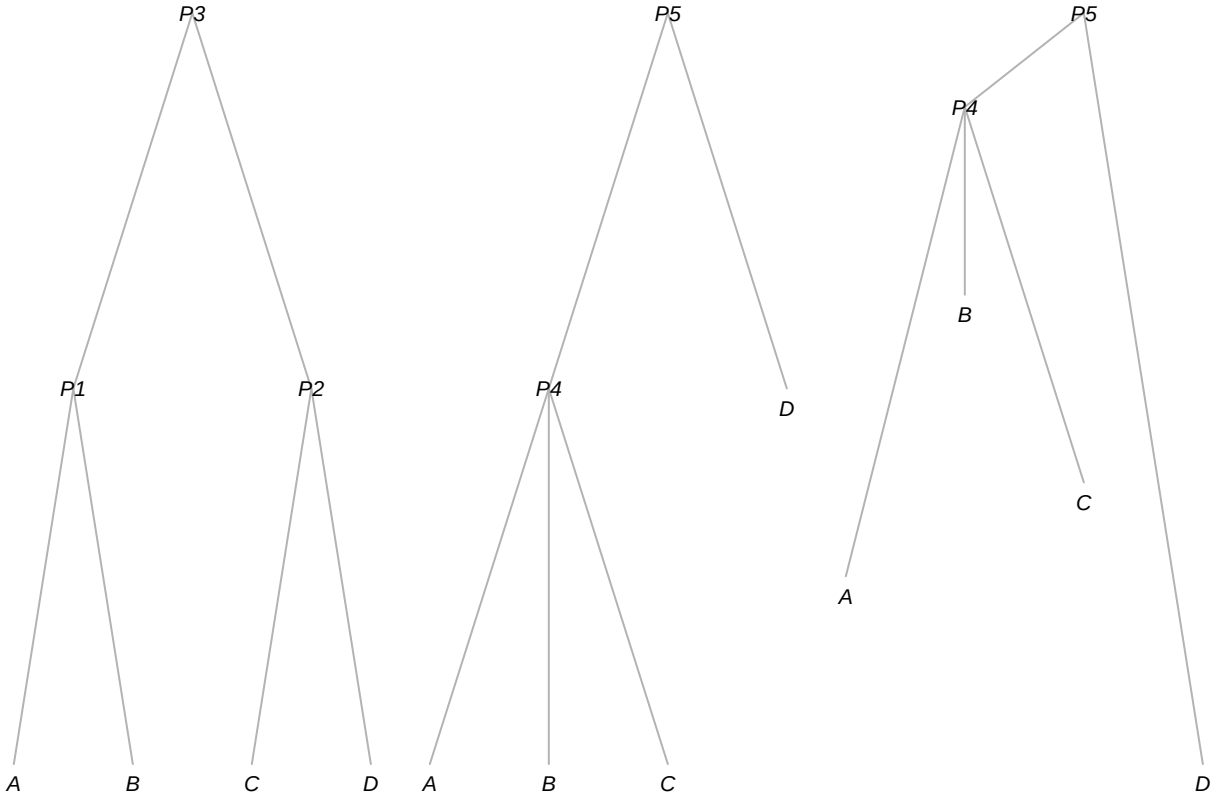

Figure 1: Three language families composed of the same four languages (A, B, C and D) but with different structures (left vs centre) and branch length (centre vs right). Time flows downwards from the proto-language at the top (*P3*, *P5* and *P5* respectively) towards the attested languages at the bottom. For example, in the leftmost tree languages A and B are more closely related than any is to language C. In the rightmost tree, language B has changed least since its most recent common ancestor (*P4*) with languages A and B.

- specifically for methods (e.g., phylogenetic) that take into account not only the toplogy of the tree but also the amount of change, the general absence of branch length estimates.

This paper offers a solution to these issues by proposing a standardized representation of language family trees from several classifications using the *de facto* standard *Newick* tree format[2], with added brach length estimates using multiple methods[3]. Here I briefly describe the data sources, methods and output formats, while the accompanying GitHub repository https://github.com/ddediu/lgfam-newick contains the actual primary data (wherever possible given their respective licensing terms), the `R` (R Core Team 2014) code and the resulting Newick language family trees with branch length.

2 Data, methods and outputs

The language family topologies are given by the following four widely used language classifications: the *Ethnologue* [denoted in the following as **E**; Lewis, Simons, and Fennig (2014); http://www.ethnologue.com/], the *World Atlas of Language Structures Online* [WALS, **W**; Dryer and Haspelmath (2013); http://wals.info/], *AUTOTYP* [**A**; Nichols, Witzlack-Makarevich, and Bickel (2013);http://www.autotyp.uzh.ch/] and *Glottolog* [**G**; Hammarström et al. (2014); http://glottolog.org/]. For each of these resources, I downloaded the raw data containing the language classifications and converted them to Newick trees without branch length information (resulting thuse in pure tree *topologies*).

2.1 Mapping between codes

However, before describing this transformation, it is important to discuss the issue of language *unique identifiers*. Currently, there are several methods for allocating unique (and hopefully also persistent) identifiers to linguistic entities (most often existing or recently extinct languages, but also dialects or proto-languages) and the mapping between these systems is far from a simple problem. Here, four standards are relevant: ISO 639-3 codes (tree letters, denoted in the follwing as **i**; http://www-01.sil.org/iso639-3), WALS codes (three letters, **w**; http://wals.info), AUTOTYP LIDs (numeric, **a**; http://www.autotyp.uzh.ch), and Glottocodes (alphanumeric: four letters followed by four digits, **g**; http://glottolog.org/glottolog/glottologinformation). As a first step, I mapped these codes for all the linguistic entities present in the four databases, a process made possible by the fact that some of these also give other codes besides their primary one for the linguistic entities therein (see Table 1) below.

Table 1: Codes present in the databases; most databases also give other codes besides their primary code. Legend for codes: **i** = ISO 639-3, **w** = WALS, **a** = AUTOTYP LID, and **g** = Glottocode.

Database	Primary code	Other codes
Ethnologue (**E**)	**i**	–
WALS (**W**)	**w**	**i g**
AUTOTYP (**A**)	**a**	**i g**
Glottolog (**G**)	**g**	**i**

This mapping is in the TAB-separated file `./output/code_mappings_iso_wals_autotyp_glottolog.csv` which gives for each unique linguistic entity (the rows) the corresponding ISO 639-3 code (column "ISO"),

[2]This format is described in http://evolution.genetics.washington.edu/phylip/newicktree.html.

[3]Even if for a few large families (including *Indo-European*, *Austronesian*, *Bantu* and *Uto-Aztecan*, with this list continuously growing) high-quality posterior samples of trees with branch length derived from cognacy judgments on the basic vocabulary (and even with calibaration data) using Bayesian phylogenetic methods (e.g. Bouckaert et al. 2012; Dunn et al. 2011) are available, this is currently not the case for the vast majority of the families.

WALS code (column "WALS"), AUTOTYP LID (column "AUTOTYP"), Glottocode (column "Glottolog"), the name as given by Ethnologue (column "Name.ethn"), by WALS (column "Name.wals"), by AUTOTYP (column "Name.autotyp") and by Glottolog (column "Name.glottolog"), the geographic coordinates (columns "Latitude" and "Longitude") in degrees as given by WALS and Glottolog[4], and, in the last column ("UULID") the *Universally Unique Language IDentifier* in the format "[i–**i**][w–**w**][a–**a**][g–**g**]" explained in detail in Section 2.3 The Newick trees and the naming convention.

2.2 Building the tree topologies

A second step is represented by the gathering of the raw data concerning the structure of the language families and exporting them as pure tree topologies in Newick format (without any branch length information). Each database poses its own challanges as each tends to use particular representations of the genetic relationships between languages. To standardize the process of topology extraction, conversion, exporting to and importing from file, I have written a collection of `R` (R Core Team 2014) types and functions (file `FamilyTrees.R`) which extend the *de facto* standard for representing phylogenetic trees in `R` as objects of class `phylo` (inmplemented in package `ape`; Paradis, Claude, and Strimmer 2004).

The list below summarizies the format of the raw data and its acquisition:

- **Ethnologue** (Lewis, Simons, and Fennig 2014) as opposed to the other three databases, the language classification data here is not provided in an easily downloadable form; instead, the Ethnologue website provides[5] (as of February 2015) a webpage (http://www.ethnologue.com/browse/families) containing a list with all the language families and links to their respective webpages (e.g., http://www.ethnologue.com/subgroups/afro-asiatic). These family webpages were further downloaded and parsed in order to extract the tree structure of the family, as well as the group names and the language names and ISO 639-3 codes[6];
- **WALS Online** (Dryer and Haspelmath 2013) provides the whole database (including language name, codes, geographic coordinates but also values for more than 130 typological features; http://wals.info/static/download/wals-language.csv.zip) under a Creative Commons Attribution-NonCommercial-NoDerivs 2.0 Germany (CC BY-NC-ND 2.0 DE; http://creativecommons.org/licenses/by-nc-nd/2.0/de/deed.en); here the important columns are WALS, ISO 639-3 and Glottolog codes, the languages' name, "genus" and "family", resulting in a rather flat three-levels structure;
- **AUTOTYP** (Nichols, Witzlack-Makarevich, and Bickel 2013) the AUTOTYP trees are freely available for download (http://www.autotyp.uzh.ch/available.html), use and distribution provided that their source is clearly cited; the format of the language families is similar to the WALS in the sense that each language (row) contains the language names, the AUTOTYP LID, the Glottolog and the ISO 639-3 codes, as well as the "stock", "mbranch", "sbranch", "ssbranch" and "lsbranch" names, each denoting more and more superficial levels (i.e., the "stock" is the highest level corresponding to the language family), and in some cases intermediate levels might be missing;
- **Glottolog** (Hammarström et al. 2014) as opposed to the other three databases, Glottlog provides the family trees already in a standardized Newick format (http://glottolog.org/static/trees/tree-glottolog-newick.txt) under a Creative Commons Attribution-ShareAlike 3.0 Unported License (CC BY-SA 3.0]; http://creativecommons.org/licenses/by-sa/3.0) license; here I only expanded the language codes with WALS and AUTOTYP.

The basic idea behind building the standardized tree topologies from these diverse formats[7] is to maintain a forest of (partially) built language family trees to which a new full path from a proto-language to a language is added. The algorithm first tries to identify an already present tree that contains the deeper part of the

[4]When there is a discrepancecy greater than 1° between the two, WALS wins.

[5]Under a set of conditions contained in the Terms of Use (www.ethnologue.com/terms-use) which allow "portions" of the data to be used for "research or educational purposes".

[6]The data used in this paper and included in the supplementary materials was harvested in Feburary 2015.

