Computer-Assisted Language Comparison: State of the Art

Authors: Wu, Mei-Shin[1]; Schweikhard, Nathanael E.[1]; Bodt, Timotheus A.[2]; Hill, Nathan W.[2]; List, Johann-Mattis[1]

Affiliations:

1. Department of Linguistic and Cultural Evolution, Max Planck Institute for the Science of Human History, Jena, Germany

2. SOAS, University of London, London, UK

Abstract

Historical language comparison opens windows onto a human past, long before the availability of written records. Since traditional language comparison within the framework of the comparative method is largely based on manual data comparison, requiring the meticulous sifting through dictionaries, word lists, and grammars, the framework is difficult to apply, especially in times where more and more data have become available in digital form. Unfortunately, it is not possible to simply automate the process of historical language comparison, not only because computational solutions lag behind human judgments in historical linguistics, but also because they lack the flexibility that would allow them to integrate various types of information from various kinds of sources. A more promising approach is to integrate computational and classical approaches within a *computer-assisted framework*, “neither completely computer-driven nor ignorant of the assistance computers afford” [1, p. 4]. In this paper, we will illustrate what we consider the current state of the art of computer-assisted language comparison by presenting a workflow that starts with raw data and leads up to a stage where sound correspondence patterns across multiple languages have been identified and can be readily presented, inspected, and discussed. We illustrate this workflow with the help of a newly prepared dataset on Hmong-Mien languages. Our illustration is accompanied by Python code and instructions on how to use additional web-based tools we developed so that users can apply our workflow for their own purposes.

Keywords: computer-assisted, language comparison, historical linguistics, Hmong-Mien language family

1 Introduction

There are few disciplines in the humanities that show the impact of quantitative, computer-based methods as strongly as historical linguistics. While individual scholarship and intuition had played a major role for a long time, with only minimal attempts to formalize or automatize the painstaking methodology, the last twenty years have seen a rapid increase in quantitative applications. Quantitative approaches are reflected in the proposal of new algorithms that automate what was formerly done by inspection alone [2], in the publication of large cross-linguistic databases that allow for a data-driven investigation of linguistic diversity [3], and in numerous publications in which the new methods are used to tackle concrete questions on the past of the world's languages (for recent examples, see [4, 5]).

While it is true that — due to increasing amounts of data — the classical methods are reaching their practical limits, it is also true that computer applications are still far from being able to replace experts' experience and intuition, especially in those cases where data are sparse (as they are still for many language families). If computers cannot replace experts and experts do not have enough time to analyse the massive amounts of data, a new framework is needed, neither completely computer-driven nor ignorant of the assistance computers provide. Current machine translation systems, for example, are efficient and consistent, but they are by no means accurate, and no one would use them in place of a trained expert. Trained experts, on the other hand, do not necessarily work consistently and efficiently. In order to enhance both the quality of machine translation and the efficiency and consistency of human translation, a new paradigm of computer-assisted translation has emerged [6].

Following the idea of computer-assisted frameworks in translation and biology, scholars have begun to propose frameworks for *computer-assisted language comparison* (CALC), in which the flexibility and intuition of human experts is combined with the efficiency and consistency of computational approaches. In this study, we want to introduce what we consider

the state of the art[1] in this endeavor, and describe a workflow that starts from raw, cross-linguistic data. These raw data are then consistently lifted to the level of an etymologically annotated dataset, using advanced algorithms for historical language comparison along with interactive tools for data annotation and curation.

2 A workflow for computer-assisted language comparison

Our workflow consists of 5 stages, as shown in Figure 1. It starts from *raw data* (tabular data from fieldwork notes or data published in books and articles) which we re-organize and re-format in such a way that the data can be automatically processed (Step 1). Once we have lifted the data to this stage, we can infer sets of etymologically related words (*cognate sets*) (Step 2). In this first stage, we only infer cognates inside the same *meaning slot*. That means that all cognate words have the same meaning in their respective languages. Once this has been done, we *align* all cognate words *phonetically* (Step 3). Since we only infer cognate words that have the same meaning in Step 2, we now use a new method to infer cognates *across meanings* by employing the information in the aligned cognate sets (Step 4). Finally, in Step 5, we employ a recently proposed method for the detection of correspondence patterns [7] in order to infer sound correspondences across the languages in our sample.

1 By “state of the art”, we refer to approaches that have been developed during the past two decades and are available in the form of free software packages that can be used on all major computing platforms and have shown to outperform alternative proposals in extensive tests. These approaches themselves build on both qualitative and quantitative considerations that have been made in the field of historical linguistics during the past two centuries (for early quantitative and formal approaches, compare, for example, Hoenigswald [40] and Kay [41]).

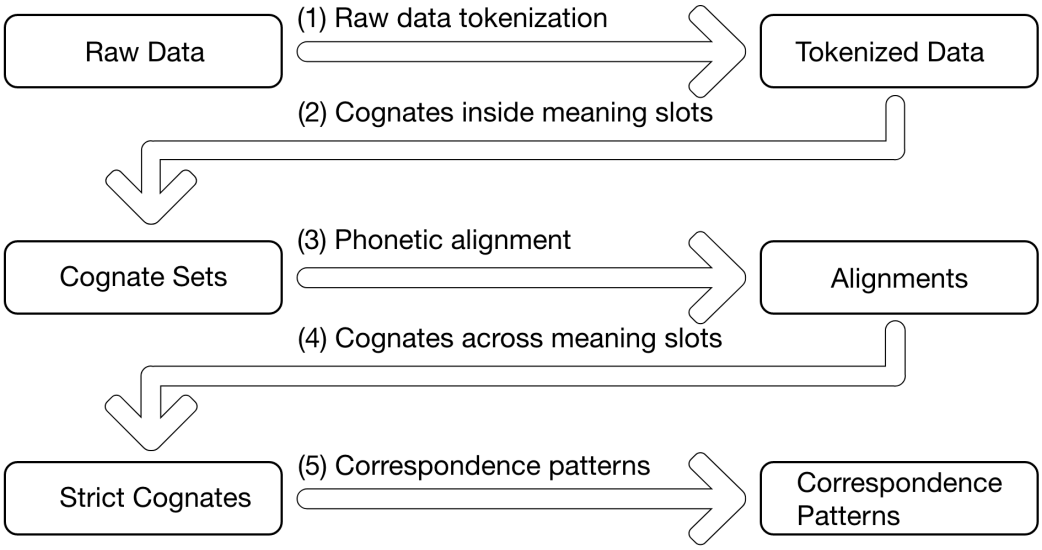


Figure 1: An overview of the workflow.

Our workflow is strictly *computer-assisted,* and by no means solely *computer-based*. That means that during each stage of the workflow, the data can be manually checked and modified by experts and then used in this modified form in the next stage of the workflow. Our goal is not to replace human experts, but to increase the efficiency of human analysis by providing assistance especially in those tasks which are time consuming, while at the same time making sure that any manual input is checked for internal consistency.

Our study is accompanied by a short tutorial along with code and data needed to replicate the studies illustrated in the following. The workflow runs on all major operating systems. In addition, we have prepared a Code Ocean capsule[2] to allow users to test the workflow without installing the software.

3 Illustration of the workflow

3.1 Dataset

The data we use was originally collected by Chén (2012) [8], later added in digital form to the

2 The permanent link of the Code Ocean Capsule is : https://codeocean.com/capsule/8178287/tree/v2

SEALANG project [9], and was then converted to a computer-readable format as part of the CLICS database (https://clics.clld.org, [10]). Chén's collection comprises 885 concepts translated into 25 Hmong-Mien varieties. Hmong-Mien languages are spoken in China, Thailand, Laos and Vietnam in Southeast Asia. Scholars divide the family into two main branches, Hmong and Mien. The Hmong-Mien languages have been developing in close contact with neighboring languages from different language families (Sino-Tibetan, Tai-Kadai, Austroasiatic, and Austronesian [11, p. 224]). Chén's study concentrates on Hmong-Mien varieties spoken in China.

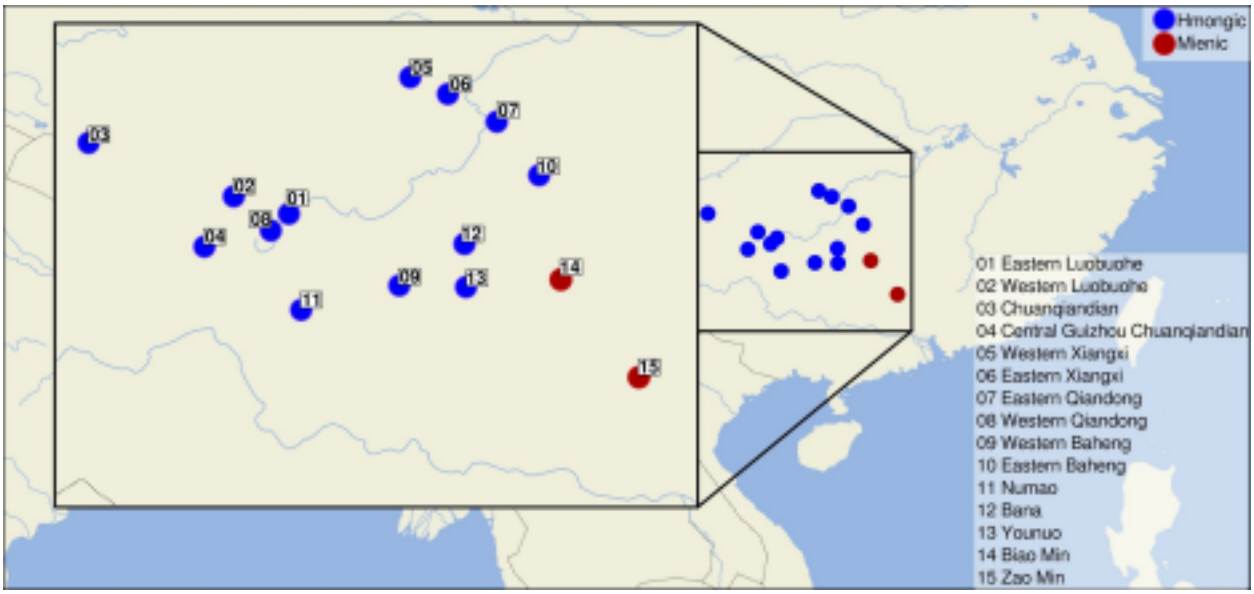


Figure 2: The geographic distribution of the Hmong-Mien languages selected for our sample.

In order to make sure that the results can be easily inspected, we decided to reduce the data by taking a subset of 502 concepts of 15 varieties from the dataset. While we selected the languages due to their geographic distribution and their representativeness with respect to the Hmong-Mien language family, we selected the concepts for reasons of comparability with previous linguistic studies. We focus both on concepts that are frequently used in general studies in historical linguistics (reflecting the so-called *basic vocabulary* [12-15]), and concepts that have been specifically applied in studies on Southeast Asian languages [4, 16-19]. The 15

varieties are shown in their geographic distribution in Figure 2. While the reduction of the data is done for practical reasons, since smaller datasets can be more easily inspected manually, the workflow can also be applied to the full dataset, and we illustrate in the tutorial how the same analysis can be done with all languages in the original data sample.

3.2 Workflow

3.2.1 From raw data to tokenized data

As a first step, we need to lift the data to a format in which they can be automatically digested. Data should be human- and machine-readable at the same time. Our framework works with data in *tabular form*, which is usually given in a simple text file in which the first line serves as table header and the following lines provide the content. In order to apply our workflow, each word in a given set of languages must be represented in one row of the data table, and four obligatory values need to be supplied: an identifier (ID), the name of the language variety (DOCULECT), the elicitation gloss for the concept (CONCEPT), and a phonetic transcription of the word form, provided in tokenized form (TOKENS). Additional information can be flexibly added by placing it in additional columns. Table 1 gives a minimal example for four words in Germanic languages.

ID	DOCULECT	CONCEPT	VALUE	TOKENS
1	English	house	house	h aʊ s
2	German	house	Haus	h au s
3	Dutch	house	huis	h ʊɪ s
4	Swedish	house	hus	h ʉː s

Table 1 A minimal example for four words in four Germanic languages, given in our minimal tabular format. The column VALUE (which is not required) provides the orthographical form of each word [20, 21].

As can be seen from Table 1, the main reference of our algorithms is the phonetic transcription in its *tokenized form* as provided by the column TOKENS. Tokenized, in this context, means that the transcription explicitly marks what an algorithm should treat as one

sound segment. In Table 1, for example, we have decided to render *diphthongs* as one sound. We could, of course, also treat them as two sounds each, but since we know that diphthongs often evolve as a single unit we made this explicit decision with respect to the tokenization.

Transcriptions are usually not provided in tokenized form. The tokenization thus needs to be done prior to analyzing the data further. While one can easily manually tokenize a few words as shown in Table 1, it becomes tedious and error-prone to do so for larger datasets. In order to increase the consistency of this step in the workflow, we recommend using ***orthography profiles*** [22]. An orthography profile can be thought of as a simple text file with two columns in which the first column represents the values as one finds them in the data, and the second column allows to convert the exact sequence of characters that one finds in the first column into the desired format. An orthography profile thus allows tokenizing a given transcription into meaningful units. It can further be used to modify the original transcription by replacing tokenized units with new values.[3] How an orthography profile can be applied is illustrated in more detail in Figure 3.

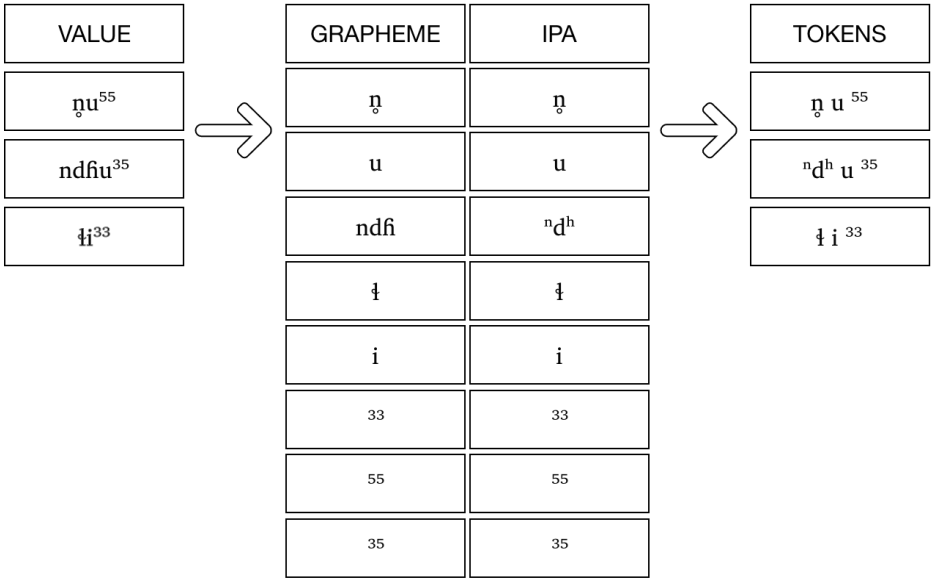


Figure 3: An example to illustrate the usage of orthography profiles to tokenize the phonetic transcriptions.

