Automatic Inference of Sound Correspondence Patterns Across Multiple Languages

Sound correspondence patterns play a crucial role for linguistic reconstruction. Linguists use them to prove language relationship, to reconstruct proto-forms, and for classical phylogenetic reconstruction based on shared innovations.Cognate words which fail to conform with expected patterns can further point to various kinds of exceptions in sound change, such as analogy or assimilation of frequent words. Here we present an automatic method for the inference of sound correspondence patterns across multiple languages based on a network approach. The core idea is to represent all columns in aligned cognate sets as nodes in a network with edges representing the degree of compatibility between the nodes. The task of inferring all compatible correspondence sets can then be handled as the well-known minimum clique cover problem in graph theory, which essentially seeks to split the graph into the smallest number of cliques in which each node is represented by exactly one clique. The resulting partitions represent all correspondence patterns which can be inferred for a given dataset. By excluding those patterns which occur in only a few cognate sets, the core of regularly recurring sound correspondences can be inferred. Based on this idea, the paper presents a method for automatic correspondence pattern recognition, which is implemented as part of a Python library which supplements the paper. To illustrate the usefulness of the method, various tests are presented, and concrete examples of the output of the method are provided. In addition to the source code, the study is supplemented by a short interactive tutorial that illustrates how to use the new method and how to inspect its results.

1. Introduction

One of the fundamental insights of early historical linguistic research was that – as a result of systemic changes in the sound system of languages – genetically related languages exhibit structural similarities in those parts of their lexicon which were commonly inherited from their ancestral languages. These similarities surface in form of *correspondence relations* between sounds from different languages in cognate words. English *th* [θ], for example, is usually reflected as *d* in German, as we can see from cognate pairs like English *thou* vs. German *du*, or English *thorn* and German *Dorn*. English *t*, on the other hand, is usually reflected as *z* [ts] in German, as we can see from pairs like English *toe* vs. German *Zeh*, or English *tooth* vs. German *Zahn*. The identification of these *regular sound correspondences* plays a crucial role in historical language comparison, serving not only as the basis for the *proof of genetic relationship* (Dybo and Starostin 2008; Campbell and Poser 2008) or the *reconstruction of proto-forms* (Hoenigswald 1960, 72-85, Anttila 1972, 229-263), but (indirectly) also for classical

subgrouping based on shared innovations (which would not be possible without identified correspondence patterns).

Given the increasing application of automatic methods in historical linguistics after the "quantitative turn" (Geisler and List 2013, 111) in the beginning of this millennium, scholars have repeatedly attempted to either directly infer regular sound correspondences across genetically related languages (Kondrak 2009, 2003; Brown, Holman, and Wichmann 2013; Kay 1964) or integrated the inference into workflows for automatic cognate detection (Guy 1994; List 2012a, 2014; List, Greenhill, and Gray 2017). What is interesting in this context, however, is that almost all approaches dealing with regular sound correspondences, be it early formal – but classically grounded – accounts (Grimes and Agard 1959; Hoenigswald 1960) or computer-based methods (Kondrak 2003, 2002; List 2014) only consider sound correspondences between *pairs* of languages.

A rare exception can be found in the work of Anttila (1972, 229-263), who presents the search for regular sound correspondences across multiple languages as the basic technique underlying the comparative method for historical language comparison. Anttila's description starts from a set of cognate word forms (or morphemes) across the languages under investigation. These words are then arranged in such a way that corresponding sounds in all words are placed into the same column of a matrix. The extraction of regularly recurring sound correspondences in the languages under investigation is then based on the identification of similar patterns recurring across different columns within the cognate sets. The procedure is illustrated in Figure 1, where four cognate sets in Sanskrit, Ancient Greek, Latin, and Gothic are shown, two taken from Anttila (1972, 246) and two added by me.

Two points are remarkable about Anttila's approach. First, it builds heavily on the *phonetic alignment* of sound sequences, a concept that was only recently adapted in linguistics (Covington 1996; Kondrak 2000; List 2014), building heavily on approaches in bioinformatics and computer science (Wagner and Fischer 1974; Needleman and Wunsch 1970), although it was implicitly always an integral part of the methodology of historical language comparison (compare Fox 1995, 67f, Dixon and Kroeber 1919). Second, it reflects a concrete technique by which regular sound correspondences for multiple languages can be detected and employed as a starting point for linguistic reconstruction. If we look at the framed columns in the four examples in Figure 1, which are further labeled alphabetically, for example, we can easily see that the patterns A, E, and F are remarkably similar, with the missing reflexes in Gothic in the patterns E and F as the only difference. The same holds, however, for columns C, E, and F. Since A and C differ regarding the reflex sound of Gothic (*u* vs. *au*), they cannot be assigned to the same correspondence set at this stage, and if we want to solve the problem of finding the regular sound correspondences for the words in the figure, we need to make a decision which columns in the alignments we assign to the same correspondence sets, thereby 'imputing' missing sounds where we miss a reflex. Assuming that the "regular" pattern in our case is reflected by the group of A, E, and F, we can make predictions about the sounds missing in Gothic in E and F, concluding that, if ever we find the missing reflex in so far unrecognised sources of Gothic in the future, we would expect a *-u-* in the words for 'daughter-in-law' and 'red'.[1]

1 If we ever found a new Gothic text in which these words are attested but do not contain a *-u-* where we expect it, this would force us to revise our hypothesis, but as long as we lack the data, we trust in the predictive power of our investigation.

We can easily see how patterns of sound correspondences across multiple languages can serve as the basis for linguistic reconstruction. Strictly speaking, if two alignment columns are identical (ignoring missing data to some extent), they need to reflect the same proto-sound. But even if they are not identical, they could be assigned to the same proto-sound, provided that one can show that the differences are conditioned by phonetic context. This is the case for Gothic *au* [o] in pattern *C*, which has been shown to go back to *u* when preceding *h* (Meier-Brügger 2002, 210f). As a result, scholars usually reconstruct Proto-Indo-European **u* for A, C, E, and F.

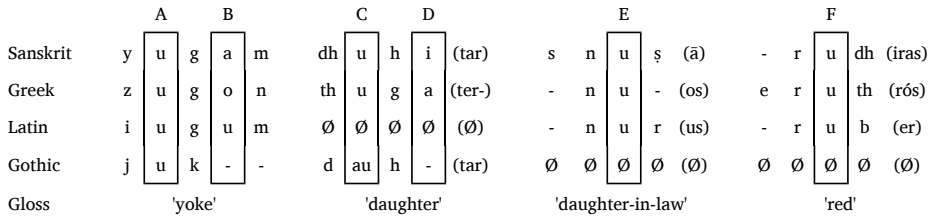

Figure 1
Regular sound correspondences across four Indo-European languages, illustrated with help of alignments along the lines of Anttila (1972: 246). In contrast to the original illustration, lost sounds are displayed with help of the dash "-" as a gap symbol, while missing words (where no reflex in Gothic or Latin could be found) are represented by the "Ø" symbol.

While it seems trivial to identify sound correspondences across multiple languages from the few examples provided in Figure 1, the problem can become quite complicated if we add more cognate sets and languages to the comparative sample. Especially the handling of missing reflexes for a given cognate set becomes a problem here, as missing data makes it difficult for linguists to decide which alignment columns to group with each other. This can already be seen from the examples given in Figure 1, where we have two possibilities to group the patterns A, C, E, and F.

The goal of this paper is to illustrate how a manual analysis in the spirit of Anttila can be automatized and fruitfully applied – not only in purely computational approaches to historical linguistics, but also in computer-assisted frameworks that help linguists to explore their data before they start carrying out painstaking qualitative comparisons (List 2016b). In order to illustrate how this problem can be solved computationally, I will first discuss some important general aspects of sound correspondences and sound correspondence patterns in Section 2, introducing specific terminology that will be needed in the remainder. In Section 3, I will show that the problem of finding sound correspondences across multiple languages can be modeled as the well-known *clique-cover problem* in an undirected network (Bhasker and Samad 1991). While this problem is *hard* to solve in an exact way computationally,[2] fast approximate solutions exist (Welsh and Powell 1967) and can be easily applied. Based on these findings, I will introduce a fully automated method for the recognition of sound correspondence patterns across multiple languages in Section 4. This method is implemented in form of a Python library and can be readily applied to multilingual wordlist data as it is also required by software packages such as LingPy (List, Greenhill, and Forkel 2017) or software tools such as EDICTOR (List 2017). In Section 5, I will then illustrate how the method can be applied and evaluate its performance both qualitatively and quantitatively. The application of the

2 Both the clique-cover problem and its inverse problem, the graph coloring problem, have been shown to be *np-complete* (Bhasker and Samad 1991).

new method is further explained in an accompanying interactive tutorial available from the supplementary material, which also shows how an extended version of the EDICTOR interface can be used to inspect the inferred correspondence patterns interactively. The supplementary material also provides code and data as well as instructions on how to replicate all tests carried out in this study.

2. Preliminaries on Sound Correspondence Patterns

In the introduction, I have tried to emphasize that the comparative method is itself less concerned with regular sound correspondences attested for language pairs, but for all languages under consideration. In the following, I want to substantiate this claim further, while at the same time introducing some major methodological considerations and ideas which are important for the development of the new method for sound correspondence pattern recognition that I want to introduce.

2.1 From Sound Correspondences to Sound Correspondence Patterns

Sound correspondences are most easily defined for pairs of languages. Thus, it is straightforward to state that German [d] regularly corresponds to English [θ], that German [ts] regularly corresponds to English [t], and that German [t] corresponds to English [d]. We can likewise expand this view to multiple languages by adding another Germanic language, such as, for example, Dutch to our comparison, which has [d] in the case of German [d] and English [θ], [t] in the case of German [ts] and English [t], and [d] in the case of German [t] and English [d]. Examples for all forms are given along with proto-forms in Proto-Germanic in Table 1.

Gloss	Proto-Germanic		German		English		Dutch	
'dead'	**daudaz*	daudaz	*tot*	toːt	*dead*	dɛd	*doot*	doːt
'deed'	**dēdiz*	deːdiz	*Tat*	taːt	*deed*	diːd	*daad*	daːt
'thick'	**þekuz*	θekuz	*dick*	dɪk	*thick*	θɪk	*dik*	dɪk
'thorn'	**þurnuz*	θurnuz	*Dorn*	dɔrn	*thorn*	θɔːn	*doorn*	doːrn
'tongue'	**tuŋgōn*	tuŋgoːn	*Zunge*	tsʊŋə	*tongue*	tʌŋ	*tong*	tɔŋ
'tooth'	**tanþs*	tanθs	*Zahn*	tsaːn	*tooth*	tuːθ	*tand*	tɑnt

Table 1
Comparing correspondence patterns for Proto-Germanic reflexes of **d-*, **þ-*, and **t-* in German, English, and Dutch (Germanic proto-forms follow Kroonen 2013).

