Estimating and visualizing language similarities using weighted alignment and force-directed graph layout

Gerhard Jäger
University of Tübingen, Department of Linguistics
gerhard.jaeger@uni-tuebingen.de

Abstract

The paper reports several studies about quantifying language similarity via phonetic alignment of core vocabulary items (taken from Wichman's Automated Similarity Judgement Program data base). It turns out that weighted alignment according to the Needleman-Wunsch algorithm yields best results.

For visualization and data exploration purposes, we used an implementation of the Fruchterman-Reingold algorithm, a version of force directed graph layout. This software projects large amounts of data points to a two- or three-dimensional structure in such a way that groups of mutually similar items form spatial clusters.

The exploratory studies conducted along these ways lead to suggestive results that provide evidence for historical relationships beyond the traditionally recognized language families.

1 Introduction

The *Automated Similarity Judgment Program* (Wichmann et al., 2010) is a collection of 40-item Swadesh lists from more than 5,000 languages. The vocabulary items are all given in a uniform, if coarse-grained, phonetic transcription.

In this project, we explore various ways to compute the pairwise similarities of these languages based on sequence alignment of translation pairs. As the 40 concepts that are covered in the data base are usually thought to be resistant against borrowing, these similarities provide information about genetic relationships between languages.

To visualize and explore the emerging patterns, we make use of *Force Directed Graph Layout*. More specifically, we use the CLANS[1] implementation of the Fruchterman-Reingold algorithm (Frickey and Lupas, 2004). This algorithm takes a similarity matrix as input. Each data point is treated as a physical particle. There is a repelling force between any two particles — you may think of the particles as electrically charged with the same polarity. Similarities are treated as attracting forces, with a strength that is positively related to the similarity between the corresponding data points.

All data points are arranged in a two- or three-dimensional space. The algorithm simulates the movement of the particles along the resulting force vector and will eventually converge towards an energy minimum.

In the final state, groups of mutually similar data items form spatial clusters, and the distance between such clusters provides information about their cumulative similarity.

This approach has proven useful in bioinformatics, for instance to study the evolutionary history of protein sequences. Unlike more commonly used methods like SplitsTree (or other

[1] **Cl**uster **AN**alysis of **S**equences; freely available from http://www.eb.tuebingen.mpg.de/departments/1-protein-evolution/software/clans

phylogenetic tree algorithms), CLANS does not assume an underlying tree structure; neither does it compute a hypothetical phylogenetic tree or network. The authors of this software package, Tancred Frickey and Andrei Lupas, argue that this approach is advantageous especially in situations were a large amount of low-quality data are available:

> "An alternative approach [...] is the visualization of all-against-all pairwise similarities. This method can handle unrefined, unaligned data, including non-homologous sequences. Unlike phylogenetic reconstruction it becomes more accurate with an increasing number of sequences, as the larger number of pairwise relationships average out the spurious matches that are the crux of simpler pairwise similarity-based analyses." (Frickey and Lupas 2004, 3702)

This paper investigates two issues, that are related to the two topics of the workshop respectively:

- Which similarity measures over language pairs based on the ASJP data are apt to supply information about genetic relationships between languages?
- What are the advantages and disadvantages of a visualization method such as CLANS, as compared to the more commonly used phylogenetic tree algorithms, when applied to large scale language comparison?

2 Comparing similarity measures

2.1 The LDND distance measure

In (Bakker et al., 2009) a distance measure is defined that is based on the Levenshtein distance (= edit distance) between words from the two languages to be compared. Suppose two languages, $L1$ and $L2$, are to be compared. In a first step, *the normalized Levenshtein distances* between all word pairs from $L1$ and $L2$ are computed. (Ideally this should be 40 word pairs, but some data are missing in the data base.) This measure is defined as

$$\mathrm{nld}(x, y) \doteq \frac{d_{Lev}(x, y)}{\max(l(x), l(y))}. \tag{1}$$

The normalization term ensures that word length does not affect the distance measure.

If $L1$ and $L2$ have small sound inventories with a large overlap (which is frequently the case for tonal languages), the distances between words from $L1$ and $L2$ will be low for non-cognates because of the high probability of chance similarities. If $L1$ and $L2$ have large sound inventories with little overlap, the distance between non-cognates will be low in comparison. To correct for this effect, (Bakker et al., 2009) normalize the distance between two synonymous words from $L1$ and $L2$ by defining the normalized Levenshtein distance by the average distance between all words from $L1$ and $L2$ that are non synonymous ($39 \times 40 = 1,560$ pairs if no data are missing). The **NDLD** distance between $L1$ and $L2$ is defined as the average doubly normalized Levenshtein distance between synonymous word pairs from $L1$ and $L2$. (LDND is a distance measure rather than a similarity measure, but it is straightforward to transform the one type of measure into the other.)

