An automated framework for fast cognate detection and Bayesian phylogenetic inference in computational historical linguistics

Taraka Rama
Department of Linguistics
University of North Texas
taraka.kasi@gmail.com

Johann-Mattis List
Dep. of Ling. and Cult. Evolution (DLCE)
MPI-SHH (Jena)
list@shh.mpg.de

Abstract

We present a fully automated workflow for phylogenetic reconstruction on large datasets, consisting of two novel methods, one for fast detection of cognates and one for fast Bayesian phylogenetic inference. Our results show that the methods take less than a few minutes to process language families that have so far required large amounts of time and computational power. Moreover, the cognates and the trees inferred from the method are quite close, both to gold standard cognate judgments and to expert language family trees. Given its speed and ease of application, our framework is specifically useful for the exploration of very large datasets in historical linguistics.

1 Introduction

Computational historical linguistics is a relatively young discipline which aims to provide automated solutions for those problems which have been traditionally dealt with in an exclusively manual fashion in historical linguistics. Computational historical linguists thus try to develop automated approaches to detect historically related words (called "cognates"; Jäger et al. 2017; List et al. 2017; Rama et al. 2017; Rama 2018a), to infer language phylogenies ("language trees"; Rama et al. 2018; Greenhill and Gray 2009), to estimate the time depths of language families (Rama, 2018b; Chang et al., 2015; Gray and Atkinson, 2003), to determine the homelands of their speakers (Bouckaert et al., 2012; Wichmann et al., 2010), to determine diachronic word stability (Pagel and Meade, 2006; Rama and Wichmann, 2018), or to estimate evolutionary rates for linguistic features (Greenhill et al., 2010).

Despite the general goal of automating traditional workflows, the majority of studies concerned with phylogenetic reconstruction (including studies on dating and homeland inference) still make use of expert judgments to determine cognate words in linguistic datasets, because detecting cognates is usually regarded as hard to automate. The problem of manual annotation is that the process is very time consuming and may show a lack of objectivity, as inter-annotator agreement is rarely tested when creating new datasets. The last twenty years have seen a surge of work in the development of methods for automatic cognate identification. Current methods reach high accuracy scores compared to human experts (List et al., 2017) and even fully automated workflows in which phylogenies are built from automatically inferred cognates do not differ a lot from phylogenies derived from expert's cognate judgments (Rama et al., 2018).

Despite the growing amount of research devoted to automated word comparison and fully automated phylogenetic reconstruction workflows, scholars have so far ignored the computational effort required to apply the methods to large amounts of data. While the speed of the current workflows can be ignored for small datasets, it becomes a challenge with increasing amounts of data, and some of the currently available methods for automatic cognate detection can only be applied to datasets with maximally 100 languages. Although methods for phylogenetic inference can handle far more languages, they require enormous computational efforts, even for small language families of less than 20 varieties (Kolipakam et al., 2018), which make it impossible for scholars perform exploratory studies in Bayesian frameworks.

In this paper, we propose an automated framework for *fast* cognate detection and *fast* Bayesian phylogenetic inference. Our cognate detection algorithm uses an alignment-free technique based on character skip-grams (Järvelin et al., 2007), which has the advantage of neither requiring handcrafted nor statistically trained matrices of proba-

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 6225–6235
Florence, Italy, July 28 - August 2, 2019.

ble sound correspondences to be supplied.[1] Our fast approach to Bayesian inference uses a simulated annealing variant (Andrieu et al., 2003) of the original MCMC algorithm to compute a *maximum-a-posteriori* (MAP) tree in a very short amount of time.

Testing both our fast cognate detection and our fast phylogenetic reconstruction approach on publicly available datasets, we find that the results presented in the paper are comparable to the alternative, much more time-consuming algorithms currently in use. Our automatic cognate detection algorithm shows results comparable to those achieved by the SCA approach (List, 2014), which is one of the best currently available algorithms that work without inferring regular sound correspondences prior to computation (List et al., 2017). Our automatically inferred MAP trees come close to the expert phylogenies reported in Glottolog (Hammarström et al., 2017), and are at least as good as the phylogenies inferred with MrBayes (Ronquist et al., 2012), one of the most popular programs for phylogenetic inference. In combination, our new approaches offer a fully automated workflow for phylogenetic reconstruction in computational historical linguistics, which is so fast that it can be easily run on single core machines, yielding results of considerable quality in less than 15 minutes for datasets of more than 50 languages.

In the following, we describe the fast cognate detection program in Section 2. We describe both the regular variant of the phylogenetic inference program and our simulated annealing variant in Section 3. We present the results of our automated cognate detection and phylogenetic inference experiments and discuss the results in Section 4. We conclude the paper and present pointers to future work in Section 5.

2 Fast Cognate Detection

Numerous methods for automatic cognate detection in historical linguistics have been proposed in the past (Jäger et al., 2017; List, 2014; Rama et al., 2017; Turchin et al., 2010; Arnaud et al., 2017). Most of them are based on the same general workflow, by which – in a first stage – all possible pairs of words within the same meaning slot of a wordlist are compared with each other in order to compute a matrix of pairwise distances or similarities. In a second stage, a flat cluster algorithm or a network partitioning algorithm is used to partition all words into cognate sets, taking the information in the matrix of word pairs as basis (List et al., 2018b). Differences between the algorithms can be found in the way in which the pairwise word comparisons are carried out, to which degree some kind of pre-processing of the data is involved, or which algorithm for flat clustering is being used.

