Statistical and computational elaborations of the classical comparative method

Gerhard Jäger and Johann-Mattis List

April 11, 2016

Due to the increasing amount of large digitally available datasets, computational approaches play an increasingly important role in historical linguistics, and many attempts have been made to computerize various aspects of the classical comparative method for language comparison. The article gives an overview on popular and important approaches which have been developed in the last two decades. These include approaches to sequence comparison and phylogenetic reconstruction. The former cover the tasks of cognate and sound correspondence identification in the classical comparative method. The latter address the genetic classification of language families. We conclude our description by pointing to recent approaches to borrowing detection and semantic reconstruction.

1 Introduction

The central method in historical linguistics is the *comparative method* (Meillet 1954, Weiss 2014). It has successfully elucidated the history of a wide range of language families of varying size and age (Baldi 1990, Campbell and Poser 2008) and external evidence has often confirmed the validity of the findings (McMahon and McMahon 2005:10-14). The comparative method is not just a simple technique, but rather an *overarching framework* to study language history (Klimov 1990, Ross and Durie 1996, Fox 1995, Jarceva 1990). This framework has an underlying workflow that scholars implicitly follow (see Figure 1, following Ross and Durie 1996). The most crucial part is the identification of *cognate words* ② and regular *sound correspondences* ③. The *iterative character* of the workflow requires repetition in all steps. Iteration is important to address circularity problems: *cognate words* ② can, for example, only be identified with help of regular *sound correspondences* ③, but sound correspondences themselves occur only in cognate words. An iterative procedure circumvents this problem by starting with an initial hypothesis regarding sound correspondences and cognate words which is then constantly revised.

Despite its benefit and its successful application, the comparative method has a couple of drawbacks. Its application is very slow and requires highly trained historical linguists. The procedure itself lacks transparency, in so far as the scholars' intuition still plays a major role (Schwink 1994). It also shows a certain lack of reliability, since neither formal guidelines nor statistical tests are used to arrive at the hypotheses (Baxter and Manaster Ramer 2000:169-172), which makes it difficult to guarantee that scholars working independently will arrive at the same conclusions (McMahon and McMahon 2005:26-29). Given the drawbacks of the manual comparative method and the ever increasing availability of digital data in historical linguistics, it is not surprising that many attempts have been made to get aid from computers. These attempts are reflected in a *quantitative turn* in historical linguistics which started in the beginning of the second millenium and surfaced until now in form of

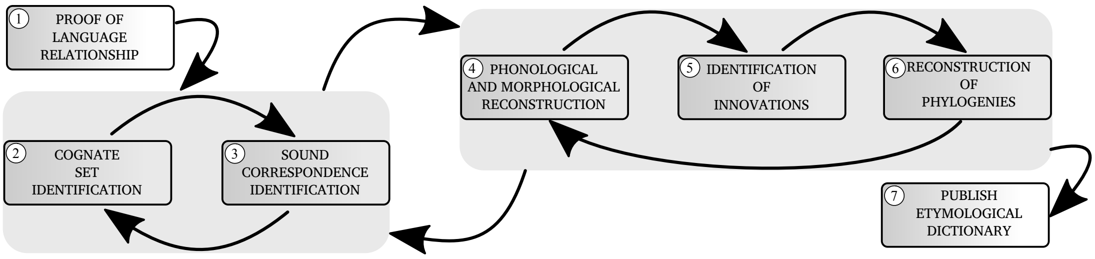

① PROOF OF LANGUAGE RELATIONSHIP
② COGNATE SET IDENTIFICATION
③ SOUND CORRESPONDENCE IDENTIFICATION
④ PHONOLOGICAL AND MORPHOLOGICAL RECONSTRUCTION
⑤ IDENTIFICATION OF INNOVATIONS
⑥ RECONSTRUCTION OF PHYLOGENIES
⑦ PUBLISH ETYMOLOGICAL DICTIONARY

Figure 1: Workflow for the comparative method by Ross and Durie (1996) with two major and multiple minor stages of iteration.

many approaches that automate certain parts of the classical workflow of the comparative method. Given the complex workflow of the classical comparative method, it is obvious that none of the automatic approaches proposed so far has ever tried to replicate it entirely. Instead, automatic approaches often also set an additional focus and follow different paths. As an example, Table 1 contrasts the modules of the classical workflow, as given in Figure 1, with popular automatic approaches. As can be seen from the table, nearly all of the major modules of the comparative method are addressed in at least one published approach. However, there is no strict overlap between any of the classical "modules" and the modern automatic approaches.

#	Classical HL	Computational HL	Examples
①	proof of language relationship	probability testing	Baxter and Manaster Ramer (2000), Kessler (2001), Ringe (1992)
		phonetic distance	Jäger (2015)
②	cognate set identification	matching sound classes	Turchin et al. (2010)
		phonetic distance and partitioning	List (2012*a*, 2014*b*), Steiner et al. (2011)
③	sound correspondence identification	phonetic alignments	Kondrak (2000), List (2012*b*), Prokić et al. (2009), Prokić and Cysouw (2013)
④	linguistic reconstruction	probabilistic string transducer	Bouchard-Côté et al. (2013)
⑤	identification of innovations	various methods for lexical, gramm., and morphol. data	Chang et al. (2015), Gray and Atkinson (2003), Jäger (2015), Longobardi et al. (2013a), Ringe et al. (2002)
⑥	phylogenetic reconstruction		
⑦	etymologies	(borrowing detection)	van der Ark et al. (2007), List et al. (2014a), Nelson-Sathi et al. (2011)
		(ancestral state reconstruction)	Jäger and List (2016), List (2015)

Table 1: Comparing computational approaches in historical linguistics with the classical comparative method: Approaches in brackets in the "Computational HL" column reflect only certain aspects of the original workflow.

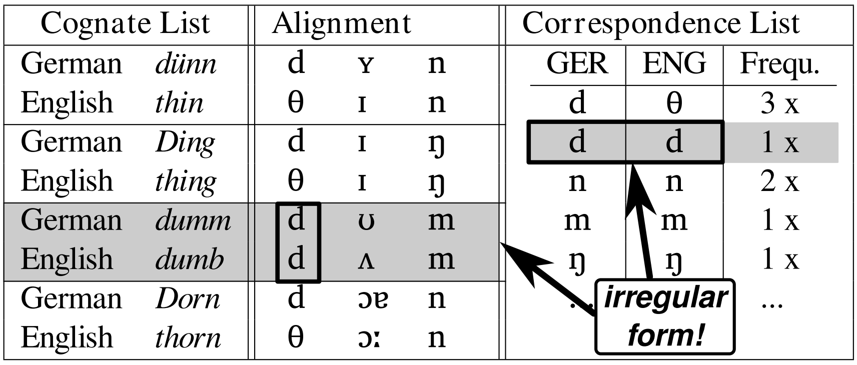
Judging from their accessibility, accuracy, and acceptance, the most developed approaches in computational historical linguistics are approaches to *sequence comparison* and *phylogenetic reconstruction*, which can be roughly identified with working steps ② and ⑥ of the workflow by Ross and Durie (1996). In the following, we will briefly introduce the main ideas and the major methods and algorithms behind these approaches. In a further section we will then point to recent promising attempts to tackle further challenges in automatic language comparison.

2 Sequence Comparison

The basis of the classical comparative method, the identification of regularly corresponding sounds and cognate words in genetically related languages, is essentially a very specific task of *sequence comparison*, since the phonic substance of words, morphemes, and also sentences manifests itself in dependence of time (de Saussure 1916:103), and our linguistic theories of phonology and morphology allow us to cut these streams into units which distinguish or constitute meaning. For this reason, it seems legitimate to make use of general approaches to sequence comparison, developed in computer science and evolutionary biology, to compare linguistic sequences. Note that in this context, we follow the working steps ② and ③ of the comparative method in taking words and morphemes as our primary linguistic sequences.

2.1 Alignment Analyses

Comparing sequences at an abstract level requires the identification of those segments which *match* across sequences, that is, those segments which are identical or share a common history. For example, when comparing the sound sequences English *daughter* [dɔːtɚ] with Greek *thigatera* [θiɣatɛra], we know from the historical development of the words that English [d] corresponds with Greek [θ], as does English [t] with Greek [t]. This kind of analysis is at the core of all endeavour in historical linguistics, since it is the only way to identify regular sound correspondences across cognate words in different languages (see Figure 2). *Alignment analyses* are a very general and convenient way to model differences between sequences. In alignment analyses, sequences are arranged in the rows of a matrix in such a way that all corresponding segments occur in the same column (Gusfield 1997:216). In order to ease the visualization, it is furthermore common to fill empty cells in the matrix with *gap symbols* (usually a dash: -). Empty cells result from segments which do not match with other segments, such as the two instances of [a] in Greek *thigatera*, which do not have an English counterpart.



Cognate List		Alignment			Correspondence List		
German	*dünn*	d	ʏ	n	GER	ENG	Frequ.
English	*thin*	θ	ɪ	n	d	θ	3 x
German	*Ding*	d	ɪ	ŋ	d	d	1 x
English	*thing*	θ	ɪ	ŋ	n	n	2 x
German	*dumm*	d	ʊ	m	m	m	1 x
English	*dumb*	d	ʌ	m	ŋ	ŋ	1 x
German	*Dorn*	d	ɔɐ	n	...		...
English	*thorn*	θ	ɔː	n			

irregular form!

Figure 2: Sequence comparison as the basis for sound correspondence detection. The figure shows how correspondence counts are derived from the alignments of putative cognate words. When correspondences occur only sporadically, as the one between [d] and [d] in German and English, this is provides evidence that the words are not regularly related (German *dumm* is an irregular reflex of Old High German *tumb*, probably under influence of Low German varieties).

Alignment analyses are a very common way to model differences between sequences and regularly used across different scientific fields, such as molecular biology (Durbin et al. 2002), spelling correction (Oflazer 1996), or plagiarism detection (Horton et al. 2010). Implicitly, the use of alignment analyses dates back to the founding days of historical linguistics, when Rasmus Rask (1787–1832) and Jacob Grimm (1785–1863) laid the foundation of the notion of *sound laws* (Rask 1818, Grimm 1822), although the earliest explicit visualization of sound correspondences with help of alignment analyses we could find so far dates back to the beginning of the 20th century (Dixon and Kroeber 1919). Since the middle of the 1990s automatic alignment analyses, developed in biology and computer sciences from the 1970s onwards (Needleman and Wunsch 1970, Wagner and Fischer 1974) have also been increasingly applied in historical linguistics and dialectology (Kessler 1995, Nerbonne et al. 1996, Covington 1996, Kondrak 2000), and today, one can say that they play a crucial role in the quickly developing field of quantitative linguistics. Figure 3 illustrates the basic ideas behind pairwise alignment analyses in historical linguistics with help of the two words English *daughter* and Greek *thigatera*.

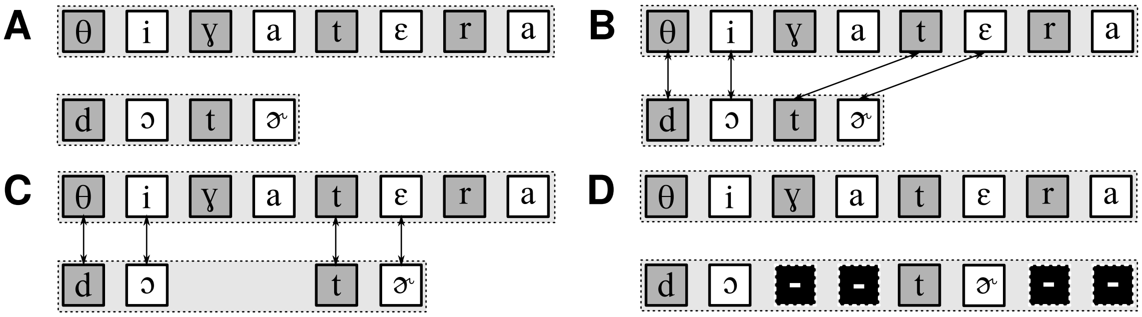


Figure 3: Alignment of English and Greek words for 'daughter'. A shows the starting point with both words unaligned. B shows the matching process in which cognate sounds are identified between both words. C shows how the words are re-arranged so that cognate sounds appear in the same column of a fictive matrix. D shows the resulting alignment in which dashes as gap symbols have been introduced to fill those slots in which a sound in one word does not have a matching counterpart in the other words.

2.1.1 Pairwise Phonetic Alignment

For reasons of complexity it is common to distinguish between pairwise and multiple alignment analyses. Even the pairwise alignment of two strings can become really complex, since the number of possible alignments increases drastically with the length of the sequences (Rosenberg and Ogden 2009). While there are only 681 possibilities for the alignment of two strings of the length 5 and 4, there are 8 097 453 possibilities for two strings of length 10 and 10 (Torres et al. 2003). For this reason, automatic algorithms cannot simply test all possible alignments between two sequences but need to employ a smart search strategy that minimizes the search space instead. The development of the general strategy to tackle this problem, which is still used today, goes back to the 1970s, when biologists (Needleman and Wunsch 1970) and computer scientists (Wagner and Fischer 1974) independently proposed an efficient solution for the *global alignment problem*. Due to its different origins, this algorithm is usually called Needleman-Wunsch algorithm (NW algorithm) in the context of biology, and Wagner-Fischer algorithm in computer science. Although both algorithms do not differ in their basic strategy, they differ in their output. While the Needleman-Wunsch algorithm yields a *similarity score* between two sequences, the Wagner-Fischer algorithm yields a *distance score*. In historical linguistics, the Needleman-Wunsch algorithm is the preferred variant in computational applications (Kondrak 2000, List 2012*c*). In computational dialectology, the Wagner-Fischer algorithm is commonly used.[1]

The basic idea of the Needleman-Wunsch and the Wagner-Fischer algorithm is to reduce the problem of finding an optimal alignment of two sequences by 'using previous solutions for optimal alignments of smaller subsequences' (Durbin et al. 2002:19). This approach is known as *dynamic programming* and defines a family of algorithms with very similar characteristics (Eddy 2004). It would go beyond the scope of this chapter to present the dynamic programming algorithm for pairwise alignment analyses in all detail. For a detailed description of the Needleman-Wunsch algorithm

[1] In computational dialectology, the algorithm is often falsely labelled as *Levensthein algorithm*, named after V. I. Levensthein. While Levenshtein proposed a distance measure for the comparison of two sequences in 1965, he never published the algorithm to automatically compute it.

along with many examples, we refer the readers to Kondrak (2002:20-65) and List (2014*b*:77-82). An interactive demo of the Wagner-Fischer algorithm is presented in (List 2016*b*) and can be directly accessed at `http://lingulist.de/pyjs/demos/wf-demo.html`.

The major components of the algorithm are a *scoring function* which handles the similarity between segments, and the *main loop* which manages how sequences are compared in general. Following these two components, one can therefore make a distinction between *substantial* and *structural* extensions to the basic algorithm which play both a crucial role in phonetic alignment analyses in historical linguistics. Substantial extensions define how sounds are compared by the algorithm. In its simplest form, only two kinds of differences are defined: two segments are either identical, or different. When dealing with distance scores, as in the Wagner-Fischer algorithm, this could be expressed by giving a score of 0 for segment identity and a score of 1 for segment difference. When dealing with similarities, one usually gives a negative value to different segments and a positive value to similar ones. The score for a whole alignment between two sequences is usually identical with the sum of the distance or similarity scores for all segment pairs in an alignment. Applied to the alignment in Figure 3, the distance score would sum up to 7, since there is only one segment pair out of 8 pairs in the alignment which is identical.

As we can see from this high score for two words which are actually cognate, this distinction is not very satisfying, since we know that sounds may exhibit very fine-grained *degrees* of similarity, and trained historical linguists would probably agree that the difference between a [p] and an [f] is quite different from the difference between a [p] and a [k]. One seemingly natural solution would be the use of distinctive features to describe each sound and a rough comparison of the features, using, for example, the Hamming distance (Hamming 1950) to derive a similarity score for individual sound pairs. The disadvantage of this naive feature approach is that all features are given the same weight, although we intuitively know that certain features are more relevant for historical comparison than others. The ALINE algorithm proposed by Kondrak (2000) addresses this problem explicitly by proposing *multi-valued features* from which individual weights for sound pairs are derived. An alternative to feature-approaches is to reduce the phonetic space by clustering sounds into classes which frequently occur in correspondence relation in genetically related languages (Dolgopolsky 1964). The advantage of sound classes is that they are very flexible and very easy to handle. All that needs to be defined is a mapping from a phonetic transcription to a simpler sound class transcription. Following Dolgopolsky's sound class approach, for example, English *daughter* could be rendered as "TVTV", and Greek *thigatera* could be rendered as "TVKVTVRV", and aligning the words with the classical Needleman-Wunsch algorithm would yield the correct alignment. Furthermore, transitions between sound classes can be easily defined and passed as an extended scoring function to the alignment algorithm. Sound classes are used in different versions across different research projects. Turchin et al. (2010) and Kassian et al. (2015) use Dolgopolsky's original sound class system of 10 consonant classes for cognate detection. The ASJP project (Holman et al. 2008*b*) employs a sound class system of 40 classes (34 consonants and 6 vowels), and the SCA algorithm employs an expanded Dolgopolsky system of 28 classes (List 2012*c*). The three sound class models are contrasted in Figure 4.

