Quantitative Analyses of Typological Data

Von der Fakultät für Mathematik und Informatik
der Universität Leipzig
angenommene

DISSERTATION

zur Erlangung des akademischen Grades

DOCTOR RERUM NATURALIUM
(Dr. rer. nat.)

im Fachgebiet

INFORMATIK

vorgelegt von

Mihai Albu

geboren am 06.09.1976 in Bucharest
Romania

Acknowledgements

First of all, I would like to thank my supervisors, Peter Stadler and Michael Cysouw, for the amazing support and availability, and for sharing their scientific expertise with me. Michael Cysouw provided my research with deep knowledge of typological analyses, with insights in the problems of evaluating linguistic data, as well as with rigorous suggestions for interpretations of the methods and the results. Peter Stadler combined the excitement and enthusiasm of scientific research. His group meetings and workshops gave me a pleasant feeling of enjoying the research activity, as well as appreciating his valuable advice. I would also like to thank the members of the Linguistic Department in the Max Planck Institute for Evolutionary Anthropology and the Interdisciplinary Centre for Bioinformatics, for both the cordial environments as well as the academic discussions they offered me.

Special thanks go to Andreas Dress, who inspired me to begin the research activity, and sustained it during the last 5 years with valuable advice and suggestions.

This thesis is dedicated to my son Gabriel and my wife Silvia, as a small thanks for their understanding and acceptance of me being away for so long.

Thank you

Abstract

The current computational era heralds a multitude of challenges for linguists, mathematicians and computer scientists alike. The investigation of linguistic feature consistencies, their implications in historical linguistics and the detection of possible geographical and phylogenetical influences in feature behaviors will steadily unveil the importance of each language characteristic and its associated historical role in language evolution. The wish to develop new methods has made the integration of various disciplines necessary. Consequently, the role of computer science (and more specifically bioinformatics) has risen in the study of linguistic processes. By combining the strengths of bioinformatics algorithms, mathematical procedures, statistical knowledge and databases developing methodologies, various strategies have been developed and applied to the phylogenetic reconstruction of language evolution, but up till now only on small scales. The analysis of a worldwide typological database has many potential applications for the study of language universals, feature consistencies, and of phylogenetic implications of linguistic differences. In this research I highlight recent progress of strategies related to typological data, on the basis of the data offered by the World Atlas of Language Structures (WALS, [41]).

This thesis attempts to find suitable analyses for dealing with linguistic typological data. These methods are focused on discovering "good" data for phylogenetic reconstruction algorithms, detecting (inter-) dependencies between the characteristics of languages, and on performing multiple statistical methods for selecting the data that correlates best with either genealogical or geographical relationships.

While the aim of qualitative analysis is a complete, detailed description of the data, and no attempt is made to assign frequencies to the linguistic features which are identified in the data, in quantitative research one classifies features, counts

them, and even constructs statistical models in an attempt to explain what is observed. Findings can be generalized to a larger population, and direct comparisons can be made between two data sets, so long as valid sampling and significance techniques have been used. Thus, quantitative analysis allows the discovering of which phenomena are likely to be genuine reflections of the behavior of a set of languages or varieties, and which are merely chance occurrences. However, the picture of the data which emerges from quantitative analysis is often less precise than those obtained from qualitative analysis. In quantitative analysis, an item either belongs to class X or it does not. In some cases, quantitative analysis is therefore an idealization of the data. Also, quantitative analysis tends to sideline rare occurrences. To ensure that certain statistical tests (such as the chi-square test) provide reliable results, it is essential that minimum frequencies are obtained - meaning that categories may have to be collapsed together, therefore resulting in a loss of data richness.

This research will concentrate on quantitative analysis, by selecting appropriate measurements in order to detect relevant phylogenetic information. I tried to discover consistent features with the world-wide distribution of languages presented in the WALS, as well as phylogenetic informative ones. Using such characteristics in future phylogenetic analyses would hopefully enhance the results.

This thesis is structured as follows. Chapter 1 introduces the basic computational/bioinformatics and linguistic background. The WALS data set is presented in Chapter 2 together with the coding problems found. Chapter 3 gives detailed information about various methodologies for improving the distance measurements, while the content of Chapter 4 describes my approach on detecting good "phylogenetic" features. In Chapter 5 new phylogenetic reconstruction methods are specified as well as a new approach to detect the fitness of a data set to a phylogenetic tree. In Chapter 6 I specify the QALD software of the thesis, describing its utility and functionality. This software can be used to analyze various languages, features, and the according results of the methods presented in this thesis. The conclusions of this research are summarized in Chapter 7.

All research ultimately has a qualitative grounding
(Donald Campbell)
There's no such thing as qualitative data. Everything is either 1 or 0
(Fred Kerlinger)

Contents

Chapter 1

Introduction

1.1 Linguistic background

1.1.1 Typology

1.1.1.1 Definition of linguistic typology

Linguistic typology (in this work I will henceforth simply refer to 'typology'), represents the attempt to classify languages into types. This definition of typology implies that the research has to discover shared patterns across languages. Based on such shared patterns, a language is considered as belonging to a specific type, while a typology of languages is a classification of languages into those types. These definitions already suggest a close connection to a cross linguistic comparison. As presented in [17], typology can be classified into two main kinds, according to their object of study:

- holistic typology: classification of whole languages into types and subtypes on the basis of shared structural characteristics (not to be confused with genealogical or areal classification);
- partial typology: classification of specific structural features across languages (e.g. word order or case marking).

Holistic typology implies that there are various ways to classify languages. The kind of classification depends on the intended purpose and on the linguistic elements that are analyzed:

- genealogical: classification into families descended from a common ancestor, e.g. Sino-Tibetan languages, Austronesian languages;
- areal: classification by geographical region, e.g. the languages of Southeast Asia;
- typological: classification by shared structural features, e.g. tone languages, SVO languages, ergative languages.

These classifications are, in principle, independent of each other: Chinese is an SVO language, Tibetan an SOV language, while languages belonging to the Indo-European family may be VSO (Welsh), SVO (Italian) or SOV (Armenian). But in practice, the classifications are not entirely independent (for historical reasons): most languages of the Austronesian family are VSO or SVO (due to shared inheritance), while most languages of mainland Southeast Asia are tonal (due to areal diffusion). Nevertheless, I am interested in genealogical language classification using typological data and, but I acknowledge the importance of the geographical influences.

1.1.1.2 The history of genealogical classification and linguistic typology

Of the two subjects, linguistic typology and genealogical language classification, the latter has the much older tradition. Its beginning can be traced back to early 13th-century attempts concerned with establishing genealogical relationships among languages [13]. The short *Diatriba de Europaeorum linguae*, written in 1599 by Joseph Justus Scaliger [68], is one of the most cited examples of an early attempt of genealogical classification of languages. Scaliger established - apart from Greek which was frequently seen as the source language of Latin - the Romance, Germanic, and Slavic language families, on the basis of shared vocabulary items among the languages belonging to the particular group, e.g. the word for 'god' (*deus*, *god*, and *bog*, respectively). Nevertheless, there was no hint in his work that these three families in turn were related. However, the Czech Sigismund Gemenius (1497-1554) had shown in a comparative dictionary, several generations before Scaliger, that Greek, Latin, Germanic and Slavic were genetically related, cf. [34].

Like much older scientific work in the study of languages, linguistic typology - though essentially developed during the 19th century - appears to have forerunners, as first attempts can be traced far back to the 16th century. For instance Coseriu [16], makes references to a 40-page essay on "compounded" and "original" languages from 1761 by the political economist Adam Smith (1723-1790) as a source for Friedrich Schlegel's typological classifications. Schlegel's brother

August Whilhelm (1767-1845) referred to Smith in his own effort at language typology in 1818. August Wilhelm, in his 1818 monograph, gives full and exclusive credit to his brother's exposé, at the same time introducing a further subgrouping on the bases of an "analytical/synthetic" distinction. Others [47, 61] have pointed to Gabriel Girard's (1677-1748) distinction between "languages analogues" (i.e., those which have a fixed word order, like French) and "languages transpositives" (i.e., those which have a flexible word order, like Latin) as the immediate source of Smith's "compounded/uncompounded" dichotomy, without proving that Girard had any influence on either of the Schlegel brothers. However, even if the Schlegel brothers were not the inventors of language typology, but synthesizers of proposals by their predecessors or contemporaries, it remains safe to say that the beginning of a "scientific" attempt at language classification on the basis of morphological structure has its origin in their work.

1.1.1.3 Typology classification. Implicational universals

Historically, the first manifestation of typology in modern linguistics is typology classification, i.e. the process of describing the various linguistic types found across languages. The linguistic types (or strategies) are the linguistic structures that are found across languages based on an external definition of a category. Mainly, patterns are defined and then found in various languages, being the starting point in typological classification. Starting with Greenberg [38], a more reductionistic approach has been developed in which only parts of linguistic structure are classified. Some samples of these types, like word order, positions of the Genitive, etc. can be found in [64].

The next obvious question is whether the resulting typologies (classifications) of different characteristics (domains) correlate with each other or not. Whether these correlations are directly dependent on genealogical or geographical patterns is an open question that this research tries to answer. The goal of such research is to uncover regularities or even universals of linguistic structure. An often used analysis is the so-called *implicational universal*.

An implicational universal states a dependency between two logically independent parameters, e.g. one may state that if a language has a type X, then it will also have type Y. The implicational universal then become a major tool for expressing typological generalisations. However, as Cysouw describes in [20], there is a major problem with this generalisation, mainly that a frequency in a sample that appears to be remarkably high or low does not necessarily mean anything. This is due to the fact that the saliency of a frequency in a typological sample depends on the deviation from the statistical expectation, not on the absolute number of occurrences.

1.1.1.4 Areal-linguistic implications for typology

Areal patterns have strong implications for work in typology and universals, though usually unrecognized in the earlier linguistic literature. A self-explanatory example is the case of word - order typology. It has been argued that certain word-order types come into existence in languages only through areal influence and borrowing [15]. Both [38] and [42] dealt with the 24 possible basic word-order types, of which only 15 were thought to be actually represented by existing languages. However, it turns out that for some of these 15 types, and for certain others for which representative languages were subsequently discovered, all the exemplifying languages owe crucial aspects of their basic word-order to areal borrowing; this applies to Greenberg's types 7, 18, 19 and 20. Type 7 (Verb-first/Postpositional - Noun/Genitive - Noun/Adjective - Noun) is represented only by Zoque (a Mixe-Zoquean language of southern Mexico); Zoque borrowed VOS word order from neighbouring Mayan languages, creating its odd-type 7 combination. Type 18 (SOV/Prepositional/Noun - Genitive/Adjective - Noun), not previously recognized to have exemplifying languages, is represented by Tigre (Ethiopian Semitic), which is like non-Ethiopian Semitic languages, except for SOV, which is acquired from Cushitic. Type 19 and 20 have also been proven to be areal influenced, e.g. by samples of Amharic borrowing from Cushitic, and Northern Tajik borrowing from Turkic, respectively.

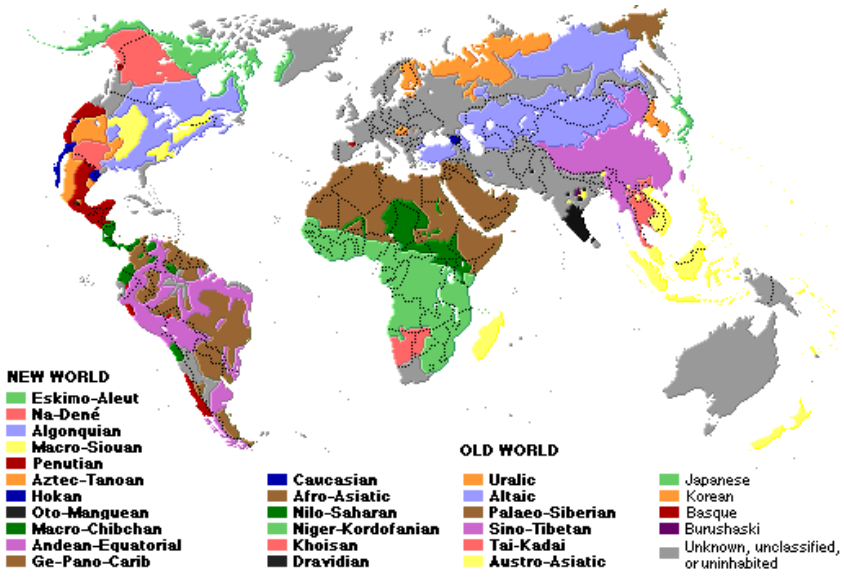


Figure 1.1: The geography of the main language families in the world.

Both typology and areal linguistics are important tools for historical reconstruction in linguistics. Typology helps us to understand expected changes and

constraints on possible changes, and thus is very important for reconstruction. Areal linguistics helps to recover aspects of linguistics history that are due to diffusion and convergence (Figure 1.1). However, the two are alike in one respect: both can hinder linguistic reconstruction. In the comparative method, corresponding forms shared among related languages are the basis for postulating ancestral forms in the proto-language. Nevertheless, undetected areal borrowings can exhibit similarities, seemingly corresponding forms, that sometimes are assumed to be the results of common inheritance, and are erroneously reconstructed as features of the parent-language [33].

1.1.2 Previous work

1.1.2.1 Typological and lexical databases

Over the last decades, typological databases started to gain an important role, but they are mostly subject oriented. WALS, on the other way, does not concentrate on a specific geographical location or on specific linguistic fields. Two important problems are usually still found in the typological databases [9] :

- Typological databases typically rely on a static and pre-defined category list which tends to conflict with the data as more languages are entered. Further, this restricts the database to research that is completely sanctioned by the category list.

- Typological databases are typically integrated into a single file containing a wide variety of information making it difficult (if not impossible) to re-use any part of this information in other databases or to search for typological correlations across databases.

The need for resolving these disadvantages suggests that more research activity is needed from computer science experts and database scientists.

Over the last few decades, a large amount of new lexical resources has arisen: machine-readable dictionaries, lexical databases, full-form lexicons, morphological databases, semantic networks, dictionary databases, etc. Most of these lexical systems have been modelled after lexicographic sources. A lexical database is a lexical resource system meant primarily for computational exploitation. This can be used in a search engine providing human users with lexical information, but also in NLP (Natural Language Processing) applications, computer - aided language - learning systems, computer - aided linguistic research, etc. There is still an open problem of how to compare word lists and make them interoperable.

1.1.2.2 The trend in current typology

An important part of my research was conducted in the spirit of trying to detect possible correlations with either geographical or genealogical patterns, as well as to detect which of the data points are appropriate for each kind of correlation.

Since the late 1980s, and most prominently through the work of Dryer [28, 29] and Nichols [58], it has become clear that hardly any typological variable is evenly distributed in the world. Most distributions are subject to non-accidental geographical skewing. For example, testing the hypothesis that verb-final or free word order correlates with dependent-marking in transitive subjects (*A*) or objects (*P*) against a genealogically balanced sample from AUTOTYP [9] and WALS [8, 41] presented a careful and thorough analysis. There is a significant association (Fisher Exact p = .014, N = 179), if one examines the entire data together. But if a closer look is taken at the distribution of the available data, continent-by-continent, it turns out that only in Eurasia is the association significant. Everywhere else the distribution can be predicted from marginal frequencies of the two variables. A large number of such examples can easily be found, and they underline Dryer's warning [28] that, unless geographical factors are controlled for, a statistical association does not support a hypothesis of universal preference. This is not surprisingly, as it is known that large areas like Eurasia have an intricate history of type spread [45, 58], and in general, the history of language contact and population movements substantially affects typological distributions.

Clearly many current typological distributions can only be understood as the result of actual (pre-)history, both locally and globally. Findings from anthropological and historical disciplines might provide valuable information about historical signals. Nevertheless, the most plausible available explanations of statistically significant macro-areas, such as those around the Pacific, or those covering Eurasia [10, 11, 58, 59, 60] suggest that they are the surviving traces of distributions that were formed at early periods of large scale population movement and language spreads.

But, as argued by Maslova [57], the distributions determine the threshold above which one accepts universals that are due to the nature of language rather than to the nature of human population history: an association of variables must not only be statistically significant in a representative sample and independent of known geographical and genealogical affiliation [28, 63], but it must also be shown to be independent of earlier (or even initial) stages at which there could have been significant skewing at work. In other words, associations can be taken to reflect strictly linguistic universals only if they can be shown to be sufficiently instable historically so that one can assume a stationary distribution for the current situation. This again requires a fundamentally diachronic understanding of what causes typological distributions, viz. different type shift probabilities [8].

Large datasets almost invariably reveal exceptions to universals, and this, together with a substantial increase of newly described languages and assisted by prominent conceptual argumentation [18, 27], have practically eliminated notions of absolute universals and impossibilities. Modern studies of typological distributions involve statistical methods, from association tests (cf. Cysouw in press, for recent review [21]) to multivariate scaling methods [19, 52]. Regarding the studies on areal data, typology has seen the introduction of new mathematical methods (e.g. the Isopleth Method: [73]), and current attempts to integrate Geographical Information Systems bring bright hope for progress in this domain.

One common property of all these methods is that they work with independently and narrowly defined variables, instead of the gross types (active language, agglutinative language) of classical holistic typology, or categorical notions of a so called Sprachbund. The general assumption is that if there are large-scale connections between linguistic structures, or between linguistic structures and geography, they consist in probabilistic (and therefore exception-ridden) correlations between independently measured variables; they are not expected to follow from absolutely defined or ideal types. In a similar vein, modern typology has moved away from analyzing entire languages and instead takes individual structural patterns (constructions, rules, constraints etc.) as objects of study. Linguistic diversity is captured by large sets of fine-grained variables, not by grand type notions.

The analysis of such variables poses statistical problems shared by other historical population sciences – most prominently, one has access to only less than 1% of all languages that have ever been spoken by our species, and so the current population with all its historically-grown distributional biases will always be overrepresented in our samples. Moreover, in typological sampling, one typically attempts exhaustive and well-balanced coverage of known genealogical diversity, so that signals of universal preference or areal population history are not disturbed by relatively recent inheritance effects. In response to these problems, typologists are now adopting Monte-Carlo-like methods, and first steps have also been undertaken toward randomization-based reliability tests on coding [46]. Unlike classical distribution-based methods, these methods do not support statistical inference to an underlying population of all human languages. All statistical inference is limited to the current sample at hand. Modern typology is a discipline that develops variables for capturing cross-linguistic similarities and differences (qualitative typology), explores universal and local skewings in the distribution of these variables (quantitative typology) and proposes theories that explain the skewings (theoretical typology). The ultimate goal is to understand *what is*, *where*, and *why*, and this makes it clear that major contributions that typology offers are not confined to Cognitive Science as narrowly understood. The goals of the 21st century typology are embedded in a much broader anthropological perspective: to help understand how the variants of one key social institution are distributed in the

world, and what general principles and what incidental events are the historical causes for these distributions.

1.1.2.3 Phylogenetic attempts using linguistic data

The word-list analyses became an increased research activity, varying from comparison of word-lists to dictionaries, texts, or books. While Cysouw performed analyses of biblical fragments (the choice of the material being obvious from the variety and consistency point of view), other are just using the Swadesh word list [72], even if it was extremely controversially. Nevertheless, many attempts are starting with this list that it is filled for a various number of languages, and then different comparative methods are used.

A comparative analysis of manuscripts copies was performed during the workshop in Louvain-de-Neuve, where a dataset consisting in codings of the differences between different copies were given to the participants and they were challenged to try to reconstruct the history of the manuscript. The whole experiment was using not a real historical evolution, but a *simulated* one, meaning that scholars with different background and different native languages were asked to copy a two page *manuscript*. The *mistakes* and *errors* are believed to be the same as the real history a few centuries ago. Nevertheless, the important conclusion of the meeting was that the manuscript histories can be very similar with the biological evolution (e.g., correspondence with the lateral gene transfers process, evolution 'mistakes', evolution improvements). As [4] showed, a simple use of 'modern' phylogenetic methods (Figure 1.2) provided a very close result to the real history of the manuscript (Figure 1.3), even if according to the general opinion the main problem was that one of the first copier was not a native speaker of the manuscript's language, and this might induce disturbance in those analysis. The real history of the copiers was the following: the initial manuscript was given to 3 copiers (**g**, **a** and **f**), from which **a** was not a native speaker. My results failed to place **a** as a common ancestor of **c** and **d**, and also missed the **g**, **h** and **i** history, placing **i** as the predecessor copier (the organizer's explanation was that the **g** copier was a really 'bad' one, and this might explain my results).

In Nature, R. Gray and Q. Atkinson [37] published a rigorous analysis of word lists for the Indo-European family, concluding not only in the phylogenetic relationships, but also in detecting the time depth of languages splits (Figure 1.4).

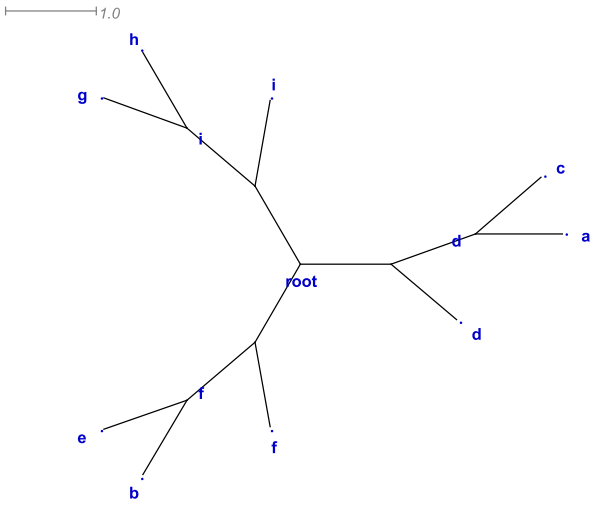


Figure 1.2: NNet result using Hamming distance.

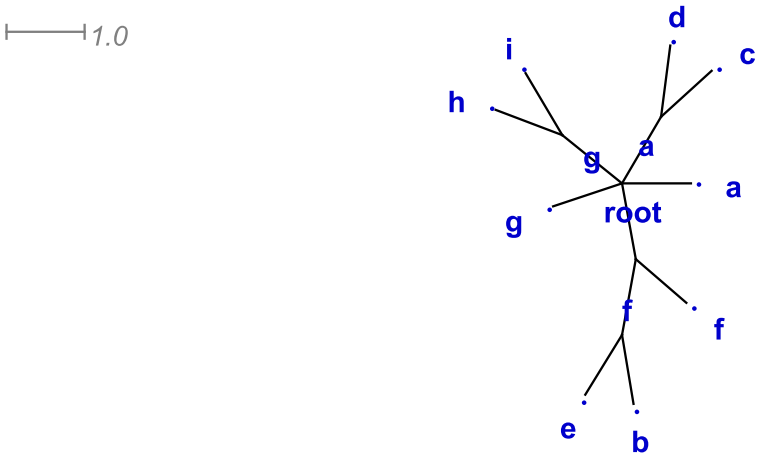


Figure 1.3: The real history of manuscripts.

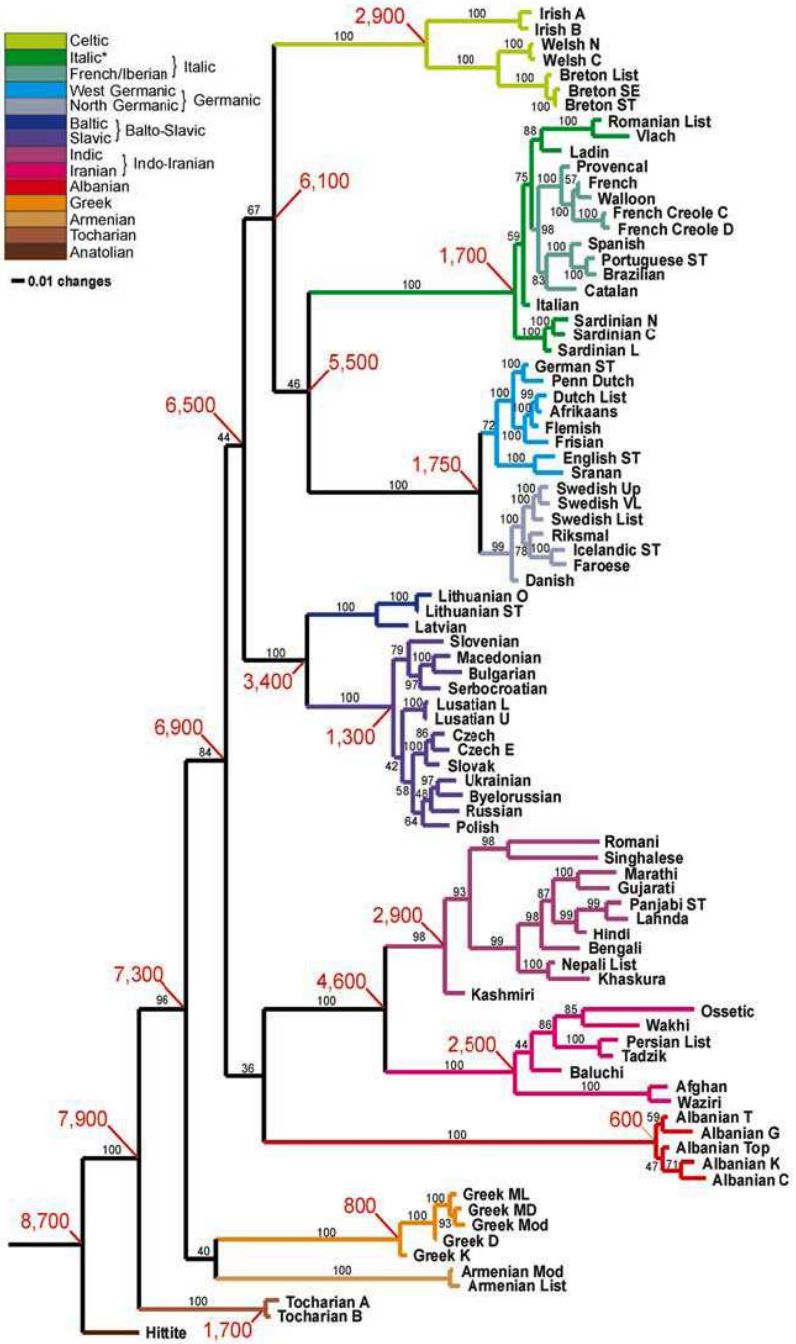


Figure 1.4: Results from [37] with time-depth estimation. Consensus tree and divergence-time estimates. Majority-rule consensus tree based on the MCMC (Markov Chain Monte Carlo) sample of 1,000 trees.

1.2 Phylogenetic background

1.2.1 Computer science and biology

Previously bioinformatics was defined as an interdisciplinary field involving biology, computer science, mathematics and statistics to analyze biological sequence data, genome content, and arrangements to predict the function and structure of macromolecules. With the advent of the genome era, bioinformatics now plays added roles in biological and medical research and accounts for an increasing number of publications each year [55]. Can bioinformatics tools also help the historical linguists? Is there any similarity between biological data and linguistic data? Do I need new tools for analyzing linguistics data, can I use the already recognized bioinformatics ones, or should I carefully modify them in order to be field appropriate? This research tries to give an answer to these questions with the hope of being a starting point for future research.

Clearly, in the WALS case I do not have to worry about aligning sequences (data), as the WALS dataset can be easily seen as a rectangular table with languages on rows and features on columns (I should specify that the aligning sequences problems can be found in linguistic analyses, for example when comparing word lists). But I am dealing with similar problems as in biology, like missing data (gaps), like methods of building distance matrices or like phylogenetic reconstruction issues. Even worse, I have no information on feature values interdependencies, nor other properties that bioinformatics tools account for , e.g. physical and chemical properties of amino acids [7]. Therefore I am mainly interested in methods that analyze sequences and that can be applied to the WALS data set. While in bioinformatics the databases evolved exponentially, being today an enormous interconnected amount of information, the scale of linguistics databases is still small. Nevertheless, using the already recognized scientific methods from bioinformatics and statistics, I hope for a faster research development, and I am positive that this work will simplify future analyses.

1.2.1.1 Trees, distance matrices, alignments, networks

Common bioinformatics terminology will often be used in this research, so, in the following, short descriptions of the most important terms are presented.

Phylogenetic trees. A *tree* is a mathematical structure which is used to model the evolutionary history of a group of objects (sequences, organisms, languages). This actual pattern of historical relationships is the **phylogeny** or **evolutionary tree** which one tries to estimate [62]. A tree consists of **nodes** connected by **branches** (or **edges**) (Figure 1.5) **Terminal nodes** (also called leaves, OTUs [Operational Taxonomic Units] or terminal taxa) represent the objects of interest for

which one has data, and they can be either extant or extinct. **Internal nodes** represent hypothetical ancestors, and the one that comprises the whole tree is the **root** of the tree.

Distance Matrices. Measures of OTU's dissimilarities may be used to estimate the number of evolutionary changes that occurred between two OTUs since they last shared a common ancestor. These measures quantify the evolutionary distance between the two OTUs. Trees themselves can also be represented by distances, and this link has motivated a range of tree-building methods that seek to convert pairwise distances between objects into evolutionary trees. A *perfect* distance measure must satisfy some basic requirements: it must be *ultrametric* (Equation 1.4). Nevertheless, real complex data never produces such a distance measure, no matter which method is used to obtain it.

Let $D(a, b)$ be the distance between two OTUs a and b. The following properties (Equations 1.1 - 1.5):

$$D(a, b) \geq 0 \tag{1.1}$$

$$D(a, b) = D(b, a) \tag{1.2}$$

$$D(a, c) \leq D(a, b) + D(b, c) \tag{1.3}$$

$$D(a, c) \leq max(D(a, b), D(b, c)) \tag{1.4}$$

$$a = b \Rightarrow D(a, b) = 0 \tag{1.5}$$

are necessary for a tree-like matrix, e.g. a distance matrix that can be represented by a tree, with the length of the edges corresponding to the distances between OTUs. A *non tree-like* distance matrix is depicted in Table 1.1, as the condition

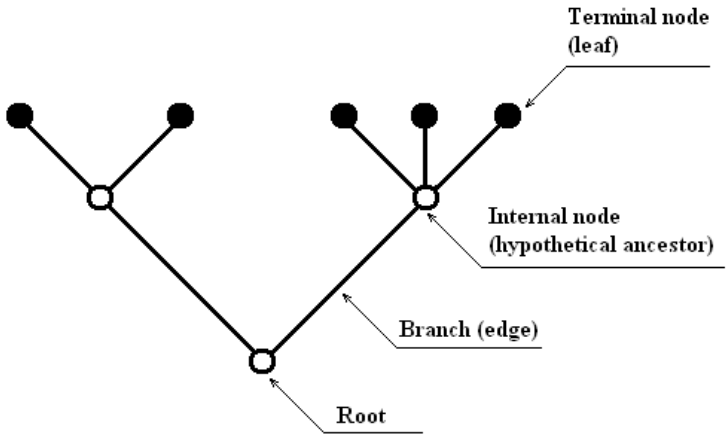


Figure 1.5: Tree elements.

1.4 is not fulfilled.

Alignments. Alignments are used in order to: (1) organise data to reflect sequence homology, (2) estimate evolutionary distance, (3) infer phylogenetic trees from homologous sites, (4) highlight conserved sites/regions, (5) highlight variable sites/regions, (6) uncover changes in gene structure, (7) discover for evidence of selection, etc.

Networks. As stated above, real data can not produce metric and additive distance matrices. Most of the phylogenetic reconstruction algorithms use any kind of distance matrix and try to build the phylogenetic tree, but this clearly implies that some tie-decisions have to be taken, or some information must be disregarded. Clearly, if one has the distance measures between four OTUs as in shown Table 1.1, a simple tree-like diagram can not correctly represent this distance measure, as the ultrametric condition is not fulfilled (Figure 1.6). Split networks are used to

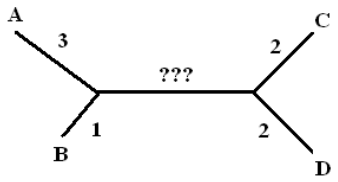


Figure 1.6: Incompatible tree for data in Table 1.1.

represent incompatible and ambiguous signals in a data set. In such a network parallel edges, rather than single branches, are used to represent the splits computed from the data. To be able to accommodate incompatible splits, it is often necessary that a split network contains nodes that do not represent ancestral species. Thus, split networks provide only an implicit representation of evolutionary history. Therefore, it will be easy to use the network representations to represent the distance matrix shown in Table 1.1 as it is depicted in Figure 1.7.

Networks are not only more *flexible* in accepting problematic data sets but, by their constructions and representations, they offer valuable information about

DistMatr	A	B	C	D
A	0	4	5	7
B	4	0	7	5
C	5	7	0	4
D	7	5	4	0

Table 1.1: A non-tree like distance matrix

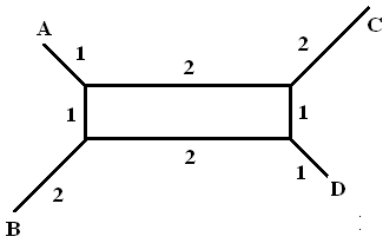


Figure 1.7: Compatible network for data in Table 1.1.

the relationships between the objects analyzed. Figure 1.7 can be interpreted as follows: there is a greater support (evidence) for the split (A, B) vs (C, D) than for the split (A, C) vs (B, D), as the length (weight) of the edge that separates (A, B) and (C, D) is bigger than the one that separates (A, C) and (B, D).

1.2.2 Computer science and linguistics

Phylogenetic analyses of linguistic data have become a usual research in detecting languages genealogy. Even if the data analyzed consists in word lists, or sets of language features, recent methods provided new outcomes in linguistic phylogeny. The [37] paper dates the initial divergence of the Indo-European language family back to 8700 years ago, with Hittite as the first language to split off. This they take to support the theory that Indo-European originated in Anatolia and that Indo-European languages arrived in Europe with the spread of agriculture. They take this to argue against the alternative "Kurgan hypothesis", according to which the "Kurgan Culture" of the steppes was Indo-European speaking, though they say that it is consistent with the view that the Kurgan people represented a branch of Indo-European.

While this paper rigorously analyzed the Indo-European family, proposing an intelligent method for building distance measurements from word lists, Dunn et al [30] suggested another type of analysis, based on binary grammatical features, that were a posteriori weighted after applying a maximum-likelihood approach to the data.

To address the problem of detecting deep signal, Dunn et al. borrowed two tools from their colleagues in biology. First, they constructed a database of 125 structural features for 16 Austronesian and 15 Papuan languages. This enabled them to avoid the charge that they merely selected a few features that happened to fit their hypotheses. The number of possible family trees of descents for an even quite small numbers of languages is vast. Dunn et al.'s second methodological borrowing from biology was the use of a computer program to find the set of optimal trees for the Austronesian and Papuan data sets. To test whether the struc-

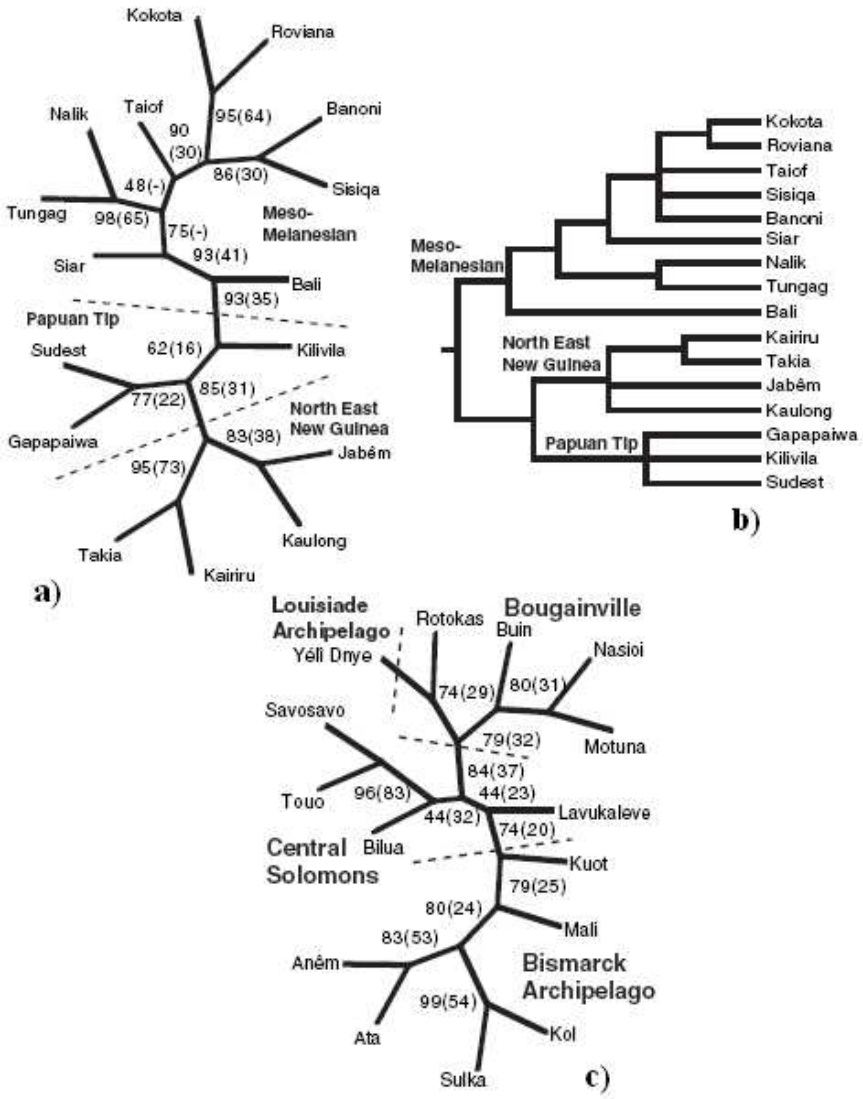


Figure 1.8: The results from [30]. a) Reconstructed phylogeny of the languages of the Meso-Melanesian, Papuan Tip, and North New Guinea groups based on the linguistic comparative method [56, 66]. b) Unrooted parsimony tree showing relationships among the Meso-Melanesian and Papuan Tip groups based on grammatical traits only (that is, discarding abundant lexical evidence) (the figure shows reweighted and raw bootstrap values). c)Maximum parsimony tree of Island Melanesian Papuan languages with reweighted and raw bootstrap values.

tural features contain a historical signal, Dunn et al. compared the Austronesian structure tree with the traditional classification of these languages. The resulting Austronesian structure tree matched the traditional classification quite well, which suggests that the structural features contained some historical link or signal for at least the 4000-year time depth that the Austronesian of languages studied by Dunn et al. are thought to have.