Since any automated word comparison that starts from the comparison of word pairs needs to calculate similarities or distances for all $\frac{n^2-n}{2}$ possible word pairs in a given concept slot, the computation cost for all algorithms which employ this strategy exponentially increases with the number of words being compared. If methods additionally require to pre-process the data, for example to search across all language-pairs for language-specific similarities, such as regularly corresponding sounds (List et al., 2017; Jäger et al., 2017), the computation becomes impractical for datasets of more than 100 languages.

A linear time solution was first proposed by Dolgopolsky (1964). Its core idea is to represent all sound sequences in a given dataset by their *consonant classes*. A consonant class is hereby understood as a rough partitioning of speech sounds into groups that are conveniently used by historical linguistics when comparing languages (such as *velars*, [k, g, x], *dentals* [t, d, θ], or *liquids* [r, l, ʁ], etc.). The major idea of this approach is to judge all words as cognate whose initial two consonant classes *match*. Given that the method requires only that all words be converted to their first consonant classes, this approach, which is now usually called *consonant-class matching* approach (CCM, Turchin et al. 2010), is very fast, since its computation costs are linear with respect to the number of words being compared. The task of assigning a given word to a given cognate set is already fulfilled by assigning a word a given string of consonant classes.

The drawback of the CCM approach is a certain lack of accuracy. While being quite conservative when applied to words showing the same meaning, the method likewise misses many valid matches and thus generally shows a low recall. This is most likely due to the fact that the method does not not

[1] Although Rama (2015) uses skip-grams, the approach in the paper requires hand-annotated data which we intend to overcome in this paper.

contain any *alignment* component. Words are converted to sound-class strings and only complete matches are allowed, while good partial matches can often be observed in linguistic data, as can be seen from the comparison of English *daughter*, represented as TVTVR in sound classes compared to German *Tochter* TVKTVR.

In order to develop an algorithm for automatic cognate detection which is both fast and shows a rather high degree of accuracy, we need to (1) learn from the strategy employed by the CCM method in avoiding any pairwise word comparison, while – at the same time – (2) avoiding the problems of the CCM method by allowing for a detailed sequence comparison based on some kind alignment techniques. Since the CCM method only compares the first two consonants per word, it cannot identify words like English *daughter* and German *Tochter* as cognate, although the *overall* similarity is obvious when comparing the whole strings.

A straightforward way to account for our two requirements is using *skip-grams* of sound-class representations and to represent words and sound-class skip-grams in a given dataset in form of a *bipartite network*, in which words are assigned to one type of node, and skip-grams to another one. In such a network, we could compute multiple representations of TVTVR and TVKTVR directly and later see, in which of them the two sequences match. If, for example, we computed all n-grams of length 5 allowing to skip one, we would receive TVTVR for English (only possible solution) and VKTVR, TKTVR, TVTVR, TVKVR, TVKTR, and TVKTV for German, with TVTVR matching the English word, and thus being connected to both words by an edge in our bipartite network (see Figure 1).

Similarly, when computing a modified variant of skip-grams based on n-grams of size 3, where only consonants are taken into account, and in which we allow to replace up to one segment systematically by a gap-symbol ("-"), we can see from Table 1 that the structure of matching n-grams directly reflects the cognate relations, with Greek çɛri "hand" opposed to German *Hand* and English *hand* (both cognate), as well as Russian [ruka], Polish rɛ̃ŋka (both cognate).

Note that the use of skip-grams here mimics the alignment component of those automatic cognate detection methods in which alignments are used. The difference is that we do not compute the alignments between a sequence pair only, but project each word to a potential (and likewise also restricted) alignment representation. Note also that – even if skip-grams may take some time to compute – our approach presented here is essentially *linear* in computation time requirements, since the skip-gram calculation represents a constant factor. When searching for potential cognates in our bipartite network, we can say that (A) all connected components correspond to cognate sets, or (B) use some additional algorithm to partition the bipartite network into our putative cognate sets. While computation time will be higher in the latter case, both cases will be drastically faster than existing popular methods for automatic cognate detection, since our bipartite-graph-based approach essentially avoids pairwise word comparisons.

Following these basic ideas, we have developed a new method for *fast cognate detection using bipartite networks of sound-class-based skip-grams* (*BipSkip*), implemented as a Python library (see SI 1). The basic working procedure is extremely straightforward and consists of three stages. In a first stage, a bipartite network of words and their corresponding skip-grams is constructed, with edges drawn between all words and their corresponding skip-grams. In a second, optional stage, the bipartite graph is refined by deleting all skip-gram nodes which are linked to fewer word nodes than a user-defined threshold. In a third stage, the bipartite graph is projected to a monopartite graph and partitioned into cognate sets, either by its connected components, or with help of graph partitioning algorithms such as, e.g., Infomap (Rosvall and Bergstrom, 2008).