[7]Except for Glottolog, which provides a Newick format that requires only very light processing.

path (i.e., say adding "Indo-European → Germanic → North-West Germanic → English" would identify an already existing partial Indo-European tree) and, if so, adds the new (recent) part of the path to the tree. In this manner, the forest of all language families in the database is iteratively built from the ground up (Figure 2).

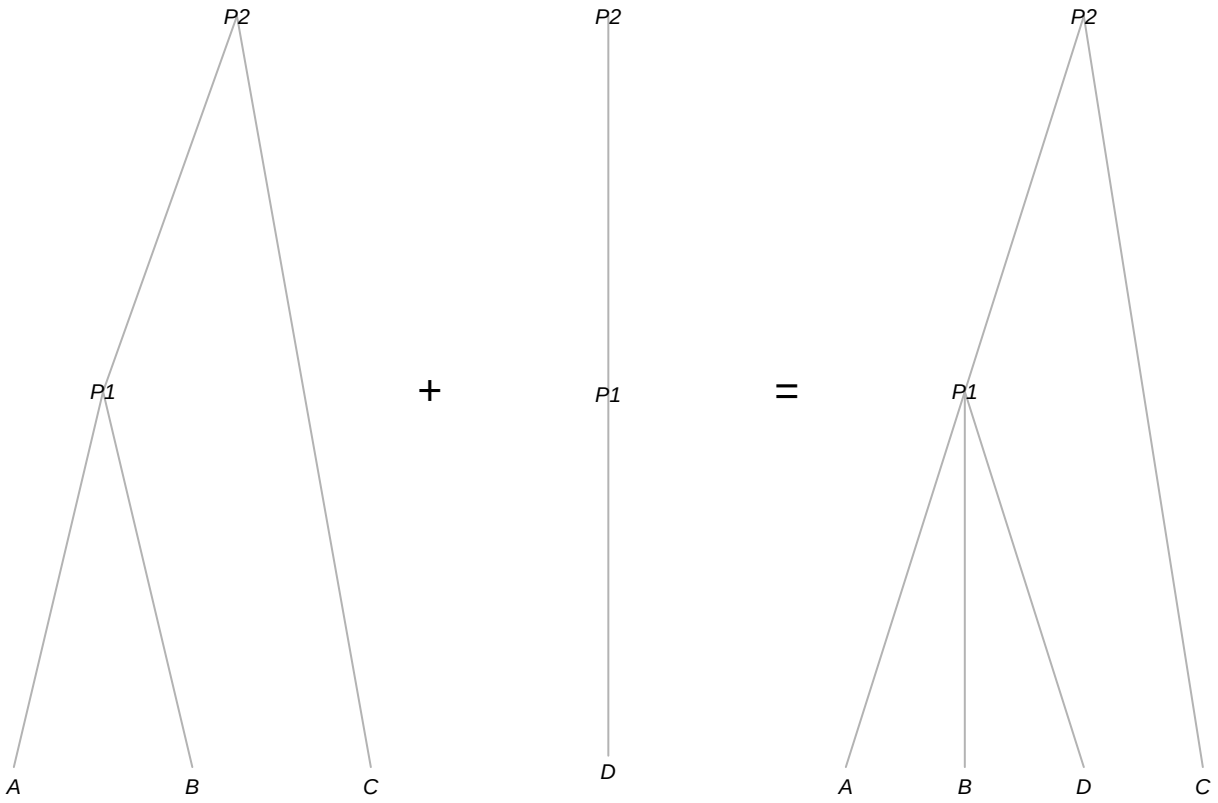

Figure 2: The leftmost partial family tree already exists in the forest when a new language *D* from subfamily *P1* in family *P2* (thus with full path *P2* → *P1* → *D*) is added, resulting in the rightmost tree.

Table 2 gives various summaries concerning the language family tree topologies successfully converted for each database.

Table 2: Various summaries concerning the topologies (no branch length) of the language family trees extracted from the four databases; **E** = Ethnologue, **W** = WALS, **A** = AUTOTYP, and **G** = Glottolog; '#' stands for 'number of...'; 'μ' is the 'average of...', 'm' is the 'minimum of...' and 'M' is the 'maximum of...'; 'lgs' stands for the leaves (or non-internal nodes) in the language family tree which are various types of lects (most often languages); 'lvl' represent the levels in the language family tree.

Class	**# trees**	**# lgs**	μ **lgs**	M **lgs**	m **lvl**	μ **lvl**	M **lvl**
E	147	7492	51.0	1545	3	4.8	16
W	214	2607	12.2	371	4	4.0	4
A	403	2926	7.3	340	3	3.4	7
G	435	15772	36.3	3254	3	4.5	20

2.3 The Newick trees and the naming convention

An interesting question concerns the format in which these tree topologies (and later, branch lengths) should be exported. I opted for the *de facto* standard Newick tree format[8] widely used in evolutionary biology, read and exported by many software packages and libraries, and able to represent rooted and unrooted trees, with our without leaf and internal node names, and with or without branch lengths. The basic idea is that subtrees are enclosed within parentheses "()" and the (optional) branch length is given as a number immediately following the branch and separated from it by ":". For example, the leftmost tree in Figure 2 can be represented as (language = leaf, proto-languages or groups = internal nodes, for simplicity all branches have the same length of 1):

Representation	Comments
(,,);	just the structure
(*A*,*B*),*C*);	with leaf names
(*A*,*B*)*P1*,*C*)*P2*;	with group names
(*A*:1,*B*:1),*C*:1);	with branch length
(*A*:1,*B*:1)*P1*:1,*C*:2)*P2*:1;	with everything

The language and group/proto-language names must include not only the actual name as given by the particular classification (which could very well differ between classifications; as a trivial example, Ethnologue calls the language with code ISO 639-3 "English" while Glottolog calls it "Standard English", but there are much more dramatic differences between the databases), but also the various unique identifiers this linguistic entity might have. Therefore, I opted for a standardized node name that follows the convention:

'NAME [i-I][w-W][a-A][g-G]'

where CAPITAL LETTERS denote variables and the full node name is usually included within single quotes. NAME is the entity name as given by the classification[9], followed by a SPACE and the four unique codes I (ISO 639-3), W (WALS), A (AUTOTYP) and G (Glottocode), where each and all can be missing or can have multiple values (in which case the values are separated by "-"). A few examples are (from the WALS classification, the Indo-European family):

- 'German {Zurich} [i-gsw][w-gzu][a-1305-1306-1307-1308-1309-1310][g-swis1247]'
- 'Urdu [i-urd][w-urd][a-2671][g-urdu1245]'
- 'Romani {Sepecides} [i-][w-rse][a-][g-]'
- 'Germanic [i-][w-][a-][g-]'.

2.4 The branch length methods

The methods I used to add branch lengths to the tree topologies can be divided into:

(a) methods that depend only on the topology: (1) constant, (2) proportional and (3) grafen,
(b) methods that generate the branch length and topology from a distance matrix: (4) nj, and
(c) methods that map a given distance matrix onto the topology: (5) nnls and (6) ga.

The methods of type (a) only need a tree topology T (and possibly a numeric constant $k > 0$). Method (1) computes branch lengths such that the sum of the branch lengths for every *root* → *leaf* path in the tree is

[8] See http://evolution.genetics.washington.edu/phylip/newicktree.html and http://en.wikipedia.org/wiki/Newick_format for details on the actual format.

[9] Given that some characters have a special meaning in the Newick format, I have enforced the following character substitutions: ,→. '→‘ (→{)→} TAB→SPACE :→| ;→| and characters with diacritics into their plain form (e.g., á→a and ã→a) while leaving unaltered the other characters.

equal to the constant k, meaning that the same amount of evolution k has happened on all branches. For example, for the leftmost tree in Figure 2 and k=1.0, the resulting tree is

$$((A : 0.667, B : 0.667)P1 : 0.333, C : 1)P2;$$

Method (2) simply gives each branch the same length k such that the amount of evolution on a path is proportional to the number of splits on that path; here the result is

$$((A : 1, B : 1)P1 : 1, C : 1)P2;$$

Method (3) is a reimplementation of Grafen (1989) whereby first each node is given a "height" defined as the number of leaves of its subtree minus 1 (0 for the leaves), after which branch lengths are computed as the difference between the height of the lower and the upper nodes of the branch; our tree is then:

$$((A : 1, B : 1)P1 : 1, C : 2)P2;$$

Method (4) is the only one of type (b) used here and is a clasic method in phylogenetics, the so-called "Neighbor-Joining" (or NJ) alorithm (Saitou and Nei 1987), essentially a clustering method that iteratively joins taxa into higher groupings (see http://en.wikipedia.org/wiki/Neighbor_joining for a good explanation). Given a language family topology T and a distance matrix between a set of languages D, I extract the languages in T and the submatrix of distances between them D_T (*NB*: it is possible that not for all pairs of languages there is a distance defined in D, resulting in a submatrix D_T with missing data for those pairs of languages), and then use NJ (as implemented by `R`'s function `njs()` in package `ape`; Paradis, Claude, and Strimmer 2004) to construct the corresponding phylogenetic tree. Thus, this method does not consider the actual topology in T but only the set of languages and the distances between them. For our example and the distance matrix (please note that the distances are given only between the languages – the leaves – and do not concern the proto-languages – the internal nodes)

$$D = \begin{array}{c|ccc} & A & B & C \\ A & 0 & 2.1 & 3.9 \\ B & 2.1 & 0 & 4.2 \\ C & 3.9 & 4.2 & 0 \end{array}$$

which approximates the distances between the three languages in the right-most tree of Figure 2 assuming method (1) with $k = 2.0$, we have the NJ tree

$$(C : 3, B : 1.2, A : 0.9);$$

It is important to note that NJ does not know anything about the internal structure of the original family tree (in this case about the *P1* internal node) and it might produce very different topologies from the ones given by the actual classifications.