3 Orthography profiles proceed in a greedy fashion, converting grapheme sequences in the reverse order of their length, thus starting from the longest grapheme sequence.

Our data format can be described as a *wide-table format* [23-25] and conforms to the strict principle of entering only *one value per cell* in a given data table. This contrasts with the way in which linguists traditionally code their data, as shown in Table 2, where we contrast the original data from Chén with our normalized representation. To keep track of the original data, we reserve the column VALUE to store the original word forms, including those cases where multiple values are placed in the same cell. The separated forms are placed in the column FORM, which itself is converted into a tokenized transcription with help of orthography profiles.

English	Chinese	Bana	Numao	Zao Min	Biao Min
moon	月亮	la^{04}la^{35}	ɬo^{44}	lo^{42}	la^{53}gwaŋ^{33}
sun	太陽	la^{04}ni^{13}	ma^{42}n̥aŋ^{33}	ʔa^{53}nai^{44}	n̥i^{21}tau^{31}
mother	母親	ʔa^{04}ŋa^{313}	mai^{33}	ni^{44}; ʑe^{44}	n̥a^{31}

a) Raw data as given in the digitized version of Chéns (2012) book.

ID	DOCULECT	SUBGROUP	CONCEPT	VALUE	TOKENS
1	Bana	Hmongic	moon	la^{04}la^{35}	l a $^{0/4}$ + l a ^{35}
2	Numao	Hmongic	moon	ɬo^{44}	ɬ o ^{44}
3	ZaoMin	Mienic	moon	lo^{42}	l o ^{42}
4	BiaoMin	Mienic	moon	la^{53}gwaŋ^{33}	l a ^{53} + g w a ŋ ^{33}
5	Bana	Hmongic	sun	la^{04}ni^{13}	l a $^{0/4}$ + n i ^{13}
6	Numao	Hmongic	sun	ma^{42}n̥aŋ^{33}	m a ^{42} + n̥ a ŋ ^{33}
7	ZaoMin	Mienic	sun	ʔa^{53}nai^{44}	ʔ a ^{53} + n ai ^{44}
8	BiaoMin	Mienic	sun	n̥i^{21}tau^{31}	n̥ i ^{21} + t au ^{31}
9	Bana	Hmongic	mother	ʔa^{04}ŋa^{313}	ʔ a $^{0/4}$ + ŋ a ^{313}
10	Numao	Hmongic	mother	mai^{33}	m ai ^{53}
11	ZaoMin	Mienic	mother	**ni^{44}**; ʑe^{44}	**n i ^{44}**
12	ZaoMin	Mienic	mother	ni^{44}; **ʑe^{44}**	**ʑe^{44}**

13	BiaoMin	Mienic	mother	nˌa³¹	nˌ a ³¹

b) Long-table format in which tokenized forms (TOKENS) have been added, and language names have been normalized.

Table 2: The transformation from raw to machine-readable data. As illustrated in Table 1, the VALUE column displays the raw form. The tokenized forms are added to the TOKENS column.

In order to make sure that our data is comparable with other datasets, we follow the recommendations by the Cross-Linguistic Data Formats initiative (CLDF, https://cldf.clld.org, [24]) and link our languages to the Glottolog database (https://glottolog.org, [26]), our concepts to the Concepticon (https://concepticon.clld.org, [27]), and follow the transcription standards proposed by the Cross-Linguistic Transcription Systems initiative (https://clts.clld.org, [28]).

In the accompanying tutorial, we show how the data can be retrieved from CLDF format and converted into plain tabular format. We also show how the original data can be tokenized with the help of an orthography profile (TUTORIAL 3.1).

3.2.2 From tokenized data to cognate sets

Having transformed the original data into machine-readable format, we can start to search for words in the data which share a common origin. These *etymologically related* words (also called *cognates*) are the first and most crucial step in historical language comparison. The task is not trivial, especially when dealing with languages that diverged a long time ago. A crucial problem is that words are often not entirely cognate across languages [29]. What we find instead is that languages share *cognate* ***morphemes***[4] (word parts). When languages make frequent use of *compounding* to coin new words, such as in Southeast Asian languages, *partial* ***cognacy*** is rather the norm than the exception, which is well-known to historical linguists working in this area [30]. We explicitly address partial cognacy by adopting a numerical annotation in which each morpheme instead of each word form is assigned to a specific cognate set [31], as shown in Figure 4.

4 Linguistic terms which are further explained in our glossary, submitted as part of the supplementary information, are marked in bold font the first time they are introduced.

DOCULECT	CONCEPT	TOKENS	COGID	COGIDS
Chuanqiandian	SUN	n̥ o ⁴³	1	1
Numao	SUN	m a ⁴² + n̥ a ŋ ³³	2	2 1
ZaoMin	SUN	ʔ a ⁵³ + n ai ⁴⁴	3	3 1
EasternBaheng	SUN	l a ⁰/³ + n̥ e ³⁵	4	4 1

Figure 4: The comparison of full cognates (COGID) and partial cognate sets (COGIDS). While none of the four words is entirely cognate with each other, they all share a common element. Note that the IDs for full cognates and partial cognates are independent from each other. For reasons of visibility, we have marked the partial cognates shared among all language varieties in red font.

In order to infer partial cognates in our data, we make use of the partial cognate detection algorithm proposed by List et al. [32], which is, so far, the only algorithm available that has been proposed to address this problem. In the tutorial submitted along with this paper, we illustrate in detail how partial cognates can be inferred from the data and how the results can be inspected (TUTORIAL 3.2). In addition, the tutorial quickly explains how the web-based EDICTOR tool (https://digling.org/tsv/, [33]) can be used to manually correct the partial cognates identified by the algorithm (TUTORIAL 3.2).

3.2.3 From cognate sets to alignments

An alignment analysis is a very general and convenient way to compare sequences of various kinds. The basic idea is to place two sequences into a matrix in such a way that corresponding segments appear in the same column, while placeholder symbols are used to represent those cases where a corresponding segment is lacking (Figure 5) [34]. As the core of historical language comparison lies in the identification of regularly recurring sound correspondences across cognate words in genetically related languages, it is straightforward to make use of alignment analyses once cognates have been detected in order to find patterns of corresponding sounds. In addition to building the essential step for the identification of sound

correspondences, alignment analyses also make it easier for scholars to inspect and correct algorithmic findings.

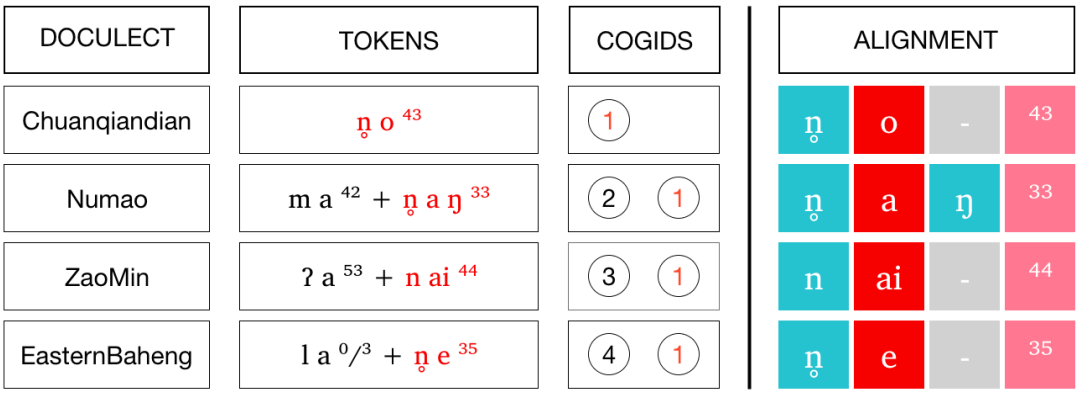


Figure 5: The alignment of 'sun' (cognate ID 1) among 4 Hmong-Mien languages, with segments colored according to their basic sound classes. The table on the left shows the cognate identifiers for cognate morphemes, as discussed in Figure 4. The table on the right shows how the cognate morphemes with identifier 1 (basic meaning sun") are aligned.

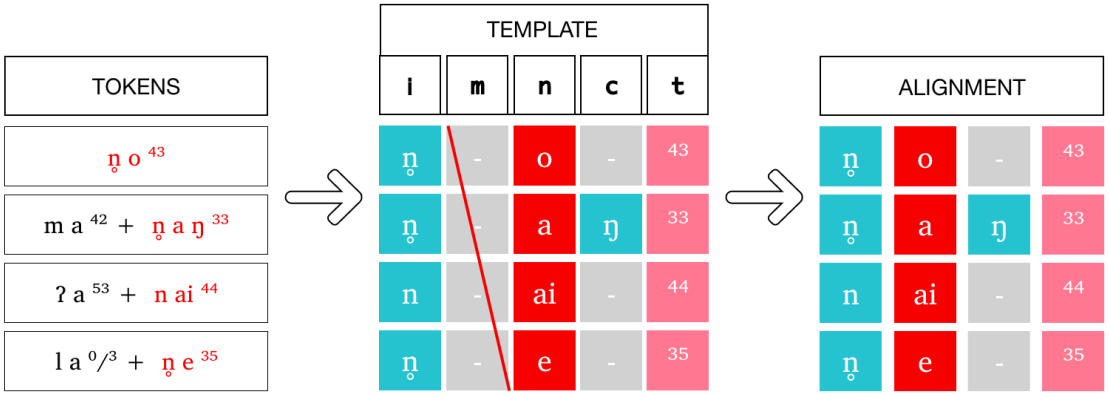
Phonetic alignment algorithms have greatly improved during the last 20 years. The most popular alignment algorithms used in the field of historical linguistics today all have their origin in alignment applications developed for biological sequence comparison tasks, which were later adjusted and modified for linguistic purposes [34].

While the currently available alignment algorithms are all very complex, scholars often forget that the same amount of algorithmic complexity is not needed for all languages. Since most Southeast Asian languages have fixed *syllable templates*, alignments are often predicted by the syllable structure. As a result, one does not need to employ complicated sequence comparison methods in order to find the right matchings between cognate morphemes. All one needs to have is a template-representation of each morpheme in the data.

As an example, consider the typical template for many Southeast Asian languages [35]: syllables consist maximally of an initial consonant (i), a medial glide (m), a nucleus vowel (n), a coda consonant (c), and the tone (t). Individual syllables do not need to have all these positions filled, as can be seen in the following example in Figure 6a.[5]

5 Note that this template of *i(nitial) m(edial) n(ucleus) c(oda)* and *t(one)* is generally sufficient to represent

a) Representing prosodic structure reflecting syllable templates for each morpheme in the data.



b) Aligning tokenized transcriptions to templates, and deleting empty slots.

Figure 6: Illustration of the template-based alignment procedure.

Once the templates of all words are annotated, aligning any word with any other word is extremely simple. Instead of aligning the words with each other, we simply align them to the

all syllables in the Hmong-Mien data we consider here. Seemingly complex cases, such as *ntsæn*^{22} “clear”, for example, can be handled by treating *nts* as one (initial) sound, resulting in a phonetic transcription of [ⁿts æ n ^{22}].

template, by filling those spots in the template which have no sounds with gap symbols ("-"). We can then place all words that have been aligned to a template in our alignment and only need to delete those columns in which only gaps occur, as illustrated in Figure 6b.

Our accompanying tutorial illustrates how template-based alignments can be computed from the data (TUTORIAL 3.3). In addition, we also show how the alignments can be inspected with the help of the EDICTOR tool (TUTORIAL 3.3).

3.2.4 From alignments to cross-semantic cognates

As in many Southeast Asian languages, most morphologically complex words in Hmong-Mien languages are *compounds*, as shown in Table 3. The word for 'fishnet' in Northeast Yunnan Chuanqiandian, for example, is a combination of the morpheme meaning 'bed' [dzʱaɯ³⁵] and the morpheme meaning 'fish' [ⁿpə⁵³].[6] The word for 'eagle' in Dongnu is composed of the words [po⁵³] 'father' and [tɬəŋ⁵³] 'hawk'. As can be seen from the word for 'bull' in the same variety, [po⁵³vɔ²³¹], [po⁵³] can be used to denote male animals, but in the word for 'eagle' it is more likely to denote strength [8, p. 328]. As a final example, Younuo lexicalizes the concept 'tears' as [ki⁵⁵mo³²ʔŋ⁴⁴], with [ki⁵⁵mo³²] meaning 'eye' and [ʔŋ⁴⁴] meaning 'water'.

An important consequence of the re-use of word parts in order to form new words in highly isolating languages of Southeast Asia is that certain words are not only cognate *across* languages, but also *inside* one and the same language. However, since our algorithm for partial cognate detection only identifies those word parts as cognate which appear in words denoting the same meaning, we need to find ways to infer the information on ***cross-semantic cognates*** in a further step.

DOCULECT	GLOSS	VALUE	TOKENS	MORPHEMES
Northeast-Yunnan-Chuanqiandian	fishnet	dzʱaɯ³⁵mpə³³	dzʱ aɯ ³⁵ + ⁿp ə ³³	bed fish
	fish	mpə³³	ⁿp ə ³³	fish

6 We are aware of the fact that the transcriptions by Chén are not entirely "phonetic", but since they are much less phonologically abstract than, for example, the transcriptions provided by Ratliff [11], we prefer to place them in phonetic rather than phonological brackets.

	bed	dzɦaɯ^{35}	dz^{ɦ} aɯ ^{35}	bed
	net	dzɦo^{33}	dz^{ɦ} o ^{35}	net
Dongnu	bull	po^{53}vɔ^{231}	p o ^{53} + v ɔ ^{231}	father cow
	eagle	po^{53}tɬəŋ^{53}	p o ^{53} + tɬ ə ŋ ^{53}	father hawk
	father	po^{53}	p o ^{53}	father
	bovine	vɔ^{231}	v ɔ ^{231}	cow
	hawk	tɬəŋ^{53}	tɬ ə ŋ ^{53}	hawk
Younuo	tear	ki^{55}mo^{32}ʔŋ^{44}	k i ^{55} + m o ^{32} + ʔ ŋ ^{44}	ki-suffix eye water
	water	ʔŋ^{44}	ʔ ŋ ^{44}	water
	eye	ki^{55}mo^{32}	k i ^{55} + m o ^{32}	ki-suffix eye

Table 3: Examples of ***compound words*** in Hmong-Mien languages. The column MORPHEMES uses morpheme glosses [31] in order to indicate which of the words are cognate inside the same language. The form for 'net' in the table serves to show that 'bed' and 'net' are not colexified, and that instead 'fishnet' is an analogical compound word.

