Multiple sequence alignments in linguistics

Jelena Prokić
University of Groningen
The Netherlands
j.prokic@rug.nl

Martijn Wieling
University of Groningen
The Netherlands
m.b.wieling@rug.nl

John Nerbonne
University of Groningen
The Netherlands
j.nerbonne@rug.nl

Abstract

In this study we apply and evaluate an iterative pairwise alignment program for producing multiple sequence alignments, ALPHAMALIG (Alonso et al., 2004), using as material the phonetic transcriptions of words used in Bulgarian dialectological research. To evaluate the quality of the multiple alignment, we propose two new methods based on comparing each column in the obtained alignments with the corresponding column in a set of gold standard alignments. Our results show that the alignments produced by ALPHAMALIG correspond well with the gold standard alignments, making this algorithm suitable for the automatic generation of multiple string alignments. Multiple string alignment is particularly interesting for historical reconstruction based on sound correspondences.

1 Introduction

Our cultural heritage is studied today not only in museums, libraries, archives and their digital portals, but also through the genetic and cultural lineaments of living populations. Linguists, population geneticists, archaeologists, and physical and cultural anthropologists are all active in researching cultural heritage on the basis of material that may or may not be part of official cultural heritage archives. The common task is that of understanding the histories of the peoples of the world, especially their migrations and contacts. To research and understand linguistic cultural heritage we require instruments which are sensitive to its signals, and, in particular sensitive to signals of common provenance. The present paper focuses on pronunciation habits which have been recognized to bear signals of common provenance for over two hundred years (since the work of Sir William Jones).

We present work in a research line which seeks to submit pronunciation data to phylogenetic analysis (Gray and Atkinson, 2003) and which requires an alignment of the (phonological) segments of cognate words. We focus in this paper on evaluating the quality of multi-aligned pronunciations.

In bioinformatics, sequence alignment is a way of arranging DNA, RNA or protein sequences in order to identify regions of similarity and determine evolutionary, functional or structural similarity between the sequences. There are two main types of string alignment: pairwise and multiple string alignment. Pairwise string alignment methods compare two strings at a time and cannot directly be used to obtain multiple string alignment methods (Gusfield, 1997, 343-344). In multiple string alignment all strings are aligned and compared at the same time, making it a good technique for discovering patterns, especially those that are weakly preserved and cannot be detected easily from sets of pairwise alignments. Multiple string comparison is considered to be *the holy grail* of molecular biology (Gusfield, 1997, 332):

> It is the most critical cutting-edge tool for *extracting and representing* biologically important, yet faint or widely dispersed, commonalities from a set of strings.

Multiple string comparison is not new in linguistic research. In the late 19th century the Neogrammarians proposed the hypothesis of the regularity of sound change. According to THE NEOGRAMMARIAN HYPOTHESIS sound change occurs regularly and uniformly whenever the appropriate phonetic environment is encountered (Campbell, 2004). Ever since, the understanding of sound change has played a major role in the comparative method that is itself based on the simultaneous comparison of different languages, i.e. lists of cognate terms from the related languages. The correct analysis of sound changes

Proceedings of the EACL 2009 Workshop on Language Technology and Resources for Cultural Heritage, Social Sciences, Humanities, and Education –LaTeCH – SHELT&R 2009, pages 18–25,
Athens, Greece, 30 March 2009. ©2009 Association for Computational Linguistics

requires the simultaneous examination of corresponding sounds in order to compare hypotheses about their evolution. Alignment identifies which sounds correspond. Historical linguists align the sequences manually, while we seek to automate this process.

In recent years there has been a strong focus in historical linguistics on the introduction of quantitative methods in order to develop tools for the comparison and classification of languages. For example, in his PhD thesis, Kondrak (2002) presents algorithms for the reconstruction of proto-languages from cognates. Warnow et al. (2006) applied methods taken from phylogenetics on Indo-European phonetic data in order to model language evolution. Heeringa and Joseph (2007) applied the Levensthein algorithm to the Dutch pronunciation data taken from *Reeks Nederlandse Dialectatlassen* and tried to reconstruct a 'proto-language' of Dutch dialects using the pairwise alignments.