The task of making accurate inferences about our past is a demanding one that requires the integration and triangulation of inferences from genetic, linguistic, and archaeological data [49]. The approach of Dunn et al. is an important step forward in this interdisciplinary endeavor and sets new standards for the systematic collection and analysis of structural features.

Both papers, equally successful and controversial, produced an important step in phylogenetic analyses of languages. Nevertheless, in both cases only a small sample of languages were used, belonging, more or less, to the same family of languages, and there was no inconvenience in, for example, determining the sample range when you are dealing with variate data sets (from the genealogical point of view). In general, in order to have a reliable method, one must use various datasets, that incorporate both genealogical and geographical variance, a well accepted rigorously method and a scientifically recognized algorithm for displaying the results. This is exactly what my research tries to achieve.

1.2.3 Software, availability and conclusions

Standard bioinformatics tools are various and their availability is usually free. One can use web versions or stand-alone versions. As the problems that are attempted to be solved are so different, there are specific software programs for each/multiple issue(s).

There are programs for dealing with: multiple alignment (BLAST), phylogenetic reconstruction (Phylip, SplitsTree), database and database interrogations (JJCB, etc), converting types of input data for phylogenetic analyses (i.e. converting alignments to distance matrices, converting distance matrices to quartet representations, converting alignments to splits systems, etc). My research focused on adjusting the right method of converting the alignments to distance matrices, considering that I am dealing with linguistic data, or discovering the appropriate method to obtain a ‘good’ linguistic distance matrix, while for phylogenetic representation of the objects, I used the NNet algorithm implemented in SplitsTree4 [43], because I needed as much information as possible out of the WALS database. Nevertheless, I tried to compare the phylogenetic algorithms (cf. Section 5.1.1, and Section 5.1.2) with the neighbour-joining method [67], and the analyses of the results can be performed in the QALD software (cf. Chapter 6).

Chapter 2

WALS dataset

2.1 Introduction

WALS (World Atlas of Language Structures, [41]) represents the first attempt of a research program to map a large number of linguistic features (141 different features, grouped mainly in categories of phonology, morphology, nominal categories, nominal syntax, verbal categories, word order, simple clauses, complex sentences, lexicon) found in a large number of languages (2560). The work of Nichols [58] is the only comparable published work in terms of aims and scope, and that deals with 174 languages and 10 multivalued linguistic features, all morphological. WALS significantly increases the degree to which linguists are exposed to typological mapping.

WALS is the largest database of structural properties of languages gathered from descriptive materials (such as reference grammars) by a team of more than 40 authors (many of them leading authorities on the subject). Some of these languages (265 in number) appear on only one map, while some, such as English, appear on most of the maps.

Each map (sample depicted in Figure 2.1) shows between 112 (map number 123) and 1370 (map number 83) languages, each language being represented by a dot, and different colored dots showing different values for the features. Altogether 2650 languages are shown on the maps and more than 58000 dots give information on features in particular languages. The World Atlas of Language Structures thus gives information on the structural diversity of the world's languages available to a large audience, including interested non-linguists as well as

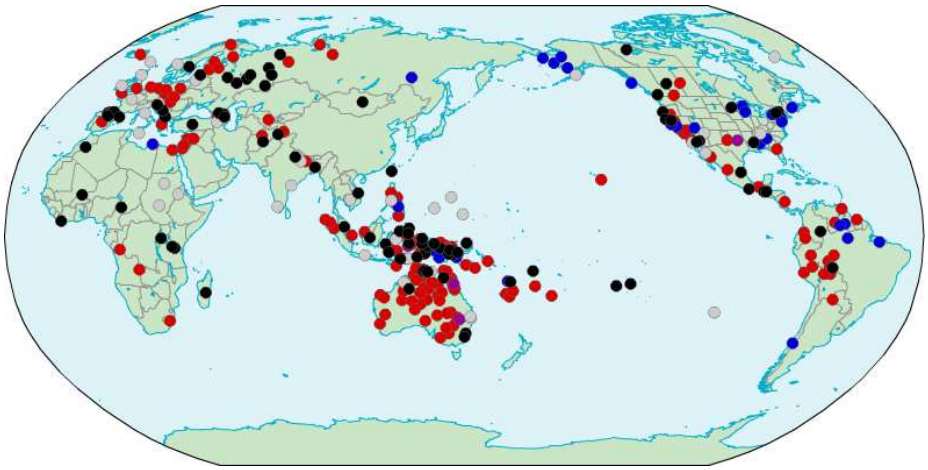


Figure 2.1: WALS sample map. Map number 17 *Rhythm Types*. Coding colours: red = trohaic, blue = iambic, lilac = dual, grey = undetermined, black = no rhythmic stress.

linguists who would not normally read grammars of exotic languages or specialized works by comparative linguists.

The original data was converted in a MySQL [1] database for easier data manipulation. The new MySQL database offers the possibility of 'in-hand' data extractions, visualisations and modifications. For this, SQL (Structured Query Language) statements, more or less embedded were used, and a software program was implemented in order to have a fast, safe and reliable interaction with WALS data. The main part of the database used in the analyses was the 'x_lang_feat' table, that contained the actual data points defined for languages and features (around 58200 records). Other tables needed in various analysis were $genus$, $language$, and $family$, in order to have an overview of the language distribution over the world and/or over the language distribution within families/genus classifications. The main problem, often mentioned in this work, is the poor data coverage:
2560 languages $\times$ 141 features = 360960 data points (theoretical) but, I have only 58200, that means that the actually available data is $= 58200/360960 = 18\%$ (Figure 2.2).

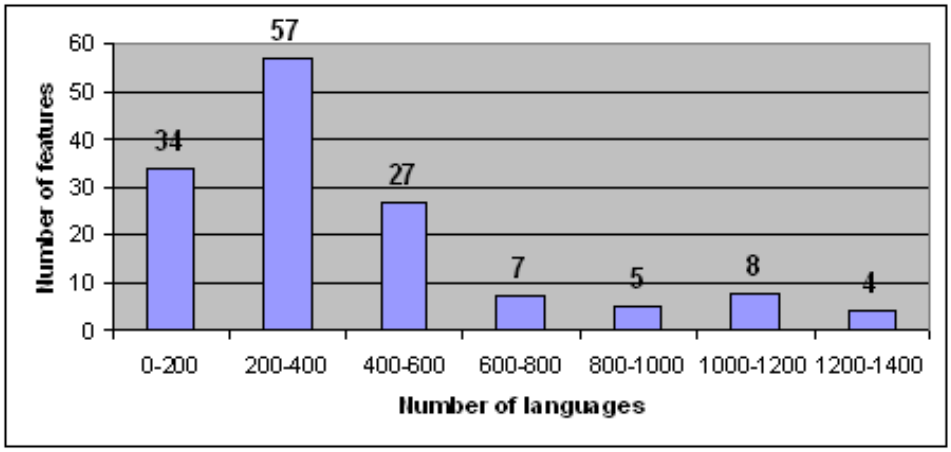


Figure 2.2: WALS distribution of data points.

2.2 Coding structure of the WALS dataset

2.2.1 Can WALS map to a linguistic ontology?

Two broad questions being addressed by this part of the project [1] are:

- What conceptual and design problems need to be solved in order to build an ontology internal to WALS which may allow for a high degree of interoperability among the WALS features?
- How can the WALS categories be related to a general ontology?

In this section, I discuss some of the general challenges raised by WALS as I have tried to determine how specific WALS concepts should be linked to concepts in the GOLD ontology (General Ontology for Linguistic Description, [31]). These can be placed into three broad classes:

- non-encoded internal structure of features;
- non-canonical concepts;
- lateral relationships holding among concepts across different WALS features.

[1] A full description of this analysis can be found in [23].

2.2.1.1 Non-encoded internal structure

The data available in WALS is an enormously valuable source of information for linguistic research. However, in its current form it cannot be used for certain computational and statistical approaches to language typology. The problem is that the database does not encode logical dependencies between concepts referred to by terms found in the databases. Such implicit dependencies can be found both within the values for a single typological feature in WALS and among values found in different features.

To illustrate this problem, it is first useful to consider a case where the values for a given typological feature have no logical dependencies. For the typological feature voicing in plosives and fricatives (map number 4), Figure 2.3 reflects the feature values distributions.

These four possibilities clearly represent the intersection of two independent dimensions: voicing in plosives and voicing in fricatives. There is no a priori reason why these two characteristics should show any dependency on each other – that is, there is nothing about the definitions of plosive, fricative, and voicing, which would imply that there should be any correlation between plosive voicing and fricative voicing in the worlds languages. The fact that there is an apparent correlation between the two parameters in the data (Fishers Exact p = .000037 when counting genera) is an empirical observation of potential interest. However, Dryers test [29] shows only marginal significance in three out of six geographic macro-areas (Africa, Australia/New Guinea, and South America), indicating that the overall significance is not a world-wide effect, but only regionally important. These sorts of correlations are potentially interesting ,but they are only linguistically meaningful if one knows that the relevant parameters are logically independent of each other.

Unlike the data represented in Figure 2.3, the values for the data represented in Figure 2.4 show a high degree of logical interdependence. For example, a language missing both /p/ and /g/ is also a language missing /p/, but the database does not encode this. Similarly, a language missing no sounds in /p t k b d g/ cannot be a language missing /p/, missing /g/, or having both missing. From the perspective of a human user, these logical dependencies are obvious. However, a computational algorithm designed to discover correlations among values in the various features will find spurious patterns without an explicit machine-readable encoding of such dependencies.

The logical dependencies holding among the values for the feature voicing and gaps in plosive systems are depicted in the tree in Figure 2.5. Figure 2.5 of course, includes possible typological feature values not found in the data represented in Figure 2.4, and it also includes a number of higher-level categories. The sort of information represented in Figure 2.5 can be easily expressed using an ontology.

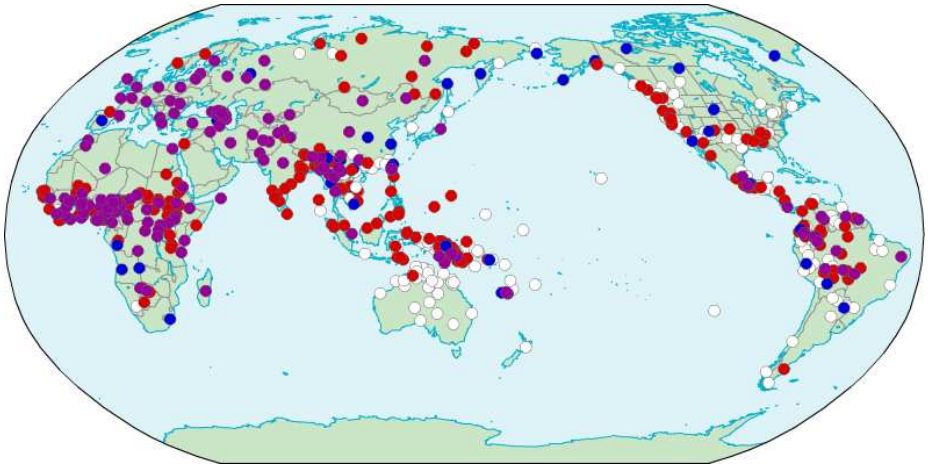


Figure 2.3: Map number 4. *Voicing in Plosives and Fricatives*. No voicing contrast in plosives and fricatives (white), voicing contrast in plosives alone (red), voicing contrast in fricatives alone (blue), voicing contrast in both plosives and fricatives (lilac)

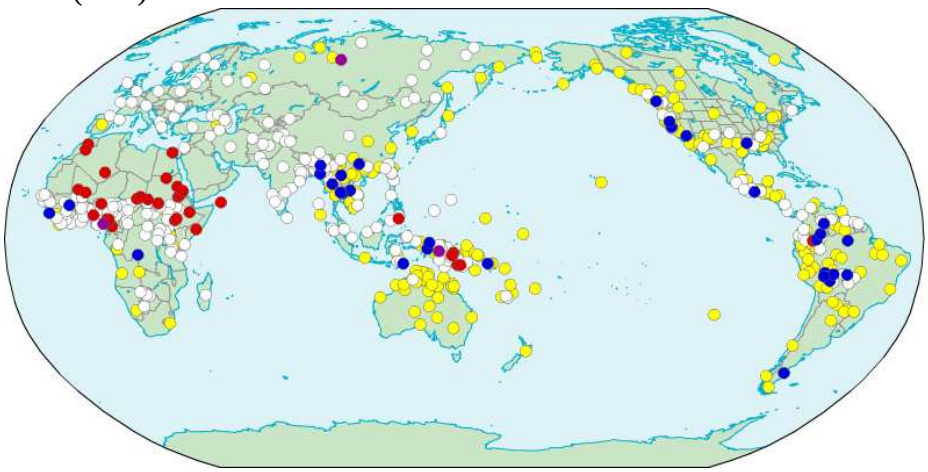


Figure 2.4: Man Number 5. *Voicing and Gaps in Plosive Systems*. Other = yellow, none missing in /p t k b d g/ = white, missing /p/ = red, missing /g/ = blue, missing both = lilac.

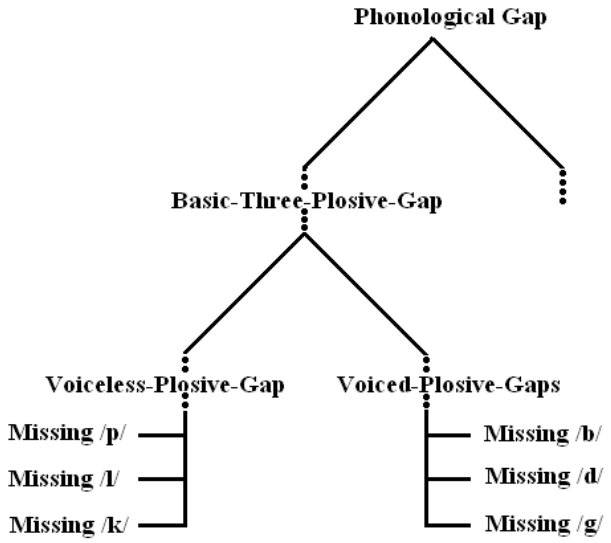


Figure 2.5: Logical Structure of *Voicing and Gaps in Plosive Systems*.

An important part of the WALS ontology project is to enumerate the logical dependencies holding among the concepts found in WALS and build appropriate ontological resources for encoding them. Of the problems the WALS ontology project has encountered with respect to linking WALS concepts to a general ontology, implicit logical dependencies have required the greatest deal of human labor. However, from an ontological perspective, they are relatively easy to deal with.

2.2.1.2 Non-canonical concepts

As a resource designed for use in language typology instead of use in individual language description, WALS makes use of many concepts which are quite distinct from the concepts found in a typical grammar or annotated text. Three such classes concepts seem worthy of mention here:

- absence concepts;
- numerical concepts;
- fuzzy concepts.

I label these concepts as non-canonical. They contrast with canonical concepts by not being straightforwardly expressible using instance of relationships with re-

spect to concepts in an ontology. I discuss each of these non-canonical concepts in turn.

Absence concepts are found throughout WALS. They refer to a concept explicitly defined as not being an instance of another kind of concept. In ontological terms, the concept of no case marking means that, in some language, there is no grammatical structure which can be claimed as instantiating case marking. Crucially, an absence category is quite different from inferring the absence of some grammatical phenomenon in a language simply because it is unattested or because there is no discussion of it in a grammatical description. The former is an explicit statement about the properties of a languages grammar and can, therefore, be taken directly as linguistic data, while the latter cannot.

Some sense of the variety of possible absence concepts can be achieved through a simple enumeration of some of the ones that are found in WALS. They include: no action nominals, no adpositions, no antipassive, no bilabials, no case, no distributive numerals, no fricatives, no gender distinctions, no glottalized consonants, no grammatical evidentials, no independent subject pronouns, no irregular negatives, no laterals, no nasals, no obligatorily possessed nouns, no perfect, no person marking, no plural, no possessive affixes, no productive reduplication, no question particle, no suppletion in tense or aspect, no tense-aspect inflection, no tones, no uvulars, and no velar nasal.

Looking through the definitional statements as given by the authors, some more absence concepts can be found. They include: non-agreeing, non-benefactive, non-bound, non-declarative, non-derived, non-finite, non-head, non-human, non-iconic, non-inflecting, non-inflectional, non-number, non-obligatory, non-paradigmatic, non-periphrastic, non-possessible, non-pronominal, non-realized, non-reduction, non-referential, non-reflexive, non-relativizable, non-sex-based, non-sibilant, non-singular, non-subject, non-syntactic, and non-verbal.

Clearly, absence concepts are important for typological description. The existence of absence concepts within WALS leads to a simple recommendation with respect to the relationship between a general ontology like GOLD and a community-specific ontology like the WALS ontology. In addition to allowing concepts in the community ontology to be related to the general ontology via positive relationships like language shows *instances of*, it is also necessary to allow them to be related via negative relationships like this language does not show instances of. While this would not seem to put a particular burden on the development of a general or a community-specific ontology, it would seem to put a burden on software designers building ontologically-intelligent search tools to ensure that their tools can deal with absence categories in a way which is useful to linguists.

It seems worthwhile to point out here that the general problem of encoding absence concepts may be more complex than is reflected in the WALS data. In WALS all absence concepts are assertions about a property of a languages gram-

mar. However, there is at least one other important kind of absence: *no information on*. Here, again, I need to contrast inference and explicit statements. If the only descriptive linguist who has worked on a particular language states that, quite simply, there is no data which would allow a language to be classified one way or another typologically, that is information of quite different value from discovering that a particular resource happens to have no information on a given topic. The former would not seem to call for looking for other resources to see if they contain the relevant information, while the latter would.

While I have encountered numerous absence concepts within WALS, the WALS ontology project has not attempted to exhaustively enumerate all possible kinds of absence concepts which might be useful as linguistic annotation. This seems like a worthwhile area for future research.

Another frequently occurring non-canonical concept found in WALS is the usage of features with countable values. For example, the feature number of genders distinguishes languages with no gender from language with two, three, four, or five or more genders.

Such *numerical concepts* are found rather frequently among the WALS features. Illustrating this approach are, for example, features covering the number of distance contrasts in demonstratives (map number 41) , the number of cases (map number 49), the number of classes of possessive classification (map number 59), and the number of degrees of remoteness as distinguished in the past tense (map number 66). Such overt examples nicely illustrate the importance of counting in typological parameters. However, there are also more covert examples of counts being used in the definitional details of typological parameters.

With respect to linking WALS concepts to the GOLD ontology, the existence of numerical concepts would seem to necessitate concepts which can directly refer to cardinal numbers. The usage of numbers for countable phenomena has to be distinguished from numbers used to divide more or less continuous parameters into discrete values. For example, in the feature depicting vowel/consonant ratios (map number 3), the value low is defined as having a ratio of two or less. The fact that the cut-off point is a whole number is clearly just an arbitrary decision, as the ratio results in a quasi-continuous parameter (see Cysouw, forthcoming, for a discussion of such quasi-continuous parameters in typology). The division of such continuous parameters into discrete, numerically-defined classes is related to the notion of fuzzy concepts, to which I turn next.

The third class of non-canonical concepts in WALS are what I call *fuzzy* concepts. These are concepts which cannot be straightforwardly related to other relevant concepts via a logical relation. This is not to say they are unrelated to other concepts, but rather some consistent policy needs to be developed for determining how to annotate such relationships which makes the fuzziness ontologically tractable. Some examples of fuzzy concepts found in WALS are: small con-

sonant inventory (map number 1), complex syllable structure (map number 12), borderline case marking (map number 49), weakly suffixing (map number 26), and highly differentiated genitives, adjectives, and relative clauses (map number 60).

The hallmark of a fuzzy concept is the use of a modifier like *small* or *like* which is open to a subjective or relative interpretation. For example, a small consonant inventory can only be considered small in reference to all the known consonant inventories and even then, there is still a subjective element to determining the boundary between, say, small and moderate. As another example, consider descriptions using the modifier *like*. Going through the definitional statements as presented in WALS, I found the following terms used: adjective-like, agent-like, case-like, patient-like, vowel-like, we-like. These can only be understood as referring to an unidentified deviation from the more prototypical meaning of the head-term.

Fuzzy concepts, then, can be distinguished from simply idiosyncratic concepts which combine categories in unexpected ways but which, in principle, are logically definable without an explicit statement of interpretation. One such idiosyncratic concept is the value pronouns avoided for politeness in the database for the feature politeness distinctions in pronouns (map number 45). This concept incorporates the notions of pronoun, avoidance, and politeness into a single concept in a way which would be unlikely to be specifically anticipated by developers of a general ontology. Nevertheless, assuming that an ontology contains these basic concepts, there is nothing fuzzy about them and it should therefore, be possible to relate a concept combining them to a general ontology using standard logical relations.

There would seem to be two broad strategies available for linking fuzzy concepts to a general ontology. The first is to always associate the entire concept to a reasonable non-fuzzy definition and treat relative and subjective terms like small or borderline as useful abbreviated conventions with no real ontological status. Thus, for example, a *small* consonant inventory could be defined as meaning less than fourteen consonants (which is, in fact, the definition given in map number 1). A second strategy would be to relate the fuzzy modifiers themselves to a general, concrete definition. Then, perhaps, *small* would be defined as two or more standard deviations away from the average for a countable quantity.

In looking at the prose descriptions accompanying the WALS maps, what I find is that in general, authors did in fact associate apparently fuzzy concepts with a non-fuzzy definition explaining. For example, how they specifically interpreted terms like *small* or *borderline* with respect to particular categories in particular languages. Thus, common practice in the creation of WALS was to take the first of the two strategies outlined above. The WALS ontology project is following this common practice and adopting it as its general strategy for dealing with fuzzy

concepts.

To make the discussion more concrete, in Figure 2.6 is depicted the map representing the data collected for the feature syllable structure. This feature has three values in WALS: simple, moderately complex, and complex. All these values refer to fuzzy concepts. However, within a prose description accompanying the map in the published version of WALS, map number 1 associates each fuzzy concept with more concrete definitions. These definitions do not explicitly appear in map number but are adapted from his prose descriptions of the relevant categories.

- Simple Syllable Structure: Describes a language which only allows syllable structures conforming to a (C)V pattern
- Moderately Complex Syllable Structure: Describes a language which only allows syllable structures conforming to (C)V(C) or CWV(C) patterns (where W stands for a liquid or glide)
- Complex Syllable Structure: Describes a language which allows syllable structures other than those described as permitted in simple syllable structure or moderately-complex syllable structure languages

The strategy of associating each fuzzy concept with a non-fuzzy definition, instead of devising a general definition for each attested type of fuzzy modifier, is a fairly comfortable one for the WALS ontology project since each of the typological feature databases was essentially conceived as its own internally coherent research project with terms defined for that project alone. I leave open the question as to whether or not some other project might find it worthwhile to deal with fuzzy concepts by giving concrete definitions to the fuzzy modifiers themselves. Whether this is considered necessary, it would seem to necessitate the inclusion of notions like average or standard deviation within a general ontology for linguistic description (perhaps linked to an upper ontology also containing such concepts).

2.2.1.3 Lateral relationships among concepts

By the term *lateral relationship*, I mean a relationship holding among two concepts in different resources. Of course, lateral relationships abound among categories in linguistic resources since it is what makes the data they contain comparable in the first place. Thus, for example, when one encounters the term nominative case in one resource and ergative case in another, one assumes a lateral relationship holding between those concepts where they both have something in common (being instances of case) and something not in common (being instances of different kinds of case). In principle, lateral relationships can all be encoded by linking linguistic concepts to their appropriate place in an ontology - in fact, this is one of

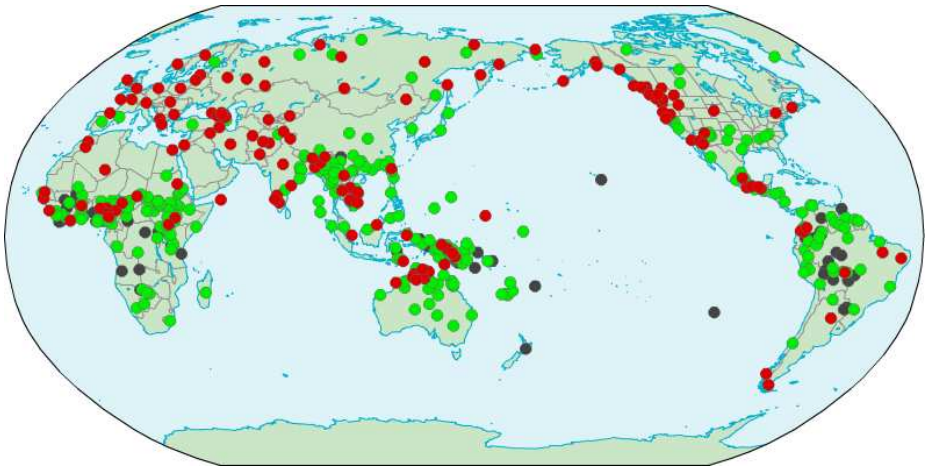


Figure 2.6: Map number 12. *Syllable structure*. Simple = black, moderately complex = green, complex = red.

the tasks ontologies were designed for.

However, in the WALS ontology project, I have often found it useful to make note of lateral relationships among concepts. The reason for this is a simple one: Sometimes it is possible to determine a lateral relationship holding between two categories before it is possible to relate each of those categories to a higher-level ontology. Thus, encoding lateral relationships allows us to indicate some of what I know about a category at a given time even if I do not know enough about the category to link it to an ontology.

Some of the classes of lateral links I have found useful in annotation are given below:

- Similar term names, but different concept
- Theoretically (almost) same concept, but certain languages classified differently
- Same concept, but no appropriate ontology concept(s) found.

In principle, a controlled vocabulary for such links could be developed. However, at this time, since there is no tool designed to exploit lateral links in developing an ontology, they are always inspected by hand, and I have found no need to develop a machine-readable annotation system for them.

2.2.1.4 The recoded WALSX

The recoded WALSX (WALS eXtended) tries to solve the above presented problems, in order to have a stable, quantitative-appropriate data set. Even if the oper-

ations needed imply an impressive volume of work, I decided that it is worth implementing. I did a careful analysis in order to (i) identify the problems described in Section 2.2.1, (ii) decide on the appropriate solution, (iii) careful implement the solution. These steps were performed for each feature, and a *conversion* file was created for each of them, while a software was implemented for creating the new WALSX and for analysing these files together with the original WALS data.

A typical conversion file contains the following sections, used to define each new feature and each new feature value, set the required transformation 'rules', and give the necessary information about the new feature, feature values and descriptions of the feature values.

FeatID	**Action type**
9	split + copy

Table 2.1: Sample Action Type.

The *action type* section presents the recoded feature id, together with the conversion type. This can be one of :

- *copy* = the old feature is copied without modifications;
- *split* = the old feature is split in 2 or 3 distinct new features without any dependencies;
- *split* + *copy* = usually, 2 new features are created one being a yes/no feature and one just a copy of the initial feature(Figure 2.8); implicit dependencies are noted;
- *divide* = the new recoded features are mostly dependent.

NewXFeatID	**NameXFeat**	**DescriptionXFeat**
15	The velar nasal	Velar Nasal
16	Type of velar nasal	Velar nasal, initial or not

Table 2.2: Features Description.

In the feature description section I am describing the new features, noting their new ID, names and descriptions. The new feature values are described in *features values* subdivision (Table 2.3).

The table *transformations* describe the 'rules' to obtain the new feature values based on the old ones (Table 2.4).

NewXFeatID	ValueXFeat	Description
15	1	No Velar nasal
15	2	Velar nasal
16	1	No Velar nasal
16	2	Velar nasal, also initially
16	3	Velar nasal, but no initially

Table 2.3: Features Values table describes each unique combination of feature id and feature value.

InitialFeatID	InitialFeatVal	NewXFeatID	ValXFeat
9	1	15	2
9	1	16	2
9	2	15	2
9	2	16	3
9	3	15	1
9	3	16	1

Table 2.4: The transformations table sets the ‘rules’ of creating the new features and new features values, i.e., a language with an initial feature value 1 for feature 9 will be set to have the feature value 2 for the new feature 15 and feature value 2 for the new feature 16 respectively.

It is easily to see that there will be

$$NrNewXFeatures \times NrInititalFeatureValues$$

records, as for each old feature value I define a new set of records for each correspondence to the new features.

The *Chance* and *Dependencies* tables provide useful information for the internal coding of the data, that the initial WALS was unable to describe. These fields are not necessarily required and they are filled only in the appropriate case. In the *chance* fields I define the character distance matrix for this feature, and in the *dependencies* sections I note the inter-dependencies between these feature values and other (if any) features values corresponding to the same or another features.

FeatXID1	**FeatXVal1**	**FeatXID2**	**FeatXVal2**
15	1	16	1
15	1	16	1
16	2	15	2
16	3	15	2

Table 2.5: Dependencies table. A feature value 1 for feature 15 implies feature value 1 for feature 16 (this relationship is bidirectional). The feature value 2 or 3 for feature 16 implies the feature value 2 for feature 15 (this relationship is unidirectional).

As mentioned above, these analyses were also useful in order to detect a 'character distance matrix' for some features. For example, consider feature 1, the 'Consonant Inventories', with 5 possible feature values: *small* (from 6 to 14 consonants), *moderately small* (15-18), *average*(19-25), *moderately large* (26-33), and *large* (34 or more consonants). It is natural to believe that a language who fell in the first category (*small*), will evolve easier to an intermediate category (*moderate small* or even *average*), than going directly to the category *large*. Therefore, a character distance matrix for this feature would look like the one in Table 2.6.

Of course, in some cases I did not know the exact relationship between feature or features values, and I must emphasis that these recoding is only based on personal knowledge of people involved in this research. Nevertheless, the new WALSX contains now 257 features and a lot of coded dependencies and transitions that I believe that are more appropriate to statistical approaches.

A small software program (Appendix 8.6) was implemented in order to:

- read the transformation file,

Feature1	Small	Moderately Small	Average	Moderately Large	Large
Small	0	1	2	3	4
Moderately small	1	0	1	2	3
Average	2	1	0	1	2
Moderately large	3	2	1	0	1
Large	4	3	2	1	0

Table 2.6: Character distance matrix for feature 1.

- extract the old data from WALS,
- insert the new data into new tables by following the transformations 'rules',
- create a *dot* file with the info in order to be able to use the "twopi" program [36] to display the transformations.

In Figures 2.7 - 2.9, 3 examples of such kind of conversions are presented, describing the *split* transformation (*Voicing in plosives and fricatives*), the *split + copy* transformation (*Tone*) and the *divide* transformation (*Fixed stress location*) respectively.

I also tried to test if the WALSX will provide a better phylogenetic picture. I analyzed 33 languages, grouped by families, and I used the data from WALS and WALSX respectively. The picture resulted from WALS (Figure 2.10) is similar with the one from WALSX (Figure 2.11). The analysis can be systematized as the following:

- both data-sets cluster the the languages from Indo-European, Altaic, Uralic, Trans New Guinea, and Nakh Daghestanian families;
- WALS data provides the entire grouping of Austronesian languages;
- WALSX manage to get closer the languages from Sino-Tibetan family, as well as the ones from Nilo-Saharan family.

Nevertheless, in this comparison I did not make use of any information discovered during the recoding, i.e. the distance between feature values, the dependencies, etc, but just used the data (from WALS and WALSX respectively) and created distance matrices for it.

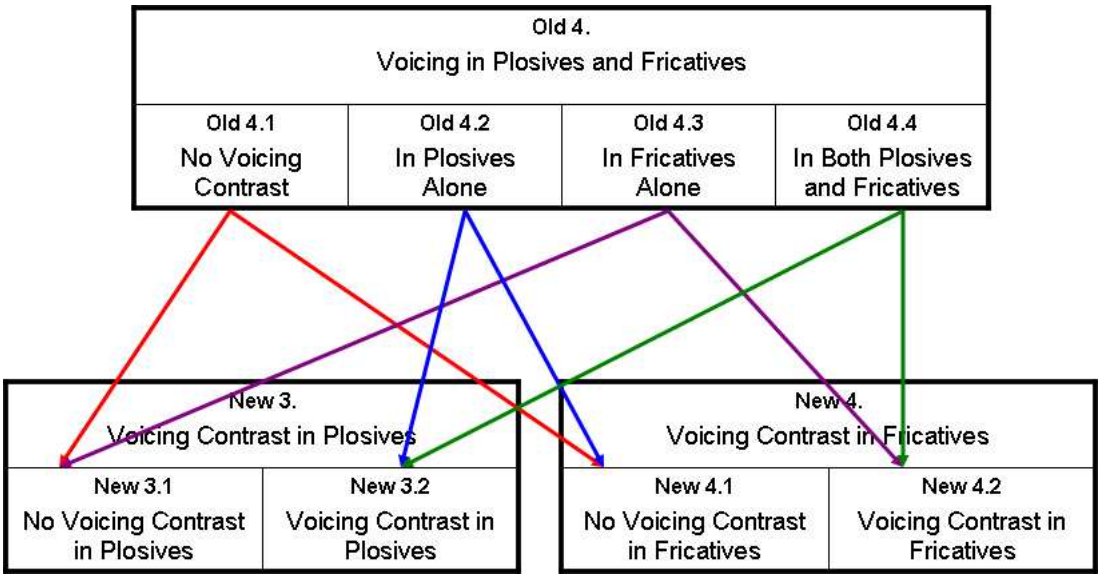


Figure 2.7: Example of *split* conversion

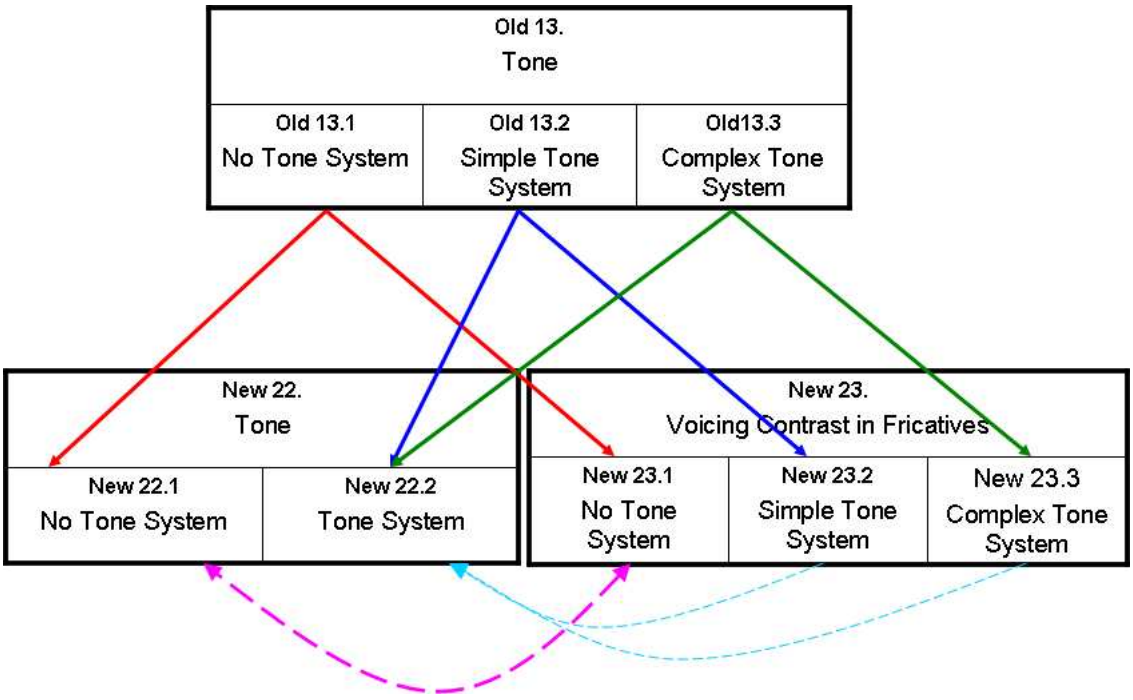


Figure 2.8: Example of *split* + *copy* conversion. Note the bidirectional dependency between the new feature values 22.1 and 23.1 and the unidirectional dependencies from 23.2 and 23.3 to 22.2.

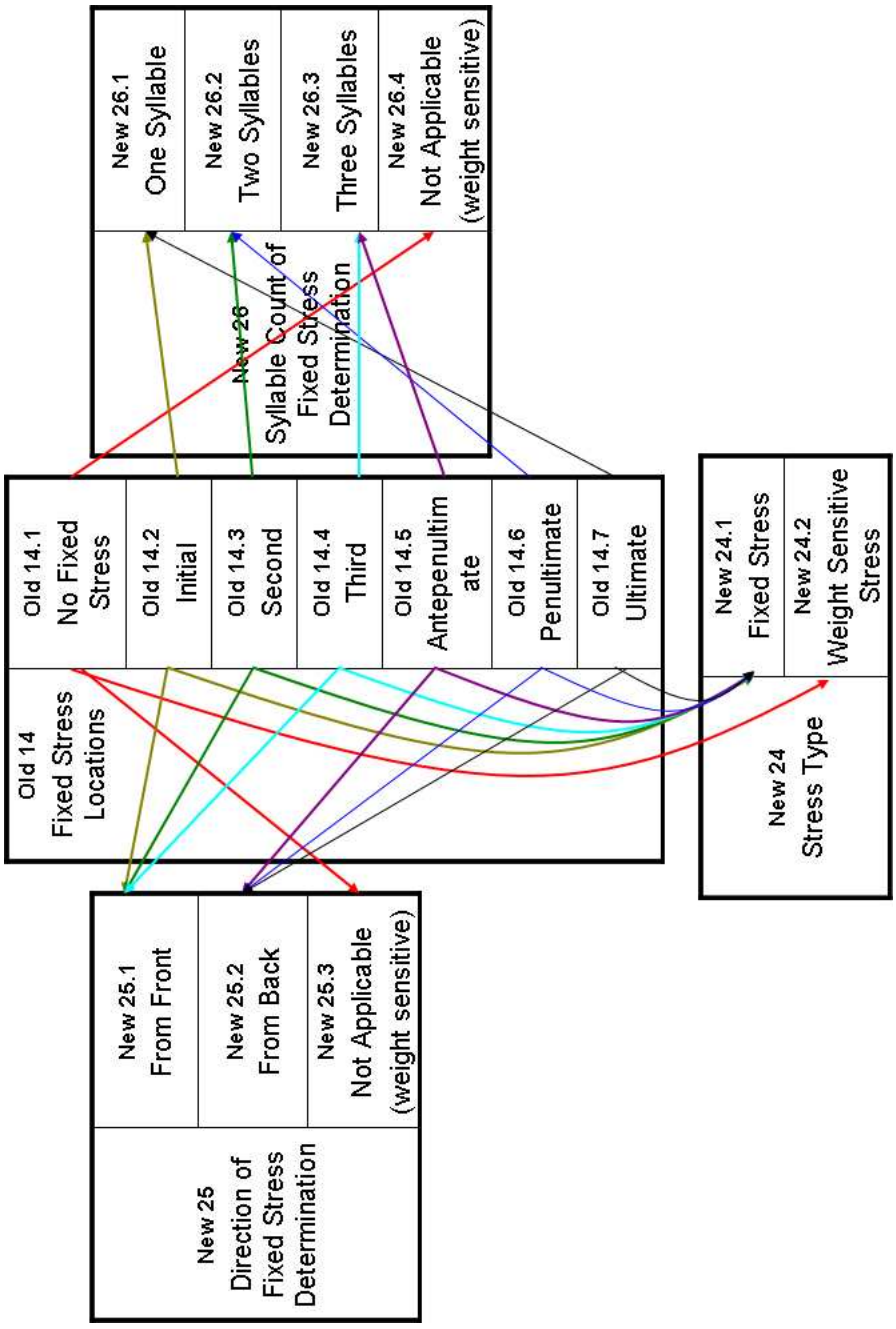


Figure 2.9: Example of *divide* conversion. Note the dependencies between new feature values 24.2, 25.3 and 26.4 respectively.