As an example, consider the data for 'son' and 'daughter' in five language varieties of our illustration data. As can be seen immediately, two languages, Chuanqiandian and East Qiandong, show striking partial ***colexifications*** for the two concepts. In both cases, one morpheme recurs in the words for the two concepts. In the other cases, we find different words, but if we compare the overall cognacy, we can also see that all five languages share one cognate morpheme for 'son' (corresponding to the Proto-Hmong-Mien *tu̯ɛn in Ratliff's reconstruction [11]), and three varieties share one cognate morpheme for 'daughter' (corresponding to *mphje D in Ratliff's reconstruction), with the morpheme for 'son' occurring also in the words for 'daughter' in East Qiandong and Chuanqiandian, as mentioned before.

DOCULECT	CONCEPT	FORM	Cognacy	Cross-Semantic
EasternBaheng	SON	taŋ^{35}	1	1
EasternBaheng	DAUGHTER	p^{ɦ}je^{53}	2	2
WesternBaheng	SON	ʔa$^{3/0}$ + taŋ^{35}	3 1	3 1
WesternBaheng	DAUGHTER	ta^{55} + qa$^{3/0}$ + t^{ɦ}jei^{53}	4 5 6	4 5 6

Chuanqiandian	SON	to^{43}	1	1
Chuanqiandian	DAUGHTER	^{n}ts^{h}ai^{33}	7	7
CentralGuizhouChuanqiandian	SON	tə$^{2/0}$ + tə̃^{24}	8 1	8 1
CentralGuizhouChuanqiandian	DAUGHTER	tə̃^{24} + ^{n}p^{h}e^{42}	9 2	1 2
EasternQiandong	SON	tei^{24}	1	1
EasternQiandong	DAUGHTER	tei^{24} + p^{h}a^{35}	9 2	1 2

Table 4: Two glosses, ‘son’ and ‘daughter’, in [8] are displayed here as an example to compare the differences between cognates inside and cognates across meaning slots.

While a couple of strategies have been proposed to search for cognates across meaning slots [36, 37], none of the existing algorithms is sensitive to partial cognate relations as shown in Table 4. In order to address this problem in our workflow, we propose a novel approach that is relatively simple, but surprisingly efficient. We start from all *aligned cognate sets* in our data, and then systematically compare all alignments with each other. Whenever two alignments are *compatible*, i.e., they have (1) at least one morpheme in one language occurring in both aligned cognate sets, which is (2) identical, and (3) no shared morphemes in two alignments which are not identical, we treat them as belonging to one and the same cognate set (see Figure 7). We iterate over all alignments in the data algorithmically, merging the alignments into larger sets in a greedy fashion, and re-assigning cognate sets in the data.

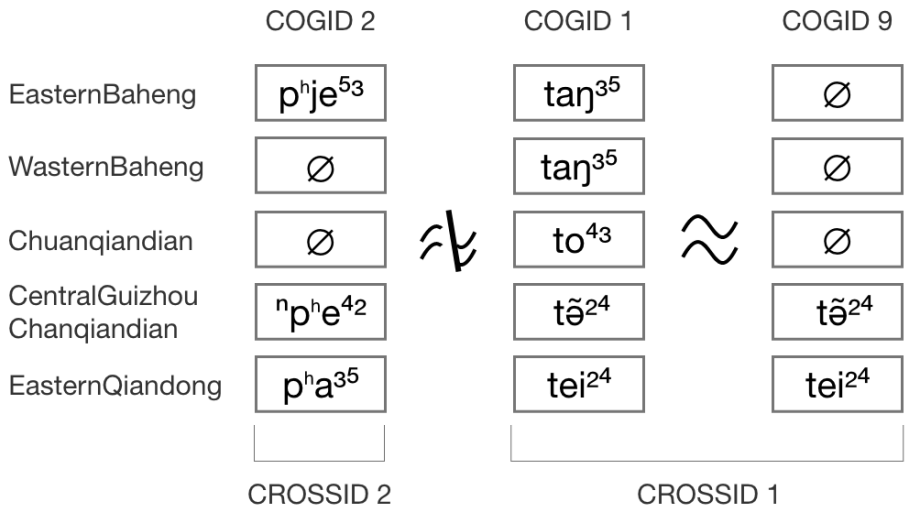


Figure 7: Compare alignments for morphemes meaning ‘‘son’ and ‘daughter’ as an example to illustrate how cross-semantic cognates can be identified. The cognate sets in which the forms in the languages are identical are clustered together and assigned a unique cross-semantic cognate identifier (CROSSID). Those which are not compatible as the cognate sets 2 and 1 in our example are left separate.

The results can be easily inspected with the help of the EDICTOR tool, for example, by inspecting cognate set distributions in the data, as illustrated in detail in the tutorial (TUTORIAL 3.4). When inspecting only those cognate sets which occur in at least 10 language varieties in our sample, we find already quite a few interesting cases of cross-semantic cognate sets: morphemes denoting the concept ‘one’, for example, recur in the words for ‘hundred’ (indicating that hundred is a compound of ‘one’ plus ‘hundred’ in all languages); morphemes recur in ‘snake’ and ‘earthworm’ (reflecting that words for ‘snake’ and ‘earthworm’ are composed of a morpheme ‘worm’); and ‘left’ and ‘right’ share a common morpheme (indicating an original meaning of ‘side’ for this part, such as ‘left side’ vs. ‘right side’).

3.2.5 From cross-semantic cognates to sound correspondence patterns

Sound correspondences, and specifically sound ***correspondence patterns*** across multiple languages, can be seen as the *core objective* of the classical comparative method and build the basis of further endeavors such as the reconstruction of proto-forms or the reconstruction of phylogenies. Linguists commonly propose *sound correspondence sets*, that is, collections of sound correspondences which reconstruct back to a common proto-sound (or sequence of

proto-sounds) in the ancestor language, as one of the final stages of historical language comparison. In Hmong-Mien languages, for example, Wang proposed 30 sets [38] and Ratliff reduced the quantity of correspondence sets to 28 [11].

	1	2	3	4	5	6	7	8	9	10	11
blood [*ntshjamX]	ɕhaŋ³	nʱtɕhi³	ɳtʂha³	ntsua^{3b}	nʔtshen^{B}	θi³	nʱe³	ɕam³	sa:m³	san³	dzjɛm³
head louse [*ntshjeiX]	ɕhu³	nʱtɕhi³	ɳtsau^{3b}	ntsɔ^{3b}	nʔtshu^{B}	-	tɕhi³	ɕeib³	tθei³	-	dzɛi³
to fear/be afraid [*ntshjeX]	ɕhi¹	-	ɳtʂai⁵	ntse^{5b}	nʔtshe^{C}	ɳtʃei¹	nʱɛ⁵	dʑa⁵	dʱa$^{5'}$	dʱa⁵	dzjɛ⁵
clear [*ntshjiəŋ]	ɕhi¹	-	ɳtʂia¹	ntsæin^{1b}	nʔtshe^{A}	-	nĩ¹	dzaŋ¹	-	-	-

Table 5: An example of correspondence sets in the classical literature, following Ratliff [11, p.75], reconstructed forms for Proto-Hmong-Mien are preceded by an asterisk.

An example for the representation of sound correspondence sets in the classical literature [11] is provided in Table 5. The supposed proto-sound **ntshj-* in proto-Hmong-Mien is inferred from the initials of four words in 11 contemporary Hmong-Mien languages.

Although this kind of data representation is typical for classical accounts on sound correspondence patterns in historical language comparison, it has several shortcomings. First, the representation shows only morphemes, and we are not informed about the full word forms underlying the patterns. This is unfortunate, since we cannot exclude that compound words were already present in the ancestral language, and it may likewise be possible that processes of compounding left traces in the correspondence patterns themselves. Second, since scholars tend to list sound correspondence patterns merely in an exemplary fashion, with no intent to provide full frequency accounts, it is often not clear how strong the actual evidence is, and whether the pattern at hand is exhaustive, or merely serves to provide an example. Third, we are not being told where a given sound in a given language fits a general pattern less well. Thus, we can find two different ***reflexes*** in language 8 in the table, [ɕ] and [dʑ], but without

further information, we cannot tell if the differences result from secondary, conditioned sound changes, or whether they reflect irregularities that the author has not yet resolved.

To overcome these shortcomings, we employ a two-fold strategy. We first make use of a new method for sound correspondence pattern detection [7] in order to identify exhaustively, for each column in each alignment of our data, to which correspondence pattern it belongs. In a second step, we use the EDICTOR tool to closely inspect the patterns identified by the algorithm and to compare them with those patterns proposed in the classical literature.

The method for correspondence pattern identification starts by assembling all ***alignment sites*** (all columns) in the aligned cognate sets of the data, and then clusters them into groups of compatible sound correspondence patterns. Compatibility essentially makes sure that no language has more than one reflex sound in all partitioned alignment sites (see [7] for a detailed explanation of this algorithm).

Table 6 provides some statistics regarding the results of the correspondence pattern analysis. The analysis yielded a total of 1392 distinct sound correspondence patterns (with none of the patterns being compatible with any of the other 1392 patterns). While this may seem a lot, we find that 234 patterns only occur once in the data only once (probably reflecting borrowing events, erroneously coded cognates, or errors in the data).[7] Among the non-singleton patterns, we find 302 corresponding to initials, 74 to medials, 389 to nucleus vowels, 95 to the codas, and 298 to the tone patterns. These numbers may seem surprising, but one should keep in mind that phonological reconstruction will assign several distinct correspondence patterns to the same proto-form and explain the divergence by means of conditioning context in sound change.[8] So far, there are few studies on the numbers of distinct correspondence patterns one should expect, but the results we find for the Hmong-Mien dataset are in line with previous

7 In cases of very intensive language contact, one would expect to find recurring correspondence patterns that include borrowings, but in the case of sporadic borrowings, they will surface as exceptions.

8 How this step of identifying conditioning context can be done in concrete is not yet entirely clear to us. Computational linguists often use *n-gram* representations in order to handle context of preceding and following sounds, but this would not allow us to handle situations of remote context.

studies on other language families [7]. More studies are needed in order to fully understand what one ought to expect in terms of the numbers of correspondence patterns in datasets of various sizes and types.

Position	**'Regular' Patterns**	**Singletons**
Initial	165	106
Medials	45	23
Nucleus	213	57
Coda	66	13
Tone	164	29
Total	653	228

Table 6: A summary of the result of the sound correspondence pattern inference algorithm applied to our data. The numbers below each item are the quantities of sound correspondence patterns detected at each position in the syllables.

Language	'blood'		'fear (be afraid)'	
Numao	ⁿtsʰ	a n ¹³	ⁿtsʰ	ei ³³
Western Luobuohe	ⁿtsʰ	e n ⁴⁴	ⁿtsʰ	e ³⁵
Biao Min	s	a n ³⁵	Ø	
Zao Min	ʑ	a m ²⁴	ʑ	a ⁴²
Younuo	tsʰ	u n ³³	tsʰ	i ⁴⁴
Western Xiangxi	ⁿtɕʰ	i ⁴⁴	ⁿtɕʰ	a ⁵³
Eastern Luobuohe	ⁿtsʰ	e n ⁴⁴	ⁿtsʰ	e ²⁴
Bana	Ø		dʑ	i ¹³
Eastern Xiangxi	tsʰ	i ⁵⁵	Ø	
Western Qiandong	ɕʰ	ẽ ¹³	ɕʰ	e ⁴⁴

Eastern Baheng	ⁿtɕʰ	e ³¹³	Ø	
Chuanqiandian	ⁿtʂʰ	a ŋ ⁵⁵	ⁿtʂʰ	ai ⁴⁴
Western Baheng	Ø		Ø	
Central Guizhou Chuanqiandian	ⁿsʰ	õ ¹³	ⁿsʰ	e ⁴²
Eastern Qiandong	ɕ	a n ³³	ɕ	a ²⁴

Table 7: Cells shaded in blue indicate the initial consonants belonging to a common correspondence pattern, with missing reflexes indicated by a Ø.

While the representation in textbooks usually breaks the unity of morphemes and word forms, our workflow never loses track of the words, although it enables users to look at the morphemes and at the correspondence patterns in isolation. Our accompanying tutorial shows not only how the correspondence patterns can be computed (TUTORIAL 3.5), but also how they can be inspected in the EDICTOR tool (TUTORIAL 3.5), where we can further see that our analysis uncovers the correspondence pattern shown in Table 5 above, as we illustrate in Table 7. Here, we can see that our approach confirms Ratliff's pattern by clustering initial consonants of cognates for 'blood' and 'fear (be afraid)' into one correspondence pattern.[9]

4 Discussion

Although our workflow represents what we consider the current state of the art in the field of computational historical linguistics, it is not complete yet, and it is also not perfect. Many more aspects need to be integrated, discussed, and formalized. Based on a quick discussion of the general results of our study, we will discuss three important aspects, namely, (a) the current performance of the existing algorithms in our workflow, (b) possible improvements of the algorithms, and (c) general challenges for all future endeavors in computer-assisted or

9 The other two cognate sets in Ratliff's data could not be confirmed, because they do not occur in our sample.

computational historical linguistics.

4.1 Current performance

Historical language comparison deals with the reconstruction of events that happened in the past and can rarely be directly verified. Our knowledge about a given language family is constantly evolving. At the same time, debate on language history is never free of disagreement among scholars, and this is also the case with the reconstruction of Hmong-Mien.[10] As a result, it is not easy to provide a direct evaluation of the performance of the computational part of the workflow presented here.

In addition to these theoretical problems, evaluation faces practical problems. First, classical resources on historical language comparison of Hmong-Mien are not available in digital form (and digitizing them would be beyond the scope of this study). Second, and more importantly, however, even when having recent data on Hmong-Mien reconstruction in digital form we could not compare them directly with our results due to the difference in the workflows. All current studies merely consist of morphemes which were taken from different sources without giving reference to the original words [31]. Full words, which are the starting point in our study, are not reported and apparently not taken into account. For a true evaluation of our workflow, however, we would need a manually annotated dataset that would show the same completeness in terms of annotation as the one we have automatically produced. Furthermore, since our workflow is explicitly thought of as a computer-assisted, not a purely computational workflow, the question of algorithmic performance is rather aesthetical than substantial, given that the computational approaches are merely used to ease the labor of the experts.