Apart from the substantial extensions using feature scores or sound class models, various structural extensions to the basic algorithm have been proposed and tested in the past. While the basic algorithm, for example, compares two sequences globally, thus trying to match all segments completely, *local alignment*, first proposed by Smith and Waterman (1981), allows to search for the best scoring subsequence between to sequences instead. Essentially, a local alignment may refuse to completely align to strings, ignoring prefixes and suffixes. Thus, while a global alignment analysis of English *strawberry* [strɔːbərɪ] and German *Erdbeere* [eːrtbeːrə] would try by all means to align not only the cognate parts of the words (*-berry* and *-beere*), but also the unrelated morphemes *straw-* and *Erd-*. A local alignment algorithm, however, would simply leave those parts unaligned. Similar to local

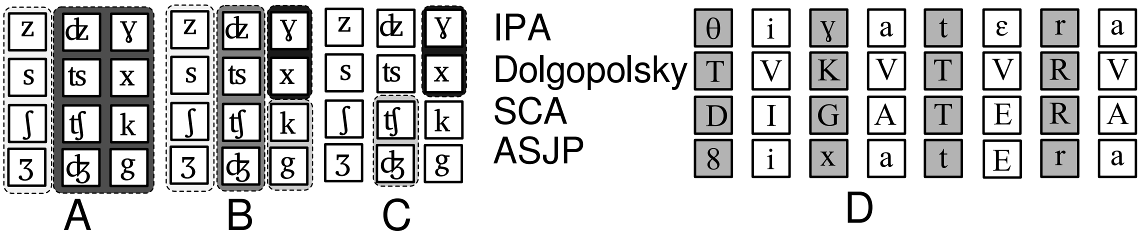


Figure 4: Illustrating the differences between different sound class systems. The graphic shows different sound class systems and how they reduce phonetic space. A is the model by Dolgopolsky (1964), B is the model by List (2012), and C is the model by Holman et al. (2008). D illustrates the sound class conversion for Greek *thigatera*.

but less strict are *semi-global* alignment analyses (Durbin et al. 2002:26f). In semi-global alignment analyses, prefixes or suffixes in either of the sequences can be ignored, but it is not possible to strip off two prefixes or two suffixes in both sequences. As a result, a semi-global alignment analysis of *strawberry* and *Erdbeere* would try to match the [t] in *straw* with the [t] in the phonetic transcription of *Erd* since the overall similarity of the sequences would be higher.

As a very specific modification of the basic algorithm, List (2012*c*) proposes *secondary alignment* (List 2014*b*:88-91). In contrast to traditional alignment analyses, be they global, local, or semi-global, secondary alignment allows to define a secondary layer of segmentation, like, for example, syllable or morpheme boundaries. The core idea of the secondary alignment extension of the basic algorithm is that these boundaries are preserved during the whole alignment process. As a result, no single morpheme in one sequence can be aligned with two other morphemes in the other sequence. This is especially important for alignment analyses of South East Asian languages, where the majority of all words consist of more than one morpheme. Aligning Hǎikǒu Chinese 日 [zit^3] 'sun' with Běijīng Chinese 日頭 [ʐʅ^{51}t^{h}ou^1] 'sun', for example, normal alignment algorithms would certainly match the [t] in Hǎikǒu with the [t^h] in Běijīng, ignoring that the latter belongs to another morpheme. Provided that morpheme boundaries are indicated in the words, secondary alignment correctly aligns both words, since the alignment of Hǎikǒu [t] and Běijīng [t^h] would contradict the rule that one morpheme in one word can only be aligned with one morpheme in the other word.

Different software packages and algorithms for alignment analyses in historical linguistics and dialectology have been proposed in the past. Table 2 roughly compares those which are most frequently mentioned in the literature for a couple of different aspects, such as the basic method, the modes (structural extensions), the scoring function (substantial extensions) and the availability.

2.1.2 Multiple Phonetic Alignment

Pairwise alignment algorithms themselves are not of a great interest for historical linguistic applications when considering only the task of aligning to words with each other in isolation, since this may well be done faster manually than to load one of the different programs mentioned above, not to speak of the fact that a trained linguist will usually outperform the computer. When carrying out large-scale comparisons of 20 and more languages or 100 and more dialect points, however, automatic pairwise alignment approaches can be very useful to aggregate linguistic distances between languages and dialects. Even more interesting, however, are multiple alignments, since they allow linguists to get a very fast impression of the diversity for a given set of cognate words, but also and especially, since they may bring in additional evidence, which could be overlooked when only considering words from the perspective of sequence pairs (Haas 1969:41, Fox 1995:68). The major problem of multiple alignment analyses is the problem of increasing complexity. While the dynamic programming solution for pairwise alignment is fast enough to make an exhaustive search for the op-

Algorithm	Author	Method	Modes	Scoring	Availability
Covington	Covington 1996	tree search	global	rudimentary scoring scheme disfavoring vowel-consonant matches	-
JAKARTA	Oakes 2000	greedy strategy	global	different sound change types with unified penalties	-
ALINE	Kondrak 2000	dynamic programming, Needleman-Wunsch	global, semi-global, local	multi-valued features	C++, Python, https://sourceforge.net/projects/pyaline/
GabMap	Nerbonne et al. 2011	dynamic programming, Wagner-Fischer	global	identity scorer preventing matching of vowels and consonants	server application, http://www.gabmap.nl
ASJP	Holman et al. 2011	dynamic programming, Wagner-Fischer	global	identity scorer applied to sound class model of 40 classes, only distances are computed, no alignments returned	Fortran code and Windows executable, http://asjp.clld.org
SCA	List 2012	dynamic programming, Needleman-Wunsch	global, semi-global, local, secondary	different sound class models with extended scoring function	Python library, http://lingpy.org
PMI	Jäger 2013	dynamic programming, Needleman-Wunsch	global	weighted alignment with scoring function for ASJP sound classes inferred from pairwise language comparisons	Python implementation

Table 2: Comparing different pairwise alignment algorithms in historical linguistics

timal alignment (given the assumptions which are encoded in the scoring function and the structural extensions), extending this algorithm to multiple sequences would yield computation times that grow exponentially with the number of sequences being analyzed (Bilu et al. 2006). For this reason, all algorithms for multiple alignment analyses are usually based on heuristics which are not guaranteed to find an optimal solution, but perform well enough in practice.

Among the most popular algorithms used for multiple alignment analyses are *progressive alignment techniques* (Feng and Doolittle 1987, Thompson et al. 1994). Progressive alignment consists of two stages. First, a *guide trees* is constructed, representing the distances between the sequences. Second, moving from the branches to the root, the sequences are successively aligned with each other. For the construction of the guide tree, different cluster algorithms can be used (cf. see also Section 3). Most biological algorithms use either UPGMA (Sokal and Michener 1958) or Neighbor-Joining (Saitou and Nei 1987). Both algorithms require a matrix of pairwise distances between all sequences as input. These distances are usually calculated by computing the pairwise alignments between all sequences. Figure 5 illustrates the process of climbing up the guide tree until all sequences are aligned for the three cognate words English *daughter*, German *Tochter*, and Greek *thigatera*.

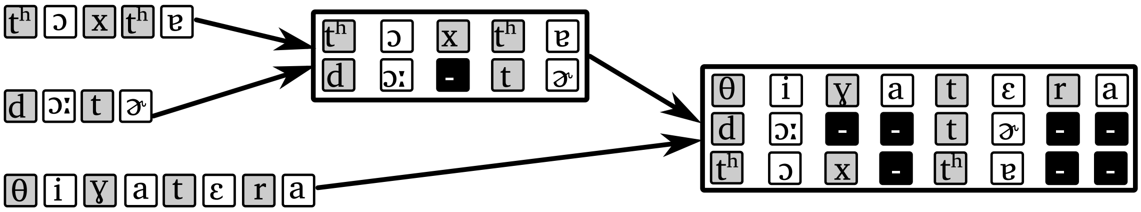


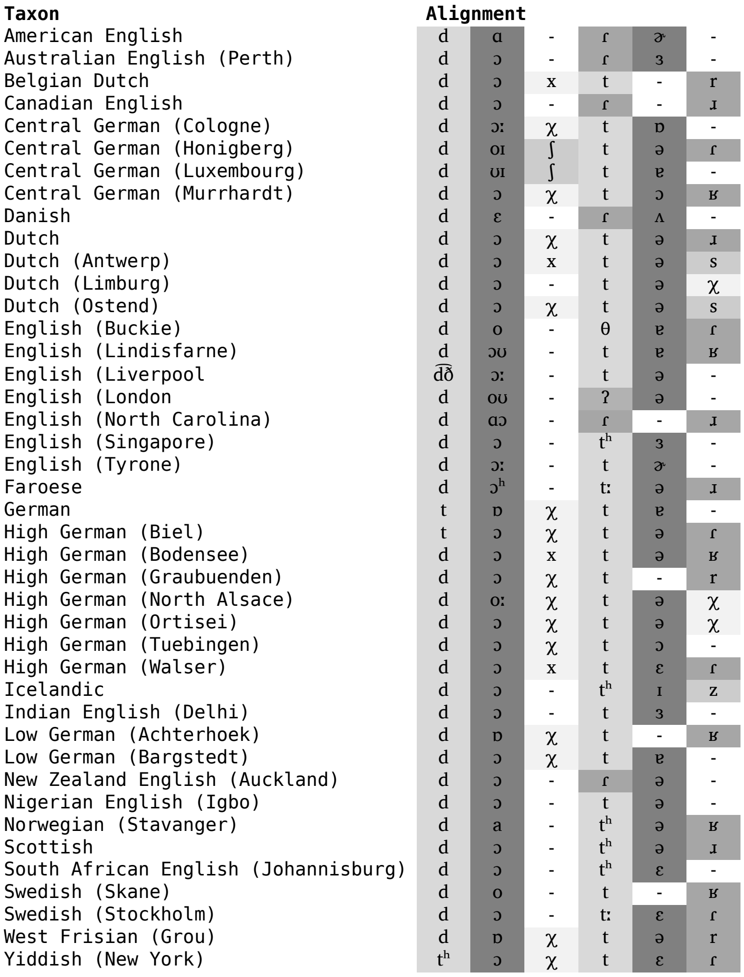
Figure 5: Progressive alignment of three words for *daughter* in English, German, and Greek.

Progressive alignment can be further enhanced by using specific methods to pre- and post-process the data. A very useful preprocessing approach that is quite popular in biological alignment algorithms is the idea of *consistency-based alignments* which was first proposed along with the T-Coffee algorithm for multiple sequence alignment in biology (Notredame et al. 2000). The basic idea of consistency-based alignments is that a good multiple alignment should be maximally consistent with a set of independently computed pairwise alignments of the sequences. This set of pairwise alignments is called a *library*, and the library itself can be filled by using different alignment approaches, such as, for example, global, and local alignment analyses between the sequence pairs. Since global and local alignments may well differ, especially in cases of very diverse sequences or in linguistics in cases of words which are only partially cognate, the inclusion of global and local information may drastically change the result of an alignment analysis. The T-Coffee algorithm assembles the information in the library in a two-stage approach. First, a new scoring function is initialized for all segments in the data, based on the pairwise alignments in the library. Second, a set of composite alignments is constructed by aligning each pair of sequences in the data *through* the rest of all sequences. The composite alignments are then used to further modify the scoring function. The newly derived scoring function is then used along with a classical progressive approach to compute the multiple alignment of all sequences. Various post-processing methods have been proposed in biology. Among the most popular strategies are methods for *iterative refinement* (Barton and Sternberg 1987, Gotoh 1996, Do et al. 2005). In these approaches, a given multiple alignment is split into two or more parts and then re-aligned. This procedure is repeated until the general alignment converges (Rosenberg 2009:58).

Despite the increased application of pairwise alignment analyses, the application of multiple alignment analyses is still in its infancy in historical linguistics and dialectology. Covington (1998) was the first to propose an algorithm for multiple phonetic alignment analyses, but the approach employed an inefficient tree-search and was only tested on up to three sequences. Later in 2009, Prokić et al. used the ALPHAMALIG algorithm (Alonso et al. 2004) to align cognate words in a large dataset of 152 words reflected in 192 Bulgarian dialects. The algorithm was originally designed to study discourse structure in NLP tasks and employed an iterative strategy that was not further explicated in the paper presenting the algorithm. However, comparing the automatic alignments with a manually compiled gold standard, the authors reported a high accuracy.

List (2012*b*) presented a progressive alignment implementation of the SCA algorithm for pairwise alignments, based on Dolgpolsky and ASJP sound classes as representation format. In addition to the sound class representation, the algorithm introduced *prosodic profiles* to account for the fact that different positions of a word show different degrees of strength and weakness with respect to change (Geisler 1992). These profiles which assign each sound segment in a word to one of 7 different classes of prosodic strength are used to individually adjust the scoring of gaps. As a result, the algorithm tries to avoid to leave initial consonants of a word unaligned, while final consonants and vowels are more easily tolerated. As a new method for post-processing the new method also contained a routine to automatically search for instances of metathesis. A test on the Bulgarian gold standard by Prokić et al. (2009) showed that the new algorithm largely outperformed the ALPHAMALIG approach List (2012*b*). List (2012*c*) further expanded the SCA algorithm by using an improved sound class model of 28 sequences and employing the T-Coffee method for pre-processing and iterative strategies for post-processing. The expanded version of the algorithm was tested on an enlarged gold standard of 750 multiple alignments (List and Prokić 2014) and showed a very high accuracy with more than 90 percent agreement with the gold standard. Jäger and List (2015) presented a fully automated workflow for language comparison in which they compare the SCA algorithm with a new version of the T-Coffee algorithm that was integrated into Jäger's (2013) PMI algorithm for pairwise alignments. The comparison of the accuracy of phylogenetic reconstruction inferred from the alignments, showed that phylogenetic trees inferred from words aligned by the PMI-T-Coffee algorithm came closer to expert judgments than trees constructed with help of the SCA algorithm.

Although they are still only rarely applied, multiple alignment analyses bear a great potential for quantitative historical linguistics and computational dialectology. The algorithms show a high accuracy in comparison with experts alignments. The computation is rather fast, and alignments of more than 200 words can be easily computed within seconds. Furthermore, multiple alignments are visually easy to process and straightforward in the representation of sequence differences. Along with



Taxon	Alignment					
American English	d	ɑ	-	ɾ	ɚ	-
Australian English (Perth)	d	ɔ	-	ɾ	ɜ	-
Belgian Dutch	d	ɔ	x	t	-	r
Canadian English	d	ɔ	-	ɾ	-	ɹ
Central German (Cologne)	d	ɔː	χ	t	ɒ	-
Central German (Honigberg)	d	oɪ	ʃ	t	ə	ɾ
Central German (Luxembourg)	d	ʊɪ	ʃ	t	ɐ	-
Central German (Murrhardt)	d	ɔ	χ	t	ɔ	ʁ
Danish	d	ɛ	-	ɾ	ʌ	-
Dutch	d	ɔ	χ	t	ə	ɹ
Dutch (Antwerp)	d	ɔ	x	t	ə	s
Dutch (Limburg)	d	ɔ	-	t	ə	χ
Dutch (Ostend)	d	ɔ	χ	t	ə	s
English (Buckie)	d	o	-	θ	ɐ	ɾ
English (Lindisfarne)	d	ɔʊ	-	t	ɐ	ʁ
English (Liverpool	d͡ð	ɔː	-	t	ə	-
English (London	d	oʊ	-	ʔ	ə	-
English (North Carolina)	d	ɑɔ	-	ɾ	-	ɹ
English (Singapore)	d	ɔ	-	t^h	ɜ	-
English (Tyrone)	d	ɔː	-	t	ɚ	-
Faroese	d	$ɔ^h$	-	tː	ə	ɹ
German	t	ɒ	χ	t	ɐ	-
High German (Biel)	t	ɔ	χ	t	ə	ɾ
High German (Bodensee)	d	ɔ	x	t	ə	ʁ
High German (Graubuenden)	d	ɔ	χ	t	-	r
High German (North Alsace)	d	oː	χ	t	ə	χ
High German (Ortisei)	d	ɔ	χ	t	ə	χ
High German (Tuebingen)	d	ɔ	χ	t	ɔ	-
High German (Walser)	d	ɔ	x	t	ɛ	ɾ
Icelandic	d	ɔ	-	t^h	ɪ	z
Indian English (Delhi)	d	ɔ	-	t	ɜ	-
Low German (Achterhoek)	d	ɒ	χ	t	-	ʁ
Low German (Bargstedt)	d	ɔ	χ	t	ɐ	-
New Zealand English (Auckland)	d	ɔ	-	ɾ	ə	-
Nigerian English (Igbo)	d	ɔ	-	t	ə	-
Norwegian (Stavanger)	d	a	-	t^h	ə	ʁ
Scottish	d	ɔ	-	t^h	ə	ɹ
South African English (Johannisburg)	d	ɔ	-	t^h	ɛ	-
Swedish (Skane)	d	o	-	t	-	ʁ
Swedish (Stockholm)	d	ɔ	-	tː	ɛ	ɾ
West Frisian (Grou)	d	ɒ	χ	t	ə	r
Yiddish (New York)	t^h	ɔ	χ	t	ɛ	ɾ

Figure 6: Alignment analysis of 42 words for *daughter* across different Germanic languages and dialects. The alignment was manually prepared as part of the Benchmark Database for Phonetic Alignments (List and Prokić 2014). The visualization was plotted with help of the LingPy Python library for quantitative historical linguistics (List and Moran 2013).