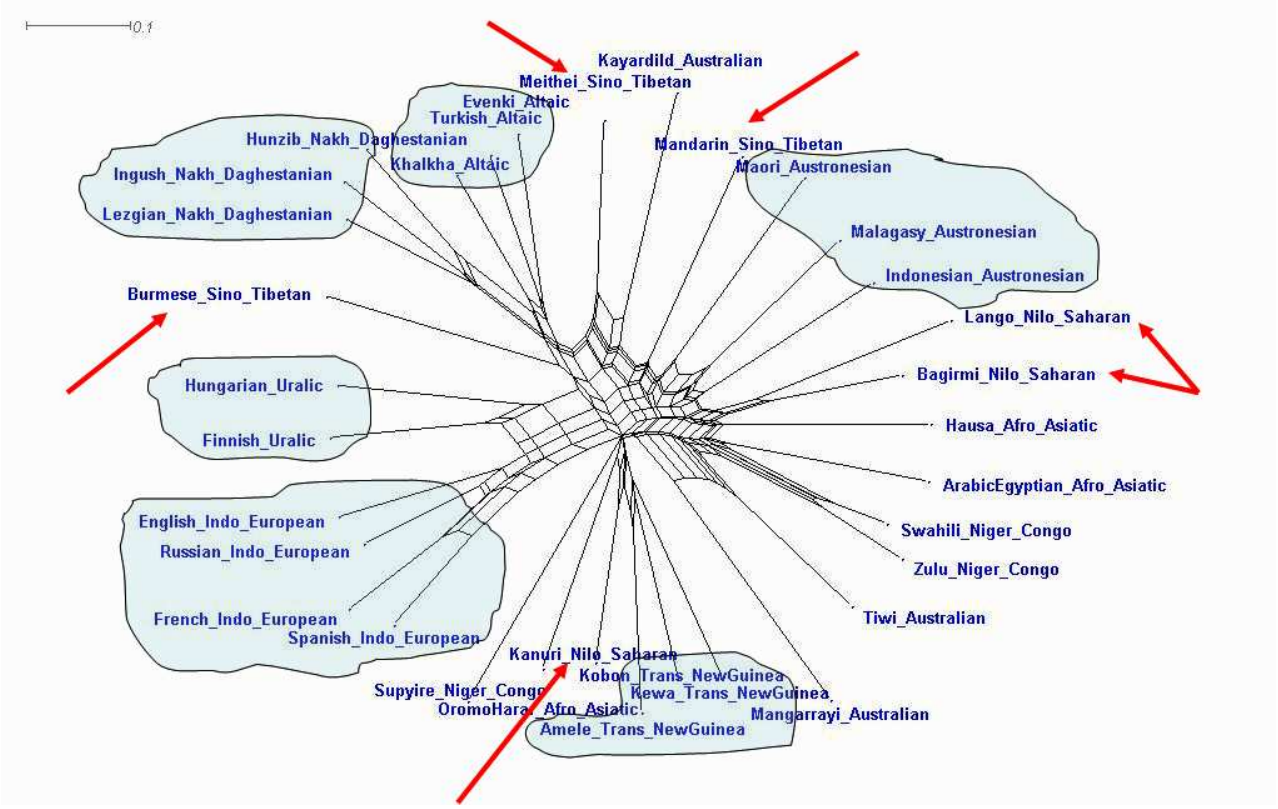


Figure 2.10: NNet using the original WALS data.

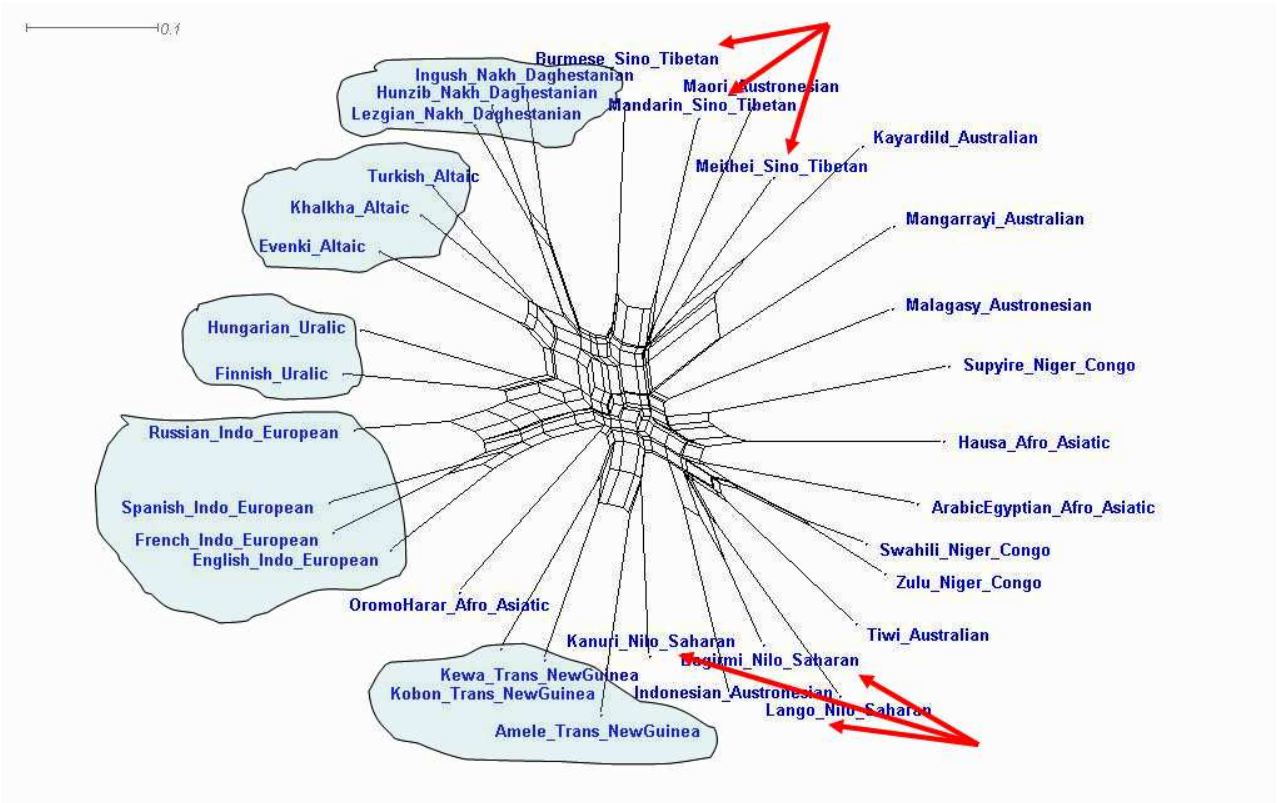


Figure 2.11: NNet using the recoded WALSX data.

2.3 Multidimensional scaling

2.3.1 Introduction

Another obvious question that was arisen when the first phylogenetic picture was obtained is, whether the relationships depicted are due to the phylogeny or just geographical influence. It is well known that languages can influence each other, if they are closely located, not only lexical, but also typological (Figure 2.12 shows a nice geographical distribution for feature 87 = Order of Adjective and Noun).

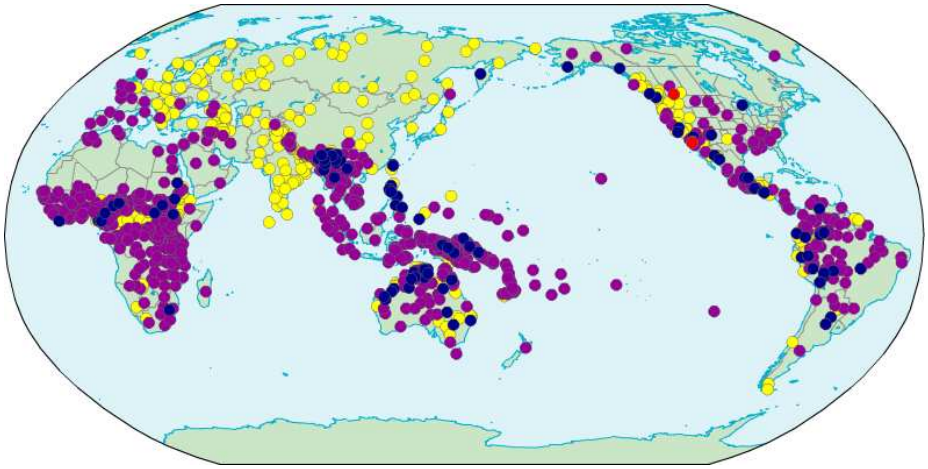


Figure 2.12: Geographical influence on typological features.

Moreover, in case of the languages in Laos country, as shown in Figure 2.13, there are 4 genera depicted (red, white, green, blue). The Kam-Tai genera contains Lu, Lao and Saek languages [2]. Why and how Lu, so far away geographically, belongs to the Kam-Tai genus and not to Palamung-Khmuic one? A revealing and simple explanation can be found if the geographical elements are considered, e.g. there is a river which probably was used by the population to travel faster. (Figure 2.14).

2.3.2 Results

Multidimensional scaling (MDS) is a set of related statistical techniques often used in data visualisation for exploring similarities or dissimilarities in data. An MDS algorithm starts with a matrix of item-to-item similarities, then assigns a location of each item in a low-dimensional space, suitable for graphing or 3D

[2]Thanks to Hans-Jörg Bibiko for pointing out this.

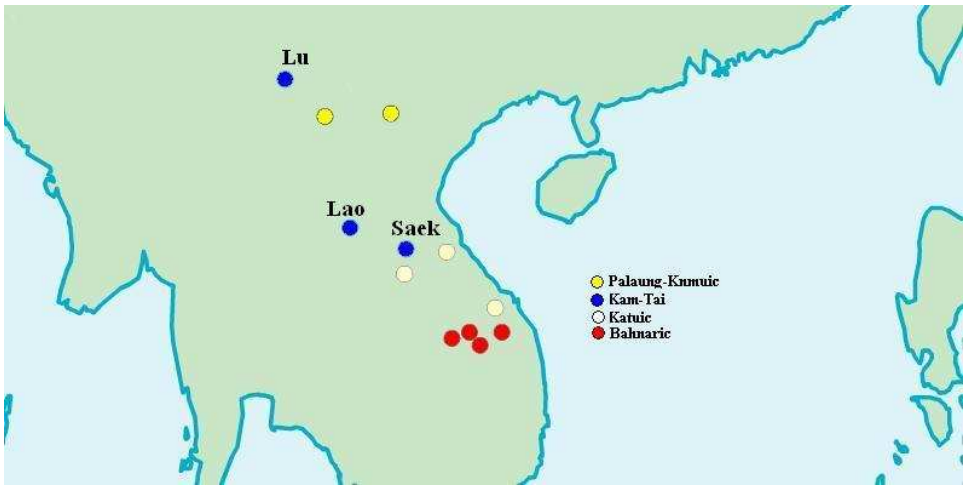


Figure 2.13: The languages in Laos, four genera (Palaung-Knmuic, Kam-Tai, Katuic, Bahnaric). Lu, Lao and Saek languages (blue) are part of the same Kam-Tai genera.

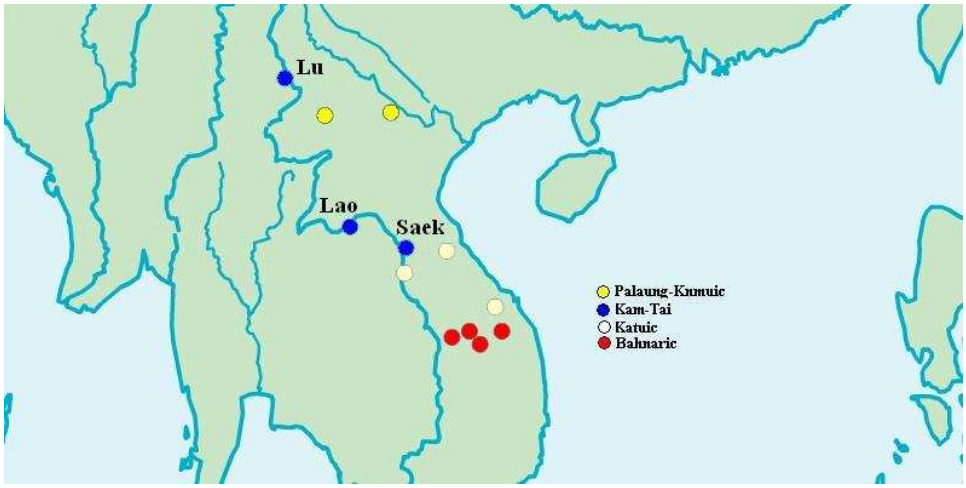


Figure 2.14: A possible explanation of Kam-Tai genera relationship for languages located geographical far apart: geographical elements (river).

visualisation. I used the multidimensional scaling (ALSCAL) from SPSS (Statistical Package for the Social Sciences) software package [70].

For this analysis I choose the case of Oceania, a region very diverse from the language grouping point of view as well as very interesting if a closer look of the geography is taken. There are mainly three language families (Papua New Guinea, Australian and Austronesian). While the languages from Papua New Guinea are very closed located, the Australian family is spread over the entire continent, and Austronesian is really diverse from the geographical point of view (Figure 2.16). A multidimensional scaling of the typological distances (Figure 2.15) shows a remarkable similarity with the geographical location[3], suggesting that many features in WALS are influenced by the location of the languages in the world (Figure 2.16).

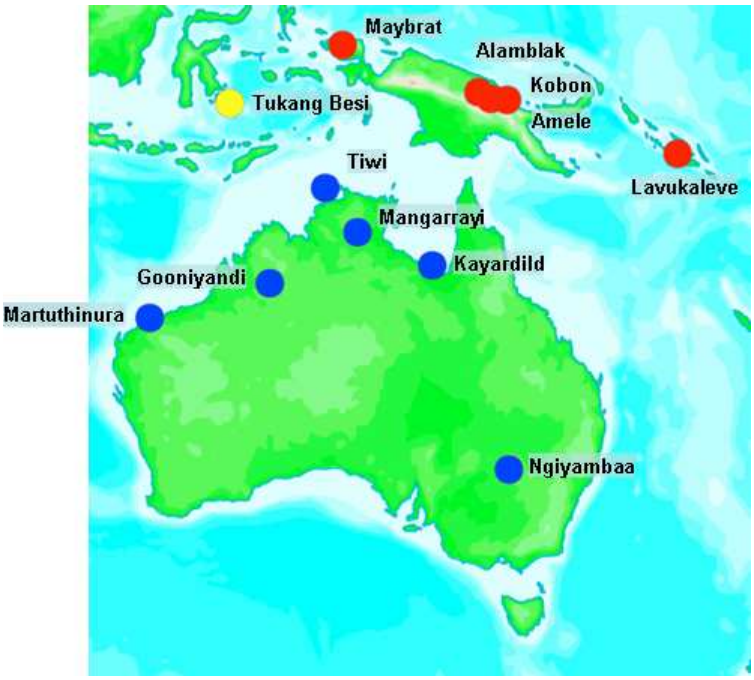


Figure 2.15: The geography of Oceania languages. Austronesian family = yellow, Australian family= blue, Trans-New Guinea family= red.

[3]Thanks to Michael Cysouw for discovering this.

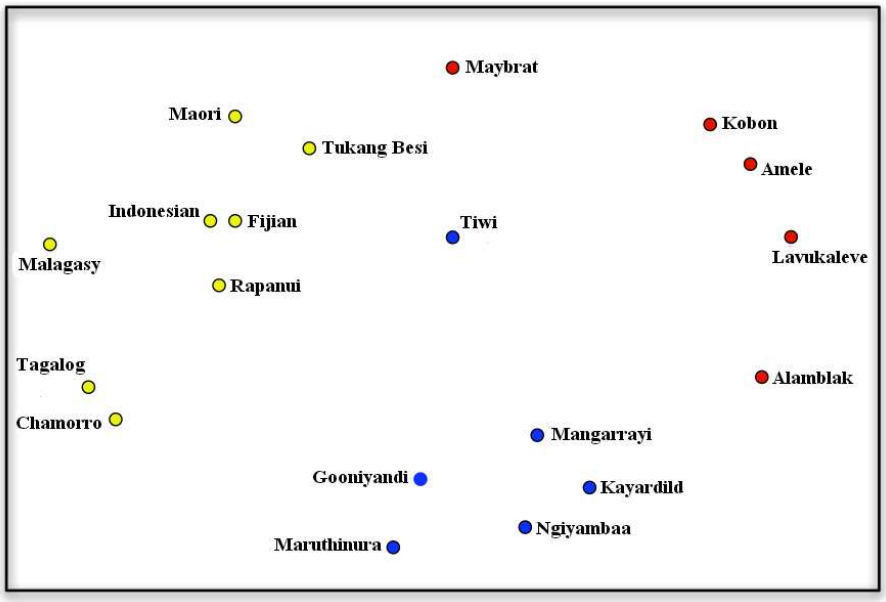


Figure 2.16: Multidimensional scale of Oceania languages.

2.4 The typological-geographical relationship

2.4.1 Analysis

It comes natural after the above results, in order to detect the geographical and typological influences, to compare the measurements based on these two criteria. I constructed two sets of distance matrices corresponding to the typological distances and geographical distances respectively. For the typological ones I constructed the distance measure described in Section 4.1, and for the geographical distances I used the *Haversine* formula [69] presented below:

$$\triangle lat = lat_2 - lat_1 \tag{2.1}$$

$$\triangle long = long_2 - long_1 \tag{2.2}$$

$$a = sin^2(\triangle lat/2) + cos(lat_1) * cos(lat_2) * sin^2(\triangle long/2) \tag{2.3}$$

$$C = 2 * atan2(\sqrt{a}, \sqrt{(1-a)}) \tag{2.4}$$

$$dist = R * C \tag{2.5}$$

where $R = 6.371$ km. Presuming a spherical Earth with radius R , and that the locations of the two points in spherical coordinates (longitude and latitude) are long1, lat1 and long2, lat2, then the Haversine Formula will give mathematically and computationally exact results. The intermediate result c is the great circle

distance in radians. The great circle distance d will be in the same units as R.

The analyses were conducted for different data sets, as following:

- Build typological distance matrix (D_T) and geographical distance matrix (D_G) respectively;
- Remove the biggest distances (distortion);
- Take the extremes of the rapport $D_T(i,j)/D_G(i,j)$ for each i and j in the data set;
- Interpret very low values as linguistically (too) similar;
- Interpret very high values as linguistically (too) diverse.

2.4.2 Results

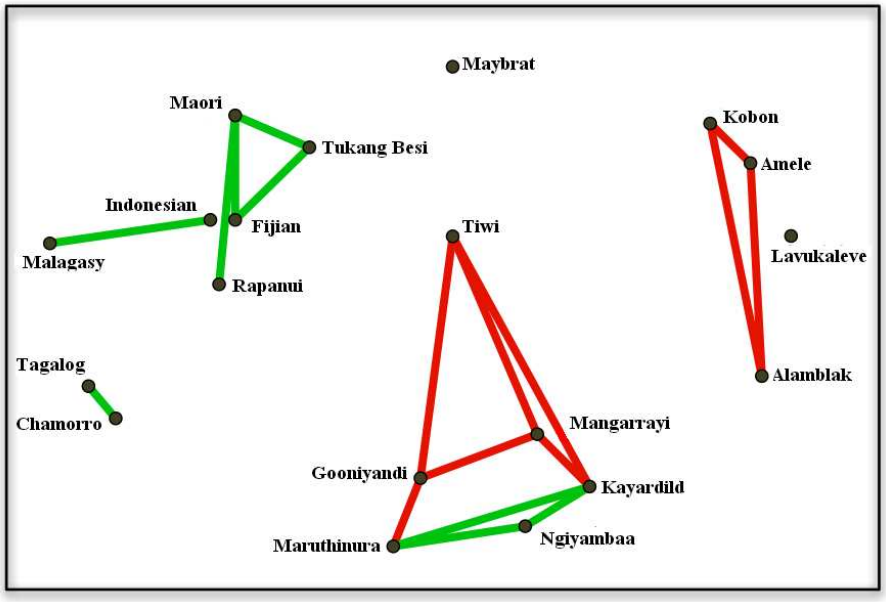


Figure 2.17: Multidimensional scale of Oceania languages. Red lines show 'unexplained' typological dissimilarities. Green lines show 'unexplained' typological similarities.

The green lines in the Figure 2.17 show a too high similarity from the typological point of view compared with a larger geographical distance. That is, there is no reason for this languages to be so similar, due to their geographical position, but their similarity might be explain by a typological influence, which might

imply genealogical relationship. Reversely, the red lines show a close geographical position with a high dissimilarity between languages. Both cases are very interesting in linguistic research.

2.5 Feature clustering

Another useful approach that I performed was to identify, if any, the feature clusters. In other words, I tried to detect if some features can be grouped together based on their world wide distributions of features values, or based on their consistency between families and/or geographical positions.

Using the CADM software [50, 51] I was able to measure the consistency of the features with the overall distribution of features values in the entire dataset, and also between each of the features (see also Section 4.2.1). The aim of this analysis was to create distance matrices for each feature (based on a set of n languages), and to detect by using the CADM software, which of these matrices are more correlated with the others. I achieved this by comparing their *similarity* to the overall matrix (obtained by using a distance measure for the same set of languages but for all the features).

The permutations tests were performed for all features using a number of 9999 permutations and the results were used as input data for a multidimensional scaling (in order to obtain a graphical picture of their similarity). The first picture obtained from the multidimensional scaling for all 141 features did not showed any defined clusters (Figure 2.18). I decided that the number of features analyzed should be reduced (the graphical resolution was not interpretable), therefore I arrived to a set of 17 features for which I performed again the CADM analyses and used MDS to display the results. The 17 features selected are the following: 51=Position of Case Affixes, 57=Position of Pronominal Possessive Affixes, 69=Position of Tense-Aspect Affixes, 82=Order of Subject and Verb, 83=Order of Object and Verb, 84=Order of Object, Oblique, and Verb, 85=Order of Adposition and Noun Phrase, 86=Order of Genitive and Noun, 87=Order of Adjective and Noun, 88=Order of Demonstrative and Noun, 89=Order of Numeral and Noun, 90=Order of Relative Clause and Noun, 91=Order of Degree Word and Adjective, 92=Position of Polar Question Particles, 93=Position of Interrogative Phrases in Content Questions, 94=Order of Adverbial Subordinator and Clause, 104=Order of Person Markers on the Verb. The correlation values, as well as the permutation indexes obtained from the CADM analyses allow us to draw lines between the points that have a higher correlation (Figure 2.19).

As the Figure 2.19 shows, features 83, 85, and 86 are very well connected not only between them, but also with the majority of the other features. The cluster 87 and 88 (Noun positions) is surprisingly grouped with 51 (case affixes). The fea-

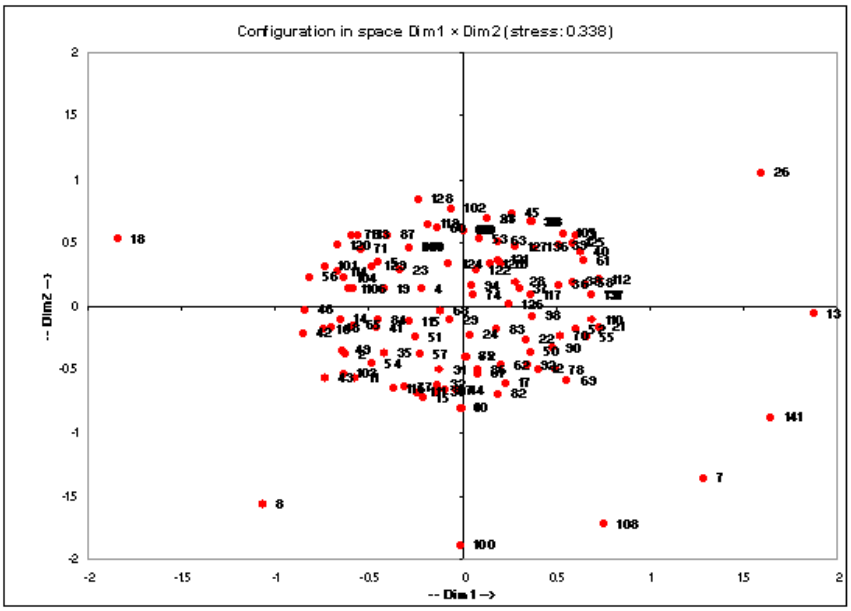


Figure 2.18: Multidimensional scale of 141 features. No clear clustering, and no possible interpretation if more graphical elements are added.

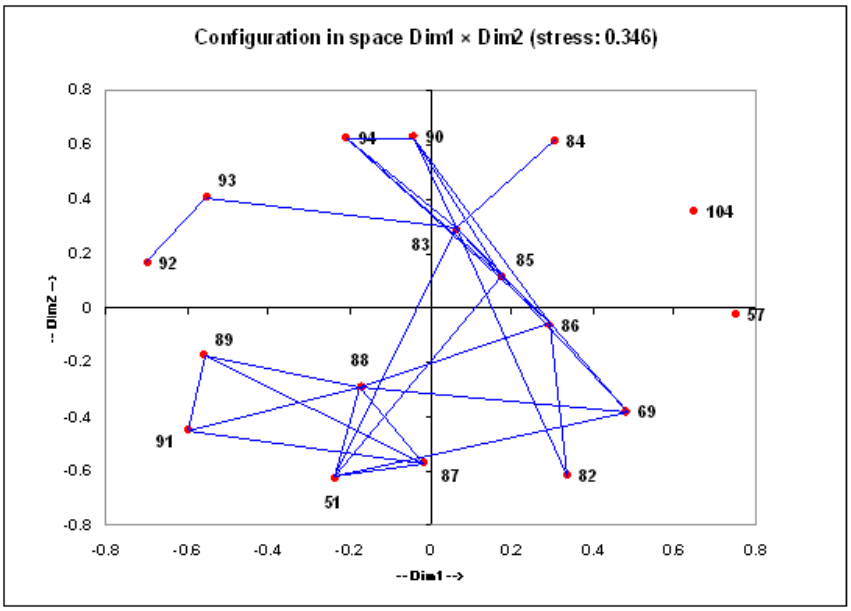


Figure 2.19: Multidimensional scale of 17 features. Features 83, 85, and 86 form a central cluster as well as the clusters formed by features 51, 87, and 88. All the other features, except 57 and 104, are connected through these clusters.

tures 89 and 91 are connected through this cluster to the 83, 85, 86 (the 'central') one. The features 57 and 104 do not show any relationship to the others at this level. Feature 84 (object/oblique/verb) is nicely connected with 83 (object/verb) as well as is the feature 92 with the feature 93.

One can set various initial grouping of features of interest (my suggestion is to be less than 20 [4].) and perform the above analyses in order to detect possible similarities/clusters between them. A simple software was implemented in order to read the output from the MDS analysis (points coordinates) and draw the connections lines (Appendix 8.5).

Nevertheless, one might use this approach in order to detect geographically related features. For this, the above analysis must be changed only in the fact that the overall distance matrix that is used as a reference point for comparing the features distance matrices, should be build as a real geographical distance between languages (I recommend the Haversine formula for this). In this case the similarities observed will provide information about the clustering of the features from the geographical influence point of view, which is still a controversial discussion in these days. These results might be a starting point for further analyses that tries to find/explain feature correlation with the world wide distributions, as well as the geographical influence on typological features.

[4]In order to proper interpret the results. Above 20 it is really difficult to distinguish the graphical elements

Chapter 3

‘Improving’ missing data. Improving distance measurements

3.1 Missing data

WALS has a low coverage of data points (a data point means a value of a feature assigned to a language), only 20% filled, but this 20% means more than 50.000 data points.

Clearly, one can complain about the missing data, but the existing data points should still provide enough information for various kind of analyses. Nevertheless, one of the possible approach when dealing with missing data is to try to fill it in. But, because of the enormous financial and research efforts to fill in even only a few additional data points [1], I decided to renounce on the strict method and to try to recover, if possible, some information regarding the missing data. That is also due to the fact that if one will apply a statistical method on only 20% data, its results cannot be rigourously sustained. In general, there are two possibilities when trying to fill in missing data:

- detect (and use) the most probable value,
- fill in the missing values with all possible combinations and check which variant is the best one.

[1] Some preliminary estimates indicate that the data in WALS costs various hundreds Euros per data point (cf. Michael Cysouw, p.c.).

The first approach is very easy to implement, as in the WALS data, I can look at the most frequent feature value of the languages that belong to sets defined on families/genera criterion. If, for example, 95 percent of the Indo-European languages have the feature value V_1 for feature F, and the other 5 percent are not defined, one might assume that these 5 percent have also the same feature value (i.e., V_1). Of course, this approach is a little dangerous as the level of trust depends very much on the distribution of the features values in the family/genera set (e.g what conclusion can be obtained if 40% of languages have feature value V_1, 40% have feature values V_2, and the rest are not coded? What will be their assumed feature value?). The second method, which I would like to call "searching in the space of missing values", implies to fill out the missing data with all possible appearances of the feature values. Again, there is a difficult problem in those cases where feature values belong to a large interval, and/or the missing data are often present. Two cases are possible:

- only a single cell per column is noted as missing,
- there exist more than one missing datum per column.

In the first case, each column that contains missing data is expanded to all possible variations of its values. As shown in Figure 3.1.**a**, if one has three columns (features) with possible ranges of values of 2, 4, and 5 respectively, the expanded alignment will contain now 11 columns instead of 3 (Figure 3.1.**b**).

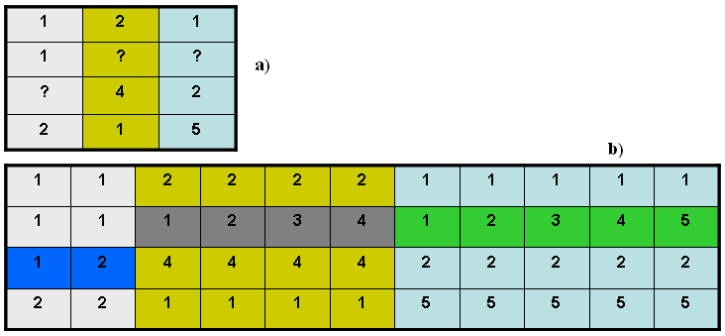
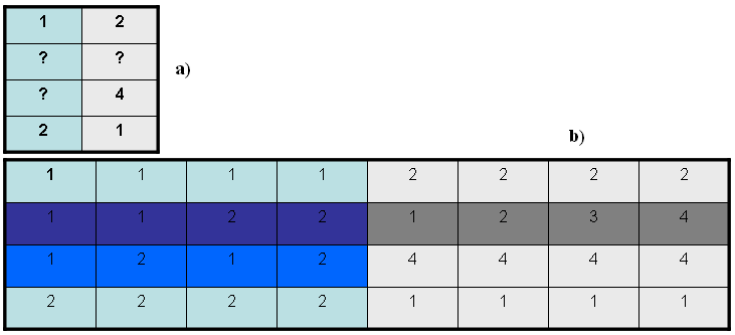


Figure 3.1: Missing data. One per column. **a)** initial alignment, **b)** expanded alignment.

For the second case, with more than one missing data point per column, each feature will be expanded for all the possible combination of feature values between the missing data. As shown in Figure 3.2 **a**, the first alignment, contains 2 columns. The first column contains 2 missing cells, and the range of feature values is [1, 2], while the second column (Figure 3.2 **b**) has 1 missing value but the range of feature values in [1, 4]. Therefore the initial alignment will be expanded



1	2
?	?
?	4
2	1

1	1	1	1	2	2	2	2
1	1	2	2	1	2	3	4
1	2	1	2	4	4	4	4
2	2	2	2	1	1	1	1

Figure 3.2: Missing data. More than one per column. **a)** initial alignment, **b)** expanded alignment.

to a 8 columns wide alignment. The general formula to calculate the length of the final alignment is presented in Equation 3.1.

$$nrResColumns = \sum_{columns} nrIntervalRange^{nrGaps} \tag{3.1}$$

Again, this is hazardously from the point of view of complexity and the level of incertitudes, especially in the case where one is dealing with very large interval ranges and incomplete data, but I concluded that it is might still be worthwhile to analyze.
Performing the entropy algorithm described in Section 5.2 on these "full" alignments might provide useful information about features, even if they contain missing data. Nevertheless, the method just assumes the 'best' options with a certain level of trust. For example, let assume that this tree is analyzed [2]:

$$((Finnish, Hungarian), (Romanian, (Italian, (Spanish, Portuguese)))); \tag{3.2}$$

and a feature has the following feature values distribution for this grouping:

$$((1, 1), (2, (2, (?, 2)))); \tag{3.3}$$

(where question mark denotes missing data), with the possible features values being in the range [1,2]. By replacing the question mark with 1 and 2 respectively, the entropy algorithm (Section 5.2) detects that having the value 2 instead of the question mark results in a 'good' phylogenetic feature and one may notice that the value of 2 is also matching the majority of the feature values analyzed. Again, this is just an assumption, but one that is strongly supported (it would be interesting for

[2]The strings in Expressions 3.2 and 3.3 are called *Newick tree expressions* and are often used in phylogenetics to represent tree structures.

further research to investigate the 'predicted' values, e.g., does Spanish have the feature value 2?). To run the entropy algorithm on a selection of, for example 20 languages for all the 141 features from WALS, will imply too much computational effort (remember that I have 80% missing data and the interval range of feature values can vary from 2 up to 9). Nevertheless, I adopted some variants of this method on restricted set of data:

- Remove features that have missing data. Features that contain missing data are removed from the analyses.
- Remove leaves that have missing data for all features. The sequences are removed whenever they contain missing data in any position.
- Remove leaves that have missing data for each feature. The sequences are removed whenever they contain missing data for one feature.

In the above last two cases, the algorithm must modify the initial tree, by removing the leaves in question and correctly adjust the parenthesis format of the tree. Moreover, in the last situation, this adjustment has to be done for every feature analyzed.

3.2 Distance measurements

Clearly, distance matrices are important data input for phylogenetic analyses. As it is easy to calculate a geographical distance between two points on the globe, using for example the Haversine formula [69], it is not as easy to establish the correct distance measure for species, languages or other objects of interests. This issue is more difficult, as it usually presents multiple problems: missing data, wrong/default mistakes in data/measurement, rescaling, etc. A very important issue in obtaining the correct (or most reliable) results in phylogenetic analysis is to perform a good method for constructing the distance matrix. What is good, and how to measure the 'reliability' of the methods? Different approaches were analyzed and the most significant are described in the next section, with the hope that this might be applied to a general dataset.

3.2.1 Methods

3.2.1.1 Hamming distance. *Relative* Hamming distance

Hamming distance [40] is probably one of the most common distance measurements, but as presented next, it has a huge drawback when dealing with missing

data. The distance is calculated by counting the elements that are different for any two objects. In my case:

$$D_F(L_1, L_2) = \begin{cases} 1 & \text{if } 0 \neq F(L_1) \neq F(L_2) \neq 0, \\ 0 & \text{else (i.e., if } 0 \neq F(L_1) = F(L_2)). \end{cases} \tag{3.4}$$

$$D(L_1, L_2) = \sum_{i=1}^{141} D_F(L_1, L_2) \tag{3.5}$$

How to deal with missing data, is a problematic issue. A self-explaining sample is presented next, where I assumed that I analyze three languages and five features as in Table 3.1 (question marks represent unknown data). Now, there are

	Feat1	Feat2	Feat3	Feat4	Feat5
German	1	2	1	3	?
English	1	2	1	4	1
Romanian	3	?	1	6	1

Table 3.1: Sample data table

at least three ways to build variants of the Hamming distance matrix:

- *Treat '?' as different* - '?' is different from everything else.
  D(G,E) = 0+0+0+1+1=2
  D(G,R) = 1+1+0+1+1=4
  D(E,R) = 1+1+0+1+0=3

- *Ignore '?'* - Ignore the entire column that contains '?'.
  D(G,E) = 0+0+0+1+0=1
  D(G,R) = 1+0+0+1+0=2
  D(E,R) = 1+0+0+1+0=2

- *Ignore '?' but count the available data (Relative Hamming distance)* - Same as above but divided by the number of columns that have values defined for both objects (the actual number of cases where one has the possibility of measurements).
  D(G,E) = 1/4 = 0.25
  D(G,R) = 2/3 = 0.66
  D(E,R) = 2/4 =0.5

There are a lot of differences in the results and this suggest a comparison with the output of other methods in order to detect which method provides the most significant results. In general, I will adopt the following distance measure (Equation 3.6).

$$dist(L1, L2) = \sum_{F \in features} \frac{\#dissim(F(L1), F(L2))}{\#dissim(F(L1), F(L2)) + \#sim(F(L1), F(L2))} \tag{3.6}$$

where $\#dissim$ and $\#sim$ represents the number of differences in coded data, respectively the number of similar coded data.