Nevertheless, to some degree, we can evaluate the algorithms which we assembled for our workflow here, and it is from these evaluations that have been made in the past, that we draw confidence in the overall usefulness of our workflow. Partial cognate detection, as outlined

10 Compare, for example, the debate about regular epenthesis in Proto-Hmong-Mien among Ratliff [42] and Ostapirat [43].

in Section 3.2, for example, has been substantially evaluated with results ranging between 90% (Chinese dialects) and 94% (Bai dialects) compared to expert judgments. The alignment procedure we propose is supposed to work as good as an expert, provided that experts agree on the prosodic structure we assign to all morphemes. For the cross-semantic cognate set detection procedure we propose, we do not yet have substantial evaluations, since we lack sufficient test data. The correspondence pattern detection algorithm, finally, has been indirectly evaluated, by testing how well so far unobserved cognate words could be predicted (see also [39]), showing an accuracy between 59% (Burmish languages) and 81% (Polynesian languages) for trials in which 25% of the data was artificially deleted and later predicted.

As another quick way to check if the automated aspects of our workflow are going into the right direction, we can compute a phylogeny based on shared cross-semantic cognates between all language pairs and see if the phylogeny matches with those proposed in the literature. This analysis, which can be inspected in detail in the accompanying tutorial (TUTORIAL 4.2), shows that the automated workflow yields a tree that correctly separates not only Hmongic from Mienic languages but also identifies all smaller subgroups commonly recognized.

4.2 Possible improvements

The major desideratum in terms of possible improvements is the inclusion of further integration of our preliminary attempts for *semi-automated reconstruction*, starting from already identified sound correspondence patterns. Experiments are ongoing in this regard, but we have not yet had time to integrate them fully.[11] In general, our workflow also needs a clearer integration of automatic and manual approaches, ideally accompanied by extensive tutorials that would allow users to start with the tools independently. This study can be seen as a first step in this

11 A specific problem in semi-automated reconstruction consists in the importance of handling conditioning context in sound change. To our knowledge, no approaches that would sufficiently deal with this problem have been proposed so far. This reflects one apparent problem of common alignment approaches, as they cannot handle cases of *structural equivalence* which require information on conditioning context [44].

direction, but much more work will be needed in the future.

4.3 General challenges

General challenges include the full-fledged *lexical reconstruction of words*, i.e., a reconstruction that would potentially also provide compounds in etymological dictionaries. This might help to overcome a huge problem in historical language comparison in the Southeast Asian area, where scholars tend to reconstruct only morphemes, and rarely attempt at the reconstruction of real word forms in the ancestral languages [31]. Furthermore, we will need a convincing annotation of sound change that would ideally allow us to even check which sounds changed at which time during language history.

5 Outlook

This article provides a detailed account on what we consider the current state-of-the-art in computer-assisted language comparison. Starting from raw data, we have shown how these can be successively lifted to higher levels of annotation. While our five-step workflow is intended to be applied in a computer-assisted fashion, we have shown that even with a purely automatic approach, one can already achieve insightful results that compare favourably to results obtained in a purely manual approach. In the future, we hope to further enhance the workflow and make it more accessible to a wider audience.

6 Acknowledgements

This research was funded by the ERC Starting Grant 715618 “Computer-Assisted Language Comparison” (CALC, http://calc.digling.org, MSW, NES, JML), the ERC Synergy Grant 609823 “Beyond Boundaries: Religion, Region, Language and the State” (ASIA, NWH), and the Grant of P2BEP1_181779 “Reconstruction of Proto-Western Kho-Bwa” of the Swiss National Science Foundation (TAB). The workflow was presented in the workshop “Recent Advances in

Comparative Linguistic Reconstruction" in SOAS, London. We thank the workshop participants for giving valuable feedback regarding several aspects of the workflow in their studies. In addition, we thank Christoph Rzymski and Tiago Tresoldi who provided technical support on setting up our Code Ocean capsule.

7 References

1. List J-M. Computer-Assisted Language Comparison: Reconciling Computational and Classical Approaches in Historical Linguistics [Internet]. *Jena: Max Planck Institute for the Science of Human History*. 2016. Available from: https://hcommons.org/deposits/item/hc:25045/

2. List, J-M, Greenhill, SJ, Gray, RD. The potential of automatic word comparison for historical linguistics. *PLOS ONE*. 2017;12(1):1–18.

3. Dellert J, Daneyko T, Münch A, Ladygina A, Buch A, Clarius N, et al. NorthEuraLex: A wide-coverage lexical database of northern eurasia. *Language Resources and Evaluation*. 2020;54(1):273–301.

4. Sagart L, Jacques G, Lai Y, Ryder R, Thouzeau V, Greenhill SJ, et al. Dated language phylogenies shed light on the ancestry of Sino-Tibetan. *Proceedings of the National Academy of Science of the United States of America [Internet]*. 2019;116(21):10317–22. Available from: https://www.pnas.org/content/early/2019/04/30/1817972116

5. Kolipakam V, Jordan FM, Dunn M, Greenhill SJ, Bouckaert R, Gray RD, et al. A Bayesian phylogenetic study of the Dravidian language family. *Royal Society Open Science*. 2018;5(171504):1–17.

6. Barrachina S, Bender O, Casacuberta F, Civera J, Cubel E, Khadivi S, et al. Statistical approaches to computer-assisted translation. *Computational Linguistics*. 2008;35(1):3–28.

7. List J-M. Automatic inference of sound correspondence patterns across multiple languages. *Computational Linguistics [Internet].* 2019;1(45):137–61. Available from: https://www.mitpressjournals.org/doi/full/10.1162/coli_a_00344

8. Chén, Q 陈其光 [Chen Q]. Miàoyáo yǔwén 苗瑶语文 [Mao and Yao Language]. *Běijīng* 北京*: Zhōngyāng Mínzú Dàxué* 中央民族大学出版社 *[Central Institute of Minorities]*; 2012 [cited 2019 Feb 23]. Available from: https://en.wiktionary.org/wiki/Appendix:Hmong-Mien_comparative_vocabulary_list

9. Cooper D. Data Warehouse, Bronze, Gold, STEC, Software. In: *Proceedings of the 2014 Workshop on the Use of Computational Methods in the Study of Endangered Languages*. 2014. pp. 91–9.

10. Rzymski C, Tresoldi T, Greenhill S, Wu M-S, Schweikhard NE, Koptjevskaja-Tamm M, et al. The Database of Cross-Linguistic Colexifications, reproducible analysis of cross- linguistic polysemies. *Scientific Data [Internet]*. 2020;7(13):1–12. Available from: https://www.nature.com/articles/s41597-019-0341-x

11. Ratliff M. Hmong-Mien language history. *Canberra: Pacific Linguistics*; 2010.

12. Swadesh M. Lexico-statistic dating of prehistoric ethnic contacts: With special book to north american indians and eskimos. *Proceedings of the American Philosophical Society [Internet]*. 1952;96(4):452–63. Available from: http://www.jstor.org/stable/3143802

13. Swadesh M. Towards greater accuracy in lexicostatistic dating. *International Journal of American Linguistics [Internet]*. 1955;21(2):121–37. Available from: http://www.jstor.org/stable/1263939

14. Comrie B, Smith N. Lingua Descriptive Series: Questionnaire. *Lingua*. 1977;42:1–72.

15. Liú Lìlǐ 刘俐李, Wáng Hóngzhōng 王洪钟, Bǎi Yíng 柏莹. Xiàndài Hànyǔ fāngyán héxīncí, tèzhēng cíjí 现代汉语方言核心词·特征词集 [Collection of basic vocabulary words and characteristic dialect words in modern Chinese dialects]. *Nánjīng* 南京*: Fènghuáng* 凤凰; 2007.

16. So-Hartmann H. Notes on the Southern Chin languages. *Linguistics of the Tibeto-Burman Area*. 1988;11(2):98–119.

17. Matisoff JA. Variational semantics in Tibeto-Burman. The “organic” approach to linguistic comparison. *Institute for the Study of Human Issues*; 1978.

18. Blust R. Variation in retention rate among Austronesian languages. *Unpublished paper presented at the Third International Conference on Austronesian Linguistics*, Bali, January 1981; 1981.

19. Běijīng Dàxué 北京大学北京大学 [BD] (ed.). Hànyǔ fāngyán cíhuì 汉语方言词汇 [Chinese dialect vocabularies]. *Běijing* 北京*: Wénzì Gǎigé* 文字改革; 1964.

20. Baayen RH, Piepenbrock R, Gulikers L, editors. *The CELEX Lexical Database*. Philadelphia: University of Pennsylvania; Linguistic Data Consortium; CD-ROM; 1995.

21. PONS.Eu Online-Wörterbuch. *Stuttgart: Pons GmbH*; [Accessed 2019 October 24].

22. Moran S, Cysouw M. The Unicode Cookbook for Linguists: Managing writing systems using

orthography profiles [Internet]. *Berlin: Language Science Press*; 2018. Available from: http://langsci-press.org/catalog/book/176

23. Wickham H, others. Tidy data. *Journal of Statistical Book.* 2014;59(10):1–23.

24. Forkel R, List J-M, Greenhill SJ, Rzymski C, Bank S, Cysouw M, et al. Cross-Linguistic Data Formats, advancing data sharing and re-use in comparative linguistics. *Scientific Data [Internet]*. 2018;5(180205):1–10. Available from: https://www.nature.com/articles/sdata2018205

25. Broman KW, Woo KH. Data organization in spreadsheets. *The American Statistician.* 2018;72(1):2–10.

26. Hammarström H, Haspelmath M, Forkel R. *Glottolog. Version 4.0*. Jena: Max Planck Institute for the Science of Human History; 2019. Available from: https://glottolog.org

27. List JM, Rzymski C, Greenhill S, Schweikhard N, Pianykh K, Tjuka A, et al. Concepticon. A resource for the linking of concept lists (Version 2.3.0) [Internet]. *Jena: Max Planck Institute for the Science of Human History*; 2020. Available from: https://concepticon.clld.org/

28. List J-M, Anderson C, Tresoldi T, Rzymski C, Greenhill S, Forkel R. Cross-Linguistic Transcription Systems. Version 1.3.0. *Jena: Max Planck Institute for the Science of Human History*; 2019.

29. List J-M. Beyond cognacy: Historical relations between words and their implication for phylogenetic reconstruction. *Journal of Language Evolution [Internet]*. 2016;1(2):119–36. Available from: http://jole.oxfordjournals.org/content/1/2/119

30. Matisoff JA. On the uselessness of glottochronology for the subgrouping of Tibeto-Burman. In: Renfrew C, McMahon A, Trask L, editors. Time depth in historical linguistics. *Cambridge: McDonald Institute for Archaeological Research*; 2000. pp. 333–71.

31. Hill NW, List J-M. Challenges of annotation and analysis in computer-assisted language comparison: A case study on Burmish languages. *Yearbook of the Poznań Linguistic Meeting [Internet]*. 2017;3(1):47–76. Available from: https://www.degruyter.com/view/j/yplm.2017.3.issue-1/yplm-2017-0003/yplm-2017-0003.xml

32. List J-M, Lopez P, Bapteste E. Using sequence similarity networks to identify partial cognates in multilingual wordlists. In: *Proceedings of the Association of Computational Linguistics 2016 (Volume 2: Short Papers) [Internet].* Berlin: Association of Computational Linguistics; 2016. pp. 599–605. Available from: http://anthology.aclweb.org/P16-2097

33. List J-M. 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 [Internet]. Valencia: Association for Computational Linguistics; 2017. pp. 9–12. Available from: http://edictor.digling.org

34. List J-M, Walworth M, Greenhill SJ, Tresoldi T, Forkel R. Sequence comparison in computational historical linguistics. *Journal of Language Evolution [Internet]*. 2018;3(2):130–44. Available from: https://academic.oup.com/jole/article/3/2/130/5050100?guestAccessKey=cf8fe64e-3996-4cb1-ba2c-317a7cd81bf4

35. Wang WS-Y. Linguistic diversity and language relationships. In: *Huang C-tJ, editor*. New horizons in Chinese linguistics. Dordrecht: Kluwer; 1996. pp. 235–67. (Studies in natural language and linguistic theory).

36. Arnaud AS, Beck D, Kondrak G. Identifying cognate sets across dictionaries of related languages. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics; 2017. pp. 2509–18.

37. Wahle J. An approach to cross-concept cognacy identification. In: Bentz C, Jäger G, Yanovich I, editors. *Proceedings of the Leiden Workshop on Capturing Phylogenetic Algorithms for Linguistics* [Internet]. Tübingen: Eberhard-Karls University; 2016. Available from: http://dx.doi.org/10.15496/publikation-10060

38. Wang F 王辅世. Miáoyǔ gǔyīn gòunǐ 苗语古音构拟 [reconstruction of the sound system of proto-miao]. *Tokayo: Institute for the Study of languages; Cultures of Asia; Africa*; 1994.

39. Bodt TA, List J-M. Testing the predictive strength of the comparative method: An ongoing experiment on unattested words in Western Kho-Bwa languages. *Papers in Historical Phonology [Internet]*. 2019;4(1):22–44. Available from: http://journals.ed.ac.uk/pihph/article/view/3037

40. Hoenigswald HM. Phonetic similarity in internal reconstruction. *Language [Internet]*. 1960;36(2):191–2. Available from: http://www.jstor.org/stable/410982

41. Kay M. The logic of cognate recognition in historical linguistics. *Santa Monica: The RAND Corporation*; 1964.

42. Ratliff M. Against a regular epenthesis rule for hmong-mien. *Papers in Historical Phonology.* 2018 Dec;3.

43. Ostapirat W. Issues in the reconstruction and affiliation of proto-miao-yao. *Language and Linguistics.* 2016;17(1):133–45.

44. List J-M. Beyond Edit Distances: Comparing linguistic reconstruction systems. *Theoretical Linguistics [Internet]*. 2019;45(3-4):1–10. Available from:

https://www.degruyter.com/view/j/thli.2019.45.issue-3-4/tl-2019-0016/tl-2019-0016.xml?format=INT

Supplementary information and material

The appendix that is submitted along with this study consists of two parts. First, there is a glossary explaining the most important terms that were used throughout this study. Second, there is a tutorial explaining the steps of the workflow in detail. Additionally to this supplementary information, we provide supplementary material in the form of data and code. The data which is used in this study is archived on Zenodo (DOI: 10.5281/zenodo.3741500) and curated on GitHub (Version 2.1.0, https://github.com/lexibank/chenhmongmien). The code along with the tutorial has also been archived on Zenodo (DOI: 10.5281/zenodo.3741771) and is curated on GitHub (Version 1.0.0, https://github.com/lingpy/workflow-paper). Additionally, our Code Ocean Capsule allows users to run the code without installing anything on their machine, it can be accessed from https://codeocean.com/capsule/8178287/ (Version 2).