Methods (5) and (6) try to use both the given language family's tree topology T and the information contained in the inter-language distance matrix D by computing branch lengths that best approximate the original distances in D (i.e., if one creates a new distance matrix between the languages D' by adding up the total branch lengths one needs to travel in the tree from one language to the other, then $D' \approx D$). Method (5) computes the branch lengths by using a non-negative least squares approach as implemented by `R`'s function `nnls.tree()` in package `phangorn` (Schliep 2011), resulting in this case in

$$((A : 1.05, B : 1.05)P1 : 0.975, C : 2.02)P2;$$

Finally, method (6) estimates the branch lengths using a standard genetic algorithm as implemented by `R`'s function `ga()` in package `GA` (Scrucca 2013). Given a topology T with n branches, I need to compute n real positive numbers, each representing the length of a branch in T such that the resulting distance matrix D' is a good approximation of the original distances D. In this genetic algorithm approach, I defined the "genome" as composed of n real-valued "genes", and the "fitness function" for a particular such genome $G = (g_1, g_2, \ldots, g_n)$ computes the SSE (sum of squared errors) between the original distances D and the current distances

D' between languages if the topology T had the branch lengths g_1, g_2, ... g_n. The genetic algorithm finds the best solution $G^* = (g^*_1, g^*_2, \ldots, g^*_n)$ that minimizes the fitness function (the SSE) using a population size of 100 individuals for at most 10,000 iterations (or when the fitness does not change for 100 iterations). For our example, some possible trees could be

$$((A : 0.899, B : 1.2)P1 : 1.43, C : 1.57)P2;$$

$$((A : 0.901, B : 1.2)P1 : 1.1, C : 1.9)P2;$$

$$((A : 0.902, B : 1.2)P1 : 1.61, C : 1.39)P2;$$

Please note that due to the random nature of the genetic algorithm and possibly the non-uniqueness of the solution (multiple optima), the best solution might vary between runs. Methods (5) and (6) have similar goals and produce very similar results, but approach them in different ways; method (5) is less robust than method (6) (it fails for certain topologies and distance matrices), while method (6) is much slower, especially for very large trees, and might produce non-unique solutions.

2.5 Topology preservation by restoring the collapsed single nodes

Methods (5) and (6) use internally functions from the `ape` package that do not currently deal well with so-called "single nodes"; these are internal nodes that have a single descendant in the tree such as node *P1* in the mid (degenerated) tree in Figure 2. These functions require that the single nodes have been removed from the argument(s) prior to their call and, to this end, `ape` provides the function `collapse.singles()` that takes a single phylogeny of class `phylo` and returns it with the single nodes collapsed (i.e., these nodes are removed and their single child is directly connected to their parent). Unfortunately, this means that for those language families that contain single nodes (which is the case quite often) the topology with branch length is not really the original topology anymore.

To address this problem, I extended the `collpase.nodes()` function to a pair of functions named `collapse.singles.reversible()` and `reverse.collapse.singles()` that ensure that the single nodes are removed prior to applying functions that cannot deal with them but are correctly added back in afterwards so that the result fully reflects the original topology. Briefly[10], `collapse.singles.reversible()` behaves exactly as `collapse.singles()` but returns not only the tree without single nodes but also information on how to restore these collapsed nodes. After the processing (in this context involving branch length computation), this information is used by `reverse.collapse.singles()` to insert back the removed single nodes into the tree, and the user can also specify how the new branch lengths are to be computed (keeping the original proportion, splitting the branch in two, or forcing the parent $\rightarrow$ single node branch to have length 0). Thus, by using this pair of functions, we ensure that the resulting tree with brach length preserves the original topology faithfully.

2.6 The distance matrices

There are many potentially meaningful distances between languages, and while the framework and `R` code introduced here can accomodate new ones, I have used here the following:

a. distances based on vocabulary: (1) ASJP16,
b. distances based on geography: (2) great-circle,
c. distances based on WALS: (3) gower and (4) euclidean, with and without missing data imputation,
d. distance based on AUTOTYP: (5) gower with missing data using only the variables with a single datapoint per language (this distance was computed by Balthasar Bickel), and
e. distances based on the tree topology: Maurits and Griffiths (2014)'s "genetic method" applied to the WALS (6), Ethnologue (7), Glottolog (8) and AUTOTYP (9) classifications.

[10] For more details on this process please see my blog post Tired of what ape's collapse.singles() does? (https://bitsaying.wordpress.com/2015/09/22/tired-of-what-apes-collapse-singles-does-heres-a-way-to-put-those-nodes-back/).

Method (1) uses the distances between languages provided by The Automated Similarity Judgment Program version 16 (ASJP16; Wichmann et al. 2013) and the ASJP software (version 2.1), freely available under a Creative Commons Attribution 3.0 (CC BY 3.0, http://creativecommons.org/licenses/by/3.0) license from the authors' website (http://asjp.clld.org). These distances are computed on the basis of standardized short wordlists transcribed in a reduced set of symbols using a normalized Levenstein distance (for details see Bakker et al. 2009). I further processed and converted this database into a distance matrix between languages using ISO 639-3 codes as language identifiers[11], resulting in a 3932 × 3932 matrix with no missing data.

Method (2) computes the geographic (great circle) distances between the languages' geographic coordinates using `R`'s function `distm()` in package `geosphere` (Hijmans 2014), resulting in a 7494 × 7494 matrix with no missing data.

Methods (3) and (4) use the WALS typological database to compute distances between languages using their feature values. I used the methods implemented by `R`'s function `daisy()` in package `cluster` (Maechler et al. 2015), namely "gower" (method 3; Gower 1971) which standardizes each feature to the [0,1] interval, and "euclidean" (method 4) which computes the standard Euclidean distance in an n-dimensional real space. Given the enormously high proportion of missing data in the WALS database (85.1% cells), I have computed these distances also doing a simple missing data imputation whereby the missing data was replaced by the mode (i.e., the most frequent value) of the corresponding typological variable. With these, I obtained the follwing distance matrices: gower with missing data (2679 × 2679, 48.9% missing data cells), gower with missing data imputation (2679 × 2679, no missing data), euclidean with missing data (2679 × 2679, 48.9% missing data cells), and euclidean with missing data imputation (2679 × 2679, no missing data).

Method (5) uses the AUTOTYP typological database to compute distances between languages using their feature values. This method also uses `R`'s function `daisy()` in package `cluster` (Maechler et al. 2015) with argument "gower" (Gower 1971), without missing data imputation, resulting in a 2928 × 2928 distance matrix with 57.6% missing data cells.

Methods (6) – (9) use the "genetic method" described in Maurits and Griffiths (2014) whereby branch lengths are computed from the topology of the family tree so that languages sharing k intermediate nodes on their paths from the root are separated by the distance $d = M - \sum_{i=1}^{n} \alpha^i$ (with M being the maximum possible distance and α being empirically fixed at 0.69); it is important to note that by defintion these distances cannot be computed for pairs of languages beloging to different families, but are defined for any pair of languages that belong to the same family (therefore the percent of missing data is uninformative here). I reimplemented this method in `R`[12] and applied it to each of the four classifications, resulting in the follwing distance matrices: MG2015 using the WALS classification (2607 × 2607), the Ethnologue classification (7492 × 7492), the Glottolog classification (15772 × 15772), and the AUTOTYP classification (2926 × 2926).

2.7 The family trees with branch length

Thus, for each combination of *classification* (Ethnologue, WALS, AUTOTYP, Glottolog) by *method* (no branch length, constant, proportional, grafen, nj, nnls, ga) and, for the last three methods, also by *distance* matrix (asjp16, great-circle, wals-gower, wals-gower+imputation, wals-euclidean, wals-euclidean+imputation, autotyp-gower, mg2015+wals, mg2015+ethnologue, mg2015+glottolog, mg2015+autotyp[13]), I produced a set of phylogenetic trees in Newick format as described above. Each of these sets was saved in two formats: a TAB-separated CSV file, and a Nexus file, containing essentially the same information.

The first format is a standard TAB-separated CSV file with a standardized name of the form `CLASSIFICATION-newick-METHOD&PARAMETERS.csv` (e.g., `autotyp-newick-nj+autotyp.csv` and `glottolog-newick-nnls+wals(gower,mode).csv`) in the directory `./output/CLASSIFICATION/`. It

[11] This conversion required limited manual editing including the replacement of some non-ASCII characters in the language descriptors and some of the 26-character language identifiers exported by the ASJP v2.1 software.

[12] Many thanks to Luke Maurits for helping to clarify the inner workings of the method.

[13] Please note that given that MG2015 is computed using only the tree topologies, it makes little sense (as far as I can tell) to use the MG2015 computed on one classification for branch length computation on another classification; therefore only mg2015+wals was used for **W**, mg2015+ethnologue for **E**, mg2015+autotyp for **A**, and mg2015+glottolog for **G**.

contains the language families (one per row, except the first row which is the header), and for each family it gives the family name (as defined by the classification), the success or failure of the method, some relevant comments (for examply, why the methods has failed), and the actual tree in Newick format (or an empty string "" if the method has failed).

The second format is a standard Nexus file (D. R. Maddison, Swofford, and Maddison 1997) with a standardized name of the form `CLASSIFICATION-nexus-METHOD\&PARAMETERS.nex` (e.g., `autotyp-nexus-nj+autotyp.nex` and `glottolog-nexus-nnls+wals(gower,mode).nex`) in the `./output/CLASSIFICATION/` directory. These Nexus files contain only the `trees` block with the `translate` list[14] and the named trees (the language family names as given by the classification) in Newick format.

Summaries about these trees are given in Appendix A.

To explore the (dis)agreement between the trees produced by these methods, for each language family (within a given classification as families do not generally mean the same thing across classifications) and pair of methods, I computed two distances[15] between these corresponding trees: one that considers only how similar they are in their topology ("PH85"; Penny and Hendy 1985; Rzhetsky and Nei 1992) and another that also takes into account the branch length ("score"; Kuhner and Felsenstein 1994). For details please see Appendix B.

3 Conclusions

This paper describes a flexible method for producing standardized language family trees in the Newick format with branch length using a variety of linguistic classifications, methods and distances between languages. Accompanying this paper a Github https://github.com/ddediu/lgfam-newick repository containing (where possible given the licensing terms) the input data, the `R` code, the output files, and the source of this paper (written using `R Markdown` Allaire et al. 2015), as well as short descriptions (`ReadMe.txt` files) and license terms. My own `R` code is released under a GPL v2 license, is relatively well-commented and tested, and is free to use and modify as long as the terms of the license are respected and this paper is cited[16]. Especially the high-level functions in `./code/FamilyTrees.R` might be useful for manipulating such trees and applying new distance matrices to family tree topologies; the file `./code/StandardizedTrees.R` can be consulted as an example of using them and it also contains some useful high-level functions. Updates and bug fixes will be posted on the Github https://github.com/ddediu/lgfam-newick repository.

4 Acknowledgments

Many thanks to the authors of the databases used here for making their data freely available, to Balthasar Bickel for computing the AUTOTYP distance and agreeing to making it freely available with this paper, to Luke Maurits for clariying their "genetic method", and to Harald Hammarström and Seán Roberts for discussions and feedback. During this project I was funded by an NWO (Netherlands Organisation for Scientific Research) VIDI grant number 016.124.315.

5 Appendix A: Family trees summaries

This Appendix contains summaries concerning the language family trees generated using the classifications, methods and distances discussed in this paper, split into four Tables by classification used.

[14] The `R` script is capable of generating or not the `translate` list, by default it does so in order to increase human readablity and make the files importable by some phylogenetic software.

[15] As implemented by `R`'s function `dist.topo()` in package `ape` (Paradis, Claude, and Strimmer 2004).