Since it is difficult to assess which kinds of skip-grams and which kinds of sound-class systems would yield the most promising results, we conducted an exhaustive parameter training using the data of List (2014, see details reported in SI 2). This resulted in the following parameters used as default for our approach: (1) compute skip grams exclusively from consonant classes, (2) compute skip-grams of length 4, (3) include a *gapped* version of each word form (allowing for matches with a replacement), (4) use the SCA sound class model (List, 2014), and (5) prune the graph by deleting all skip-gram nodes which link to less than 20% of the median degree of all skip-gram nodes in the data. This setting

Figure 1: Bipartite graph of English *daughter*, German *Tochter*, and their corresponding sound-class-based skip-grams of size 5.

yielded F-scores of 0.854 (connected components partitioning) and 0.852 (Infomap partitioning) on the training data (using B-Cubes as measure, cf. Amigó et al. 2009 and section 4.2), suggesting that our BipSkip method performs in a manner comparable to the SCA method for automatic cognate detection (List, 2014), which is based on pairwise sequence comparison methods using improved sound class models and alignment techniques. This also means that it clearly outperforms the CCM approach on the training data (scoring 0.8) as well as the computationally rather demanding edit distance approach (scoring 0.814, see List et al. 2017).

IPA	çeri	hant	hænd	ruka	rɛ̃ŋka
Cognacy	1	2	2	3	3
Sound Classes	CERI	HANT	HENT	RYKA	RENKA
H-T	-	+	+	-	-
HN-	-	+	+	-	-
HNT	-	+	+	-	-
R-K	-	-	-	+	+

Table 1: Shared skip-grams in words meaning "hand" in Greek, German, English, Russian, and Polish reflect the known cognate relations of the word.

3 Fast Phylogenetic Inference

Methods for Bayesian phylogenetic inference in evolutionary biology and historical linguistics (Yang and Rannala, 1997) are all based on the following Bayes rule:

$$f(\Psi|X) = \frac{f(X|\Psi)f(\Psi)}{f(X)}, \qquad (1)$$

where each state Ψ is composed of τ the tree topology, $\mathbf{T}$ the branch length vector of the tree, and θ the substitution model parameters where X is a binary cognate data matrix where each column codes a cognate set as a binary vector. The posterior distribution $f(\Psi|X)$ is difficult to calculate analytically since one has to sum over all the possible rooted topologies ($\frac{(2L-3)!}{2^{L-2}(L-2)!}$) increases factorially with the number of languages in the sample. Therefore, Markov Chain Monte Carlo (MCMC) methods are used to estimate the posterior probability of Ψ.

The Metropolis-Hastings algorithm (a MCMC algorithm) is used to sample the parameters from the posterior distribution. This algorithm constructs a Markov chain by proposing a new state Ψ^* and then accepting the proposed state Ψ^* with the probability given in equation 2 where, $q(.)$ is the proposal distribution.

$$r = \frac{f(X|\Psi^*)f(\Psi^*)}{f(X|\Psi)f(\Psi)}\frac{q(\Psi|\Psi^*)}{q(\Psi^*|\Psi)} \qquad (2)$$

The likelihood of the data to the new parameters is computed using the pruning algorithm (Felsenstein, 2004, 251-255), which is a special case of the variable elimination algorithm (Jordan et al., 2004). We assume that the parameters $\tau, \mathbf{T}, \theta$ are independent of each other. In the above procedure, a Markov chain is run for millions of steps and sampled at regular intervals (called thinning) to reduce autocorrelation between the sampled states. A problem with the above procedure is that the chain can get stuck in a local maxima when the posterior has multiple peaks. A different approach known as Metropolis-coupled Markov Chain Monte-Carlo methods (MC3) has been applied to phylogenetics to explore the tree space efficiently (Altekar et al., 2004).

3.1 MC3

In the MC3 approach, n chains are run in parallel where $n-1$ chains are heated by raising the posterior probability to a power $1/T_i$ where T_i is the temperature of ith chain defined as $1+\delta(i-1)$ where $\delta > 0$. A heated chain ($i > 1$) can explore peaks more efficiently than the cold chain since the posterior density is flattened. The MC3 approach swaps the states between a cold chain and a hot chain at regular intervals using a modified Metropolis-Hastings ratio. This swapping procedure allows the cold chain to explore multiple peaks in the tree space successfully. The MC3

procedure is computationally expensive since it requires multiple CPU cores to run the Markov chains in parallel. As a matter of fact, Rama et al. (2018) employ the MC3 procedure (as implemented in MrBayes; Ronquist et al., 2012) to infer family phylogenetic trees from automatically inferred cognate judgments.

3.2 Simulated Annealing

In this paper, we employ a computationally less intensive and a fast procedure inspired from simulated annealing (Andrieu et al., 2003) to infer the *maximum-a-posteriori* (MAP) tree. We refer the simulated annealing MCMC as MAPLE (MAP estimation for Language Evolution) in the rest of the paper. In this procedure, the Metropolis-Hastings ratio is computed according to the equation 3. In this equation, the initial temperature T_0 is set to a high value and then decreased according to a cooling schedule until $T_i \rightarrow 0$. The final state of the chain is treated as the *maximum-a-posteriori* (MAP) estimate of the inference procedure. We implement our own tree inference software in Cython which is made available along with the paper.

$$r = \left(\frac{f(X|\Psi^*)f(\Psi^*)}{f(X|\Psi)f(\Psi)} \right)^{1/T_i} \frac{q(\Psi|\Psi^*)}{q(\Psi^*|\Psi)} \quad (3)$$

All our Bayesian analyses use binary datasets with states 0 and 1. We employ the Generalized Time Reversible Model (Yang, 2014, Ch.1) for computing the transition probabilities between individual states $(0, 1)$. The rate variation across cognate sets is modeled using a four category discrete Γ distribution (Yang, 1994) which is sampled from a Γ distribution with shape parameter α.