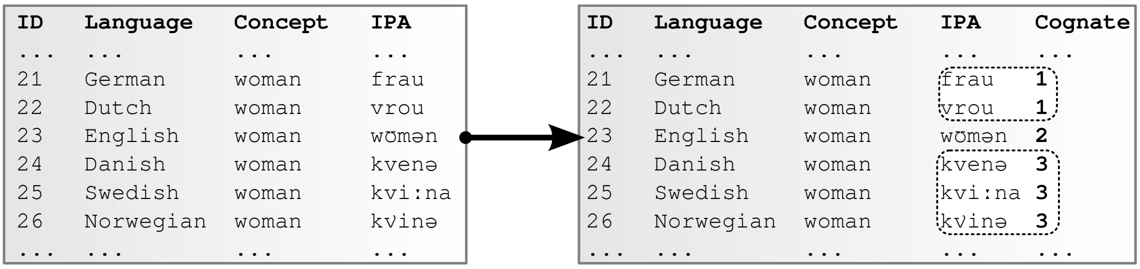
enhanced visualization techniques as they are now available in software packages such as LingPy (List and Moran 2013) where alignments can be plotted as HTML or PDF files with colors highlighting the sound classes of the phonetic values, they offer an immediate look at the diversity in the data. As an example, Figure 6 shows a multiple alignment of 42 words for *daughter* in different Germanic languages, taken from the Germanic subset of the *Benchmark Database for Phonetic Alignments* (List and Prokić 2014) (which is based on Renfrew and Heggarty 2009).

2.2 Cognate Detection

In the previous section we tried to illustrate how classical approaches to historical linguistics could profit from automatic alignment analyses, both as a tool that helps to visualize linguistic data in its complexity and to formalize those assumptions which are so far mostly made implicitly. Taken alignment analyses alone, however, there is not much we can gain when trying to establish computational models of major workflows of the comparative method, since the performance of alignment algorithms relies on what we feed them. So, while an algorithm would align no matter which words we present it, our interest in alignments is restricted to alignments of those words which are actually historically related, that is, words which are *cognate*. This brings us to one of the bigger task of quantitative historical linguistics, which can likewise be considered as one of its “holy grails” (List 2014*b*), the task of *automatic cognate detection*. In the following, we will try to shed some light on the major ideas behind recent automatic approaches to cognate detection, as well as the major challenges which have not yet been sufficiently solved.

2.2.1 The Automatic Cognate Detection Task

In order to get a clearer view on how the cognate detection task can be handled automatically, it is helpful to state it in terms of *input* and *output*, that is, what data we feed to an algorithm, and what data we hope to get back. In the following, we will assume that the input is a *multilingual word list*. A multilingual word list is hereby understood as a list that is organized onomasiologically by giving a set of meanings and their translations in different languages. Since cognate detection deals with phonetic sequences, the translations should be given in some form of phonetic transcription, preferably in IPA. Furthermore, since IPA is often ambiguously used, especially regarding the treatment of affricates, which may often resemble two sounds (compare [ts] which is used to denote the affricate [t͡s] and the the combination of [t] with [s]), but also regarding certain diacritics (compare [ʰ] which may denote pre- and post-aspiration), we will assume that the phonetic transcription is explicitly segmented, for example, by using a space to mark phoneme boundaries. Regarding the concepts in the word list, we can think of a typical *Swadesh list*, like the 200 item list proposed by Swadesh (1952), but it should be clear that in many cases, 100 or 200 items may just not provide enough information to sufficiently identify cognates and sound correspondences (List 2014*a*). For the output we want to have when applying an algorithm for automatic cognate detection is a clustering of all words in the data into sets of cognate words. For the sake of simplicity, we will assume that cognate sets will be restricted to words denoting the same meaning, but it is clear that ultimately, it would be desirable to search for all cognates in the data regardless of the meaning of the words, since according to the classical definition of cognacy, cognate words do by no means need to have the same meaning (Trask 2000:64). The fundamental input and output requirements for the automatic cognate detection task are illustrated in Figure 7.



ID	Language	Concept	IPA
...	...	...	...
21	German	woman	frau
22	Dutch	woman	vrou
23	English	woman	wʊ̆mən
24	Danish	woman	kvenə
25	Swedish	woman	kviːna
26	Norwegian	woman	kʋinə
...	...	...	...

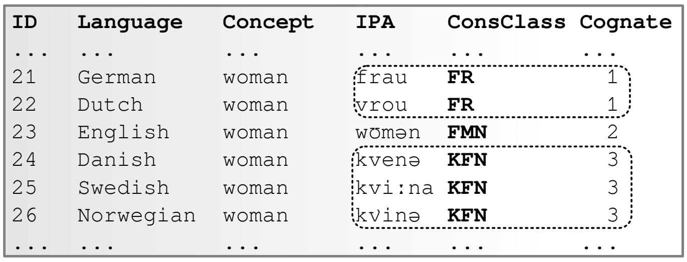
ID	Language	Concept	IPA	Cognate
...	...	...	...	...
21	German	woman	frau	1
22	Dutch	woman	vrou	1
23	English	woman	wʊ̆mən	2
24	Danish	woman	kvenə	3
25	Swedish	woman	kviːna	3
26	Norwegian	woman	kʋinə	3
...	...	...	...	...

Figure 7: Input and output of the automatic cognate detection task. The input is a multilingual word list with words reflecting the translations of a set of meanings into different languages. The output is a word list in which words with the same meaning are clustered into cognate set. Clustering decisions are represented by adding cluster numbers in the “Cognate” column. Words with the same cluster number are assigned to the same cognate sets.

2.2.2 Basic Approaches to Automatic Cognate Detection

Essentially, cognate detection is a *clustering task*, since the goal is to cluster words into cognate classes. More precisely, it is a *partitioning task*, since we do not necessarily assume any hierarchical ordering inside or among the different classes of cognate words, we only want to have the different parts of the data, as if we cut a piece of paper into different pieces.

There are different ways how a partitioning of words into cognate classes can be achieved. A first and very early approach was presented in Dolgopolsky (1964) and is based on the above-mentioned idea of sound classes. When using a very rough sound class system, as the system of ten consonant classes proposed by Dolgopolsky, one could assign all words to the same cognate set which share the same sound classes. This idea was later followed up by scholars from the comparative linguistics circle in Moscow and even implemented as part of the STARLING database system Burlak and Starostin (2005:270-275). As a general rule, all approaches assign words which match in their first two consonant classes to the same cognate set.[2] Turchin et al. (2010) employed this *Consonant Class Matching* approach (CCM), using a modified sound class model of 9 consonant classes, along with additional probability tests to test the Altaic hypothesis. Kassian et al. (2015) use the same approach to test deeper relations between Indo-European and Uralic languages. In both cases, scholars reported a low rate of false positives produced by this method. This was confirmed in List (2012*a*), where an explicit comparison of the CCM approach and alternative approaches was carried out. However, this study also showed that the CCM approach tends to produce many false negatives, that is, it misses many valid cognates. Figure 8 illustrates this method by showing how it would cluster the data of Figure 7 into cognate sets. Implementations of the method are currently online available as part of the STARLING software package (Starostin 2000), online at `http://starling.rinet.ru`, and as part of the LingPy Python library (List and Moran 2013), online available at `http://lingpy.org`.



ID	Language	Concept	IPA	ConsClass	Cognate
...	...	...	...	...	...
21	German	woman	frau	**FR**	1
22	Dutch	woman	vrou	**FR**	1
23	English	woman	wʊ̄mən	**FMN**	2
24	Danish	woman	kvenə	**KFN**	3
25	Swedish	woman	kviːna	**KFN**	3
26	Norwegian	woman	kʋinə	**KFN**	3
...	...	...	...	...	...

Figure 8: The Consonant Class Matching method for automatic cognate detection

The major advantage of the CCM approach is its simplicity. As a result, computation is really fast, which makes it a perfect method to be applied to very large datasets or inside lightweight computer-assisted workflows in which linguists first use an automatic approach to search for cognates and then manually correct the results. The major drawback of the CCM approach is that it misses many valuable cognate sets. This lack in resolution power results from two problems: First, consonant classes are treated as absolute entities which can only be identical or different. Second, restricting the matching consonant classes to the first two consonants of the words deprives the approach of valuable information. Comparing English *daughter* and German *Tochter*, for example, the CCM method will classify both words as not being cognate with each other, since the first consonant classes of the former ("TT") do not match those of the latter ("TK"). Using an alignment algorithm instead of the static matching procedure, for example, would immediately show that there are two valuable matches of "T" and one mismatch of "K" in German, which might give a good hint regarding common ancestry of the words. As another example, consider English *tooth* ("TT") and German *Zahn* ("KN"), which look completely different regarding their consonant class representation although the sound changes between both words are completely regular. While it seems useful to state a certain closeness between alveolar affricates and velars, it would be at least as useful to state a closeness between alveolar

[2] Word-initial vowels are hereby assigned to the same consonant class as word-initial glottal stops.

affricates and alveolar stops.

An alternative family of approaches to cognate detection circumvents these problems by first calculating distances or similarities between all word pairs in the data, and then feeding those distance scores to a flat clustering algorithm which partitions the words into cognate sets. This workflow is very common in evolutionary biology, where it is used to detect homologous genes and proteins (Bernardes et al. 2015). While distances can be calculated in many different ways, the most straightforward way to calculate them is to use pairwise alignment analyses. Many algorithms for data partitioning based on pairwise distance matrices are available in the literature. One possibility here is to employ hierarchical clustering algorithms like UPGMA (Sokal and Michener 1958) and terminate them once a certain threshold of pairwise similarities or distances is reached. Another possibility is to use graph-based partitioning algorithms (Andreopoulos et al. 2009). In these methods, words are represented as nodes in a network and links between them are drawn when the pairwise similarity exceeds a certain threshold. Graph-based clustering algorithms then further try to partition the nodes in the network into groups by adding or removing links (Frey and Dueck 2007, van Dongen 2000). Figure 9 gives an illustrative example on how the words shown in Figure 7 can be clustered into cognate sets with help of a flat hierarchical cluster algorithm.

	German	English	Danish	Swedish	Dutch	Norwegian
German [frau]	0.00	0.95	0.81	0.70	0.34	1.00
English [wʊmən]	0.95	0.00	0.78	0.90	0.80	0.80
Danish [kvenə]	0.81	0.78	0.00	0.17	0.96	0.13
Swedish [kvinːa]	0.70	0.90	0.17	0.00	0.86	0.10
Dutch [vrɑu]	0.34	0.80	0.96	0.86	0.00	0.89
Norwegian [kʋinə]	1.00	0.80	0.13	0.10	0.89	0.00

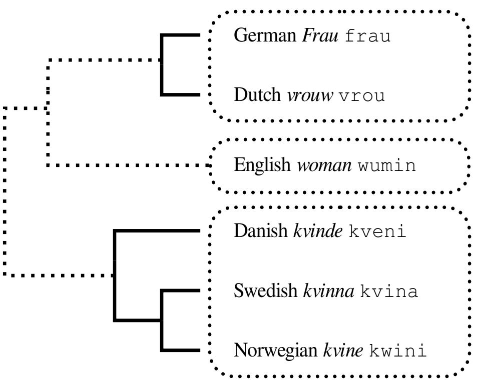


Figure 9: Flat clustering strategy to automatic cognate detection. Pairwise word distance on the left are used to derive an hierarchical cluster of the words. The clustering process stops when a certain threshold is reached (here indicated by dotted lines in the cluster on the right).

To our knowledge, Bergsma and Kondrak (2007) were the first to present a cognate detection approach based on a clustering algorithm applied to pairwise word distances. Their approach was based on the *longest common subsequence ration*, which is derived from a global pairwise alignment of two strings by dividing the number of identical sounds in two words by the length of the longer word. They then use an integer linear programming approach to partition the words into cognate sets. Steiner et al. (2011) compute Needleman-Wunsch alignments between all word pairs in a meaning slot and then use a cluster algorithm which is not further specified for the task of cognate partitioning. Their approach is interesting in so far, as it is part of an iterative pipeline which learns scores from pairwise alignments and even searches for cognate sets across different meanings in the word list. Hauer and Kondrak (2011) employ a machine learning approach that is trained with different pairwise sequence similarities to decide whether two words are cognate or not. They then use a flat version of the UPGMA clustering algorithm that terminates when clusters reach a certain threshold of average similarities. List (2012*a*, 2014*b*) employs a similar flat clustering algorithm but computes word similarities with help of an iterative approach that first searches uses global and local alignment analyses to search for potential sound correspondences in all language pairs and then uses these pairs to derive a language-specific scoring function. This function is used to realign all words, and the alignment scores are then passed to the scoring function. List (2014*b*) compares this LexStat approach with the CCM approach and two further clustering approaches, one based on the *normalized edit distance* (NED) and one based on distances derived from SCA alignments. The test, carried out on a gold standard of expert cognate judgments on six datasets covering five language families and a total of more than 16 000 words showed that the LexStat performed best, followed by the SCA and the NED clustering. The CCM method performed worst, due to a very high rate of false negatives. Jäger and Sofroniev (2016) develop an approach where various variables derived from string similarities computed by means of Jäger's (2013) PMI algorithm are used for supervised training of a *Support Vector Machine* (a machine-learning algorithm for automatic classification), trained with a collection of data manually annotated for cognacy. This method performed slightly better than the LexStat approach when applied to unseen data.

Hall and Klein (2010, 2011) and Bouchard-Côté et al. (2013) present an alternative family of approaches to cognate detection which is essentially based on a phylogenetic model that reconstructs how words evolved along a phylogenetic tree, distinguishing between mutations (instances of sound change during which the word retains its cognate class) and innovations (lexical replacement). The authors describe different models of varying complexity, ranging from simple global alignments up to complex models which may even include rudimentary ways to handle phonetic context (Bouchard-Côté et al. 2013). All of these approaches requires a reference phylogeny of the languages under investigation to be known in advance. Due to the complexity of the problem of detailed evolutionary scenarios for the development of characters along a tree, they also require the use of sophisticated machine learning techniques. The authors present flavors of this basic idea and test it on different datasets for Austronesian languages, reporting high scores of cognate recovery. In addition to cluster-based approaches to cognate detection or variants of the CCM method, these phylogeny-based approaches to cognate detection also reconstruct ancestral word forms, which makes it possible to test the realism of the models by comparing reconstructions based on the comparative method with the automatically produced reconstructions.

All approaches to cognate detection mentioned above have their advantages and disadvantages. The CCM method is very easy to understand, very straightforward to implement in software, and very fast in application. It is thus the recommended method for large datasets which are not analyzable with help of complex and time-consuming algorithms, but also very useful for computer-assisted workflows in which all automatically computed output is manually corrected by trained experts. The drawback of the CCM method is its high rate of false negative judgments. Cluster-based approaches to cognate detection offer a more elaborated alternative to CCM approaches. They usually outperform the CCM method, but their lower rate of false negatives may come to the price of a higher rate of false positives, especially when naive alignment algorithms, such as the normalized edit distance are being used. The increased complexity requires longer computation times, which makes it difficult to integrate the methods in lightweight applications for computer-assisted frameworks. The increased accuracy, which may reach almost perfect agreement with human experts in smaller datasets of shallow time depths (List 2014*b*), however, is a great advantage, especially in exploratory applications of understudied language families. Phylogeny-based approaches to cognate detection are the computationally most advanced of the methods which have been proposed so far. Their advantage is their explicitness regarding the processes they model, and their output which does not only yield decisions regarding the cognacy of words, but also distinct evolutionary scenarios regarding the way the words in the data evolved into their current shape. Their disadvantage is their complexity which requires the application of complex and time-consuming machine learning approaches. Furthermore, phylogeny-based approaches cannot be applied for the purpose of data-exploration, since they require all languages in the data to be known to be related. While the LexStat method, for example, could be used to test a relationship hypothesis between two or more languages (List 2014*b*:203-205), this is not possible for phylogeny-based approaches to cognate detection.