3.2.1.2 Refining the similarities

What if, instead of 'doing nothing' when two objects are the 'same' for some characteristics (the case of 'adding' 0 in Equation 3.4), one tries to refine this similarity. For example in a group of 20 persons, only two of them are male. The characteristic of sex (male/female) has a bigger influence if the *value* is male than if the *value* is female. In other words, in this case, the males are more similar from the 'sex' point of view than females. Or, to put it in another way, a shared common characteristic should not be counted as the same as a shared rare characteristic. The method implies to count for each feature:

- F_A = frequency of value A, e.g. the number of cases where the feature is defined as having value A
- F_T = frequency of available data points for this feature, e.g. the number of cases where the feature is defined.

or

$$F_A = \#\{L \mid F(L) = A\} \tag{3.7}$$

and

$$F_A = \#\{L \mid F(L) \neq 0\} \tag{3.8}$$

I calculate the similarity measure as:

$$NormS(A) = F_A / F_T \tag{3.9}$$

while building the distance between two languages will become now:

$$dist(L1, L2) = \sum_{F \in features} \frac{\#dissim(F(L1), F(L2))}{\#dissim(F(L1), F(L2)) + (1 - NormS(F(L1)))} \tag{3.10}$$

where $NormS$ is calculated when $F(L1) = F(L2)$, which implies $NormS(F(L1)) = NormS(F(L2))$. For a better understanding, the following example (Figure 3.3) is presented, where a feature value distribution is presented: 84% of languages have feature value 1, 12% of languages have feature value 2, and 4% of languages have feature value 3 respectively:

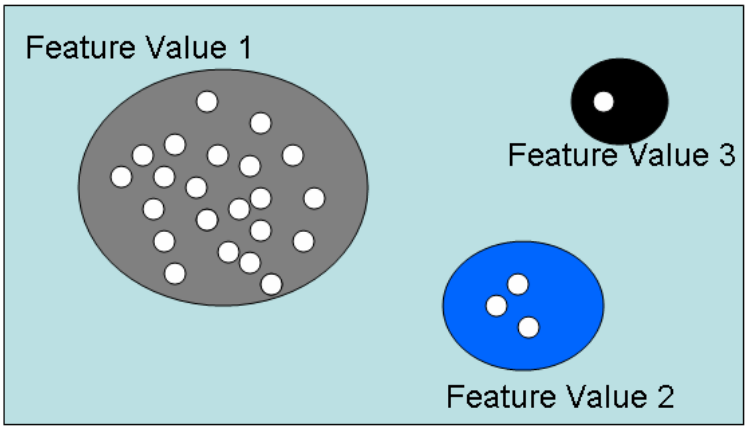


Figure 3.3: Sample of feature values distribution. The languages (the dots) are grouped by feature value.

My idea implies that two languages (objects) that share the feature value 2 are more similar than two languages (objects) that share the feature value 1. For example, in the case of map 51 from WALS (Position of Case Affixes):

- value 1 (Case suffixes) = 431 cases out of 934 available. $\Longrightarrow$ NormS = 0.4603
- value 4 (Case tone) = 4 cases out of 934 available. $\Longrightarrow$ NormS = 0.0043

$\Longrightarrow$ so the results assume that two languages that share the value 4 for the feature 51 are more similar than two languages that share the value 1.
Another formulae used is presented in Equation 3.11

$$NormS(A) = -log(F_A/F_T) \tag{3.11}$$

and then the formulae 3.10 will be transformed in :

$$dist(L1, L2) = \sum_{F \in features} \frac{\#dissim(F(L1), F(L2))}{\#dissim(F(L1), F(L2)) + NormS(F(L1))} \tag{3.12}$$

3.2.1.3 Refining the dissimilarities

Are all the differences the same [3]? For example analyzing two human groups, is being different in height the same as is being different in the number of eyes? In my analysis I am grouping the languages by genus and count , for every two feature values $val1$ and $val2$ the number of genera where:

- a = neither $val1$ nor $val2$ are present
- b = $val1$ is present and $val2$ is not present
- c = $val2$ is present and $val1$ is not present
- d = both $val1$ and $val2$ are present

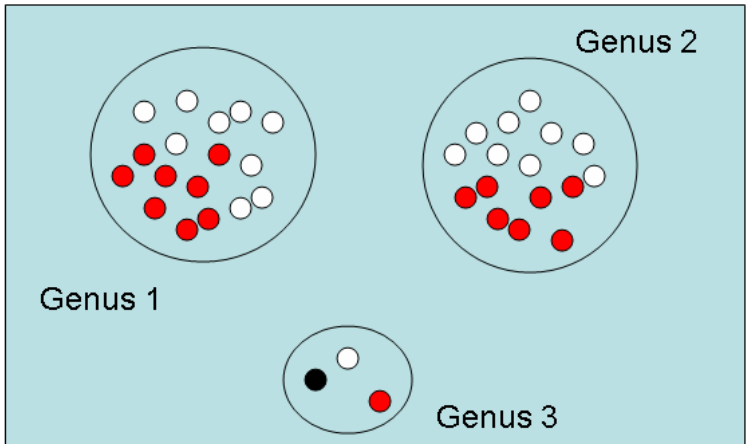


Figure 3.4: Feature values distribution for languages grouped by genera. Different colours represent different feature values.

Then, I used the following formulae to obtain a measure of dissimilarities:

$$NormD(val1, val2) = \frac{b+c}{a+b+c+d} \tag{3.13}$$

or

$$NormD(val1, val2) = \frac{1 - \frac{ad-bc}{\sqrt{(a+b)(c+d)(a+c)(b+d)}}}{2} \tag{3.14}$$

Instead of adding a value of 1 to the distance between two languages $L1$ and $L2$ that have $val1$ and $val2$ respectively defined for a feature F, I will add the value

[3]Various formulas (for both similarities and dissimilarities) were tried due to Michael Cysouw's suggestions.

of $NormD(val1, val2)$. Of course, $NormD(val1, val2) = NormD(val2, val1)$ and the overall distance measure will now be as defined in Equation 3.15 :

$$dist(L1, L2) = \sum_{F \in features} \frac{NormD(F(L1), F(L2))}{NormD(F(L1), F(L2)) + \#sim(F(L1))} \quad (3.15)$$

In cases similar with ours, when the dataset might be divided in sets of interests (geographical, genealogical), it becomes clear that applying both refinements of similarity and dissimilarity respectively, by using the Formulae 3.16 when building the distance matrix, improves, even if not in the amount expected, the results obtained. The general formula will now be defined as in Equation 3.16.

$$dist(L1, L2) = \sum_{F \in features} \frac{NormD(F(L1), F(L2))}{NormD(F(L1), F(L2)) + 1 - NormS(F(L1), F(L2))} \quad (3.16)$$

3.2.2 Results

All distance measurements were compared, in the way of 'how good they fit the required data', with a typological distance matrix. The typological distance matrices were build, in a simplistic way, from the *ethnologue* information [35], by weighting each internal edge with a value of 1. For example, if the tree in Figure 3.5 is analyzed then the associated distance matrix in Table 3.2 is constructed.

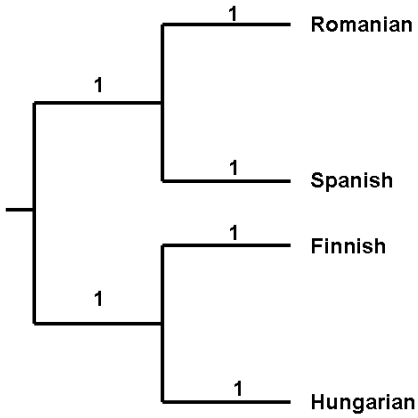


Figure 3.5: Weighting uniformly the edges of the tree with a value of 1.

The comparison (Figure 3.6) was processed for 5 language families, and it showed a very good performance of the relative Hamming distance. This measure should definitely be used when other improvements are not possible. Nevertheless, Formulae 3.12 allows with only a small amount of effort, to obtain improved

	Romanian	Spanish	Finish	Hungarian
Romanian	0	2	4	4
Spanish	2	0	4	4
Finish	4	4	0	2
Hungarian	4	4	2	0

Table 3.2: The distance matrix associated

results. The second variant of NormS (Formulae 3.11), as well as the variants of the NormD showed the same criterion of affinities to the ethnologue matrix (the same variance of the results). Surprisingly, the combination of both best NormS and NormD variants showed not only no improvement, but even worse, a depreciations of the results. [4]

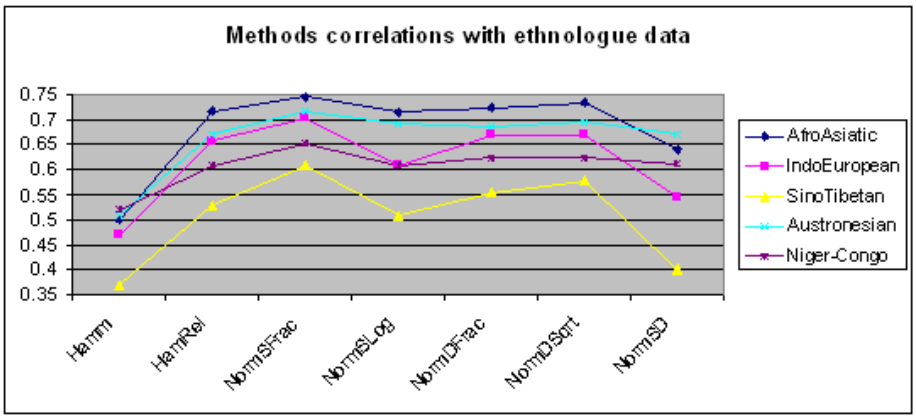


Figure 3.6: Comparison of various distance measurements with the phylogenetic information.

3.2.3 Phylogenetic comparisons

For the phylogenetic comparison, I selected a set of 25 languages (the best coded ones) grouped in 5 families, and I applied the different methods presented in this chapter to build the distance matrices as an input to NNet algorithm. The results showed a very good improvement using the relative Hamming distance (Figure 3.8), compared with the simple Hamming distance (Figure 3.7), which is also suggested by the plot in Figure 3.6. The NormS (Figure 3.9) and NormD (Figure

[4] I am still puzzled why the combination of the best variants of NormS and NormD did not produce an improvement, but worse, it depreciates the results.

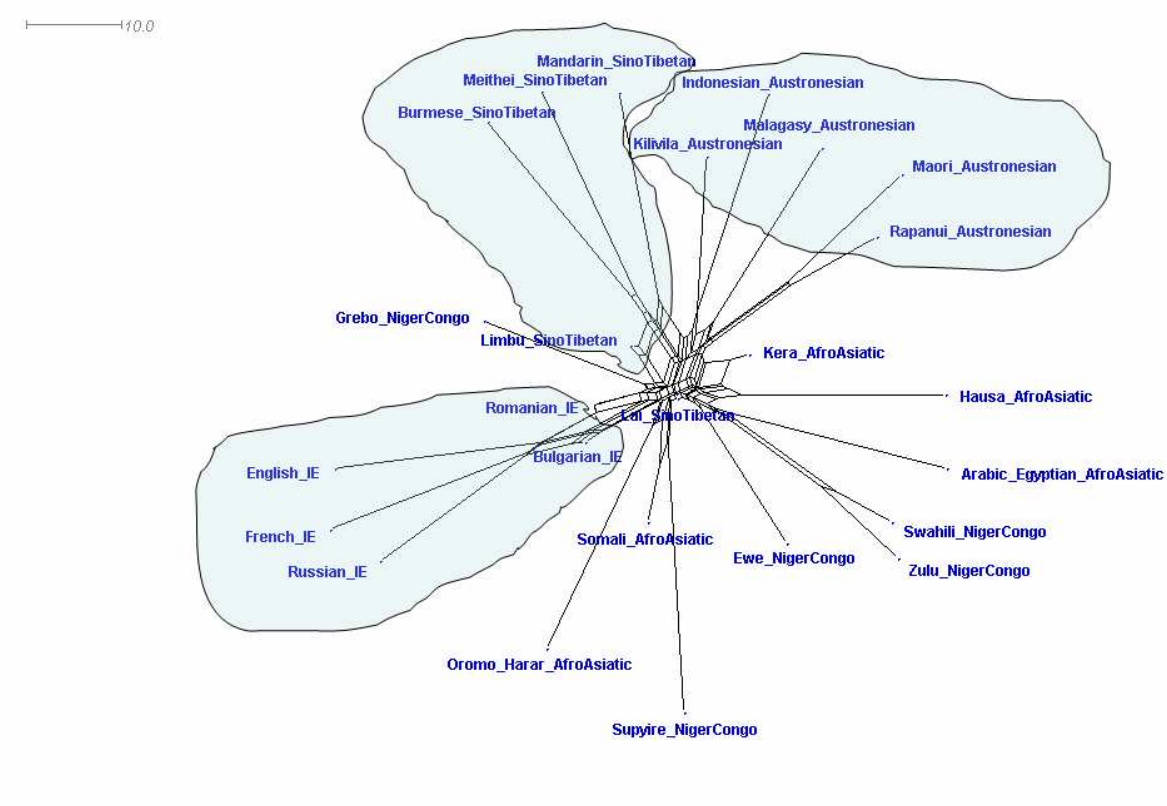


Figure 3.7: NNet using Hamming distance.

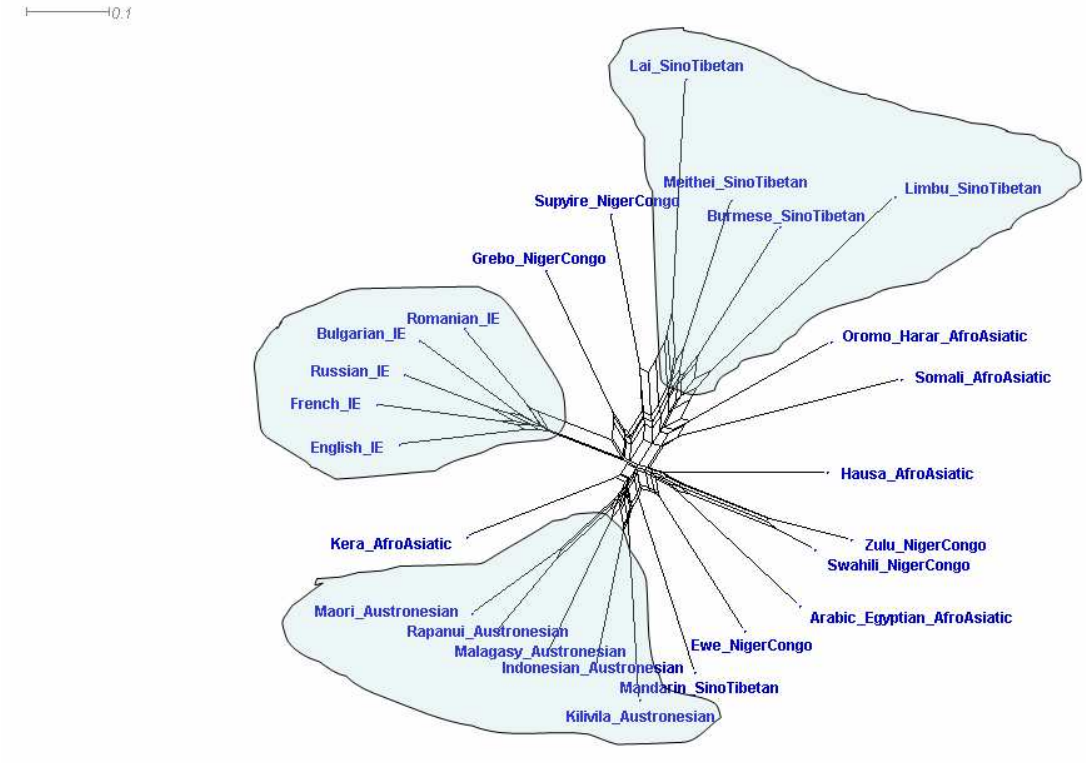


Figure 3.8: NNet using relative Hamming distance.

3.10) variants showed small improvements, as NormS brings the Sino-Tibetan languages together, as well as nice small clusters for Niger-Congo and Afro-Asiatic languages.

Of course, the next obvious question was if by combining the NormS and NormD, I will improve the clustering or, as the plot in Figure 3.6 suggests, I will get worse results. It actually came out a NNet picture almost identical with the one obtained by using Hamming relative measurement, and it seems the combination of both NormS and NormD somehow cancel each other. Why, and how this can be improved might be an interesting topic for future investigations. Nevertheless, the NNet phylogenetic analyses proved to be consistent with my distance measurements, even if I used a simplistic method to build the *ethnologue* distance matrix (as in Section 3.2.2).

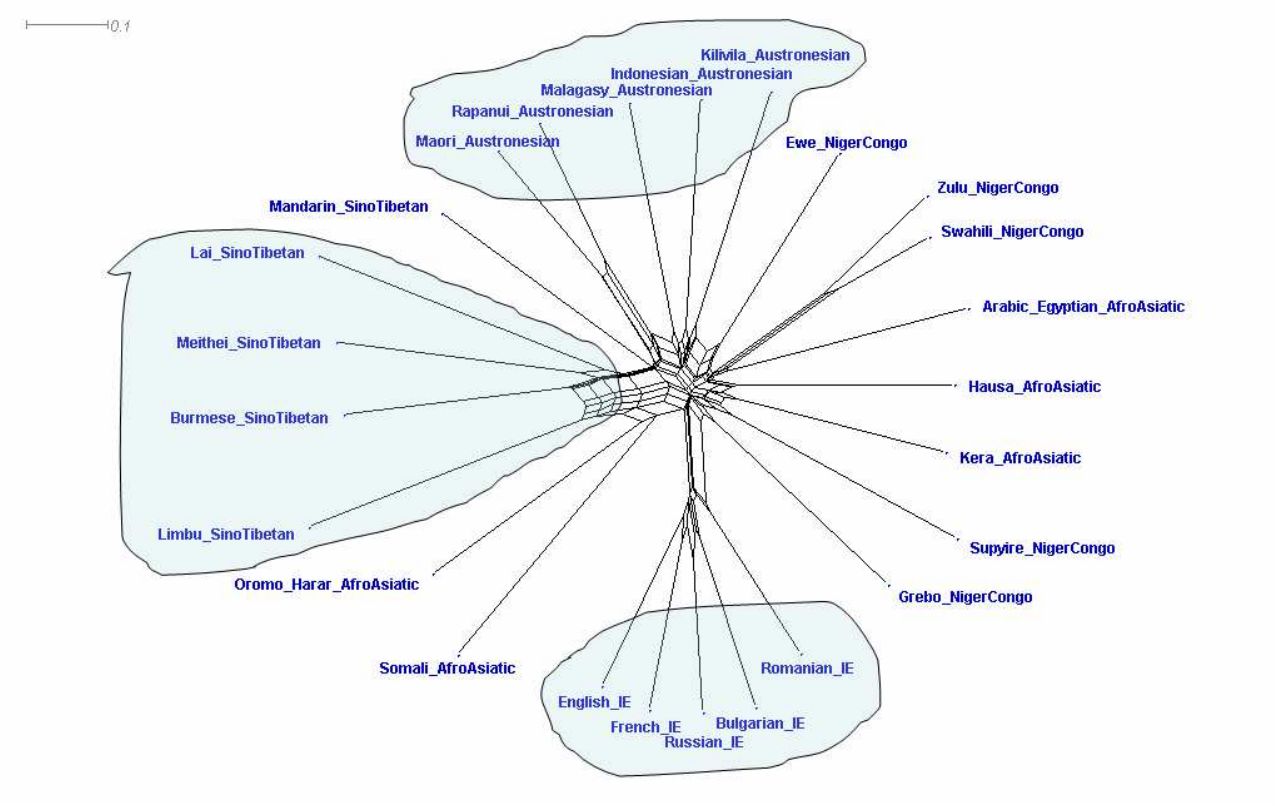


Figure 3.9: NNet using NormS distance.

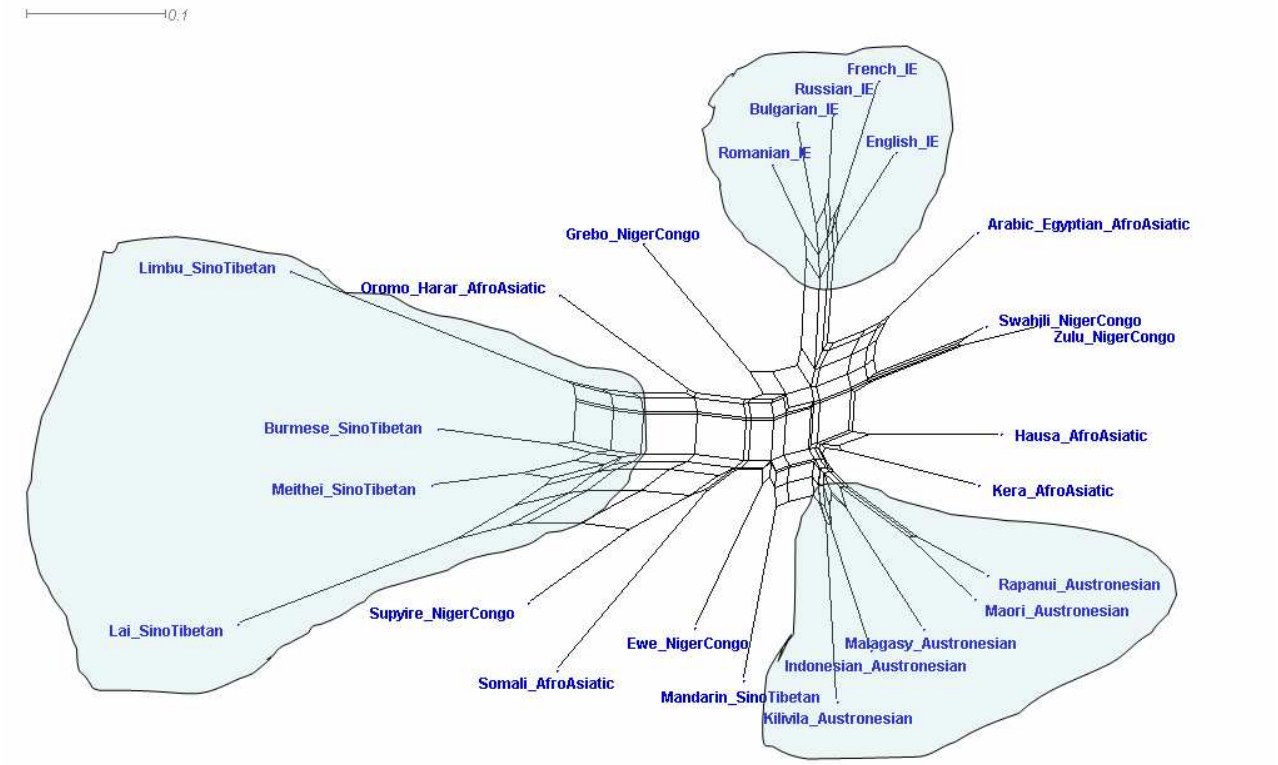


Figure 3.10: NNet using NormD distance.

Chapter 4

Good or bad data?

4.1 Selecting representative features

Obviously, the WALS dataset has much valuable phylogenetic information, as well as 'noise' from the point of view of evolutionary analyses. The question is how to detect which of the characteristics are phylogenetic relevant. Are there maybe features which are more stable than others all over the world? Are some of them only regionally relevant? Are there any linguistic features that are indicative of the overall structure of a language? Can highly consistent features also be genealogical representative ones? I have approached these questions by performing different methods.

Given any language L and feature F, WALS assigns to L and F a value $F(L)$ – an integer between 1 and at most 9 – in case that feature is defined for L. For the most cases (80%) where no value is assigned I put $F(L) := 0$. Based on this data, I will define a distance matrix describing the pairwise distances between all pairs of languages. Given a finite set N, then an $N \times N$ distance matrix D is a two-dimensional arrays with rows and columns indexed by the elements in N, containing the distances or dissimilarities, taken pairwise between any two elements of N. The entries of D are denoted by $D(L1, L2)$ where $L1, L2 \in$ N. A distance matrix is a symmetric $N \times N$ matrix, containing non-negative real numbers as entries and in general zeros along its diagonal: $D(L, L) = 0$.

The preparations of the data set resulted in a selection of 134 distance matrices D_F for each individual feature (I excluded features 3, 25, 95, 96, 97, 139, 140,

and 141 [1]), and one overall distance matrix D for all features together. The goal is to identify those features F that somehow harmonize with the overall distance matrix. The idea behind this goal is to find out which features are best predictors for the overall similarities between languages.

To investigate the relation between an individual feature F and the overall dataset, I compared each D_F matrix with the overall D matrix. Based on the features from WALS and a selection L of languages that have a good data coverage, I define the following $L \times L$ distance matrices.

First, I define for every single feature F and any two languages $L1$ and $L2$, the distance matrix for this one particular feature $D_F(L1, L2)$ as shown in Equation 4.1. In words, the distance is set at two (i.e. the languages are really different) when the languages are different and both are defined for this features in the WALS. The distance is zero when the languages are the same and both languages are defined for this features. The distance is defined as one when either (or both) of the two languages is not coded for this particular feature. I prefer this definition to a simpler definition (i.e. considering the cases with unavailable data as either all similar or all different) because I want to differentiate the cases for which there is missing information in the WALS.

$$D_F(L1, L2) = \begin{cases} 2 & \text{if } F(L1) \neq F(L2) and F(L1), F(L2) \neq 0, \\ 1 & \text{if } F(L1) = 0 or F(L2) = 0, \\ 0 & \text{else.} \end{cases} \tag{4.1}$$

Second, I define an overall distance matrix D for all features together. In this matrix, the distance between two languages is computed only on the basis of available information. To achieve this, I denote by F the set of all features and define $F(L)$ as the collection of all features F for which data is available for language L in the WALS, as shown in Equation 4.2. Then, I define the normalized distance between any two languages $L1$ and $L2$ as shown in Equation 4.3. In words, for the distance between $L1$ and $L2$ only those features are considered for which data is available for both languages (i.e. all cases where $D_F(L1, L2)$ = 1 are ignored). Then, the distances for all these available features are summarized, and divided by the number of available features. This procedure assures that the available data in the WALS is completely used. For every distance between two languages a different set of features is used, depending on the available information.

$$\mathbf{F}(L) := \{F \in \mathbf{F} : F(L) \neq 0\}, \tag{4.2}$$

$$D(L_1, L_2) = \frac{\sum_{F \in (\mathbf{F}(L_1) \bigcap \mathbf{F}(L_2))} D_F(L_1, L_2)}{\# \left(\mathbf{F}(L_1) \bigcap \mathbf{F}(L_2)\right)} \tag{4.3}$$

[1] This selection is explained in Section 4.3.

4.2 Methods

4.2.1 Mantel's test

Any $N \times M$ matrix X (N the index set for the rows and M that for the columns) can be viewed as providing records regarding a collection M of experiments (measurements) applied to candidates (objects) from a set N. Dividing M into two nonempty disjoint subsets M_1 and $M_2(\#M_1 > 0, \#M_2 > 0, \text{ and } \#M_1 + \#M_2 = \#M)$, it is natural to ask for the correlation between the results obtained by performing the experiments in M_1 and those obtained performing by the experiments in M_2. To answer this question, the first step is to derive from the $N \times M_1$ data matrix X_1 and the $N \times M_2$ data matrix X_2 corresponding dissimilarity (or distance) matrices, say D_1 and D_2. The main idea is that if two rectangular matrices contain concordant information, the distances derived from them should be significantly correlated. It is reasonable to ask, for example, whether genetic distance is correlated with geographical distance.

One can not use a simple correlation coefficient because the cases are not independent (e.g. the distance between two objects A and B is not independent of the distance between object A and another object C because A is involved in both). It was shown in [50] that, by calculating Kendalls W coefficient of concordance among matrices, Friedmans χ^2 statistics, and the associated P-value, one can obtain a measure of correlation, say R_0. Now, I permute the rows an arbitrary number k of times within one of the two matrices X_1 and X_2 and recalculate the correlation coefficients $R_1, \rightarrow, R_k$. If the original matrices had been correlated, the disruption caused by the permutations should reduce the correlation coefficient. The measure of significance is the number of times that the original correlation coefficient (R_0) was exceeded by the permuted values. For example, if one did 1000 permutations and only 1 of the resulting coefficients exceeded R_0, this would give us a significance of 0.001. Conversely, if the matrices were uncorrelated, there is no reason to assume that the permutations would decrease the correlation coefficient. They may indeed as well increase it. Here, I use the following variant of Mantels test:

Given the matrix $X = (F(L))F \in F, L \in L$, I construct for every $F \in F$ the extended matrix $(X \mid F)$ in which the Fs column $(F(L))L \in L$ occurs twice. I split this into the original matrix X and the one-column matrix $(F(L))L \in L$. Then, Mantels test is applied to the above defined overall distance matrix D and the matrix D_F . I used for my analyses the CADM (Congruence among distance matrices) software [51], and special routines had to be implemented in order to prepare the data for CADM (15 feature matrices per file plus the overall one for each file), and also to extract the relevant information out of the output files.

4.2.2 Coherence method

Next, I try to measure [2] for each of the feature matrices D_F corresponding to one of the 141 features, its degree of coherence with the overall matrix D. I propose to do this by calculating the triangle coherence index for each feature matrix D_F relative to D. To do this, I first define the excess of any two elements L_1, L_2 in a set N, with respect to an $N \times N$ distance matrix D, relative to any element $L_3 \in N$ (Equation 4.4).

$$exc_D(L_1L_2|L_3) = D(L_1, L_3) + D(L_2, L_3) - D(L_1, L_2) \tag{4.4}$$

The excess is, roughly spoken, the extra distance to be traveled between L_1 and L_2 when the route is taken via L_3, instead of taking the direct path from L_1 to L_2. If L_1, L_2, L_3 are the leaves of a weighed tree representing their distances, and if l is the median of L_1, L_2, L_3 in that tree, the excess $exc_D(L_1L_2|L_3)$ of L_1 and L_2 relative to L_3 is twice the distance between l and L_3 in that tree (cf. Figure 4.1).

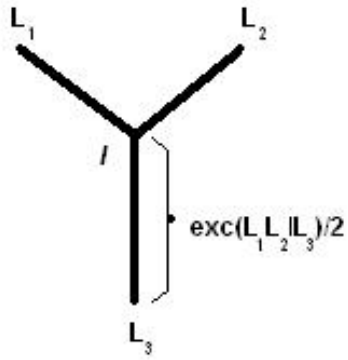


Figure 4.1: Excess value interpretation.

Based on that concept, the *triangle coherence index* ($\triangle_{coh\_index}$) is then defined as shown in Formulae 4.5. For every triplet of languages L_1, L_2, L_3, the quotient is taken of the excess relative to the overall matrix D and the excess relative to the single feature matrix D_F. The triangle coherence index for a feature F then is the minimum of all these quotients. A larger triangle coherence index will indicate a higher degree of coherence of a feature matrix D_F with the overall matrix D:

$$\triangle_{coh-index}(F) = sup(\lambda \in R_{\geq 0} : \ L_1, L_2, L_3 \in \mathbf{L}) \tag{4.5}$$

[2] Thanks to Andreas Dress for suggesting and describing the coherence and rank methods.

which implies

$$\Rightarrow exc_D(L_1L_2|L_3) \geq \lambda\, exc_{D_F}(L_1L_2|L_3) \tag{4.6}$$

My measure of coherence was first based on

$$\triangle_{coh-index}(F) = min\left(\frac{exc_D(L_1L_2|L_3)}{exc_{D_F}(L_1L_2|L_3)} \mid L_1, L_2, L_3 \in \mathbf{L}\right) \tag{4.7}$$

but because there are too many cases when this value goes down (due to the poor data coverage), a more appropriate measure was implemented (Equation 4.8).

$$\triangle_{coh-index}(F) = avg\left(\frac{1 + exc_D(L_1L_2|L_3)}{1 + exc_{D_F}(L_1L_2|L_3)} \mid L_1, L_2, L_3 \in \mathbf{L}\right) \tag{4.8}$$

4.2.3 Rank method

The rank $rk_{L_1}(L_2)$ of a language L_2 with respect to a language L_1 associated with the distance matrix D is defined by:

$$rk_{L_1}(L_2) = rk^D_{L_1}(L_2) := \#\{L_3 \in \mathbf{L}|D(L_1, L_3) \leq D(L_1, L_2)\}, (L_1, L_2 \in \mathbf{L}). \tag{4.9}$$

In other words, for every element L_1, I am counting the elements that have a distance measure relative to L_1 smaller or equal than the distance measure between L_1 and L_2. Metaphorically speaking, I am looking for the elements that are at least as good a friend of L_1 as is L_2 [3, 25]. This map yields a new matrix, the rank matrix:

$$R_D(D(L_1, L_2)), \forall L_1, L_2 \in \mathbf{L} \tag{4.10}$$

which is not necessarily symmetric, but has in general the value 1 along its diagonal. For example, the distance matrix in Table 4.1 will be transformed in the

0	4	2
4	0	1
4	1	0

Table 4.1: Sample distance matrix

following rank matrix (Table 4.2).
Next, I consider for each feature F, the sets of elements (languages) that share the same feature value:

$$\mathbf{L}(L, F) = \{L' \in \mathbf{L}|F(L) = F(L')\} \tag{4.11}$$

1	3	2
3	1	2
3	2	1

Table 4.2: The rank matrix associated

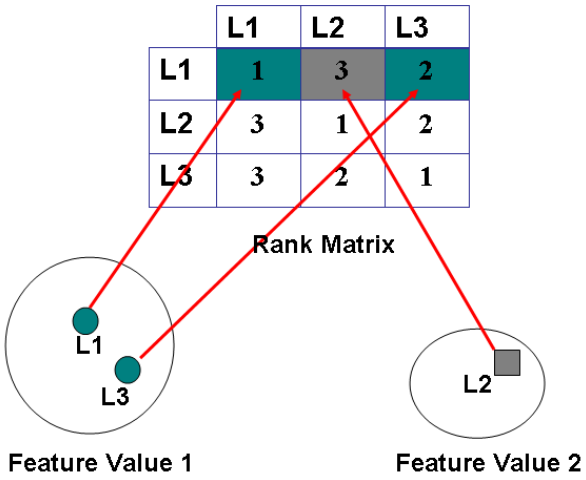


Figure 4.2: This feature *correctly* maps to the rank associations (for L1) because it groups together the L1's best friend (L3) and L2 is more far apart.

That means, that for every feature with X possible values, and a set of languages L, I may obtain a maximum number of clusters X, with various cardinality for each grouping. Now, I am trying to map this clusters into the rank matrix by measuring how well these clusters are identified by the rank matrix (Figure 4.2).

If the feature F fits well into the overall data structure encoded by the distance matrix D, the relative ranks $rk_L(L')$ of the languages L' in the subset $\mathbf{L}(L, F)$ should be significantly smaller for any given $L \in \mathbf{L}$ than the ranks $rk_L(L")$ of the languages $L"$ in the complement $\mathbf{L} \setminus \mathbf{L}(L, F)$ of $\mathbf{L}(L, F)$. Consequently, I propose the following measure of fitness using the ranking procedure:

$$\rho(F) := \frac{1}{\#\mathbf{L}} \sum_{L \in \mathbf{L}} \frac{\sum_{L' \in \mathbf{L}(L,F)} rk_L(L')}{\binom{1+\ \#\mathbf{L}(L,F)}{2}} \tag{4.12}$$

4.3 Conclusions

To test the three methods I decided to select the most dated languages from WALS, and also choose one language per genus in order to have a good distribution of the languages all over the world. Further, I also had to select a subset of the available features. From the 141 features in WALS I excluded features concerning sign languages (139 and 140), the features that replicates data from another maps (3, 25, 95, 96, and 97), and the feature dealing with writing systems (141). Having the ranked list of 150 languages (after this the number of data points per language decreased dramatically), I randomly divided this in three datasets (50 languages per dataset) to obtain a uniform distribution of the number of data points per dataset (please note that even so, many languages had a poor data coverage).

I run each of the three methods for each dataset and the ranked results are presented below, grouped by the methods (for lack of space, only the first 10 ranked feature numbers are presented).

Wichmann and Saunders performed a different analysis with the same purpose [75]. Using their formula for calculating p-values (implemented in [74]) for each genus/feature dataset in WALS, and by averaging the p-values found for each feature they generated a ranked-list of features where the (averaged) p-value determines the rank-order of the corresponding feature in terms of its usefulness for genealogical analyses (Table 4.6), cf. Equation 4.13 - 4.15, where k = number of possible values for a given feature, n = the number of languages in the genus, and r = the number of times that the most significantly feature occurs.

$$p(n, k, r) = \frac{C(n, k, r)}{k^n} \tag{4.13}$$

Feature Number	Mantel Coefficient	Feature Description
83	0.421	Order of Object and Verb
85	0.386	Order of Adposition and Noun Phrase
81	0.343	Order of Subject, Object and Verb
86	0.302	Order of Genitive and Noun
49	0.263	Number of Cases
88	0.239	Order of Demonstrative and Noun
98	0.238	Alignment of Case Marking of Full Noun Phrases
51	0.237	Position of Case Affixes
50	0.234	Asymmetrical Case-Marking
102	0.231	Verbal Person Marking

Table 4.3: Ranking of WALS features using the Mantel method

Feature Number	Coherence Coefficient	Feature Description
11	0.505	Front Rounded Vowels
18	0.475	Absence of Common Consonants
73	0.447	The Optative
19	0.432	Presence of Uncommon Consonants
107	0.432	Passive Constructions
82	0.422	Order of Subject and Verb
6	0.421	Uvular Consonants
10	0.408	Vowel Nasalization
7	0.406	Glottalized Consonants
80	0.406	Verbal Number and Suppletion
13	0.402	Tone
44	0.400	Gender Distinctions in Independent Personal Pronouns

Table 4.4: Ranking of WALS features using the coherence method

Feature Number	Rank Coefficient	Feature Description
11	1.306	Front Rounded Vowels
18	1.372	Absence of Common Consonants
83	1.637	Order of Object and Verb
85	1.804	Order of Adposition and Noun Phrase
107	1.835	Passive Constructions
82	2.007	Order of Object and Verb
130	2.025	Finger and Hand
73	2.069	The Optative
115	2.102	Negative Indefinite Pronouns and Predicate Negation
86	2.128	Order of Genitive and Noun
19	2.160	Presence of Uncommon Consonants
7	2.206	Glottalized Consonants

Table 4.5: Ranking of WALS features using the rank method

$$C(n,k,r) = k\binom{n}{k}Q(n-r,k-1,r+1) - \sum_{i-max(2,n-k(r-1)}^{n/r}\left((i-1)\binom{k}{i}\binom{n}{ir}\prod_{j=2}^{i}\binom{jr}{r}Q(n-ir,k-i,r)\right) \tag{4.14}$$

$$Q(n,k,r) = k^n - \sum_{i=r}^{n} C(n,k,i) \tag{4.15}$$

The ideal situation in this analyses would be that, irrespective of the method used, the same features would appear highly consistent with the overall dataset. The results however, in Table 4.7, showed no high correlations between the methods but at least they showed high correlation between the data sets for each method.