Studies in historical linguistics and dialectometry where string comparison is used as a basis for calculating the distances between language varieties will profit from tools to multi-align strings automatically and to calculate the distances between them. Good multiple alignment is of benefit to all those methods in diachronic linguistics such as the comparative reconstruction method or the so-called CHARACTER-BASED METHODS taken from phylogenetics, which have also been successfully applied in linguistics (Gray and Jordan, 2000; Gray and Atkinson, 2003; Atkinson et al., 2005; Warnow et al., 2006). The multi-alignment systems can help historical linguistics by reducing the human labor needed to detect the regular sound correspondences and cognate pairs of words. They also systematize the linguistic knowledge in intuitive alignments, and provide a basis for the application of the quantitative methods that lead to a better understanding of language variation and language change.

In this study we apply an iterative pairwise alignment program for linguistics, ALPHAMALIG, on phonetic transcriptions of words used in dialectological research. We automatically multi-align all transcriptions and compare these generated alignments with manually aligned gold standard alignments. At the same time we propose two methods for the evaluation of the multiple sequence alignments (MSA).

The structure of this paper is as follows. An example of a multiple alignment and a discussion of the advantages over pairwise alignment is given in the next section, after which we discuss our data set in section 3. Section 4 explains the iterative pairwise alignment algorithm and the program ALPHAMALIG. Section 5 discusses the gold standard and two baselines, while section 6 discusses the novel evaluation procedures. The results are given in section 7 and we end this paper with a discussion in section 8.

2 Example of Multiple Sequence Alignment

In this section we will give an example of the automatically multi-aligned strings from our data set and point out some important features of the simultaneous comparison of more than two strings.

village1	j	ˈɑ	-	-	-	-
village2	j	ˈɑ	z	e	-	-
village3	-	ˈɑ	s	-	-	-
village4	j	ˈɑ	s	-	-	-
village5	j	ˈɑ	z	e	k	a
village6	j	ˈɛ	-	-	-	-
village7	-	ˈɒ	s	-	-	-

Figure 1: Example of multiple string alignment

In Figure 1 we have multi-aligned pronunciations of the word *az* 'I' automatically generated by ALPHAMALIG. The advantages of this kind of alignment over pairwise alignment are twofold:

- First, it is easier to detect and process corresponding phones in words and their alternations (like [ˈɑ] and [ˈɛ] and [ˈɒ] in the second column in Figure 1).
- Second, the distances/similarities between strings can be different in pairwise comparison as opposed to multiple comparison. This is so because multi-aligned strings, unlike pairwise aligned strings, contain information on the positions where phones were inserted or deleted in both strings. For example, in Figure 1 the pairwise alignment of the pronunciations from village 1 and village 3 would be:

village1	j	ˈɑ	-
village3	-	ˈɑ	s

These two alignments have one matching element out of three in total, which means that the similarity between them is $1/3 = 0.33$. At the same time the similarity between these two strings calculated based on the multi-aligned strings in Figure 1 would be $4/6 = 0.66$. The measurement based on multi-alignment takes the common missing material into account as well.

3 Data set

The data set used in this paper consists of phonetic transcriptions of 152 words collected from 197 sites evenly distributed all over Bulgaria. It is part of the project *Buldialect—Measuring linguistic unity and diversity in Europe*.[1] Pronunciations of almost all words were collected from all the sites and for some words there are multiple pronunciations per site. Phonetic transcriptions include various diacritics and suprasegmentals, making the total number of unique characters (types) in the data set 98.[2]

4 Iterative pairwise alignment

Multiple alignment algorithms iteratively merge two multiple alignments of two subsets of strings into a single multiple alignment that is union of those subsets (Gusfield, 1997). The simplest approach is to align the two strings that have the minimum distance over all pairs of strings and iteratively align strings having the smallest distance to the already aligned strings in order to generate a new multiple alignment. Other algorithms use different initializations and different criteria in selecting the new alignments to merge. Some begin with the longest (low cost) alignment instead of the least cost absolutely. A string with the smallest edit distance to any of the already merged strings is chosen to be added to the strings in the multiple alignment. In choosing the pair with the minimal distance, all algorithms are greedy, and risk missing optimal alignments.