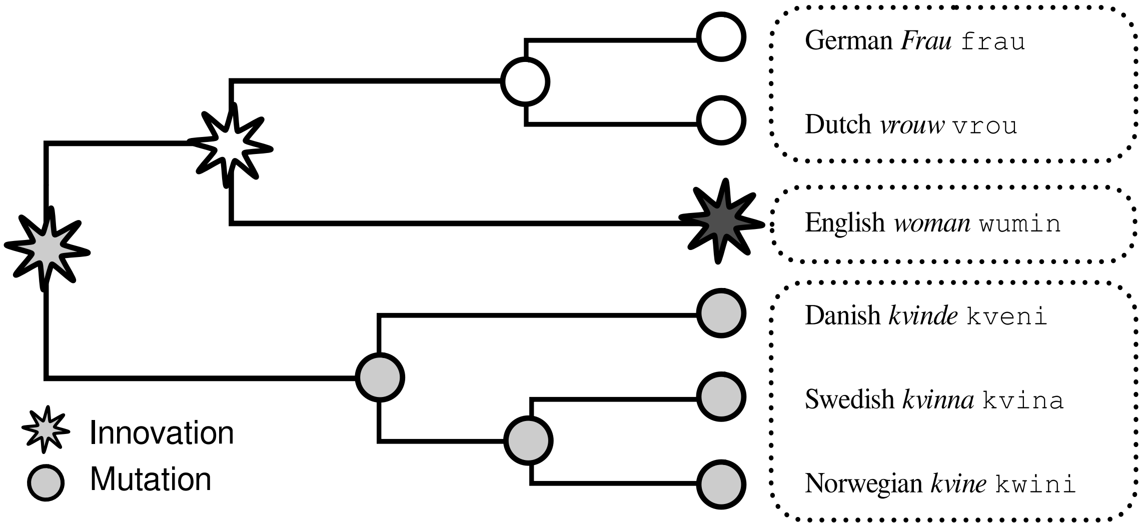


Figure 10: Phylogeny-based approach to cognate detection. Given a phylogeny (a tree), the method tries to identify optimal scenarios of character evolution by which words can either mutate (change their sound shape slightly) or innovate (be replaced by another word).

2.2.3 Future Challenges for Automatic Cognate Detection

The methods for automatic cognate detection which have been proposed so far are definitely promising and can already in their current state provide great help, especially in exploratory data analysis, but also in computer-assisted approaches to the comparative method. They rest, however, a couple of serious shortcomings which future research needs to address. As first problem to mention in this context is the problem of borrowing: None of the methods proposed so far can sufficiently handle borrowing. Language-specific (as opposed to language-independent) methods which are based on the computation of individual sound correspondences between language pairs, can rule out sporadic borrowings between languages, but they also fail when borrowing is intense. Possible solutions would require a *stratification analysis* in which sound correspondences for different parts of the lexicon are investigated and the resulting correspondence patterns compared. List (2014*a*:98f) illustrated for a dataset of English, German, Dutch, and French, that stratification analyses in which sound correspondences are only inferred for stable parts of the lexicon and then used to detect cognates across all data could help to drastically reduce the amount of erroneously classified borrowings from French to English. However, these results came at the cost of a generally increased rate of false negatives.

A further challenge are the different *shades of cognacy* which can be observed in lexical datasets (List 2016*a*). While all algorithms model cognacy as a distinct relation between words which is either present or absent, words can exhibit many more degrees of relatedness. Comparing French *soleil* with Italian *sole* 'sun', for example, it is clear that the words are cognate. While *sole*, however, goes directly back to Latin *sol*, *soleil* goes back to Vulgar Latin *soliculus* 'small sun' which itself is a derivation of *sol* (Meyer-Lübke 1911). Morphological processes which shape the form of words results in *unalignable parts* among cognate words. Apart from the secondary alignment algorithm (see Section 2.1.1) which allows to force an alignment algorithm to avoid the matching of one morpheme with two or more other morphemes, no further methods which take unalignable parts into account have been proposed so far.

A last challenge is the unification and propagation of common formats and open software applications in the field of computational historical linguistics. The majority of the proposed methods for phonetic alignment and cognate detection which have been proposed in the past have never been published in form of software packages. So far, the only approaches to cognate detection which are online available are the CMM approach which is implemented in the STARLING software package (Starostin 2000) and the LexStat approach, which is, along with other methods, implemented in the LingPy software package (List and Moran 2013). The same applies for benchmark datasets. While many methods have been tested in comparison with gold standards, there is no study which would compare the performance of the methods on the same gold standard. In addition, the majority of the tests which have been carried out did not publish the data which would be needed to replicate the analyses. In order to increase the replicability of research in the quickly evolving field of quantitative historical linguistics, it is indispensable that scholars change their attitude and start to publish data and source code along with their research papers.

3 Phylogenetic Reconstruction

3.1 General remarks

Phylogenetic reconstruction is the task to infer a family tree from language data. Computational biologists have developed a rich toolbox for the corresponding task of reconstruction evolutionary history from biomolecular or morphological data. Most of these tasks are, *mutatis mutandis*, applicable in computational historical linguistics as well.

On a general level, computational phylogenetic reconstruction has the same goal as family tree reconstruction according to the comparative method. Both approaches strive at construction of tree diagrams, with observed languages at the leaves, where internal nodes represent inferred historical language stages. The adequacy criteria for computationally derived phylogenies are somewhat different though from those for traditional family trees, which has to be kept in mind when interpreting the results.

3.1.1 Phylogenetic trees

Phylogenetic trees come in two varieties, as *unrooted* or *rooted* trees. Mathematically speaking, an *unrooted phylogenetic tree topology* is a connected undirected acyclic graph. Nodes with a degree 1 (i.e., nodes only connected to one branch) are called *leaves* or *tips*. A *rooted phylogenetic tree topology* is an unrooted topology where one node is designated as root. An unrooted topology is *binary branching* if all nodes except the tips have degree 3. Similarly, a rooted topology is binary branching if the root has degree 2 and all other nodes have degree 1 or 3. Figure 11 shows an unrooted and a rooted topology. The rooted topology is obtained from the unrooted one here by adding a root node on the branch from C to D.

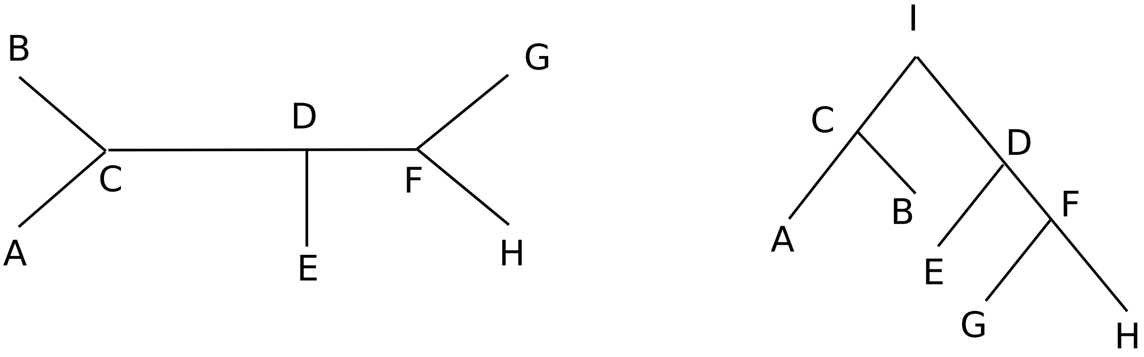


Figure 11: An unrooted (left) and a rooted (right) tree topology

In an unrooted topology, each branch induces a binary split between the set of leaves. For instance, in the left tree in Figure 11, the branch from D to F splits the leaves into the set {A,B,E} and {G,H}.

This is to be interpreted as the claim that the two sets differ with respect to some feature(s). An unrooted topology is agnostic though with regard to the direction of the time arrow. It is left open which of the two sets represents an innovation and which one a retention.

In a rooted toplogy, time flows from root to leaves.

A *phylogenetic tree* (unrooted or rooted) is obtained from a topology by assigning a (non-negative) length to each branch.

The interpretation of branch lengths is sometimes subtle. They only have a well-defined meaning in connection with a quantity r, expressing the *rate of change*. If t is the length of a branch, $r \cdot t$ expresses the amount of change that happened along that branch. How this relates to historical time depends on how much r varies between lineages.

In practice, whenever historical dating is not an issue, r is assumed to be constant, and branch lengths can directly be interpreted as a measure of the amount of change. In studies dealing explicitly with historical dating though (such as, e.g., Bouckaert et al. 2012), branch lengths express assumed historical time (and r is assumed to vary across branches).

3.1.2 Inferring trees

The literature contains a vast variety of methods for phylogenetic inference, differing both with regards to the type of data used and the required computational resources. Due to space limitations, we will discuss only two extreme ends of the spectrum here in any detail:[3] *Neighbor Joining* (Saitou and Nei 1987), a highly efficient method which is applicable to a wide range of data types, and *Bayesian phylogenetic inference*, which is highly resource intensive and requires a highly specific type of data. The latter method affords a much richer and more fine-grained interpretation than the former.

For concretness' sake, we defined three data collections over a small sample of languages, which will be used as running examples throughout this section. The sample of languages consists of twenty-five Indo-European languages: *Bengali, Breton, Bulgarian, Catalan, Czech, Danish, Dutch, English, French, German, Greek, Hindi, Icelandic, Irish, Italian, Lithuanian, Nepali, Polish, Portuguese, Romanian, Russian, Spanish, Swedish, Ukrainian*, and *Welsh*. We use three types of data:

- Swadesh lists in IPA transcription, taken from the *Indo-European Lexical Cognacy Database* (IELex; `http://ielex.mpi.nl/`, accessed on April 6, 2016),
- expert cognate classifications of Swadesh list entries (likewise taken from IELex),[4] and
- phonological, grammatical and semantic classifications of languages (taken from WALS, `http://wals.info/`, accessed on April 2, 2016; Haspelmath et al. 2008).

A small subset for each of those data collections are displayed Table 3 for illustration.

3.2 Distance-Based versus Character-Based Methods

Depending on the type of input data, phylogenetic algorithms fall into two categories. *Distance-based* methods operate on a matrix of pairwise distances between the languages to be classified. The distance between two languages is a measure of the amount of divergent changes that occurred in the two lineages since their latest common ancestor. A distance measure is useful for this purpose if on average, the distance between two languages grows monotonically with the combined time between their latest common ancestor.

[3] For a comprehensive treatment, the interested reader is referred to (Felsenstein 2004) or Section III of (Lemey et al. 2009).

[4] We only included those entries from IELex where both an IPA transcription and a cognate classification is given.

language	*phonological form* (IELex)	*cognate class* (IELex)	*order of subject, object and verb* (WALS)
Bengali	-	-	SOV
Breton	-	-	SVO
Bulgarian	muˈrɛ	sea:B	SVO
Catalan	mar; maɾ; ma	sea:B	SVO
Czech	ˈmɔr̝ɛ	sea:B	SVO
Danish	hɑw/søˀ	sea:K/sea:J	SVO
Dutch	ze	sea:J	no dominant order
English	si:	sea:J	SVO
French	mɛʀ	sea:B	SVO
German	ze:/ʼo:t͜sea:n/me:ɐ̯	sea:J/sea:E/sea:B	no dominant order
Greek	ˈθalaˌsa	sea:F	no dominant order
Hindi	-	-	SOV
Icelandic	haːv/sjouːr	sea:K/sea:J	SVO
Irish	ˈfˠæɾˠɟɪ	sea:G	VSO
Italian	ˈmare	sea:B	SVO
Lithuanian	ˈjuːrɐ	sea:H	SVO
Nepali	-	-	SOV
Polish	ˈmɔʐɛ	sea:B	SVO
Portuguese	maɾ	sea:B	SVO
Romanian	ˈmare	sea:B	SVO
Russian	ˈmɔrʲɛ	sea:B	SVO
Spanish	maɾ	sea:B	SVO
Swedish	hɑːv/ɧøː	sea:K/sea:J	SVO
Ukrainian	ˈmɔrɛ	sea:B	SVO
Welsh	-	-	VSO

Table 3: Phonetic, cognacy and typological data

Character-based methods require as input a *character matrix*, i.e. a matrix with languages as rows and *discrete characters* as columns. A discrete character is a classification criterion with a finite number of possible values. Each character must have the same range of possible values. The cognate classifications from IELex and the typological classifications in WALS violate the latter condition. They can easily be transformed into the required format though by converting them into binary format. This is achieved by treating each feature-value pair F, v as a character, with the value "1" for languages having value v for feature F, and "0" otherwise.[5]

To be useful for phylogenetic inference, a character must display a certain diachronic inertia without being completely invariant. Furthermore, phylogenetically informative character should have the *Phylogenetic Markov Property*. This can be illustrated with a schematic example. Consider the left tree topology in Figure 11, and let F be a character, with $F(L)$ being the value of F for language L. The Phylogenetic Markov Property demands that once we know which value $F(C)$ has, we cannot learn anyting about $F(A)$ from $F(B)$, $F(D)$, or $F(E)$. Formally, we have[6]

Phylogenetic Markov Property *Let $\mathcal{T}$ be a phylogeny and X, Y, and Z nodes of $\mathcal{T}$ such that Y is on the path from X to Z. Let F be a character.*

F has the Phylogenetic Markov Property *if and only if*

$$F(X) \perp\!\!\!\perp F(Z)|F(Y).$$

This condition is, for instance, violated if languages A and B have been in contact after they diverged, and A borrowed a character value from B.

Since few, if any, linguistic variables are immune to borrowing, it is questionable whether this condition is ever fully satisfied if the languages considered have been in contact. As we will see below, this does not preclude the applicability of phylogenetic inference, but language contact should be kept in mind as biasing factor when interpreting the results.

3.3 Distance-Based Methods: *Neighbor Joining*

A phylogenetic tree implicitly defines a pairwise distance between any pair of leaves as the length of the path between those two leaves. If matrix pairwise distances between languages is given (i.e., has been obtained from empirical data), the fit of a tree to the empirical data is the better the more the empirical distances coincide with the distances predicted by the tree. Distance-based phylogenetic inference is the problem to find a phylogenetic tree that has a good fit with a given matrix of distances. For brevity's sake, we will only discuss one instance of this general paradigm (there is a plethora of alternative though). Before that, we will consider the question how linguistic distances can be obtained.

3.3.1 Computing linguistic distances

Depending on the data being considered, there is a multitude of ways to estimate linguistic distances. We will consider one such method for each of our three data types.

If we work with word lists in phonetic transcriptions, a frequently used approach is to start with string similarity scores that can be obtained from pairwise sequence alignment (see Subsection 2.1.1). Similar approaches to phylogenetic inference have been used, among others, in (Holman et al. 2008*a*).

[5] If the value of F for some language L is undefined — either because F is not applicable to L or because the value is unkown —, L's value is undefined for all binary characters derived from F as well.

[6] The notation "$a \perp\!\!\!\perp b|c$" means "a and b are conditionally independent given c.

In (Jäger 2013, 2015) one such method is proposed which will briefly be discussed here.

The underlying intuition is the following: Suppose Swadesh lists for two languages, A and B, are given, but they are unordered and the word meanings are not known. If A and B are closely related, it is easy to guess which words are translations of each other because these word pairs will be cognate and therefore phonetically similar. When A and B diverge further, the similarity between cognate pairs will decrease due to phonetic change, and some words will be replaced by non-cognate words. Generally, the task of spotting translation pairs in the two word lists will be the harder the more time has passed since the latest common ancestor of A and B.