[16] Why not an `R` package, you migth ask? I feel this application is very specific and the code is mainly intended to be changed and adapted (or just serve as inspiration) for other specific problems the users might have, instead of being used as it is.

Table 4: Various summaries concerning the topologies (no branch length) of the language family trees extracted from the ethnologue database; the methods are constant, proportional, Grafen, nj, nnls and ga with the appropriate parameter (k or the distance matrix name shortened to 'asjp' [asjp16] ,'geo' [geo] ,'w:g' [wals(gower)] ,'w:e' [wals(euclidean)] ,'w:gm' [wals(gower,mode)] ,'w:em' [wals(euclidean,mode)] ,'auto' [autotyp] ,'m:e' [mg2015(ethnologue)]); '#' stands for 'number of...'; 'μ' is the 'average of...', 'm' is the 'minimum of...' and 'M' is the 'maximum of...'; 'lgs' stands for the leaves (or non-internal nodes) in the language family tree which are various types of lects (most often languages); 'lvl' represent the levels in the language family tree.

Method	**Param**	**# trees**	**# lgs**	μ **lgs**	M **lgs**	m **lvl**	μ **lvl**	M **lvl**
const	k=1.0	147	7492	51.0	1545	3	4.8	16
prop	k=1.0	147	7492	51.0	1545	3	4.8	16
grafen	-	147	7492	51.0	1545	3	4.8	16
nj	*asjp*	147	3810	46.5	789	2	7.4	33
nj	*geo*	147	7124	69.8	1510	2	17.2	220
nj	*w:g*	147	449	15.5	81	2	6.7	22
nj	*w:e*	147	449	15.5	81	2	6.2	21
nj	*w:gm*	147	2231	29.4	337	2	11.0	64
nj	*w:em*	147	2231	29.4	337	2	11.6	64
nj	*auto*	147	194	9.7	49	2	5.2	15
nj	*m:e*	147	7419	71.3	1545	2	15.8	129
nnls	*asjp*	147	3846	38.5	789	2	4.4	15
nnls	*geo*	147	7184	54.4	1510	2	4.7	15
nnls	*w:g*	147	551	9.0	58	2	3.6	7
nnls	*w:e*	147	551	9.0	58	2	3.6	7
nnls	*w:gm*	147	2273	23.4	337	2	4.3	15
nnls	*w:em*	147	2273	23.4	337	2	4.3	15
nnls	*auto*	147	476	8.5	63	2	3.5	7
nnls	*m:e*	147	7479	55.8	1545	3	4.9	16
ga	*asjp*	147	3846	38.5	789	2	4.4	15
ga	*geo*	147	7184	54.4	1510	2	4.7	15
ga	*w:g*	147	2247	26.1	337	2	4.5	15
ga	*w:e*	147	2247	26.1	337	2	4.5	15
ga	*w:gm*	147	2273	23.4	337	2	4.3	15
ga	*w:em*	147	2273	23.4	337	2	4.3	15
ga	*auto*	147	2439	28.7	369	2	4.6	15
ga	*m:e*	147	7479	55.8	1545	3	4.9	16

Table 5: Various summaries concerning the topologies (no branch length) of the language family trees extracted from the wals database; the methods are constant, proportional, Grafen, nj, nnls and ga with the appropriate parameter (k or the distance matrix name shortened to 'asjp' [asjp16] ,'geo' [geo] ,'w:g' [wals(gower)] ,'w:e' [wals(euclidean)] ,'w:gm' [wals(gower,mode)] ,'w:em' [wals(euclidean,mode)] ,'auto' [autotyp] ,'m:w' [mg2015(wals)]); '#' stands for 'number of...'; 'μ' is the 'average of...', 'm' is the 'minimum of...' and 'M' is the 'maximum of...'; 'lgs' stands for the leaves (or non-internal nodes) in the language family tree which are various types of lects (most often languages); 'lvl' represent the levels in the language family tree.

Method	**Param**	**# trees**	**# lgs**	μ **lgs**	M **lgs**	m **lvl**	μ **lvl**	M **lvl**
const	k=1.0	214	2607	12.2	371	4	4.0	4
prop	k=1.0	214	2607	12.2	371	4	4.0	4
grafen	-	214	2607	12.2	371	4	4.0	4
nj	*asjp*	214	1973	28.2	297	2	7.1	20
nj	*geo*	214	2425	30.7	370	2	11.5	67
nj	*w:g*	214	276	11.0	65	2	5.3	21
nj	*w:e*	214	276	11.0	65	2	4.9	15
nj	*w:gm*	214	2442	30.9	371	2	11.2	66
nj	*w:em*	214	2442	30.9	371	2	12.0	88
nj	*auto*	214	255	9.4	60	2	5.3	20
nj	*m:w*	214	2442	30.9	371	2	14.7	163
nnls	*asjp*	214	2015	22.1	297	2	3.4	4
nnls	*geo*	214	2483	23.0	370	2	3.9	4
nnls	*w:g*	214	1259	14.5	176	4	4.0	4
nnls	*w:e*	214	1259	14.5	176	4	4.0	4
nnls	*w:gm*	214	2502	23.0	371	4	4.0	4
nnls	*w:em*	214	2502	23.0	371	4	4.0	4
nnls	*auto*	214	1152	13.9	159	2	3.6	4
nnls	*m:w*	214	2502	23.0	371	4	4.0	4
ga	*asjp*	214	1999	22.7	297	2	3.4	4
ga	*geo*	214	2481	23.2	370	2	3.9	4
ga	*w:g*	214	2441	29.4	371	4	4.0	4
ga	*w:e*	214	2441	29.4	371	4	4.0	4
ga	*w:gm*	214	2502	23.0	371	4	4.0	4
ga	*w:em*	214	2502	23.0	371	4	4.0	4
ga	*auto*	214	2234	29.0	344	2	3.5	4
ga	*m:w*	214	2502	23.0	371	4	4.0	4

Table 6: Various summaries concerning the topologies (no branch length) of the language family trees extracted from the autotyp database; the methods are constant, proportional, Grafen, nj, nnls and ga with the appropriate parameter (k or the distance matrix name shortened to 'asjp' [asjp16] ,'geo' [geo] ,'w:g' [wals(gower)] ,'w:e' [wals(euclidean)] ,'w:gm' [wals(gower,mode)] ,'w:em' [wals(euclidean,mode)] ,'auto' [autotyp] ,'m:a' [mg2015(autotyp)]); '#' stands for 'number of...'; 'μ' is the 'average of...', 'm' is the 'minimum of...' and 'M' is the 'maximum of...'; 'lgs' stands for the leaves (or non-internal nodes) in the language family tree which are various types of lects (most often languages); 'lvl' represent the levels in the language family tree.

Method	**Param**	**# trees**	**# lgs**	μ **lgs**	M **lgs**	m **lvl**	μ **lvl**	M **lvl**
const	k=1.0	403	2926	7.3	340	3	3.4	7
prop	k=1.0	403	2926	7.3	340	3	3.4	7
grafen	-	403	2926	7.3	340	3	3.4	7
nj	*asjp*	403	2035	18.8	306	2	6.3	23
nj	*geo*	403	2547	19.9	337	2	8.6	69
nj	*w:g*	403	635	12.2	175	2	5.6	46
nj	*w:e*	403	635	12.2	175	2	5.1	30
nj	*w:gm*	403	2229	18.4	314	2	8.5	50
nj	*w:em*	403	2229	18.4	314	2	8.8	59
nj	*auto*	403	198	6.6	36	2	3.9	12
nj	*m:a*	403	2605	20.4	340	2	10.1	162
nnls	*asjp*	403	2107	14.6	306	2	3.3	7
nnls	*geo*	403	2635	15.3	337	2	3.7	7
nnls	*w:g*	403	1264	9.4	185	2	3.2	7
nnls	*w:e*	403	1264	9.4	185	2	3.2	7
nnls	*w:gm*	403	2319	14.0	314	2	3.3	7
nnls	*w:em*	403	2319	14.0	314	2	3.3	7
nnls	*auto*	403	1485	10.8	203	3	3.8	7
nnls	*m:a*	403	2697	15.5	340	3	3.9	7
ga	*asjp*	403	2091	14.8	306	2	3.3	7
ga	*geo*	403	2619	15.5	337	2	3.8	7
ga	*w:g*	403	2232	16.8	314	2	3.5	7
ga	*w:e*	403	2232	16.8	314	2	3.5	7
ga	*w:gm*	403	2299	14.2	314	2	3.4	7
ga	*w:em*	403	2299	14.2	314	2	3.4	7
ga	*auto*	403	2611	18.9	340	3	4.1	7
ga	*m:a*	403	2697	15.5	340	3	3.9	7

Table 7: Various summaries concerning the topologies (no branch length) of the language family trees extracted from the glottolog database; the methods are constant, proportional, Grafen, nj, nnls and ga with the appropriate parameter (k or the distance matrix name shortened to 'asjp' [asjp16] ,'geo' [geo] ,'w:g' [wals(gower)] ,'w:e' [wals(euclidean)] ,'w:gm' [wals(gower,mode)] ,'w:em' [wals(euclidean,mode)] ,'auto' [autotyp] ,'m:g' [mg2015(glottolog)]); '#' stands for 'number of...'; 'μ' is the 'average of...', 'm' is the 'minimum of...' and 'M' is the 'maximum of...'; 'lgs' stands for the leaves (or non-internal nodes) in the language family tree which are various types of lects (most often languages); 'lvl' represent the levels in the language family tree.

Method	**Param**	**# trees**	**# lgs**	μ **lgs**	M **lgs**	m **lvl**	μ **lvl**	M **lvl**
const	k=1.0	435	15772	36.3	3254	3	4.5	20
prop	k=1.0	435	15772	36.3	3254	3	4.5	20
grafen	-	435	15772	36.3	3254	3	4.5	20
nj	*asjp*	435	1926	22.4	430	2	5.6	26
nj	*geo*	435	4501	31.9	830	2	10.4	163
nj	*w:g*	435	197	6.8	29	2	4.4	21
nj	*w:e*	435	197	6.8	29	2	4.2	13
nj	*w:gm*	435	945	14.8	128	2	7.3	28
nj	*w:em*	435	945	14.8	128	2	8.0	42
nj	*auto*	435	154	5.3	12	2	3.7	10
nj	*m:g*	435	15507	69.9	3254	2	11.1	77
nnls	*asjp*	435	2000	16.3	430	2	4.0	15
nnls	*geo*	435	4605	23.9	830	2	4.1	17
nnls	*w:g*	435	322	4.9	29	2	3.1	7
nnls	*w:e*	435	322	4.9	29	2	3.1	7
nnls	*w:gm*	435	1019	10.1	128	2	3.7	12
nnls	*w:em*	435	1019	10.1	128	2	3.7	12
nnls	*auto*	435	278	4.6	26	2	3.3	7
nnls	*m:g*	435	15611	57.0	3254	3	5.4	20
ga	*asjp*	435	2000	16.3	430	2	4.0	15
ga	*geo*	435	4605	23.9	830	2	4.1	17
ga	*w:g*	435	966	12.5	128	2	4.1	12
ga	*w:e*	435	966	12.5	128	2	4.1	12
ga	*w:gm*	435	1012	10.2	128	2	3.7	12
ga	*w:em*	435	1012	10.2	128	2	3.7	12
ga	*auto*	435	1068	13.7	139	2	4.4	12
ga	*m:g*	435	15611	57.0	3254	3	5.4	20

6 Appendix B: Distances between methods

Across all classifications and methods the distances between corresponding trees are distributed as in Figure 3: "PH85" varies between 0.00 and 4044.00, with a mean of 31.86 (and median 5.00), while "score" varies between 0.00 and 14067.33, with a mean of 31.62 (and median 0.66).