ALPHAMALIG is an iterative pairwise alignment program for bilingual text alignment. It uses the strategy of merging multiple alignments of subsets of strings, instead of adding just one string at the time to the already aligned strings.[3] It was originally developed to align corresponding words in bilingual texts, i.e. with textual data, but it functions with any data that can be represented as a sequence of symbols of a finite alphabet. In addition to the input sequences, the program needs to know the alphabet and the distances between each token pair and each pair consisting of a token and a gap.

In order to perform multiple sequence alignments of X-SAMPA word transcriptions we modified ALPHAMALIG slightly so it could work with the tokens that consist of more than one symbol, such as ["e], ["e:] and [t_S]. The distances between the tokens were specified in such a way that vowels can be aligned only with vowels and consonants only with consonants. The same tokens are treated as identical and the distance between them is set to 0. The distance between any token in the data set to a gap symbol has the same cost as replacing a vowel with a vowel or a consonant with a consonant. Except for this very general linguistic knowledge, no other data-specific information was given to the program. In this research we do not use any phonetic features in order to define the segments more precisely and to calculate the distances between them in a more sensitive way than just making a binary 'match/does-not-match-distinction', since we want to keep the system language independent and robust to the highest possible degree.

5 Gold standard and baseline

In order to evaluate the performance of ALPHAMALIG, we compared the alignments obtained using this program to the manually aligned strings, our gold standard, and to the alignments obtained using two very simple techniques that are described next: simple baseline and advanced baseline.

5.1 Simple baseline

The simplest way of aligning two strings would be to align the first element from one string with the first element from the other string, the second element with the second and so on. If two strings are not of equal length, the remaining unaligned tokens are aligned with the gap symbol which rep-

[1] The project is sponsored by Volkswagen Stiftung. More information can be found at http://sfs.uni-tuebingen.de/dialectometry.

[2] The data is publicly available and can be found at http://www.bultreebank.org/BulDialects/index.html.

[3] More information on ALPHAMALIG can be found at http://alggen.lsi.upc.es/recerca/align/alphamalig/intro-alphamalig.html.

resents an insertion or a deletion. This is the alignment implicit in Hamming distance, which ignores insertions and deletions.

By applying this simple method, we obtained multiple sequence alignments for all words in our data set. An example of such a multiple sequence alignment is shown in Figure 2. These alignments were used to check how difficult the multiple sequence alignment task is for our data and how much improvement is obtained using more advanced techniques to multi-align strings.

j	ˈɑ	-	-
j	ˈɑ	z	e
ˈɑ	ʃ	-	-

Figure 2: Simple baseline

5.2 Advanced baseline

Our second baseline is more advanced than the first and was created using the following procedure:

1. for each word the longest string among all pronunciations is located
2. all strings are pairwise aligned against the longest string using the Levensthein algorithm (Heeringa, 2004). We refer to both sequences in a pairwise alignment as ALIGNMENT LINES. Note that alignment lines include hyphens indicating the places of insertions and deletions.
3. the alignment lines—all of equal length—are extracted
4. all extracted alignment lines are placed below each other to form the multiple alignment

An example of combining pairwise alignments against the longest string (in this case [jˈaze]) is shown in Figure 3.

5.3 Gold standard

Our gold standard was created by manually correcting the advanced baseline alignments described in the previous section. The gold standard results and both baseline results consist of 152 files with multi-aligned strings, one for each word. The pronunciations are ordered alphabetically according to the village they come from. If there are more pronunciations per site, they are all present, one under the other.

j	ˈɑ	z	e
j	ˈɑ	-	-

j	ˈɑ	z	e
-	ˈɑ	ʃ	-

j	ˈɑ	-	-
j	ˈɑ	z	e
-	ˈɑ	ʃ	-

Figure 3: Advanced baseline. The top two alignments each contain two alignment lines, and the bottom one contains three.

