An Open Source Toolkit for Quantitative Historical Linguistics

Johann-Mattis List
Research Center Deutscher Sprachatlas
Philipps-University Marburg
mattis.list@uni-marburg.de

Steven Moran
Department of General Linguistics
University of Zurich
steven.moran@uzh.ch

Abstract

Given the increasing interest and development of computational and quantitative methods in historical linguistics, it is important that scholars have a basis for documenting, testing, evaluating, and sharing complex workflows. We present a novel open-source toolkit for quantitative tasks in historical linguistics that offers these features. This toolkit also serves as an interface between existing software packages and frequently used data formats, and it provides implementations of new and existing algorithms within a homogeneous framework. We illustrate the toolkit's functionality with an exemplary workflow that starts with raw language data and ends with automatically calculated phonetic alignments, cognates and borrowings. We then illustrate evaluation metrics on gold standard datasets that are provided with the toolkit.

1 Introduction

Since the turn of the 21st century, there has been an increasing amount of research that applies computational and quantitative approaches to historical-comparative linguistic processes. Among these are: phonetic alignment algorithms (Kondrak, 2000; Prokić et al., 2009), statistical tests for genealogical relatedness (Kessler, 2001), methods for phylogenetic reconstruction (Holman et al., 2011; Bouckaert et al., 2012), and automatic detection of cognates (Turchin et al., 2010; Steiner et al., 2011), borrowings (Nelson-Sathi et al., 2011), and proto-forms (Bouchard-Côté et al., 2013).

In contrast to traditional approaches to language comparison, quantitative methods are often emphasized as advantageous with regard to objectivity, transparency and replicability of results. It is striking then that given the multitude of new approaches, very few are publicly available as executable code. Thus in order to replicate a study, researchers have to rebuild workflows from published descriptions and reimplement their approaches and algorithms. These challenges make the replication of results difficult, or even impossible, and they hinder not only the evaluation and comparison of existing algorithms, but also the development of new approaches that build on them.

Another problem is that quantitative approaches that have been released as software are largely incompatible with each other and they show great differences in regard to their input and out formats, application range and flexibility.[1] Given the breadth of research questions involved in determining language relatedness, this is not surprising. Furthermore, the linguistic datasets upon which many analyses and tools are based are only – if at all – available in disparate formats that need manual or semi-automatic re-editing before they can be used as input elsewhere. Scholars who want to analyze a dataset with different approaches often have to (time-consumingly) convert it into various input formats and they have to familiarize themselves with many different kinds of software. As a result, errors may occur during data conversion processes and the output from different tools must also be converted into a comparable format. For the comparison of different output formats or

[1] There is the STARLING database program for lexicostatistical and glottochronological analyses (Starostin, 2000). The Rug/L04 software aligns sound sequences and calculates phonetic distances using the Levensthein distance (Kleiweg, 2009; Levenshtein, 1966). The ASJP-Software also computes the Levenshtein distance (Holman et al., 2011), but its results are based on previously executed phonetic analyses. The ALINE software carries out pairwise alignment analyses (Kondrak, 2000). There are also software packages from evolutionary biology, which are adapted for linguistic purposes, such as MrBayes (Ronquist and Huelsenbeck, 2003), PHYLIP (Felsenstein, 2005), and SplitsTree (Huson, 1998).

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 13–18,
Sofia, Bulgaria, August 4-9 2013.

for the evaluation of competing quantitative approaches, gold standard datasets are desirable.

Towards a solution to these problems, we have developed a toolkit that (a) serves as an interface between existing software packages and data formats frequently used in quantitative approaches, (b) provides high-quality implementations of new and existing approaches within a homogeneous framework, and (c) offers a solid basis for testing, documenting, evaluating, and sharing complex workflows in quantitative historical linguistics. We call this open source toolkit LingPy.

2 Lingpy

LingPy is written in Python3 and is freely available online.[2] The Lingpy website contains an API, documentation, tutorials, example scripts, workflows, and datasets that can be used for training, testing, and comparing results from different algorithms. We use Python because it is flexible and object-oriented, it is easy to write C extensions for scientific computing, and it is approachable to non-programmers (Knight et al., 2007). Apart from a large number of different functions for common automatic tasks, LingPy offers specific modules for implementing general workflows that are used in historical linguistics and which partially mimic the basic aspects of the traditional comparative method (Trask, 2000, 64-67). Figure 1 illustrates the interaction between different modules along with the data they produce. In the following subsections, these modules will be introduced in the order of a typical workflow to illustrate the basic capabilities of the LingPy toolkit in more detail.