The more languages we add to the sample, however, the more complex the picture will get, and while we can state three (basic) patterns for the case of English, German, and Dutch, given in our example, we may get easily more patterns, due to secondary sound changes in the different languages, although we would still reconstruct only three sounds in the proto-language ([θ, t, d]). Thus, there is a one-to-*n* relationship between what we interpret as a proto-sound of the proto-language, and the regular correspondence patterns which we may find in our data. While we will reserve the term *sound correspondence* for pairwise language comparison, we will use the term *sound correspondence pattern* (or simply *correspondence pattern*) for the abstract notion of regular sound correspondences across a set of languages which we can find in the data. If the words upon which we base our inference of correspondence patterns are strictly cognate (i.e., they have not been borrowed and not undergone "irregular" changes like assimilation or analogy), a given

correspondence pattern points directly to a proto-sound in the ancestral language. A given proto-sound, however, may be reflected in more than one correspondence pattern, which can be ideally resolved by inferring the phonetic context that conditions the change from the proto-language to individual descendants.

2.2 Correspondence Patterns and Proto-Forms

Scholars like Meillet (1908, 23) have stated that the core of historical linguistics is not linguistic reconstruction, but the inference of correspondence patterns, emphasizing that 'reconstructions are nothing else but the signs by which one points to the correspondences in short form'. [3] However, given the one-to-n relation between proto-sounds and correspondence patterns, it is clear, that this is not quite correct. Having inferred regular correspondence patterns in our data, our reconstructions will add a different level of analysis by further *clustering* these patterns into groups which we believe to reflect one single sound in the ancestral language.

That there are usually more than just one correspondence pattern for a reconstructed proto-sound is nothing new to most practitioners of linguistic reconstruction. Unfortunately, however, linguists do rarely list all possible correspondence patterns exhaustively when presenting their reconstructions, but instead select the most frequent ones, leaving the explanation of weird or unexpected patterns to comments written in prose. A first and important step of making a linguistic reconstruction system transparent, however, should start from an exhaustive listing of all correspondence patterns, including irregular patterns which occur very infrequently but would still be accepted by the scholars as reflecting true cognate words.

2.3 Correspondence Patterns in the Classical Literature

What scholars do instead is providing tables which summarise the correspondence patterns in a rough form, e.g., by showing the reflexes of a given proto-sound in the descendant languages in a table, where multiple reflexes for one and the same language are put in the same cell. An example, taken with modifications[4] from Clackson (2007, 37), is given in Table 2. In this table, the major reflexes of Proto-Indo-European stops in 11 languages representing the oldest attestations and major branches of Indo-European, are listed. This table is a very typical example for the way in which scholars discuss, propose, and present correspondence patterns in linguistic reconstruction (Brown et al. 2011; Holton et al. 2012; Jacques 2017; Beekes 1995). The shortcomings of this representation become immediately transparent. Neither are we told about the frequency by which a given reflex is attested to occur in the descendant languages, nor are we told about the specific phonetic conditions which have been proposed to trigger the change where we have two reflexes for the same proto-sound. While scholars of Indo-European tend to know these conditions by heart, it is perfectly understandable why they would not list them. However, when presenting the results to outsiders to their field in this form, it makes it quite difficult for them to correctly evaluate the findings. A sound correspondence table may look impressive, but it is of no use to people who have not studied the data themselves.

3 My translation, original text: 'Les «restitutions» ne sont rien autre chose que les signes par lesquels on exprime en abrégé les correspondances'.

4 We added phonetic transcriptions, preceding the original sound given by the author, separated by a slash.

PIE	Hittite	Sanskrit	Greek	Latin	Gothic	...
*p	p	p	p	p	f b	...
*b	b p	b	b	b	p	...
*b^h	b p	$b^ɦ$/bh	p^h/ph	f b	b	...
*t	t	t	t	t	θ/þ d	...
*d	d t	d	d	d	t	...
*d^h	d t	d^h/dh h	t^h/th	f d b	d	...
...	...	...	...	...	...	...
*k^w	k^w/ku	k c	k p t	k^w/qu	h^w/hw g	...
*g^w	k^w/u	g j	g b d	g^w/gu u	q	...
*g^{wh}	k^w/ku g^w/gu	$g^ɦ$/gh h	p^h/ph t^h/th k^h/kh	f g^w/gu u	g b	...

Table 2
Sound correspondence patterns for Indo-European stops, following Clackson (2007, 37).

A further problem in the field of linguistic reconstruction is that scholars barely discuss workflows or procedures by which sound correspondence patterns can be *inferred*. For well-investigated language families like Indo-European or Austronesian, which have been thoroughly studied for hundreds of years, it is clear that there is no direct need to propose a heuristic procedure, given that the major patterns have been identified long ago and the research has reached a stage where scholarly discussions circle around individual etymologies or higher levels of linguistic reconstruction, like semantics, morphology and syntax.[5] For languages whose history is less well known and where historical language reconstruction has not even reached a stage of reconstruction where a majority of scholars agrees, however, a procedure that helps to identify the major correspondence patterns underlying a given dataset, would surely be incredibly valuable.

2.4 Correspondence Patterns and Alignments

In order to infer correspondence patterns, the data must be available in *aligned* form (for details on alignments, see List 2014, 61-118), that is, we must know which of the sound segments that we compare across cognate sets are assumed to go back to the same ancestral segment. This is illustrated in Figure 2 where the cognate sets from Table 1 are presented in aligned form, following the alignment annotations of LingPy (List, Greenhill, and Forkel 2017) and EDICTOR (List 2017), in representing zero-matches with the dash (`"-"`) as a *gap symbol*, and using brackets to indicate unalignable parts in the sequences. Scholars at times object to this claim, but it should be evident, also from reading the account by Anttila (1972) mentioned above, that without alignment analyses, albeit implicit ones that are never provided in concrete, no correspondence patterns could be proposed. Even if alignments are never mentioned in the entire book of Clackson (2007), the correspondence patterns shown in Table 2 directly reflect them, since each example that one could give for the data underlying a given correspondence pattern in the descendant languages would require the identification of unique sounds in each of the reflexes that confirm this pattern.

5 For examples, compare the very detailed etymological discussions by Meier-Brügger (2002, 173-187).

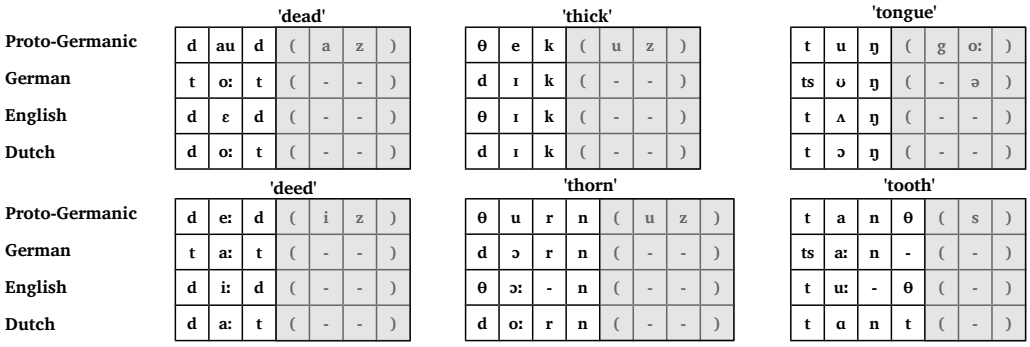

Figure 2
Alignment analyses of the six cognate sets from Table 1. Brackets around subsequences indicate that the alignments cannot be fully resolved due to secondary morphological changes.

It is important to keep in mind that strict alignments can only be made of cognate words (or parts of cognate words) that are *directly related*. The notion of directly related word (parts) is close to the notion of *orthologs* in evolutionary biology (List 2016a) and refers to words or word parts whose development have not been influenced by secondary changes due to morphological processes.[6] If we compare German *gehen* [geː.ən] 'to go' with English *go* [gəʊ], for example, it would be useless to align the verb ending *-en* in German with two gap characters in English, since we know well that English lost most of its verb endings independently. We can, however, align the initial sound and the main vowel.

Following evolutionary biology, a given column of an alignment is called an *alignment site* (or simply a *site*). An alignment site may reflect the same values as we find in a correspondence pattern, and correspondence patterns are usually derived from alignment sites, but in contrast to a correspondence pattern, an alignment site may reflect a correspondence pattern only incompletely, due to missing data in one or more of the languages under investigation. For example, when comparing German *Dorf* [dɔrf] 'village' with Dutch *dorp* [dɔrp], it is immediately clear that the initial sounds of both words represent the same correspondence pattern as we find for the cognate sets for 'thick' and 'thorn' given in Figure 2, although no reflex of their Proto-Germanic ancestor form **þurpa-* (originally meaning 'crowd', see Kroonen 2013, 553) has survived in Modern English.[7] Thanks to the correspondence patterns in Table 1, however, we know that – if we project the word back to Proto-Germanic – we must reconstruct the initial with **þ-* '[θ], since the match of German *d-* and Dutch *d-* only occurs – if we ignore recent borrowings – only in correspondence patterns in which English has *th-*.

These "gaps" due to missing reflexes of a given cognate set are not the same as the gaps inside an alignment, since the latter are due to the (regular) loss or gain of a sound segment in a given alignment site, while gaps due to missing reflexes may either reflect processes of *lexical replacement* (List 2014, 37f), or a preliminary stage of research resulting from insufficient data collections or insufficient search for potential reflexes.

6 In some sense, we can find this thought already in the work of August Schleicher, who emphasized the importance of deriving the 'mutmaßliche grundform, d. i. die gestalt' ('presumable base form, i.e. the Gestalt') before turning to a comparison of cognate words across languages (Schleicher 1852, iv).

7 Old English still knows *thorp*, but in Modern English, we only find it in names.

While I follow the LingPy annotation for gaps in alignments by using the dash as a symbol for gaps in alignment sites, I will use the character ∅ (denoting the empty set) to represent missing data in correspondence patterns and alignment sites. The relation between correspondence patterns in the sense developed here and alignment sites is illustrated in Figure 3, where the initial alignment sites of three alignments corresponding to Proto-Germanic *þ* [θ] are assembled to form one correspondence pattern.