6 Evaluation

Although multiple sequence alignments are broadly used in molecular biology, there is still no widely accepted objective function for evaluating the goodness of the multiple aligned strings (Gusfield, 1997). The quality of the existing methods used to produce multiple sequence alignments is judged by the 'biological meaning of the alignments they produce'. Since strings in linguistics cannot be judged by the biological criteria used in string evaluation in biology, we were forced to propose evaluation methods that would be suitable for the strings in question. One of the advantages we had was the existence of the gold standard alignments, which made our task easier and more straightforward—in order to determine the quality of the multi-aligned strings, we compare outputs of the different algorithms to the gold standard. Since there is no off-the-shelf method that can be used for comparison of multi-aligned strings to a gold standard, we propose two novel methods—one sensitive to the order of columns in two alignments and another that takes into account only the content of each column.

6.1 Column dependent method

The first method we developed compares the contents of the columns and also takes the column sequence into account. The column dependent evaluation (CDE) procedure is as follows:

- Each gold standard column is compared to the most similar column out of two neighboring columns of a candidate multiple alignment. The two neighboring columns depend on the previous matched column j and have indices $j + 1$ and $j + 2$ (at the start $j = 0$). It is possible that there are columns in the candidate multiple alignment which remain unmatched, as well as columns at the end of the gold standard which remain unmatched.

- The similarity of a candidate column with a gold standard column is calculated by dividing the number of correctly placed elements in every candidate column by the total number of elements in the column. A score of 1 indicates perfect overlap, while a score of 0 indicates the columns have no elements in common.

- The similarity score of the whole multiple alignment (for a single word) is calculated by summing the similarity score of each candidate column and dividing it by the total number of matched columns plus the total number of unmatched columns in both multiple alignments.

- The final similarity score between the set of gold standard alignments with the set of candidate multiple alignments is calculated by averaging the multiple alignment similarity scores for all strings.

As an example consider the multiple alignments in Figure 4, with the gold standard alignment (GS) on the left and the generated alignment (GA) on the right.

w	rʲ	ˈɛ	m	e	w	-	rʲ	ˈɛ	m	e
v	r	ˈe	m	i	v	-	r	ˈe	m	i
u	rʲ	ˈe	m	i	-	u	rʲ	ˈe	m	i
v	rʲ	ˈe	m	i	v	-	rʲ	ˈe	m	i

Figure 4: GS and ALPHAMALIG multiple string alignments, the gold standard alignment left, the ALPHAMALIG output right.

The evaluation starts by comparing the first column of the GS with the first and second column of the GA. The first column of the GA is the best match, since the similarity score between the first columns is 0.75 (3 out of 4 elements match). In similar fashion, the second column of the GS is compared with the second and the third column of the GA and matched with the third column of GA with a similarity score of 1 (all elements match). The third GS column is matched with the fourth GA column, the fourth GS column with the fifth GA column and the fifth GS column with the sixth GA column (all three having a similarity score of 1). As a consequence, the second column of the GA remains unmatched. In total, five columns are matched and one column remains unmatched. The total score of the GA equals:

$$\frac{(0.75+1+1+1+1)}{(5+1)} = 0.792$$

It is clear that this method punishes unmatched columns by increasing the value of the denominator in the similarity score calculation. As a consequence, swapped columns are punished severely, which is illustrated in Figure 5.

ˈo	rʲ	ə	j	-	ˈo	rʲ	ə	-	j
ˈo	rʲ	ə	-	u	ˈo	rʲ	ə	u	-
ˈo	rʲ	ə	f	-	ˈo	rʲ	ə	-	f

Figure 5: Two alignments with swapped columns

In the alignments in Figure 5, the first three columns of GS would be matched with the first three columns of GA with a score of 1, the fourth would be matched with the fifth, and two columns would be left unmatched: the fifth GS column and the fourth GA column yielding a total similarity score of $4/6 = 0.66$. Especially in this case this is undesirable, as both sequences of these columns represent equally reasonable multiple alignment and should have a total similarity score of 1. We therefore need a less strict evaluation method which does not insist on the exact ordering. An alternative method is introduced and discussed in the following section.