In the remainder of this section, I will propose an improvement of LDND in two aspects:

- using weighted sequence alignment based on phonetic similarity, and
- correcting for the variance of alignments using an information theoretic distance measure.

2.2 Weighted alignment

The identity-based sequence alignment that underlies the computation of the Levenshtein distance is rather coarse grained because it does not consider different degrees of similarities between sounds. Consider the comparison of the English word *hand* (/hEnd/ in the ASJP transcription) to its German translation *hand* (/hant/) on the one hand and its Spanish translation *mano* (/mano/) on the other hand. As the comparison involves

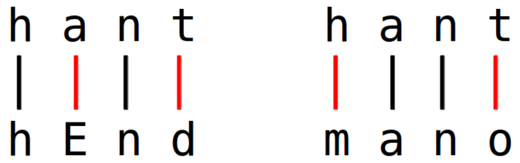


Figure 1: Simple alignment

two identical and two non-identical sounds in each case (see Figure 1), the normalized Levenshtein distance is 0.5 in both cases. It seems obvious though that /hEnd/ is much more similar to /hant/ than to /mano/, i.e. it is much more likely to find an /a/ corresponding to an /E/ in words that are cognate, and and /d/ corresponding to a /t/, than an /h/ corresponding to an /m/ or a /t/ to an /o/.

There is a parallel here to problems in bioinformatics. When aligning two protein sequences, we want to align molecules that are evolutionarily related. Since not every mutation is equally likely, not all non-identity alignments are equally unlikely. The *Needleman-Wunsch* algorithm (Needleman and Wunsch, 1970) takes a similarity matrix between symbols as an input. Given two sequences, it computes the optimal global alignment, i.e. the alignment that maximizes the sum of similarities between aligned symbols.

Following (Henikoff and Henikoff, 1992), the standard approach in bioinformatics to align protein sequences with the Needleman-Wunsch algorithm is to use the BLOSUM (*Block Substitution Matrix*), which contains the *log odds* of amino acid pairs, i.e.

$$S_{ij} \propto \log \frac{p_{ij}}{q_i \times q_j} \qquad (2)$$

Here S is the substitution matrix, p_{ij} is the probability that amino acid i is aligned with amino acid j, and q_i/q_j are the relative frequencies of the amino acids i/j.

This can straightforwardly be extrapolated to sound alignments. The relative frequencies q_i for each sound i can be determined simply by counting sounds in the ASJP data base.

The ASJP data base contains information about the family and genus membership of the languages involved. This provides a key to estimate p_{ij}. If two word x and y have the same meaning and come from two languages belonging to the same family, there is a substantial probability that they are cognates (like /hEnd/ and /hant/ in Figure 1). In this case, some of the sounds are likely to be unchanged. This in turn enforces alignment of non-identical sounds that are historically related (like /E/-/a/ and /d/-/T/ in the example).

Based on this intuition, I estimated p in the following way:[2]

- Pick a family F at random that contains at least two languages.
- Pick two languages $L1$ and $L2$ that both belong to G.
- Pick one of the forty Swadesh concepts that has a corresponding word in both languages.
- Align these two words using the Levenshtein distance algorithm and store all alignment pairs.

This procedure was repeated 100,000 times. Of course most of the word pairs involved are not cognates, but it can be assumed in these cases, the alignments are largely random (except for universal phonotactic patterns), such that genuine cognate alignments have a sufficiently large effect.

Note that language families vary considerably in size. While the data base comprises more than 1,000 Austronesian and more than 800 Niger-Congo languages, most families only consist of a handful of languages. As the procedure described above samples according to families rather than languages, languages that belong to small families are over-represented. This decision is intentional, because it prevents the algorithm from overfitting to the historically contingent properties of Austronesian, Niger-Congo, and the few other large families.

[2] A similar way to estimate sound similarities is proposed in (Prokic, 2010) under the name of *pointwise mutual information* in the context of a dialectometric study.