Author contributions

MSW, NWH, and JML initiated the study. MSW, NWH, JML, and TAB drafted the workflow. MSW and JML implemented the workflow. NES wrote the glossary. TAB, NWH and NES tested the workflow on different datasets. MSW and JML wrote the accompanying tutorial. MSW and JML wrote the first manuscript. NES, NWH and TAB helped in revising the manuscript. All authors agree with the final version of the manuscript.

Appendix A: Glossary

Alignment

An alignment is a comparison of two or more sequences which places the sequences in a matrix, indicating corresponding segments by placing them in the same row, with missing segments being represented by a gap symbol (usually a '-').

Alignment site

A column of an alignment (term adopted from molecular biology). See phonetic alignment analysis.

Basic vocabulary

Referring to concepts that are assumed to occur in all human languages and to be more resistant to replacement than other parts of the vocabulary.

Cognacy

A relation between two word forms. The relation holds when the two words go back to a common ancestor. Words that share this relationship are called cognate or etymologically related. When talking about "cognates", this usually excludes those words related by borrowing events.

Colexification

Two different concepts that are expressed by the same word are said to colexify.

Compound words

Words that are formed by combining two other words, like *correspondence pattern*

being composed from the words *correspondence* and *pattern*.

Correspondence patterns

Also sound correspondence patterns. Due to the regularity of sound change (see below), words that share the same sound in one language often also share the same (possibly different) sound in another language if they are cognate with the respective words in that language. These regularities are called correspondence patterns.

Cross-semantic cognates

Cognate words that have a different meaning due to semantic change.

Morpheme

Smallest part of a word that corresponds to a meaning or function of its own, usually by occuring in other words as well. It differs from a phonestheme in so far as all morphemes of a word taken together build the whole word but the parts of a word not belonging to a given phonestheme consist not necessarily of morphemes or phonesthemes themselves.

Orthography profile

A replacement table used to automatically convert data from one transcription system into another (e.g. into IPA) and to segment it into units (e.g. phonemes, diphthongs,...).

Phonetic alignment analysis

The comparison of sequences, e.g. of words suspected to be related. The words are therefore put into a matrix in such a way that corresponding segments appear in the same column, while placeholder symbols are used to represent those cases where a

corresponding segment is lacking.

Strict cognates

Related words that differ only by regular sound change. This means that they go back to exactly the same word form and that no borrowing event was involved in the history of these words since their common ancestor.

Reflex

The descendant of a given ancestral form. Reflex typically refers to a word form, but one can also find the term reflex sound in the literature.

Appendix B: Tutorial

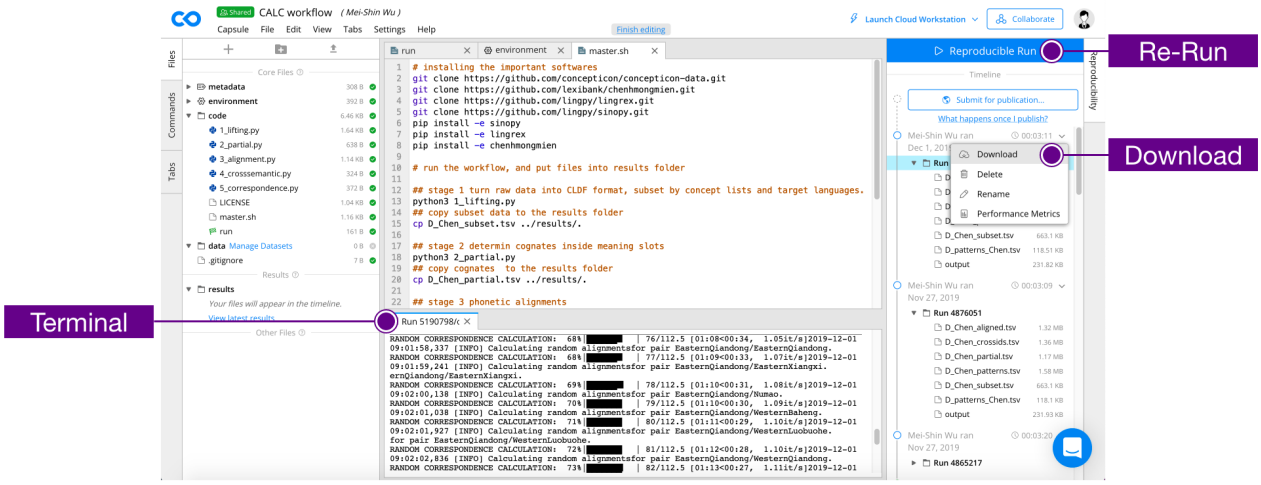
This tutorial supplements the study "Computer-Assisted Language Comparison: State of the Art". In this tutorial, we explain in detail, how our workflow can be tested and applied. The workflow consists of several Python libraries that interact, one producing the data that can be used by the other. Since the data is available in different stages, each stage allows us to intervene by correcting errors manually that were made by the automated approach.
For users who are interested in testing our workflow on their local machine or further applying it in their own research, some basic knowledge of the Python programming language and the commandline will be required. All the software offered here is available in the form of free software. For more information on LingPy, the main programming library used here, we recommend users to check the tutorial[12] accompanying the study “Sequence comparison in computational historical linguistics”[13] by List et al. (2018)[1].

1. Code Ocean Capsule

In order to facilitate it for users to quickly test our workflows without installing the software, we have set up a Code Ocean Capsule which users can use to run the code remotely. Code Ocean is an open access platform which enables researchers to reproduce their or others’ experiments. For a detailed introduction to the Code Ocean platform[14], please refer to the website. To see how our experiments can be run from within the Code Ocean Capsule, follow the following steps:

a) Navigate to the capsule: https://codeocean.com/capsule/8178287/tree/v2
b) Press the **Re-Run** button to reproduce the results.
c) View the progression in the **Terminal** panel.
d) Download all results and unzip the .zip file for further inspection on EDICTOR.

12 https://github.com/lingpy/lingpy-tutorial
13 https://academic.oup.com/jole/article/3/2/130/5050100
14 https://codeocean.com/



The following files can be found in the downloaded file:

File	Stage	Section
D_Chen_subset.tsv	From raw data to tokenized data	3.1
D_Chen_partial.tsv	From Tokenized Data to Cognate Sets	3.2
D_Chen_aligned.tsv	From Cognate Sets to Alignments	3.3
D_Chen_crossids.tsv	From Alignments to Cross-Semantic Cognates	3.4
D_Chen_patterns.tsv	From Cross-Semantic Cognates to Sound Correspondence	3.5
D_Chen_distance.dst	Validation	4.2, 4.3
D_Chen_tree.tre	Validation	4.2, 4.3

2. Installation Instructions

We assume that users who are interested in running the workflow on their local machine are familiar with the essentials of command-line operations and system administration on either Unix-like systems (such as Linux and MacOS) or Windows systems. Also, users should have Python[15] installed, including the package manager `pip`. Additionally, the version control system[16] `git` will be required. We strongly encourage users to run this code in a virtual environment. A virtual environment is a practical solution for creating independent configurations for testing and

15 https://www.python.org/, Version 3.5 or higher
16 https://git-scm.com/

experimenting, with no interference on the system-wide installation and without requiring complex virtualization or containerization solutions. The Python Packaging User Guide[17] gives clear instructions on setting up a virtual environment on Windows, Linux and macOS.
We start by installing the dependencies from the commandline. In order to do so, we first download the code that we will use with help of `git`.

```
$ git clone https://github.com/lingpy/workflow-paper.git
$ cd workflow-paper
```

Now that we have done this, we can install all the packages we will need with help of `pip`.

```
$ pip install -r requirements.txt
```

Now that this has been done, we need to configure the access to reference catalogs, such as Concepticon[18] and CLTS[19] in order to make sure that they can be accessed readily by the code. This can be done with help of the `catconfig` argument submitted with the `cldfbench` package which organizes the linguistic datasets.

```
$ cldfbench catconfig
```

You will be prompted to ask if you want to clone actual versions of Concepticon, Glottolog, and CLTS, and the easiest way to deal with this is to agree and type “y” in all cases.

3. Getting Started

There are two basic ways in which you can run our workflow:

1. You can run it by downloading a set of Python scripts and running them directly on your computer.
2. You can use the `cldfbench` package to run the commands via the commandline, without downloading the data directly.

The advantage of solution 2 is that you do not have to download extra data, since we have integrated the code directly in the `lexibank` version of the dataset of Hmong Mien languages by Chén (2012)[2]. Once this dataset has been installed (and this is the first package we have

17 https://packaging.python.org/guides/installing-using-pip-and-virtual-environments/
18 https://github.com/concepticon/concepticon-data
19 https://github.com/cldf-clts/clts/

installed in the previous section as part of all dependencies needed), you can type commands on your commandline, and the code will be carried out. The disadvantage is that the code example itself is not that easy to process for people less experienced with Python. For this reason, we will only note the commands in each of the steps we discuss in the following, and not explain them in more detail.

3.1 From Raw Data to Tokenized Data

The first script essentially loads the data from the repository and creates a wordlist that contains a subselection of all the data that was used. Some aspects of the more difficult "lifting" of data have already been done and distributed along with the original data package[20], which specifically also contains the orthography profile in the file `etc/orthography.tsv` and can be automatically applied with help of the `cldfbench` package.

```
$ cldfbench lexibank.makecldf chenhmongmien
```

But since the data is available in the form of a `cldf` package with the original orthography already tokenized to the formats we need, you can also skip this step and convert the data to the wordlist format required by the `lingpy` package.

```
$  python 1_select.py
```

If you want to test the version from the CLDF-repository directly with `cldfbench`, you can type:

```
$  cldfbench chenhmongmien.wf_select.
```

This will select a part of the languages and a part of the concepts, as indicated in the main study and write them to a file `D_Chen_subsets.tsv`. Additionally, you will see some statistics on the terminal, specifically a table indicating the coverage for each language. If you want to select all languages, and not just a subset, type:

```
$  python 1_select.py all
```

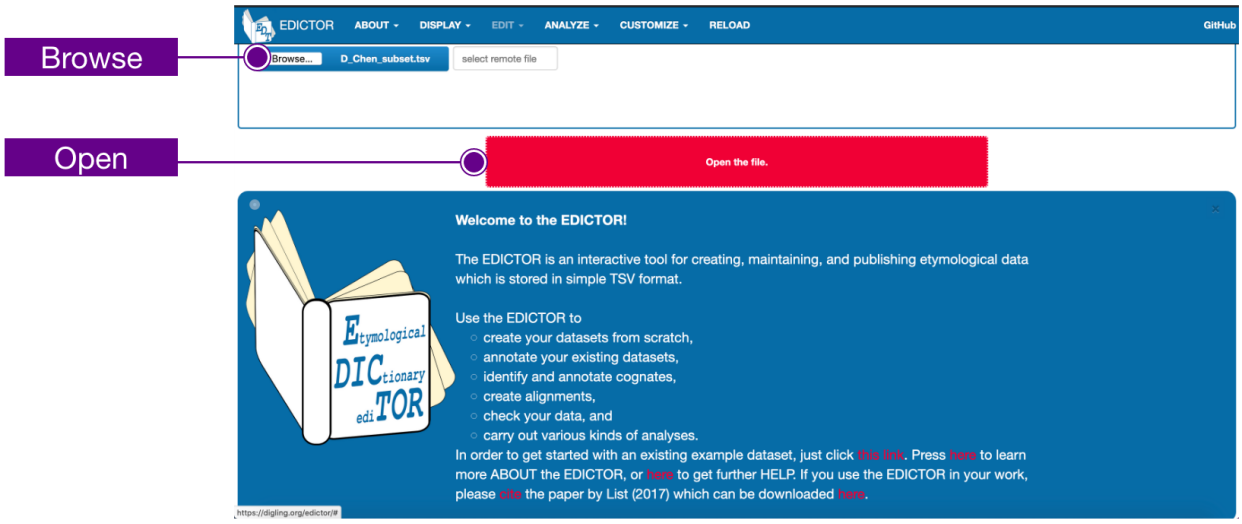
The output `A_Chen_subset.tsv` is generated due to the argument all is used. Once the argument `all` is used in the first stage, it has to be added to the rest of stages to ensure that the workflows process the correct files.

20 https://github.com/lexibank/chenhmongmien

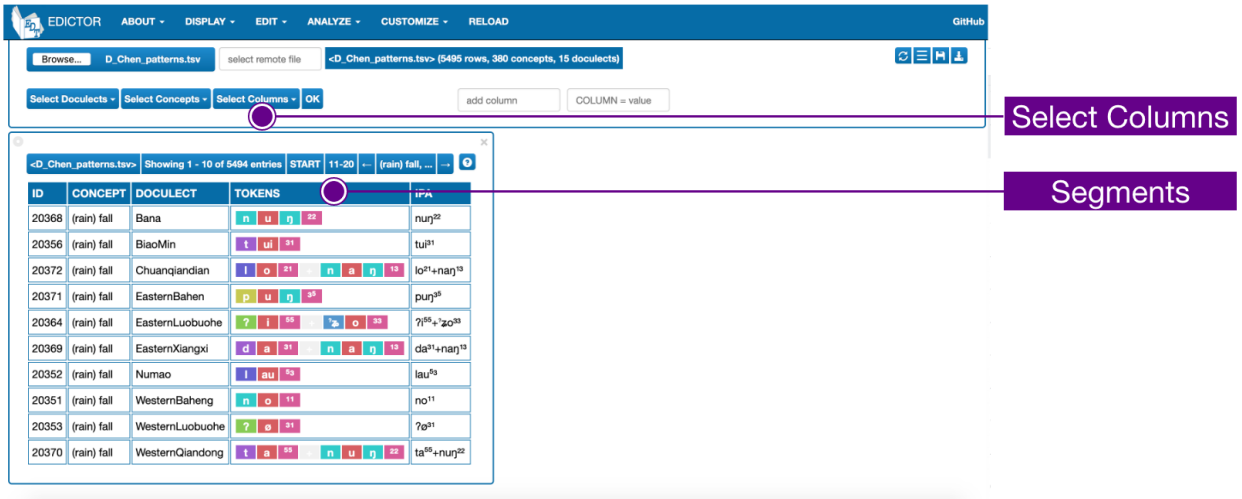
Doculect	Words	Coverage
Bana	502	1.00
BiaoMin	488	0.97
CentralGuizhouChuanqiandian	454	0.90
Chuanqiandian	501	1.00
EasternBahen	492	0.98
EasternLuobuohe	499	0.99
EasternQiandong	442	0.88
EasternXiangxi	492	0.98
Numao	490	0.98
WesternBaheng	500	1.00
WesternLuobuohe	488	0.97
WesternQiandong	494	0.98
WesternXiangxi	502	1.00
Younuo	500	1.00
ZaoMin	455	0.91

Already now you can inspect the data with the help of the EDICTOR tool. In order to do so, open the tool's website at https://digling.org/edictor/ and wait until the page is loaded (note that we recommend to browse EDICTOR in Firefox, but GoogleChrome should also not cause further problems).