2.1 Input Formats

The basic input format read by LingPy is a tab-delimited text file in which the first line (the header) indicates the values of the columns and all words are listed in the following rows. The format is very flexible. No specific order of columns or rows is required. Any additional data can be specified by the user, as long as it is in a separate column. Each row represents a word that has to be characterized by a minimum of four values that are given in separate columns: (1) ID, an integer that is used to uniquely identify the word during calculations, (2) CONCEPT, a gloss which indicates the meaning of the word and which is used to align the words semantically, (3) WORD, the orthographic

[2] `http://lingpy.org`

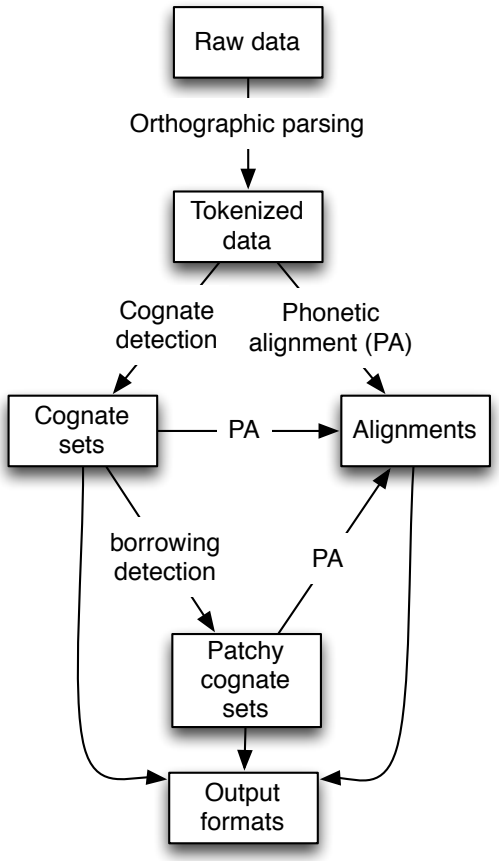

Figure 1: Basic Workflow in LingPy

representation of the word,[3] and (4) TAXON, the name of the language (or dialect) in which the word occurs. Basic output formats are essentially the same, the difference being that the results of calculations are added as separate columns. Table 1 illustrates the basic structure of the input format for a dataset covering 325 concepts translated into 18 Dogon language varieties taken from the Dogon comparative lexical spreadsheet (Heath et al., 2013).[4]

2.2 Parsing and Unicode Handling

Given a dataset in the basic LingPy input format, the first step towards sound-based normalization for automatically identifying cognates and sound changes with quantitative methods is to parse words into tokens. Orthographic tokenization is a non-trivial task, but it is needed to at-

[3] By this we mean a *textual* representation of the word, whether in a document or language-specific orthography or in some form of broad or narrow transcription, etc.

[4] This tokenized dataset and analyses that are discussed in this work are available for download from the LingPy website.

ID	CONCEPT	WORD	TAXON
...	...	...	...
1239	file (tool)	kí:rà	Toro_Tegu
1240	file (tool)	dì:sî:	Ben_Tey
1241	file (tool)	kírâl	Bankan_Tey
1242	file (tool)	dì:jú	Jamsay
...	...	...	...
1249	file (tool)	bìmbú	Tommo_So
1250	file (tool)	bìmbú	Dogul_Dom
1251	file (tool)	dì:zù	Yanda_Dom
1252	file (tool)	bí:mbyé	Mombo
...	...	...	...