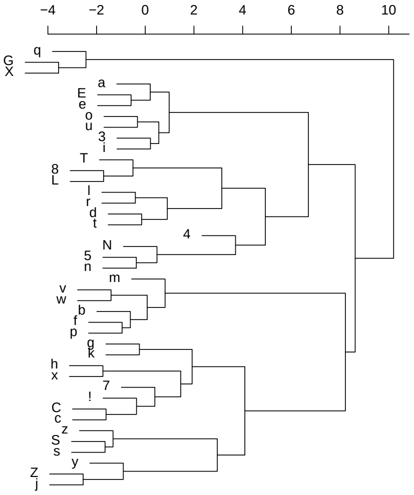


Figure 2: Sound similarities

The thus obtained log-odds matrix is visualized in Figure 2 using hierarchical clustering. The outcome is phonetically plausible. Articulatorily similar sounds — such as the vowels, the alveolar sound, the labial sounds, the dental sounds etc. — form clusters, i.e. they have high log-odds amongst each other, while the log-odds between sounds from different clusters are low.

Using weighted alignment, the similarity score for /hEnd/ $\sim$ /hant/ comes out as $\approx$ 4.1, while /hEnd/ $\sim$ /mano/ has a score of $\approx$ 0.2.

2.3 Language specific normalization

The second potential drawback of the LDND measure pertains to the second normalization step described above. The distances between translation pairs are divided by the average distance between non-translation pairs. This serves to neutralize the impact of the sound inventories of the languages involved — the distances between languages with small and similar sound inventories are generally higher than those between languages with large and/or different sound inventories.

Such a step is definitely necessary. However, dividing by the average distance does not take the effect of the variance of distances (or similarities) into account. If the distances between words from two languages have generally a low variance, the effect of cognacy among translation pairs is less visible than otherwise.

As an alternative, I propose the following similarity measure between words. Suppose s is some independently defined similarity measure (such as the inverse normalized Levenshtein distance, or the Needleman-Wunsch similarity score). For simplicity's sake, L_1 and L_2 are identified with the set of words from the respective languages in the data base:

$$s^i(x, y|L_1, L_2)$$
$$\doteq -log\frac{|\{(x',y')\in L_1\times L_2|s(x',y')\geq s(x,y)\}|}{|L_1|\times|L_2|}$$

The fraction gives the relative frequency of word pairs that are at least as similar to each other than x to y. If x and y are highly similar, this expression is close to 0. Conversely, if they are entirely dissimilar, the expression is close to 0.

The usage of the negative logarithm is motivated by information theoretic considerations. Suppose you know a word x from L_1 and you have to pick out its translation from the words in L_2. A natural search procedure is to start with the word from L_2 which is most similar to x, and then to proceed according to decreasing similarity. The number of steps that this will take (or, up to a constant factor, the relative frequency of word pairs that are more similar to each other than x to its translation) is a measure of the distance between x and its translation. Its logarithm corresponds (up to a constant factor) to the number of bits that you need to find x's translation. Its negation measures the amount of information that you gain about some word if you know its translation in the other language.

The information theoretic similarity between two languages is defined as the average similarity between its translation pairs.

2.4 Comparison

These considerations lead to four different similarity/distance measures:

- based on Levenshtein distance vs. based on Needleman-Wunsch similarity score, and

metric	*correlation*	*log-likelihood genus*	*log-likelihood family*
LDND	-0.62	-116.0	-583.6
Levenshtein^i	0.61	-110.5	-530.5
NW normalized	0.62	-108.1	-518.5
NW^i	$\mathbf{0.64}$	$\mathbf{-106.7}$	$\mathbf{-514.5}$

Table 1: Tests of the different similarity measures

- normalization via dividing by average score vs. information theoretic similarity measure.

To evaluate these measures, I defined a gold standard based on the know genetic affiliations of languages:

$$\begin{aligned} gs(L_1, L_2) &\doteq 2 \text{ if } L_1 \text{ and } L_2 \\ &\quad \text{belong to the same genus} \\ gs(L_1, L_2) &\doteq 1 \text{ if } L_1 \text{ and } L_2 \\ &\quad \text{belong to the same family} \\ &\quad \text{but not the same genus} \\ gs(L_1, L_2) &\doteq 0 \text{ else} \end{aligned}$$

Three tests were performed for each metric. 2,000 different languages were picked at random and arranged into 1,000 pairs, and the four metrics were computed for each pair. First, the correlation of these metrics with the gold standard was computed. Second, a logistic regression model was fitted, where a language pair has the value 1 if the languages belong to the same genus, and 0 otherwise. Third, the same was repeated with families rather than genera. In both cases, the log-likelihood of another sample of 1,000 language pairs according to the thus fitted models was computed.