Table 8 presents the summaries (minimum, median, mean and maximum) for the "PH85" and "score" distances between the corresponding language family trees across all four classifications. This table does not show all possible pairs of methods (the full data at the level of the language family is given in the TAB-separated file

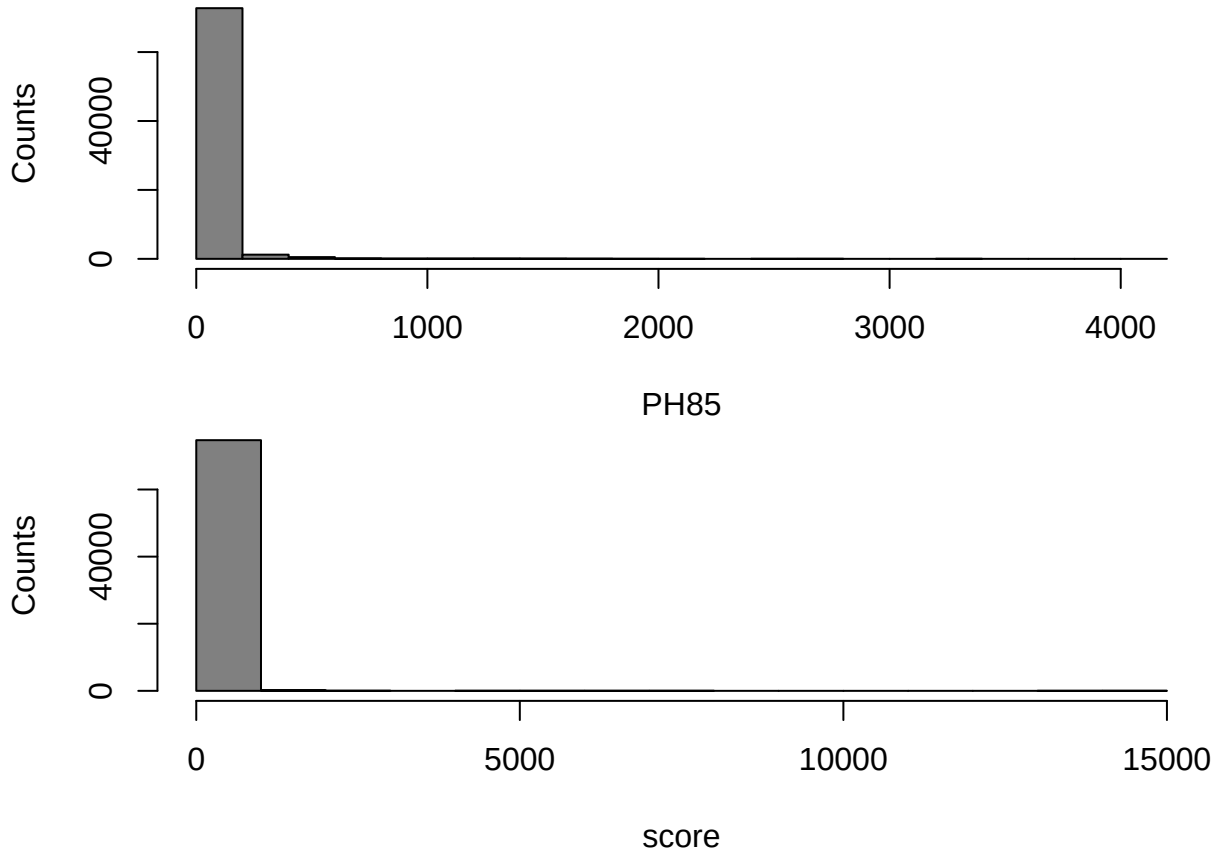

Figure 3: The distribution of distances between corresponding family trees for different pairs of methods across the whole dataset. 'PH85' (top) takes only the tree topology into account, while 'score' (bottom) also the branch lengths.

`./output/tree_comparisons_between_methods.csv`) but only the comparisons between:

- the methods *constant*, *proportional* and *grafen*,
- "constant" versus *nj* (all distances),
- all the distances for method *nj*,
- all the distances for method *nnls*,
- all the distances for method *ga*,
- the corresponding distances between the methods *nnls*, *nj* and *ga*.

Table 8: Selected pairs of methods (**Met$_1$** and **Met$_2$**) with their parameters (**Par$_1$** and **Par$_2$**) for all classifications. methods *constant* and *proportional* are shortneded to 'const' and 'prop', while the distance names are shortneed as described above. As opposed to the 'PH85' (**PH**) distances, the 'score' (**sco**) comparisons involving the *geo* distance are so large due to the measurement scale (kilometers). As above, 'μ' is the 'average of...', 'm' is the 'minimum of...' and 'M' is the 'maximum of...', with $\tilde{m}$ being the 'median of...'

Met$_1$	**Par$_1$**	**Met$_2$**	**Par$_2$**	m **PH**	$\tilde{m}$ **PH**	μ **PH**	M **PH**	m **sco**	$\tilde{m}$ **sco**	μ **sco**	M **sco**
const	*1*	**prop**	*1*	0.00	0.00	0.00	0.00	0.75	1.30	2.12	35.25
const	*1*	**grafen**	-	0.00	0.00	0.00	0.00	0.25	3.89	149.11	14066.81
prop	*1*	**grafen**	-	0.00	0.00	0.00	0.00	0.00	3.00	148.19	14059.41
const	*1*	**nj**	*asjp*	0.00	18.00	69.16	1682.00	0.14	0.53	0.62	2.58
const	*1*	**nj**	*w:g*	0.00	13.00	24.30	210.00	0.25	0.69	0.82	4.91
const	*1*	**nj**	*w:gm*	0.00	19.00	66.91	1489.00	0.25	0.50	0.57	2.23
const	*1*	**nj**	*w:e*	0.00	13.00	24.35	212.00	0.25	23.00	29.79	179.73
const	*1*	**nj**	*w:em*	0.00	19.00	66.95	1489.00	0.25	18.60	31.50	394.65
const	*1*	**nj**	*auto*	0.00	10.00	15.09	80.00	0.25	0.52	0.63	1.93
const	*1*	**nj**	*geo*	0.00	15.00	76.64	2444.00	0.25	0.44	0.53	2.22
const	*1*	**nj**	*m:w*	0.00	10.00	32.53	344.00	0.11	0.25	0.36	1.12
const	*1*	**nj**	*m:e*	0.00	8.00	68.48	1637.00	0.11	0.34	0.40	1.91
const	*1*	**nj**	*m:g*	0.00	6.00	50.46	2012.00	0.11	0.26	0.35	3.82
const	*1*	**nj**	*m:a*	0.00	10.00	26.52	323.00	0.11	0.29	0.31	1.13
nj	*asjp*	**nj**	*w:g*	0.00	11.00	25.26	281.00	0.00	0.48	0.67	4.80
nj	*asjp*	**nj**	*w:gm*	0.00	17.00	59.58	1115.00	0.00	0.24	0.30	1.48
nj	*asjp*	**nj**	*w:e*	0.00	10.50	25.63	295.00	0.00	23.35	30.16	179.61
nj	*asjp*	**nj**	*w:em*	0.00	17.00	59.63	1115.00	0.00	18.45	31.48	394.65
nj	*asjp*	**nj**	*auto*	0.00	8.00	14.06	109.00	0.00	0.38	0.49	1.87
nj	*asjp*	**nj**	*geo*	0.00	18.00	78.06	2252.00	0.00	0.22	0.29	1.47
nj	*asjp*	**nj**	*m:w*	0.00	19.00	60.89	615.00	0.00	0.33	0.39	1.57
nj	*asjp*	**nj**	*m:e*	0.00	25.00	135.52	2209.00	0.00	0.33	0.40	1.78
nj	*asjp*	**nj**	*m:g*	0.00	44.50	230.59	3550.00	0.01	0.40	0.65	4.90
nj	*asjp*	**nj**	*m:a*	0.00	17.50	42.02	632.00	0.00	0.35	0.37	1.31
nj	*w:g*	**nj**	*w:gm*	0.00	8.00	21.45	308.00	0.00	0.41	0.60	4.83
nj	*w:g*	**nj**	*w:e*	0.00	4.00	16.77	298.00	0.00	22.67	28.58	179.11
nj	*w:g*	**nj**	*w:em*	0.00	8.00	21.94	314.00	0.00	17.69	26.16	220.05
nj	*w:g*	**nj**	*auto*	0.00	2.50	11.60	119.00	0.00	0.27	0.46	2.62
nj	*w:g*	**nj**	*geo*	0.00	10.50	30.87	347.00	0.00	0.45	0.61	4.85
nj	*w:g*	**nj**	*m:w*	0.00	10.00	23.06	124.00	0.02	0.41	0.65	1.96
nj	*w:g*	**nj**	*m:e*	2.00	21.00	50.22	279.00	0.05	0.54	0.77	2.14
nj	*w:g*	**nj**	*m:g*	2.00	20.00	48.81	179.00	0.11	0.62	0.54	1.37
nj	*w:g*	**nj**	*m:a*	0.00	10.00	31.00	393.00	0.00	0.56	0.72	4.87