Table 1: Basic Input Format of LingPy

tain interoperability across different orthographies or transcription systems and to enable the comparative analysis of languages. LingPy includes a parser that takes as input a dataset and an optional orthography profile, i.e. a description of the Unicode code points, characters, graphemes and orthographic rules that are needed to adequately model a writing system for a language variety as described in a particular document (Moran, 2012, 331). The LingPy parser first normalizes all strings into Unicode Normalization Form D, which decomposes all character sequences and reorders them into one canonical order. This step is necessary because sequences of Unicode characters may differ in their visual and logical orders. Next, if no orthography profile is specified, the parser will use a regular expression match `\X` for Unicode grapheme clusters, i.e. combining character sequences typified by a base character followed by one or more Combing Diacritical Marks. However, another layer of tokenization is usually required to match linguistic graphemes, or what Unicode calls 'tailored grapheme clusters'. Table 2 illustrates the different technological and linguistic levels involved in orthographic parsing.[5]

code points	t	s	h	o	~	◌̃	◌́	s	h	i
"characters"	t	s	h	õ				s	h	i
graphemes	ts^{h}			õ				sh		i

Table 2: Tokens for the string <ts^{h}ṍshi>

So, given the dataset illustrated in Table 1 and an orthography profile that defines the phonemic units in the Dogon comparative lexicon, the LingPy parser produces the IPA tokenized output shown in Table 3.

ID	...	WORD	TOKENS	...
...	...	...	...	...
1239	...	kí:rà	# k í: r à #	...
1240	...	dì:sî:	# d ì: s î: #	...
1241	...	kírâl	# k í r â l #	...
1242	...	dì:jú	# d ì: ʤ ú #	...
...	...	...	...	...
1249	...	bìmbú	# b ì m b ú #	...
1250	...	bìmbú	# b ì m b ú #	...
1251	...	dì:zù	# d ì: z ù #	...
1252	...	bí:mbyé	# b í: m b j é #	...
...	...	...	...	...

Table 3: Orthographic Parsing in LingPy

2.3 Phonetic Alignments

Although less common in traditional historical linguistics, phonetic alignment plays a crucial role in automatic approaches, with alignment analyses being currently used in many different subfields, such as dialectology (Prokić et al., 2009), phylogenetic reconstruction (Holman et al., 2011) and cognate detection (List, 2012a). Furthermore, alignment analyses are very useful for data visualization, since they directly show which sound segments correspond in cognate words.

LingPy offers implementations for many different approaches to pairwise and multiple phonetic alignment. Among these, there are standard approaches that are directly taken from evolutionary biology and can be applied to linguistic data with only slight modifications, such as the Needleman-Wunsch algorithm (Needleman and Wunsch, 1970) and the Smith-Waterman algorithm (Smith and Waterman, 1981). Furthermore, there are novel approaches that use more complex sequence models in order to meet linguistic-specific requirements, such as the Sound-Class-based phonetic Alignment (SCA) method (List, 2012b). Figure 2 shows a plot of the multiple alignment of the counterparts of the concept "stool" in eight Dogon languages. The color scheme for the sound segments follows the sound class distinction of Dolgopolsky (1964).

2.4 Automatic Cognate Detection

The identification of cognates plays an important role in both traditional and quantitative approaches in historical linguistics. Most quantitative approaches dealing with phylogenetic reconstruction are based on previously identified cognate sets distributed over the languages being in-

[5]Note that even when a linguist transcribes a word with the International Phonetic Alphabet (IPA; a transcription system with one-to-one symbol-to-sound correspondences), explicit definitions for phonemes are needed because some IPA diacritics are encoded as Unicode Spacing Modifier Letters, i.e. characters that are not specified as how they combine with a base character, such as aspiration.

Taxon	Alignment								
Ben_Tey	t	ú	ŋ	g	ú	r	-	ú	m
Bankan_Tey	t	ú	ŋ	g	ú	r	-	ú	-
Jamsay	t	ú	ŋ	-	ú	rⁿ	-	ú	-
Perge_Tegu	t	ú	ŋ	-	ú	rⁿ	-	ú	m
Gourou	t	ú	m	-	ú	r	-	ú	-
Yorno_So	t	ɔ́	ŋ	-	ɔ́	-	-	-	-
Tommo_So	t	ú	ŋ	g	ú	r	-	ú	-
Tebul_Ure	t	ú	ŋ	g	ú	r	g	ɔ́	-

Figure 2: Multiple Phonetic Alignment in LingPy

ID	CONCEPT	WORD	TAXON	COGID
...	...	...	...	...
1239	file (tool)	kí:rà	Toro_Tegu	68
1240	file (tool)	dì:sî:	Ben_Tey	69
1241	file (tool)	kírâl	Bankan_Tey	68
1242	file (tool)	dì:jú	Jamsay	69
...	...	...	...	...
1249	file (tool)	bìmbú	Tommo_So	70
1250	file (tool)	bìmbú	Dogul_Dom	70
1251	file (tool)	dì:zù	Yanda_Dom	69
1252	file (tool)	bí:mbyé	Mombo	70
...	...	...	...	...