This idea is operationalized as a quantitative measure in the following way. All word pairs from A and B are arranged in order of decreasing string similarity. String similarity is determined via globale pairwise alignment using the parameters proposed in (Jäger 2015). (For illustration, the first fifteen pairs from this list for Russian/Lithuanian are shown in Table 4. The 1st, 14th and 20th line are translation pairs.) The assignment of translation pairs can be coded as the list of ranks of the rows with matching meanings in this list. Each rank k can be coded as a binary number with $\lceil \log_2 k \rceil$ digits, so in average we need $\sum_{i=1}^{n} \lceil \log_2 r_i \rceil / n$ bits to encode one translation pair (where n is the number of translation pairs and r_i the rank of the ith translation pair in the list). If no information about string similarities were given, the same information would have to encoded on the basis of some arbitrary order, which requires on average a number of $\lceil \log_2 N \rceil - 1$ bits, where N is the total number of word pairs. So by utilizing string similarities, we save $(\sum_{i=1}^{n} \lceil \log_2 N \rceil - 1 - \lceil \log_2 r_i \rceil) / n$ many bits. For larger values of n, this value can be approximated by the following formula, which gives a measure of the similarity between the languages in question (the subscript p indicates "phonetic"):

Russian word	*Russian meaning*	*Lithuanian word*	*Lithuanian meaning*	*similarity score*	*similarity rank*
s^{j}i’d^{j}ɛt^{j}	**‘sit’**	**s^{j}e:ˈd^{j}e:t^{j}ɪ**	**‘sit’**	**8.57**	**1**
zv^{j}ɛz’da	‘star’	lɐzˈdɐ	‘stick’	7.23	2
tʃɛrv^{j}	‘worm’	tʃ^{j}ɛ	‘here’	6.45	3
v^{j}a’zat^{j}	‘tie’	ˈvardas	‘name’	6.16	4
dut^{j}	‘blow’	ˈduɔt^{j}ɪ	‘give’	5.56	5
stʃi’tat^{j}	‘count’	ˈʃ^{j}ɪltɐs	‘warm’	5.11	6
kal^{j}ɛ’nɔ	‘knee’	ˈka:lnɐs	‘mountain’	5.08	7
’dumat^{j}	‘think’	ˈduɔt^{j}ɪ	‘give’	5.01	8
zam^{j}ɛr’zat^{j}	‘freeze’	m^{j}ɛˈd^{j}ʒ^{j}ot^{j}ɪ	‘hunt’	4.99	9
r^{j}ɛ’ka	‘river’	ɾɐŋˈkɐ	‘hand’	4.97	10
p^{j}at^{j}	‘five’	pɐs	‘at’	4.92	11
’małɔ	‘few’	ˈma:ʒɐs	‘small’	4.77	12
tʃɛ’sat^{j}	‘scratch’	tʃ^{j}ɛ	‘here’	4.74	13
sn^{j}ɛg	**‘snow’**	**s^{j}n^{j}jægɐs**	**‘snow’**	**4.67**	**14**
vɔ’da	‘water’	ˈvardas	‘name’	4.67	15
da’vat^{j}	‘give’	ˈt^{j}e:vɐs	‘father’	4.66	16
zɔˈła	‘ashes’	ʒoˈl^{j}e:	‘grass’	4.50	17
tra’va	‘grass’	ˈt^{j}e:vɐs	‘father’	4.41	18
bɔˈrɔt^{j}s^{j}a	‘fight’	ˈba:ltɐs	‘white’	4.31	19
um^{j}i’rat^{j}	**‘die’**	**ˈm^{j}iɾ^{j}t^{j}ɪ**	**‘die’**	**4.30**	**20**

Table 4: Word pairs from Russian/Lithuanian, arranged according to string similarity

$$\text{sim}_p(A, B) \doteq \frac{\sum_{i=1}^{n} - \log_2 \frac{r_i}{N}}{n} - 1$$

The similarity between A and B is maximal if $A = B$. Empirically, this value is in the range between ca. 6 and 12. For entirely unrelated languages, the expected value is 0. By making the natural assumption that similarity converges with an exponential rate towards 0 with decreasing divergence time between A and B, we arrive at the following estimate for the divergence time:

$$d_p(A,B) = -\log \frac{\text{sim}_p(A,B)}{\max_{A',B'} \text{sim}_p(A',B')}$$

This method is perhaps somewhat reminiscent of Greenberg's (e.g., Greenberg 1987) "lexical mass comparison" since it operates on superficial string similarities regardless of regular sound correspondences or other linguistic analyses. Our method avoids two of Greenberg's pitfalls though: It explicitly takes chance resemblances into account and calibrates string similarities accordingly, and it only considers word pairs with identical meanings. This considerably reduces the impact of false positives. (See Jäger 2015 for more discussion of this issue.)

Deriving a distance measure from data in character format is more straightforward, even though there is a variety of options here as well, depending on one's assumptions about the dynamics of character evolution.

For the purpose of illustration, let us consider a binary character matrix. For simplicity's sake, we assume that characters change there value according to a specific mutation rate (i.e. character values follow a continuous time Markov process) that mutation rate is equal between characters and between lineages. Furthermore we assume that a change $1 \rightarrow 0$ is common while the inverse change is negligibly rare. This makes sense for instance if characters are cognate classes, as by definition, a cognate class cannot emerge more than once in a tree (except via borrowing), while it might be lost in multiple lineages. For presence/absence of WALS-style feature values, this assumption is less obvious. It might still be a viable approximation as most WALS features have several values, so the probability of loosing a value is much higher than the probability of gaining it.

Let A and B be two languages. We only consider characters with a defined value for both languages. Let 1_A be the set of characters for which A has value 1, and likewise for 1_B. The divergence time between A and B can then be estimated[7] as

$$d_c(A,B) \doteq -\log \frac{|1_A \cap 1_B|^2}{|1_A| \cdot |1_B|}.$$

3.3.2 Computing the Neighbor Joining tree

The *Neighbor Joining* algorithm (Saitou and Nei 1987) is an agglomerative algorithm taking a matrix of pairwise distances over some set of taxa (e.g., languages) as input and computing an unrooted phylogenetic tree over those taxa as output. It can be informally sketched (discussing the precise mathematical formulation would go beyond the scope of this chapter; see for instance Lemey et al. 2009, pp. 150–153, for a detailed explanation) as follows:

- **Start** with a distance matrix D (where d_{ij} is the distance between trees i and j) and a collection of trees $\mathcal{T}$ where each tree consists of just one taxon.
- **While** $|\mathcal{T}| > 1$, **do**
  - Pick the pair of trees (a,b) such that d_{ab} is small but d_{ac} and d_{bc} are for all $c \neq a, b$, on average, large.

[7] Derivation: Suppose a character F has value 1 in B. By assumption, it must have had value 1 in the latest common ancestor of A and B, since there are no mutations $0 \rightarrow 1$. Let $t(A)$ be the time depth of A since its latest common ancestor with B. The probability of value 1 is an exponentially decreasing function of $t(A)$, i.e. $P(F(A) = 1|F(B) = 1) = e^{-rt(A)}$ for some constant rate r. This probability can be estimated as the relative frequency of characters having preserved value 1 in A, i.e. $|1_A \cap 1_B|/|1_B|$. Therefore $t(A)$ can be estimated as $-\frac{\log(|1_A \cap 1_B|/|1_B|)}{r}$. By the same argument, we have $t(B) = -\frac{\log(|1_A \cap 1_B|/|1_A|)}{r}$. Therefore $t(A) + t(B) = -\frac{\log |1_A \cap 1_B|^2/(|1_A| \cdot |1_B|)}{r}$. Dropping $1/r$ only changes this estimate by a constant factor.

  - Construct a new tree x with a root and a and b as its daughters. Compute the lengths of the new branches and d_{xc} for all other trees $c \in \mathcal{T}$.
  - Remove a, b from $\mathcal{T}$ and the corresponding rows and columns from D.

- **Output:** the single member of $\mathcal{T}$.

The output is always an unrooted binary branching tree.

Neighbor Joining is computationally very efficient; computing a tree over several hundreds of taxa does not take more than a few seconds or at most minutes on a modern personal computer. The algorithm is widely used in computational biology. It is included in all standard phylogeny software programs such as *Phylip* (Felsenstein 2005), *Paup** (Swofford 2002), *SplitsTree* (Huson 1998), MEGA (Kumar et al. 2016), or *LingPy* (List and Moran 2013).

The *Neighbor Joining* (NJ) tree is an approximation to the optimal tree according to the *Minimum Evolution* criterion (Gascuel and Steel 2006). This criterion favors trees that minimize the total sum of branch lengths in the tree — i.e., trees that assume a minimal amount of evolutionary change — while maximizing the fit to the input distance matrix. There is no guarantee that the NJ tree is the optimal tree according to this criterion though. Finding this optimal tree is computationally not feasible since there are too many different tree topologies for an exhaustive search as soon as the number of taxa exceeds ca. ten, and there is no more efficient method for this task. However, the NJ is in most cases a very good approximation to the optimal tree.

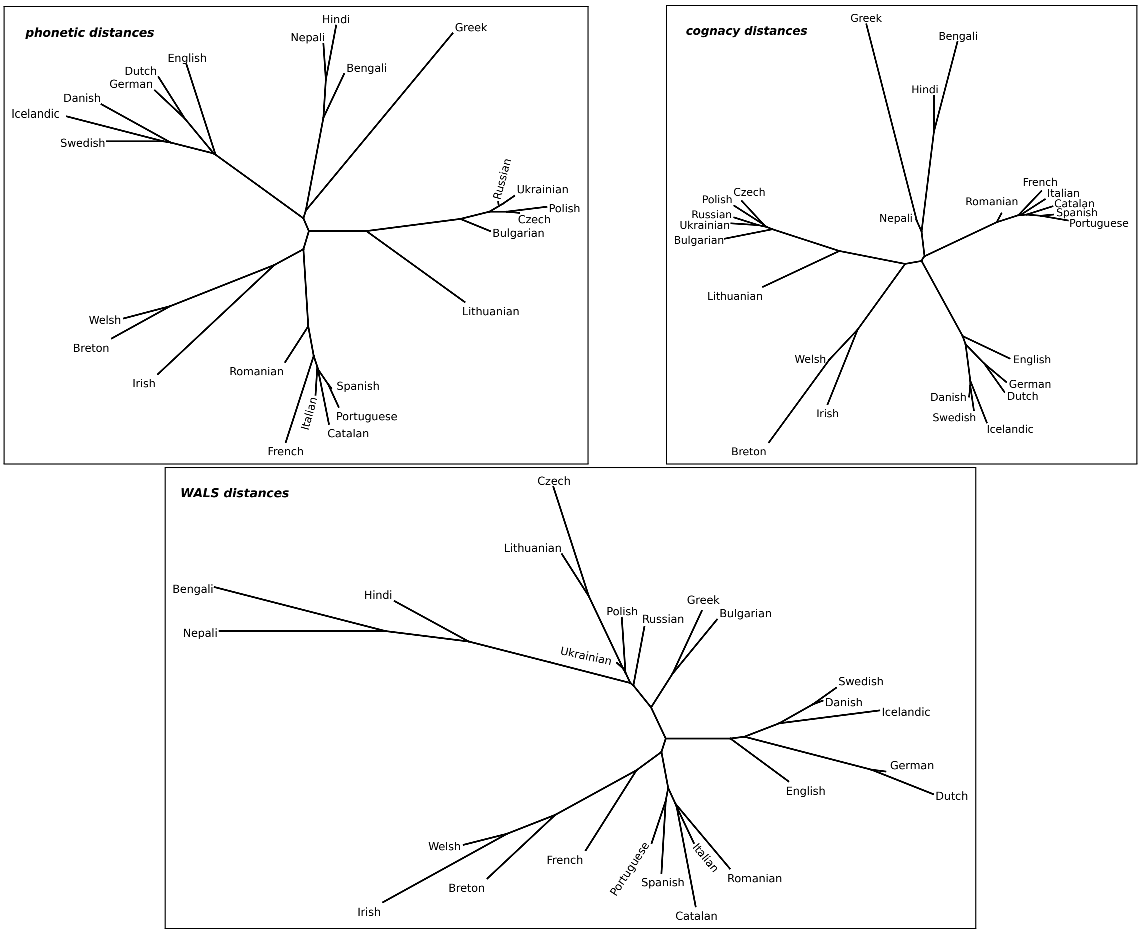


Figure 12: Neighbor Joining trees

Figure 12 shows the NJ trees for the three distance matrices from our running example. All three trees largely identify the established subgroupings of Indo-European correctly. There are some noteworthy deviations in detail though. To mention the most conspicuous ones: All three trees treat English as an outlier within the Germanic branch. According the phonetic and the cognacy tree, French is an outlier within Western Romance, and according to the WALS tree, French is even closer to the Celtic than to the remaining Romance group. In the cognacy tree, Nepali is located on the Greek rather than the Indic branch, and in the WALS tree, Lithuanian is deeply embedded within the Slavic branch, with Czech as its closest neighbor. Also, in the WALS tree Bulgarian and Greek form a group, and Russian is excluded from the Slavic group as well.

There are various possible reasons for such descrepancies. Groupings in an automatically inferred phylogeny mostly reflect common ancestry, but they may also indicate language contact or convergent evolution. Last but not least, they may be statistical artifacts due to an insufficient amount of data.

Branch lengths provide a first heuristics on the reliabilty of groupings. For instance, in the cognacy tree the branch from Nepali to the split from Greek, as well as the branch separating Greek+Nepali from the rest, are very short. This indicates that the location of Nepali in this tree is not strongly supported by the data. The Greek+Bulgarian grouping in the WALS tree, on the other hand, seems to be strongly supported.

The degree of statistical support for phylogenetic branches can be quantified via resampling techniques such as *bootstrapping*. If n data points (such as characters or concepts in Swadesh lists) are available, n data points are drawn from the original data at random *with replacement*. In the resulting sample, some original data points will occur multiple times and others will not occur at all. The resampled data are used for a complete analysis, i.e., distance estimation plus inference of a NJ tree. This procedure is repeated 100 times, leading to 100 trees. If a branch has high support in the data, a corresponding branch (i.e., a branch inducing the same bipartition of languages) will occur in many bootstrap trees, and vice versa. From this collection of trees a *majority consensus tree* is constructed. This is a tree topology which has exactly those branches which occur in at least 50% of the bootstrap trees.

Bootstrap consensus trees for our three data sets are shown in Figure 13. These topologies are to be interpreted as unrooted; they are displayed with the root at the node with the highest degree. Internal branches are annotated with *bootstrap support values*, i.e., the percentage of bootstrap trees having this branch. It turns out that most oddities of the original NJ trees have little statistical support and therefore do not figure in the consensus trees. This holds, e.g., for the groups Nepali+Greek in the cognacy tree, and French+Celtic in the WALS tree. It is also noteworthy that the WALS tree contains neither a Romance nor a Slavic group. On the other hand, a bipartition Balto-Slavic + Indic vs. the Western branches does have strong support in the WALS data. Also, the groupings Czech+Lithuanian and Bulgarian+Greek are well-supported. The latter presumably results from a selection of non-independent features (of the nine character values that are identical for Czech and Lithuanian but different for Polish, say, six pertain to the morphosyntax of negative morphemes). The Bulgarian+Greek grouping arguably reflects language contact within the Balkan Sprachbund.

The fact that Nepali comes out as isolated within Indo-European in the cognacy tree reflects data sparseness. Our data set only contains 16 entries for Nepali (as opposed to, e.g., 192 entries for English). It is noteworthy in this connection though that 16 words are sufficient for a correct classification in the phonetic tree. To appreciate this point, it should be noted that those 16 words contain 73 phonetic segments, as opposed to just 16 cognate class labels. Quite generally, the richness of the phonetic data overall afford a better classification than with cognacy or typological data, even though little linguistic knowledge is reflected in the raw data.

This discussion illustrates an important methodological point. Phylogenetic inference is a valuable tool, but it should not be treated as an infallible oracle. It has major advantages in comparison to manual methods: it does not suffer from a confirmation bias, results are replicable, the degree of

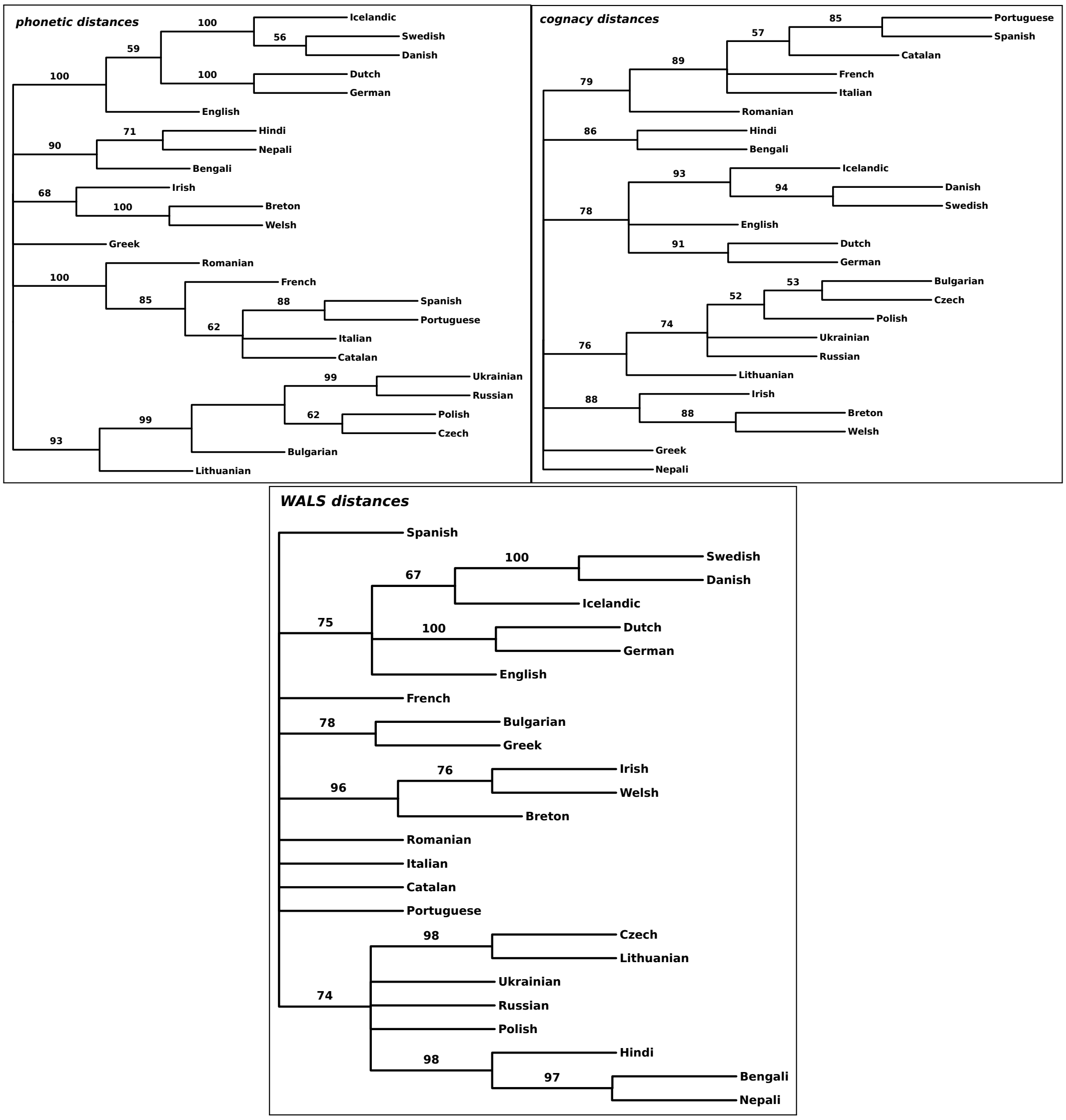


Figure 13: Bootstrap trees

support by the data for a certain hypothesis can be quantified etc. Still, its results are always in need of interpretation, which requires careful inspection of the original data.