6.2 Modified Rand Index

In developing an alternative evaluation we proceeded from the insight that the columns of a multiple alignment are a sort of PARTITION of the elements of the alignment strings, i.e., they constitute a set of disjoint multi-sets whose union is the entire multi-set of segments in the multiple alignment. Each column effectively assigns its segments to a partition, which clearly cannot overlap with the elements of another column (partition). Since every segment must fall within some column, the assignment is also exhaustive.

Our second evaluation method is therefore based on the modified Rand index (Hubert and Arabie, 1985). The modified Rand index is used in classification for comparing two different partitions of a finite set of objects. It is based on the Rand index (Rand, 1971), one of the most popular measures for comparing the degree to which partitions agree (in classification).

Given a set of n elements $S = o_1, ...o_n$ and two partitions of S, U and V, the Rand index R is defined as:

$$R = \frac{a+b}{a+b+c+d}$$

where:

- a: the number of pairs of elements in S that are in the same set (column) in U and in the same set in V

- b: the number of pairs of elements in S that are in different sets (columns) in U and in different sets in V

- c: the number of pairs of elements in S that are in the same set in U and in different sets in V

- d: the number of pairs of elements in S that are in different sets in U and in the same set in V

Consequently, a and b are the number of pairs of elements on which two classifications agree, while c and d are the number of pairs of elements on which they disagree. In our case classifications agree about concrete segment tokens only in the cases where they appear in the same columns in the alignments.

The value of Rand index ranges between 0 and 1, with 0 indicating that the two partitions (multi-alignments) do not agree on any pair of points and 1 indicating that the data partitions are exactly the same.[4] A problem with the Rand index is that it does not return a constant value (zero) if two partitions are picked at random. Hubert and Arabie (1985) suggested a modification of the Rand index (MRI) that corrects this property. It can be expressed in the general form as:

$$\text{MRI} = \frac{\text{Rand index} - \text{Expected index}}{\text{Maximum index} - \text{Expected index}}$$

The expected index is the expected number of pairs which would be placed in the same set in U and in the same set in V by chance. The maximum index represents the maximum number of objects that can be put in the same set in U and in the same set in V. The MRI value ranges between -1 and 1, with perfect overlap being indicated by 1 and values ≤ 0 indicating no overlap. For a more detailed explanation of the modified Rand index, please refer to Hubert and Arabie (1985).

We would like to emphasize that it is clear that the set of columns of a multi-alignment have more structure than a partition *sec*, in particular because the columns (subpartitions) are ordered, unlike the subpartitions in a partition. But we shall compensate for this difference by explicitly marking order.

ˈo [1]	rʲ [2]	ə [3]	j [4]	-
ˈo [5]	rʲ [6]	ə [7]	-	u [8]
ˈo [9]	rʲ [10]	ə [11]	f [12]	-

Figure 6: Annotated alignment

In our study, each segment token in each transcription was treated as a different object (see Figure 6), and every column was taken to be a subpartition to which segment tokens are assigned. Both alignments in Figure 5 have 12 phones that are put into 5 groups. We "tag" each token sequentially in order to distinguish the different tokens of a single segment from each other, but note that the way we do this also introduces an order sensitivity in the measure. The two partitions obtained are:

GS1 = {1,5,9}	GA1 = {1,5,9}
GS2 = {2,6,10}	GA2 = {2,6,10}
GS3 = {3,7,11}	GA3 = {3,7,11}
GS4 = {4,12}	GA4 = {8}
GS5 = {8}	GA5 = {4,12}

Using the modified Rand index the quality of each column is checked, regardless of whether the columns are in order. The MRI for the alignments in Figure 5 will be 1, because both alignments group segment tokens in the same way. Even though columns four and five are swapped, in both classifications phones [j] and [f] are grouped together, while sound [u] forms a separate group.