Table 4: Cognate Detection in LingPy

vestigated (Bouckaert et al., 2012; Bouchard-Côté et al., 2013). Since the traditional approach to cognate detection within the framework of the comparative method is very time-consuming and difficult to evaluate for the non-expert, automatic approaches to cognate detection can play an important role in objectifying phylogenetic reconstructions.

Currently, LingPy offers four alternative approaches to cognate detection in multilingual wordlists. The method by Turchin et al. (2010) employs sound classes as proposed by Dolgopolsky (1964) and assigns words that match in their first two consonant classes to the same cognate set. The NED method calculates the normalized edit distance between words and groups them into cognate sets using a flat cluster algorithm.[6] The SCA and the LexStat methods (List, 2012a) use the same strategy for clustering, but the distances for the SCA method are calculated with help of the SCA alignment method (List, 2012b), and the distances for the LexStat method are derived from previously identified regular sound correspondences. Table 4 shows a small section of the results from the LexStat analysis of the Dogon data. As shown, LingPy follows the STARLING approach in displaying cognate judgments by assigning cognate words the same cognate ID (COGID). In Table 4, the words judged to be cognate are shaded in the same color. The full results are posted on the LingPy website.

2.5 Automatic Borrowing Detection

Automatic approaches for borrowing detection are still in their infancy in historical linguistics. LingPy provides a full reimplementation (along with specifically linguistic modifications) of the minimal lateral network (MLN) approach (Nelson-Sathi et al., 2011). This approach searches for cognate sets which are not compatible with a given reference tree topology. Incompatible (patchy) cognate sets often point to either borrowings or wrong cognate assessments in the data. The results can be visualized by connecting all taxa of the reference tree for which patchy cognate sets can be inferred with lateral links. In Figure 3, the method has been applied again to the Dogon dataset. Cognate judgments for this analysis were carried out with help of LingPy's LexStat method. The tree topology was calculated using MrBayes.

2.6 Output Formats

The output formats supported by LingPy can be divided into three different classes. The first class consists of text-based formats that can be used for manual correction and inspection by importing the data into spreadsheet programs, or simply editing and reviewing the results in a text editor. The second class consists of specific formats for third-party toolkits, such as PHYLIP, SplitsTree, MrBayes, or STARLING. LingPy currently offers support for PHYLIP's distance calculations (DST-format), for tree-representation (Newick-format), for complex representations of character data (Nexus-format), and for the import into STARLING databases (CSV with STARLING markup). The third class consists of new approaches to the visualization of phonetic alignments, cognate sets, and phylogenetic networks. In fact, all plots in this paper were created with LingPy's output formats.

3 Evaluation

In order to improve the performance of quantitative approaches, it is of crucial importance to test and evaluate them. Evaluation is usually done by comparing how well a given approach performs on a reference dataset, i.e. a gold standard, where the results of the analysis are known in advance. LingPy comes with a module for the evaluation of

[6] The normalized edit distance is calculated by dividing the edit distance (Levenshtein, 1966) by the length of the smaller sequence, see Holman et al. (2011) for details.