Distance-based phylogenetic inference has been applied to linguistic problems, *inter alia*, in (Holman et al. 2008*a*, Jäger 2015, Jäger and Wichmann 2016, Longobardi and Guardiano 2009, Longobardi et al. 2013*a*,*b*), and (Sicoli and Holton 2014).

3.4 Character-Based Methods

Distance-based inference uses character-based data in a sub-optimal way in several respects. When computing pairwise distances from a character matrix, differences between characters are essentially brushed over. Also, the criteria that are optimized in distance-based inference — such as Minimum Evolution with Neighbor Joining — do not have an intuitive interpreation. Perhaps most severely, the output is essentially a black box. We get a phylogeny with branch lengths, but we learn nothing about the behavior of the individual characters in different parts of the tree.

3.4.1 Maximum Parsimony

Character-based inference tracks the behavior of each character and each character value individually. As distance-based inference, character-based inference comes in many flavors. Its simplest incarnation is *Maximum Parsimony* inference (Fitch 1971).

Suppose we have a character matrix plus a rooted tree topology, and a value for each character at each node. This is schematically illustrated in Figure 14. For each branch, the character state at the mother node and the daughter node are compared, and each state combination is assigned a score. In the simplest case, identical states have score 0 and non-identical ones score 1. The sum of all scores then expresses the total number of mutations the character in question underwent for the given scenario. In the example in Figure 14, we would have two mutations, i.e. a score of 2. The sum of the scores for all characters is the mutation score of the given phylogenetic scenario.

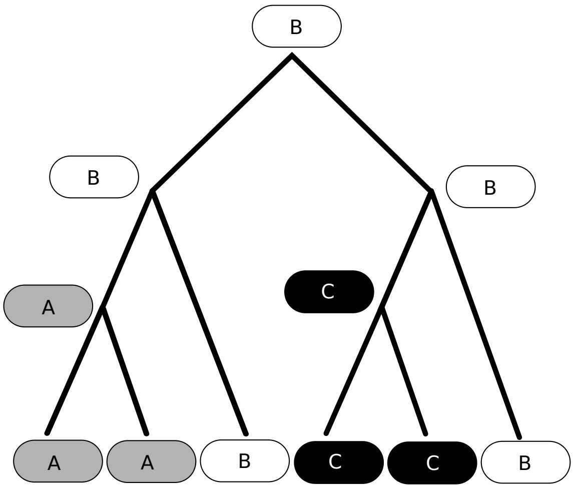


Figure 14: Tree with character states at internal nodes

For a given tree topology, the *length of the tree* is its minimal mutation score consistent with this topology. Even though the number of possible annotations of internal nodes grow exponentially with the size of the tree and the number of characters, this quantity can be computed efficiently (for instance by means of the Sankoff algorithm; cf. Sankoff 1975). The *maximum parsimony topology* is the topology with the smallest length for a given character matrix. Intuitively, it is the ancestral state reconstruction assuming the fewest number of mutations consistent with the data.

There is no efficient way to find this topology. In principle one could go through all topologies over a given set of leaves, compute the length, and then pick the best. However, the number of possible topologies over a given set of leaves grows hyper-exponentially with the number of leaves. For n leaves, the number of unrooted bifurcating topologies is given by the formula (Felsenstein 1978)

$$\frac{(2n-3)!}{2^{n-2}(n-2)!}.$$

For 20 taxa, this amounts to ca. 10^{22} different tree topologies, for 30 taxa ca. 5×10^{38}, for 50 taxa ca. 3×10^{76}. The numer of rooted trees and of non-binary trees grow even faster. Even with modern super-computers, an exhaustive search of the tree space over more than ten leaves or so is not possible.

In practice one uses optimization heuristics to find a tree with a length close to the optimal one. The search algorithm starts with some easy-to-obtain supoptimal but good tree (such as the NJ tree) and modifies the topology locally until no improvement is possible anymore. There is no guarantee though that this local optimum is the globally optimal tree.

Maximum Parsimony inference is implemented for instance in the software packages *Phylip* (Felsenstein 2005) and *Paup** (Swofford 2002). Running an analysis on a personal computer for a medium-sized (several dozens of languages) data-set may take between seconds and hours, depending on the data. As a rule of thumb, Maximum Parsimony is slow if the number of characters is small. While this might be surprising, it is due to the fact that the evaluation of a single topology is fast even for many characters, but the search space is huge. With few characters, there are many ties, which makes heuristic search hard.

A noteworthy application of Maximum Parsimony to infer linguistic phylogenies is (Dunn et al. 2005).

3.4.2 Maximum Likelihood

Maximum Parsimony is conceptually simple and appealing, but it has several drawbacks. All mutations are equally penalized. However, some character are more stable than others, so mutations of the former should be penalized more than the latter. Also, a mutation on a long branch is more likely than one on a short branch, but branch length information is not utilized.

These shortcomings are avoided by the *Maximum Likelihood* framework.[8] Again we start with a rooted tree with character state annotations at the internal nodes, as in Figure 14. Branch lengths are known. Additionally, for each character F, a *rate matrix* Q_F, and a probability distribution over character states at the root are given. The rate matrix determines the probability of a change from the state at the mother node to the state at the daughter node for each character and each branch, depending on the branch length. The overall probability of the observed character states at the leaves is the product of the transition probabilities for all characters and branches. (Since these probabilities are small numbers, in practice one sums over logarithms of probabilities instead.)

The probability of the data given just a tree and a collection of rate matrices and root probabilities, i.e. without state annotations at the internal nodes, is computed as the sum over all possible annotations.

An annotated tree topology plus a rate matrix for each branch is a statistical *model* (M), the branch lengths, rate matrix values and root character state probabilities are *parameters* ($\vec{\theta}$), and the observed characters states at the leaves are the *data* (D). The described method defines the quantity $P(D|M,\vec{\theta})$ — the probability of the data given the model and the parameters. Since the data, but not the correct model and parameter values are given, $P(D|M,\vec{\theta})$ — as a function of M and $\vec{\theta}$ — is called the *likelihood* of $(M,\vec{\theta})$. Maximum Likelihood inference attempts to find the parameterized model with maximal likelihood, i.e., the model best explaining the data.

The class of possible parameter configurations is usually suitably constrained by limiting the possible variation of rate matrices across characters and across branches.

Finding the parameter configuration $\vec{\theta}^*$ which maximizes the likelihood for known M and D can efficiently be done using standard numerical optimization techniques. This gives a maximal likelihood for a tree topology given D. Finding the topology with the maximal likelihood is again an essentially unsolvable problem. As with Maximum Parsimony, implementations of Maximum Likelihood find a locally optimal solution by a heuristic search of the tree space.

Maximum Likelihood inference is implemented for instance in *Phylip* (Felsenstein 2005), *Paup** (Swofford 2002), and *SplitsTree* (Huson 1998). *RAxML* (Stamatakis 2014) is a fairly new and highly efficient implementation. But even with RAxML, an analysis of a typical linguistic data set will take minutes to hours on a personal computer.

[8] This method was developed incrementally; (Edwards and Cavalli-Sforza 1964) is an early reference.

3.4.3 Bayesian Phylogenetic Inference

Maximum Likelihood is based on probability calculations. There are, broadly speaking, two philosophical interprations of the notion of probability. According to the frequentist school, the probability of the outcome of a process is the limit of the relative frequency of that outcome if the process is repeated over and over again. This makes sense for controlled experiments, but its application to contingent one-time events, such as those studied by historical linguistics, is dubious. It is not possible to repeat the history of the Indo-European language family 1,000 times and to check how often a certain pattern of cognacy relations emerges, say.

According to the subjective or Bayesian interpretation, the probability of an outcome quantifies the degree of certainty one has about this outcome. If, for instance, an election forecast says that candidate X has a 60% chance of winning the next election, this expresses the forecasters' degree of certainty on the basis of their knowledge, not some relative frequency. This interpretation seems well-suited for historical reconstruction as well. A statement such as "With 60% probability, Italic and Celtic form a common sub-group of Indo-European." is coherent unter the Bayesian, but not under the frequentist interpretation.

The calculations described above enable us to compute the probability of a character matrix given a parameterized model. A more interesting object of scientific inquiry is the converse, i.e., the probability of a certain model given the observed data. These two quantities are related via *Bayes Theorem*:

$$P(M,\vec{\theta}|D) = \frac{P(D|M,\vec{\theta})\cdot P(M,\vec{\theta})}{P(D)} = \frac{P(D|M,\vec{\theta})\cdot P(M,\vec{\theta})}{\sum_{M',\vec{\theta}'} P(D|M',\vec{\theta}')\cdot P(M',\vec{\theta})}$$

The quantity $P(M,\vec{\theta})$, the so-called *prior probability* of the parameterized model, expresses the degree of our belief that this parameterized model is correct before the data are considered. Fixing this number is tricky, and there is a vast literature on suitable methods for obtaining prior probabilities.

Suppose this problem is solved and we can compute the probability of a parameterized model, given the data (the so-called *posterior probability*). Let us say that the Maximum-Likelihood tree topology, or the Neighbor Joining topology, has a posterior probability of 2.2% — an entirely realistic outcome. This does not instill trust that this tree is correct.

A drawback of all methods discussed so for is that they produce *point estimates*, i.e., a single tree. Even if this tree is our best guess, it might still be highly unlikely. *Bayesian phylogenetic inference* overcomes this problem by generating a large number (usually at least 1,000) trees that are distributed according to the posterior probability distribution given the data. So if a tree has a posterior probability of 2.2%, we expect it to occur 22 times in a posterior sample of size 1,000.

Even if each individual tree in this sample has a low posterior probability, it is possible derive conclusions with high probability. Consider again our previous example: "With 60% probability, Italic and Celtic form a common sub-group of Indo-European." This is supported by a Bayesian posterior sample if 600 out of the 1,000 trees in the sample have a branch separating the Italic and Celtic languages from the rest.

While the Bayesian approach has clear advantages, there are also drawbacks. Setting up a Bayesian analysis requires the user to make many choices in advance pertaining to the class of models considered, the prior probability distribution (over tree topologies, rate variation across characters, rate variation across branches etc.), and technical details about how the posterior probability is generated. Even though there are heuristics aiding these decisions, running a Bayesian analysis is still, to some degree, an art as much as a science. Also, it is computationally highly demanding. An analysis of a sizeable data set usually takes at least hours and might easily take several days even on a powerful computer server.

Let us consider the outcome of Bayesian analyses for our running examples. The phonetic data are not in character format, so they cannot be used directly. We extracted a character matrix from phonetic strings in the following way. First all IPA strings were converted into strings of ASJP sound classes. For instance, English *year*, [jɪə] in IPA transcription, is converted to the ASJP string `yi3`. Each pair of a Swadesh concept and an ASJP sound class is treated as character. For English, the characters YEAR:Y, YEAR:I, and YEAR:3 have the value 1 (since the sound class occurs in the ASJP transcription of the English word for 'year'), while all other characters involving the concept 'year' have value 0 for English.

The analyses were carried out with the software *MrBayes* (Ronquist et al. 2012). For all three data sets, we chose a model with gamma-distributed rates and the relaxed clock. This means, i.a., that it is *a priori* assumed that trees are rooted and that all leaves have the same distance from the root. Consequently, branch length reflect (estimates of) historical time rather than amount of change. Rates are allowed to vary between branches and between characters.[9]

The outcome of a Bayesian analysis is a posterior sample of at least 1,000 trees. To visualize it, one tree is picked out from this sample which is somehow representative for the entire sample. (Note that this need not be the tree with the highest posterior probability.) For this purpose we used the software *TreeAnnotator* [10] and the criterion of *maximum clade credibility* (Drummond and Bouckaert 2015). The credibility of a clade is the relative proportion of trees in the posterior sample having that clade. The maximum clade credibility tree is the tree with the highest aggregated credibility of its clades.

The results are shown in Figure 15. The numbers at the branches indicate branch credibility in percent. Unlike the bootstrap support values used above, clade credibilities are probabilities. They give the (estimated) posterior probability that the true tree has a clade comprising the same leaves.

It is important to appreciate that the trees depicted here are each just one sample from a large posterior distribution. Each of the three topologies shown has a posterior probability of under 1%, so it is virtually certain that neither of them represent the true tree in its entirety. For most of the clades in the topologies, the probability that they are genuine is very high though.

Even though these analyses produce rooted trees, the clade credibilities of the clades close to the root is low in all three trees. This suggests the interpretation that the data used do not contain enough information to reliably infer deep branching patterns beyond the established sub-groupings.

While the Bayesian trees are largely consistent with the outcome of the distance-based bootstrap analysis shown above, it is obvious that Bayesian inference is able to pick up weaker signals than Neighbor Joining + bootstrap analysis. Despite the relative data sparseness, for instance, Nepali is correctly grouped together with Bengali and Hindi in the cognacy tree. Also, the Balto-Slavic group (with the exception of Bulgarian) and the Romance group are reliably identified in the WALS tree.

Wherever those trees diverge from the established picture with high credibility, this likely reflects patterns in the data rather than statistical flukes. This applies, arguably, to the classification of English as outlier within Germanic in the cognacy tree and the WALS tree, or the Balkan grouping in the WALS tree. In both cases, language contact is an obvious candidate for an explanation.

Bayesian inference of language phylogenies has been used extensively in recent years; landmark publications are (Bouchard-Côté et al. 2013, Bouckaert et al. 2012, Bowern and Atkinson 2012, Dunn et al. 2011, Gray and Jordan 2000, Gray and Atkinson 2003, Gray et al. 2009, Hruschka et al. 2015, Pagel et al. 2007), and (Pagel et al. 2013). Most of these studies are not primarily concerned with inferring trees *per se* but utilize Bayesian phylogenetic inference for other purposes, such as inferring rates of change of linguistic variables or estimating time depths of proto-languages.

The currently most popular software tools for Bayesian phylogenetic inference are *BayesPhylogenies* (Pagel and Meade 2004), *BEAST* (Drummond and Bouckaert 2015), and *MrBayes* (Ronquist

[9] This is a very crucial difference between modern phylogenetic inference and glottochronology, which has otherwise a certain family resemblance to the approaches discussed here.

[10] `http://beast.bio.ed.ac.uk/treeannotator`, accessed on April 10, 2016.

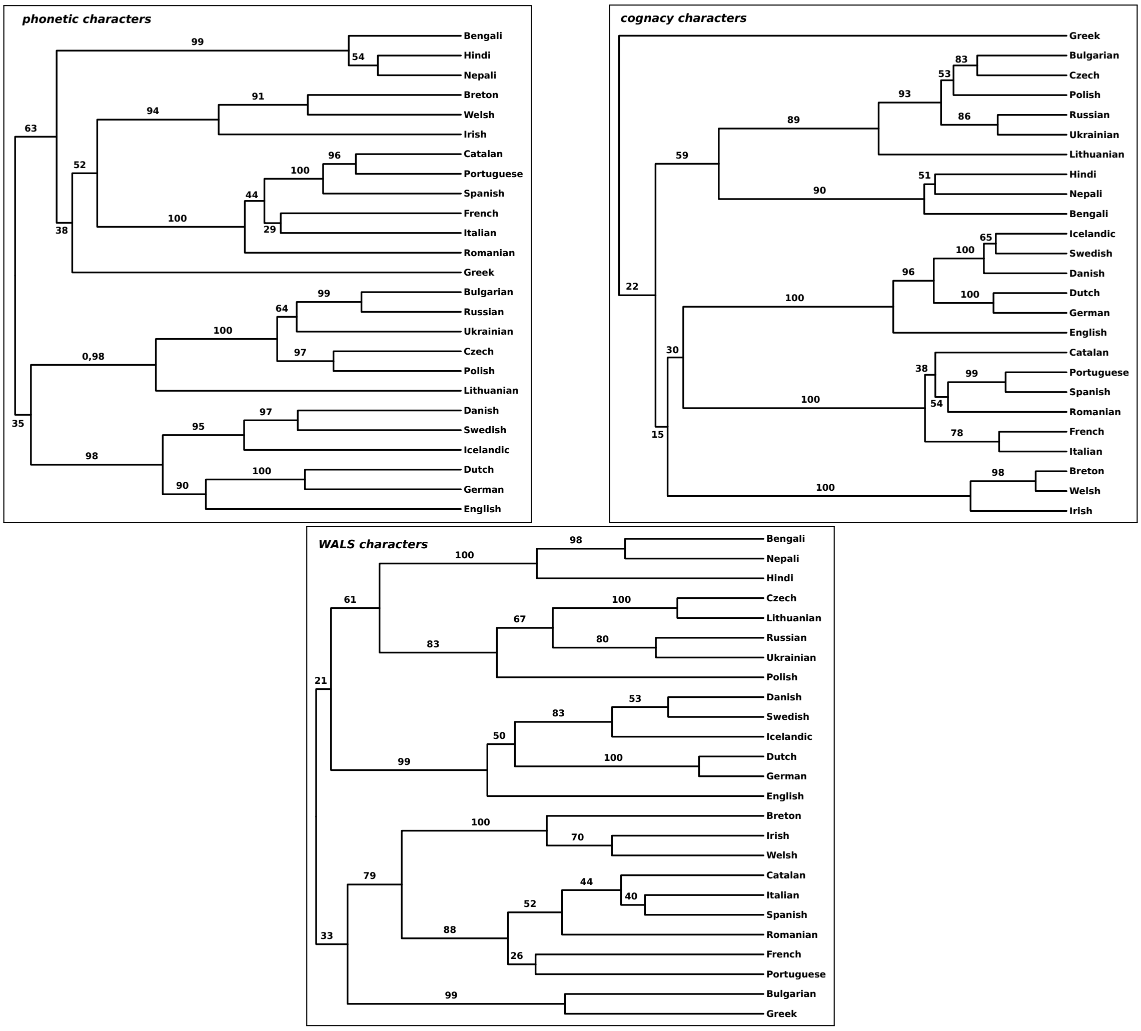


Figure 15: Bayesian trees

et al. 2012).