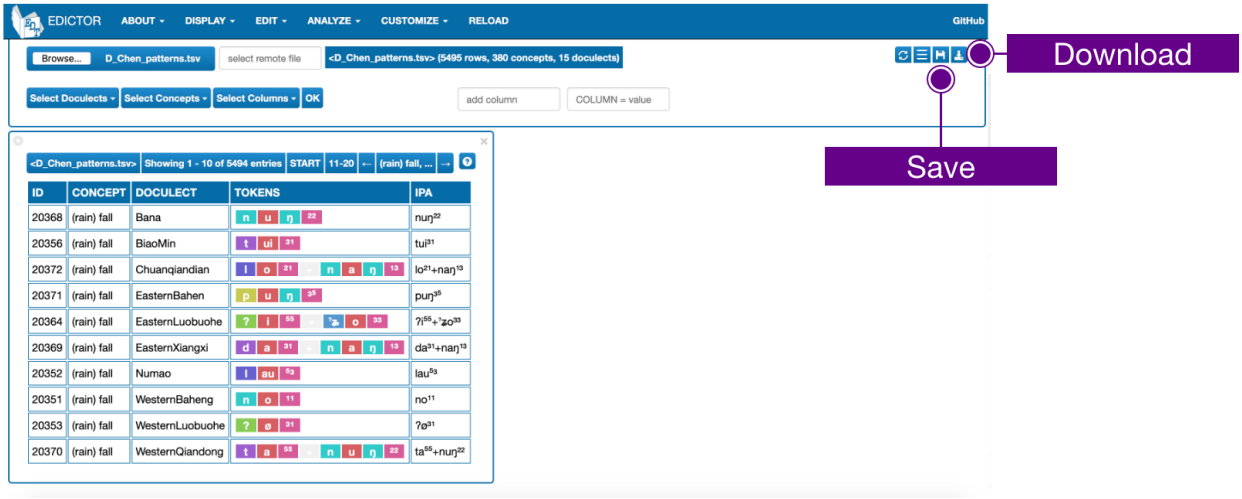
The data is in the file D_Chen_subset.tsv, in order to load it to the tool, press the **Browse** button and select the file. Once this has been done, press the **Open the file** button to examine the data, as illustrated in the following figure.



The segmented strings are displayed in the TOKENS column. Press **Select Columns** to inspect the raw forms and other aspects of the data, as shown in the following figure.



In order to save data to your computer, after you have manually edited them, you need to “download” them. This may be a bit surprising, since effectively, you do not download the data, but since the EDICTOR is working on a browser, it does not have any access to the data on your computer, and **download** is the only way to communicate with your machine. Thus, in order to save your data and load it to your machine, you first have to press the **save** icon at the top-right corner in order to to store the edited data in the web browser. When now pressing the **download** icon at the top-right, your browser will either directly download the data and store them in your download folder, or it will ask you to specify a specific file destination.



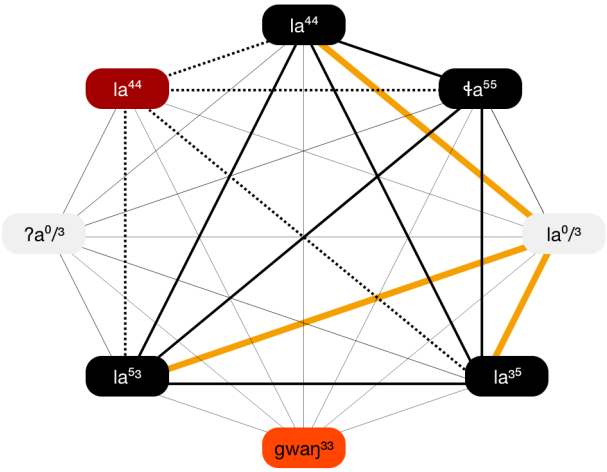
Be careful when editing data in the EDICTOR without saving and downloading them. If you close your browser, all the edits you made will be lost, so you should regularly save and download your data when working with the EDICTOR. As a shortcut, you can also type CONTROL+S to save and CONTROL+E to “export” the data (i.e., to download them).

3.2 From Tokenized Data to Cognate Sets

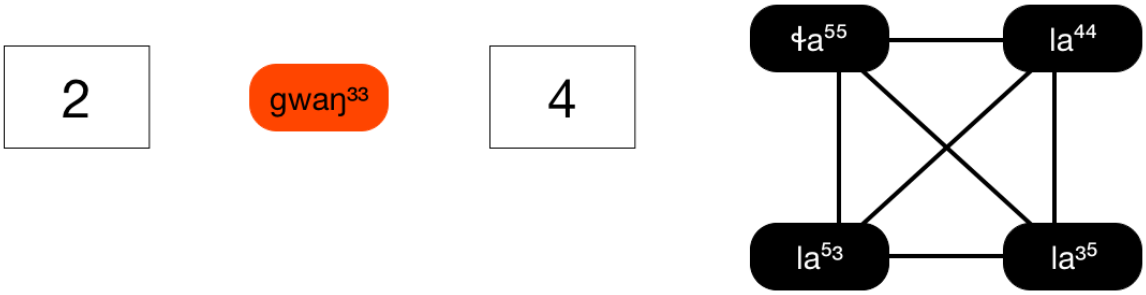
Partial cognate detection is an important task, specifically when working with Southeast Asian language data. The algorithm we use for this taks was first proposed in the study “Using Sequence Similarity Networks to Identify Partial Cognates in Multilingual Wordlists” by List et al. (2016)[3], where the algorithm is described in due detail. To illustrate how the algorithm works, we provide an example with four words for ‘moon’ in the Eastern Baheng, Eastern Qiandong, Bana and Biao Min language varieties. The major steps of the algorithm are the following:

c) Calculate the distances of all morpheme pairs.
d) Create a fully connected network from the distance scores.
e) Filter the network by deleting edges in the following fashion:
   A. Two morphemes in the same word should not be linked (see the dashed lines in the following figure).
   B. A morpheme in a word should not be linked to two morphemes in another word (see the yellow edges in the figure).
f) Remove the edges with similarity scores below a given threshold.

DOCULECT	IPA	
EasternBaheng	la$^{0/3}$	ɬa^{55}
EasternQiandong	la^{44}	la^{44}
Bana	la$^{0/4}$	la^{35}
BiaoMin	la^{53}	gwaŋ^{33}



Once this has been done, an algorithm for Community Detection in networks[4] is used to partition the network into “communities”, with each community representing one partial cognate set.



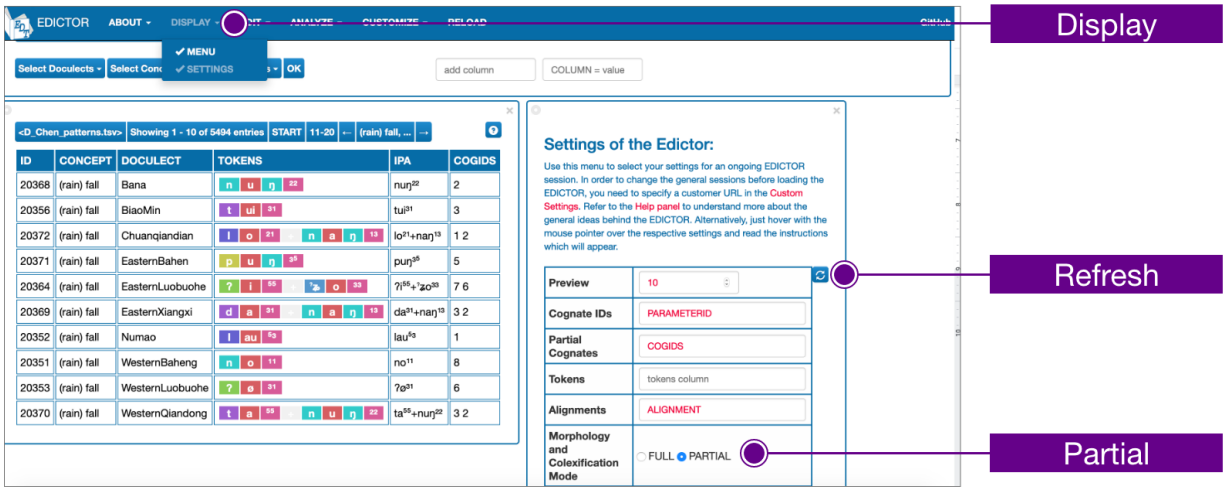
In order to calculate partial cognates, we use the algorithm as provided by the `lingpy` software package and apply it to our subselection of languages.

```
$ python 2_partial.py
```

If you want to test the version from the CLDF-repository directly with `cldfbench`, you can type:

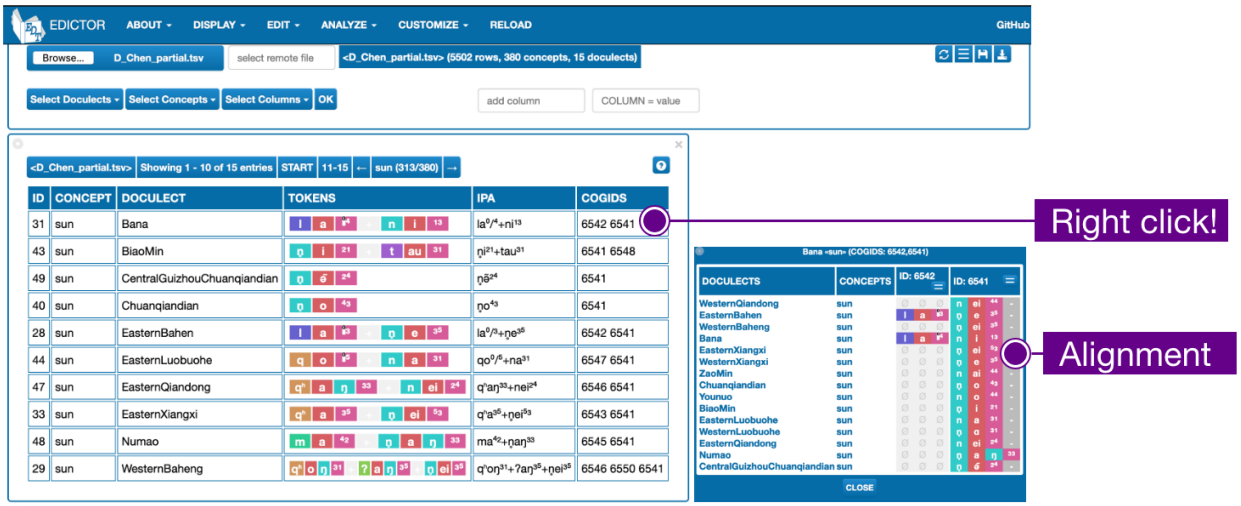
```
$ cldfbench chenhmongmien.wf_partial.
```

This will take some time when you run it the first time. The data can be found in the file `D_Chen_partial.tsv`. To inspect the data with EDICTOR, load `D_Chen_partial.tsv` as shown before. Then press **DISPLAY** to select **SETTINGS** in the drop-down menu. Select **PARTIAL** in the **Morphology and Colexification Mode** entry. Press the **Refresh** button.



In order to investigate the partial cognates, you need to select the column which stores the identifiers. To do so, press **Select Columns** and select **COGIDS** in the drop-down menu.

If you right-click on any number in the “COGIDS” column, a pop-up window will open and show all the cognate sets for a given word form in the form of an alignment. Since we have not yet aligned the data, the alignment will be wrong at this point.



3.3 From Cognate Sets to Alignments

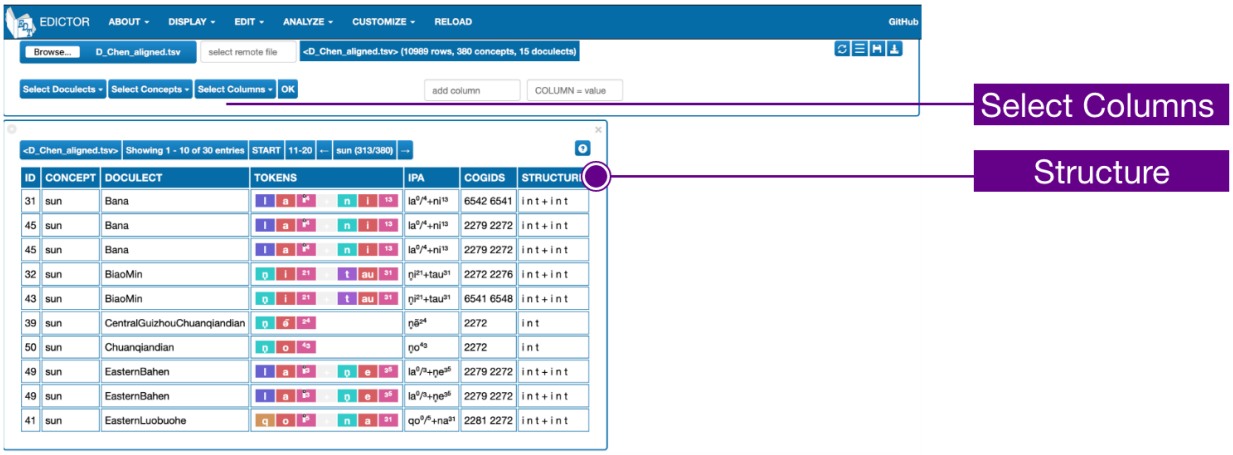
To align the data, we use the new procedure for template-based alignment, which is available from the `lingrex` package which we have installed as one of the requirements of our workflow, and the `sinopy` package, which helps us to compute syllable templates from all morphemes in the data. Running the code is again straightforward.

```
$ python 3_alignment.py
```

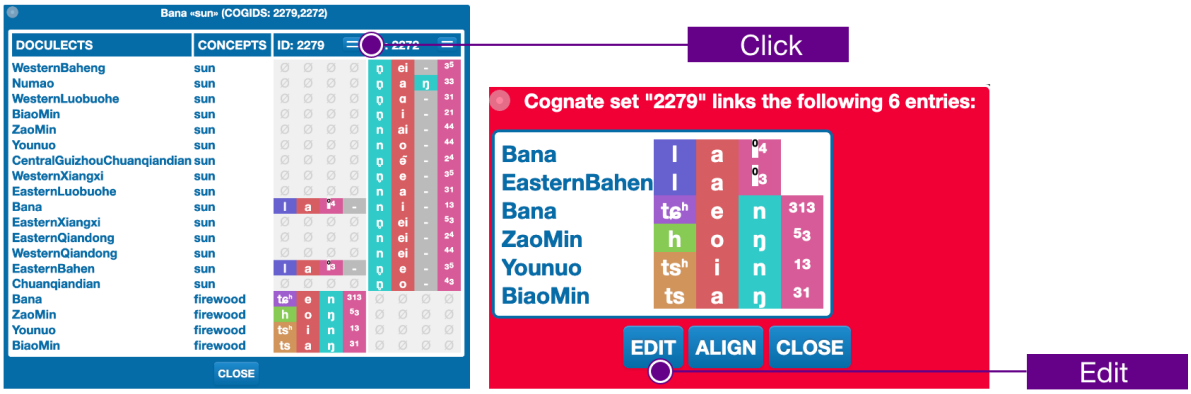
If you want to test the version from the CLDF-repository directly with `cldfbench`, you can type:

```
$ cldfbench chenhmongmien.wf_alignment
```

The aligned data will be stored in the file `D_Chen_aligned.tsv`. To inspect the alignments in EDICTOR, load this file and follow the previous steps we mentioned in Section 3.2. In addition to selecting the **COGIDS** column now, we also select the **STRUCTURE** column, since this column provides the templates for each morpheme, which we have automatically added to the data with help of `sinopy`.