$\mathbf{Met_1}$	$\mathbf{Par_1}$	$\mathbf{Met_2}$	$\mathbf{Par_2}$	m **PH**	$\tilde{m}$ **PH**	μ **PH**	M **PH**	m **sco**	$\tilde{m}$ **sco**	μ **sco**	M **sco**
nj	*w:gm*	**nj**	*w:e*	0.00	7.00	22.04	316.00	0.00	22.98	28.85	179.71
nj	*w:gm*	**nj**	*w:em*	0.00	8.00	36.78	576.00	0.00	18.06	30.88	394.65
nj	*w:gm*	**nj**	*auto*	0.00	6.00	12.74	117.00	0.00	0.32	0.42	1.80
nj	*w:gm*	**nj**	*geo*	0.00	22.00	78.20	1839.00	0.00	0.05	0.07	0.43
nj	*w:gm*	**nj**	*m:w*	0.00	20.00	68.06	728.00	0.00	0.21	0.23	0.83
nj	*w:gm*	**nj**	*m:e*	2.00	37.00	135.64	1876.00	0.01	0.32	0.37	1.42
nj	*w:gm*	**nj**	*m:g*	2.00	65.00	275.16	3363.00	0.02	0.46	0.71	4.83
nj	*w:gm*	**nj**	*m:a*	0.00	17.50	43.13	648.00	0.00	0.22	0.21	0.74
nj	*w:e*	**nj**	*w:em*	0.00	7.00	21.79	320.00	0.00	28.39	38.02	207.90
nj	*w:e*	**nj**	*auto*	0.00	4.50	12.37	121.00	0.00	20.56	20.66	73.42
nj	*w:e*	**nj**	*geo*	0.00	10.50	30.91	357.00	0.00	23.00	28.86	179.74
nj	*w:e*	**nj**	*m:w*	0.00	8.00	22.71	122.00	1.46	24.33	29.45	73.40
nj	*w:e*	**nj**	*m:e*	2.00	21.00	50.30	279.00	0.05	23.82	33.30	103.59
nj	*w:e*	**nj**	*m:g*	2.00	20.00	48.81	179.00	0.30	16.78	18.49	68.48
nj	*w:e*	**nj**	*m:a*	0.00	10.00	31.30	395.00	0.00	23.82	32.08	179.74
nj	*w:em*	**nj**	*auto*	0.00	6.00	12.69	117.00	0.00	16.30	16.16	58.58
nj	*w:em*	**nj**	*geo*	0.00	21.00	78.18	1841.00	0.00	18.08	30.91	394.65
nj	*w:em*	**nj**	*m:w*	0.00	20.00	68.22	732.00	0.00	19.99	34.93	394.65
nj	*w:em*	**nj**	*m:e*	2.00	35.00	135.76	1878.00	0.28	16.30	28.49	163.67
nj	*w:em*	**nj**	*m:g*	2.00	65.00	275.20	3363.00	0.30	13.82	23.41	157.25
nj	*w:em*	**nj**	*m:a*	0.00	17.50	43.13	646.00	0.00	19.61	33.97	220.37
nj	*auto*	**nj**	*geo*	0.00	8.00	15.80	118.00	0.00	0.32	0.42	1.80
nj	*auto*	**nj**	*m:w*	0.00	7.00	17.30	119.00	0.00	0.32	0.49	1.82
nj	*auto*	**nj**	*m:e*	2.00	10.00	22.44	110.00	0.08	0.46	0.58	1.85
nj	*auto*	**nj**	*m:g*	2.00	20.00	36.50	147.00	0.05	0.42	0.47	0.87
nj	*auto*	**nj**	*m:a*	2.00	8.00	14.00	64.00	0.08	0.41	0.45	1.14
nj	*geo*	**nj**	*m:w*	0.00	19.50	65.64	717.00	0.00	0.20	0.22	0.82
nj	*geo*	**nj**	*m:e*	0.00	18.00	145.14	3003.00	0.00	0.29	0.31	1.44
nj	*geo*	**nj**	*m:g*	0.00	26.00	169.42	4044.00	0.00	0.33	0.47	4.83
nj	*geo*	**nj**	*m:a*	0.00	15.00	41.57	663.00	0.00	0.16	0.19	0.74
nnls	*asjp*	**nnls**	*w:g*	0.00	3.50	4.66	32.00	0.00	0.12	0.16	0.78
nnls	*asjp*	**nnls**	*w:gm*	0.00	6.00	17.74	455.00	0.00	0.16	0.23	1.33
nnls	*asjp*	**nnls**	*w:e*	0.00	3.50	4.66	32.00	0.00	0.20	2.34	72.01
nnls	*asjp*	**nnls**	*w:em*	0.00	6.00	17.74	455.00	0.00	16.50	23.66	199.29
nnls	*asjp*	**nnls**	*auto*	0.00	3.00	4.42	36.00	0.00	0.11	0.16	0.58
nnls	*asjp*	**nnls**	*geo*	0.00	6.00	22.22	570.00	0.00	0.17	0.23	1.32
nnls	*asjp*	**nnls**	*m:w*	0.00	0.00	4.65	37.00	0.00	0.12	0.20	0.85
nnls	*asjp*	**nnls**	*m:e*	0.00	10.00	55.25	1109.00	0.00	0.34	0.43	1.65
nnls	*asjp*	**nnls**	*m:g*	0.00	25.00	102.48	1525.00	0.02	0.41	0.50	1.60
nnls	*asjp*	**nnls**	*m:a*	0.00	6.00	9.64	73.00	0.00	0.28	0.31	1.04
nnls	*w:g*	**nnls**	*w:gm*	0.00	0.00	0.00	0.00	0.00	0.03	0.05	0.56
nnls	*w:g*	**nnls**	*w:e*	0.00	0.00	0.00	0.00	0.00	0.00	2.04	50.47
nnls	*w:g*	**nnls**	*w:em*	0.00	0.00	0.00	0.00	0.00	10.40	19.71	460.18
nnls	*w:g*	**nnls**	*auto*	0.00	0.00	2.44	21.00	0.00	0.00	0.05	0.57
nnls	*w:g*	**nnls**	*geo*	0.00	0.00	4.09	39.00	0.00	0.01	0.06	1.00
nnls	*w:g*	**nnls**	*m:w*	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.51
nnls	*w:g*	**nnls**	*m:e*	0.00	6.00	9.38	45.00	0.00	0.25	0.27	1.01
nnls	*w:g*	**nnls**	*m:g*	0.00	18.00	23.16	76.00	0.05	0.30	0.34	0.68
nnls	*w:g*	**nnls**	*m:a*	0.00	3.00	4.41	18.00	0.00	0.20	0.21	0.51
nnls	*w:gm*	**nnls**	*w:e*	0.00	0.00	0.00	0.00	0.00	0.04	2.07	50.93
nnls	*w:gm*	**nnls**	*w:em*	0.00	0.00	0.00	0.00	0.00	15.53	23.26	460.23
nnls	*w:gm*	**nnls**	*auto*	0.00	0.00	2.71	21.00	0.00	0.03	0.06	0.54

$\mathbf{Met_1}$	$\mathbf{Par_1}$	$\mathbf{Met_2}$	$\mathbf{Par_2}$	m **PH**	$\tilde{m}$ **PH**	μ **PH**	M **PH**	m **sco**	$\tilde{m}$ **sco**	μ **sco**	M **sco**
nnls	*w:gm*	**nnls**	*geo*	0.00	3.00	20.34	579.00	0.00	0.04	0.06	1.00
nnls	*w:gm*	**nnls**	*m:w*	0.00	0.00	0.00	0.00	0.00	0.00	0.16	0.80
nnls	*w:gm*	**nnls**	*m:e*	0.00	12.50	57.42	1102.00	0.00	0.33	0.40	1.55
nnls	*w:gm*	**nnls**	*m:g*	0.00	36.00	124.67	1436.00	0.05	0.46	0.52	1.52
nnls	*w:gm*	**nnls**	*m:a*	0.00	5.00	8.65	74.00	0.00	0.24	0.27	0.90
nnls	*w:e*	**nnls**	*w:em*	0.00	0.00	0.00	0.00	0.00	9.41	18.27	452.06
nnls	*w:e*	**nnls**	*auto*	0.00	0.00	2.44	21.00	0.00	0.00	1.88	14.97
nnls	*w:e*	**nnls**	*geo*	0.00	0.00	4.09	39.00	0.00	0.02	2.04	51.00
nnls	*w:e*	**nnls**	*m:w*	0.00	0.00	0.00	0.00	0.00	0.00	1.55	14.90
nnls	*w:e*	**nnls**	*m:e*	0.00	6.00	9.38	45.00	0.05	0.42	3.03	51.00
nnls	*w:e*	**nnls**	*m:g*	0.00	18.00	23.16	76.00	0.05	0.56	2.15	7.41
nnls	*w:e*	**nnls**	*m:a*	0.00	3.00	4.41	18.00	0.00	0.32	2.43	14.62
nnls	*w:em*	**nnls**	*auto*	0.00	0.00	2.71	21.00	0.00	10.81	18.40	346.45
nnls	*w:em*	**nnls**	*geo*	0.00	3.00	20.34	579.00	0.00	15.65	22.63	346.44
nnls	*w:em*	**nnls**	*m:w*	0.00	0.00	0.00	0.00	0.00	0.00	16.40	346.23
nnls	*w:em*	**nnls**	*m:e*	0.00	12.50	57.42	1102.00	0.05	21.91	28.21	141.00
nnls	*w:em*	**nnls**	*m:g*	0.00	36.00	124.67	1436.00	0.05	10.92	21.72	130.00
nnls	*w:em*	**nnls**	*m:a*	0.00	5.00	8.65	74.00	0.00	21.13	27.30	194.44
nnls	*auto*	**nnls**	*geo*	0.00	0.00	3.17	31.00	0.00	0.02	0.06	1.01
nnls	*auto*	**nnls**	*m:w*	0.00	0.00	1.88	21.00	0.00	0.00	0.13	0.61
nnls	*auto*	**nnls**	*m:e*	0.00	5.00	6.74	42.00	0.00	0.22	0.23	1.01
nnls	*auto*	**nnls**	*m:g*	0.00	12.00	21.37	76.00	0.05	0.30	0.32	0.80
nnls	*auto*	**nnls**	*m:a*	0.00	0.00	0.00	0.00	0.00	0.20	0.20	0.51
nnls	*geo*	**nnls**	*m:w*	0.00	0.00	0.92	22.00	0.00	0.00	0.16	0.79
nnls	*geo*	**nnls**	*m:e*	0.00	0.00	16.33	646.00	0.05	0.28	0.32	1.50
nnls	*geo*	**nnls**	*m:g*	0.00	11.00	68.30	1571.00	0.05	0.31	0.36	1.50
nnls	*geo*	**nnls**	*m:a*	0.00	0.00	2.51	43.00	0.00	0.25	0.25	0.86
ga	*asjp*	**ga**	*w:g*	0.00	6.00	19.42	455.00	0.00	0.22	0.27	2.11
ga	*asjp*	**ga**	*w:gm*	0.00	6.00	17.94	455.00	0.00	0.19	0.24	1.97
ga	*asjp*	**ga**	*w:e*	0.00	6.00	19.42	455.00	0.00	4.64	11.35	534.05
ga	*asjp*	**ga**	*w:em*	0.00	6.00	17.94	455.00	0.00	9.87	26.24	798.38
ga	*asjp*	**ga**	*auto*	0.00	6.00	19.81	463.00	0.00	0.23	0.30	2.26
ga	*asjp*	**ga**	*geo*	0.00	6.00	22.39	570.00	0.00	0.20	0.25	2.64
ga	*asjp*	**ga**	*m:w*	0.00	0.00	4.78	37.00	0.00	0.13	0.18	0.70
ga	*asjp*	**ga**	*m:e*	0.00	10.00	55.25	1109.00	0.01	0.30	0.49	5.42
ga	*asjp*	**ga**	*m:g*	0.00	25.00	102.48	1525.00	0.00	0.32	0.75	11.13
ga	*asjp*	**ga**	*m:a*	0.00	6.00	9.64	73.00	0.00	0.27	0.29	0.77
ga	*w:g*	**ga**	*w:gm*	0.00	0.00	0.00	0.00	0.00	0.07	0.12	0.93
ga	*w:g*	**ga**	*w:e*	0.00	0.00	0.00	0.00	0.00	4.27	9.99	533.23
ga	*w:g*	**ga**	*w:em*	0.00	0.00	0.00	0.00	0.00	10.22	27.44	797.67
ga	*w:g*	**ga**	*auto*	0.00	4.00	9.04	141.00	0.00	0.11	0.19	1.48
ga	*w:g*	**ga**	*geo*	0.00	5.00	22.61	579.00	0.00	0.09	0.17	2.85
ga	*w:g*	**ga**	*m:w*	0.00	0.00	0.00	0.00	0.00	0.07	0.12	0.41
ga	*w:g*	**ga**	*m:e*	0.00	13.00	59.22	1102.00	0.01	0.32	0.49	5.50
ga	*w:g*	**ga**	*m:g*	0.00	36.00	127.06	1436.00	0.04	0.36	0.89	11.17
ga	*w:g*	**ga**	*m:a*	0.00	5.50	8.79	74.00	0.00	0.27	0.28	0.96
ga	*w:gm*	**ga**	*w:e*	0.00	0.00	0.00	0.00	0.00	4.32	10.09	533.86
ga	*w:gm*	**ga**	*w:em*	0.00	0.00	0.00	0.00	0.00	8.52	25.02	798.18
ga	*w:gm*	**ga**	*auto*	0.00	4.00	9.00	141.00	0.00	0.10	0.18	1.62
ga	*w:gm*	**ga**	*geo*	0.00	3.00	20.45	579.00	0.00	0.03	0.08	2.75
ga	*w:gm*	**ga**	*m:w*	0.00	0.00	0.00	0.00	0.00	0.00	0.14	0.56
ga	*w:gm*	**ga**	*m:e*	0.00	12.50	57.42	1102.00	0.00	0.29	0.46	5.43