Table 1 gives the outcomes of these tests. The information theoretic similarity measure based on the Needleman-Wunsch alignment score performs best in all three test. It achieves the highest correlation with the gold standard (the correlation coefficient for LDND is negative because it is a distance metric while the other measures are similarity metrics; only the absolute value matters for the comparison), and it assigns the highest log-likelihood on the test set both for family equivalence and for genus equivalence. We can thus conclude that this metric provides most information about the genetic relationship between languages.

3 Visualization using CLANS

The pairwise similarity between all languages in the ASJP database (excluding creoles and artificial languages) was computed according to this metric, and the resulting matrix was fed into CLANS. The outcome of two runs, using the same parameter settings, are given in Figure 3. Each circle represents one language. The circles are colored according to the genus affiliation of the corresponding language. Figure 4 gives the legend.

In both panels, the languages organize into clusters. Such clusters represent groups with a high mutual similarity. With few exceptions, all languages within such a cluster belong to the same genus. Obviously, some families (such as Austronesian — shown in dark blue — and Indo-European — shown in brown — have a high coherence and neatly correspond to a single compact cluster. Other families such as Australian — shown in light blue — and Niger-Congo — shown in red — are more scattered.

As can be seen from the two panels, the algorithm (which is initialized with a random state) may converge to different stable states with different global configurations. For instance, Indo-European is located somewhere between Austronesian, Sino-Tibetan — shown in yellow —, Trans-New-Guinea (gray) and Australian in the left panel, but between Austronesian, Austro-Asiatic (orange) and Niger-Congo (red) in the right panel. Nonetheless, some larger patterns are recurrent across simulations. For instance, the

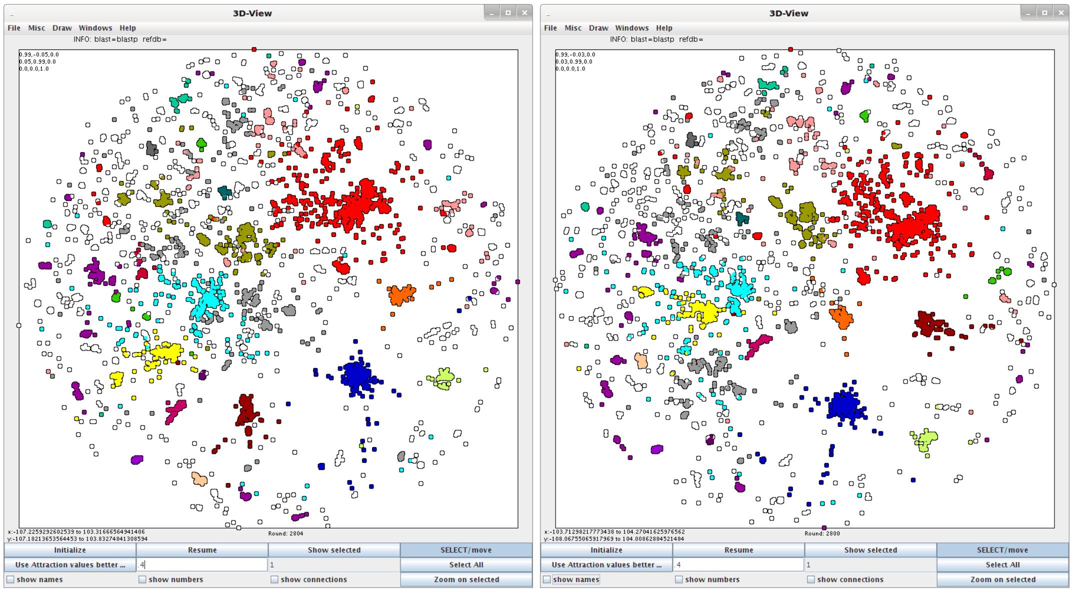
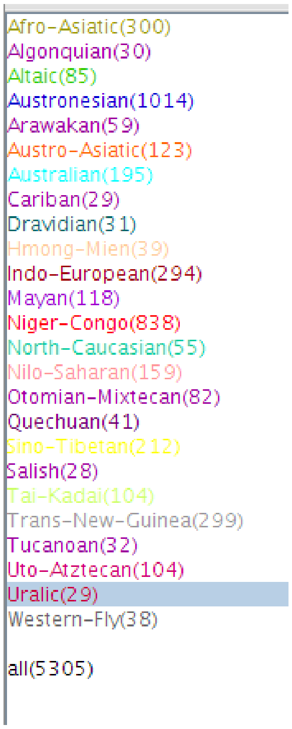