As we already mentioned, if you right-click on any number in the “COGIDS” column, a pop-up window will show the alignment. Click on the `=` sign to modify the alignment. The modification itself is very straightforward: just click on a sound segment to move it to the right, and click on a gap segment to delete this segment.



3.4 From Alignments to Cross-Semantic Cognates

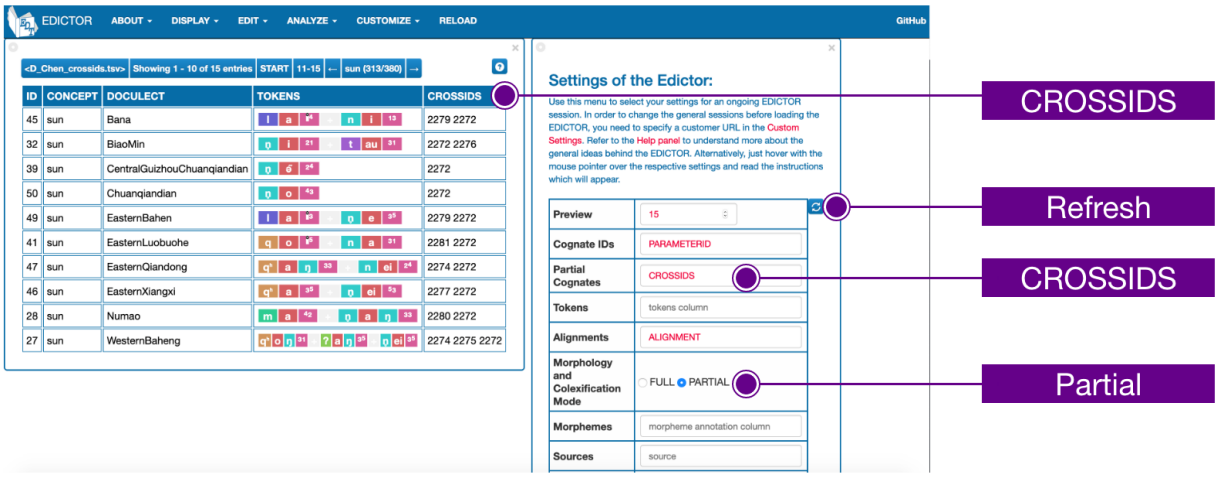
The algorithm for cross-semantic cognate detection as we propose it here is illustrated in more detail in the main study. It is implemented as part of the `lingrex` package. Again, it is straightforward to run the code.

```
$ python 4_crosssemantic.py
```

If you want to test the version from the CLDF-repository directly with `cldfbench`, you can type:

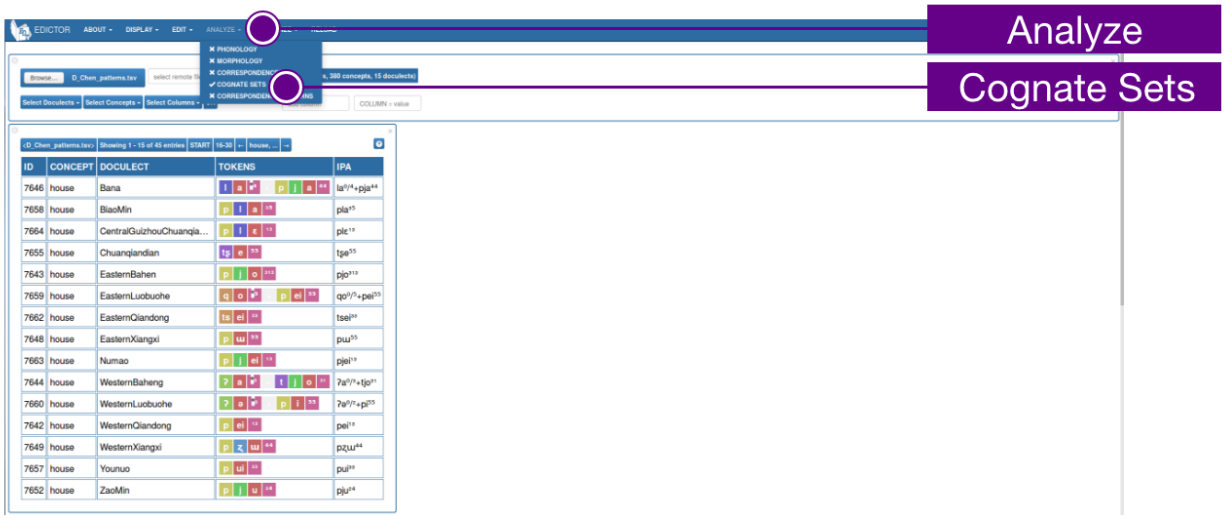
```
$ cldfbench chenhmongmien.wf_crosssemantic
```

The output file is `D_Chen_crossids.tsv`, and we load it into the EDICTOR tool, just as we did before, but when checking the **SETTINGS** in the menu this time, we need to specify that the column "CROSSIDS" holds the partial cognates. To do so, just type in **CROSSIDS** in the text field **Partial Cognates** in the settings menu and then press the **refresh** button.

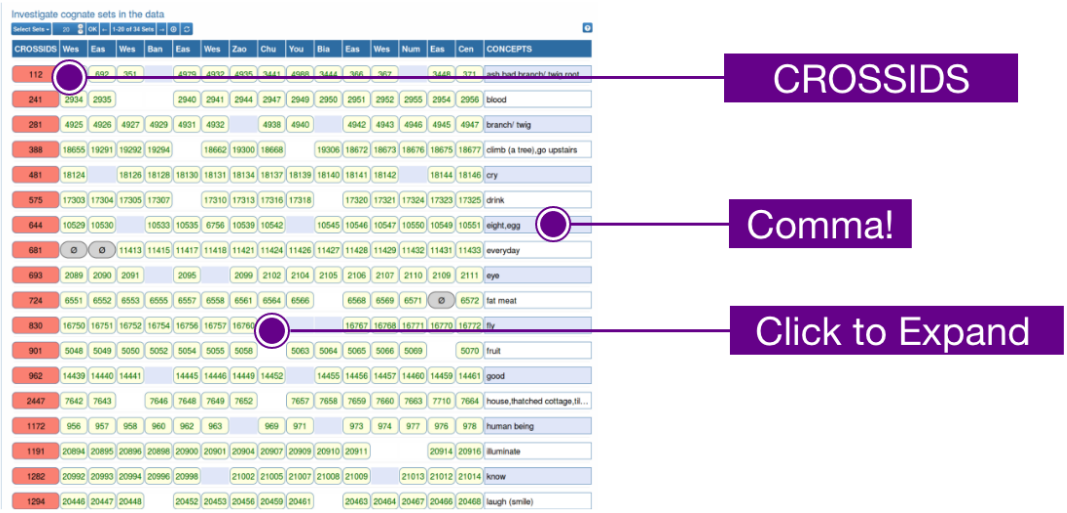


To inspect the distribution of partial cognates, press **ANALYZE** in the top-level menu and select

Cognate sets in the drop-down menu.



As a result, a new panel will open and show the distribution of all cognate sets across the different language varieties. Pressing the red button with the cognate set identifier on the left will open the alignment. Pressing the yellow buttons with the word identifiers will show you the original morpheme. On the right, in the column **CONCEPTS**, you will find those cognate sets which are attested for more than one concept as separated by a comma. Clicking on this field will modify the main wordlist panel in such a way that only the selected concepts will appear.



3.5 From Cross-Semantic Cognates to Sound Correspondence Patterns

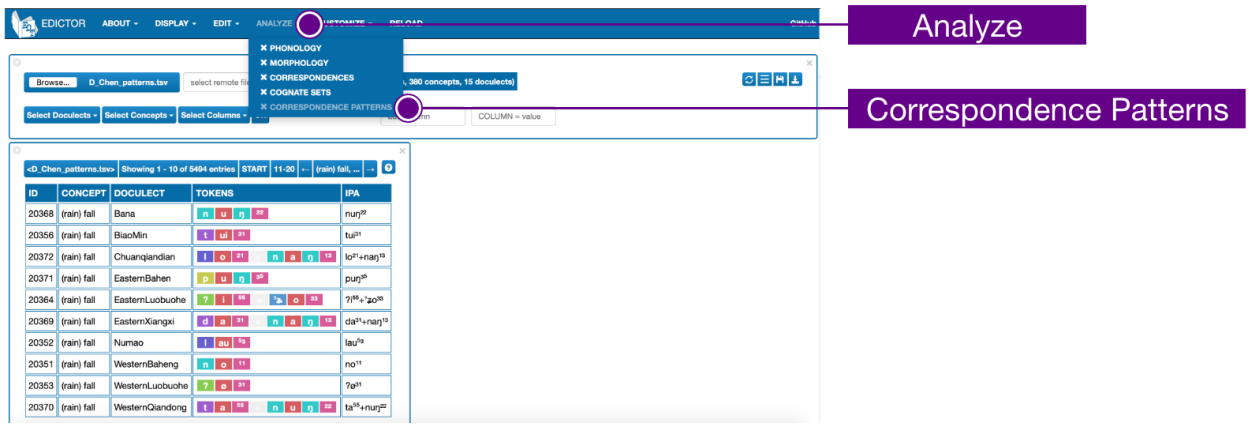
As a final step, we will try to infer the major correspondence patterns in the data, using the algorithm by List (2019)[5] which is available from the `lingrex` package. Running the code is straightforward, as before.

```
$ python 5_correspondence.py
```

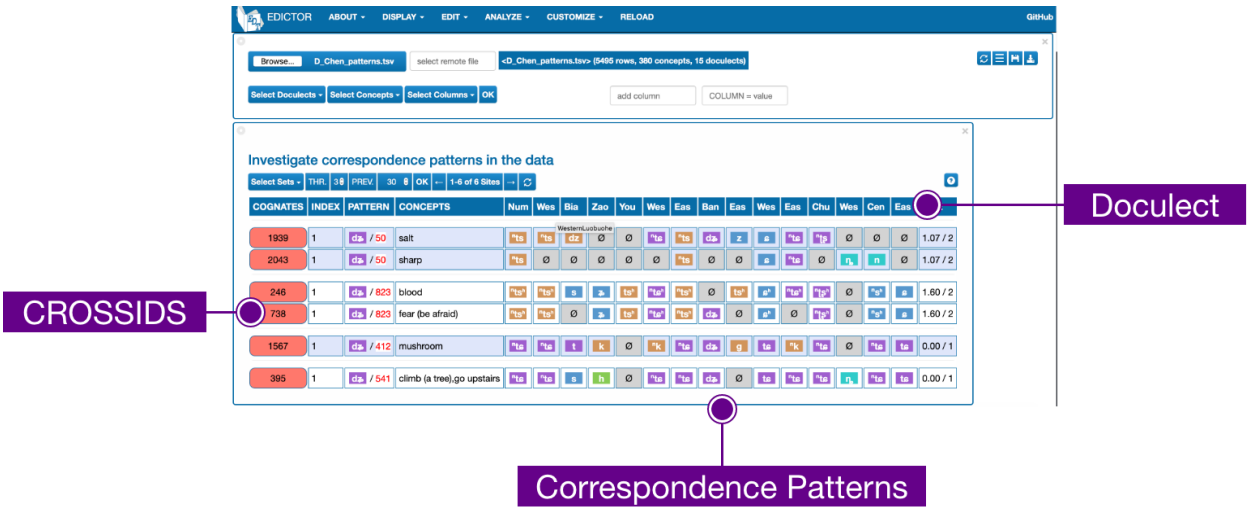
If you want to test the version from the CLDF-repository directly with `cldfbench`, you can type:

```
$ cldfbench chenhmongmien.wf_correspondence
```

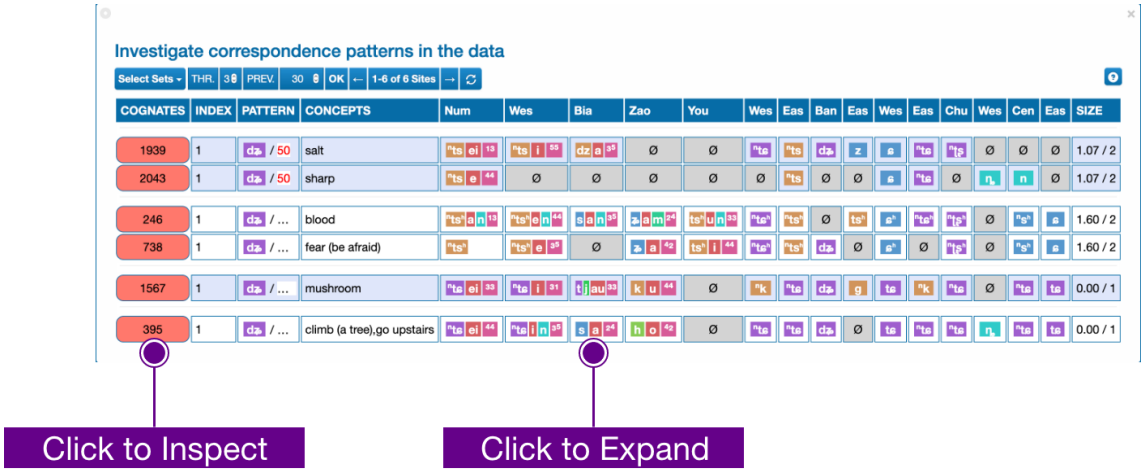
This creates two output files. One, called `D_Chen_patterns.tsv` is the file without wordlist that can be loaded by EDICTOR and inspected, and one file contains the patterns that have been inferred alone, called `D_patterns_Chen.tsv`. In order to inspect the patterns, we recommend to use the EDICTOR tool, which requires the same steps that we already applied when loading our cross-semantic cognates. Once this has been done, press the **ANALYZE** button in the top menu and select **CORRESPONDENCE PATTERNS** in the drop-down menu.