MCMC moves We employ multiple moves to sample the parameters. For continuous parameters such as branch lengths and shape parameter we use a multiplier move with exponential distribution ($\mu = 1$) as the proposal distribution. In the case of the stationary frequencies, we employ a uniform slider move that randomly selects two states and proposes a new frequency such that the sum of the frequencies of the states does not change. We use two tree moves: Nearest neighbor interchange (NNI) and a specialized Subpruning and Regrafting move that operates on leaf nodes to propose new trees (Lakner et al., 2008).

Cooling Schedule The cooling schedule is very important for the best performance of a simulated annealing algorithm (Andrieu et al., 2003). We experimented with a linear cooling schedule that starts with a high initial temperature T_0 and reduces the temperature at iteration i through $T_i = \lambda T_{i-1}$ where $0.85 <= \lambda <= 0.96$ (Du and Swamy, 2016). We decrease the value of T_i until $T_i = 10^{-5}$. In this paper, we experiment with reducing the temperature over step size s starting from an initial temperature T_0.

4 Evaluation

4.1 Materials

All the data for training and testing was taken from publicly available sources and has further been submitted along with the supplementary material accompanying this paper. For training of the parameters of our BipSkip approach for fast cognate detection, the data by List (2014) was used in the form provided by List et al. (2017). This dataset consists of six subsets each covering a subgroup of a language family of moderate size and time depth (see SI 2). To test the BipSkip method, we used both the test set of List et al. (2017), consisting of six distinct datasets of moderate size, as well as five large datasets from five different language families (Austronesian, Austro-Asiatic, Indo-European, Pama-Nyungan, and Sino-Tibetan) used for the study by Rama et al. (2018) on the potential of automatic cognate detection methods for the purpose of phylogenetic reconstruction. The latter dataset was also used to test the MAPLE approach for phylogenetic inference. The other two datasets could not be used for the phylogenetic inference task, since these datasets contain a large number of largely unresolved dialect varieties for which no expert classifications are available at the moment. More information on all datasets is given in Table 2.

4.2 Evaluation Methods

We evaluate the results of the automatic cognate detection task through B-Cubed scores (Amigó et al., 2009), a measure now widely used for the task of assessing how well a given cognate detection method performs on a given test dataset (Hauer and Kondrak, 2011; List et al., 2016; Jäger et al., 2017; List et al., 2017). B-Cubed scores are reported in form of *precision*, *recall*, and *F-scores*, with high precision indicating a high amount of

Dataset	Concepts	Languages	Cognates
Austronesian	210	20	2864
Bai	110	9	285
Chinese	140	15	1189
Indo-European	207	20	1777
Japanese	200	10	460
Ob-Ugrian	110	21	242

(a) BipSkip training data.

Dataset	Concepts	Languages	Cognates
Bahnaric	200	24	1055
Chinese	180	18	1231
Huon	139	14	855
Romance	110	43	465
Tujia	109	5	179
Uralic	173	7	870

(b) BipSkip test data.

Dataset	Concepts	Languages	Cognates
Austronesian	210	45	3804
Austro-Asiatic	200	58	1872
Indo-European	208	42	2157
Pama-Nyungan	183	67	6634
Sino-Tibetan	110	64	1402

(c) BipSkip and MAPLE test data.

Table 2: Datasets (name, concepts, and languages), used for training (a) and testing of BipSkip (b, c) and MAPLE (c). Data in (a) is from List (2014), data in (b) is from List et al. (2017), and data in (c) comes from Rama et al. (2018).

true positives, and high recall indicating a high amount of true negatives. Details along with an example on how B-Cubed scores can be inferred are given in List et al. (2017). An implementation of the B-Cubed measure is available from the LingPy Python library for quantitative tasks in historical linguistics (List et al., 2018a).

We evaluate the performance of the phylogenetic reconstruction methods by comparing them to expert phylogenies through the *Generalized Quartet Distance* (GQD), which is a variant of the quartet distance originally developed in bioinformatics (Christiansen et al., 2006) and adapted for linguistic trees by Pompei et al. (2011). A quartet consists of four languages and can either be a *star* or a *butterfly*. The quartet distance is defined as the total number of different quartets divided by the total number of possible quartets ($\binom{n}{4}$) in the tree. This definition of quartet distance penalizes the tree when the gold standard tree has non-binary nodes which is quite common in linguistic phylogenies. The GQD version disregards star quartets and computes the distance between the inferred tree and the gold standard tree as the ratio between the number of different butterflies and the total number of butterflies in the gold standard tree.

4.3 Implementation

Both methods are implemented in form of Python packages available – along with detailed installation instructions – from the supplemental material accompanying the paper (SI 1 and SI 4). While the BipSkip method for fast cognate detection is implemented in form of a plug-in for the LingPy library and thus accepts the standard wordlist formats used in LingPy as input format, MAPLE reads the data from files encoded in the Nexus format (Maddison et al., 1997).

4.4 Results

Fast Cognate Detection We tested the two variants, of the new BipSkip approach for automatic cognate detection, connected components and Infomap (Rosvall and Bergstrom, 2008), on the two test sets (see Table 2) and calculated the B-Cubed precision, recall, and F-scores. To allow for a closer comparison with cognate detection algorithms of similar strength, we also calculated the results for the SCA method for cognate detection described in List et al. (2017), and the CCM approach described in Section 2. The SCA method uses the Sound-Class-Based Alignment algorithm (List, 2014) to derive distance scores for all word pairs in a given meaning slot and uses a flat version of the UPGMA method (Sokal and Michener, 1958) to cluster words into cognate sets. Table 3 lists the detailed results for all four approaches and all 11 subsets of the two datasets, including the computation time.