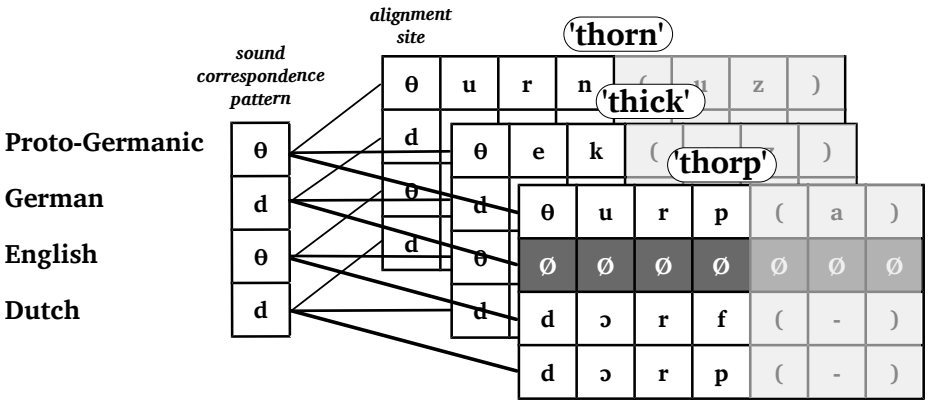

Figure 3
Alignment sites and correspondence patterns: While alignment sites are concrete representations of the presumed relations among cognate words, correspondence patterns are a further stage of abstraction.

2.5 Summary on Sound Correspondence Patterns

In this section, I have tried to introduce some basic terms, techniques, and concepts that help to set the scope for the new method for sound correspondence pattern recognition that will be presented in this paper. I first distinguished correspondence patterns from proto-forms, since one proto-form can represent multiple correspondence patterns in a given language family. I then distinguished correspondence patterns from concrete alignment sites in which the relations of concrete cognate words are displayed, by emphasizing that correspondence patterns can be seen as a more abstract analysis, in which similar alignment sites across different cognate sets, regardless of missing reflexes in the descendant languages, are assigned to the same correspondence pattern. In the next sections, I will try to show that this handling allows us to model the problem of sound correspondence pattern recognition as a network partitioning task.

3. Preliminary Thoughts on Correspondence Patterns Recognition

Before presenting the new method for automatic correspondence pattern recognition, it is important to introduce some basic thoughts about alignment sites and correspondence patterns that hopefully help to elucidate the core idea behind the method. Having established the notion of *alignment site compatibility*, I will show how alignment sites can be modelled with help of an *alignment site network*, from which we can extract regularly recurring sound correspondences.

3.1 Compatibility of Alignment Sites

If we recall the problem we had in grouping the alignment sites E and F from Figure 1 with either A or C, we can see that the general problem of grouping alignment sites to correspondence patterns is their *compatibility*. If we had reflexes for all languages under investigation in all cognate sets, the compatibility would not be a problem, since we could simply group all identical sites with each other, and the task could be considered as solved. However, since it is rather an exception than the norm to have reflexes for all languages under consideration in a number of cognate sets, we will always find alternative possibilities to group our alignment sites in correspondence patterns. In the following, I will assume that two alignment sites are compatible, if they (a) share at least one sound which is not a gap symbol, and (b) do not have any conflicting sounds. We can further *weight* the compatibility by counting how many sounds are shared among two alignment sites. This is illustrated in Figure 4 for our four alignment sites A, C, E, and F from Figure 1 above. As we can see from the figure, only two sites are incompatible, namely A and C, as they show different sounds for the reflexes in Gothic. Given that the reflex for Latin is missing in site C, we can further see that C shares only two sounds with E and F.

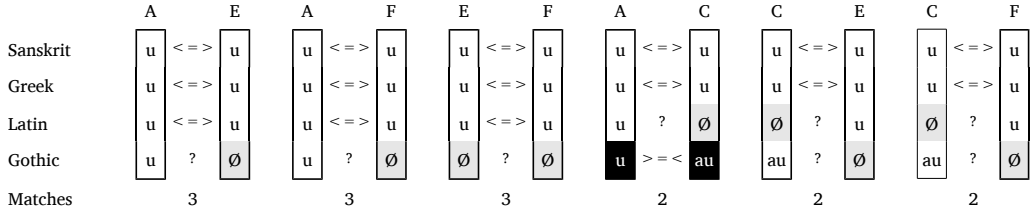

Figure 4
Assessing the compatibility of the four alignment sites from Figure 1.

3.2 Modeling Sound Correspondence Patterns in Networks

Having established the concept of *alignment site compatibility* in the previous section, it is straightforward to go a step further and model alignment sites in form of a network. Here, all sites in the data represent nodes (or vertices), and edges are only drawn between those nodes which are *compatible*, following the criterion of compatibility outlined in the previous section. We can further weight the edges in the alignment site network, for example, by using the number of matching sounds (where no missing data is encountered) to represent the strength of the connection (but we will disregard weighting in our method). Figure 5 illustrates how an alignment site network can be created from the compatibility comparison shown in Figure 4.

3.3 Correspondence Pattern Recognition as a Clique Coverage Problem

As was mentioned already in the introduction, the main problem of assigning different alignment sites to correspondence patterns is to decide about those cases where one site could be assigned to more than one patterns. Having shown how the data can be modeled in form of a network, we can rephrase the task of identifying correspondence patterns as a *network partitioning task* with the goal to split the network into non-overlapping sets of nodes. Given that our main criterion for a valid correspondence pattern is full

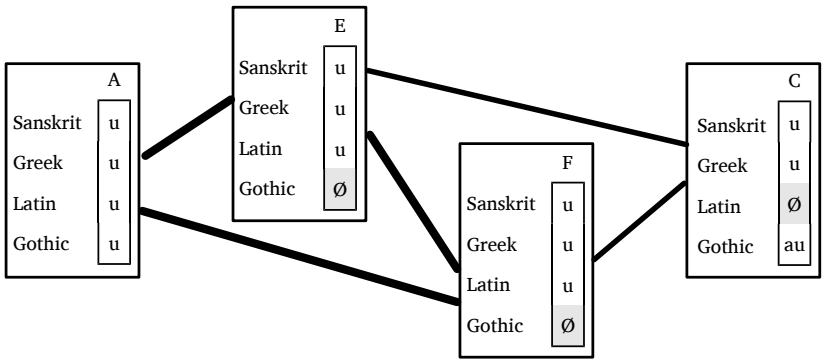

Figure 5
Representing alignment sites with help of a network. Edges are only drawn between compatible alignment sites. The width of the edges represents the number of matches per pair of alignment sites.

compatibility among all alignment sites of a given partition, we can further specify the task as a *clique partitioning task*. A *clique* in a network is 'a maximal subset of the vertices [nodes] in an undirected network such that every member of the set is connected by an edge to every other' (Newman 2010, 193). Demanding that sound correspondence patterns should form a clique of compatible nodes in the network of alignment sites is directly reflecting the basic practice of historical language comparison as outlined by Anttila (1972), according to which a further grouping of incompatible alignment sites by proposing a proto-form would require us to identify a phonetic environment that could show incompatible sites to be complementary. Partitioning our alignment site network into cliques does therefore *not* solve the problem of linguistic reconstruction, but it can be seen as its fundamental prerequisite.

It is difficult to find a linguistically valid criterion for the way in which the alignment site network should be partitioned into cliques of compatible nodes. Following a general reasoning along the lines of Occam's razor or general *parsimony* of explanation (Gauch 2003, 269-326), which is often frequented as a criterion for favoring one explanation over the other in historical language comparison, it is straightforward to state the problem of clique partitioning of alignment site networks as a *minimum clique cover problem*, i.e., the problem of identifying 'the minimum number of cliques into which a graph can be partitioned' (Bhasker and Samad 1991, 2). This means, when partitioning our alignment site graph, we should try to minimize the number of cliques to which the different nodes are assigned.

The minimum clique cover problem is a well-known problem in graph theory and computer science, although it is usually more prominently discussed in form of its inverse problem[8], the *graph coloring problem*, which tries to assign different colors to all nodes in a graph which are directly connected (Hetland 2010, 276). While the problem is generally known to be *NP-hard* (ibid.), fast approximate solutions like the Welsh-Powell algorithm (Welsh and Powell 1967) are available. Using approximate solutions seems to be appropriate for the task of correspondence pattern recognition, given that we do not (yet) have formal linguistic criteria to favor one clique cover over another. We should

8 The inverse problem of a given problem in graph theory provides a solution to the original problem for a graph in which the original edges are deleted and nodes formerly unconnected are connected.

furthermore bear in mind that an optimal resolution of sound correspondence patterns for linguistic purposes would additionally allow for uncertainty when it comes to assigning a given alignment site to a given sound correspondence pattern. If we decided, for example, that the pattern C in Figure 5 could by no means cluster with E and F, this may well be premature before we have figured out whether the two patterns (**u-u-u-u** vs. **u-u-u-au**) are *complementary* and what phonetic environments explain their complementarity. The algorithm for correspondence pattern recognition, which will be presented in the next section, accounts for this by allowing one to propose fuzzy partitions in which alignment sites can be assigned to more than one correspondence pattern.

4. An Automatic Method for Correspondence Pattern Recognition

In the following, I will introduce a method for automatic correspondence pattern recognition that takes cognate-coded and phonetically aligned multilingual wordlists as input and delivers a list of correspondence patterns as output, with each alignment site in the original data being assigned to at least one of the inferred correspondence patterns.

4.1 General Workflow

The general workflow underlying the method for automatic correspondence pattern recognition can be divided into five different stages. Starting from a multilingual wordlist in which translations for a concept list are provided in form of phonetic transcriptions for the languages under investigation, the words in the same semantic slot are manually or automatically searched for cognates (A) and (again manually or automatically) phonetically aligned (B). The alignment sites are then used to construct an *alignment site network* in which edges are drawn between compatible sites (C). The alignment sites are then partitioned into distinct non-overlapping subsets using an approximate algorithm for the minimum clique cover problem (D). In a final step, potential correspondence patterns are extracted from the non-overlapping subsets, and all individual alignment sites are assigned to those patterns with which they are compatible (E). While there are both standard algorithms and annotation frameworks for stages (A) and (B),[9], the major contribution of this paper is to provide the algorithms for stages (C), (D), and (E). The workflow is further illustrated in Figure 6. In the following sections, I will provide more detailed explanations on the different stages.

4.2 Implementation, Input Format, and Output Format

The method has been implemented as a Python package that can be used as a plugin for the LingPy library for quantitative tasks in historical linguistics (List, Greenhill, and Forkel 2017). Users can either invoke the method from within Python scripts as part of their customised workflows, or from the command line. The supplementary material offers a short tutorial along with example data illustrating how the package can be used.

The input format for the method described here generally follows the input format employed by LingPy. In general, this format is a tab-separated text file with the first row being reserved for the header, and the first column being reserved for a unique

9 For automatic cognate detection, compare for example List (2014), List, Greenhill, and Gray (2017), Arnaud, Beck, and Kondrak (2017), and Jäger, List, and Sofroniev (2017), and for automatic phonetic alignment, compare Prokić, Wieling, and Nerbonne (2009) and List (2014). For manual annotation of cognates and alignments, compare List (2017).