The MRI itself only takes into account the quality of each column separately since it simply checks whether the same elements are together in the candidate alignment as in the gold-standard alignment. It is therefore insensitive to the ordering of columns. While it may have seemed counterintuitive linguistically to proceed from an order-insensitive measure, the comparison of "tagged tokens" described above effectively reintroduces order sensitivity.

In the next section we describe the results of applying both evaluation methods on the automatically generated multiple alignments.

[4] In dialectometry, this index was used by Heeringa et al. (2002) to validate dialect clustering methods.

7 Results

After comparing all files of the baseline algorithms and ALPHAMALIG against the gold standard files according to the column dependent evaluation method and the modified Rand index, the average score is calculated by summing up all scores and dividing them by the number of word files (152).

The results are given in Table 1 and also include the number of words with perfect multi-alignments (i.e. identical to the gold standard). Using CDE, ALPHAMALIG scored 0.932 out of 1.0 with 103 perfectly aligned files. The result for the simple baseline was 0.710 with 44 perfectly aligned files. As expected, the result for the advanced baseline was in between these two results—0.869 with 72 files that were completely identical to the GS files. Using MRI to evaluate the alignments generated we obtained generally higher scores for all three algorithms, but with the same ordering. ALPHAMALIG scored 0.982, with 104 perfectly aligned files. The advanced baseline had a lower score of 0.937 and 74 perfect alignments. The simple baseline performed worse, scoring 0.848 and having 44 perfectly aligned files.

The scores of the CDE evaluation method are lower than the MRI scores, which is due to the first method's problematic sensitivity to column ordering in the alignments. It is clear that in both evaluation methods ALPHAMALIG outperforms both baseline alignments by a wide margin.

It is important to notice that the scores for the simple baseline are reasonably high, which can be explained by the structure of our data set. The variation of word pronunciations is relatively small, making string alignment easier. However, ALPHAMALIG obtained much higher scores using both evaluation methods.

Additional qualitative error analysis reveals that the errors of ALPHAMALIG are mostly caused by the vowel-vowel consonant-consonant alignment restriction. In the data set there are 21 files that contain metathesis. Since vowel-consonant alignments were not allowed in ALPHAMALIG, alignments produced by this algorithm were different from the gold standard, as illustrated in Figure 7.

The vowel-consonant restriction is also responsible for wrong alignments in some words where metathesis is not present, but where the vowel-consonant alignment is still preferred over aligning vowels and/or consonants with a gap (see for example Figure 4).

v	l	ˈɣ	k		v	l	ˈɣ	-	k
v	ˈɣ	l	k		v	-	ˈɣ	l	k

Figure 7: Two alignments with metathesis

The other type of error present in the ALPHAMALIG alignments is caused by the fact that all vowel-vowel and consonant-consonant distances receive the same weight. In Figure 8 the alignment of word *bjahme* 'were' produced by ALPHAMALIG is wrong because instead of aligning [m^j] with [m] and [m] it is wrongly aligned with [x] and [x], while [x] is aligned with [ʃ] instead of aligning it with [x] and [x].

b	ˈɛ	**ʃ**	u	**x**	-	m	e	-
b^j	ˈɑ	**-**	-	**x**	-	m	i	-
b	ˈe	**x**	-	**m**^{j}	-	-	ɣ	-

Figure 8: Alignment error produced by ALPHAMALIG

8 Discussion and future work

In this study we presented a first attempt to automatically multi-align phonetic transcriptions. The algorithm we used to generate alignments has been shown to be very reliable, produce alignments of good quality, with less than 2% error at the segment level. In this study we used only very simple linguistic knowledge in order to align strings. The only restriction we imposed was that a vowel should only be aligned with a vowel and a consonant only with a consonant. The system has shown to be very robust and to produce good quality alignments with a very limited information on the distances between the tokens. However, in the future we would like to apply this algorithm using more detailed segment distances, so that we can work without vowel-consonant restrictions. Using more detailed language specific feature system for each phone, we believe we may be able to improve the produced alignments further. This especially holds for the type of errors illustrated in Figure 8 where it is clear that [m^j] is phonetically closer to [m] than to [x] sound.