As can be seen from the results in Table 3, the BipSkip algorithm clearly outperforms the CCM method in terms of overall accuracy on both datasets. It also comes very close in performance to the SCA method, while at the same time only requiring a small amount of the time required to run the SCA analysis. An obvious weakness of our current BipSkip implementation is the performance on South-East Asian language data. Here, we can see that the exclusion of tones and vowels, dictated by our training procedure, leads to a higher amount of false positives. Unfortunately, this cannot be overcome by simply includ-

Dataset	CCM			BipSkip-CC			BipSkip-IM			SCA		
	P	R	FS	P	R	FS	P	R	FS	P	R	FS
Bahnaric	**0.92**	0.63	0.75	0.82	**0.87**	0.84	0.85	0.85	0.85	0.88	0.84	**0.86**
Chinese	**0.81**	0.74	0.78	0.66	**0.95**	0.77	0.68	0.93	0.78	0.80	0.79	**0.79**
Huon	**0.89**	0.84	0.87	0.73	**0.95**	0.80	0.73	0.93	0.81	0.79	0.93	**0.86**
Romance	**0.94**	0.61	0.74	0.91	**0.89**	**0.90**	0.92	0.86	0.89	0.93	0.81	0.87
Tujia	**0.97**	0.74	0.84	0.89	**0.95**	**0.90**	0.89	0.90	**0.90**	**0.97**	0.83	0.89
Uralic	**0.96**	0.86	0.91	0.84	**0.93**	0.88	0.84	0.93	0.88	0.91	0.91	**0.91**
TOTAL	**0.92**	0.74	0.81	0.81	**0.91**	0.85	0.82	0.90	0.85	0.88	0.85	**0.86**
TIME	0m1.400s			0m2.960s			0m5.909s			0m25.768s		

(a) Test Data from List et al. 2017

Dataset	CCM			BipSkip-CC			BipSkip-IM			SCA		
	P	R	FS	P	R	FS	P	R	FS	P	R	FS
Austro-Asiatic	**0.79**	0.64	0.71	0.61	**0.81**	0.70	0.67	0.77	0.72	0.73	0.80	**0.76**
Austronesian	**0.88**	0.58	0.70	0.72	0.72	0.72	0.77	0.68	0.72	0.82	**0.74**	**0.77**
Indo-European	**0.89**	0.64	0.75	0.82	0.73	0.77	0.86	0.69	0.77	**0.89**	**0.74**	**0.81**
Pama-Nyungan	0.64	0.82	0.72	0.71	0.79	0.75	**0.75**	0.77	**0.76**	0.59	**0.85**	0.69
Sino-Tibetan	**0.78**	0.35	0.48	0.59	**0.62**	**0.60**	0.61	0.59	**0.60**	0.73	0.46	0.56
TOTAL	**0.80**	0.61	0.67	0.69	**0.73**	0.71	0.73	0.70	0.71	0.75	0.72	**0.72**
TIME	0m2.938s			0m9.642s			0m17.642s			2m40.472s		

(b) Test Data from Rama et al. 2018

Table 3: Results of the cognate detection experiments. Table (a) presents the results for the performance of the four methods tested on the dataset by List et al. (2017): the CCM method, our new BipSkip methods in two variants (with connected components clusters, labelled CC, and the Infomap clusters, labelled IM), and the SCA method. Table (b) presents the results on the large testset by Rama et al. (2018). The column TIME indicates the time the code needed to run on a Linux machine (Thinkpad X280, i5, 8GB, ArchLinux OS), using the Unix "time" command (reporting the *real* time value).

ing tones in the skip-grams, since not all languages in the South-East Asian datasets (Sino-Tibetan and Austro-Asiatic) are tonal, and tone matchings would thus lead to an unwanted clustering of tonal and non-tonal languages in the data, which would contradict certain subgroups in which tone developed only in a few language varieties, such as Tibetan.

The most promising approach to deal consistently with language families such as Sino-Tibetan would therefore be to extend the current approach to identify *partial* instead of *complete* cognates (List et al., 2016), given the prominence of processes such as compounding or derivation in the history of Sino-Tibetan and its descendants.

Partial cognates, however, do not offer a direct solution to the problem, since we currently lack phylogenetic algorithms that could handle partial cognates (List, 2016), while approaches to convert partial into full cognates usually require to take semantic information into account (Sagart et al., 2019, 10321). In addition to any attempt to improve on BipSkip by enhancing the training of features used for South-East Asian languages, consistent approaches for the transformation of partial into complete cognate sets will have to be developed in the future.

Neither of the two BipSkip approaches can compete with the LexStat-Infomap approach, which yields F-scores of 0.89 on the first test set (see List et al. 2017) and 0.77 on the second test set (see Rama et al. 2018), but this is not surprising, given that neither of the four approaches compared here computes regular sound correspondence information. The obvious drawback of LexStat is its computation time, with more than 30 minutes for the first, and more than two hours for the second test set. While the superior results surely justify its use, the advantage of methods like BipSkip is that they can be used for the purpose of exploratory data analysis or web-based applications.