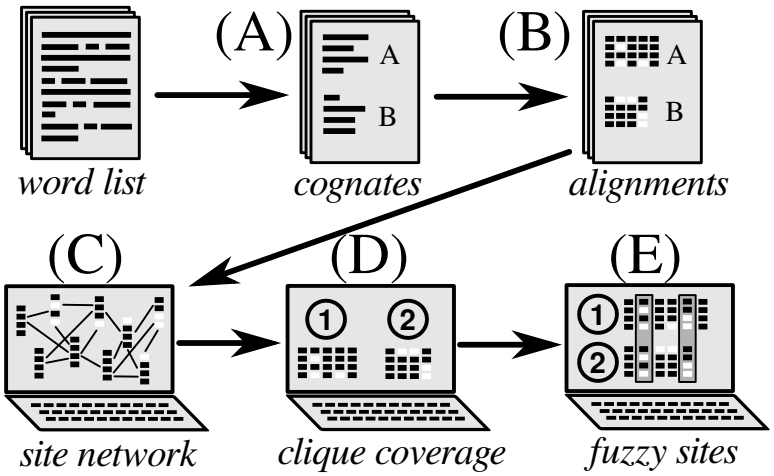

Figure 6
General workflow of the method for automatic correspondence pattern recognition. Steps (A) and (B) may additionally be provided in manually corrected form from the input data.

numerical identifier. The header specifies the entry types in the data. In LingPy, all analyses require certain entry types to be provided from the file, but the entry types can vary from method to method. Table 3 provides an example for the minimal data that needs to be provided to our method for automatic correspondence pattern recognition. In addition to the generally needed information on the identifier of each word (ID), on the language (DOCULECT), the concept or elicitation gloss (CONCEPT), the (not necessarily required) orthographic form (FORM), and the phonetic transcription provided in space-segmented form (TOKENS), the method requires information on the type of sound (consonant or vowel, STRUCTURE),[10] the cognate set (COGID), and the alignment (ALIGNMENT).

The format employed by LingPy and the method presented in this study is very similar to the format specifications developed in the *Cross-Linguistic Data Formats* (CLDF) initiative (Forkel et al. 2017), which seeks to render cross-linguistic data more comparable. The CLDF homepage (`http://cldf.clld.org`) offers more detailed information on the ideas behind the different columns mentioned above as part of the CLDF ontology. LingPy offers routines to convert to and from the format specifications of the CLDF initiative.

The method offers different output formats, ranging from the LingPy wordlist format in which additional columns added to the original wordlist provide information on the inferred patterns, or in the form of tab-separated text files, in which the patterns are explicitly listed. The wordlist output can also be directly inspected in the EDICTOR tool, allowing for a convenient manual inspection of the inferred patterns.

10 The values passed to the STRUCTURE column can be arbitrarily filled. When running the analysis, they are used to identify those positions in the alignments which should be analysed in a given run (e.g., only vowels, vs. only consonants, etc.).

ID	DOCULECT	CONCEPT	FORM	TOKENS	STRUCTURE	COGID	ALIGNMENT
1	German	tongue	Zunge	ts ʊ ŋ ə	**c v c** v	1	ts ʊ ŋ (ə)
2	English	tongue	tongue	t ʌ ŋ	c v c	1	t ʌ ŋ (-)
3	Dutch	tongue	tong	t ɔ ŋ	c v c	1	t ɔ ŋ (-)
4	German	tooth	Zahn	ts aː n	c v c	2	ts aː n -
5	English	tooth	tooth	t uː θ	c v c	2	t uː - θ
6	Dutch	tooth	tand	t ɑ n t	**c v c** c	2	t ɑ n t
7	German	thick	dick	d ɪ k	c v c	3	d ɪ k
...	...	...	...	...	...	...	...

Table 3
Input format with the basic values needed to apply the method for automatic correspondence pattern recognition. Both the information in the column COGID (providing information on the cognacy) and the ALIGNMENT column (providing the segmented transcriptions in aligned form) can be automatically computed.

4.3 Cognate Detection and Phonetic Alignment

Given that the method is implemented in form of a plugin for the LingPy library, all cognate detection and phonetic alignment methods offered in LingPy are also available for the approach and have been tested. Among automatic cognate detection methods, the users can select among the *consonant-class matching approach* (Turchin, Peiros, and Gell-Mann 2010), simple cognate partitioning with help of the *normalized edit distance* (Levenshtein 1965) or the Sound-Class-Based Alignment (SCA) method (List 2012b), and enhanced cognate detection with help of the original *LexStat* method (List 2012a) and its enhanced version, based on the *Infomap* network partitioning algorithm (Rosvall and Bergstrom 2008), as proposed in (List, Greenhill, and Gray 2017). In addition, when dealing with data which has been previously segmented morphologically, users can also employ LingPy's partial cognate detection method (List, Lopez, and Bapteste 2016). For phonetic alignments, LingPy offers two basic variants as part of the SCA method for multiple sequence alignments (List 2012b), namely "classical" *progressive* alignment, and *library-based* alignment, inspired by the T-COFFEE algorithm for multiple sequence alignment in bioinformatics (Notredame, Higgins, and Heringa 2000).

The automatic methods for cognate detection and phonetic alignments, however, are not necessarily needed in order to apply the automatic method for correspondence pattern recognition. Alternatively, users can prepare their data with help of the EDICTOR tool for creating, maintaining and publishing etymological data (List 2017), which allows users both to annotate cognates and alignments from scratch or to refine cognate sets and alignments that have been derived from automatic approaches.

Users proficient in computing do not need to rely on the algorithms offered by LingPy. Given that the number of freely available algorithms for automatic cognate detection is steadily increasing (Jäger, List, and Sofroniev 2017; Arnaud, Beck, and Kondrak 2017;

Rama et al. 2017), users can design their personal workflows, as long as they manage to export the analyses into the input formats required by the new method for correspondence pattern recognition.

4.4 Correspondence Pattern Recognition

The method for correspondence pattern recognition consists of three stages (C-E in our general workflow). It starts with the reconstruction of an alignment site network in which each node represents a unique alignment site, and links between alignments sites are drawn if the sites are compatible, following the criterion for site compatibility outlined in Section 3.1 (C). It then uses a greedy algorithm to compute an approximate minimal clique cover of the network (D). All partitions proposed in stage (D) qualify as potentially valid correspondence patterns of our data. But the individual alignment sites in a given dataset may as well be compatible with more than one correspondence pattern. For this reason, the method iterates again over all alignment sites in the data and checks with which of the correspondence patterns inferred in stage (D) they are compatible. This procedure yields a (potentially) fuzzy assignment of each alignment site to at least one but potentially more different sound correspondence patterns (E). By further weighting and sorting the fuzzy patterns to which a given site has been assigned, the number of fuzzy alignment sites can be further reduced.

As mentioned above in Section 3.3, by modeling the alignment sites in the data as a network in which edges are drawn between compatible alignment sites, we can treat the problem of correspondence pattern recognition as a network partitioning task, or, more precisely, as a specific case of the clique cover problem. Given the experimental status of this research, where it is still not fully understood what qualifies as an optimal clique cover of an alignment site graph with respect to the problem of identifying regular sound correspondence patterns in historical linguistics, I decided to use a simple approximate solution for the clique cover problem. The advantage of this approach is that it is reasonably fast and can be easily applied to larger datasets. Once more data for training and testing becomes available, the basic framework introduced here can be easily extended by adding more sophisticated methods.

The clique cover algorithm consists of two steps. In a first step, the data is sorted, using a customized variant of the Quicksort algorithm (Hoare 1962), which seeks to sort patterns according to compatibility and similarity. By iterating over the sorted patterns, all compatible patterns are assigned to the same cluster in this first pass, which provides a first very rough partition of the network. While this procedure is by no means perfect, it has the advantage of detecting major signals in the data very quickly. For this reason, it has also been introduced into the web-based EDICTOR tool, where a more refined method addressing the clique cover problem could not be used, due to the typical limitations of JavaScript running on client-side.

In a second step, an inverse version of the Welsh-Powell algorithm for graph coloring (Welsh and Powell 1967) is employed. This algorithm starts from sorting all existing partitions by size, beginning with the largest partitions. It then consecutively compares the currently largest partition with all other partitions, merging those which are compatible with each other, and keeping the incompatible partitions in the queue. The algorithm stops, once all partitions have been visited and compared against the remaining partitions.

In order to adjust the algorithm to the specific needs of correspondence pattern recognition in historical linguistics, I use a slightly modified version. The method starts by

sorting all partitions (which were retrieved from the application of the sorting algorithm) in reverse order using the number of non-missing segments in the pattern and the *density* of the alignment sites assigned to the pattern as our criterion. The density of a given correspondence pattern and the alignment site matrix (showing all alignment sites compatible with the pattern) is calculated by dividing the number of cells with no missing data in the matrix by the total number of cells in the matrix (see Figure 7 for an example). The method then selects the first element of the sorted partitions and compares it against all the remaining partitions for compatibility as defined above. If the first partition is compatible with another partition, the two partitions are merged into one and the new partition is further compared with the remaining partitions. If the partition is not compatible, the incompatible partition is appended to a queue. Once all partitions have been checked for compatibility, the pattern that was checked against the remaining patterns is placed in the result list, and the queue is sorted again according to the specific sort criteria. The procedure is repeated until all initial partitions have been checked against all others.

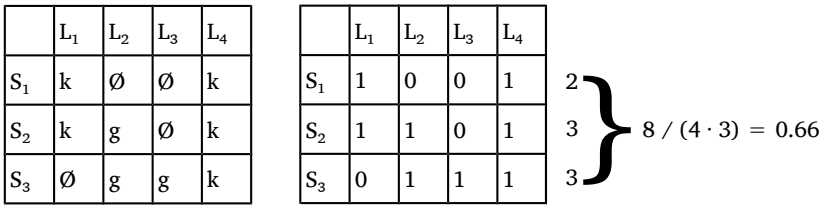

Figure 7
Calculating the alignment site density of a given correspondence pattern. The density is calculated by dividing the number of cells in the alignment site matrix with no missing data by the total number of cells in the matrix.

Figure 8 gives an artificial example that illustrates how the basic method infers the clique cover. Starting from the data in (A), the method assembles patterns A and B in (B) and computes their pattern, thereby retaining the non-missing data for each language in the pattern as the representative value. Having added C and D in this fashion in steps (C) and (D), the remaining three alignment sites, E-G are merged to form a new partition, accordingly, in steps (E) and (F).