Because there is no complete consensus between the different methods [3], I can not draw any far-reaching conclusions about general consistency between individual features and the overall data structure of the WALS. However, for further linguistic data collection it is important to get at least a rough impression about what kind of features among the WALS data show a good consistency with the overall data structure. The following interpretations are not derived by strictly defined statistical tests, but by a manual inspection of the top ranked features from

[3]Based on conclusions from [22].

Feature Number	Feature Description
85	Order of Adposition and Noun Phrase
18	Absence of Common Consonants
90	Order of Relative Clause and Noun
88	Order of Demonstrative and Noun
11	Front Rounded Vowels
51	Position of Case Affixes
89	Order of Numeral and Noun
95	Relationship between the Order of Object and Verb and the Order of Adposition and Noun Phrase
33	Coding of Nominal Plurality
87	Order of Adjective and Noun
86	Order of Genitive and Noun
81	Order of Subject, Object and Verb

Table 4.6: Ranking of WALS features using Wichmann and Saunders's method

	Coherence	Rank
Mantel	.22	.22
Coherence		.65*

Table 4.7: Correlations between the three methods in Pearson's r. Significance $p < 0.001$ are indicated with a star

Mantel	Dataset1	Dataset2	Dataset2
Dataset1	1	.0.671108566	0.687585822
Dataset2		1	0.731586636
Dataset3			1

Table 4.8: Pearson correlations between the results for the datasets using the Mantel method

Coherence	Dataset1	Dataset2	Dataset2
Dataset1	1	0.901321099	.0.902387996
Dataset2		1	0.905746747
Dataset3			1

Table 4.9: Pearson correlations between the results for the datasets using the coherence method

Rank	Dataset1	Dataset2	Dataset2
Dataset1	1	0.765369979	0.762146186
Dataset2		1	0.705553075
Dataset3			1

Table 4.10: Pearson correlations between the results for the datasets using the rank method

the three methods [4].

All three methods put various word order features high on their ranking (82=subject-word order, 83=object-word order, 85 = ad-position order, 86 = genitive-noun order, 88 = demonstrative-noun order). This agreement between methods might be due to the fact that there are various features in the WALS about word order, which are all more or less significantly correlated with each other.

Besides word order features the rank and coherence methods suggest to include the ones about the inventory of strictly defined kinds of consonants like uvular consonants (feature 6), glottalised consonants (feature 7), the velar nasal (feature 9), and also more generally, like the absence of common consonants (feature 5 and 18), the presence of common consonants (feature 19). Also, both methods agree in ranking high features like: the passive (107), inflectional optatives (73), front rounded vowels (11), and tone (13).

In the methods that I have presented to measure consistency I have not used any information about genealogical relationships. Still, I wanted to see whether consistency might be a good predictor for genealogical relationships. To test this hypothesis, I constructed a sample of eleven families from the WALS, taking three languages out of each family. The choice of families and languages was completely driven by data availability. I wanted to know how well a particular selection of features would be able to distinguish pairs of related languages from pairs of unrelated languages in this sample. To investigate this, I constructed an overall distance matrix for the 33 languages sampled on the basis of all data in the WALS (I simply put 0 as distance if the languages were part of the same family and 1 if they belonged to different families). The distance matrices were compiled using the method as described in Equation 4.3 above and I constructed various distance matrices based on a selection of the features. Each selection of features was determined by the ranking of features as given by the various measures of consistency that I have discussed. For every method I subsequently considered the most consistent 25, 50, 75 and 100 features, and constructed distance matrices on that basis. As a control, I also considered the amount of available data as a ranking, constructing distance matrices on the basis of the best coded 25, 50, 75 and 100 features. In this way, I had sixteen different distance matrices for my test sample of 33 languages. All distances in such a matrix were then divided into two groups: one group with all distances between pairs of related languages and one group with all distances between pairs of unrelated languages.

I wanted to know whether the distances between related languages are generally smaller than the distances between unrelated languages. To investigate this, I used a t-test to determine the significance of the difference between the two groups. It turned out that the two groups were significantly different for all sets

[4]Thanks to Michael Cysouw for analyzing and producing the conclusions.

of features considered. Still, there are clear differences between the various selections. This can be seen by considering the t-test statistics themselves (not the significances). These t-test statistics are summarized in Figure 4.3. Selecting features by available data (the dotted line in Figure 4.3) gives some sort of baseline to compare my various methods again. The dotted line starts low and rises continuously, though the slope flattens the more features are considered. This indicates that I am able to get a better differentiation between related and unrelated language pairs the more features I consider, though there seems to be a level of differentiation that cannot be improved upon. Looking now at the various selection of features, I see that the best 25 features as selected by the methods all show a clearly stronger differentiation between related and unrelated pairs compared to taking the best coded 25 features. Taking the best 25 features from the Mantel method even gives roughly the same differentiation as given by considering all features together. Various of the other selections even improve on this. This indicates that by selecting a set of consistent features it is possible to improve the recognition of genealogical relationships compared to simply taking all available data.

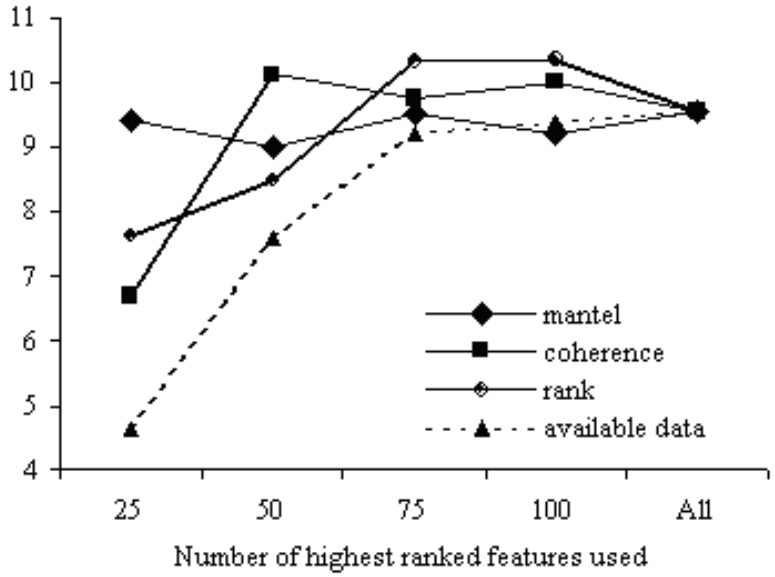


Figure 4.3: T-test statistic for the differentiation between related and unrelated pairs of languages for language distances as established by selected sets of features on the basis of the ranking of consistency.

A more appropriate approach in order to detect ‘good’ phylogenetic features would be to build the same 134 feature matrices (one per each feature). Instead

of building the overall data matrix D from WALS, one can build a good phylogenetic matrix (variants of building this matrix might include weighing each edge between languages and internal nodes as 1, count the family, subfamily and genera classification, etc.). Applying in the same way the methods presented above should result in a feature classification of how good the features map into the phylogenetic clustering. Again, if one would be interested in feature consistency from the geographical point of view, the overall distance matrix can be built starting with the geographical coordinates (as described in Section 2.4) (also available in WALS) and calculate the geographical distance between languages.

Of course, I redid the entire analyses for the WALSX recoded data. As expected, the same 'theme' came out as good fitting to the overall data, with some exceptions from the new binary features (Yes/No) that 'climb' up compared with the original ones.

I have performed additional phylogenetic analyses, in order to observe the influence of selecting different sets of features, based on each method's ranking. I selected the set of 33 languages with the most data points grouped by families, e.g. there should be at least 2 languages per family and at most 4 languages per family. I produced the distance matrices associated with a selection of best 25, 50, 75, and 100 features using the relative Hamming measurement (cf. Section 3.2.1.1). The matrices were then used in NNet algorithm and the most interesting outputs are presented below.

First, I analyzed the results obtained by using the best 25 ranked features from each method. All methods successfully recognized the clustering of languages from Nakh-Daghestanian and Trans New Guinea families. The rank method (Figure 4.4) is failing to group the Indo-European family as a cluster but amazingly, it recognizes the Australian and Austronesian families. The middle position in the comparison with the other two methods might be explained also by the structure of the network, which is mostly unresolved. The coherence method, which is ranked worst for the 25 feature selection by the plot in Figure 4.3, also produced an unstructured network and has no improvements in the classifications. On the other hand, the Mantel method is producing a well defined network and it is the only one that clusters the Altaic languages together, as well as partially the Niger-Congo languages, while the Indo-European family forms a strong cluster.

As the plot in Figure 4.3 suggested, the coherence selection of the best 50 features should provide a great improvement. Indeed, as depicted in Figure 4.7, the Indo-European family has now a well defined cluster, the Austronesian languages are grouped together, as well as the ones that belong to the Nilo-Saharan and Australian families.

The selection of the best ranked 100 features for the same coherence method, produces a well structured network, with improvements regarding the Altaic, Uralic, Sino-Tibetan and Trans New Guinea families (Figure 4.8). Nevertheless, by us-

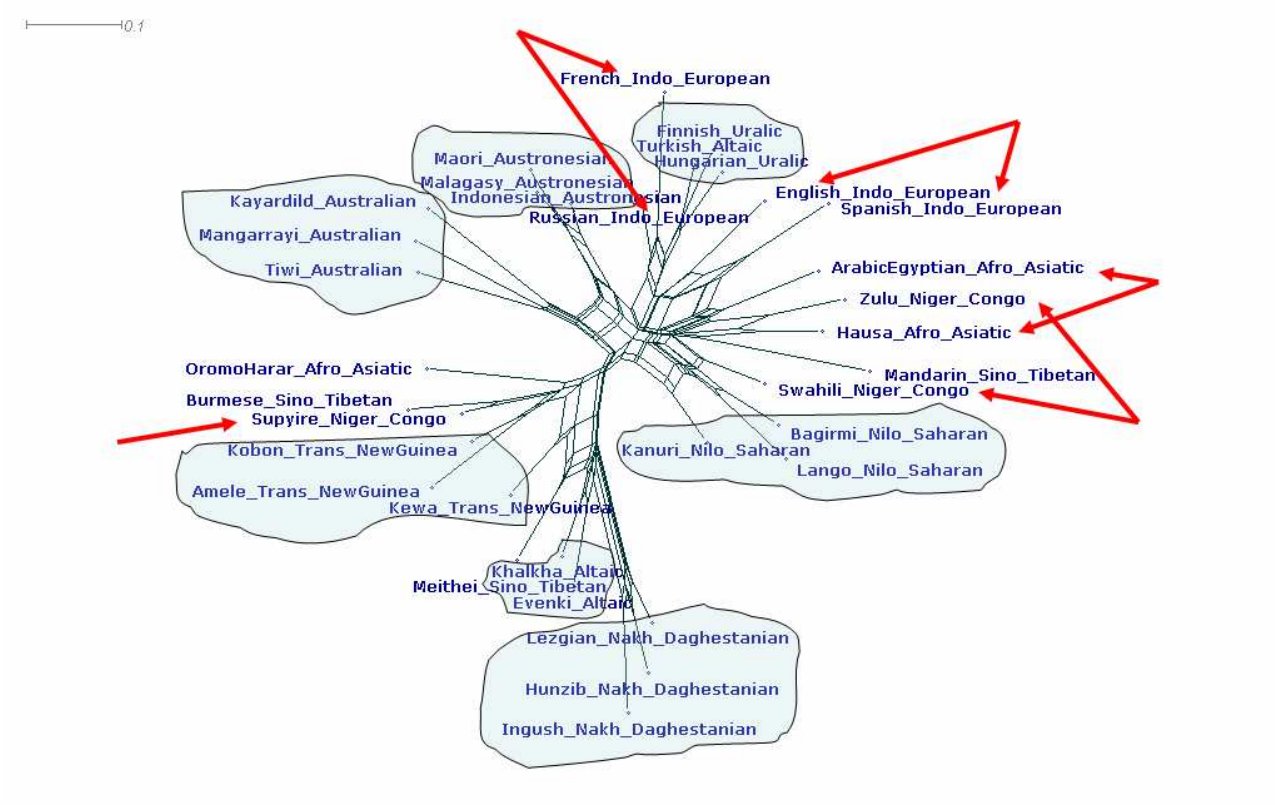


Figure 4.4: NNet using the best 25 features from the rank method.

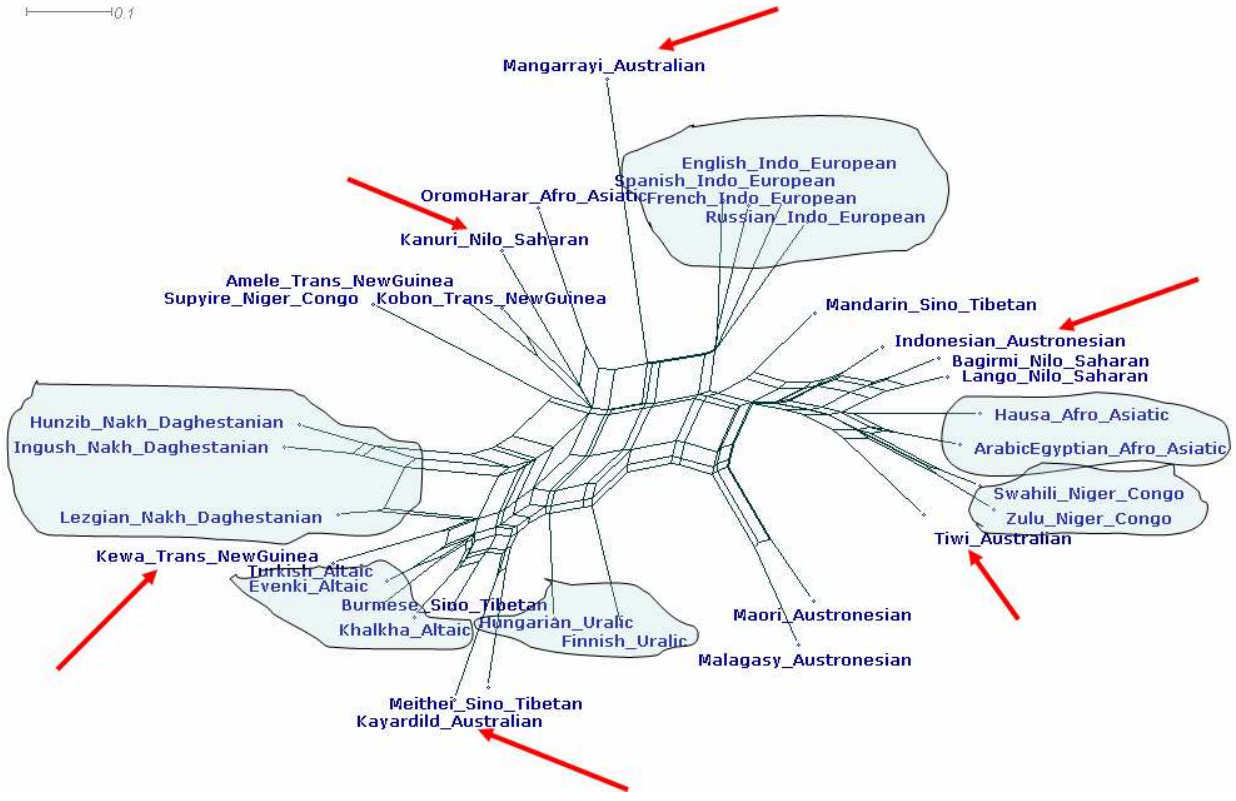


Figure 4.5: NNet using the best 25 features from the mantel method.

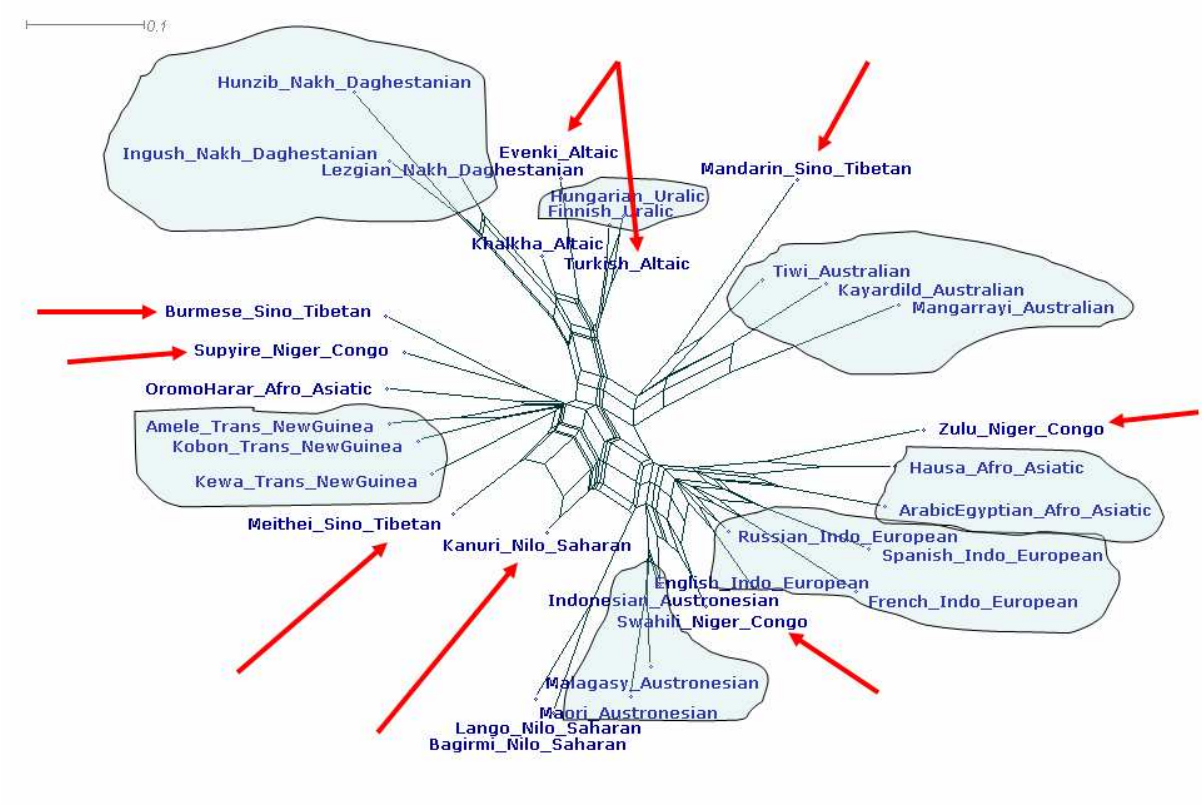


Figure 4.6: NNet using the best 25 features from the coherence method.

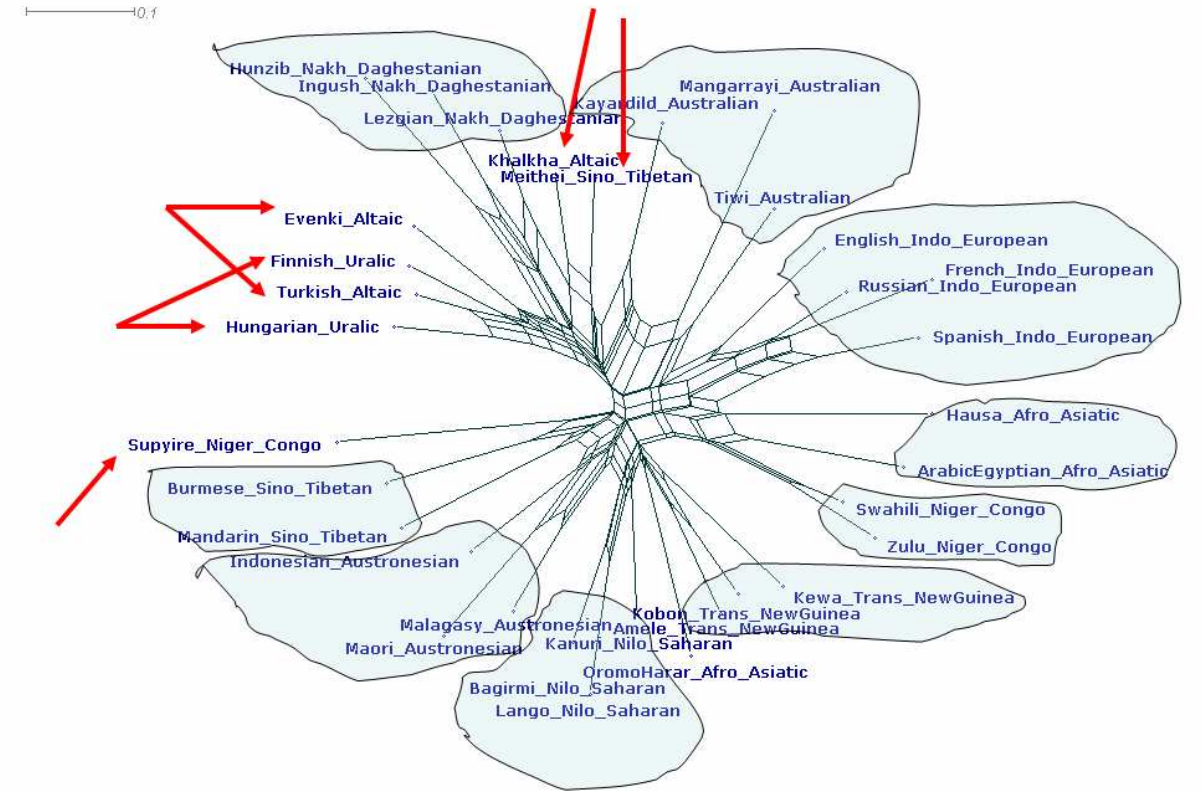


Figure 4.7: NNet using the best 50 features from the coherence method.

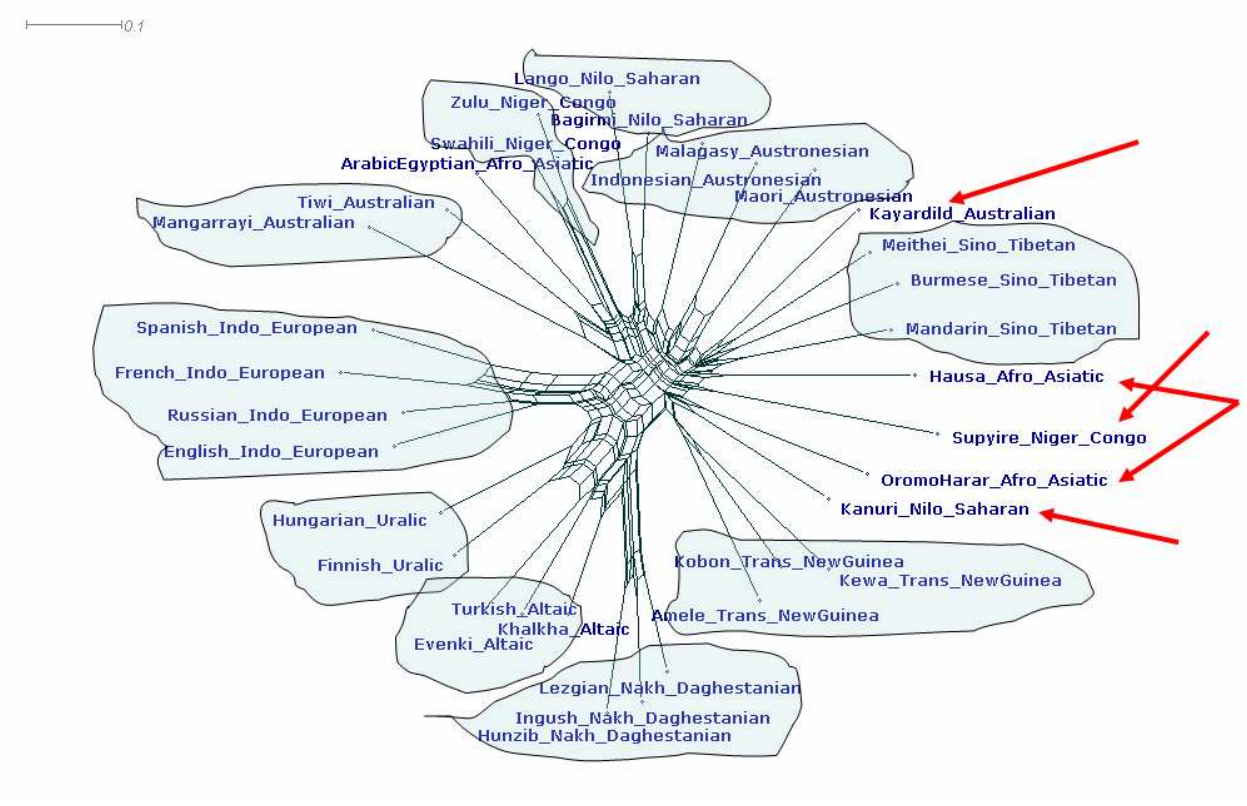


Figure 4.8: NNet using the best 100 features from the coherence method.

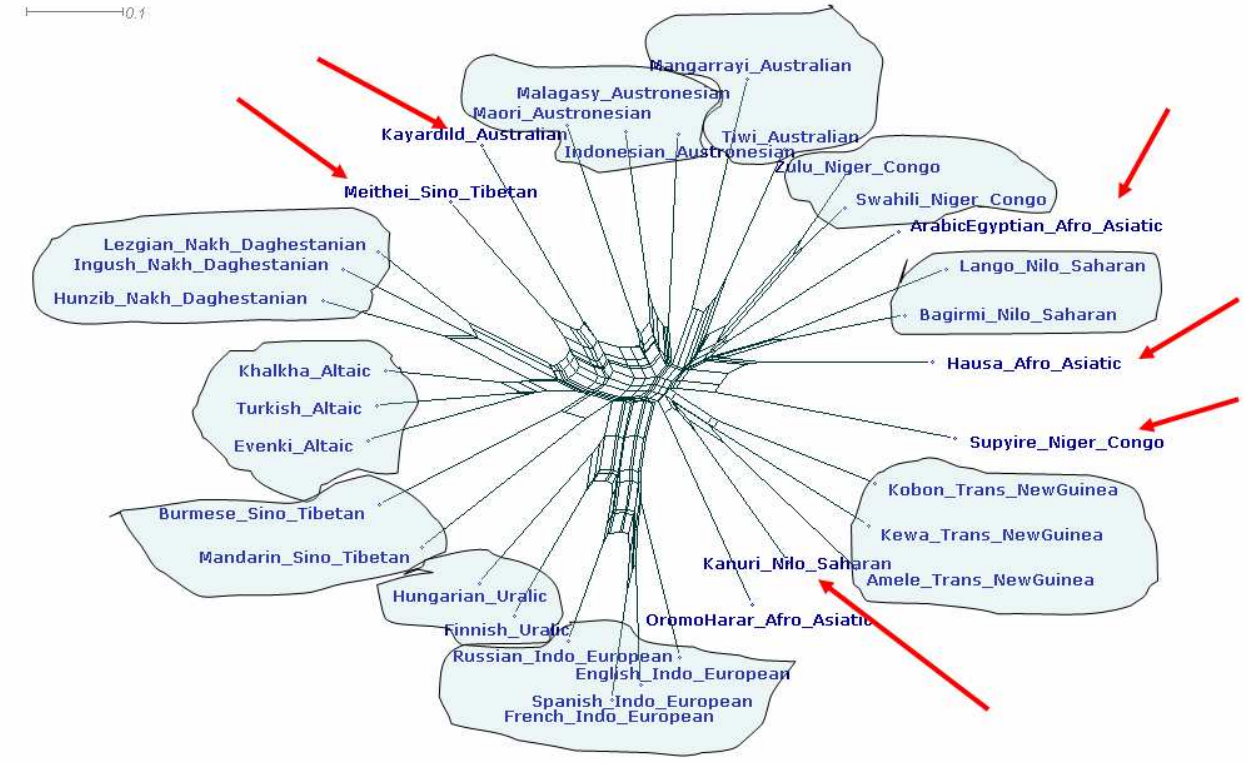


Figure 4.9: NNet using all features.

ing all features (the method is not important anymore as the dataset is the same), the resulting picture fails to cluster the Australian, Nilo-Saharan, Sino-Tibetan and partially Afro-Asiatic families, suggesting that the last ranked features added might contain noise or that they are just not phylogenetic representative.

The phylogenetic results presented in Figures (4.4 - 4.9) show a good correlation with the plot in Figure 4.3 and also indicate what I originally believed that each method manages to pick up signals from features consistencies, but these signals have different meanings.

Chapter 5

Phylogenetic Algorithms

5.1 Phylogenetic reconstruction algorithms

5.1.1 A rank-based hierarchical classification

5.1.1.1 Introduction

The purpose of this algorithm is the phylogenetic tree reconstruction from dissimilarity matrices[1]. This is a well-known problem and many algorithms were implemented so far [14, 67, 32], especially that and this is the usual situation, a dissimilarity matrix obtained from data will in general not be ultra-metric or additive; so it it has to be *tortured* to give a tree. In this case by torture I understand methodologies of modifying the original distance matrix [48], methods of choosing best alternatives in cases of tie decisions [71], or just complex algorithms in order to obtain the desired results [12, 26]. A standard task in cluster analysis is to associate, to any distance matrix D, a *Linnean hierarchy* $\aleph = \aleph(D)$, i.e. a collection $\aleph$ of subsets X_1, X_2, ... of the taxa set S, such that

$$A \cap B \neq 0 \Rightarrow \#(A \cup B) \leq max(\#A, \#B) \tag{5.1}$$

for all $A, B \in \aleph$. The central idea of Linnean hierarchies, and the associations with a partially ordered set, was used also by E. Haeckel to construct his rooted phylogenetic trees from the given knowledge of the "natural order" of the plant

[1] All the algorithms presented in this chapter were constructed following suggestions from Andreas Dress.

and the animal kingdoms just seven years after C. Darwin published *The Origins of Species* (cf. [24, 39]). The idea of deriving hierarchies from dissimilarity data was first introduced in the 1960's by A.F. Parker Rhodes and R.M. Needham [65], as well as J.D Apresjan [5, 6]. In a remarkable way, this procedure is not restricted to the conditions $D(x, y) = D(y, x) \geq 0$ and $D(x, x) = 0$ for all x, y in a set X, and it works for any such map D from $X \times X$ in any given linearly ordered set. They suggested to consider the collection

$$\hbar_D = \hbar_D(X) := \{A \subseteq X \mid a, b \in A, x \in X - A \Rightarrow D(a, b) < D(a, X)\} \quad (5.2)$$

of the subsets A of X for which, for any element a in A, any other element b in A is "closer" to a than any element $x \in X$ outside A and they observed that these subsets always form a Linnean hierarchy.

I produced a method which associates an ultra-metric matrix to an arbitrary real matrix, by an iterative procedure and I detected the hierarchies from the rank matrix. If the original matrix has no structure, the expected ultra-metric matrix is trivial, and gives rise to a star tree. This method, instead, uses ranks matrices. Why ranks? I do not have much faith in the precise values of a dissimilarity matrix, but I like to believe that the order of the these values is significant. The rank methods used the dissimilarity values to build a new, clearer matrix (the rank matrix), on which further analyses are performed. Rank methods have played an important role in statistics [48]. In [3, 25] it is described their utility in phylogenetic reconstruction. By its nature, the method that I shall describe does not give branch lengths, but only tree topology.

5.1.1.2 Method

The mathematics and many extensions are presented in [25], while a more recent paper describes the utility and applications as well as the program that was implemented to deal various situation [3]. One defines a rank matrix R of size m as an $m \times m$ matrix of integers such that: consider the i-th row of R, for $1 \leq i \leq m$. The entry R_{ij} is required to be equal to the number of entries R_{ik} such that $R_{ik} \leq R_{ij}$.

With this definition of rank, every rank matrix (for which any diagonal entry is not larger than any entry on its row) determines a rooted tree, but the current implementation provides only the clusters (hierarchies) found in a parenthesis format. Given an arbitrary real $m \times m$ matrix D, one produces a rank matrix $R = R(D)$ by : R_{ij} is the number of entries D_{ik} such that $D_{ik} \leq D_{ij}$. Usually, the hierarchies obtained by this algorithm are poor, in the way that they do not offer a full description of the clusters. For example, for 4 objects A, B, C, and D, one might obtain the following hierarchies, depending on the actual distance matrix: {A}, {B}, {C}, {D}, {A, B, C, D}. These hierarchies can not imply an objects

clustering, less a tree. To solve this, the following alternative is proposed. While associating the rank matrix $R(D)$ with a dissimilarity D gives rise to a map

$$R : \mathbf{R}^{X\times X} \to R^2(X) : D \mapsto R(D), \tag{5.3}$$

any map

$$T : R(X) \times R(X) \mapsto \mathbf{R} \tag{5.4}$$

allow us to define a map

$$D_T : R^2X \mapsto \mathbf{R}^{X\times X} : \bar{R} \mapsto D_T(\bar{R}) \tag{5.5}$$

in the opposite direction. Obviously, one may reiterate these procedures, thus deriving a whole family of dissimilarities $D^l : X \times X \to R(l = 0, 1, ..)$ from any dissimilarity $D \in R^{X\times X}$, for all $l \in N_0$. The experiments show that these iterations always run into a cycle, i.e. there exists a unique smallest number $i \in N$ such that $D^i = D^j$ holds for some $j \in N$. In my implementation I used as D_T the distance formulae from Spearmann and Kendal to calculate the new distances between the objects (rows) in the matrix.

$$D_T(i, j) = \sum_k (R(i, k) - R(j, k))^2 \tag{5.6}$$

$$D_T(i, j) = \frac{1}{N(N-1)} \sum_{a=1}^{N} \sum_{b=1}^{N} I_{i,j}(a, b) \tag{5.7}$$

where

$$I_{i,j}(a, b) = \begin{cases} +1 & \text{if } R(i, a) < R(i, b) \text{ AND } R(j, a) < R(j, b) \\ +1 & \text{if } R(i, a) > R(i, b) \text{ AND } R(j, a) > R(j, b) \\ -1 & \text{if } R(i, a) > R(i, b) \text{ AND } R(j, a) < R(j, b) \\ -1 & \text{if } R(i, a) < R(i, b) \text{ AND } R(j, a) > R(j, b) \\ +0 & \text{else} \end{cases} \tag{5.8}$$

Tests also showed that iterating until the cycle (convergence) is achieved produce much better results, because the hierarchies obtained are well defined and structured. Amazingly, the results showed that the cycles have even a step of 1, or a step of 4

$$D_T^i = D_T^{i+1} \text{ or } D_T^i = D_T^{i+4}, \tag{5.9}$$

but I could not find a proper explanation for the 4 step results. For example, in Figure 5.1, the initial rank matrix constructed for four objects is used, together with Spearmann's formula in Equation 5.6 to construct a new distance matrix, which is again reiterated. In the last step of the iteration, the distance matrix produces the same rank matrix as the initial one, therefore the cycle must stop, concluding in a step-cycle of four (note that the derived distance matrices will always have 0 on the diagonal and they will always be symmetric).

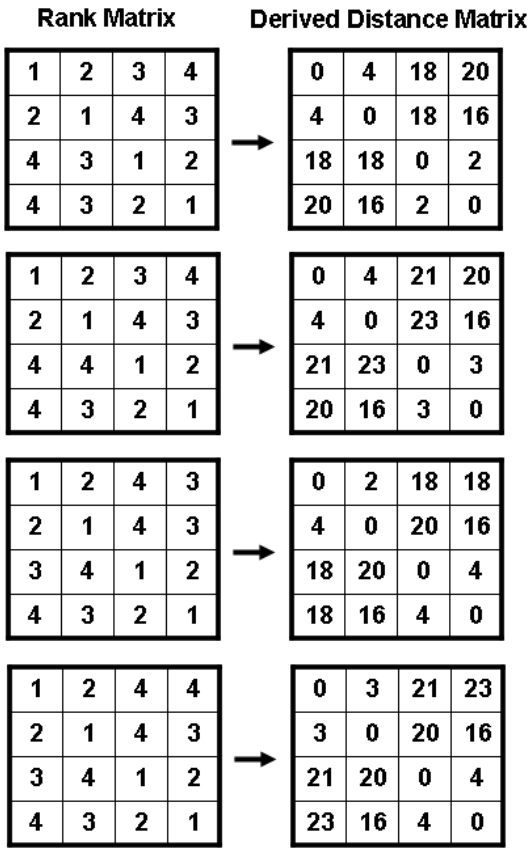


Figure 5.1: Iterations using Spearman's formula. After 4 iterations, the process is entering in a cycle.

Nevertheless, it is actually worth analyzing each cycle's output, as the evolution of size and correctness of the hierarchies might show interesting information. In some cases, there might be the situation that even after iterating, the hierarchies might not be fully descriptive.
That is why, I introduced a new improvement, a divide-and-conquer approach. Mainly, after convergence is achieved, I am looking only at the largest two disjoint hierarchies, let say H_1 and H_2, and disregard the other ones. That is also because, and extensive tests proved it, one should have greater confidence in the largest hierarchies, and be at least sceptical about the small ones. Then, for each of H_1 and H_2, I can select the objects that form each hierarchy (remember that they are disjoint), and restart the entire algorithm for each of them, by constructing from the original distance matrix, the distances matrices associated to H_1 and H_2 respectively (cf. Figure 5.2).