Fast Phylogenetic Inference We present the results of the phylogenetic experiments in Table 4. Each sub-table shows the setting for s, T_0 that yielded the lowest GQD for each cognate detection method. We experimented over a wide range of settings for $s \in \{1, 5, 10, 20, 40, 80, 100\}$ and $T_0 \in \{10, 20, \ldots, 90, 100\}$. We provide the time and the number of generations taken to infer the MAP tree for each cognate inference program and language family. We note that the longest run takes less than fifteen minutes across all the families. In comparison, the results reported by Rama et al. (2018) using MrBayes takes at least four hours on six cores for each of the language family using the SCA method.

We examined which settings of s/T_0 give the lowest results and found that low step sizes such as 1 give the lowest results for a wide range of T_0. We examined the results across the settings and found that the best results can be achieved with a step size above 20 with initial temperature set to 50. The lowest GQD distances were obtained with the SCA cognates. The BipSkip-IM method emerged as the winner in the case of the Pama-Nyungan language family. The best result for Pama-Nyungan is better than the average GQD obtained through expert cognate judgments reported in Rama et al. (2018). The weakness of the BipSkip methods with respect to the Sino-Tibetan language family is also visible in terms of the GQD distance.

Comparing the results obtained for the SCA cognates obtained with MAPLE against the ones inferred with MrBayes as reported in Rama et al. (2018), it becomes also clear that our method is at least as good as MrBayes, showing better results in Austro-Asiatic, Austronesian, and Pama-Nyungan.

MAPLE with gold standard cognates We further tested if gold standard cognates make a difference in the inferred tree quality. We find that the tree quality improves if we employ gold standard cognates to infer the trees. This result supports the research track of developing high quality automated cognate detection systems which can be employed to analyze hitherto less studied language families of the world.

Convergence We investigated if the MAPLE algorithm infers trees whose quality improves across the generations by plotting the GQD of the sampled trees against the temperature for all the five best settings of s/T_0 (in bold in Table 4) in Figure 2. The figure clearly shows that at high temperature settings, the quality of the trees is low whereas as temperature approaches zero, the tree quality also gets better for all the language fami-

Family	s/T_0	GQD	NGens	Time (s)
Austro-Asiatic	80/10	0.0155	18080	282.548
Austronesian	20/80	0.0446	5320	46.698
Indo-European	20/40	**0.0138**	5060	46.014
Pama-Nyungan	40/60	0.1476	10440	224.036
Sino-Tibetan	80/60	0.0958	20880	295.157

(a) Results for CCM cognates.

Family	s/T_0	GQD	NGens	Time (s)
Austro-Asiatic	100/90	**0.0135**	26900	439.005
Austronesian	100/80	**0.0148**	26600	285.659
Indo-European	20/80	0.0211	5320	41.544
Pama-Nyungan	80/100	0.1318	21680	435.8
Sino-Tibetan	100/10	**0.0722**	22600	235.774

(b) Results for SCA cognates.

Family	s/T_0	GQD	NGens	Time (s)
Austro-Asiatic	40/60	0.0415	10440	151.561
Austronesian	20/20	0.1022	4780	42.097
Indo-European	80/10	0.0322	18080	190.48
Pama-Nyungan	100/40	0.1647	25300	759.023
Sino-Tibetan	80/20	0.5218	19120	233.173

(c) Results for BipSkip-CC cognates.

Family	s/T_0	GQD	NGens	Time (s)
Austro-Asiatic	80/80	0.0245	21280	310.403
Austronesian	40/10	0.0927	9040	82.443
Indo-European	10/100	0.046	2710	28.691
Pama-Nyungan	80/70	**0.0777**	21120	662.447
Sino-Tibetan	40/80	0.3049	10640	129.903

(d) Results for BipSkip-IM cognates.

Table 4: Results for the MAPLE approach to fast phylogenetic inference for each method. The best step size and initial temperature setting is shown as s/T_0. NGens is the number of generations, Time is the time taken to run the inference in number of seconds on a single core Linux machine.

Family	s/T_0	GQD	NGens	Time (s)
Austro-Asiatic	100/90	0.0058	26900	476.113
Austronesian	80/80	0.0389	21280	123.167
Indo-European	10/10	0.0135	2260	16.713
Pama-Nyungan	100/10	0.061	22600	605.319
Sino-Tibetan	100/50	0.0475	25700	206.952

Table 5: Results for gold standard cognates.