$\mathbf{Met_1}$	$\mathbf{Par_1}$	$\mathbf{Met_2}$	$\mathbf{Par_2}$	m **PH**	$\tilde{m}$ **PH**	μ **PH**	M **PH**	m **sco**	$\tilde{m}$ **sco**	μ **sco**	M **sco**
ga	*w:gm*	**ga**	*m:g*	0.00	36.00	127.06	1436.00	0.09	0.35	0.87	11.17
ga	*w:gm*	**ga**	*m:a*	0.00	5.00	8.65	74.00	0.00	0.26	0.26	0.65
ga	*w:e*	**ga**	*w:em*	0.00	0.00	0.00	0.00	0.00	8.04	22.10	487.46
ga	*w:e*	**ga**	*auto*	0.00	4.00	9.04	141.00	0.00	4.59	10.72	533.17
ga	*w:e*	**ga**	*geo*	0.00	5.00	22.61	579.00	0.00	4.32	10.08	534.10
ga	*w:e*	**ga**	*m:w*	0.00	0.00	0.00	0.00	0.00	0.28	5.90	93.52
ga	*w:e*	**ga**	*m:e*	0.00	13.00	59.22	1102.00	0.21	5.33	18.19	534.08
ga	*w:e*	**ga**	*m:g*	0.00	36.00	127.06	1436.00	0.16	3.87	10.23	182.63
ga	*w:e*	**ga**	*m:a*	0.00	5.50	8.79	74.00	0.00	4.60	6.78	41.28
ga	*w:em*	**ga**	*auto*	0.00	4.00	9.00	141.00	0.00	11.60	28.56	797.69
ga	*w:em*	**ga**	*geo*	0.00	3.00	20.45	579.00	0.00	8.41	24.59	798.44
ga	*w:em*	**ga**	*m:w*	0.00	0.00	0.00	0.00	0.00	0.00	18.06	367.59
ga	*w:em*	**ga**	*m:e*	0.00	12.50	57.42	1102.00	0.17	13.24	38.75	798.39
ga	*w:em*	**ga**	*m:g*	0.00	36.00	127.06	1436.00	0.29	7.18	17.79	92.51
ga	*w:em*	**ga**	*m:a*	0.00	5.00	8.65	74.00	0.00	16.01	24.72	304.64
ga	*auto*	**ga**	*geo*	0.00	3.00	22.23	587.00	0.00	0.10	0.21	2.85
ga	*auto*	**ga**	*m:w*	0.00	0.00	4.07	31.00	0.00	0.22	0.19	0.69
ga	*auto*	**ga**	*m:e*	0.00	11.00	56.94	1103.00	0.03	0.31	0.50	5.64
ga	*auto*	**ga**	*m:g*	0.00	30.50	118.19	1440.00	0.07	0.34	0.86	11.20
ga	*auto*	**ga**	*m:a*	0.00	0.00	0.00	0.00	0.00	0.22	0.23	0.55
ga	*geo*	**ga**	*m:w*	0.00	0.00	0.92	22.00	0.00	0.00	0.16	0.74
ga	*geo*	**ga**	*m:e*	0.00	0.00	16.33	646.00	0.05	0.27	0.37	4.73
ga	*geo*	**ga**	*m:g*	0.00	11.00	68.30	1571.00	0.05	0.27	0.52	10.85
ga	*geo*	**ga**	*m:a*	0.00	0.00	2.51	43.00	0.00	0.26	0.25	0.65
nnls	*asjp*	**ga**	*asjp*	0.00	0.00	0.00	0.00	0.00	0.05	0.12	1.44
nnls	*w:g*	**ga**	*w:g*	0.00	0.00	0.00	0.00	0.00	0.04	0.08	0.90
nnls	*w:gm*	**ga**	*w:gm*	0.00	0.00	0.00	0.00	0.00	0.02	0.04	0.37
nnls	*w:e*	**ga**	*w:e*	0.00	0.00	0.00	0.00	0.00	2.43	3.88	41.45
nnls	*w:em*	**ga**	*w:em*	0.00	0.00	0.00	0.00	0.00	12.44	24.81	759.41
nnls	*auto*	**ga**	*auto*	0.00	0.00	0.00	0.00	0.00	0.05	0.10	0.60
nnls	*geo*	**ga**	*geo*	0.00	0.00	0.00	0.00	0.00	0.00	0.05	2.62
nnls	*m:w*	**ga**	*m:w*	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.39
nnls	*m:e*	**ga**	*m:e*	0.00	0.00	0.00	0.00	0.00	0.12	0.24	5.04
nnls	*m:g*	**ga**	*m:g*	0.00	0.00	0.00	0.00	0.00	0.08	0.24	10.70
nnls	*m:a*	**ga**	*m:a*	0.00	0.00	0.00	0.00	0.00	0.07	0.12	0.57
nnls	*asjp*	**nj**	*asjp*	0.00	9.00	36.76	856.00	0.00	0.20	0.33	1.39
nnls	*w:g*	**nj**	*w:g*	0.00	7.00	13.04	68.00	0.00	0.54	0.63	1.94
nnls	*w:gm*	**nj**	*w:gm*	0.00	13.00	36.38	485.00	0.00	0.07	0.18	1.00
nnls	*w:e*	**nj**	*w:e*	0.00	7.00	13.07	68.00	0.00	23.01	26.80	95.21
nnls	*w:em*	**nj**	*w:em*	0.00	12.50	36.41	487.00	0.00	27.25	42.80	401.20
nnls	*auto*	**nj**	*auto*	0.00	5.00	9.32	66.00	0.00	0.40	0.59	1.80
nnls	*geo*	**nj**	*geo*	0.00	12.50	62.28	1904.00	0.00	0.02	0.14	1.00
nnls	*m:w*	**nj**	*m:w*	0.00	10.00	32.53	344.00	0.00	0.31	0.54	1.00
nnls	*m:e*	**nj**	*m:e*	0.00	8.00	68.48	1637.00	0.00	0.21	0.24	1.03
nnls	*m:g*	**nj**	*m:g*	0.00	6.00	50.46	2012.00	0.00	0.19	0.27	4.37
nnls	*m:a*	**nj**	*m:a*	0.00	10.00	26.52	323.00	0.00	0.16	0.21	1.24
nj	*asjp*	**ga**	*asjp*	0.00	9.00	37.13	856.00	0.00	0.22	0.36	1.92
nj	*w:g*	**ga**	*w:g*	0.00	9.00	17.93	198.00	0.00	0.59	0.73	4.85
nj	*w:gm*	**ga**	*w:gm*	0.00	13.00	36.78	485.00	0.00	0.07	0.18	1.00
nj	*w:e*	**ga**	*w:e*	0.00	9.50	17.99	200.00	0.03	23.81	30.45	179.53
nj	*w:em*	**ga**	*w:em*	0.00	13.00	36.81	487.00	0.00	22.02	44.96	810.98
nj	*auto*	**ga**	*auto*	0.00	5.00	10.00	66.00	0.00	0.39	0.58	1.81

$\mathbf{Met_1}$	$\mathbf{Par_1}$	$\mathbf{Met_2}$	$\mathbf{Par_2}$	m **PH**	$\tilde{m}$ **PH**	μ **PH**	M **PH**	m **sco**	$\tilde{m}$ **sco**	μ **sco**	M **sco**
nj	*geo*	**ga**	*geo*	0.00	13.00	62.61	1904.00	0.00	0.02	0.17	2.76
nj	*m:w*	**ga**	*m:w*	2.00	11.00	33.47	344.00	0.02	0.37	0.56	1.00
nj	*m:e*	**ga**	*m:e*	0.00	9.00	69.38	1645.00	0.00	0.22	0.35	5.15
nj	*m:g*	**ga**	*m:g*	0.00	6.00	51.22	2012.00	0.00	0.18	0.36	9.20
nj	*m:a*	**ga**	*m:a*	1.00	11.00	27.42	325.00	0.00	0.18	0.22	1.00

First, it can be seen that most distributions are very skewed with the median much smaller than the mean, suggesting that for the vast majority of families and classifications, the differences between methods and distances are relatively small. Second, it must be pointed out that the "score" comparisons must be taken with care as for most methods and distances the branch lengths are relatively small, except for *geo* where they can reach tens of thousands (of kilometers). Third, it can be seen that the differences in topology ("PH84") are mostly small, but tend to be larger when the *nj* method is used; this is to be expected given that all methods (except *nj*) use the tree topology as given by the classification, with the small differences resulting from some leaf nodes (or subtrees) being dropped out due to missing data in the distances matrix. To zoom in the differences in topology induced by *nj* we must look at a comparison such as *constant* (or any other method that faithfully preserves the classification-given topologies including *proportional* and *grafen*) versus *nj* using "PH85": Table 9 shows the t-tests of the differences between pairs of *nj* distances relative to *constant*.