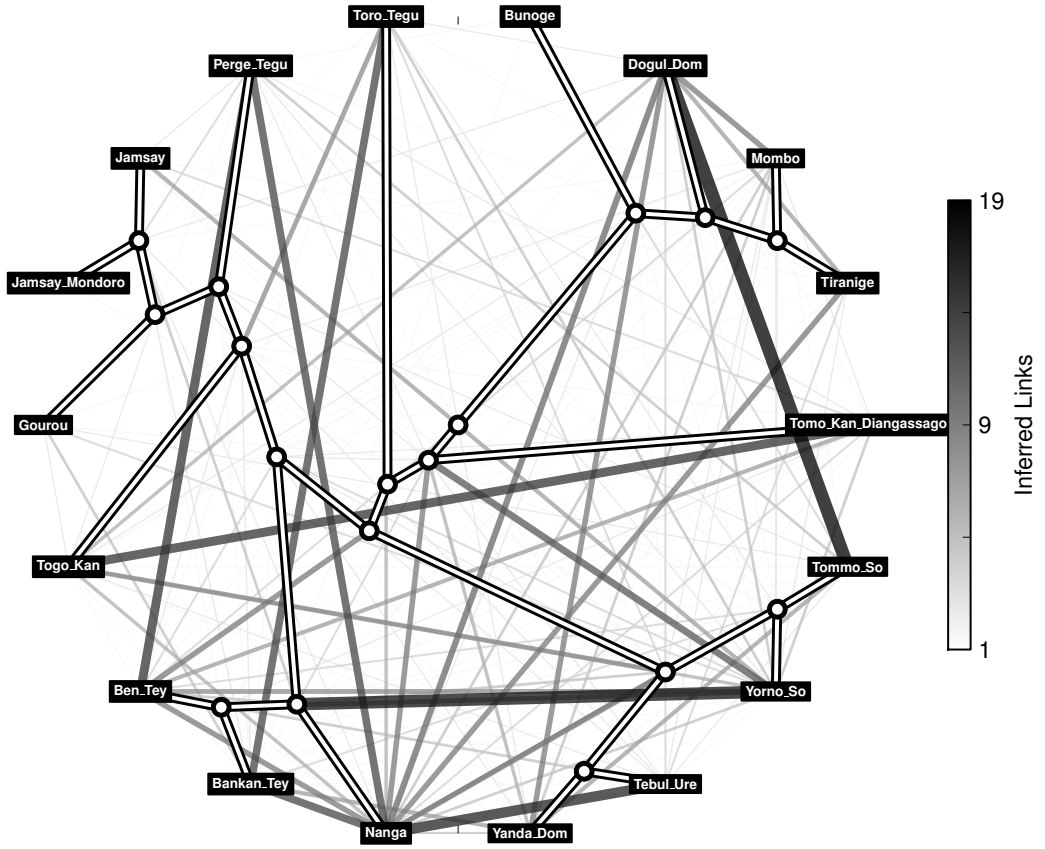

Figure 3: Borrowing Detection in LingPy

basic tasks in historical linguistics, such as phonetic alignment and cognate detection. This module offers both common evaluation measures that are used to assess the accuracy of the respective methods and gold standard datasets encoded in the LingPy input format.

In Figure 4, the performance of the four abovementioned approaches to automatic cognate detection are compared with the gold standard cognate judgments of a dataset covering 207 concepts translated into 20 Indo-European languages taken from the Indo-European Lexical Cognacy (IELex) database (Bouckaert et al., 2012).[7] The pair scores, implemented in LingPy after the description in Bouchard-Côté et al. (2013), were used as an evaluation measure. For all approaches we chose the respective thresholds that tend to yield the best results on all of the gold standards. As shown in Figure, both the SCA and LexStat methods show a higher accuracy than the Turchin and NED methods, with LexStat slightly outperforming SCA. However, the generally bad performance of all approaches on this dataset shows that there is a clear need for improving automatic cognate detection approaches, especially in cases of remote relationship, such as Indo-European.

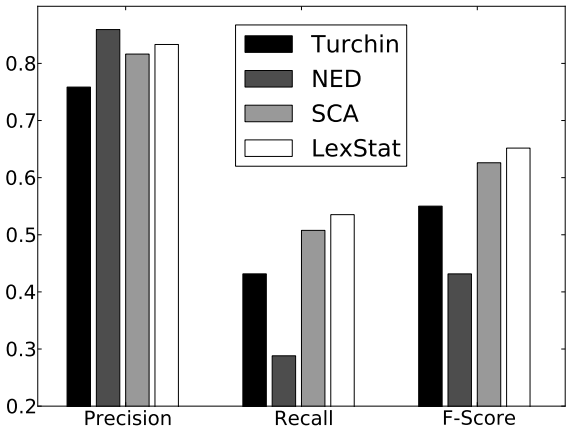

Figure 4: Evaluating Cognate Detection Methods

4 Conclusion

Quantitative approaches in historical linguistics are still in their infancy, far away from being able to compete with the intuition of trained historical

[7] Gold standard here means that the cognate judgments were carried out manually by the compilers of the IELex database.

linguists. The toolkit we presented is a first attempt to close the gap between quantitative and traditional methods by providing a homogeneous framework that serves as an interface between existing packages and at the same time provides high-quality implementations of new approaches.