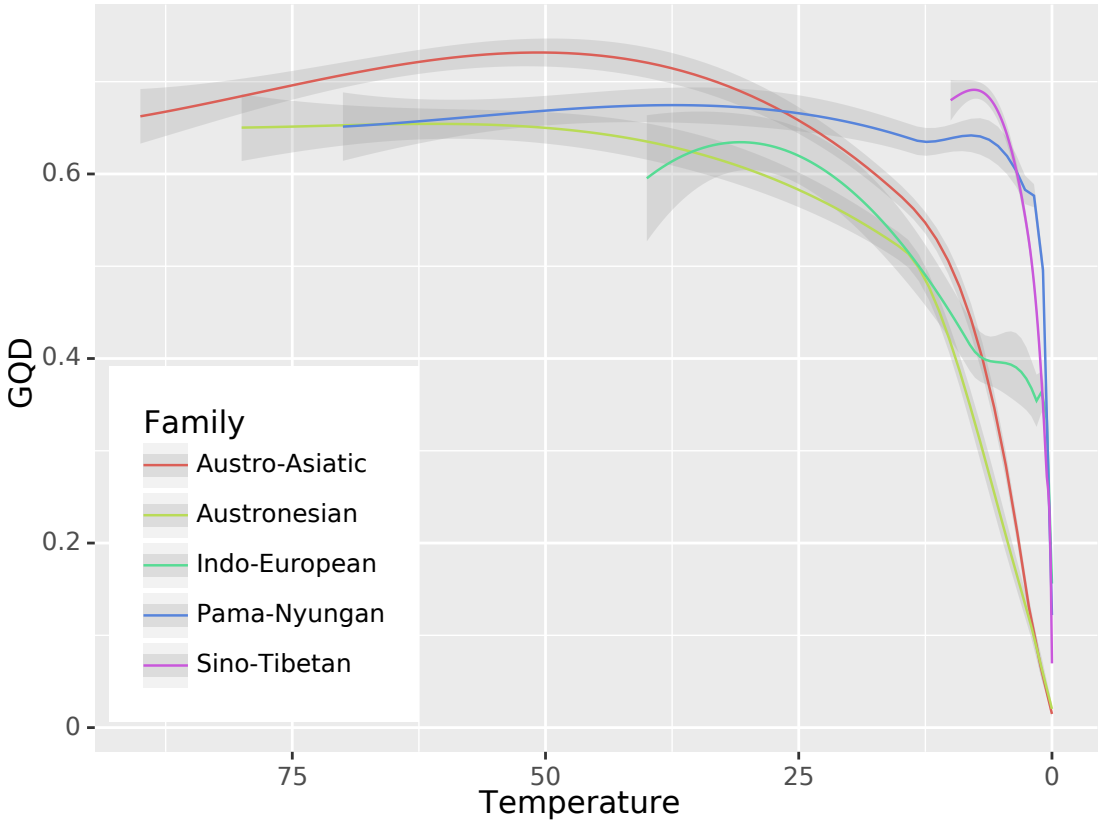

Figure 2: Lineplot of GQD against temperature for all the five different language families. The trendlines are drawn using LOESS smoothing.

lies. Moreover, the curves are monotonically decreasing once the temperature is below 12.

5 Conclusion

In this paper we proposed an automated framework for very fast and still highly reliable phylogenetic reconstruction in historical linguistics. Our framework introduces two new methods. The BipSkip approach uses bipartite networks of sound-class-based skip-grams for the task of automatic cognate detection. The MAPLE approach makes use of simulated annealing technique to infer a MAP tree for linguistic evolution. Both methods are not only very fast, but – as our tests show – also quite accurate in their performance, when compared to similar, much slower, algorithms proposed in the past. In combination, the methods can be used to assess preliminary phylogenies from linguistic datasets of more than 100 languages in less than half an hour on an ordinary single core machine.

We are well aware that our framework is by no means perfect, and that it should be used with a certain amount of care. Our methods are best used for the purpose of exploratory analysis on larger datasets which have so far not yet been thoroughly studied. Here, we believe that the new framework can provide considerable help to future research, specifically also, because it does not not require the technical support of high-end clusters.

Both methods can be further improved in multiple ways. Our cognate detection method's weak performance on South-East Asian languages could be addressed by enabling it to detect partial cognates instead of complete cognates. At the same time, new models, allowing for a consistent handling of multi-state characters and a direct handling of partial cognates, could be added to our fast Bayesian phylogenetic inference approach.

Acknowledgments

We thank the three reviewers for the comments which helped improve the paper. TR took part in the BigMed project (https://bigmed.no/) at University of Oslo when the work was performed. JML's work was supported by the ERC Starting Grant 715618 "Computer-Assisted Language Comparison" (http://calc.digling.org).

References

Gautam Altekar, Sandhya Dwarkadas, John P Huelsenbeck, and Fredrik Ronquist. 2004. Parallel metropolis coupled Markov chain Monte Carlo for Bayesian phylogenetic inference. *Bioinformatics*, 20(3):407–415.

Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. 2009. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12(4):461–486.

Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. 2003. An introduction to MCMC for machine learning. *Machine learning*, 50(1-2):5–43.

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

Remco Bouckaert, Philippe Lemey, Michael Dunn, Simon J. Greenhill, Alexander V. Alekseyenko, Alexei J. Drummond, Russell D. Gray, Marc A. Suchard, and Quentin D. Atkinson. 2012. Mapping the Origins and Expansion of the Indo-European Language Family. *Science*, 337(6097):957–960.

Will Chang, Chundra Cathcart, David Hall, and Andrew Garrett. 2015. Ancestry-constrained phylogenetic analysis supports the Indo-European steppe hypothesis. *Language*, 91(1):194–244.

Chris Christiansen, Thomas Mailund, Christian NS Pedersen, Martin Randers, and Martin Stig Stissing. 2006. Fast calculation of the quartet distance between trees of arbitrary degrees. *Algorithms for Molecular Biology*, 1(1).

Aron B. Dolgopolsky. 1964. Gipoteza drevnejšego rodstva jazykovych semej severnoj evrazii s verojatnostej točky zrenija. *Voprosy Jazykoznanija*, 2:53–63.

Ke-Lin Du and MNS Swamy. 2016. Simulated annealing. In *Search and Optimization by Metaheuristics*, pages 29–36. Springer.

Joseph Felsenstein. 2004. *Inferring phylogenies*. Sinauer Associates, Sunderland, Massachusetts.

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

Simon J. Greenhill, Quentin D. Atkinson, Andrew Meade, and Russell D. Gray. 2010. The shape and tempo of language evolution. *Proceedings of the Royal Society B: Biological Sciences*, 277(1693):2443–2450.