As our data set was relatively simple (indicated by the reasonable performance of our simple baseline algorithm), we would very much like to evaluate ALPHAMALIG against a more complex data

	CDE	CDE perfect columns	MRI	MRI perfect columns
Simple baseline	0.710	44	0.848	44
Advanced baseline	0.869	72	0.937	74
ALPHAMALIG	**0.932**	**103**	**0.982**	**104**

Table 1: Results of evaluating outputs of the different algorithms against the GS

set and try to replicate the good results we obtained here. On one hand, high performance of both baseline algorithms show that our task was relatively easy. On the other hand, achieving perfect alignments will be very difficult, if possible at all.

Additionally, we proposed two methods to evaluate multiple aligned strings in linguistic research. Although these systems could be improved, both of them are giving a good estimation of the quality of the generated alignments. For the examined data, we find MRI to be better evaluation technique since it overcomes the problem of swapped columns.

In this research we tested and evaluated ALPHAMALIG on the dialect phonetic data. However, multiple sequence alignments can be also applied on the sequences of sentences and paragraphs. This makes multiple sequence alignment algorithm a powerful tool for mining text data in social sciences, humanities and education.

Acknowledgements

We are thankful to Xavier Messeguer of the Technical University of Catalonia who kindly supplied us with the source code of ALPHAMALIG. We also thank Therese Leinonen and Sebastian Kürschner of the University of Groningen and Esteve Valls i Alecha of the University of Barcelona for their useful feedback on our ideas.

References

Laura Alonso, Irene Castellon, Jordi Escribano, Xavier Messeguer, and Lluis Padro. 2004. Multiple Sequence Alignment for characterizing the linear structure of revision. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*.

Quentin Atkinson, Geoff Nicholls, David Welch, and Russell Gray. 2005. From words to dates: water into wine, mathemagic or phylogenetic inference. *Transcriptions of the Philological Society*, 103:193–219.

Lyle Campbell. 2004. *Historical Linguistics: An Introduction*. Edinburgh University Press, second edition.

Russel D. Gray and Quentin D. Atkinson. 2003. Language-tree divergence times support the Anatolian theory of Indo-European origin. *Nature*, 426:435–339.

Russel D. Gray and Fiona M. Jordan. 2000. Language trees support the express-train sequence of Austronesian expansion. *Nature*, 405:1052–1055.

Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*. Cambridge University Press.

Wilbert Heeringa and Brian Joseph. 2007. The relative divergence of Dutch dialect pronunciations from their common source: An exploratory study. In John Nerbonne, T. Mark Ellison, and Grzegorz Kondrak, editors, *Proceedings of the Ninth Meeting of the ACL Special Interest Group in Computational Morphology and Phonology*.

Wilbert Heeringa, John Nerbonne, and Peter Kleiweg. 2002. Validating dialect comparison methods. In Wolfgang Gaul and Gunter Ritter, editors, *Classification, Automation, and New Media. Proceedings of the 24th Annual Conference of the Gesellschaft für Klassifikation e. V., University of Passau, March 15-17, 2000*, pages 445–452. Springer, Berlin, Heidelberg and New York.

Wilbert Heeringa. 2004. *Measuring Dialect Pronunciation Differences using Levenshtein Distance*. Ph.D. thesis, Rijksuniversiteit Groningen.

Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of Classification*, 2:193–218.

Grzegorz Kondrak. 2002. *Algorithms for Language Reconstruction*. PhD Thesis, University of Toronto.

William M. Rand. 1971. Objective criteria for the evaluation of clustering methods. *Journal of American Statistical Association*, 66(336):846–850, December.

Tandy Warnow, Steven N. Evans, Donald Ringe, and Luay Nakhleh. 2006. A stochastic model of language evolution that incorporates homoplasy and borrowing. In Peter Forster and Colin Renfrew, editors, *Phylogenetic Methods and the Prehistory of Languages*. MacDonald Institute for Archaeological Research, Cambridge.