4 Future Challenges

The methods for sequence comparison and phylogenetic reconstruction which we discussed in the previous parts are but a small snapshot of a vast field of topics which have been addressed in the field of computational historical linguistics during the last decades. The problem of borrowing detection, for example, has been intriguing scholars for some time now, and different methods have been proposed. Phylogeny-based approaches infer borrowed words by searching for characters which are in conflict with a given reference phylogeny (Minett and Wang 2003, Nelson-Sathi et al. 2011, List, Nelson-Sathi, Geisler and Martin 2014, List, Nelson-Sathi, Martin and Geisler 2014, List 2015, Köllner and Dellert 2016). Sequence-based approaches identify potential borrowings by searching for highly similar words in either distantly related or unrelated languages (van der Ark et al. 2007, Boc et al. 2010). Phylogenetic network approaches try to infer both a phylogenetic tree and potential borrowing events from lexical character data (Nakhleh et al. 2005). Automatic borrowing detection is still in its infancy, and potentially a task far more difficult than the task of automatic cognate detection. It is very interesting what future research will bring.

Another interesting task are different approaches to the handling of semantic change. Above, we mentioned that most approaches to cognate detection restrict the task to words with the same meaning. Ideally, however, approaches to sequence comparison as well as approaches to phylogenetic reconstruction should be able to handle semantic change. Recent automatic and data-driven approaches to semantic change began to employ the *semantic map* approach of Haspelmath (2003) and apply it to synchronically attested polysemies (François 2008, Perrin 2010, Cysouw 2010*b*,*a*). Since semantic change proceeds in stages of polysemic expansion and reduction (Wilkins 1996), synchronically attested polysemies may provide immediate hints on semantic change processes. Steiner et al. (2011) presented an automatic approach to derive semantic closeness ratings from large cross-linguistic wordlists and even included it into their cognate detection workflow. List et al. (2013) built on this approach to derive a polysemy network which they further partition into communities of densely connected concepts. Their results are available in form of a web application that allows users to investigate the data interactively (List, Mayer, Terhalle and Urban 2014, Mayer et al. 2014). Münch and Dellert (2015) compared the manually edited *Database of Semantic Shifts* (Zalizniak et al. 2012) with automatically inferred polysemy networks and showed that the networks show a high similarity. Dellert (2016) illustrated how methods for causal inference can be applied to cross-linguistic polysemy data in order to infer directions of semantic change. Youn et al. (2016) used cross-linguistic polysemy data to investigate the universality of lexical semantic structures. The quantitative investigation of semantic change patterns is a very interesting and thrilling field. A future challenge will be the unification and normalization of existing datasets and approaches, and their integration into the algorithms for sequence comparison and phylogenetic reconstruction.

5 Conclusion

Computational historical linguistics is a young and highly active field, and new results and approaches surface every year. As spelled out in this chapter, a substantial part of its overall research agenda overlaps with and is inspired by the classical comparative method. Both paradigms utilize phonetic similarity patterns between languages suspected to be related to incrementally build up classes of cognate words, possibly by utilizing regular sound correspondence, and explain the observed patterns of linguistic variation via tree diagrams reflecting past diversification events. Horizontal transfer under language contact, as well as parallel innovations, are challenges for both schools since the tree model does not directly account for it.

On the other hand, there are substantial differences between the two methods which go beyond the contrast between manual versus computerized data exploration and model construction. One of the most obvious ones — which is especially prone to generate misunderstandings — concerns the adequacy criteria for family trees. According to the comparative method, a clade is justified if its component nodes share a (possibly reconstructed) innovation which is absent outside this clade. Trees constructed this way tend to be multiply-branching since it is often not possible to identify/reconstruct a shared innovation for each diversification event.

Automatically inferred phylogenetic trees, on the other hand, are almost always binary branching, for the simple reason that the standard algorithms "don't do" multiply branching trees. While it is possible to identify poorly supported branches (via bootstrap confidence values, posterior probabilities or similar techniques), even clades with solid statistical support do not always correspond to shared innovations. For instance, in a scenario with overlapping isoglosses, phylogenetic algorithms will pick the tree topology minimizing the number of shared innovations. Traditional historical linguistics would either refrain from a decision or give priority to especially informative variables (such as regular sound laws, morphological changes, or shared aberrancies).

This points to another major difference. The comparative method strongly focuses on regular sound laws and grammatical (especially morphological) properties to infer historical relationships.

Lexical information is considered less reliable since words are more easily borrowed than sound laws or paradigms. Computational historical linguistics is agnostic in this respect; in principle any kind of evidence can be used as long as cross-linguistically comparable data are available. Almost all extant approaches focus on lexical data simply because those are easiest to get hold of. This more promiscuous approach is justifiable though, since it is possible to quantitatively assess the phylogenetic informativeness of different variables (see for instance Pagel et al. 2007).

To summarize, the intellectual goals of the comparative method and of modern computational historical linguistics overlap, but they are not identical. To formulate it in a pointed way, the comparative methods strives to reconstruct the *true* history of languages in their entirety while statistical approaches search for *probable* or at least *useful* models of the observed patterns in some well-defined partial range of data. Despite these differences, they can benefit from each other. Computational approaches utilize the findings of the comparative method both as raw data and as goldstandard to validate their findings. Conversely, computational approaches are well-suited to generate initial hypotheses especially about understudied languages, to be evaluated manually by human experts.

References

Alonso, L., Castellon, I., Escribano, J., Xavier, M. and Padro, L. (2004), Multiple sequence alignment for characterizing the linear structure of revision, *in* 'Proceedings of the 4th International Conference on Language Resources and Evaluation', pp. 403–406.

Andreopoulos, B., An, A., Wang, X. and Schroeder, M. (2009), 'A roadmap of clustering algorithms: finding a match for a biomedical application', *Briefings in Bioinformatics* **10**(3), 297–314.

Baldi, P., ed. (1990), *Linguistic change and reconstruction methodology*, Mouton de Gruyter, Berlin and New York.

Barton, G. J. and Sternberg, M. J. E. (1987), 'A strategy for the rapid multiple alignment of protein sequences', *Journal of Molecular Biology* **198**(2), 327 – 337.

Baxter, W. H. and Manaster Ramer, A. (2000), Beyond lumping and splitting. Probabilistic issues in historical linguistics, *in* C. Renfrew, A. McMahon and L. Trask, eds, 'Time depth in historical linguistics', McDonald Institute for Archaeological Research, Cambridge, pp. 167–188.

Bergsma, S. and Kondrak, G. (2007), Multilingual cognate identification using integer linear programming, *in* 'Proceedings of the RANLP Workshop on Acquisition and Management of Multilingual Lexicons', pp. 656–663.

Bernardes, J. S., Vieira, F. R., Costa, L. M. M. and Zaverucha, G. (2015), 'Evaluation and improvements of clustering algorithms for detecting remote homologous protein families', *BMC Bioinformatics* **16**(1), 1–14.

Bilu, Y., Agarwal, P. K. and Kolodny, R. (2006), 'Faster algorithms for optimal multiple sequence alignment based on pairwise comparisons', *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **3**(4), 408–422.

Boc, A., Di Sciullo, A. M. and Makarenkov, V. (2010), Classification of the Indo-European languages using a phylogenetic network approach, *in* H. Locarek-Junge and C. Weihs, eds, 'Classification as a tool fo research', Springer, Berlin and Heidelberg, pp. 647–655.

Bouchard-Côté, A., Hall, D., Griffiths, T. L. and Klein, D. (2013), 'Automated reconstruction of ancient languages using probabilistic models of sound change', *Proceedings of the National Academy of Sciences* **110**(11), 4224–4229.

Bouckaert, R., Lemey, P., Dunn, M., Greenhill, S. J., Alekseyenko, A. V., Drummond, A. J., Gray, R. D., Suchard, M. A. and Atkinson, Q. D. (2012), 'Mapping the origins and expansion of the Indo-European language family', *Science* **337**(6097), 957–960.

Bowern, C. and Atkinson, Q. D. (2012), 'Computational phylogenetics of the internal structure of Pama-Nguyan', *Language* **88**, 817–845.

Burlak, S. A. and Starostin, S. A. (2005), *Sravnitel'no-istoričeskoe jazykoznanie*, Akademia, Moscow.

Campbell, L. and Poser, W. J. (2008), *Language classification: History and method*, Cambridge University Press, Cambridge.

Chang, W., Cathcart, C., Hall, D. and Garret, A. (2015), 'Ancestry-constrained phylogenetic analysis ssupport the Indo-European steppe hypothesis', *Language* **91**(1), 194–244.

Covington, M. A. (1996), 'An algorithm to align words for historical comparison', *Computational Linguistics* **22**(4), 481–496.

Covington, M. A. (1998), Alignment of multiple languages for historical comparison, *in* 'Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics', pp. 275–279.

Cysouw, M. (2010*a*), 'Drawing Networks from Recurrent Polysemies', *Linguistic Discovery* **8**(1), 281–285.

Cysouw, M. (2010*b*), 'Semantic maps as metrics on meaning', *Linguistic Discovery* **8**(1), 70–95.

de Saussure, F. (1916), *Cours de linguistique générale* [Course in general linguistics], Payot, Lausanne.

Dellert, J. (2016), Using causal inference to detect directional tendencies in semantic evolution, *in* S. Roberts, C. Cuskley, L. McCrohon, L. Barceló-Coblijn, O. Fehér and T. Verhoef, eds, 'The Evolution of Language: Proceedings of the 11th International Conference (EVOLANGX11)', Online at `http://evolang.org/neworleans/papers/139.html`.

Dixon, R. B. and Kroeber, A. L. (1919), *Linguistic families of California*, University of California Press, Berkeley.

Do, C. B., Mahabhashyam, M. S. P., Brudno, M. and Batzoglou, S. (2005), 'ProbCons', *Genome Research* **15**, 330–340.

Dolgopolsky, A. B. (1964), 'Gipoteza drevnejšego rodstva jazykovych semej Severnoj Evrazii s verojatnostej točky zrenija [A probabilistic hypothesis concering the oldest relationships among the language families of Northern Eurasia]', *Voprosy Jazykoznanija* **2**, 53–63.

Drummond, A. J. and Bouckaert, R. R. (2015), *Bayesian evolutionary analysis with BEAST*, Cambridge University Press, Cambridge, UK.

Dunn, M., Greenhill, S. J., Levinson, S. C. and Gray, R. D. (2011), 'Evolved structure of language shows lineage-specific trends in word-order universals', *Nature* **473**(7345), 79–82.

Dunn, M., Terrill, A., Reesink, G., Foley, R. A. and Levinson, S. C. (2005), 'Structural Phylogenetics and the Reconstruction of Ancient Language History', *Science* **309**(5743), 2072–2075.

Durbin, R., Eddy, S. R., Krogh, A. and Mitchinson, G. (2002), *Biological sequence analysis. Probabilistic models of proteins and nucleic acids*, 7 edn, Cambridge University Press, Cambridge.

Eddy, S. R. (2004), 'Where did the BLOSUM62 alignment score matrix come from?', *Nature Biotechnology* **22**(8), 1035–1036.

Edwards, A. W. F. and Cavalli-Sforza, L. L. (1964), Reconstruction of evolutionary trees, *in* V. H. Heywood and J. McNeill, eds, 'Phenetic and Phylogenetic Classification', Systematics Association Publisher, London, pp. 67–76.

Felsenstein, J. (1978), 'The number of evolutionary trees', *Systematic Biology* **27**(1), 27–33.

Felsenstein, J. (2004), *Inferring Phylogenies*, Sinauer Associates, Sunderland.

Felsenstein, J. (2005), *PHYLIP (Phylogeny Inference Package)*, University of Washington, Seattle. URL: `http://evolution.genetics.washington.edu/phylip/`.

Feng, D. F. and Doolittle, R. F. (1987), 'Progressive sequence alignment as a prerequisite to correct phylogenetic trees', *Journal of Molecular Evolution* **25**(4), 351–360.

Fitch, W. M. (1971), 'Toward defining the course of evolution: minimum change for a specific tree topology', *Systematic Zoology* **20**(4), 406–416.

Fox, A. (1995), *Linguistic reconstruction*, Oxford University Press, Oxford.

François, A. (2008), Semantic maps and the typology of colexification: intertwining polysemous networks across languages, *in* M. Vanhove, ed., 'From polysemy to semantic change', Benjamins, Amsterdam, pp. 163–215.

Frey, B. J. and Dueck, D. (2007), 'Clustering by Passing Messages Between Data Points', *Science* **315**, 972–976.

Gascuel, O. and Steel, M. (2006), 'Neighbor-joining revealed', *Molecular Biology and Evolution* **23**(11), 1997–2000.

Geisler, H. (1992), *Akzent und Lautwandel in der Romania*, Narr, Tübingen.

Gotoh, O. (1996), 'Significant improvement in accuracy of multiple protein sequence alignments by iterative refinement as assessed by reference to structural alignments', *Journal of Molecular Biology* **264**, 823–838.

Gray, R. D. and Atkinson, Q. D. (2003), 'Language-tree divergence times support the Anatolian theory of Indo-European origin', *Nature* **426**(6965), 435–439.

Gray, R. D., Drummond, A. J. and Greenhill, S. J. (2009), 'Language Phylogenies Reveal Expansion Pulses and Pauses in Pacific Settlement', *Science* **323**(5913), 479–483.

Gray, R. D. and Jordan, F. M. (2000), 'Language trees support the express-train sequences of Austronesian expansion', *Nature* **405**(6790), 1052–1055.

Greenberg, J. H. (1987), *Language in the Americas*, Stanford University Press, Stanford.

Grimm, J. (1822), *Deutsche Grammatik*, Vol. 1, 2 edn, Dieterichsche Buchhandlung, Göttingen.

Gusfield, D. (1997), *Algorithms on strings, trees and sequences*, Cambridge University Press, Cambridge.

Haas, M. R. (1969), *The prehistory of languages*, Mouton, The Hague and Paris.

Hall, D. and Klein, D. (2010), Finding cognate groups using phylogenies, *in* 'Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics', Stroudsburg, pp. 1030–1039.

Hall, D. and Klein, D. (2011), Large-scale cognate recovery, *in* 'Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing', pp. 344–354.

Hamming, R. W. (1950), 'Error detection and error detection codes', *AT&T TECH J* **29**(2), 147–160.

Haspelmath, M. (2003), The geometry of grammatical meaning: semantic maps and cross-linguistic comparison, *in* M. Tomasello, ed., 'The new psychology of language', Lawrence Erlbaum, Mahwah, NJ, pp. 211–242.

Haspelmath, M., Dryer, M. S., Gil, D. and Comrie, B. (2008), 'The World Atlas of Language Structures Online', Max Planck Digital Library, Munich. http://wals.info/.

Hauer, B. and Kondrak, G. (2011), Clustering semantically equivalent words into cognate sets in multilingual lists, *in* 'Proceedings of the 5th International Joint Conference on Natural Language Processing', pp. 865–873.

Holman, E. W., Wichmann, S., Brown, C. H., Velupillai, V., Müller, A. and Bakker, D. (2008*a*), Advances in automated language classification, *in* A. Arppe, K. Sinnemäki and U. Nikann, eds, 'Quantitative Investigations in Theoretical Linguistics', University of Helsinki, Helsinki, pp. 40–43.

Holman, E. W., Wichmann, S., Brown, C. H., Velupillai, V., Müller, A. and Bakker, D. (2008*b*), 'Explorations in automated lexicostatistics', *Folia Linguistica* **20**(3), 116–121.