In order to allow for a good display, the doculect names are all abbreviated. Hovering the mouse cursor on an abbreviation will show you the full name.



Clicking on a cell in the correspondence pattern panel will allow you to see not only the sound in question, but the full morpheme in which this sound occurs.



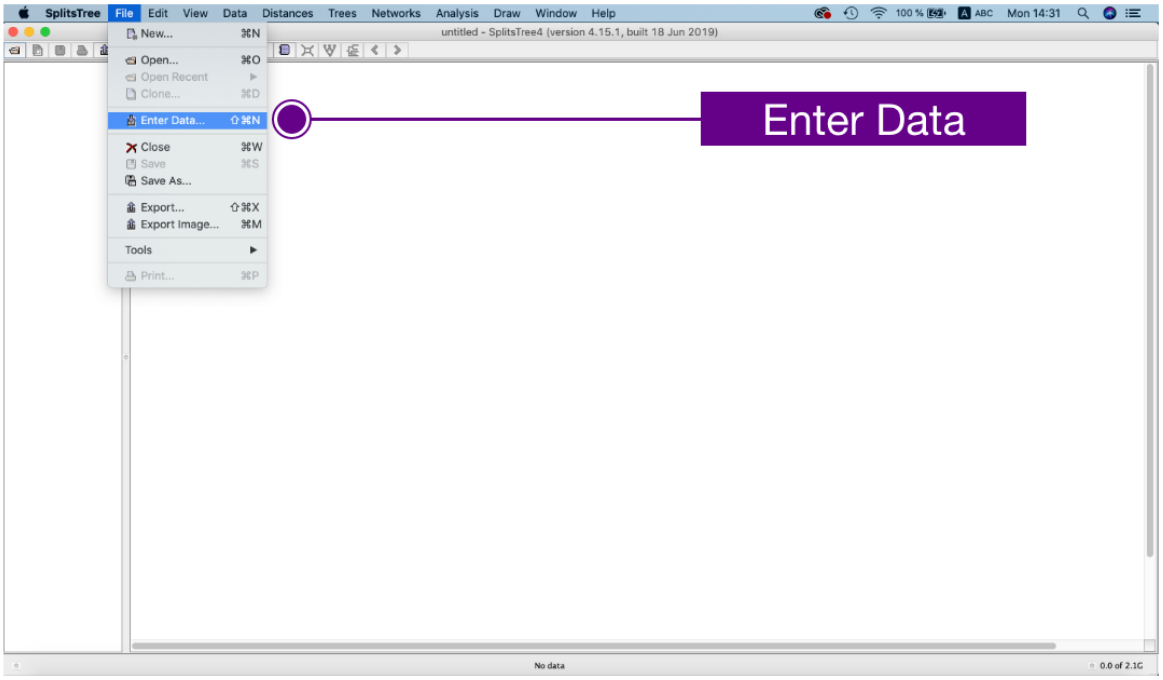
4. Validation

We calculate the shared cognates between language pairs and output the scores in the form of a pairwise distance matrix. The script `6_phylogeny.py` gives two documents, a distance matrix (`A_Chen_distance.dst` or `D_Chen_distance.dst`) and a tree file, based on a Neighbor-Joining analysis (`A_Chen_tree.tre` or `D_Chen_tree.tre`). There are many ways to work with the distance matrix , here, we give one of the approaches to visualize the matrix as a neighbor-net network with the help of SplitsTree. To get started, first make sure to install SplitsTree[21] [6] and follow the installation instructions. In order to compute the distance matrix with our code, use the command line (here we compute it for the entire dataset, so we run it with the keyword `all`)

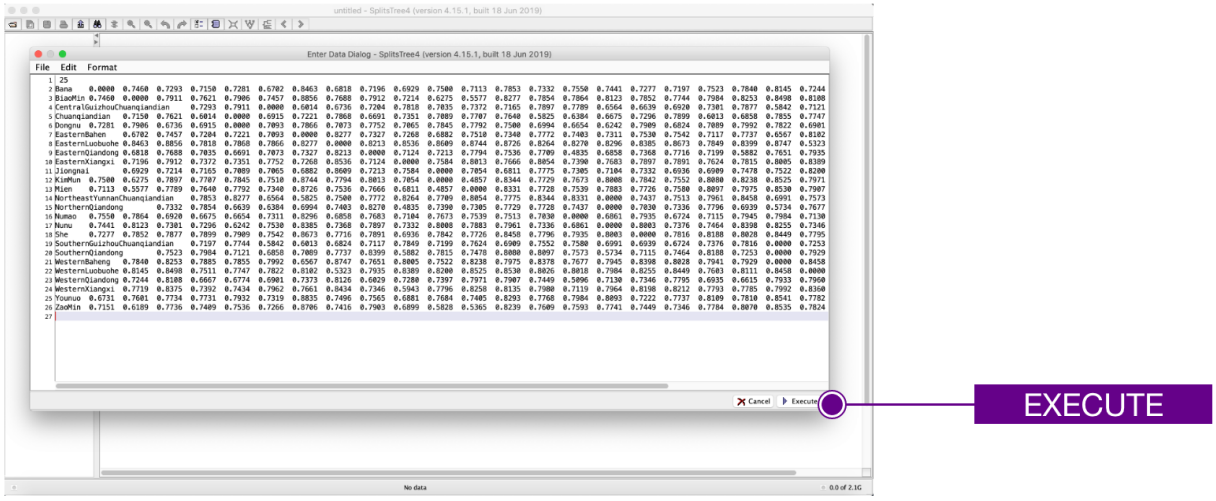
```
$ python 6_phylogeny.py all
```

To generate a Neighbor-Net from the distance matrix, open the file `A_Chen_distance.dst` or `D_Chen_distance.dst` with any plain text editor and start the SplitsTree software. Then click on **File** and **Enter Data**, as shown in the image below.

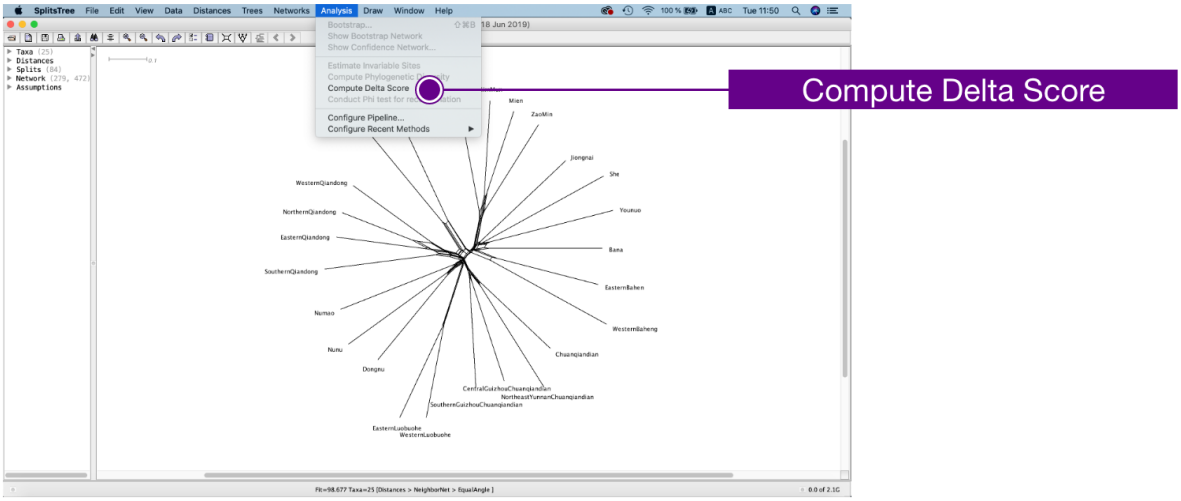
21 https://software-ab.informatik.uni-tuebingen.de/download/splitstree4/welcome.html



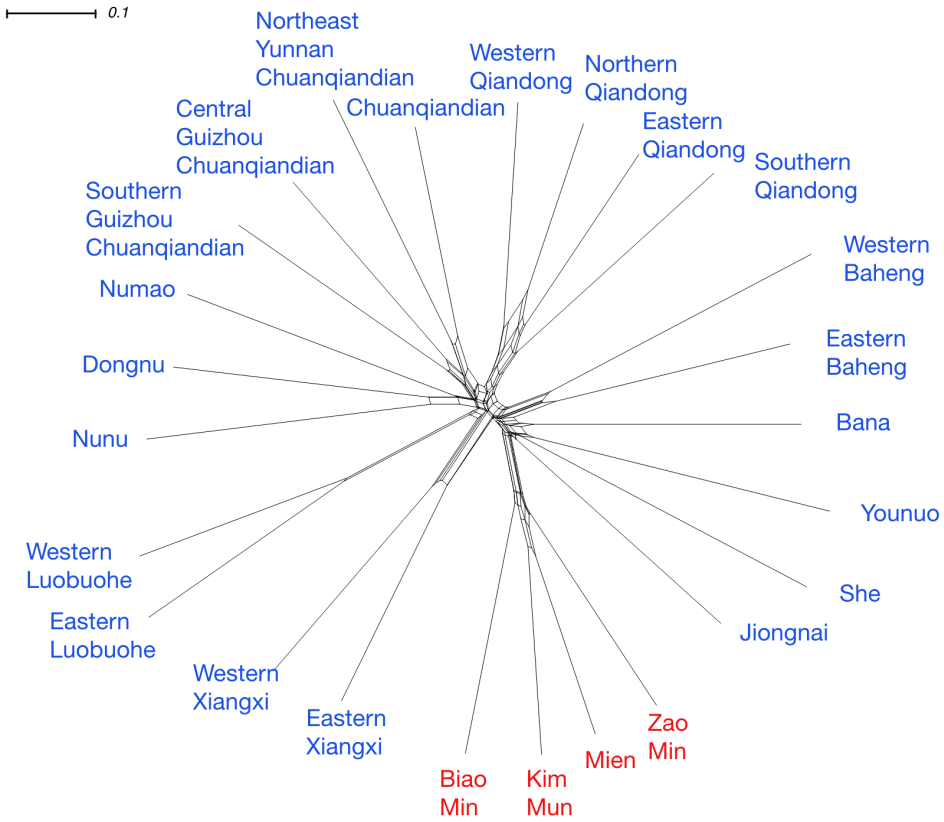
Then copy the distance matrix and paste it into the **Enter Data Dialog**, and press **Execute.**



You can now inspect the network. To analyze the data further, you can compute the delta scores, showing the degree of reticulation in the data, by pressing **Analysis** and then **Compute Delta Score**, as shown below.



The resulting Neighbor-Net is shown in the following figure. For the purpose of illustration, the Mienic language varieties are colored in red, the Hmongic group is highlighted in blue.



The following table shows the delta scores we computed from the data.

Taxon	Delta score
Bana	0.34706
Biao Min	0.27289
Central Guizhou Chuanqiandian	0.29924
Chuanqiandian	0.29172
Dongnu	0.32416
Eastern Baheng	0.32056
Eastern Luobuohe	0.33529
Eastern Qiangong	0.32083
Eastern Xiangxi	0.33736
Jiongnai	0.32644
Kim Mun	0.26992
Mien	0.25672
Northeast Yunnan Chanqiandian	0.29748
Northern Qiandong	0.28447
Numao	0.34185
Nunu	0.32375
She	0.31671
Southern Guizhou Chuanqiandian	0.34376
Southern Qiandong	0.30988
Western Baheng	0.35259
Western Luobuohe	0.3211
Western Qiandong	0.31137
Western Xiangxi	0.35174
Younuo	0.2996
Zao Min	0.26797

The average delta score is 0.313. As mentioned before, the distances between taxa are calculated via shared cognates. The shorter the distances between two taxa, the higher the similarities between them. If the taxa share cognates not only within their group but also outside their groups, the network finds it challenging to determine the best cluster for them. The larger the reticular structure, or the less tree-like the data is, the higher is the delta score. For one particular language variety's delta score this means that this specific language contributes to a certain amount of conflict in the data.

5. Conclusion

In this tutorial, we provided details of how to execute our workflow for Computer-Assisted Language comparison, using the scripts we wrote, while at the same time illustrating how the results can be manually inspected and modified. We have not discussed the details of the code we wrote, but we recommend users proficient in Python to have a look.

6. References

1. List J-M, Walworth M, Greenhill SJ, Tresoldi T, Forkel R. Sequence comparison in computational historical linguistics. Journal of Language Evolution [Internet]. 2018;3(2):130–44. Available from: https://academic.oup.com/jole/article/3/2/130/5050100?guestAccessKey=cf8fe64e-3996-4cb1-ba2c-317a7cd81bf4
2. 陳其光 CQ. Miàoyáo yǔwén [Internet]. Běijīng: Zhōngyāng Mínzú Dàxué 中央民族大学 [Central Institute of Minorities]; 2012. Available from: https://en.wiktionary.org/wiki/Appendix:Hmong-Mien_comparative_vocabulary_list
3. List J-M, Lopez P, Bapteste E. Using sequence similarity networks to identify partial cognates in multilingual wordlists. In: Proceedings of the Association of Computational Linguistics 2016 (Volume 2: Short Papers) [Internet]. Berlin: Association of Computational Linguistics; 2016. pp. 599–605. Available from: http://anthology.aclweb.org/P16-2097
4. Rosvall M, Bergstrom CT. Maps of random walks on complex networks reveal community structure. Proc Natl Acad Sci USA. 2008;105(4):1118–23.
5. List J-M. Automatic inference of sound correspondence patterns across multiple languages. Computational Linguistics [Internet]. 2019;1(45):137–61. Available from: https://www.mitpressjournals.org/doi/full/10.1162/coli_a_00344
6. Huson DH. SplitsTree: Analyzing and visualizing evolutionary data. Bioinformatics. 1998;14(1):68–73.