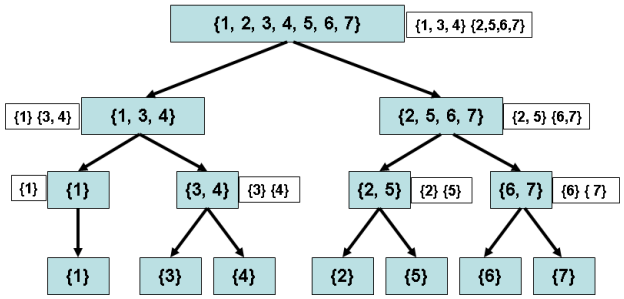


Figure 5.2: Divide and conquer search for hierarchies.

5.1.1.3 Applications

In this section I present an interesting example of a 'problematic' distance matrix, and I compare the results from the rank algorithm with other phylogenetic reconstruction algorithms based on distance matrices. The input distance matrix is based on the picture in Figure 5.3, in which the distance between the two clusters $\{$A, B, C$\}$ and $\{$a, b, c$\}$ is 2, which is equal with the distance between A and C, or a and c.

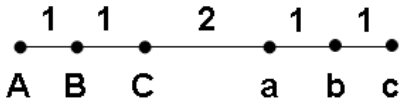


Figure 5.3: Sample input data for algorithms comparisons .

Then, the associated distance matrix is the one showed in Table 5.1, as the length

	A	B	C	a	b	c
A	0	1	2	4	5	6
B	1	0	1	3	4	5
C	2	1	0	2	3	4
a	4	3	2	0	1	2
b	5	4	3	1	0	1
c	6	5	4	2	1	0

Table 5.1: Distance matrix associated

of the path between the points
Using the Neighbour-Joining, UPGMA and DQuartets algorithms implemented in SplitsTree package, I obtained the following pictures displayed in Figures 5.4 - 5.8.

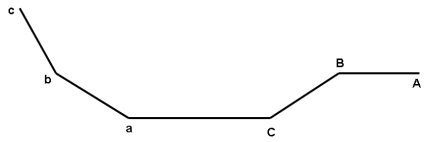


Figure 5.4: The result from the NJ algorithm.

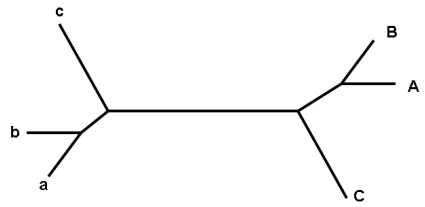


Figure 5.5: The result from the UPGMA algorithm.

Of course, this is not what I really wanted. Even if NJ (Figure 5.4) showed the reality, I am more interested in detecting the clusters {A, B, C} and {a, b, c} (of course, one can still interpret the NJ result as two clusters). The DQuartets (Figure 5.6 - 5.8) is not showing the correct clusters even if various values of the threshold parameter are used. The UPGMA algorithm (Figure 5.5) interprets correctly the problematic distance matrix, but somehow the picture seems distorted (this algorithm proposes only tree topology, not the branch lengths). The rank algorithm is somehow in the middle, as it correctly finds the {A, B, C} and {a, b, c} clusters but as I mentioned, the algorithm does not make any assumptions on the length of the branches, but only on the typology (Figure 5.9).

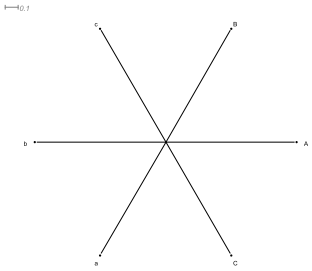


Figure 5.6: The result from the DQuartets algorithm.Thresholds = 0.0.

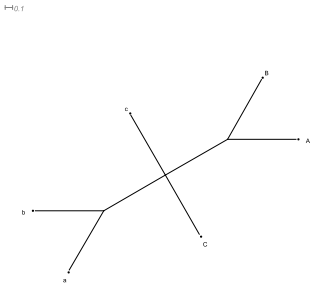


Figure 5.7: The result from the DQuartets algorithm.Thresholds = 1.0.

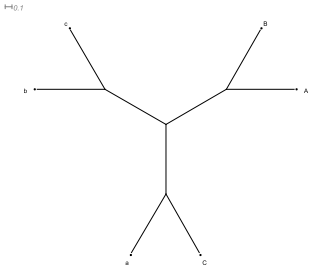


Figure 5.8: The result from the DQuartets algorithm.Thresholds = 2.0.

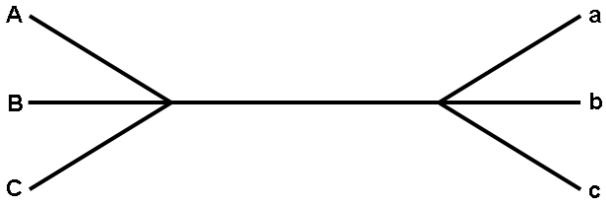


Figure 5.9: The result from the rank algorithm.

5.1.2 A Distance-Quartet Puzzling algorithm

Given a distance matrix, it seems reasonable to expect that the two species with the smallest distance will have diverged most recently. In addition, the next smallest distance in the matrix should indicate which two species diverged just before that, and so forth. For example, if species *a* and *b* are the closest[2], and species *a* and *c* are the next closest, the conclusion would be that species *a* and *b* diverged most recently, and that an ancestor of these two diverged from species *c* just before that.

This clustering approach can be used for phylogenetic tree reconstruction. Once the two closest species have been identified, one can define the distances between their virtual ancestor and all the remaining species. If species *a* and species *b* are the closest and species *x* is their ancestor, then one can define the distance between that ancestor *x* and any other species *c* using one of the following formulae:

$$D_{min}(x,c) = min(D(a,c), D(b,c)) \tag{5.10}$$

$$D_{max}(x,c) = max(D(a,c), D(b,c)) \tag{5.11}$$

$$D_{avg}(x,c) = \frac{1}{2}(D(a,c) + D(b,c)) \tag{5.12}$$

The algorithm using (5.10) is called 'nearest-neighbour clustering' or 'single linkage method', the one using (5.11) is referred to as 'furthest-neighbours clustering' or 'complete linkage method', and (5.12) is often called 'weighed pair group method using arithmetic averages', or WPGMS [44].

5.1.2.1 Distance-Quartet Puzzling

Given a phylogenetic X-tree T and four leaves a, b, c, d such that exactly one edge in T separates a, b from c, d, the length of the unique internal edge of the induced $\{a,b,c,d\}$-Tree is given by

$$w_T(ab|cd) := \frac{1}{2}\left(D_T(a,d) + D_T(b,c) - D_T(a,b) - D_T(c,d)\right) \tag{5.13}$$

$$= \frac{1}{2}\left(D_T(a,c) + D_T(b,d) - D_T(a,b) - D_T(c,d)\right) \tag{5.14}$$

where $D_T(x,y)$ denotes the sum of edge weights on the path that connects x and y, $x, y \in X$.

[2]This pair is often named a *cherry*.

Thus, given a not necessary tree-like distance D, this suggests to consider the number

$$w_D(ab|cd) = \frac{1}{2}(max(D(a,c)+D(b,d), D(a,d)+D(b,c)) - D(a,b) - D(c,d)) \tag{5.15}$$

as a weight of the quartet tree $ab|cd$ relative to D as depicted in Figure 5.10.

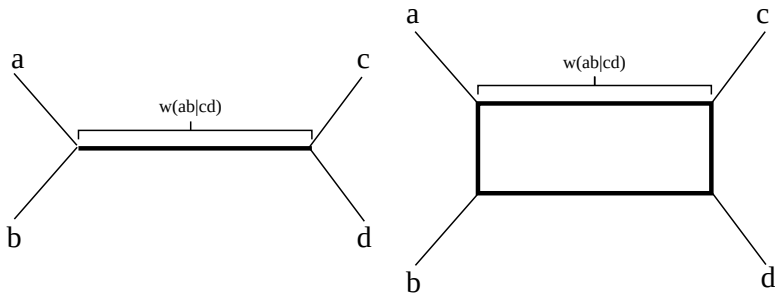


Figure 5.10: Weight calculations for a) tree-like data and b) non tree-like data.

5.1.2.2 Linkage algorithms for weighed quartets

I now consider

$$W_{max}(ab) = max\{w(ab|cd) \mid c,d \in X \setminus \{a,b\},\ c \neq d\}, \tag{5.16}$$

that denotes the maximum weight of all quartets separating a, b from any two other elements. Now, the procedure is continued as follows: first, one finds the pair $\{a,b\}$ contained in the quartet with maximal weight.

$$(a,b) = \underset{(x,y)\in X^2}{\arg\max}\{W_{max}(xy)\} \tag{5.17}$$

Then, assuming that the pair (a,b) forms a cherry in the phylogenetic X-tree that one plans to construct, one collapses the leaves a and b into one element $\{a,b\}$ and update the weights according to one of the formulae:

$$w(\{a,b\}c|de) \in \left\{ \begin{array}{l} min(w(ac|de), w(bc|de)), \\ max(w(ac|de), w(bc|de)), \\ avg(w(ac|de), w(bc|de)) \end{array} \right\}. \tag{5.18}$$

For whatever formula is used, one can now replace the two leaves by only one of them and define its distances to the remaining leaves as indicated in equation

5.18. The algorithm subsequently defines new clusters of leaves and distances, until only four leaves are left. For these four leaves, one chooses the quartet tree configuration with the highest weight among the three possibilities. Then, one builds the tree based on the information saved every time a leaf was deleted. The three variants produce, in the worst case, three different trees, but finding the same cluster in at least two of them, will suggest that this cluster has some phylogenetic relevance. Of course, this conclusion will be more convincing if it is found in all three trees.

Another variant is to build the weighted system as

$$W_{sum}(ab) := \sum_{\substack{c,d \in X \setminus \{a,b\} \\ c \neq d}} w(ab|cd) \quad \forall a, b \in X; a \neq b. \tag{5.19}$$

and to find the pair (a,b) that has the largest weight. For the update procedure, one can choose one of the three alternatives (5.18) and compare $W_{sum}(a'b')$ after performing the update, or just use

$$W_{sum}(\{a, b\}, c) := \max \begin{pmatrix} W_{sum}(a, c) \\ W_{sum}(b, c) \end{pmatrix}. \tag{5.20}$$

The other parts of the algorithm (collapsing of cherries and tree reconstruction) remain unchanged. One can explore which variant of the algorithm should be used. Thus, the analysis of the phylogenetic trees obtained by applying different variants on the same data set might be of interest in phylogenetic analysis (e.g. checking the Molecular Clock Hypothesis [53]).

5.1.2.3 Algorithm

In the algorithm, one keeps track of the pairs later forming the cherries. Given a set $X = \{$a, b, c, d, e, f$\}$ of taxa and a distance matrix D on X, we compute the phylogenetic X-tree as follows:

If (a, b) is the pair to be collapsed into $\{a, b\}$, one renames $\{a, b\}$ to a, push b onto the stack $Stack_{removed}$ of elements removed, and push a onto the stack of pairs $Stack_{pairs}$. Next, if the pair (d, e) is to be collapsed into d, the stacks contain: $Stack_{removed} = e, b$ and $Stack_{pairs} = d, a$. Whenever an element is pushed onto $Stack_{removed}$, it is also deleted from the initial data set. At the last step of this part, one remains only with four elements: a, c, d, and f and starts with the backtracking phase (Figure 5.11). If $(ac|df)$ forms the quartet configuration with the largest weight $w(ac|df)$, one first pops the stacks ($e \leftarrow pop(Stack_{removed})$, $d \leftarrow pop(Stack_{pairs})$) and replaces the leaf d in the already constructed phylogenetic X-tree by the cherry $\{d, e\}$. Analogously, the leaf a is replaced in the last step by the cherry $\{a, b\}$.

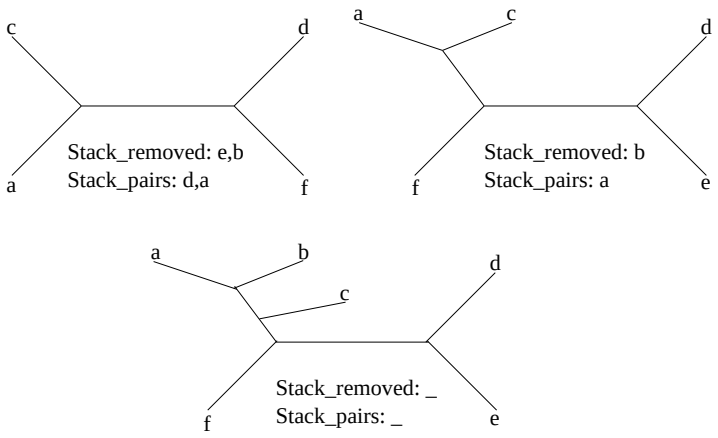


Figure 5.11: Backtracking phase for the phylogenetic X-tree reconstruction (Algorithm 1, lines 14-19) given a set $X = \{a, b, c, d, e, f\}$ of taxa.

5.1.2.4 Time and space complexity

I now give a detailed analysis for the time and space complexity of this Distance Quartet Puzzling algorithm.

Given a set X of taxa, there exist $3\binom{|X|}{4}$ quartets. Therefore, there are $3\binom{|X|}{4}$ weights to be computed (Algorithm 1, lines 1-5) that all have to be touched in the second loop (Algorithm 1, lines 6-8). The third loop (Algorithm 1, lines 9-16) is executed $|X| - 4$ times. In the $i - th$ iteration, the computation of the pair with maximal weight takes $\binom{|X|-i+1}{2}$ steps, the number of pairs of remaining taxa in X. The operations in Algorithm 1 lines 11-14 take constant time. In Algorithm 1, line 15 the update procedure (Algorithm 2) for weights of quartets containing a can be performed in $\binom{|X|-i-1}{3}$ steps. The reconstruction of the phylogenetic X-tree (Algorithm 1, lines 18-25) can simply be performed in linear time. Hence, the overall time complexity is:

$$\begin{aligned} T_{DistQ} &= O\left(\binom{|X|}{4} + \sum_{i=1}^{|X|-4}\left(\binom{|X|-i+1}{2} + c_1 + \binom{|X|-i-1}{3}\right) + n\right) \\ &= O(n^4) \end{aligned}$$

The overall space complexity is $O(n^4)$ due to the storage of the weights of all $3\binom{|X|}{4}$ quartets.

5.1.2.5 Applications

I applied the DistQ algorithm on a distance matrix constructed as in Section 3.2.1.1 for 12 languages grouped in 6 families (2 languages per family). I processed

input : Distance matrix D, set of taxa X
output: Phylogenetic X-tree

forall $\{x, y, v, z\} \in \binom{X}{4}$ **do**
  $w(xy|vz) \leftarrow$ `CalculateWeight`$(xy|vz)$
  $w(xv|yz) \leftarrow$ `CalculateWeight`$(xv|yz)$
  $w(xz|vy) \leftarrow$ `CalculateWeight`$(xz|vy)$
end
forall $\{a, b\} \in \binom{X}{2}$ **do**
  compute $W(ab)$
end
for $i \leftarrow 1$ **to** $\#(X) - 4$ **do**
  $(a, b) \leftarrow \arg\max_{x,y \in X}\{W(xy)\}$
  $X \leftarrow X - \{b\}$
  $\text{Stack}_{\text{removed}} \leftarrow$ `push`$(\text{Stack}_{\text{removed}}, b)$
  $\text{Stack}_{\text{pairs}} \leftarrow$ `push`$(\text{Stack}_{\text{pairs}}, a)$
  $a \leftarrow$ `CollapsePair`(a, b)
  $w \leftarrow$ `UpdateWeights`$(a, b|X|y)$
end
/* $V = \{a, b, c, d, u, v\}$, $E = \{\{a, u\}, \{b, u\}, \{c, v\}, \{d, v\}, \{u, v\}\}$ */
$(ab \mid cd) \leftarrow \arg\max_{x,y,v,z \in X}\{w(xy \mid vz)\}$
$\text{tree} \leftarrow treeFromQuarted(ab|cd)$
for $j \leftarrow \#(X) - 4$ **to** 1 **do**
  $y \leftarrow$ `pop`$(\text{Stack}_{\text{removed}})$
  $z \leftarrow$ `pop`$(\text{Stack}_{\text{pairs}})$
  $\text{cherry} \leftarrow$ `formCherry`(y, z)
  tree
  $\leftarrow$ `replaceTreeNodeZwithCherry`$(\text{tree}, z, \text{cherry})$
end

Algorithm 1: Distance quartet puzzling algorithm.

input : Taxa a, b, data set X and weighted system w
output: Updated Weights

forall $x \in X \setminus \{a, b\}$ **do**
  forall $y, z \in X \setminus \{a, b, x\}$ **do**
    $w(ax|yz) \leftarrow UpdateFunction(w(bx|yz), w(ax|yz))$
      where UpdateFunction $\in \{min, max, avg\}$
  end
end

Algorithm 2: Update Weights (a,b | X | w).

the matrix with the DistQ variants explained in Formulae 5.18 (using minimum), and in Formulae 5.1.2.2, and the trees obtained are depicted in Figure 5.12, and Figure 5.13 respectively. Compared with the results obtained by using the NJ algorithm (Figure 5.14) on the same distance matrix, the outputs of the two algorithms bear an obvious similarity to each other.

This algorithm as well as the NJ correctly clusters the languages from Indo-European, Altaic, Australian and Nakh-Daghestanian families, while the same problem for Austronesian and Afro-Asiatic families is encountered by both algorithms. Interestingly, the DistQ algorithm has nicely clustered the Indo-European, Altaic and Nakh-Daghestanian together, while NJ placed the Indo-European family opposite to the other two. Note that DistQ will always provide binary trees (because using the 'cherries' methodology) and it does not assume any information about the length of the branches as the output that the algorithm produces is just the simple tree Newick expression (without lengths).

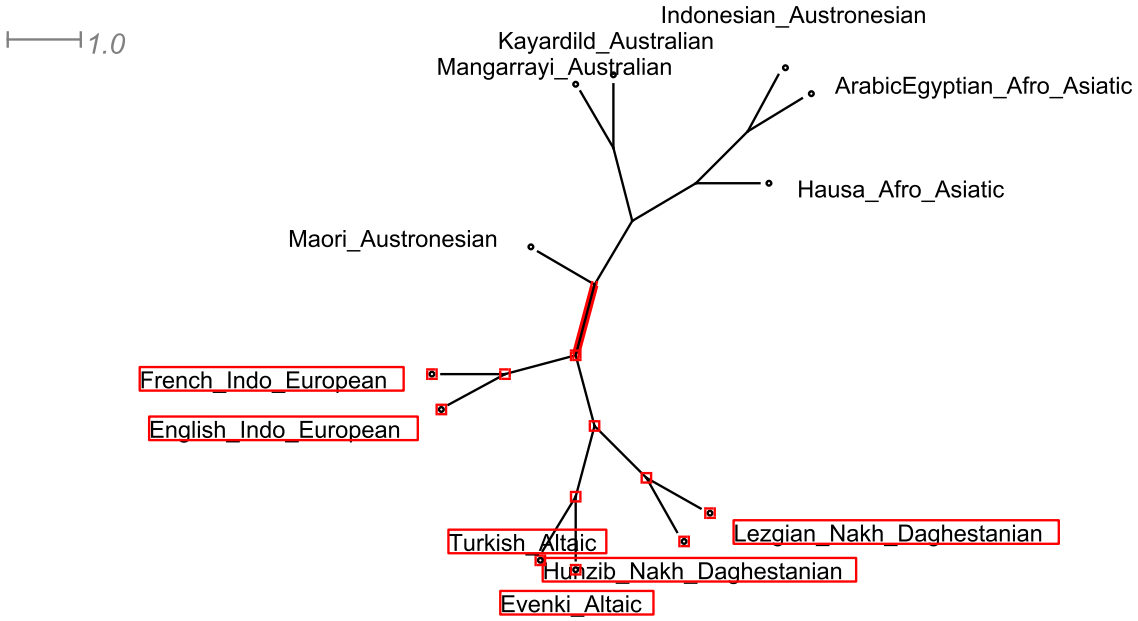


Figure 5.12: The result from the DistQ algorithm using the *minimum weight* variant.

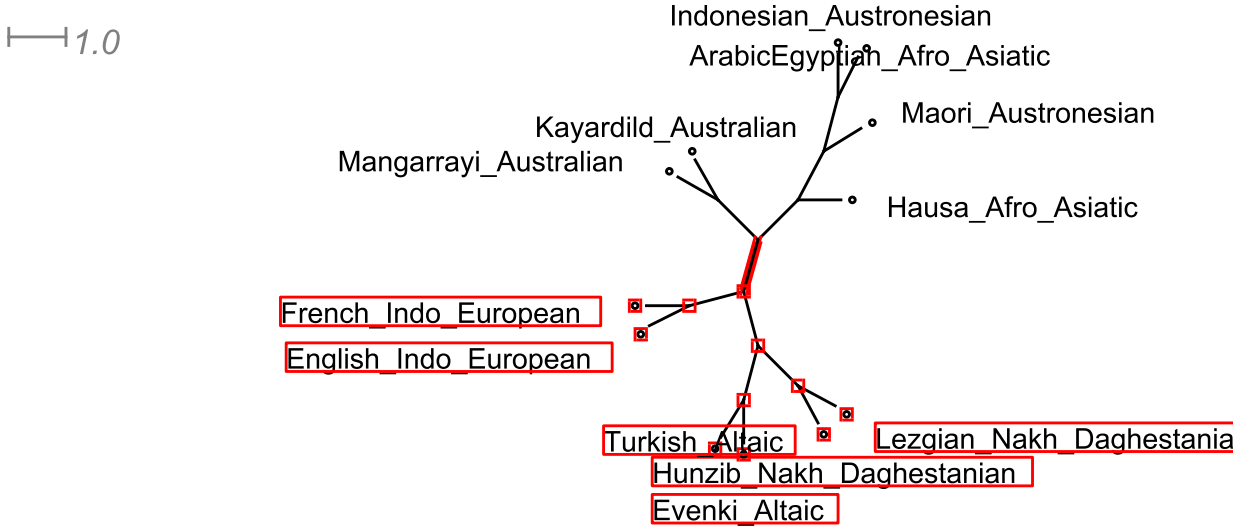


Figure 5.13: The result from the DistQ algorithm using the *global weight* variant.

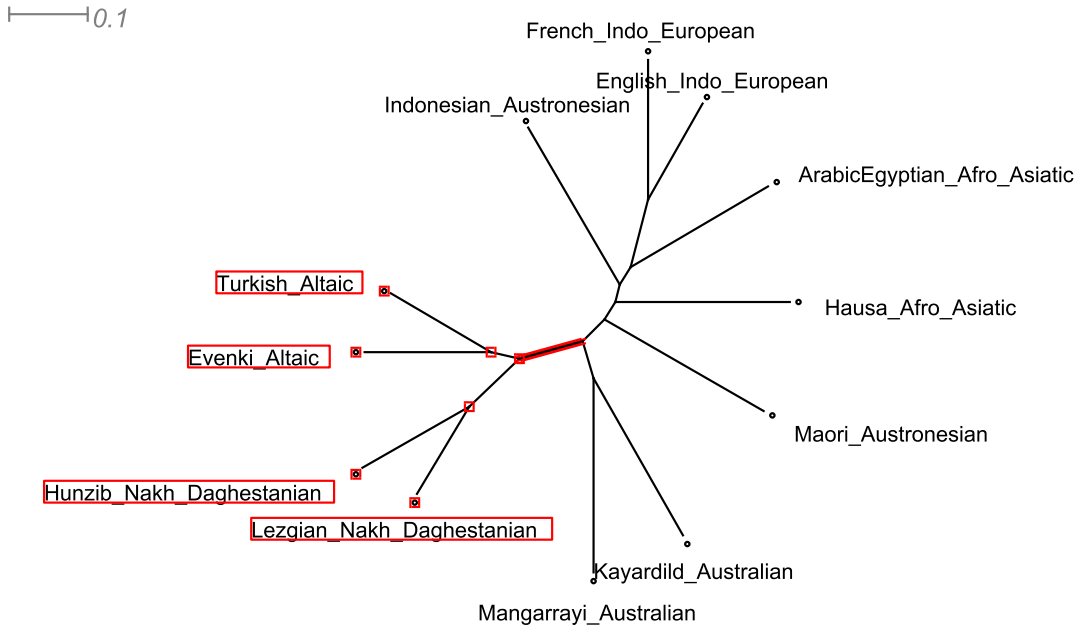


Figure 5.14: The result from the NJ algorithm.

5.2 The entropy algorithm

5.2.1 Description

Considering a system S consisting of a finite number of subsystems with finite state spaces $S_1, .., S_n$, and an assignment of an energy E(s) for each global state $s = (s_1, , s_n) \in S_1 \times .. \times S_n$ of the system S, the associated partition function

$$\Phi(T) = \Phi_E(T) := \sum_{s \in S} exp(-E(s)/T) \tag{5.21}$$

and the average energy of the system

$$E(T) := \frac{\sum_{s \in S} E(S) exp(-E(s)/T)}{\Phi(T)} \tag{5.22}$$

can be computed in a number of steps that grow linearly with n, with a constant depending on the tree-width of the associated interaction system (cf. [6]).

Given:

- a finite alphabet U, a dissimilarity map

$$D = U \times U \longrightarrow R_{\geq 0} : (\alpha, \beta) \mapsto D(\alpha, \beta) \tag{5.23}$$

with

$$D(\alpha,\alpha)=0, D(\alpha,\beta)=D(\beta,\alpha) \tag{5.24}$$

- a finite set $X \subseteq U^N$ of aligned sequences and an X-tree $\bar{T}=(V,E)$, i.e., a finite tree with vertex set V and edge set E without vertices of degree 2 for which X coincides with its leaf set

$$V_1(\bar{T}) := \{v \in V : \#\{e \in E : v \in e\} = 1\} \tag{5.25}$$

- and a map $\psi : X -> U$.

Then, one can consider the system S whose state space is the set of all extensions $\varphi : V -> U$ of ψ, and associate, to any such φ, its internal energy

$$E(\varphi) := \sum_{\{u,v\}\in E} D(\varphi(u),\varphi(v)) \tag{5.26}$$

Invoking concepts due to Boltzmann, one can then estimate the quality of the tree $\bar{T}$ relative to ψ by its average energy

$$E(\psi,\bar{T}) = E(\psi,\bar{T})_T = \sum_{\varphi} E(\varphi)p(\varphi) = \sum_{\varphi} E(\varphi)\frac{exp(-E(\varphi)/T)}{\Phi_E(\bar{T})} \tag{5.27}$$

or, as well, by the closely related entropy

$$entr(\psi,\bar{T}) = entr(\psi,\bar{T})_T := \sum_{\varphi} -ln(p_T^{\bar{T}}(\varphi))p_T^{\bar{T}}(\varphi) = ln(\Phi(T)) + E(\psi,\bar{T})/T) \tag{5.28}$$

of the probability distribution

$$p = p_T^{\bar{T}} := (p_T(\varphi))_\varphi = \frac{exp(-E(\varphi,\bar{T})/T)}{\Phi_E(T)} \tag{5.29}$$

Remarkably, using a simple recursive dynamic programming scheme (based on the fact that, after all, a tree has tree-width one), these terms can be computed algorithmically in linear time relative to $\#X$, with a constant depending on $\#U^2$, cf. [5]. So, considering a family of N maps $\psi_1,,\psi_N,: X -> U$ (corresponding to an X-labeled family of aligned sequences in U^N), one can associate to this - and compute for every X-tree $\bar{T}$ and every $T \in R_{>0}$ - the sum

$$\sum_{i=1}^{N} E(\psi_i,\bar{T}_i)_T \tag{5.30}$$

It is this sum that I suggest to use as a measure of fit.

Using the approach presented above, I can now compute the resulting quantities $E(\bar{T})_K$ and $\Phi_K^{\bar{T}}$. To this end, I define maps

$$\pi_i : \{1, 2, ..., i\} \times U \to R_{>0} : (j, \alpha) \mapsto \pi_i(j, \alpha) \tag{5.31}$$

and

$$\sigma_i : \{1, 2, ..., i\} \times U \to R_{>0} : (j, \alpha) \mapsto \sigma_i(j, \alpha) \tag{5.32}$$

for all $i = n, n-1, ..., 1$ (anti-)recursively as follows:
For $i := n$ I put $\pi_n := 1$ and $\sigma_n := 0$ and assuming that π_{i+1} and σ_{i+1} have been defined already for some $i < n$, I put

$$\pi_i(j, \alpha) := \pi_{i+1}(j, \alpha) \tag{5.33}$$

and

$$\sigma_i(j, \alpha) := \sigma_{i+1}(j, \alpha) \tag{5.34}$$

for all $j \in 1, 2, ..., i - j(i)$ while, for $j := j(i)$, I first define maps π_i^* and σ_i^* from $U \times U$ into $R_{>0}$ and R, respectively by

$$\pi_i^*(\alpha, \beta) := \pi_{i+1}(j, \alpha) \times \pi_{i+1}(j, \beta) \times exp(-D_{j,i+1}(\alpha, \beta)/T) \tag{5.35}$$

and

$$\sigma_i^*(\alpha, \beta) := \sigma_{i+1}(j, \alpha) + \sigma_{i+1}(i + 1, \beta) + D_{j,i+1}(\alpha, \beta) \tag{5.36}$$

that I used to define the required quantities $\pi_i(j, \alpha)$ and $\sigma_i(j, \alpha)$ distinguishing the cases $i \geq k$ (i.e. $i + 1 \in V_i$) and $i < k$. In case $i \geq k$ I put

$$\pi_i(j, \alpha) := \pi_i^*(\alpha, \psi(i + 1)) \tag{5.37}$$

and

$$\sigma_i(j, \alpha) := \sigma_i^*(\alpha, \sigma(i + 1)), \tag{5.38}$$

in case $i < k$, I put

$$\pi_i(j, \alpha) := \sum_\beta \pi_i^*(\alpha, \beta) \tag{5.39}$$

and

$$\sigma_i(j, \alpha) := \sum_\beta \sigma_i^*(\alpha, \beta)\pi_i^*(\alpha, \beta)/\pi_i(j, \alpha) \tag{5.40}$$

The formulae presented previously allow us now to conclude that

$$\Phi_T^{\bar{T}}(\psi) = \sum_\alpha \pi_1(1, \alpha) \tag{5.41}$$

and

$$E(\psi, \bar{T})_T = \sum_{\alpha} \sigma_1(1,\alpha)\pi_1(1,\alpha)/\Phi_T^{\bar{T}}(\psi) \tag{5.42}$$

must hold.
In other words, I can compute these two quantities by a sequence of altogether

$$2a(n-k-1)+(k-1)(2a^2+a-1)+(k-1)(2a^2+2a)+3a-1, a = \#U \tag{5.43}$$

summations and products over functions defined in U. Now I would like to sketch some implementation details of the algorithm. Assume that the following numbering of the tree nodes holds $V = 1, 2, ..., n$ and $V_1 = k+1, k+2, ..., n$ and for $\vee i \in \{1, 2, ..., n-1\}$, $\exists$ exactly one vertex $j = j(i), j \leq i$, with $\{j, i+1\} \in E$. The example tree in Figure 5.15 illustrates this numbering convention.

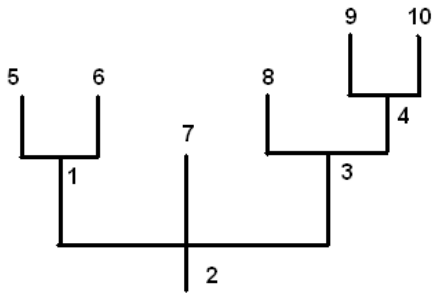


Figure 5.15: Tree encoding sample.

I associate with every node i, a corresponding row in arrays
$\pi : 1, 2, ..., n \times a$ and $\sigma : 1, 2, ..., n \times a$
At the initial step, π is filled with 1-s and σ with 0-s. Arrays π serve for "accumulation" of $exp(-KE)$ and σ for "accumulation" of E.
I have to compute

$$\Phi(\chi \mid T, K) = \sum_{\hat{\chi}} exp(-KE(\hat{\chi})) \tag{5.44}$$

and

$$\sum_{\hat{\chi}} E(\hat{\chi}) exp(-KE(\hat{\chi})) \tag{5.45}$$

To this end, I "remove" nodes from the tree (and corresponding rows from arrays π and σ) sequentially, beginning with node n. By removing of the node $i+1$, I make the following update of the array (for instance, π):

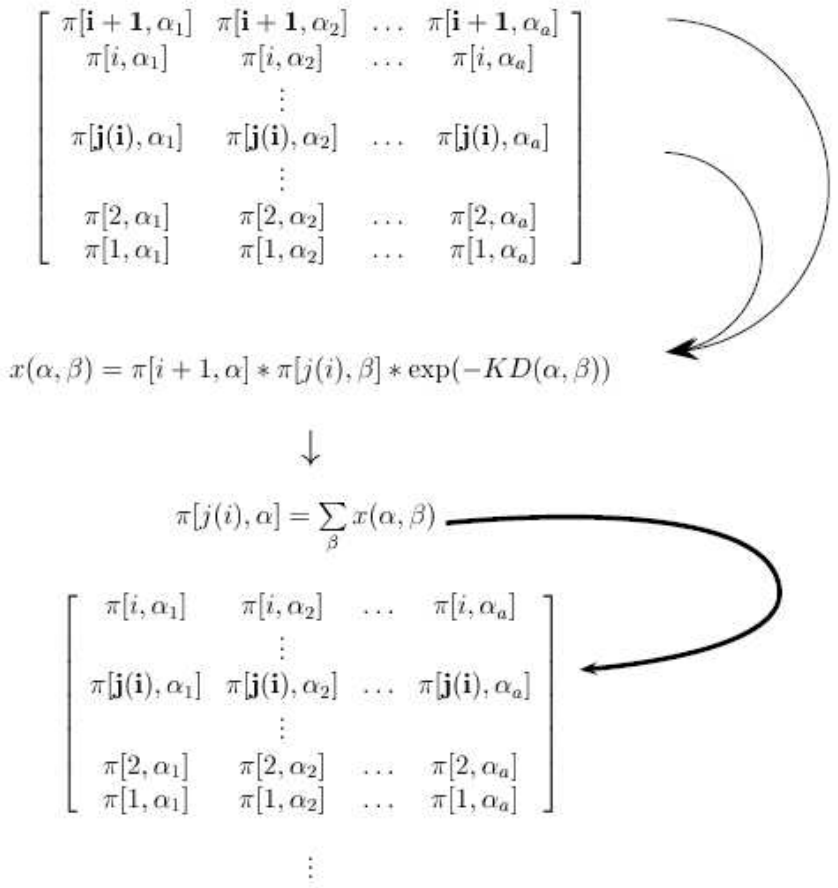


Figure 5.16: Update process overview. Each internal node/leaf is used to update its parent. Same procedure is applied to the sigma matrix.

After each step one goes one row "deeper" in the above matrix π. Analogues manipulations have to be done with the array σ too. It follows from the above considerations that at the end of the whole procedure one is at the n-th row of π and σ; these rows then are used to compute the averaged energy

$$\Phi_K^T(\psi) = \sum_\alpha \pi(1,\alpha) \tag{5.46}$$

$$E(\psi,T)_K = \sum_\alpha \sigma(1,\alpha)\pi(1,\alpha)/\Phi_K^T(\psi) \tag{5.47}$$

The algorithm is implemented in QALD software and can be easily used for any given tree and sequences. Attention must be payed on the exact correspondence between the number of leaves in the sequence file and the number of leaves in the *Newick* tree expression[3].

5.2.2 Applications

5.2.2.1 Analyzing Dunn's paper

In this Science paper [30], Dunn et al. used 120 binary grammatical features (Yes/No) in order to :

- check the phylogenetic information contained in these features for Oceanic languages
- weigh and use accordingly the most relevant features in order to detect the genealogical relationships for Papua New Guinea languages

It is somehow similar to my approach, as they first produce a maximum-likelihood analysis of the known data and the traditional tree, and then weigh the 'best' features obtained for another language family. The presented algorithm produced the following most relevant features (Table 5.2), while their approach leads to the results in Table 5.3 with the Pearson 's correlation coefficient between ranked features of $r = 0.880014$.

[3]The *Newick* tree expression is automatically converted to the internal numbering convention explained in Figure 5.15.

FeatNr	FeatDescr
97	VS Intransitive Clauses
59	S Prefix marking
61	A Prefix Marking
44	Decimal Numerals
14	Article-Noun Fix Order
66	Verb Variation ClauseType
67	Verb Variation Person
12	Definite Or Specific Articles
98	Verb Initial Transitive Clauses
52	Postpositions
85	Verb Classifiers
35	Possessive Classifiers
41	Marked Possessor
13	Indefinite Or Non-Specific Articles
29	Plural-Marked Noun

Table 5.2: My list of best features

FeatNr	FeatDescr
97	VS Intransitive Clauses
59	S Prefix Marking
61	A Prefix Marking
66	Verb Variation Clause Type
67	Verb Variation Person
98	Verb Initial Transitive Clauses
50	Oblique Case Marking
83	Reflexive Morphology
74	Recipient Object
44	Decimal Numerals
35	Possessive Classifiers
52	Postpositions
29	Plural-Marked Noun
12	Definite Or Specific Articles
14	Article-Noun Fix Order

Table 5.3: Dunn’s list of best features

5.2.2.2 Detecting the best tree for a sequence file

The algorithm can also be used on the opposite direction. Given a set of trees and one alignment, the algorithm tries to match the sequences to each tree, and therefore detects the 'best' one. An illustrative example of two different tree topologies is presented in Figure 5.17. If the alignment from Table 5.4 is analyzed together

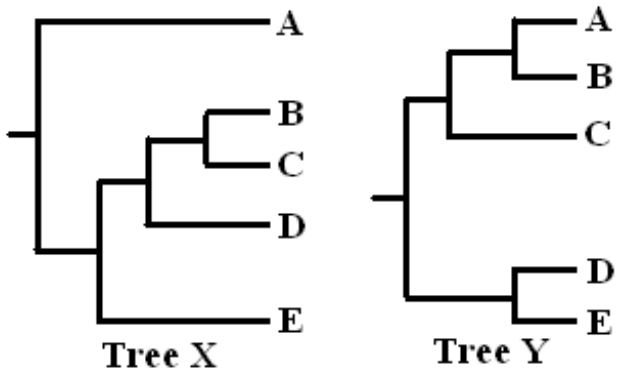


Figure 5.17: 2 different tree topologies.

with these two trees, it is clear that Tree Y is a better fitting to this data as both

A	1	2	2
B	1	2	2
C	1	1	2
D	2	3	1
E	2	3	1

Table 5.4: Sample data set.

column 1 and column 3 in the table support the split {A,B,C} vs {D,E}, while column 2 induces the split {A,B} vs {C} and {A, B, C} vs {D,E}.