In this context, it is important to note that the originally selected pattern may change during the merge procedure, since missing spots can be filled by merging the pattern with a new alignment site. For this reason, it is possible that this procedure, when only carried out one time, may not result in a true clique cover (in which all compatible alignment sites are merged). For this reason, the procedure is repeated several times (3 times is usually enough), until the resulting partitioning of the alignment site graph represents a true clique cover. Obviously, this algorithm only approximates the clique cover problem. However, as we will see in Section 5, it works reasonably well, at least for the smaller datasets which were considered in the tests.

In the final stage of assigning alignment sites to correspondence patterns, our method first assembles all correspondence patterns inferred from the greedy clique cover analysis and then iterates over all alignment sites, checking again whether they are compatible with a given pattern or not. Since alignment sites may suffer from missing data, their assignment is not always unambiguous. The example alignment from Figure 1, for example, would yield two general correspondence patterns, namely **u-u-u-au** vs. **u-u-u-u**. While the assignment of the alignment sites A and C in the figure would be unambiguous,

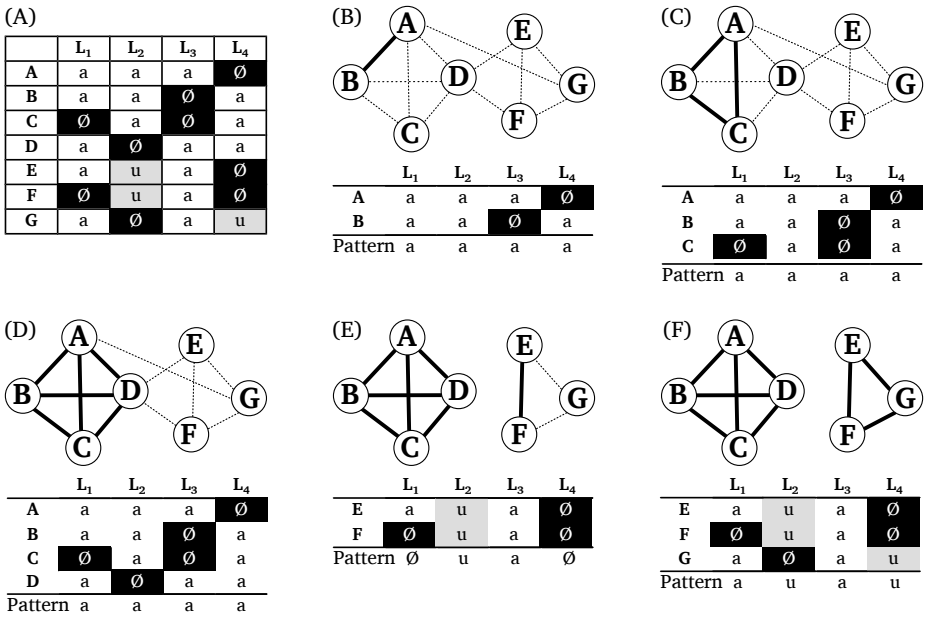

Figure 8
Example for the basic method to compute the clique cover of the data. (A) shows all alignment sites in the data. (B-D) show how the algorithm selects potential edges step by step in order to arrive at a first larger clique cover. (E-F) show how the second cover is inferred. In each step during which one new alignment site is added to a given pattern, the pattern is updated, filling empty spots. While there are two missing data points in (E), where only alignment sites E and F are merged, these are filled after adding G.

the sites E and F would be assigned to both patterns, since, judging from the data, we could not tell what correspondence pattern they represent in the end.

5. Testing the Method for Correspondence Pattern Recognition

Given that the perspective on sound correspondences and sound correspondence patterns presented in this study does not have – at least to my best knowledge – predecessors in form of quantitative studies, it is difficult to come up with a direct test of the suitability of the approach. Since classical linguists have never discussed all correspondence patterns in their data exhaustively, we have no direct means to carry out an evaluation study into the performance of the new approach as compared to an expert-annotated gold standard.

What can be done, however, is to test specific characteristics of the method by contrasting the findings when varying certain parameters, or by introducing certain distortions and testing how the method reacts to them. Last not least, we can also carry out a deep qualitative analysis of the results by manually inspecting proposed correspondence patterns. Before looking into these aspects in more detail, however, it is useful to look at some general statistics and results when applying the method to different datasets.

Dataset	Source	Languages	Concepts	Cognates	Density
Bahnaric	Sidwell (2015)	24	200	1055	0.76
Chinese	Běijīng Dàxué (1964)	18	180	1231	0.68
Huon	McElhanon (1967)	14	139	855	0.48
Romance	Saenko (2015)	43	110	465	0.90
Tujia	Starostin (2013)	5	109	179	0.63
Uralic	Syrjänen et al. (2013)	7	173	870	0.39

Table 4
Basic statistics the test data to test the new method. The training data is listed in the appendix and was only used for initial trials when developing the method.

5.1 Training and Test Data

For the tests, I use the benchmark database for automatic cognate detection compiled for the study of List, Greenhill, and Gray (2017). This database offers a training and a test set, consisting of six subsets each, with data from different subgroups of different language families. In general, the datasets are rather small, ranging from 5 to 43 language varieties and from 109 to 210 concepts with a moderate genetic diversity. For our purpose, small datasets of rather closely related languages are very useful, not only because it is easier to evaluate them manually, but also because we can rely on automated alignments when searching for sound correspondence patterns. Table 4 provides an overview of the datasets along with basic information regarding the original data sources, the number of languages, concepts, and cognate sets.

I also introduce a new measure, which I call *cognate density*, which provides a rough estimate on the genetic diversity of a given dataset.The cognate density D can be calculated with help of the formula

$$D = 1 - \frac{1}{m} \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} \frac{1}{\text{cognates}(w_{ij})} \tag{1}$$

where m is the number of concepts, n_i is the number of words in concept slot m_i, w_{ij} is the j-th word in the i-th concept slot, and cognates(w_{ij}) is the size of the cognate set to which w_{ij} belongs. If the cognate density is high, this means that the words in the data tend to cluster in large cognate sets. If it is low, this means that many words are isolated. If no words in the data are cognate, the density is zero. The cognate density measure is potentially useful to inspect specific strengths and weaknesses of the method proposed here, and one should generally expect that the method will work better on datasets with a high cognate density, since datasets with low density will have many sparse cognate sets which will be difficult to assign consistently to unambiguous correspondence patterns.

5.2 General Characteristics

As a first test, the method was applied to the test data and some basic statistics were calculated. Since the datasets are cognate-coded, but not yet phonetically aligned, I computed phonetic alignments for all datasets using the SCA algorithm in LingPy's

default settings,[11] before applying the correspondence pattern recognition method in three different versions, one inferring correspondence patterns from all alignment sites, regardless of whether they reflect a vowel or a consonant, one where only consonants are considered, and one where only sites containing vowels are compared. The results of this analysis are summarized in Table 5, which lists the number of alignment sites (St.), the number of inferred correspondence patterns (Pt.), the number of unique (singleton) patterns which cover only one alignment site and cannot be assigned to any other pattern (Sg.) and the fuzziness of the patterns (Fz.), which is the average number of different patterns to which each individual site can be attached, for all three variants (all patterns, only consonants, and only vowels) for each of the six datasets.

	All Patterns				**Consonants**				**Vowels**			
Dataset	**St.**	**Pt.**	**Sg.**	**Fz.**	**St.**	**Pt.**	**Sg.**	**Fz.**	**St.**	**Pt.**	**Sg.**	**Fz.**
Bahnaric	2659	865	385	4.59	1651	480	222	4.53	1008	382	167	4.52
Chinese	3205	584	191	5.78	1118	207	79	3.81	1308	298	108	7.28
Huon	1572	271	104	4.07	873	154	58	2.98	699	115	40	5.42
Romance	1656	874	587	3.67	940	496	345	3.51	716	379	250	3.85
Tujia	952	272	130	2.66	323	118	62	1.71	347	84	41	2.71
Uralic	1346	326	131	3.35	763	180	74	2.75	583	141	45	4.16

Table 5
General statistics on the patterns inferred from the test sets.

What we can see from these results is that the method seems to be successful in drastically reducing the number of alignment sites by assigning them to the same pattern. What is also evident, but not necessarily surprising, is the large proportion of unique patterns across all datasets. A further aspect worth mentioning is that, apart from the case of Bahnaric, the fuzziness of the assignment of alignment sites to the inferred correspondence patterns seems to be generally higher for vowels than for consonants. This is generally not surprising, as it is well known that sound correspondences among vowels are much more difficult to establish than for consonants.

Correspondence patterns wich represent only one alignment site in the data can be regarded as *irregular* with respect to the datasets, as they do not offer enough evidence to conclude whether they are representative for the languages under investigation or not. Obviously, irregular correspondence patterns may arise for different reasons. Among these are (1) errors in the data (e.g., resulting from mistaken transcriptions), (2) errors in the cognate judgments (simple lookalikes and undetected borrowings), (3) errors in the alignments (assuming that correspondence patterns can only be inferred strictly by aligning the words in question), (4) irregular sound change processes (especially assimilation of frequently recurring words, often triggered by morphological processes, but also cases like metathesis), (5) analogy (in a broader sense, referring not only to inflectional paradigms, but also to more abstract interferences among word families in a given language), and (6) missing data that renders regular sound change processes irregular (e.g., if scholars have not searched thoroughly enough for more examples, or

11 The default settings use the progressive version of the SCA alignments (as opposed to library-based alignments), and an extended sound-class model (called *SCA model* in LingPy) of currently 29 symbols.

if there is truly only one example left or available in the data).[12] Given the multiple reasons by which singleton correspondence patterns can emerge, it is difficult to tell without inspecting the data in detail, what exactly they result from.

A potentially general problem, which can be easily tested, is that the alignments were carried out automatically, while the cognate sets were assigned manually. This may lead to considerable distortions since manual cognate coders that disregard alignments usually do not pay much attention to questions of partial cognacy or morphological differences among cognate words due to derivation processes. As a result, any automatic alignment method applied to historically diverse cognate words will necessarily align parts which a human would simply exclude from the analysis. We can automatically approximate this analysis by taking only those sites of the alignments in the data into consideration in which the number of gaps does not exceed a certain threshold. A straightforward threshold excludes all alignment sites where the number of gaps is in the majority, compared to the frequency of any other character in the site. The advantage of this criterion is that it is built-in in LingPy's function for the computation of *consensus sequences* from phonetic alignments. Consensus sequences represent for each site of an alignment the most frequently recurring segment (Schneider 2002). To exclude all sites in which gaps are most frequent, it is therefore enough to compute a consensus sequence for all alignments and disregard those sites for which the consensus yields a gap when carrying out the correspondence pattern recognition analysis. The results of this analysis are shown in Table 6. As can be seen easily, the analysis in which alignment sites with a considerable number of gaps are excluded produces considerably lower proportions of singleton correspondence patterns for all six test sets. The fact that the number of alignment sites is also drastically reduced in all datasets further illustrates how important it may be to invest the time to manually align cognate sets and mark affixes as non-alignable parts.