References

A. Bouchard-Côté, D. Hall, T. L. Griffiths, and D. Klein. 2013. Automated reconstruction of ancient languages using probabilistic models of sound change. *PNAS*, 110(11):4224–4229.

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

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

J. Felsenstein. 2005. Phylip (phylogeny inference package) version 3.6. Distributed by the author. Department of Genome Sciences, University of Washington, Seattle.

J. Heath, S Moran, K. Prokhorov, L. McPherson, and B. Canslter. 2013. Dogon comparative lexicon. URL: `http://www.dogonlanguages.org`.

E. W. Holman, C. H. Brown, S. Wichmann, A. Müller, V. Velupillai, H. Hammarström, S. Sauppe, H. Jung, D. Bakker, P. Brown, O. Belyaev, M. Urban, R. Mailhammer, J.-M. List, and D. Egorov. 2011. Automated dating of the world's language families based on lexical similarity. *Current Anthropology*, 52(6):841–875.

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

B. Kessler. 2001. *The significance of word lists. Statistical tests for investigating historical connections between languages*. CSLI Publications, Stanford.

P. Kleiweg. 2009. RuG/L^{04}. Software for dialectometrics and cartography. Distributed by the Author. Rijksuniversiteit Groningen. Faculteit der Letteren, September.

R. Knight, P. Maxwell, A. Birmingham, J. Carnes, J. G. Caporaso, B. Easton, M. Eaton, M. Hamady, H. Lindsay, Z. Liu, C. Lozupone, D. McDonald, M. Robeson, R. Sammut, S. Smit, M. Wakefield, J. Widmann, S. Wikman, S. Wilson, H. Ying, and G. Huttley. 2007. PyCogent. A toolkit for making sense from sequence. *Genome Biology*, 8(8):R171.

G. Kondrak. 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*, NAACL 2000, pages 288–295, Stroudsburg, PA, USA. Association for Computational Linguistics.

V. I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

J.-M. List. 2012a. LexStat. Automatic detection of cognates in multilingual wordlists. In *Proceedings of the EACL 2012 Joint Workshop of LINGVIS & UNCLH*, pages 117–125. Association for Computational Linguistics.

J.-M. List. 2012b. SCA. Phonetic alignment based on sound classes. In M. Slavkovik and D. Lassiter, editors, *New directions in logic, language, and computation*, number 7415 in LNCS, pages 32–51. Springer, Berlin and Heidelberg.

S. Moran. 2012. *Phonetics information base and lexicon*. Ph.D. thesis, University of Washington.

S. B. Needleman and C. 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, July.

S. Nelson-Sathi, J.-M. List, H. Geisler, H. Fangerau, R. D. Gray, W. Martin, and T. Dagan. 2011. Networks uncover hidden lexical borrowing in Indo-European language evolution. *Proceedings of the Royal Society B*, 278(1713):1794–1803.

J. Prokić, M. Wieling, and J. 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. Association for Computational Linguistics.

F. Ronquist and J. P. Huelsenbeck. 2003. MrBayes 3. Bayesian phylogenetic inference under mixed models. *Bioinformatics*, 19(12):1572–1574.

T. F. Smith and M. S. Waterman. 1981. Identification of common molecular subsequences. *Journal of Molecular Biology*, 1:195–197.

S. A. Starostin. 2000. The STARLING database program. URL: `http://starling.rinet.ru`.

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

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

P. Turchin, I. Peiros, and M. Gell-Mann. 2010. Analyzing genetic connections between languages by matching consonant classes. *Journal of Language Relationship*, 3:117–126.