Horton, R., Olsen, M. and Roe, G. (2010), 'Something borrowed: Sequence Alignment and the identification of similar passages in large text collections', *Digital Studies* **2**(1).

Hruschka, D. J., Branford, S., Smith, E. D., Wilkins, J., Meade, A., Pagel, M. and Bhattacharya, T. (2015), 'Detecting regular sound changes in linguistics as events of concerted evolution', *Curr. Biol.* **25**(1), 1–9.

Huson, D. H. (1998), 'SplitsTree: Analyzing and visualizing evolutionary data', *Bioinformatics* **14**(1), 68–73.

Jäger, G. and List, J.-M. (2016), Investigating the potential of ancestral state reconstruction algorithms in historical linguistics, *in* C. Bentz, G. Jäger and I. Yanovich, eds, 'Proceedings of the Leiden Workshop on Capturing Phylogenetic Algorithms for Linguistics'.

Jarceva, V. N., ed. (1990), *Lingvističeskij ènciklopedičeskij slovar* [Linguistical encyclopedical dictionary], Sovetskaja Enciklopedija, Moscow.

Jäger, G. (2013), 'Phylogenetic inference from word lists using weighted alignment with empirical determined weights', *Lang. Dyn. Change* **3**(2), 245–291.

Jäger, G. (2015), 'Support for linguistic macrofamilies from weighted alignment', *Proceedings of the National Academy of Sciences* **112**(41), 12752–12757.

Jäger, G. and List, J.-M. (2015), Factoring lexical and phonetic phylogenetic characters from word lists, *in* H. Baayen, G. Jäger, M. Köllner, J. Wahle and A. Baayen-Oudshoorn, eds, 'Proceedings of the 6th Conference on Quantitative Investigations in Theoretical Linguistics'.

Jäger, G. and Sofroniev, P. (2016), Automatic cognate classification with a Support Vector Machine. Manuscript, University of Tübingen.

Jäger, G. and Wichmann, S. (2016), Inferring The World Tree Of Languages From Word Lists, *in* S. G. Roberts, C. Cuskley, L. McCrohon, L. Barceló-Coblijn, O. Fehér and T. Verhoef, eds, 'The Evolution of Language: Proceedings of the 11th International Conference (EVOLANGX11)', Online at `http://evolang.org/neworleans/papers/147.html`.

Kassian, A., Zhivlov, M. and Starostin, G. S. (2015), 'Proto-Indo-Euroipean-Uralic comparison from the probabilistic point of view', *J. Indo-Eur. Stud.* **43**(3-4), 301–347.

Kessler, B. (1995), Computational dialectology in Irish Gaelic, *in* 'Proceedings of the seventh conference on European chapter of the Association for Computational Linguistics', pp. 60–66.

Kessler, B. (2001), *The significance of word lists*, CSLI Publications, Stanford.

Klimov, G. A. (1990), *Osnovy lingvističeskoj komparativistiki* [Foundations of comparative linguistics], Nauka, Moscow.

Köllner, M. and Dellert, J. (2016), Ancestral state reconstruction and loanword detection, *in* C. Bentz, G. Jäger and I. Yanovich, eds, 'Proceedings of the Leiden Workshop on Capturing Phylogenetic Algorithms for Linguistics'.

Kondrak, G. (2000), A new algorithm for the alignment of phonetic sequences, *in* 'Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference', pp. 288–295.

Kondrak, G. (2002), Algorithms for language reconstruction, phdthesis, University of Toronto, Toronto.

Kumar, S., Stecher, G. and Tamura, K. (2016), 'MEGA7: Molecular Evolutionary Genetics Analysis version 7.0 for bigger datasets', *Molecular Biology and Evolution* p. msw054.

Lemey, P., Salemi, M. and Vandamme, A.-M., eds (2009), *The Phylogenetic Handbook. A Practical Approach to Phylogenetic Analysis and Hypothesis Testing*, Cambridge University Press, Cambridge.

Levenshtein, V. I. (1965), 'Dvoičnye kody s ispravleniem vypadenij, vstavok i zameščenij simvolov', *Doklady Akademij Nauk SSSR* **163**(4), 845–848.

List, J.-M. (2012*a*), LexStat. Automatic detection of cognates in multilingual wordlists, *in* 'Proceedings of the EACL 2012 Joint Workshop of Visualization of Linguistic Patterns and Uncovering Language History from Multilingual Resources', pp. 117–125.

List, J.-M. (2012*b*), Multiple sequence alignment in historical linguistics, *in* E. Boone, K. Linke and M. Schulpen, eds, 'Proceedings of ConSOLE XIX', pp. 241–260.

List, J.-M. (2012*c*), SCA. Phonetic alignment based on sound classes, *in* M. Slavkovik and D. Lassiter, eds, 'New directions in logic, language, and computation', Springer, Berlin and Heidelberg, pp. 32–51.

List, J.-M. (2014*a*), 'Investigating the impact of sample size on cognate detection', *J. Lang. Relationship* **11**, 91–101.

List, J.-M. (2014*b*), *Sequence comparison in historical linguistics*, Düsseldorf University Press, Düsseldorf.

List, J.-M. (2015), 'Network perspectives on Chinese dialect history', *Bull. Chin. Linguist.* **8**, 42–47.

List, J.-M. (2016*a*), 'Beyond cognacy: Historical relations between words and their implication for phylogenetic reconstruction', *Journal of Language Evolution* **1**. Article accepted for publication in volume 1, number 2.

List, J.-M. (2016*b*), 'Wagner-Fischer Demo', *figshare* **3158836**.

List, J.-M., Mayer, T., Terhalle, A. and Urban, M. (2014), *CLICS: Database of Cross-Linguistic Colexifications*, Forschungszentrum Deutscher Sprachatlas, Marburg. URL: `http://clics.lingpy.org`.

List, J.-M. and Moran, S. (2013), An open source toolkit for quantitative historical linguistics, *in* 'Proceedings of the ACL 2013 System Demonstrations', pp. 13–18.

List, J.-M., Nelson-Sathi, S., Geisler, H. and Martin, W. (2014), 'Networks of lexical borrowing and lateral gene transfer in language and genome evolution', *Bioessays* **36**(2), 141–150.

List, J.-M., Nelson-Sathi, S., Martin, W. and Geisler, H. (2014), 'Using phylogenetic networks to model Chinese dialect history', *Lang. Dyn. Change* **4**(2), 222–252.

List, J.-M. and Prokić, J. (2014), A benchmark database of phonetic alignments in historical linguistics and dialectology., *in* N. C. Chair, K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk and S. Piperidis, eds, 'Proceedings of the Ninth International Conference on Language Resources and Evaluation', European Language Resources Association (ELRA), pp. 288–294.

List, J.-M., Terhalle, A. and Urban, M. (2013), Using network approaches to enhance the analysis of cross-linguistic polysemies, *in* 'Proceedings of the 10th International Conference on Computational Semantics – Short Papers', pp. 347–353.

Longobardi, G. and Guardiano, C. (2009), 'Evidence for syntax as a signal of historical relatedness', *Lingua* **119**(11), 1679–1706.

Longobardi, G., Guardiano, C., Silvestri, G., Boattini, A. and Ceolin, A. (2013*a*), 'Toward a syntactic phylogeny of modern Indo-European languages', *J. Hist. Linguist.* **3**(1), 122–152.

Longobardi, G., Guardiano, C., Silvestri, G., Boattini, A. and Ceolin, A. (2013*b*), 'Toward a syntactic phylogeny of modern Indo-European languages', *Journal of Historical Linguistics* **3**(1), 122–152.

Mayer, T., List, J.-M., Terhalle, A. and Urban, M. (2014), An interactive visualization of cross-linguistic colexification patterns, *in* 'Visualization as added value in the development, use and evaluation of Linguistic Resources. Workshop organized as part of the International Conference on Language Resources and Evaluation', pp. 1–8.

McMahon, A. and McMahon, R. (2005), *Language classification by numbers*, Oxford University Press, Oxford.

Meillet, A. (1954), *La méthode comparative en linguistique historique* [The comparative method in historical linguistics], reprint edn, Honoré Champion, Paris.

Meyer-Lübke, W. (1911), *Romanisches etymologisches Wörterbuch* [Etymological dictionary of Romance], Winter, Heidelberg.

Minett, J. W. and Wang, W. S.-Y. (2003), 'On detecting borrowing', *Diachronica* **20**(2), 289–330.

Münch, A. and Dellert, J. (2015), Evaluating the potential of a large-scale polysemy network as a model of plausible semantic shifts, *in* H. Baayen, G. Jäger, M. Köllner, J. Wahle and A. Baayen-Oudshoorn, eds, 'Proceedings of the 6th Conference on Quantitative Investigations in Theoretical Linguistics', Eberhard-Karls University, Tübingen.

Nakhleh, L., Ringe, D. and Warnow, T. (2005), 'Perfect Phylogenetic Networks: A new methodology for reconstructing the evolutionary history of natural languages', *Language* **81**(2), 382–420.

Needleman, S. B. and Wunsch, C. D. (1970), 'A gene method applicable to the search for similarities in the amino acid sequence of two proteins', *J. Mol. Biol.* **48**, 443–453.

Nelson-Sathi, S., List, J.-M., Geisler, H., Fangerau, H., Gray, R. D., Martin, W. and Dagan, T. (2011), 'Networks uncover hidden lexical borrowing in Indo-European language evolution', *Proc. R. Soc. London, Ser. B* **278**(1713), 1794–1803.

Nerbonne, J., Colen, R., Gooskens, C., Kleiweg, P. and Leinonen, T. (2011), 'Gabmap – A web application for dialectology', *Dialectologia* **Special Issue II**, 65–89.

Nerbonne, J., Heeringa, W., van den Hout, E., van de Kooi, P., Otten, S. and van de Vis, W. (1996), Phonetic distance between Dutch dialects, *in* G. Durieux, W. Daelemans and S. Gills, eds, 'Proceedings of the CLIN '95 meeting', pp. 185–202.

Notredame, C., Higgins, D. G. and Heringa, J. (2000), 'T-Coffee', *J. Mol. Biol.* **302**, 205–217.

Oakes, M. P. (2000), 'Computer estimation of vocabulary in a protolanguage from word lists in four daughter languages', *Journal of Quantitative Linguistics* **7**(3), 233–243.

Oflazer, K. (1996), 'Error-tolerant finite-state recognition with applications to morphological analysis and spelling', *Comput. Linguist.* **22**(1), 73–89.

Pagel, M., Atkinson, Q. D., CAlude, A. S. and Meade, A. (2013), 'Ultraconserved words point to deep language ancestry across Eurasia', *Proceedings of the National Academy of Sciences* **110**(21), 8471–8476.

Pagel, M., Atkinson, Q. D. and Meade, A. (2007), 'Frequency of word-use predicts rates of lexical evolution throughout Indo-European history', *Nature* **449**, 717–720.

Pagel, M. and Meade, A. (2004), 'A Phylogenetic Mixture Model for Detecting Pattern-Heterogeneity in Gene Sequence or Character-State Data', *Systematic Biology* **53**(4), 571–581.

Perrin, L.-M. (2010), 'Polysemous qualities and universal networks, invariance and diversity', *Linguistic Discovery* **8**(1), 259–280.

Prokić, J. and Cysouw, M. (2013), 'Combining regular sound correspondences with geographic spread', *Lang. Dyn. Change* **3**(2), 147–168.

Prokić, J., Wieling, M. and Nerbonne, J. (2009), Multiple sequence alignments in linguistics, *in* 'Proceedings of the EACL 2009 Workshop on Language Technology and Resources for Cultural Heritage, Social Sciences, Humanities, and Education', pp. 18–25.

Rask, R. K. (1818), *Undersögelse om det gamle Nordiske eller Islandske sprogs oprindelse* [Investigation of the origin of the Old Norse or Icelandic language], Gyldendalske Boghandlings Forlag, Copenhagen.

Renfrew, C. and Heggarty, P. (2009), 'Languages and Origins in Europe'. URL: `http://www.languagesandpeoples.com/`.

Ringe, D. A. (1992), 'On calculating the factor of chance in language comparison', *Transactions of the American Philo-*

sophical Society **82**(1), 1–110.

Ringe, D., Warnow, T. and Taylor, A. (2002), 'Indo-European and computational cladistics', *Transactions of the Philological Society* **100**(1), 59–129.

Ronquist, F., Teslenko, M., van der Mark, P., Ayres, D. L., Darling, A., Höhna, S., Larget, B., Liu, L., Suchard, M. A. and Huelsenbeck, J. P. (2012), 'MrBayes 3.2: efficient Bayesian phylogenetic inference and model choice across a large model space', *Systematic Biology* **61**(3), 539–542.

Rosenberg, M. S. (2009), Sequence alignment. Concepts and history, *in* M. S. Rosenberg, ed., 'Sequence alignment. Methods, models, concepts, and strategies', University of California Press, Berkeley and Los Angeles and London, pp. 1–22.

Rosenberg, M. S. and Ogden, T. H. (2009), Simulation approaches to evaluating alignment error and methods for comparing alternate alignments, *in* M. S. Rosenberg, ed., 'Sequence alignment. Methods, models, concepts, and strategies', University of California Press, Berkeley and Los Angeles and London, pp. 179–207.

Ross, M. and Durie, M. (1996), Introduction, *in* M. Durie, ed., 'The comparative method reviewed. Regularity and irregularity in language change', Oxford University Press, pp. 3–38.

Saitou, N. and Nei, M. (1987), 'The neighbor-joining method: A new method for reconstructing phylogenetic trees', *Mol. Biol. Evol.* **4**(4), 406–425.

Sankoff, D. (1975), 'Minimal mutation trees of sequences', *SIAM Journal on Applied Mathematics* **28**(1), 35–42.

Schwink, F. (1994), *Linguistic typology, universality and the realism of reconstruction*, Institute for the Study of Man, Washington.

Sicoli, M. A. and Holton, G. (2014), 'Linguistic phylogenies support back-migration from Beringia to Asia', *PloS One* **9**(3), e91722.

Smith, T. F. and Waterman, M. S. (1981), 'Identification of common molecular subsequences', *J. Mol. Biol.* **1**, 195–197.

Sokal, R. R. and Michener, C. D. (1958), 'A statistical method for evaluating systematic relationships', *University of Kansas Scientific Bulletin* **28**, 1409–1438.

Stamatakis, A. (2014), 'RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies', *Bioinformatics* pp. 1–2.

Starostin, S. A. (2000), 'The STARLING database program', Software. URL `http://starling.rinet.ru`.

Steiner, L., Stadler, P. F. and Cysouw, M. (2011), 'A pipeline for computational historical linguistics', *Lang. Dyn. Change* **1**(1), 89–127.

Swadesh, M. (1952), 'Lexico-statistic dating of prehistoric ethnic contacts', *Proceedings of the American Philosophical Society* **96**(4), 452–463.

Swofford, D. (2002), *Phylogenetic analysis using parsimony (* and other methods)*, Sinauer Associates, Sunderland.

Thompson, J. D., Higgins, D. G. and Gibson, T. J. (1994), 'CLUSTAL W', *Nucleic Acids Research* **22**(22), 4673–4680.

Torres, A., Cabada, A. and Nieto, J. J. (2003), 'An exact formula for the number of alignments between two DNA sequences', *DNA Sequence* **14**(6), 427–430.

Trask, R. L. (2000), *The dictionary of historical and comparative linguistics*, Edinburgh University Press, Edinburgh.

Turchin, P., Peiros, I. and Gell-Mann, M. (2010), 'Analyzing genetic connections between languages by matching consonant classes', *J. Lang. Relationship* **3**, 117–126.

van der Ark, R., Mennecier, P., Nerbonne, J. and Manni, F. (2007), Preliminary identification of language groups and loan words in Central Asia, *in* 'Proceedings of the RANLP Workshop on Acquisition and Management of Multilingual Lexicons', pp. 13–20.

van Dongen, S. M. (2000), Graph clustering by flow simulation, phdthesis, University of Utrecht.

Wagner, R. A. and Fischer, M. J. (1974), 'The string-to-string correction problem', *J. Assoc. Comput. Mach.* **21**(1), 168–173.

Weiss, M. (2014), The comparative method, *in* C. Bowern and N. Evans, eds, 'The Routledge Handbook of Historical Linguistics', Routledge, New York, pp. 127–145.

Wilkins, D. P. (1996), Natural tendencies of semantic change and the search for cognates, *in* M. Durie, ed., 'The comparative method reviewed. Regularity and irregularity in language change', Oxford University Press, New York, pp. 264–304.

Youn, H., Sutton, L., Smith, E., Moore, C., Wilkins, J. F., Maddieson, I., Croft, W. and Bhattacharya, T. (2016), 'On the universal structure of human lexical semantics', *Proceedings of the National Academy of Sciences* .

Zalizniak, A. A., Bulakh, M., Ganenkov, D., Gruntov, I., Maisak, T. and Russo, M. (2012), 'The catalogue of semantic shifts as a database for lexical semantic typology', *Linguistics* **50**(3), 633–669.