Dataset	**Sites**	**Patterns**	**Singletons**	**Fuzziness**	**Non-Gappy**	**Gappy**
Bahnaric	2006	516	201	4.85	0.39	0.47
Chinese	2906	475	139	5.88	0.29	0.34
Huon	1478	213	74	3.88	0.35	0.41
Romance	1174	476	270	4.70	0.57	0.68
Tujia	820	219	110	2.75	0.50	0.51
Uralic	1168	251	94	3.46	0.37	0.41

Table 6
Calculating correspondence patterns from alignment sites with a limited number of gaps. The last two columns contrast the proportions of singleton correspondence patterns in the original analysis reported in Table 5 above (Gappy) with the results obtained for the refined analysis in which gappy alignment sites are excluded (Non-Gappy).

5.3 Specific Characteristics

In the previous section, I have mentioned different factors that may influence the correspondence pattern analysis. Although we lack gold standards against which the

12 It is even possible that the proto-language had one specific sound only in a particular word, which would render the detection of "regular" sound correspondences impossible, unless indirect evidence from the phonological system is available.

method could be compared, we can design experiments which mimic various challenges for the correspondence pattern recognition analysis. In the following, I will discuss three experiments in which the data is artificially modified in a controlled way in order to see how the method reacts to specific challenges.

5.3.1 Dealing with Artificially Seeded Borrowings. As a first experiment, let us consider cases of undetected borrowings in the data. While it is impossible to simulate borrowings realistically for the time being, we can use a simple workaround inspired by Dessimoz, Margadant, and Gonnet (2008) and tested on linguistic data in List (2015). This approach consists in the "seeding" of false borrowings among a certain number of language pairs in the data. Our version of this approach takes a pre-selected number of donor-recipient pairs and a pre-selected number of events as input and then randomly selects language pairs and word pairs from the data. For each event, one word is transferred from the donor to the recipient, and both items are marked as cognate. If an original counterpart is missing in the recipient language, the empty slot is filled by adding the word from the donor language.

In order to test the impact that the introduction of borrowings has on the analysis, I introduce a rough measure of cognate set regularity derived from the inferred correspondence patterns. This measure, which I call *pattern regularity* (PR) for convenience, uses the above-mentioned alignment site density scores for the correspondence patterns to which each alignment site in a given cognate set is attached and scores their regularity using a user-defined threshold. If less then half of all alignment sites are judged to be regular according to this procedure, the whole cognate set is assumed to be regular. If we encounter a cognate set in the data which is judged to be irregular according to this criterion, it is split up by assigning all words in the cognate sets to independent cognate sets. If a dataset is highly irregular, it will loose many cognate sets after applying this procedure, and accordingly, its cognate density will drop. By comparing the cognate density of the original dataset after applying the PR measure with a dataset that was distorted by artificial borrowings, it is possible to test the impact of undetected borrowings on the method directly.

Table 7 presents the results of this test. Based on tests with the training data, I set the PR threshold to 0.25 and ran 100 trials for each dataset, each time comparing the density in the original dataset and the dataset with the artificial borrowings for a controlled number of language pairs and a controlled number of borrowing events. The number of language pairs may seem rather high. This was intended, however, as I wanted to simulate spurious borrowings rather than intensive borrowings between only a few varieties (which would necessarily increase the pattern regularity). Based on the positive experience with the exclusion of gapped alignment sites, the same variant was used for these tests. As can be seen from the results in the table, the cognate density drops for most datasets when applying the PR measure. The only exception is Uralic, where density increases after adding the borrowings. The only explanation I have for this behaviour at the moment is that it results from the generally low cognate density of the dataset and the low phonetic diversity of the languages. If the languages are phonetically similar, borrowings do not surface as irregular correspondence patterns or cognate sets, and it is impossible to tell whether words have been regularly inherited or not. In the other cases, however, I am confident that the approach reflects the expected behaviour: if the data contains a considerable amount of undetected borrowings, this will disturb the correspondence patterns and decrease the pattern regularity of a dataset.

	Unmodified		Modified		Diff.	Lg.	Ev.
Dataset	**Orig. Ds.**	**PR Ds.**	**Orig. Ds.**	**PR Ds.**			
Bahnaric	0.76	0.51	0.77	0.47	0.04	288	331.78
Chinese	0.68	0.55	0.69	0.47	0.09	162	235.57
Huon	0.48	0.19	0.50	0.18	0.01	98	112.88
Romance	0.90	0.38	0.91	0.31	0.07	924	410.93
Tujia	0.63	0.59	0.64	0.56	0.03	12	48.95
Uralic	0.39	0.38	0.41	0.39	-0.01	24	72.09

Table 7
Comparing pattern regularity for artificially seeded borrowings in the data. The table contrasts the original density (Orig. Ds.) with the density after applying the pattern regularity measure (PR Ds.), both to the unmodified and the modified dataset. The last two columns show the number of languages pairs (Lg.) in which borrowings were introduced and the number of borrowing events (Ev.).

5.3.2 Dealing with Wrongly Assigned Cognates. In addition to undetected borrowings, the data can also suffer from wrong cognate assignments independent of borrowing, be it due to lookalikes which were erroneously judged to be cognate, or due to simple errors resulting from the annotation process. We can simulate these cases in a similar manner as was done with the seeding of artificial borrowings, by seeding erroneous words into the cognate sets in the data. In order to distinguish this experiment from the experiment on borrowings, but also to make it more challenging, I used LingPy's in-built method for word generation. This method takes a list of words as input and returns a generator (a Markov Chain) that generates new words from the input data with similar phonotactics. The method is by no means exact, employing a simple bigram model consisting of the original sound segment and a symbol indicating its prosodic position, following the prosodic model outlined in (List 2014, 119-134). For our purpose, however, it is sufficient, as we do not need the best possible model for the creation of pseudo-words, and the input data we can provide is in any case rather limited.

	Unmodified		Modified		Diff.	Lg.	Ev.
Dataset	**Orig. D.**	**PR D.**	**Orig. D.**	**PR D.**			
Bahnaric	0.76	0.51	0.76	0.47	0.04	4.0	400.0
Chinese	0.68	0.55	0.68	0.49	0.06	3.0	270.0
Huon	0.48	0.19	0.48	0.21	-0.02	2.0	134.75
Romance	0.90	0.38	0.90	0.24	0.15	8.0	440.0
Tujia	0.63	0.59	0.63	0.55	0.04	1.0	54.0
Uralic	0.39	0.38	0.39	0.37	0.01	1.0	78.99

Table 8
Comparing pattern regularity for artificially seeded neologisms in the data. The table contrasts the original density (Orig. D.) with the density after applying the pattern regularity measure (PR D.). The last two columns show the number of languages (L.) in which neologisms were introduced and the number of replacement events (Ev.).

The results of this second experiment are reported in Table 8. As can be seen from the table, the density drops at different degrees in all datasets except from Huon. We have to admit that we could not find an explanation for this outlier. All we can suspect

is that the very simple syllable structure of the languages may in fact yield words which are very similar to the words they were supposed to replace. Why this would lead to a slight increase of cognate density, however, is still not entirely clear for us. Nevertheless, in the other cases we are confident that our method picks up correctly the signals of disturbance in the data. The more erroneously assigned cognate sets we find in a given dataset, the more difficult it will be to find regular correspondence patterns.

5.3.3 Testing the Predictive Force of Correspondence Patterns. As a final experiment to be reported in this section, let us investigate the predictive force of correspondence patterns. Since the method for correspondence pattern recognition imputes missing data in its core, it can in theory also be used to predict how a given word should look in a given language if the reflex of the corresponding cognate set is missing. An example for the prediction of forms has been given above for the cognate set Dutch *dorp* and German *Dorf*. Since we know from Table 1 that the correspondence pattern of *d* in Dutch and German usually points to Proto-Germanic **þ*, we can propose that the English reflex (which is missing im Modern English) would start with *th*, if it was still preserved.[13] Since the method for correspondence pattern recognition assigns one or more correspondence patterns to each alignment site, even if the site has missing data for a certain number of languages, all that needs to be done in order to predict a missing entry is to look up the alignment pattern and check the value that is proposed for the given language variety.

How well the correspondence patterns in a given dataset predict missing reflexes can again be tested in a straightforward way by artificially introducing missing reflexes into the datasets. To make sure that the reflexes which should be predicted are in fact *predictable*, it is important to restrict both the number of reflexes which are deleted from a given dataset, as well as to delete only those reflexes from the data which appear in cognate sets of a certain size. In this way, we can guarantee that the method has a fair chance to identify missing data.

Following these considerations, the experiment was designed as follows: in 100 different trials, regular words from each dataset were excluded and the correspondence patterns were inferred from the modified datasets. The number of words to be excluded was automatically derived for each dataset by (a) selecting cognate sets whose size was at least half of the number of languages in the datasets, and (b) selecting one reflex of one third of the preselected cognate sets. As in some of the previous experiments, highly gapped sites were excluded from the analysis. The prediction rate per reflex was then computed by dividing the number of correctly predicted sites by the total number of sites for a given reflex. Given that the methods may assign one alignment site to more than one correspondence pattern, the number of correctly predicted sites was adjusted by taking the average number of correctly predicted sites when a fuzzy site was encountered. In order to learn more about the type of sounds which are best predicted by the method, the predictive force was computed not only for all sites, but also for vowels and consonants in separation.

The results of this experiment are provided in Table 9. As can be seen from the table, the prediction based on inferred correspondence patterns does not work overwhelmingly well, with only a small amount of the missing reflexes being correctly assigned. This does, however, not invalidate the method itself, but rather reflects the general problems

13 We ignore deliberately in this context that the alternative of the correspondence in Dutch and German would be a borrowing from Dutch, Frisian, or English to German.