Table 9: Differences in the topology of the trees generated by *nj* when applied to pairs of distances relative to classification-given true topology (embodied by *constant*) across all classifications. The first two columns contain the distance matrices supplied to *nj*: for each of these the distribution of 'PH85' differences versus *constant* was computed and columns 3, 4 and 5 show the t-test between these distributions; a significant p-value means that *nj* applied to the two distance matrices produces very different topologies relative to *constant* (and **bolds** the corresponding row in the table).

$\mathbf{Distance_1}$	$\mathbf{Distance_2}$	t	df	p
asjp	***w:g***	**3.31**	**288.49**	**0.0011****
asjp	*w:gm*	0.13	483.86	0.9
asjp	***w:e***	**3.30**	**288.73**	**0.0011****
asjp	*w:em*	0.13	483.87	0.9
asjp	***auto***	**4.10**	**261.78**	**5.4e-05*****
asjp	*geo*	-0.39	564.42	0.7
asjp	***m:w***	**2.38**	**298.65**	**0.018***
asjp	*m:e*	0.03	137.25	0.98
asjp	*m:g*	0.91	383.56	0.36
asjp	***m:a***	**2.94**	**314.57**	**0.0035****
w:g	***w:gm***	**-3.59**	**293.48**	**0.00038*****
w:g	*w:e*	-0.01	160.00	0.99
w:g	***w:em***	**-3.59**	**293.48**	**0.00038*****
w:g	***auto***	**2.07**	**121.31**	**0.04***
w:g	***geo***	**-3.56**	**357.45**	**0.00042*****
w:g	*m:w*	-0.90	90.72	0.37
w:g	*m:e*	-1.85	87.55	0.067
w:g	*m:g*	-1.61	203.12	0.11
w:g	*m:a*	-0.30	111.51	0.77
w:gm	***w:e***	**3.59**	**293.75**	**0.00039*****
w:gm	*w:em*	-0.00	488.00	1

Distance$_1$	Distance$_2$	t	df	p
w:gm	***auto***	**4.54**	**260.61**	**8.5e-06*****
w:gm	*geo*	-0.54	554.04	0.59
w:gm	***m:w***	**2.47**	**271.52**	**0.014***
w:gm	*m:e*	-0.06	122.84	0.95
w:gm	*m:g*	0.85	346.01	0.4
w:gm	***m:a***	**3.13**	**308.07**	**0.0019****
w:e	***w:em***	**-3.59**	**293.75**	**0.00039*****
w:e	***auto***	**2.08**	**121.09**	**0.04***
w:e	***geo***	**-3.55**	**357.69**	**0.00043*****
w:e	*m:w*	-0.90	90.91	0.37
w:e	*m:e*	-1.85	87.59	0.067
w:e	*m:g*	-1.60	203.26	0.11
w:e	*m:a*	-0.29	111.77	0.77
w:em	***auto***	**4.55**	**260.61**	**8.4e-06*****
w:em	*geo*	-0.54	554.04	0.59
w:em	***m:w***	**2.47**	**271.52**	**0.014***
w:em	*m:e*	-0.06	122.84	0.95
w:em	*m:g*	0.85	346.01	0.4
w:em	***m:a***	**3.13**	**308.07**	**0.0019****
auto	***geo***	**-4.29**	**329.55**	**2.3e-05*****
auto	***m:w***	**-2.05**	**71.35**	**0.044***
auto	***m:e***	**-2.26**	**84.36**	**0.026***
auto	***m:g***	**-2.22**	**188.49**	**0.028***
auto	*m:a*	-1.70	80.42	0.093
geo	***m:w***	**2.69**	**359.67**	**0.0075****
geo	*m:e*	0.30	149.23	0.77
geo	*m:g*	1.23	432.14	0.22
geo	***m:a***	**3.22**	**381.95**	**0.0014****
m:w	*m:e*	-1.44	102.63	0.15
m:w	*m:g*	-1.01	242.66	0.32
m:w	*m:a*	0.58	120.05	0.57
m:e	*m:g*	0.64	159.94	0.53
m:e	*m:a*	1.72	95.06	0.088
m:g	*m:a*	1.41	229.50	0.16

References

Allaire, JJ, Joe Cheng, Yihui Xie, Jonathan McPherson, Winston Chang, Jeff Allen, Hadley Wickham, and Rob Hyndman. 2015. *Rmarkdown: Dynamic Documents for R*. http://CRAN.R-project.org/package=rmarkdown.

Bakker, Dik, André Müller, Viveka Velupillai, Søren Wichmann, Cecil H. Brown, Pamela Brown, Dmitry Egorov, Robert Mailhammer, Anthony Grant, and Eric W. Holman. 2009. "Adding Typology to Lexicostatistics: A Combined Approach to Language Classification." *Linguistic Typology* 13 (1). doi:10.1515/LITY.2009.009.

Bouckaert, Remco, Philippe Lemey, Michael Dunn, Simon J. Greenhill, Alexander V. Alekseyenko, Alexei J. Drummond, Russell D. Gray, Marc A. Suchard, and Quentin D. Atkinson. 2012. "Mapping the Origins and Expansion of the Indo-European Language Family." *Science* 337 (6097): 957–60. http://www.sciencemag.org/content/337/6097/957.

Bowern, Claire, and Bethwyn Evans. 2014. *The Routledge Handbook of Historical Linguistics*. Routledge.

Campbell, Lyle, and William J. Poser. 2008. *Language Classification: History and Method*. Cambridge University Press.

Chang, Will, Chundra Cathcart, David Hall, and Andrew Garrett. 2015. "Ancestry-Constrained Phylogenetic Analysis Supports the Indo-European Steppe Hypothesis." *Language* 91 (1): 194–244.

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

Dunn, Michael, Simon J. Greenhill, Stephen C. Levinson, and Russell D. Gray. 2011. "Evolved Structure of Language Shows Lineage-Specific Trends in Word-Order Universals." *Nature* 473 (April): 79–82. doi:10.1038/nature09923.

Gower, J. C. 1971. "A General Coefficient of Similarity and Some of Its Properties." *Biometrics* 27 (4): 857–71. doi:10.2307/2528823.

Grafen, A. 1989. "The Phylogenetic Regression." *Philosophical Transactions of the Royal Society of London. B, Biological Sciences* 326 (1233): 119–57. doi:10.1098/rstb.1989.0106.

Hammarström, Harald, Robert Forkel, Martin Haspelmath, and Sebastian Nordhoff, eds. 2014. *Glottolog 2.3*. Leipzig: Max Planck Institute for Evolutionary Anthropology. http://glottolog.org.

Hijmans, Robert J. 2014. *Geosphere: Spherical Trigonometry*. http://CRAN.R-project.org/package=geosphere.

Kuhner, M. K., and J. Felsenstein. 1994. "A Simulation Comparison of Phylogeny Algorithms Under Equal and Unequal Evolutionary Rates." *Molecular Biology and Evolution* 11 (3): 459–68.

Ladd, D. Robert, Seán G. Roberts, and Dan Dediu. 2015. "Correlational Studies in Typological and Historical Linguistics." *Annual Review of Linguistics* 1 (1): 221–41. doi:10.1146/annurev-linguist-030514-124819.

Lewis, M. Paul, Gary F. Simons, and Charles D. Fennig, eds. 2014. *Ethnologue: Languages of the World*. 17th ed. Dallas, Tex.: SIL International. http://www.ethnologue.com.

Mace, R., and M. Pagel. 1994. "The Comparative Method in Anthropology." *Current Anthropology* 35: 549–64.

Maddison, David R., David L. Swofford, and Wayne P. Maddison. 1997. "Nexus: An Extensible File Format for Systematic Information." *Systematic Biology* 46 (4): 590–621. doi:10.1093/sysbio/46.4.590.

Maechler, Martin, Peter Rousseeuw, Anja Struyf, Mia Hubert, and Kurt Hornik. 2015. "Cluster: Cluster Analysis Basics and Extensions."

Maurits, Luke, and Thomas L. Griffiths. 2014. "Tracing the Roots of Syntax with Bayesian Phylogenetics." *Proceedings of the National Academy of Sciences* 111 (37): 13576–81. doi:10.1073/pnas.1319042111.

Nichols, Johanna, Alena Witzlack-Makarevich, and Balthasar Bickel. 2013. "The AUTOTYP Genealogy and Geography Database: 2013 Release."

Paradis, E., J. Claude, and K. Strimmer. 2004. "APE: Analyses of Phylogenetics and Evolution in R Language." *Bioinformatics* 20: 289–90.

Penny, David, and M. D. Hendy. 1985. "The Use of Tree Comparison Metrics." *Systematic Biology* 34 (1): 75–82. doi:10.1093/sysbio/34.1.75.

R Core Team. 2014. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. http://www.R-project.org.

Roberts, Seàn, and James Winters. 2013. "Linguistic Diversity and Traffic Accidents: Lessons from Statistical Studies of Cultural Traits." *PLoS ONE* 8 (8): e70902. doi:10.1371/journal.pone.0070902.

Rzhetsky, Andrey, and Masatoshi Nei. 1992. "A Simple Method for Estimating and Testing Minimum-Evolution Trees." *Mol. Biol. Evol* 9 (5): 945–67. http://test.scripts.psu.edu/nxm2/1992%20Publications/1992-rzhetsky-nei.pdf.

Saitou, N., and M. Nei. 1987. "The Neighbor-Joining Method: A New Method for Reconstructing Phylogenetic Trees." *Molecular Biology and Evolution* 4 (4): 406–25. http://mbe.oxfordjournals.org/content/4/4/406.

Schliep, K.P. 2011. "Phangorn: Phylogenetic Analysis in R." *Bioinformatics* 27 (4): 592–93.

Scrucca, Luca. 2013. "GA: A Package for Genetic Algorithms in R." *Journal of Statistical Software* 53 (4): 1–37. http://www.jstatsoft.org/v53/i04/.

Wichmann, Søren, André Müller, Annkathrin Wett, Viveka Velupillai, Julia Bischoffberger, Cecil H. Brown, Eric W. Holman, et al. 2013. "The ASJP Database (Version 16)." http://asjp.clld.org/.