5.2.2.3 Detecting the distances between trees

A new and interesting idea[4], even if not completely tested, was implemented to discover the distances between trees, meaning how different/distant are two trees , with respect with their topologies. Having two trees T1 and T2, and their data sets

[4]Thanks to Sören Wichmann for collaborating on this idea.

D1 and D2 (I assume here that the trees 'perfectly' fit to their data[5]), I propose a new distance measure between trees as shown in Equation 5.48.

$$dist(T1, T2) = Entr(T1, D2) + Entr(T2, D1) \tag{5.48}$$

Therefore, having the trees topologies in Figure 5.18 and their perfect data sets,

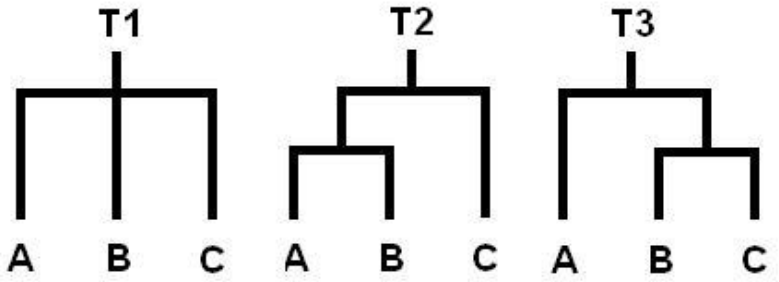


Figure 5.18: 3 different tree topologies.

D1 for T1, D2 for T2 and D3 for T3 respectively as shown in Table 5.5 I will

	D1	D2	D3
A	1	1	2
B	1	1	1
C	1	2	1

Table 5.5: Data set

obtain the following distances between trees as presented in Table 5.6.

dist(T1,T2)	1.00045
dist(T1,T3)	1.00045
dist(T2,T3)	2.00013

Table 5.6: Distances between trees

[5]I generate the 'perfect' data set by supplying the necessary columns to match each cluster in the tree topology.

5.2.2.4 Future improvements

The algorithm can be improved for future research. One drawback is the impossibility of dealing with missing data. This might be solved in a simplistic way by supplying the universal missing symbols (e.g., “?”, “-”) with all the possible values found in the corresponding column of the alignment, or more complex, by applying specific mathematical procedures [54]. Another improvement might be achieved by performing statistics on the features energies values, and/or on each feature values distribution. The distance measurements between trees might be improved if specific details about the trees and the alignment are provided. Character distance matrix can be incorporated in the algorithm, and this will provide a general valuable procedure for various datasets.

Nevertheless, I believe that this algorithm, even as simple as it is, might provide deep hidden information about a data set and the phylogenetic tree associated.

Chapter 6

QALD software

Most of the analyses and methods described in this thesis were implemented in a software package called QALD (Quantitative Analyses of Linguistic Data) that interacts with data from both the original WALS and the recoded WALSX. Using the MySql engine (Appendix 8.1) [1], appropriate conversions of the data were made for an easier interaction between the Visual Studio environment[1] and the databases. The aim of the QALD package is to be a tool for both database interrogation and phylogenetic analyses of the data contained in the WALS database. Moreover, QALD offers various algorithms for data analysis, classification of languages by family or geographical location, and methods for selecting the most informative features. The QALD software is available as a stand-alone version, and can be downloaded from *http://lingweb.eva.mpg.de/phylogenetictools/*.

It was implemented in Visual C 6.0 and it uses OpenGL functions for graphical display. It is a Single Document Interface (SDI) project type and classes were created to deal with various issues:

- entropy algorithm (CEntropy),
- WALS analyses (CWalsDB),
- MySql interaction (CMySqlMap),
- graphical manipulation (CBody3D, CVertex3D).

[1]Visual Studio 6.0 offers a friendly user interface to programming in VisualC, Visual Basic, Visual Java, as well as in-line help for predefined classes and various projects types.

In this way, the source code became very well structured and easy to be integrated if required, in other software implementations. Graphical manipulations are a new set of features to modify a tree by using the mouse, while the entropy algorithm (see Section 5.2) can be applied to the trees. Because it is using a Document-View architecture, all the graphical elements are saved in the *CDocument* class, while the actual drawing operation is implemented in the *OnDraw* function from *CView* class, by performing the necessary checking on the document elements, thus allowing real-time updating of the graphical elements. This approach permits easier implementation of functions like *Undo*, *Save*, *Print* or *Modify Tree*. Many of the graphical ideas implemented here were actually part of the author's university thesis [2].

6.1 Loading, displaying and modifying tree structures

As shown in Figure 6.1, the graphical user interface (GUI) of the software consists in a display/modify window, a set of utility buttons displays in a left toolbar, menus, as well as dialog boxes to set parameters, run the methods and display the results. All the buttons have correspondences in menus and shortcuts are also provided. The drawing area is used to display the tree (loaded from file or obtained from phylogenetic algorithms), as well as modifying them graphically (by mouse action). Using then the *Save tree* button, one can use the modified tree for further investigation.

The graphical drawing is not a very esthetically one, but it is one of the few implementations to display, modify (by mouse action) and save the trees using the 'Newick' tree expression described in
http://evolution.genetics.washington.edu/phylip/newick_doc.html, or as exemplified in 6.1.

$$((Finnish, Hungarian), (Romanian, (Italian, (Spanish, Portuguese))));\tag{6.1}$$

Nevertheless, converting such a simple parenthesis format to a graphical representation of the tree implies some preprocessing steps (Algorithm 3).

This implementation deals with two major problems:

- construct the tree structure based on the tree expression (*ReadTreeFile, AnalyzeTreeExpression*)
- assign the correct coordinates of the graphical elements, so it can be displayed and allow mouse input modifications (*CalculateTreePosition*). Many tree representations will not fit in a user screen (as the size has to be large

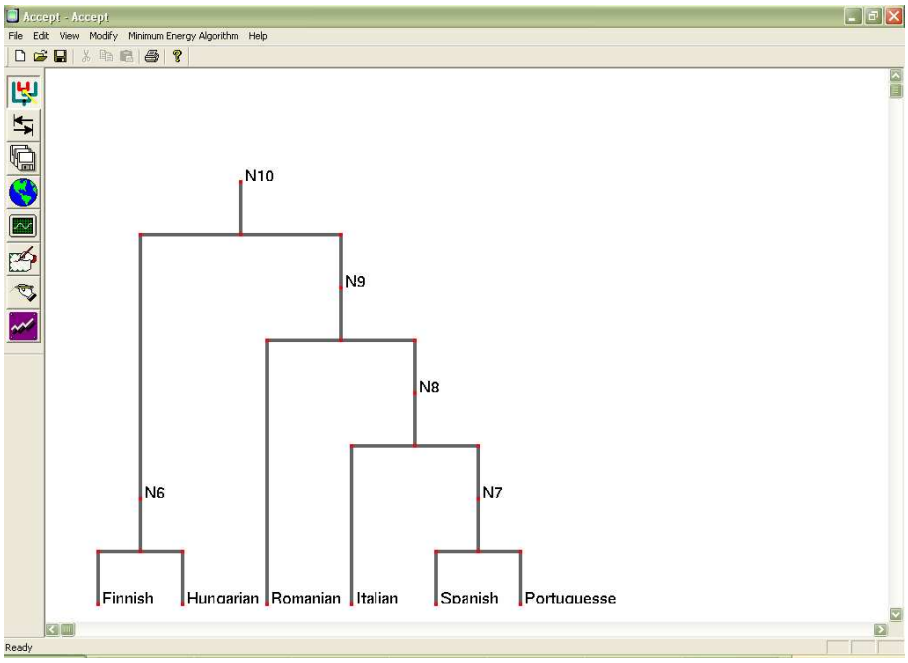


Figure 6.1: Main user interface of the QALD software.

input : The file name that contains the tree expression *CString* **strFileName**
output: Tree structure *CTreeNode* **gTreeRoot**

```
1 strTreeExpression = ReadTreeFile(strFileName)
2 gTreeRoot =AnalyzeTreeExpression(strTreeExpression, 0)
3 gTreeRoot = CalculateTreePosition()
```

Algorithm 3: Conversion from tree expression to graphical elements.

enough in order to be able to distinguish the structure and to modify it), so scroll bars (both horizontal and vertical) were necessarily implemented. This implies more calculations of coordinates (especially on the *select* operation) but unfortunately complete reliability tests were not performed. [2]

While in the function *ReadTreeFile* only a string analysis is performed (based on identifying the node separators, e.g. comma sign, left and right signs) to detect the leaves and assign them unique ID-s. In contrast, the *AnalyzeTreeExpression* routine is a more complex one. This is a recursive function to detect the actual internal structure of the tree, e.g. the parent nodes and its children (an illustrative example is presented in Figure 6.2). Before calling this function for the first

input : The tree expression *CString* **strTreeExpression**, the level in the tree *CInt* **nLevel**
output: Tree structure *CTreeNode* **gTreeRoot**

```
1 structureListPair = FindParanthesisPair(strTreeExpression)
2 strUpdatedTreeExpresion ← strTreeExpression
3 forall pairs ∈ structureListPair do
4 |   strUpdatedTreeExpresion = ReplaceInString
  |   (strUpdatedTreeExpresion, pairs, nextInternalNode)
5 |   gTreeRoot = InsertParentNode(pairs, nextInternalNode)
6 |   nextInternalNode ← nextInternalNode + 1
7 end
8 nLevel ← nLevel + 1
9 AnalyzeTreeExpression(strUpdatedTreeExpression, nLevel)
```

Algorithm 4: Building the tree structure.

time, the structure *gTreeRoot* is initialized as a node of the type *root* with all the leaves as its children (cf. *Level 0* in Figure 6.2)[3]. In the first step, cf Algorithm 4, line 1, a list of current replaceable pairs are detected (in my sample, **F,H** and **S,P** are found). These pairs are replaced in *Level 1* with the internal nodes **N6** and **N7** respectively (the next available numbering after including the leaves). By replacing I mean the search in the tree structure to find both elements to be replaced, find their common parent (on this level, their parent will be the *root*), and replace the *root* pointers to these elements to a pointer to the new element to be inserted, while the new element's pointers to the children would be inherited from

[2] If scrollbars are used, the drawing must be updated with the correct coordinates in order to be able to recognize the correct selection point (the point where the left mouse click was performed).

[3] The *nLevel* variable is saved in the *tree structure* and it will be used in the assigning of graphical coordinates to the nodes.

the above parent, the *root* in this case). This is done until all the pairs on the current level are replaced, then the *nLevel* counter is increased (Algorithm 4, line 8), and a new call to the *AnalyzeTreeExpression* function is made (Line 9) with the updated tree expression. This procedure continue until there are no more pairs found in the continuously updated tree expression, with the remark that before the end, an unique ID is also assigned to the *root*, as in Figure 6.2 the **N10** ID.

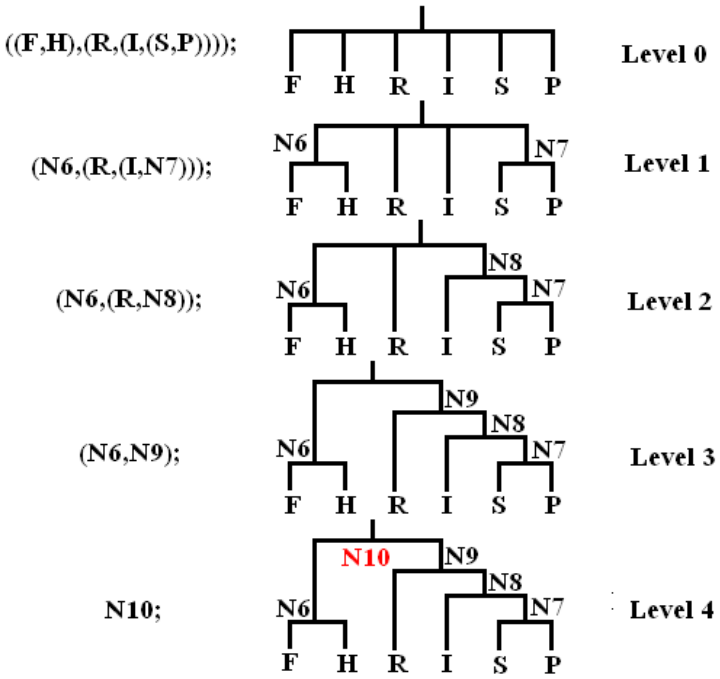


Figure 6.2: Building trees from string expression.

The definition of the *CTreeNode* class is described in Appendix 8.3. It is derived from class *CObject* for better serialization (save/load) operations. In the drawing, red squares indicate the modifiable objects, e.g. the internal nodes that one can select, move or delete in order to obtain the desired results. These actions can be performed by selecting the appropriate buttons on the left panel (*Select and Move* and *Select and Delete*) and by choosing the internal nodes that must support the operation. Please note that the usual behavior is top-down. A selection of an internal node together with a specific operation will imply that the same operation will be applied to the entire subtree that is dependent on the chosen node, e.g. to all its children.

Now, the calculus of the graphical coordinates of the nodes is an in-hand procedure, the only problem being to deal with scroll bar action if the entire tree does not have enough space to be display in one screen, or only part of it is to be visu-

alized.

Because there is no need to worry about where exactly to put the the leaves, the algorithm is free to decide on the size of the picture [4]. Analyzing each node (starting with the leaves and continuing with the internal nodes until the algorithm arrives at the root), it is possible to determine, if the size of the *draw area* is known, what coordinates each of the nodes should have. Then, following the relationships between nodes (parent-child), it is trivial to detect how to draw the lines between them. In this case, the choice was similar to cladogram (straight lines, not obliques), but this can be easily changed into more fancy variants.

Nevertheless, it is possible to save the picture back to the 'Newick' tree expression format (useful if a tree was changed) by pressing the *Save* button or by choosing the *Save Tree* from the *Edit* menu. The procedure is based on a tree parsing methodology, which starts with the root node, and then replaces its ID with the ID's of the children in a parentheses format, i.e. surrounded by parentheses and separated by commas. Then, recursively, the same method is applied to all its children until the leaf nodes are reached.

6.2 Algorithms

6.2.1 Entropy algorithm

The entropy algorithm (described in Section 5.2.1) can be accessed from the menu *Minimum Energy Algorithm* or by pressing directly *ALT+E*. This will result in displaying the parameter settings dialog box, where the user must specify the name and the location of the tree file (by pressing the *Please upload the graph file* button), the sequence file (by pressing the *Please upload the sequence file* button), the value of the temperature constant and the name of the output file. By pressing the *Perform* button, the algorithm is applied to the specified files. As soon as the operation is finished, a small dialog box containing the main results is displayed. As depicted in Figure 6.4, the summary shows the fitting values for each character (i.e. each column in the alignment file) to the specified tree. These information are also saved in a text format, with tab-separated fields, in the file specified as the output.

[4] I chose $r.bottom = gnNrLevels * 155$ and $r.right = gnNrLeaves * 100$ where *r* is a rectangle structure (with $r.top = 0$ and $r.left = 0$) that represent the drawing dimension. The number of leaves (*gnNrLeaves*) and the number of levels (*gnNrLevels*) are global variables determined apriori

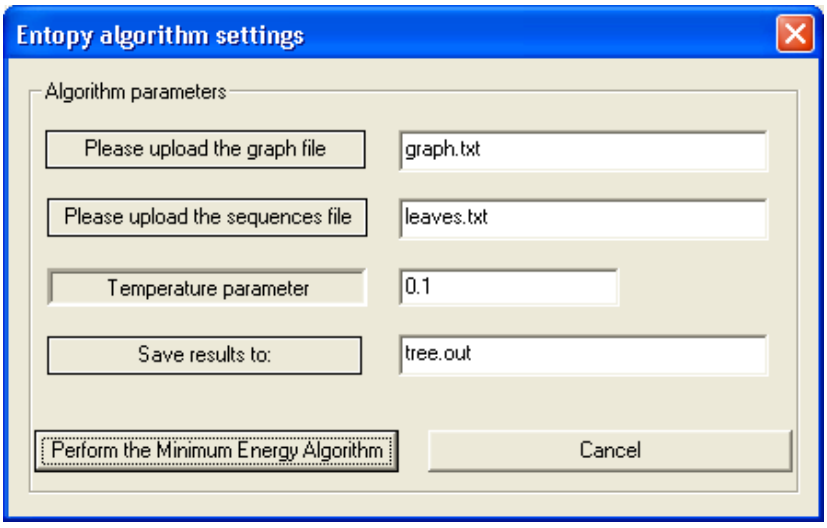


Figure 6.3: The dialog box for the entropy algorithm.

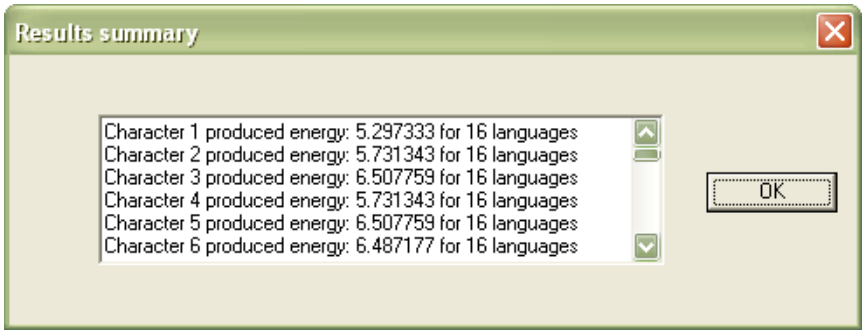


Figure 6.4: Results summary for the entropy algorithm.

6.2.2 Phylogenetic algorithms

The phylogenetic reconstruction algorithms implemented can be accessed from the menu *Phylogenetic Algorithms*. The dialog box as shown in Figure 6.5 will then be displayed. The user must choose the distance matrix file (in standard for-

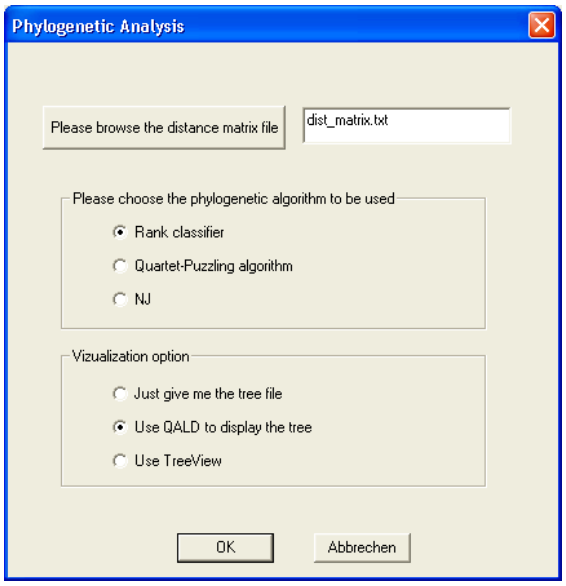


Figure 6.5: Phylogenetic algorithms dialog box.

mat [32]) by opening a browse file dialog box from the *Browse* button, the phylogenetic algorithm to be performed on the distance matrix (rank classifier, distance quartet puzzling or the standard neighbour-joining) and the method for the visualization of the tree produced by the algorithm selected. By pressing the *Ok* button, the appropriate algorithm is performed (with possible intermediate steps for setting the necessary parameters, e.g. the rank classifier method, the distance quartet-algorithm).

6.2.2.1 Rank classifier algorithm

The rank classifier algorithm needs a series of parameters that can be set in the dialog box presented in Figure 6.6 (it can be accessed after the user selects this algorithm from the options in Figure 6.5). A short description of the parameters (as they are saved in the parameter file) is presented below:

- $file\_format$ = the format of the input matrix. The matrices can be similarity matrices or dissimilarity matrices, and can include, or not, information about the number and the names of the objects. When no names are specified, numbers are assigned to identify the objects in the output tree.

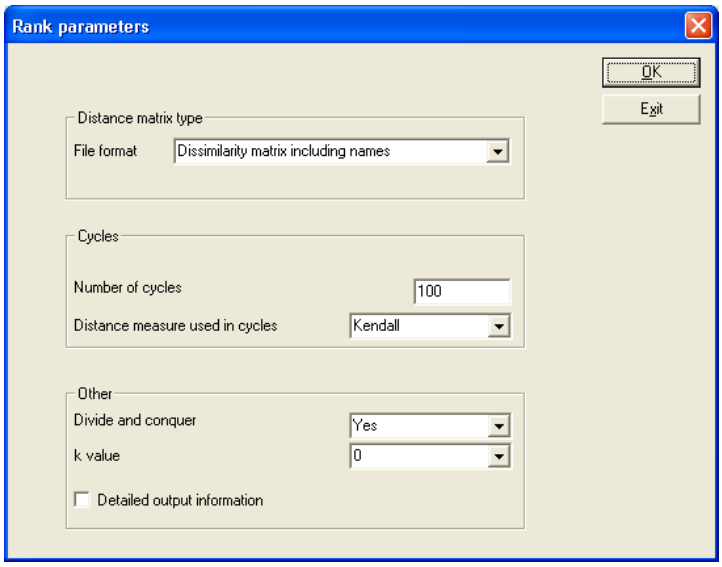


Figure 6.6: Rank classifier algorithm dialog box.

3			
Human	0	1	2
Gorilla	1	0	2
Whale	2	2	0

Table 6.1: Phylip matrix format including the number and the names of the species

- nr_cycles = the maximum number of cycles to be performed if the convergence it is not achieved initially.
- $distance$ = the formula used to build a new matrix based on a previous distance (rank) matrix, for every cycle. Two methods are implemented: using the Spearman distance formula or using the Kendall distance formula respectively.
- $divide$ = the divide and conquer parameter to specify whether this should be performed or not. A Yes/No option is provided.
- $output\_type$ = the type of output that one might need. It can be either of the variants: (i) only a Newick tree expression that can be used in tree-view programs or, (ii) a detailed (if the check box is selected) cycle-by-cycle output, if one wants to investigate the various rank classifications of the objects.

6.2.2.2 Distance quartet-puzzling algorithm

The parameters for the distance quartet-puzzling algorithm (DistQ) are fewer, and can be set in the dialog box displayed in Figure 6.7. As the distance matrix file was already specified in a previous step, this dialog box is only used to specify which variant of the algorithm should be performed: the minimum, the maximum, the average or using the sum, as described in 5.1.2.

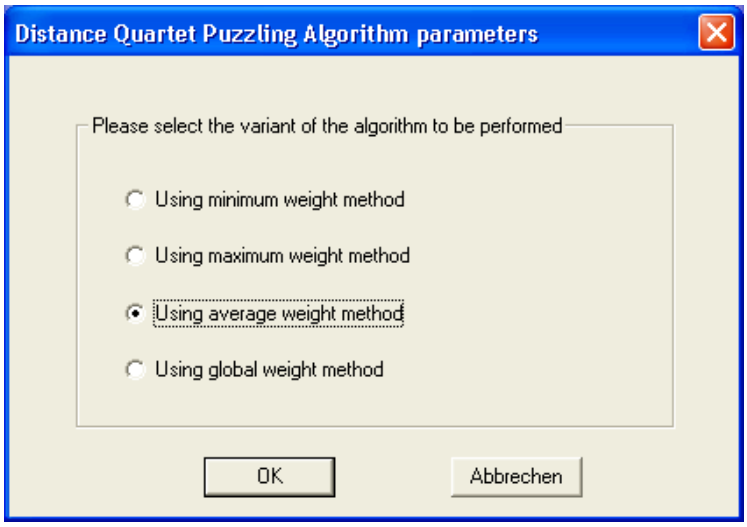


Figure 6.7: Distance-quartet algorithm dialog box.

6.2.3 Results

The results are usually files that contain the tree expression format of the inputed data. Their name, if no other decisions are made, is composed from the initial distance matrix file name, to which the extension *.out* is added. To visualize these expressions, the user might choose between the QALD software and the TreeView program from phylip package [32].

Of course, one can use the tree expression from the file as an input data to any phylogenetic drawing program, as the parenthesis format contained in this file is widely accepted by such programs. Unfortunately, the saving and printing functions of the QALD software are not implemented because they required too much effort, and I have concentrated on the algorithmic part.

6.3 Dealing with missing data

As explained in Section 3.1, there are many ways to deal with missing data.

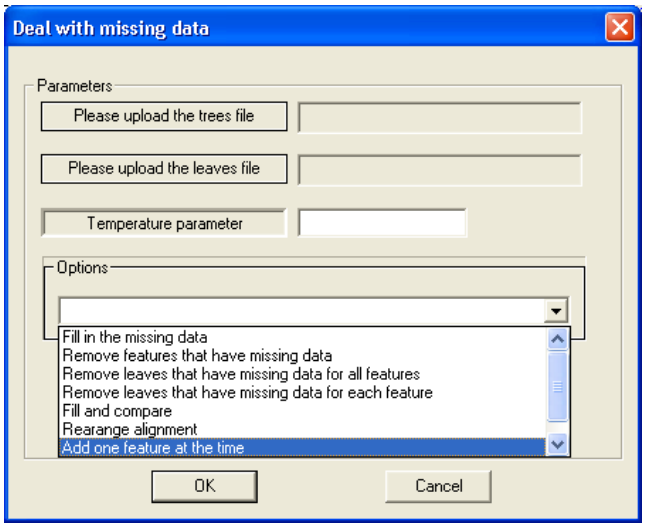


Figure 6.8: Dealing with missing data.

The entropy algorithm as implemented in the QALD package offers the possibility of various ways to deal with missing data (cf. Figure 6.8):

- Fill the missing data. It is worth to use this is the range of feature values is as small as possible (preferably 0/1).
- Remove features that have missing data. Features that contain missing data are removed from the analyses.
- Remove leaves that have missing data for all features. The sequences are removed whenever they contain missing data in all positions.
- Remove leaves that have missing data for each feature. The sequences are removed whenever they contain missing data for a feature.
- Add one feature at the time. Perform the entropy algorithm by analyzing each feature at the time. This approach is useful when the features are rank ordered by their known fitting to the data [5].

6.4 Languages analyses

This part of the QALD package offers a quick and informative view of the data in WALS. The user might select a group of languages, genera or families to be

[5] If one has a list of best ranked features, by applying the algorithm and observing the resolution of the results, one can detect a level of confidence above which the features added are *damaging* the result.

visualized, and the displayed pictures are obtained from the ethnologue file [35] as shown in Figure 6.9.

Figure 6.9: Sample information from *ethnologue*.

There is also the possibility of performing the algorithms and distance methods presented in this thesis (as depicted in Figure 6.10), and the results can be easily compared with the NJ or ethnologue ones. A choice of distance matrix building procedure is offered (as explained in Section 3.2) by allowing the *Hamming relative*, *NormD*, *NormS* and *NormS+NormD* variants. Also, the user may choose which features of WALS to be used: the 'best' phylogenetic ones, all, a predefined group as found in WALS or any set of individually selected features. The visualization of the results are the same as in Section 6.2.3.

Please note that the selections of languages and features are cumulative, meaning that by selecting the Turkic language and the Indo-European family, the methodologies will analyze all the languages grouped in the Indo-European family plus the Turkic language. Please also note that by selecting a family or a genus, the software is evaluating all the languages in that family or genus respectively, and in many cases each of these groupings contain poor coded languages. By checking the *Use Wals Extended* check box, the feature list is updated with the 266 extended features obtained as explained in Chapter 2, while un-checking it will

use the original 141 features for further analysis.

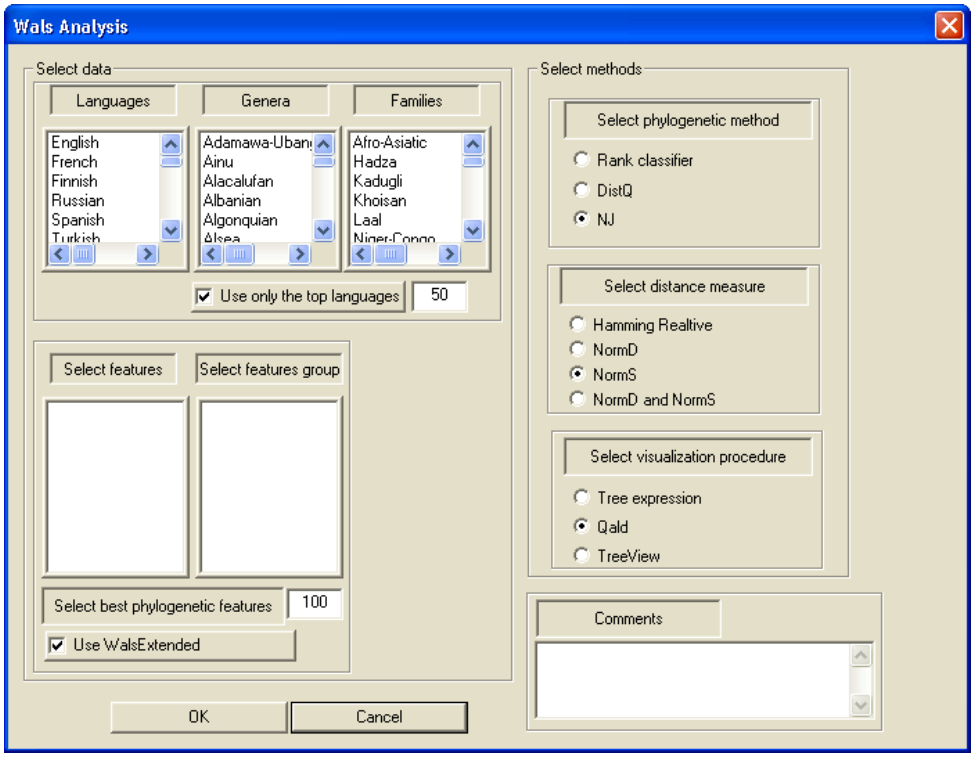


Figure 6.10: The phylogenetic analyses of WALS data.

Chapter 7

Conclusions

The extensive use of computational tools for analyzing various data sets in linguistic community indicates recognition of their potential in this field. Well-designed experiments with a clearly defined objective and with data stored in sufficient detail in different formats should provide a valuable resource for discovering many different aspects of language characteristics, universality and history. In this research I showed that by combining these data sets with various mathematical models and computational algorithms, one can obtain improved results that could not be reached by traditional research methods. These improvements can be further developed and applied/tested to other data sets, in order to better understand the methodologies and to be able to better interpret the results.

Meanwhile, the results of the present study imply that more research must be pursued in this area. This thesis has offered the results of explorations of a typological data set using computationally-driven phylogenetic software developed largely for the use of biologists.

As mentioned, the utility of typological databases for historical linguistic research cannot be fully assessed until more extensive databases have been constructed. Nor can we hope to bring our results to bear on actual empirical problems before relevant databases have been enlarged. The WALS database provides a good beginning, as long as the problems of overlap and wastebasket categories are taken into account. Simply filling out the missing data in the WALS matrix for the set of languages that one would like to compare would already constitute a useful step forward.

The typological (dis-)similarities in WALS provide useful information in historical analysis of languages. For example, selecting a set of 39 languages from

the Indo-European family and using the NormS variant for building the distance matrix produced the network depicted in Figure 7.1. The result is showing a clear split between the Indic languages and the European languages, as well as a correct grouping of most of the genera.

I showed various methodologies for analyzing typological data. First, I have presented a method for quantifying the influence of different typological features for establishing linguistic genealogies. I showed that by performing a careful analysis and by recoding the initial data, important discoveries of various problems/advantages can be detected. I presented methodologies of improving distance measurements, and the results obtained by using these variants showed a good improvement. Each of the approaches improved the basic results, therefore it must be further considered in order to completely understand the results obtained, and maybe to be able to build a new, complete and appropriate distance measurement method. The entropy algorithm (Section 5.2) proved to be an useful tool when dealing with various data sets and phylogenetic trees. Its applications are multiple, and therefore further investigations/improvements must be considered. The QALD software implemented in this research will help scientists to have a clearer picture on both the universal accepted phylogenetic relationships between languages, and on various methodologies and algorithms that one can perform for a specific data set.

A final phylogenetic analysis was performed by trying to integrate the distances between feature values for the recoded WALSX. Consequently, I chose a data set of 13 genera, with two best coded languages selected for each genera in order to have a good variability and world-wide distribution. A comparison of the results can be consulted in Figure 7.2 (using original data and relative Hamming distance) and Figure 7.3 (using WALSX and the character distance measures, cf. Section 2.2.1.4).

Further research should be focused on the degree of confidence in order that one can accept hypotheses concerning genealogical relationships generated by a given algorithm on the basis of typological data. A good deal of theoretical work will need to go into exploring adequate ways of comparing trees in order to correctly assessed the validity of a phylogenetic tree obtained using typological data.

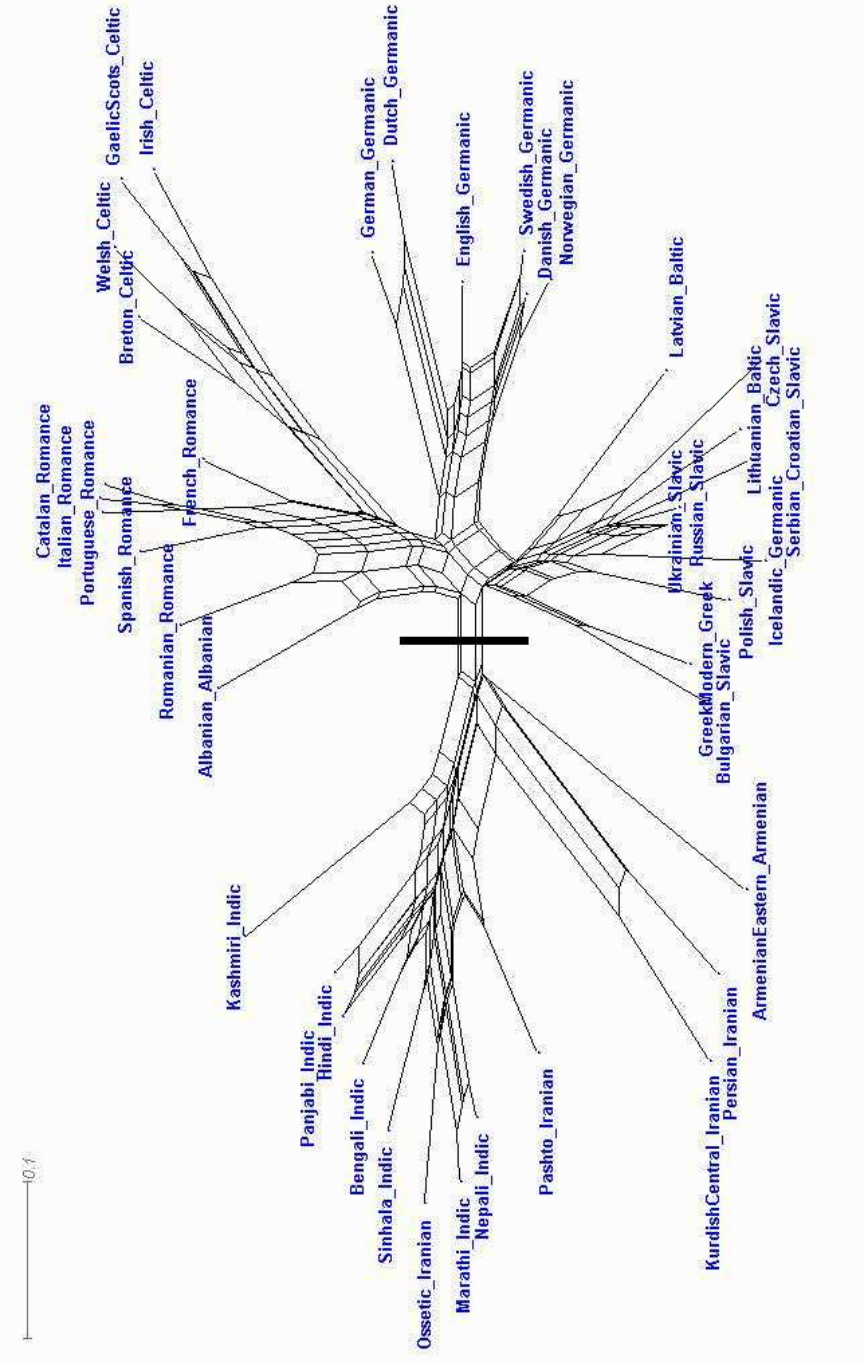


Figure 7.1: NNet using the Indo-European data set.