			Predicted				
Dataset	**MSS**	**MR**	**All Sts.**	**Con. Sts.**	**Vow. Sts.**	**Ds.**	**Fz.**
Bahnaric	12	54	0.43	0.52	0.15	0.76	4.76
Chinese	9	52	0.51	0.58	0.31	0.68	5.79
Huon	7	13	0.48	0.54	0.32	0.48	3.93
Romance	21	51	0.45	0.48	0.27	0.90	4.73
Tujia	2	47	0.47	0.50	0.32	0.63	2.88
Uralic	3	31	0.45	0.50	0.29	0.39	3.44

Table 9
Predicting missing reflexes from the data. Column *MSS* shows the minimal size of cognate sets that were considered for the experiment. Column *MR* points to the number of reflexes which were excluded, *Ds.* provides the cognate density of the dataset, and *Fz.* the fuzziness of the assignment of patterns to alignment sites. In addition to the predictive force for all sites, consonants, and vowels, the density and the fuzziness of the alignment sites for each dataset are also reported.

we encounter when working with datasets of limited size in historical linguistics. Since the datasets in the test and training data are all of a smaller size, ranging between 110 and 210 concepts only, it is not generally surprising that the prediction of missing reflexes based on previously inferred regular correspondence patterns cannot yield highest accuracy scores. That we are dealing with general regularity issues (of small wordlists or of sound change processes in general) is also reflected in the fact that the prediction rate for consonants is much higher than the one for vowels. Given the limited design space of vowels opposed to consonants, vowel change is much more prone to idiosyncratic behavior than consonant change. This is also reflected in the experiment on the predictive force of automatically inferred correspondence patterns.

5.4 Examples

Inspecting the results of the analyses in due detail would go largely beyond the scope of this paper. To illustrate, however, how the analysis can aid in practical work on linguistic reconstruction, I want to provide an example from the Chinese test set. The Chinese data has the advantage of offering quick access to Middle-Chinese reconstructions for most of the items. Since Middle Chinese is only partially reconstructed on the basis of historical language comparison, and mostly based on written sources, such as ancient rhyme books and rhyme tables (Baxter 1992), the reconstructions are not entirely dependent on the modern dialect readings.

In Table 10, I have listed all patterns inferred by the method for correspondence pattern recognition for a reduced number of dialects (one of each major subgroup), which can all be reconstructed to a dental stop in Middle Chinese (*t, *tʰ or *d). If we only inspect the first four patterns in the table, we can see that the MC *d corresponds to two distinct patterns (# 85 and #135). Sūzhōu (SZ), one of the dialects of the Wú group, which usually inherit the three-stop distinction of voiceless, aspirated, and voiced stops in Middle Chinese, shows voiced [d] as expected in both patterns, but Běijīng, Guǎngzhōu and Fúzhōu have contrastive outcomes in both patterns ([tʰ] vs. [t]). When inspecting the tones which are reconstructed for the different words in Middle Chinese, we can easily find a conditioning context why the reflexes differ. The *píng* (flat) tone category in Middle Chinese correlates with aspiration, while the other tone categories correlate with

#	Cogn.	MC	MC Tones	BJ	SZ	CS	NC	MX	GZ	FZ	Dens.
177	14	*t	PS	t	t	t	t	t	t	t	6.43
76	13	*th	PSR	t^h	t^h	t^h	t^h	t^h	t^h	t^h	6.86
85	11	*d	P	t^h	d	t	t^h	t^h	t^h	t^h	6.86
135	5	*d	QR	t	d	t	t^h	t^h	t	t	3.00
197	4	*d	P	t^h	d	t	t^h	t^h	t^h	l	3.00
26	2	*d	S	t	Ø	Ø	t^h	s	t	l	0.86
220	1	*d	P	t^h	d	t	t^h	Ø	Ø	n	0.00

Table 10
Contrasting inferred correspondence patterns with Middle Chinese reconstructions (MC) and tone patterns (MC Tones: P: píng (flat), S: shǎng (rising), Q: qù (falling), R: rù (stop coda)) for representative dialects of the major groups (Běijīng, Sūzhōu, Chángshā, Nánchāng, Měixiàn, Guǎngzhōu, Fúzhōu).

devoicing in the three dialects.[14] If we had no knowledge of Middle Chinese, it would be harder to understand that both patterns correspond to the same proto-sound, but once assembled in such a way, it would still be much easier for scholars to search for a conditioning context that allows them to assign the same proto-sound to the two patterns in questions.

In pattern #197, we can easily see that Fúzhōu is showing an unexpected sound when comparing it with the other patterns in the table. If Fúzhōu had a [t^h] instead of the [l], we could merge it with pattern #85. The conditioning context for the deviation, which can again be quickly found when inspecting the data more closely, is due to a weakening of syllable-initial sounds in non-initial syllables in Fúzhōu, which can easily be seen when comparing the compound Fúzhōu [suɔʔ^4 lau^{52}] 'stone' (lit. 'stone-head') vs. the word [t^hau^{52}] 'head' in isolation. The same process can also be found in pattern #26, with the difference that the pattern corresponds to pattern #135, as the Middle Chinese words have one of the oblique tones. The reflex [s] in Méixiàn is irregular, though, resulting from an erroneous cognate judgment that links Fúzhōu [liaʔ^{23}] with Méixiàn [sɛ^{44}] 'to lick'. Although the final pattern looks irregular, given that it occurs only once, it can also be shown to be a variant of #85, since the reflex in Fúzhōu is again due to the weakening process, but this time resulting in assimilation with the preceding nasal (compare Fúzhōu [seiŋ^{52} nau^{31}] 'the front (front side)' with additional tone sandhi).

The example shows that, as far as the Middle Chinese dental stops are concerned, we do not find explicit exceptions in our data, but can rather see that multiple correspondence patterns for the same proto-sound may easily evolve. We can also see that a careful alignment and cognate annotation is crucial for the success of the method, but even if the cognate judgments are fine, but the data are sparse, the method may propose erroneous groupings. In contrast to manual work on linguistic reconstruction, where correspondence patterns are never regarded in the detail in which they are presented here, the method is a boost, especially in combination with tools for cognate annotation, like EDICTOR, to which we added a convenient way to inspect inferred correspondence patterns interactively. Since linguists can run the new method on their data and then directly inspect the consequences by browsing all correspondence patterns conveniently

14 This phenomenon most likely goes back to an earlier phonation contrast between the first (*píng*) tone in Middle Chinese and the other tones.

in the EDICTOR, the method makes it a lot easier for linguists to come up with first reconstructions or to identify problems in the data.

6. Conclusion and Outlook

In this study I have presented a new method for the inference of sound correspondence patterns in multi-lingual wordlists. Thanks to its integration with the LingPy software package, the methods can be applied both in the form of fully automated workflows where both cognate sets, alignments, and correspondence patterns are computed, or in computer-assisted workflows where linguists manually annotate parts of the data at any step in the workflow. Having shown that the inference of correspondence patterns can be seen as the crucial step underlying the reconstruction of proto-forms, the method presented here provides a basis for many additional approaches in the fields of computational historical linguistics and computer-assisted language comparison. Among these are (a) automatic approaches for linguistic reconstruction, (b) alignment-based approaches to phylogenetic reconstruction, (c) the detection of borrowings and erroneous cognates, and (d) the prediction of missing reflexes in the data. The approach is not perfect in its current form, and many kinds of improvements are possible. Given its novelty, however, I consider it important to share the approach its current form, hoping that it may inspire colleagues in the field to expand and develop it further.

Supplementary Material

The supplementary material contains the Python package, a short tutorial (as interactive Jupyter notebook and HTML) along with data illustrating how to use it, all the code that is needed to replicate the analyses discussed in this study along with usage instructions, the test and training data, and the expanded EDICTOR version in which correspondence patterns can be inspected in various interactive ways. The supplementary material has been submitted to the Open Science Framework for anonymous review. It can be accessed from the link `https://osf.io/mbzsj/?view_only=b7cbceac46da4f0ab7f7a40c2f457ada`.

References

Anttila, Raimo. 1972. *An introduction to historical and comparative linguistics*. Macmillan, New York.

Arnaud, Adam S., David Beck, and Grzegorz Kondrak. 2017. Identifying cognate sets across dictionaries of related languages. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2509–2518, Association for Computational Linguistics.

Baxter, William H. 1992. *A handbook of Old Chinese phonology*. de Gruyter, Berlin.

Beekes, Robert S. P. 1995. *Comparative Indo-European linguistics. An introduction*. John Benjamins, Amsterdam and Philadelphia.

Bhasker, J. and Tariq Samad. 1991. The clique-partitioning problem. *Computers & Mathematics with Applications*, 22(6):1 – 11.

Brown, Cecil H., David Beck, Grzegorz Kondrak, James K. Watters, and Søren Wichmann. 2011. Totozoquean. *International Journal of American Linguistics*, 77(3):323–372.

Brown, Cecil H., Eric W. Holman, and Søren Wichmann. 2013. Sound correspondences in the world's languages. *Language*, 89(1):4–29.

Campbell, Lyle and William John Poser. 2008. *Language classification: History and method*. Cambridge University Press, Cambridge.

Clackson, James. 2007. *Indo-European linguistics*. Cambridge University Press, Cambridge.

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

Dessimoz, C., D. Margadant, and G. H. Gonnet. 2008. DLIGHT – Lateral gene transfer detection using pairwise evolutionary distances in a statistical framework. In M. Vingron and L. Won, editors, *Research in Computational Molecular Biology*. Springer, Berlin and Heidelberg, pages 315–330.

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

Dunn, Michael. 2012. *Indo-European lexical cognacy database (IELex)*. Max Planck Institute for Psycholinguistics, Nijmegen.

Dybo, Anna and George S Starostin. 2008. In defense of the comparative method, or the end of the Vovin controversy. In I. S. Smirnov, editor, *Aspekty komparativistiki*, volume 3. RGGU, Moscow, pages 119–258.

Forkel, Robert, Johann-Mattis List, Michael Cysouw, and Simon J. Greenhill. 2017. *CLDF. Cross-Linguistic Data Formats. Version 1.0*. Max Planck Institute for the Science of Human History, Jena.

Fox, Anthony. 1995. *Linguistic reconstruction*. Oxford University Press, Oxford.

Gauch, Hugh G. 2003. *Scientific method in practice*, 1st edition. Cambridge University Press, Cambridge.

Geisler, H. and J.-M. List. 2013. Do languages grow on trees? The tree metaphor in the history of linguistics. In Heiner Fangerau, Hans Geisler, Thorsten Halling, and William Martin, editors, *Classification and evolution in biology, linguistics and the history of science. Concepts – methods – visualization*. Franz Steiner Verlag, Stuttgart, pages 111–124.

Greenhill, Simon J., Robert Blust, and Russell D. Gray. 2008. The Austronesian Basic Vocabulary Database: From bioinformatics to lexomics. *Evolutionary Bioinformatics*, 4:271–283.

Grimes, Joseph E. and Frederick B. Agard. 1959. Linguistic Divergence in Romance. *Language*, 35(4):598–604.

Guy, Jacques B. M. 1994. An algorithm for identifying cognates in bilingual wordlists and its applicability to machine translation. *Journal of Quantitative Linguistics*, 1(1):35–42.

Hetland, Magnus Lie. 2010. *Python algorithms. Mastering basic algorithms in the Python language*. Apress, New York.

Hoare, C. A. R. 1962. Quicksort. *The Computer Journal*, 5(1):10–16.