Simon J. Greenhill and Russell D. Gray. 2009. Austronesian language phylogenies: Myths and misconceptions about Bayesian computational methods. *Austronesian Historical Linguistics and Culture History: A Festschrift for Robert Blust*, pages 375–397.

Harald Hammarström, Robert Forkel, and Martin Haspelmath. 2017. *Glottolog*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Bradley Hauer and Grzegorz Kondrak. 2011. Clustering semantically equivalent words into cognate sets in multilingual lists. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 865–873. AFNLP.

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

Anni Järvelin, Antti Järvelin, and Kalervo Järvelin. 2007. s-grams: Defining generalized n-grams for information retrieval. *Information Processing & Management*, 43(4):1005–1019.

Michael I Jordan et al. 2004. Graphical models. *Statistical Science*, 19(1):140–155.

Vishnupriya Kolipakam, Fiona M. Jordan, Michael Dunn, Simon J. Greenhill, Remco Bouckaert, Russell D. Gray, and Annemarie Verkerk. 2018. A Bayesian phylogenetic study of the Dravidian language family. *Royal Society Open Science*, 5:171504.

Clemens Lakner, Paul Van Der Mark, John P Huelsenbeck, Bret Larget, and Fredrik Ronquist. 2008. Efficiency of Markov chain Monte Carlo tree proposals in Bayesian phylogenetics. *Systematic biology*, 57(1):86–103.

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

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

Johann-Mattis List, Simon Greenhill, Tiago Tresoldi, and Robert Forkel. 2018a. *LingPy. A Python library for quantitative tasks in historical linguistics*. Max Planck Institute for the Science of Human History, Jena.

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

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

Johann-Mattis List, Mary Walworth, Simon J. Greenhill, Tiago Tresoldi, and Robert Forkel. 2018b. Sequence comparison in computational historical linguistics. *Journal of Language Evolution*, 3(2):130–144.

David R Maddison, David L Swofford, and Wayne P Maddison. 1997. NEXUS: an extensible file format for systematic information. *Syst. Biol.*, 46(4):590–621.

Mark Pagel and Andrew Meade. 2006. Estimating rates of lexical replacement on phylogenetic trees of languages. In Peter Forster and Colin Renfrew, editors, *Phylogenetic Methods and the Prehistory of Languages*, pages 173–182. McDonald Institute Monographs, Cambridge.

Simone Pompei, Vittorio Loreto, and Francesca Tria. 2011. On the accuracy of language trees. *PloS one*, 6(6):e20109.

Taraka Rama. 2015. Automatic cognate identification with gap-weighted string subsequences. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.*, pages 1227–1231.

Taraka Rama. 2018a. Similarity dependent chinese restaurant process for cognate identification in multilingual wordlists. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 271–281.

Taraka Rama. 2018b. Three tree priors and five datasets. *Language Dynamics and Change*, 8(2):182 – 218.

Taraka Rama, Johann-Mattis List, Johannes Wahle, and Gerhard Jäger. 2018. Are automatic methods for cognate detection good enough for phylogenetic reconstruction in historical linguistics? In *Proceedings of the North American Chapter of the Association of Computational Linguistics*, pages 393–400.

Taraka Rama, Johannes Wahle, Pavel Sofroniev, and Gerhard Jäger. 2017. Fast and unsupervised methods for multilingual cognate clustering. *arXiv preprint arXiv:1702.04938*.

Taraka Rama and Søren Wichmann. 2018. Towards identifying the optimal datasize for lexically-based bayesian inference of linguistic phylogenies. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1578–1590.

Fredrik Ronquist, Maxim Teslenko, Paul van der Mark, Daniel L Ayres, Aaron Darling, Sebastian Höhna, Bret Larget, Liang Liu, Marc A Suchard, and John P Huelsenbeck. 2012. MrBayes 3.2: efficient Bayesian phylogenetic inference and model choice across a large model space. *Systematic biology*, 61(3):539–542.

Martin Rosvall and Carl T Bergstrom. 2008. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123.

Laurent Sagart, Guillaume Jacques, Yunfan Lai, Robin Ryder, Valentin Thouzeau, Simon J. Greenhill, and Johann-Mattis List. 2019. Dated language phylogenies shed light on the ancestry of sino-tibetan. *Proceedings of the National Academy of Science of the United States of America*, 116:1–6.

Robert. R. Sokal and Charles. D. Michener. 1958. A statistical method for evaluating systematic relationships. *University of Kansas Scientific Bulletin*, 28:1409–1438.

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

Søren Wichmann, André Müller, and Viveka Velupillai. 2010. Homelands of the world's language families: A quantitative approach. *Diachronica*, 27(2):247–276.

Ziheng Yang. 1994. Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: Approximate methods. *Journal of Molecular evolution*, 39(3):306–314.

Ziheng Yang. 2014. *Molecular evolution: A statistical approach*. Oxford University Press, Oxford.

Ziheng Yang and Bruce Rannala. 1997. Bayesian phylogenetic inference using DNA sequences: a Markov Chain Monte Carlo method. *Molecular biology and evolution*, 14(7):717–724.

A Supplemental Material

The supplemental material was submitted along with this paper and also uploaded to Zenodo (https://doi.org/10.5281/zenodo.3237508). The packages provide all data needed to replicate the analyses, as well as detailed instructions in how to apply the methods. In the paper, we point to the relevant sections in the supplemental material.