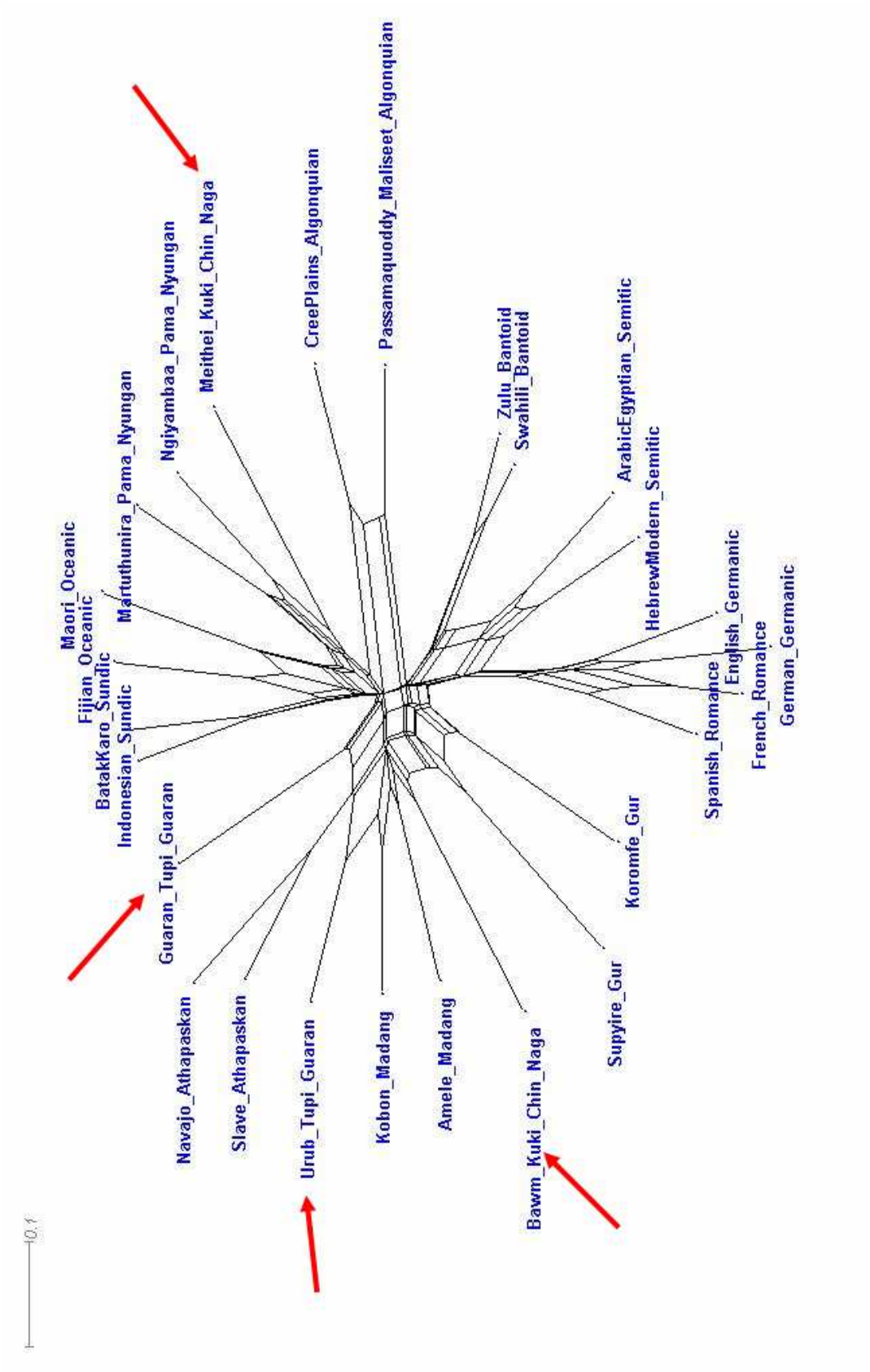


Figure 7.2: NNet using the original WALS data.

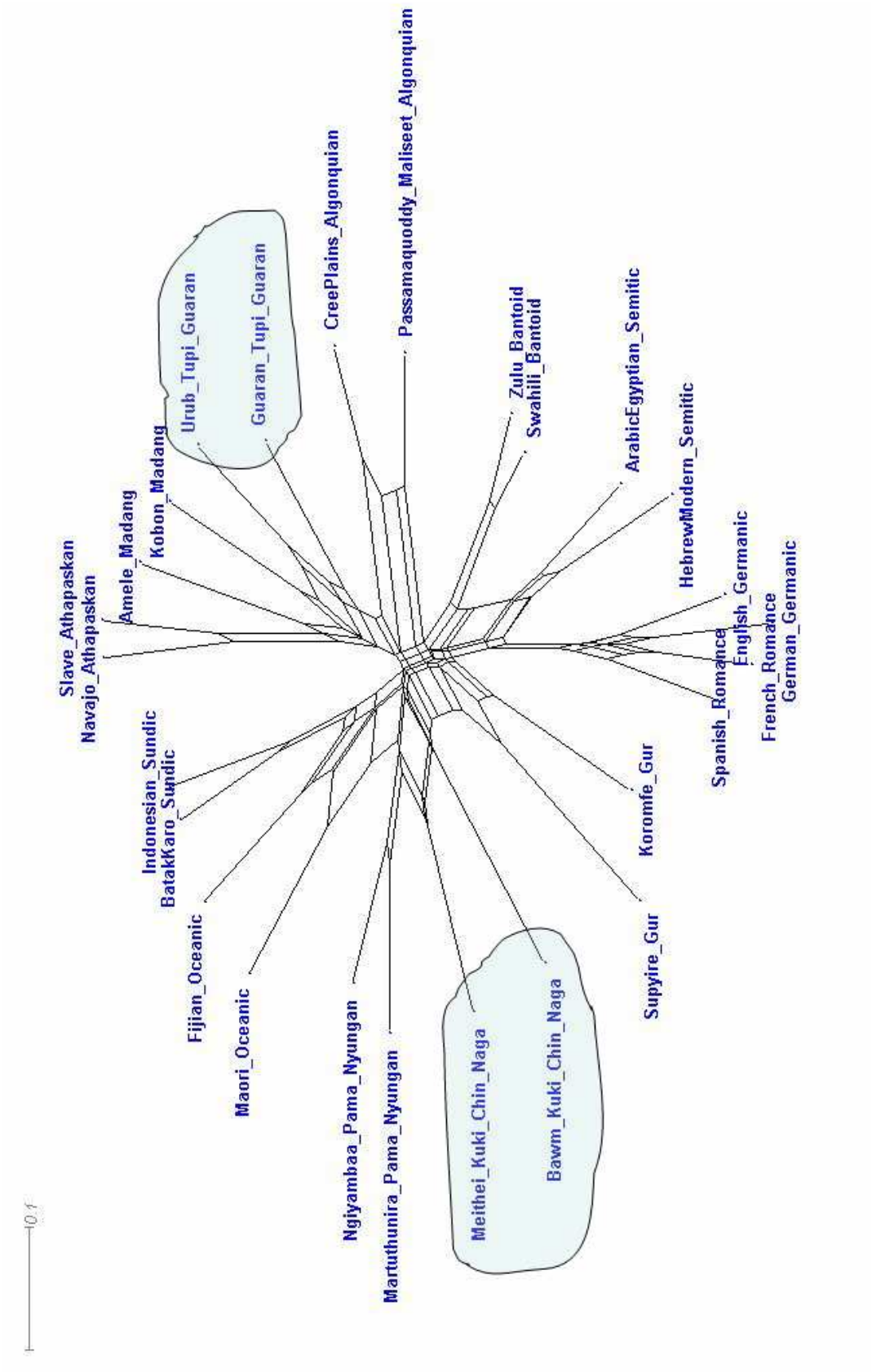


Figure 7.3: NNet using the recoded WALSX data.

Chapter 8

Appendix

8.1 MySql script for building WALS

Building the database
Created by...: Mihai Albu $albu@eva.mpg.de$
Description..: Create WalsX database together with all the tables
Orig Date....: 27/01/05
Modified Date: 27/06/06
delete database if exists
DROP DATABASE IF EXISTS WalsTemp;
create a new database
CREATE DATABASE WalsTemp;
specify that we are using it
USE WalsX;
create tables
CREATE TABLE language (language_ ID INT UNIQUE, standard_ name CHAR(100), wals_ code char(100), longitude_ text char(100), latitude_ text char(100), longitude_ num INT, latitude_ num INT, genus_ ID INT, macro_ area_ ID INT, location CHAR(100), amount_ of_ refs INT, comment CHAR(255), amount_ of_ datapoints INT, all_ countries CHAR(255), all_ othernames CHAR(255), all_ routledgenames CHAR(255), all_ ruhlennames CHAR(255), all_ regions CHAR(255), othernames_ label CHAR(255));
Copying the data All the data were initially saved in *TAB* files, as exported from the original database. Then the next script helped copying the data into the new

MySql database.
Created by...: Mihai Albu albu@eva.mpg.de
Description..: insert values in tables from tab delimited files
Orig Date....: 27/01/05
Modified Date: 27/06/06
USE WalsTemp;
LOAD DATA INFILE ' $Tabs\ add\_ref.tab' INTO TABLE add\_ref$;
SHOW WARNINGS;

The above two last lines were repeated for all the tables in the database.

8.2 Features from WALSX that allow *character distance* implementation

Table 8.1: Features from WALSX that allow *character distance* implementation

Feature	FeatValue	Description
1. Consonant inventories	1	Small consonant inventories
	2	Moderately small consonant inventories
	3	Average consonant inventories
	4	Moderately large consonant inventories
	5	Large consonant inventories
2. Vowel quality inventories	1	Small vowel inventories
	2	Average vowel inventories
	3	Large vowel inventories
17. Type of velar nasal	1	No velar nasal
	2	Velar nasal initially
	3	Velar nasal, but no initially
21. Syllable structure	1	Simple syllable structure
	2	Moderately complex syllable structure
	3	Complex syllable structure
23. Types of tone	1	No tone system
	2	Simple tone system

Continued on next page

Table 8.1 – continued from previous page

Feature	FeatValue	Description
	3	Complex tone system
48. Inflectional synthesis of the verb	1	0-1 category per word
	2	2-3 categories per word
	3	4-5 categories per word
	4	6-7 categories per word
	5	8-9 categories per word
	6	10-11 categories per word
	7	12-13 categories per word
56. Prefixing and suffixing in inflectional morphology	1	Strongly suffixing
	2	Weakly suffixing
	3	Intermediate
	4	Weakly prefixing
	5	Strongly prefixing
58. Reduplication	1	No productive reduplication
	2	Full reduplication only
	3	Productive full and partial reduplication
64. Number of genders	1	None
	2	Two
	3	Three
	4	Four
	5	Five or more
71. Plural types in human nouns	1	No plural
	2	Plural optional
	3	Plural obligatory
73. Plural types in inanimates nouns	1	No plural
	2	Plural optional
	3	Plural obligatory
90. Distance contrast in demonstratives	1	No distance constrast
	2	Two way contrast
	3	Three way contrast
	4	Four way contrast

Continued on next page

Table 8.1 – continued from previous page

Feature	FeatValue	Description
	5	Five or more way contrast
110. Number of cases	1	No or borderline case
	2	2 case categories
	3	3 case categories
	4	4 case categories
	5	5 case categories
	6	6-7 case categories
	7	8-9 case categories
	8	10 or more case categories
117. Comitatives and intrumentals	1	Identity
	2	Mixed
	3	Differentiation
125. Presence of numeral classifier	1	Absent
	2	Optional
	3	Obligatory
132. Type of possesive classification	1	No possessive classification
	2	Two classes
	3	Three to five classes
	4	More than five classes
133. Genitives, Adjectives and Relatives Clauses	1	Weakly differentiated
	2	Moderately differentiated
	3	Higly differentiated
169. Order of subject and verb	1	Subject precedes verb
	2	Both orders with either dominant
	3	Subject follows verb
170. Order of object and verb	1	Object precedes verb
	2	Both orders with either dominant
	3	Object follows verb
174. Order of genitive and noun	1	Genitive–Noun
	2	Both orders with either dominant
	3	Noun–Genitive

Continued on next page

Table 8.1 – continued from previous page

Feature	FeatValue	Description
176. Order of demonstratives and noun	1	Demonstratives before noun
	2	No dominant order
	3	Demonstratives after noun
181. Order of degree word and adjective	1	Degree word precedes adjective
	2	No dominant order
	3	Degree word follows adjective
184. Position of interrogative phrases in content question	1	Interrogative phrases obligatory initial
	2	Mixed
	3	Interrogative phrases not obligatory initial
202. Verbal marking	1	None
	2	One argument
	3	Two arguments
226. Predicative adjectives description	1	Predicative adjectives have verbal encoding
	2	Predicative adjectives have mixed encoding
	3	Predicative adjectives have non verbal encoding
237. Purpose clause	1	Balanced purpose clause
	2	Balanced/deranked purpose clause
	3	Deranked purpose clause
238. When clause	1	Balanced when clause
	2	Balanced/deranked when clause
	3	Deranked when clause
239. Reason clause	1	Balanced reason clause
	2	Balanced/deranked reason clause
	3	Deranked reason clause
240. Utterance complement clause	1	Balanced utterance complement clause
	2	Balanced/deranked utterance complement clause

Continued on next page

Table 8.1 – continued from previous page

Feature	FeatValue	Description
	3	Deranked utterance complement clause
244. Number of non-derived basic colour categories	1	3 categories
	2	Between 3 and 4 categories
	3	4 categories
	4	Between 4 and 5 categories
	5	5 categories
	6	Between 5 and 6 categories
	7	6 categories
245. Number of basic colour categories	1	3 , between 3 and 4, or 4 categories
	2	Between 4 and 5, 5, or between 5 and 6 categories
	3	6 or between 6 and 7 categories
	4	7 or between 7 and 8 categories
	5	8 or between 8 and 9 categories
	6	9 or between 9 and 10 categories
	7	More than 10 categories

8.3 Definition of class *CTreeNode*

```
class CTreeNode :   public CObject
{
public:
  CTreeNode(); \* the class constructor *\
  virtual ~CTreeNode();\* the class destructor *\
  CString strParentID;\* the unique stringID of the parent *\
  int nNrChildren;\* the number of children *\
  CString strLabel;\* the actual node name *\
  CString strType;\* the type of the structure=node, leaf, root *\
  int *strArrayChildren;\* array of pointers to
                the children nodes*\
  CString strParent;\* the name of the parent node *\
  int nLevel;\* the level in the hierarchy tree *\
  int nID;\* unique ID number *\
  CString strStatus;\* selected, unselected, deleted *\
```

```
  int xPos;\* X coordinate of the graphical display *\
  int yPos;\* Y coordinate of the graphical display *\
}
```

8.4 Implemetation of class *CTreeEnergy*

```
CTreeEnergy::CTreeEnergy()
{
gnSequenceSize = 0;
gnAlphSize = 0;
gnGraphSize = 0;
gnNrLines = 0;
nSequenceSize = 0;
nAlphabetSize = 0;
dTemperature = 0.0;
}
int CTreeEnergy::LoadGraph(char *strGraphFileName)
{
char buffer[1024] ={0};
int  nNodeValue =0;
int  nCursor =0;
FILE * hndFile = NULL;
hndFile = fopen(strGraphFileName, "r");
if (hndFile != NULL)
{
 while(fgets(buffer, sizeof(buffer), hndFile)!=NULL)
  nCursor ++;
 fclose(hndFile);
}
nCursor ++;
pGraph = (int*) calloc(nCursor, sizeof(int));
nCursor = 0;
hndFile = fopen(strGraphFileName, "r");
if (hndFile != NULL)
{
 while(fgets(buffer, sizeof(buffer), hndFile)!=NULL)
 {
  nNodeValue = atoi(buffer);
  pGraph[nCursor] = nNodeValue;
  nCursor ++;
```

```
 }
 fclose(hndFile);
}
nCursor ++;
gnGraphSize = nCursor;
return 1;
}
int CTreeEnergy::LoadLeaves(char *strLeavesFileName)
{
FILE * hndFile = NULL;
char  buffer[18000] = {0};
int nrLine = 0;
int i = 0;
int j = 0;
long len = 0;
char chInt[2] = {0};
hndFile = fopen(strLeavesFileName, "r");
if (hndFile != NULL)
{
 fgets(buffer, sizeof(buffer), hndFile);
 gnAlphSize = atoi(buffer);
 while(fgets(buffer, sizeof(buffer), hndFile)!=NULL)
 {
  nrLine ++;
  len = strlen(buffer);
 }
}
fclose(hndFile);
hndFile = fopen(strLeavesFileName, "r");
ppSequences = (int**) calloc(gnGraphSize + 1, sizeof(int*));
    for( i=0; i<gnGraphSize + 1; i++)
        ppSequences[i] = (int *)calloc(len, sizeof(int));
gnNrLines = nrLine ;
gnSequenceSize = len-1;
if (hndFile != NULL)
{
 fgets(buffer, sizeof(buffer), hndFile);
 gnAlphSize = atoi(buffer);
 nrLine = gnGraphSize - gnNrLines+1;
 while(fgets(buffer, sizeof(buffer), hndFile)!=NULL)
 {
```

```
  for (i = 0; i< len-1; i++)
  {
   strncpy(chInt, &buffer[i], 1);
   ppSequences[nrLine][i] = atoi(chInt);
  }
  nrLine ++;
 }
}
fclose(hndFile);
return gnSequenceSize;
}
int CTreeEnergy::FirstLeaf()
{
int nRetLeaf = 0;
nRetLeaf = gnGraphSize - gnNrLines +1;
return nRetLeaf;
}
void CTreeEnergy::GraphEnergy()
{
double** MatrixPi;
double** MatrixSigma;
int i = 0;
int j = 0
int nIndexSeqSize = 0;
int nIndexGraphSize =0;
int nFirstLeaf = 0;
int nCrtNode = 0;
double * nRowPi;
double * nRowSigma;
double dPi = 0.0;
double dSigma = 0.0;
int nIndexAlpha = 0;
int nIndexBeta = 0;
double dTemp = 0.0;
double Z0 = 0.0;
double E0 = 0.0;
int dSeq = 0;
double dist = 0.0;
double expp = 0.0;

MatrixPi =
```

```
   (double**) calloc(gnGraphSize+1, sizeof(double*));
    for( i=0; i<gnGraphSize+1; i++)
        MatrixPi[i] =
       (double *)calloc(gnAlphSize, sizeof(double));

MatrixSigma =
   (double**) calloc(gnGraphSize+1, sizeof(double*));
    for( i=0; i<gnGraphSize+1; i++)
        MatrixSigma[i] =
       (double *)calloc(gnAlphSize, sizeof(double));

dEnergy = (double *) calloc(gnSequenceSize+1, sizeof(double));


nFirstLeaf = FirstLeaf();
for (nIndexSeqSize=0; nIndexSeqSize < gnSequenceSize;
                   nIndexSeqSize++)
{
 for ( i = 0; i< gnGraphSize; i++)
 {
  for (j = 0; j < gnAlphSize; j ++)
  {
   MatrixPi[i][j] = 1.0;
   MatrixSigma[i][j] = 0.0;
  }
 }
 for(nIndexGraphSize = gnGraphSize; nIndexGraphSize > 1;
                   nIndexGraphSize --)
 {
  nCrtNode = pGraph[nIndexGraphSize - 2];
  nRowPi = (double *) calloc(gnAlphSize+1, sizeof(double));
  nRowSigma = (double *) calloc(gnAlphSize+1, sizeof(double));

   for( i = 1; i <= gnAlphSize; i ++)
  {
   nRowPi[i] = MatrixPi[nCrtNode-1][i-1];
   nRowSigma[i] = MatrixSigma[nCrtNode-1][i-1];
  }
  dPi = 0.0;
  dSigma = 0.0;
  for (nIndexAlpha = 1; nIndexAlpha <= gnAlphSize; nIndexAlpha ++)
```

```
 {
  if (nIndexGraphSize >= nFirstLeaf)
    {
   dSeq = ppSequences[nIndexGraphSize][nIndexSeqSize];
   if(dSeq == 0)
    dSeq = 1;
   dist = Distance(nIndexAlpha, dSeq);
   expp = exp(-dist/dTemperature);
   MatrixPi[nCrtNode-1][nIndexAlpha-1] =
               nRowPi[nIndexAlpha] *
               MatrixPi[nIndexGraphSize-1][dSeq-1] *
               expp;
   MatrixSigma[nCrtNode-1][nIndexAlpha-1] =
               nRowSigma[nIndexAlpha] +
               MatrixSigma[nIndexGraphSize-1][dSeq-1]  +
               Distance(nIndexAlpha, dSeq);
  }
  else
  {
   for (nIndexBeta = 1; nIndexBeta <= gnAlphSize; nIndexBeta ++)
   {
    dTemp = nRowPi[nIndexAlpha] *
               MatrixPi[nIndexGraphSize-1][nIndexBeta-1] *
               exp(-Distance(nIndexAlpha, nIndexBeta)/
               dTemperature);
    dPi += dTemp;
    dSigma +=(nRowSigma[nIndexAlpha] +
               MatrixSigma[nIndexGraphSize-1][nIndexBeta-1] +
               Distance(nIndexAlpha, nIndexBeta)) * dTemp;
   }
   MatrixPi[nCrtNode-1][nIndexAlpha-1] = dPi;
   MatrixSigma[nCrtNode-1][nIndexAlpha-1] = dSigma/dPi;
  }
 }
 free(nRowPi);
 free(nRowSigma);
}
Z0 = 0.0;
E0 = 0.0;
for ( i = 1; i <= gnAlphSize; i ++)
{
```

```
  Z0 += MatrixPi[0][i-1];
  E0 += MatrixSigma[0][i-1] * MatrixPi[0][i-1];
 }
 E0 = E0/Z0;
 dEnergy[nIndexSeqSize+1] = E0;
 printf("Energy for feature:\%d is \%.20f $ n$",
               nIndexSeqSize, E0);
 dEnergy[0] += E0;
}
for (i=0;i<gnGraphSize+1;i++)
   free(MatrixPi[i]);
free(MatrixPi);
for (i=0;i<gnGraphSize+1;i++)
   free(MatrixSigma[i]);
free(MatrixSigma);
}
double CTreeEnergy::Distance(int a, int b)
{
if (a == b)
 return 0.0;
else
 return 1.0;
}
void CTreeEnergy::CleanMemory()
{
int i = 0;
for (i=0;i<gnGraphSize + 1;i++)
    free(ppSequences[i]);
free(ppSequences);
free(dEnergy);
free(pGraph);
}
```

8.5 Drawing function for MDS output

```
Private Sub Form_Load()
Dim j, i As Integer
Dim xDim As Double
Dim yDim As Double
Dim xCenter As Double
```

```
Dim yCenter As Double
Dim xLeft As Double
Dim yLeft As Double
Dim xRight As Double
Dim yRight As Double
Dim strLine As String
Dim x(100) As Double
Dim y(100) As Double
Dim dScale As Double
Dim k As Integer
Dim varsplits() As String
Dim varArrayStruct(127) As ArrayStruct
’Need this so that a box is drawn when frm loads.
Me.AutoRedraw = True
Me.BackColor = RGB(255, 255, 255)
Me.ForeColor = RGB(0, 0, 0)
Me.DrawStyle = 0  ’DrawStyle is solid line
Me.Cls
Me.DrawWidth = 1
xLeft = 300
yLeft = 300
xRight = 15000
yRight = 10500
Me.Line (xLeft, yLeft)-(xRight, yRight), , B   ’Draw a box
xCenter = (xRight - xLeft) / 2
yCenter = (yRight - yLeft) / 2
Me.Line (xCenter, yLeft + 100)-(xCenter, yRight - 100)
Me.Line (xLeft + 100, yCenter)-(xRight - 100, yCenter)
Me.Circle (xCenter, yCenter), 100
dScale = 2000
i = 0
Open App.Path & "$\ Outs\ coord_50_distall.TXT$" For Input As #1
Do While EOF(1) = False
     Line Input #1, strLine
     varsplits = Split(strLine, vbTab)
     Me.Circle (xCenter + dScale * varsplits(1),
yCenter - dScale * varsplits(2)), 30, vbRed
     Me.CurrentX = xCenter + dScale * varsplits(1)
     Me.CurrentY = yCenter - dScale * varsplits(2)
     Me.Print varsplits(0)
     varArrayStruct(i).id = CInt(varsplits(0))
```

```
     varArrayStruct(i).x = xCenter + dScale * varsplits(1)
     varArrayStruct(i).y = yCenter - dScale * varsplits(2)
     i = i + 1
Loop
Close #1
Open App.Path & "$\ Outs\ congr_ 50_ distall.txt$" For Input As #1
j = 0
Do While EOF(1) = False
     Line Input #1, strLine
     varsplits = Split(strLine, vbTab)
     For i = 1 To UBound(varsplits) - 1
          If CDbl(varsplits(i)) <= 0.0001 Then
               Me.DrawStyle = 0
               Me.DrawWidth = 1
               Me.ForeColor = vbRed
          End If
         If CDbl(varsplits(i)) <= 0.0001 Then
             Me.Line (varArrayStruct(j).x, varArrayStruct(j).y)-
(varArrayStruct(i - 1).x, varArrayStruct(i - 1).y)
               k = k + 1
         End If
     Next i
     j = j + 1
Loop
Close #1

End Sub
```

8.6 Conversion from WALS to WALSX

```
Dim strfile As String
Dim strLine As String
ReDim gnewFeaturesID(0)
strfile = filename
FileOpen(1, strfile, OpenMode.Input)
System.Windows.Forms.Application.DoEvents()
gError = 0
strLine = LineInput(1)
txtComment.Text = ""
Do While InStr(strLine, "#features description") = 0
```

```
 strLine = LineInput(1)
 If InStr(strLine, "#") = 0 And strLine <> "" Then
  txtComment.Text = txtComment.Text & vbCrLf & strLine
 End If
Loop
strLine = LineInput(1)
'txtComment.Text = txtComment.Text & vbCrLf & strLine
strLine = LineInput(1)
txtComment.Text = txtComment.Text & vbCrLf
'now we have features description
Do While InStr(strLine, "#values description") = 0
 If strLine $<>$ "" Then
  SaveFeaturesDescriptions((strLine))
  If gError = 1 Then
   Exit Sub
  End If
  txtComment.Text = txtComment.Text & vbCrLf & strLine
 End If
 strLine = LineInput(1)
Loop
strLine = LineInput(1)
'txtComment.Text = txtComment.Text & vbCrLf & strLine
strLine = LineInput(1)
txtComment.Text = txtComment.Text & vbCrLf
Do While InStr(strLine, "#features values") = 0
 If strLine $<>$ "" Then
  SaveValuesDescriptions((strLine))
  txtComment.Text = txtComment.Text & vbCrLf & strLine
 End If
 strLine = LineInput(1)
Loop
strLine = LineInput(1)
'txtComment.Text = txtComment.Text & vbCrLf & strLine
strLine = LineInput(1)
txtComment.Text = txtComment.Text & vbCrLf
Do While InStr(strLine, "#chance") = 0
 If strLine $<>$ "" Then
  SaveFeaturesValues((strLine))
  txtComment.Text = txtComment.Text & vbCrLf & strLine
 End If
 strLine = LineInput(1)
```

```
Loop
strLine = LineInput(1)
’txtComment.Text = txtComment.Text & vbCrLf & strLine
strLine = LineInput(1)
txtComment.Text = txtComment.Text & vbCrLf
Do While InStr(strLine, "#dependencies") = 0
  If strLine $<>$ "" Then
  SaveChanceInfo((strLine))
 txtComment.Text = txtComment.Text & vbCrLf & strLine
 End If
vstrLine = LineInput(1)
Loop
strLine = LineInput(1)
’txtComment.Text = txtComment.Text & vbCrLf & strLine
strLine = LineInput(1)
txtComment.Text = txtComment.Text & vbCrLf
Do While InStr(strLine, "#end") = 0
 If strLine $<>$ "" Then
  SaveDependenciesInfo((strLine))
 txtComment.Text = txtComment.Text & vbCrLf & strLine
 End If
 strLine = LineInput(1)
Loop

FileClose(1)
Exit Sub
```

Bibliography

[1] MySql AB. Mysql database systemv5.1, 2006. www.mysql.com.

[2] M. Albu and A. Chivu. *Asketch - Assistant Sketch for Architects. Dissertation.* Politechnica University of Bucharest, Faculty of Computer Science, 1999.

[3] M. Albu, C. Davauchelle, A. Dress, and A. Grossmann. A rank based approach to phylogenetics. unpublished manuscipt, 2006.

[4] M. Albu and A. Dress. The evolution of texts: confronting stemmatological and genetical methods. proceedings of the international workshop held in louvain-la-neuve. *Linguistica Computazionale*, 2004.

[5] J.D. Apresjan. Algoritmy postroenija klassov po matritse rasstojaiij (algorithms for building classes on the matrix of distances). *Masinnyi Perevof i Prikladnaja Lingvistika*, 9:3–18, 1966.

[6] J.D. Apresjan. *Experimental'noe issledovanie semantiki russkogo glagola (Experimental investigation of the Semantics of the Russian Verb).* Nauka, 1967.

[7] W.R. Atchley, J. Zhao, A. Fernandes, and T. Drueke. Solving the sequence metric problem. *Proc. Natl. Acad. Sci*, pages 6395–6400, 2005.

[8] B. Bickel. Typology in the 21st century: major current developments, 2006.

[9] B. Bickel and J. Nichols. The autotyp database network. electronic database, 1966. http://www.uni-leipzig.de/ autotyp.

[10] B. Bickel and J. Nichols. Typological enclaves. paper presented at the 5th biannual conference of the association for linguistic typology, 2003.

[11] B. Bickel and J. Nichols. Inclusive/exclusive as person vs. number categories worldwide. clusivity, 2005.

[12] S. Boecker and A. Dress. Maximal hierarchies. *Adv. Math.*, 151:270–282, 2000.

[13] G. Bonfante. Ideas on the kinship of the european languages from 1200 to 1800. *Cahiers d'Histoire Mondiale*, pages 679–699, 1954.

[14] P. Buneman. The recovery of trees from measures of dissimilarity. *Math. in Archaeological and Historical Sciences*, pages 387–395, 1971.

[15] L. Campbell, V. Bubenik, and L. Saxon. Word order universals: refinements and clarifications. *Canadian Journal of Linguistics*, pages 209–230, 1988.

[16] E. Coseriu. *Adam Smith und die Anfange deer Sprachtypologie*. Herbert E. Brekle - Leonhard Lipka, 1968.

[17] W. Croft. *Typology and Universals. (Cambridge Textbooks in Linguistics.)*. Cambridge University Press, 1990.

[18] W. Croft. *Typology and universals*. Cambridge: Cambridge University Press, 2002.

[19] W. Croft and K.T. Poole. *Inferring universals from grammatical variation: multidimensional scaling for typological analysis*. Ms., Center for Advanced Studies in the Behavioral Sciences, Stanford, 2004.

[20] M. Cysouw. Against implicational universals. *Linguistic Typology*, pages 89–10, 2003.

[21] M. Cysouw. *Quantitative methods in typology. Quantitative linguistics: an international handbook.* Berlin: Mouton de Gruyter, 2006.

[22] M. Cysouw, M. Albu, and A. Dress. Analyzing feature consistency using dissimilarity matrices. *Sprachtypologie und Universalienforschung*, 2006. submitted.

[23] M. Cysouw, J. Good, M. Albu, and H.J. Bibiko. Retrofitting an ontology onto the world atlas of language structures. proceedings of the e-meld workshop 2005: Morphosyntactic annotation and terminology: Linguistic ontologies and data categories for linguistic resources. 2005. http://emeld.org/workshop/2005/papers/good-paper.doc.

[24] C. Darwin. *On the Origin of Species by Means of Natural Selection*. John Murray, 1859.

[25] C. Devauchelle, A. Dress, A Grossman, S. Gruenewald, and A. Henaut. Constructing hierarchical set systems. *Annals of Combinatorics*, 8:441 – 456, 2004.

[26] A. Dress, B. Holland, K.T. Huber, J.H. Koolen, V. Moulton, and J. Weyer-Menkhoff. Delta additive and delta ultra-additive maps, gromov's trees, and the farris transform. *Discrete Applied Mathematics*, 146:51–73, 2005.

[27] M. Dryer, editor. *Why statistical universals are better than absolute universals. Papers from the 33rd Annual Meeting of the Chicago Linguistic Society*. Chicago Linguistic Society, 1998.

[28] M. S. Dryer. Large linguistic areas and language sampling. *Studies in Language*, pages 257–292, 1989.

[29] M. S. Dryer. The greenbergian word order correlations. *Language*, pages 81–138, 1992.

[30] M. Dunn, A. Terrill, G. Reesink, R. A. Foley, and S. C. Levinson. Structural phylogenetics and the reconstruction of ancient language history. *Science*, 309:2072–2075, 2005.

[31] S. Farrar and T. Langendoen. A linguistic ontology for the semantic web. *GLOT International*, 7:97–100, 2003.

[32] J Felsenstein, 2001. PHYLIP: Phylogeny Inference Package. Version 3.6.

[33] J. Fisiak. *Linguistic reconstruction and typology*. Mouton de Gruyter. Berlin, 1997.

[34] V. Flajshans. *Pisenictvi ceske slovem i obrazem od najdavnsejsuch dob az po nase casy[Czech literature in word and picture from the earliest days until our times]*. Prague: Grosman & Svoboda, 1901.

[35] R.G. Gordon. Ethnologue: Languages of the world, 2005. http://www.ethnologue.com/.

[36] GraphViz. Graphviz - graph visualization software, February 2006. http://www.graphviz.org/.

[37] R.D. Gray and Q. Atkinson. Language-tree divergence times support the anatolian theory of indo-european origins. *Nature*, pages 435–439, 2003.

[38] J.H Greenberg. *Some universals of grammar with particular reference to the order of meaningful elements. In Universals of Language*. Cambridge, Mass.: MIT Press., 1963.

[39] E. Haeckel. *Generelle Morphologie der Organismen*. G. Reimer, 1866.

[40] R.W. Hamming. Error-detecting and error-corecting codes. *Bell System ZTechnicaal Journal*, 29(2):147–160, 1950.

[41] M. Haspelmath, M.S. Dryer, D. Gil, and B. Comrie. *The world atlas of language structures*. Oxford: Oxford University Press, 2005.

[42] J.A. Hawkins. *Word order universals*. New York: Academic Press, 1983.

[43] D. H. Huson and D. Bryant. Application of phylogenetic networks in evolutionary studies. *Mol. Biol. Evol.*, 23(2):254–267, 2006.

[44] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, 1988.

[45] R. Jakobson. *K xarakteristike evrazijskogo jazykovogo sojuza. Selected Writings*. The Hague: Mouton, 2005.

[46] D. Janssed, B. Bickel, and F. Zuniga. *Randomization tests in language typology*. Ms. University of Leipzig (www.uni-leipzig.de/ bickel/research/papers), 2005.

[47] B. Kaltz. *Les vrais principes de la langue francoise de l'abbe Girard devant la critique du XVIIIe siecle a nos jours*. Konrad Koerner, 1980.

[48] M. Kendall. *Theory and Applications of Rank Order Statistics*. Charles Griffins & Co Ltd, 4th edition, 1970.

[49] P.V. Kirch and R. C. Green. Hawaiki, ancestral polynesia: An essay. *Historical Anthropology*, 2001.

[50] P. Legendre and F.-J. Lapointe. Assessing the congruence among distance matrices: single malt scotch whiskies revisited. *Australian and New Zealand Journal of Statistics*, 46:615–629, 2004.

[51] P. Legendre and F.J. Lapointe. Cadm software : Congruence among distance matrices, 2004. http://www.bio.umontreal.ca/casgrain/en/labo/cadm.html.

[52] S.C. Levinson. *Space in language and cognition*. Cambridge: Cambridge University Pres, 2003.

[53] W. H. Li, M. Tanimura, and P. M. Sharp. An evaluation of the molecular clock hypothesis using mammalian dna sequences. *J. Mol. Evol.*, 4:330–42, 1987.

[54] J.R.A. Little and D.B. Rubin. *Statistical Analysis with Missing Data*. John Wiley and Sons. New York, 1987.

[55] N. M. Luscombe, D. Greenbaum, and M. Gerstein. What is bioinformatics? a proposed definition and overview of the field. *Method Inform Med*, pages 346–58, 2001.

[56] J. Lynch, M. Ross, and T. Crowley. *The Oceanic Languages*. Curzon Press, Richmond, UK, 2002.

[57] E. Maslova. A dynamic approach to the verification of distributional universals. *Linguistic Typology*, page 33, 2000.

[58] J. Nichols. *Language diversity in space and time*. Chicago: The University of Chicago Press, 1992.

[59] J. Nichols. Modeling ancient population structures and population movement in linguistics and archeology. *Annual Review of Anthropology*, page 84, 1997.

[60] J. Nichols and D.A. Peterson. The amerind personal pronouns. *Language*, page 71, 1996.

[61] J. Noordegraaf. A few remarks on adam smith's 'dissertation'(1761). *Historiographia Linguistica*, pages 59–67, 1977.

[62] R.D.M. Page and E.C.Holmes. *Molecular Evolution. A Phylogenetic Approach*. Oxford University Press, 2005.

[63] R.D. Perkins. Statistical techniques for determining language sample size. *Studies in Language*, page 315, 1989.

[64] F. Planck. *Typology by the end of the 18th century. History of the Language Sciences: An International Handbook on the Evolution of the Study of Language from the Beginnings to the Present*, volume 2, pages 400–401. Berlin: de Gruyter, thirteenth edition, 1982.

[65] A.F. P. Rhodes and R.M. Needham. A reduction method for non-arithmetic data, and its application to thesauric translation. *Proceedings of an International UNESCO Conference on Information Processing*, pages 321–325, 1960.

[66] M. Ross. Proto oceanic and the austronesian languages of western melanesia. *Pacific Linguistics*, 1988.

[67] N. Saitou and M. Nei. The neighbour-joining method: A new method for reconstruction of phylogenetic trees. *Molecular Biology and Evolution*, 4:406–425, 1987.

[68] J.J. Scaliger. *"Diatriba de Europaeorum linguae". Appendix to Opuscula varia, antehac non edita*. Paris: H. Beys, 1610.

[69] R. W. Sinnott. Virtues of the haversine. *Sky and Telescope*, 68:159, 1984.

[70] SPSS. Statistical package for the social sciences, 2005. www.spss.com.

[71] K. Strimmer and A. Von Haeseler. Quartet puzzling: A quartet maximum-likelihood method for reconstructing tree topologies. *Molecular Biology And Evolution*, 13(7):964 –, 1996.

[72] M. Swadesh. Lexico-statistical dating of prehistoric ethnic contacts: With special reference to north american indians and eskimos. *Proceedings of the American Philosophical Society*, pages 452–463, 1952.

[73] J. van der Auwera. Revisiting the balkan and meso-american linguistic areas. *Language Sciences*, page 70, 1998.

[74] S. Wichmann and D. Kamholz. Evaluating the strength of typological features for phylogenetic analyses. manuscript, 2005.

[75] S. Wichmann and A. Saunders. How to use typological databases in historical linguistic research. *manuscript*, 2005.

Selbständigkeitserklärung

Hiermit erkläre ich, die vorliegende Dissertation selbständig und ohne unzulässige fremde Hilfe angefertigt zu haben. Ich habe keine anderen als die angeführten Quellen und Hilfsmittel benutzt und sämtliche Textstellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Schriften entnommen wurden, und alle Angaben, die auf mündlichen Auskünften beruhen, als solche kenntlich gemacht. Ebenfalls sind alle von anderen Personen bereitgestellten Materialien oder erbrachten Dienstleistungen als solche gekennzeichnet.

Leipzig, den 11.09.2006

Mihai Albu