Hoenigswald, Henry Max. 1960. *Language change and linguistic reconstruction*, 4. aufl. 1966 edition. The University of Chicago Press and Univ. of Chicago Press, Chicago.

Holton, Gary, Marian Klamer, František Kratochvíl, Laura C. Robinson, and Antoinette Schapper. 2012. The historical relations of the Papuan languages of Alor and Pantar. *Oceanic Linguistics*, 51(1):86–122.

Hóu, Jīngī, editor. 2004. *Xiàndài Hànyǔ fāngyán yīnkù [Phonological database of Chinese dialects]*. Shànghǎi Jiàoyù, Shànghǎi.

Jacques, Guillaume. 2017. A reconstruction of Proto-Kiranti verb roots. *Folia Linguistica Historica*, 38(1):177–215.

Jäger, Gerhard, Johann-Mattis List, and Pavel Sofroniev. 2017. Using support vector machines and state-of-the-art algorithms for phonetic alignment to identify cognates in multi-lingual wordlists. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics. Long Papers*, pages 1204–1215, Association for Computational Linguistics, Valencia.

Kay, Martin. 1964. *The logic of cognate recognition in historical linguistics*. The RAND Corporation, Santa Monica.

Kondrak, Grzegorz. 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*, pages 288–295.

Kondrak, Grzegorz. 2002. Determining Recurrent Sound Correspondences by Inducing Translation Models. In *Nineteenth International Conference on Computational Linguistics (COLING 2002)*, pages 488–494, Taipei.

Kondrak, Grzegorz. 2003. Identifying complex sound correspondences in bilingual wordlists. In Alexander Gelbukh, editor, *Computational linguistics and intelligent text processing*. Springer, Berlin, pages 432–443.

Kondrak, Grzegorz. 2009. Identification of cognates and recurrent sound correspondences in word lists. *Traitement Automatique des Langues*, 50(2):201–235.

Kroonen, Guus. 2013. *Etymological dictionary of Proto-Germanic*. Number 11 in Leiden Indo-European Etymological Dictionary Series. Brill, Leiden and Boston.

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

List, Johann-Mattis. 2012a. 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*, pages 117–125, Stroudsburg.

List, Johann-Mattis. 2012b. SCA: Phonetic alignment based on sound classes. In Marija Slavkovik and Dan Lassiter, editors, *New directions in logic, language, and computation*. Springer, Berlin and Heidelberg, pages 32–51.

List, Johann-Mattis. 2014. *Sequence comparison in historical linguistics*. Düsseldorf University Press, Düsseldorf.

List, Johann-Mattis. 2015. Network perspectives on Chinese dialect history. *Bulletin of Chinese Linguistics*, 8:42–67.

List, Johann-Mattis. 2016a. Beyond cognacy: Historical relations between words and their implication for phylogenetic reconstruction. *Journal of Language Evolution*, 1(2):119–136.

List, Johann-Mattis. 2016b. Computer-Assisted Language Comparison: Reconciling Computational and Classical Approaches in Historical Linguistics. Technical report, Max Planck Institute for the Science of Human History, Jena.

List, Johann-Mattis. 2017. A web-based interactive tool for creating, inspecting, editing, and publishing etymological datasets. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics. System Demonstrations*, pages 9–12, Association for Computational Linguistics, Valencia.

List, Johann-Mattis, Simon Greenhill, and Robert Forkel. 2017. *LingPy. A Python library for quantitative tasks in historical linguistics*. Max Planck Institute for the Science of Human History, Jena.

List, Johann-Mattis, Simon J. Greenhill, and Russell D. Gray. 2017. The potential of automatic word comparison for historical linguistics. *PLOS ONE*, 12(1):1–18.

List, Johann-Mattis, Philippe Lopez, and Eric Bapteste. 2016. Using sequence similarity networks to identify partial cognates in multilingual wordlists. In *Proceedings of the Association of Computational Linguistics 2016 (Volume 2: Short Papers)*, pages 599–605, Association of Computational Linguistics, Berlin.

McElhanon, Kenneth A. 1967. Preliminary Observations on Huon Peninsula Languages. *Oceanic Linguistics*, 6(1):1–45.

Meier-Brügger, Michael. 2002. *Indogermanische Sprachwissenschaft*, 8 edition. de Gruyter, Berlin and New York.

Meillet, Antoine. 1908. *Les dialectes Indo-Européens*. Librairie Ancienne Honoré Champion, Paris.

Needleman, Saul B. and Christan D. Wunsch. 1970. A gene method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48:443–453.

Newman, M. E. J. 2010. *Networks. An Introduction*. Oxford University Press, Oxford.

Notredame, Cédric, Desmond G. Higgins, and Jaap Heringa. 2000. T-Coffee. *Journal of Molecular Biology*, 302:205–217.

Prokić, Jelena, Martijn Wieling, and John Nerbonne. 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*, pages 18–25.

Rama, Taraka, Johannes Wahle, Pavel Sofroniev, and Gerhard Jäger. 2017. Fast and unsupervised methods for multilingual cognate clustering. *CoRR*, abs/1702.04938.

Rosvall, M. and C. T. Bergstrom. 2008. Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci. U.S.A.*, 105(4):1118–1123.

Saenko, Mikhail. 2015. Annotated Swadesh wordlists for the Romance group (Indo-European family). In George Starostin, editor, *The Global Lexicostatistical Database*. RGGU, Moscow.

Schleicher, August. 1852. *Die Formenlehre der kirchenslawischen Sprache. Erklärend und vergleichend dargestellt*. H. B. König, Bonn.

Schneider, T. D. 2002. Consensus sequence Zen. *Applied Bioinformatics*, 1(3):111–119.

Sidwell, Paul. 2015. Austroasiatic dataset for phylogenetic analysis: 2015 version. *Mon-Khmer Studies (Notes, Reviews, Data-Papers)*, 44:lxviii–ccclvii.

Starostin, George S. 2013. Annotated Swadesh wordlists for the Tujia group (Sino-Tibetan family). In George Starostin, editor, *The Global Lexicostatistical Database*. RGGU, Moscow.
Syrjänen, Kaj, Terhi Honkola, Kalle Korhonen, Jyri Lehtinen, Outi Vesakoski, and Niklas Wahlber. 2013. Shedding more light on language classification using basic vocabularies and phylogenetic methods. *Diachronica*, 30(3):323–352.
Turchin, Peter, Ilja Peiros, and Murray Gell-Mann. 2010. Analyzing genetic connections between languages by matching consonant classes. *Journal of Language Relationship*, 3:117–126.
Wagner, Robert A. and Michael J. Fischer. 1974. The string-to-string correction problem. *Journal of the Association for Computing Machinery*, 21(1):168–173.
Wang, William Shi-Yuan. 2006. *Yǔyán, yǔyīn yǔ jìshù*. Xiānggǎng Chéngshì Dàxué, Shànghǎi .
Welsh, D. J. A. and M. B. Powell. 1967. An upper bound for the chromatic number of a graph and its application to timetabling problems. *The Computer Journal*, 10(1):85–86.
北京大学, Běijīng Dàxué. 1964. *Hànyǔ fāngyán cíhuì* 汉语方言词汇 *[Chinese dialect vocabularies]*. Wénzì Gǎigé 文字改革, Běijīng 北京.

Appendix A: Training Data

The following table gives a summary on the training data used in the study.

Dataset	Source	Languages	Concepts	Cognates	Density
Austronesian	Greenhill et al. (2008)	20	210	2864	0.34
Bai	Wang (2006)	9	110	285	0.73
Chinese	Hóu (2004)	15	140	1189	0.60
IndoEuropean	Dunn (2012)	20	207	1777	0.60
Japanese	Hattori (1973)	10	200	460	0.70
ObUgrian	Zhivlov (2011)	21	110	242	0.88
Bahnaric	Sidwell (2015)	24	200	1055	0.76
Chinese	Běijīng Dàxué (1964)	18	180	1231	0.68
Huon	McElhanon (1967)	14	139	855	0.48
Romance	Saenko (2015)	43	110	465	0.90
Tujia	Starostin (2013)	5	109	179	0.63
Uralic	Syrjänen et al. (2013)	7	173	870	0.39

Appendix B: Inspecting Correspondence Patterns in EDICTOR

The following screenshots shows how the modified version of the EDICTOR allows for an enhanced inspection of sound correspondence patterns inferred by the method.

Investigate correspondence patterns in the data

Select Sets | THR. 4 | PREV. 54 | OK | ← | 1-25 of 25 Sites | →

COGNATES	INDEX	PATTERN	CONCEPTS	Bei	Suz	Cha	Nan	Mei	Gua	Fuz	SIZE
646	1	tʰ / 85	the body hair (hair or fur)	tʰ	d	t	tʰ	Ø	tʰ	tʰ	5.14 / 7
649	1	tʰ / 85	the hair (of the head)	tʰ	d	t	tʰ	Ø	tʰ	tʰ	5.14 / 7
740	1	tʰ / 85	the big frog	tʰ	d	t	tʰ	tʰ	tʰ	Ø	5.14 / 7
189	5	tʰ / 85	the fish (one piece of fish)	tʰ	d	t	Ø	tʰ	tʰ	Ø	5.14 / 7
607	5	tʰ / 85	the wood (material)	tʰ	d	t	tʰ	tʰ	Ø	Ø	5.14 / 7
948	5	tʰ / 85	the upper level (above)	tʰ	d	t	tʰ	Ø	Ø	Ø	5.14 / 7
965	5	tʰ / 85	the lower level (below)	tʰ	d	t	tʰ	Ø	Ø	Ø	5.14 / 7
1038	1	tʰ / 76	to hear	tʰ	tʰ	tʰ	tʰ	tʰ	tʰ	tʰ	4.00 / 5
1097	1	tʰ / 76	to pull	tʰ	tʰ	tʰ	tʰ	tʰ	tʰ	tʰ	4.00 / 5
1058	1	tʰ / 76	to lick	tʰ	tʰ	tʰ	tʰ	Ø	tʰ	Ø	4.00 / 5
1085	1	tʰ / 76	to push	tʰ	tʰ	tʰ	tʰ	Ø	tʰ	Ø	4.00 / 5
380	1	tʰ / 76	to make a journey	Ø	tʰ	tʰ	tʰ	Ø	tʰ	Ø	4.00 / 5
598	6	t / 177	the sickle	t	t	t	t	Ø	t	Ø	2.57 / 4
667	5	t / 177	the flower (one piece of flower)	t	t	t	t	Ø	t	Ø	2.57 / 4
673	4	t / 177	the ear	t	t	t	t	Ø	Ø	Ø	2.57 / 4
432	1	t / 177	all	t	Ø	Ø	t	t	Ø	t	2.57 / 4