Figure 3: Languages of the world



Afro-Asiatic(300)
Algonquian(30)
Altaic(85)
Austronesian(1014)
Arawakan(59)
Austro-Asiatic(123)
Australian(195)
Cariban(29)
Dravidian(31)
Hmong-Mien(39)
Indo-European(294)
Mayan(118)
Niger-Congo(838)
North-Caucasian(55)
Nilo-Saharan(159)
Otomian-Mixtecan(82)
Quechuan(41)
Sino-Tibetan(212)
Salish(28)
Tai-Kadai(104)
Trans-New-Guinea(299)
Tucanoan(32)
Uto-Atztecan(104)
Uralic(29)
Western-Fly(38)

all(5305)

Figure 4: Legend for Figure 3

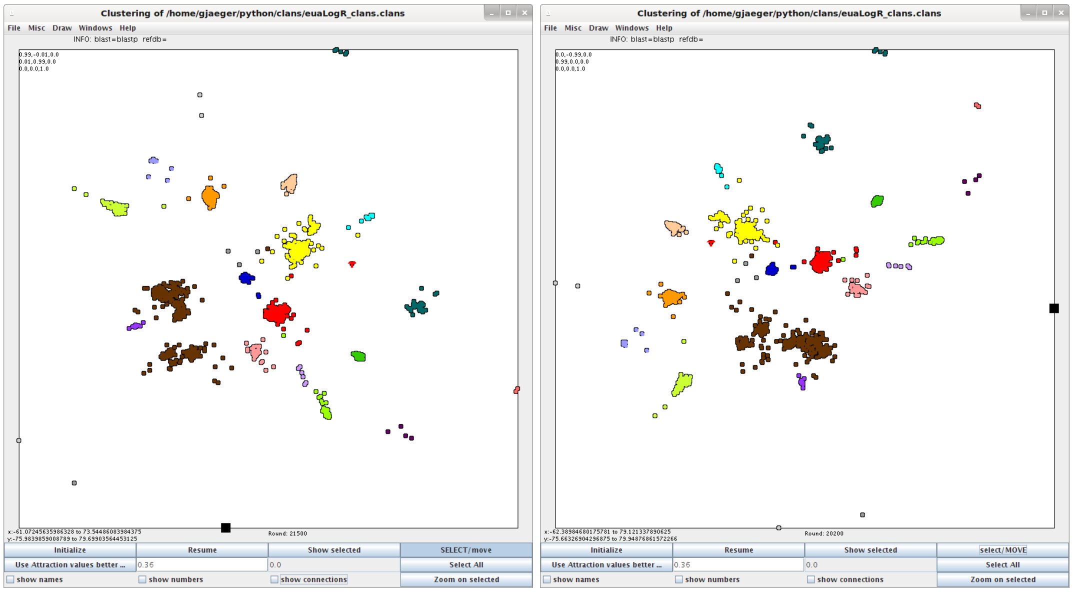
Tai-Kadai languages (light green) always end up in the proximity of the Austronesian languages. Likewise, the Nilo-Saharan languages (pink) do not always form a contiguous cluster, but they are always near the Niger-Congo languages.

It is premature to draw conclusions about deep genetic relationships from such observations. Nonetheless, they indicate the presence of weak but non-negligible similarities between these families that deserve investigation. Visualization via CLANS is a useful tool to detect such weak signals in an exploratory fashion.

4 The languages of Eurasia

Working with all 5,000+ languages at once introduces a considerable amount of noise. In particular the languages of the Americas and of Papua New Guinea do not show stable relationships to other language families. Rather, they are spread over the entire panel in a seemingly random fashion. Restricting attention to the languages of Eurasia (also including those Afro-Asiatic languages that are spoken in Africa) leads to more pronounced global patterns.

In Figure 5 the outcome of two CLANS runs is shown. Here the global pattern is virtually identical across runs (modulo rotation). The Dravid-



Clustering of /home/gjaeger/python/clans/euaLogR_clans.clans

INFO: blast=blastp refdb=

0.99,-0.01,0.0
0.01,0.99,0.0
0.0,0.0,1.0

x:-61.07245635986328 to 73.54486083984375
y:-75.9839859008789 to 79.69903564453125

Round: 21500

Use Attraction values better ... 0.36 0.0

□ show names □ show numbers □ show connections

Clustering of /home/gjaeger/python/clans/euaLogR_clans.clans

INFO: blast=blastp refdb=

0.0,-0.99,0.0
0.99,0.0,0.0
0.0,0.0,1.0

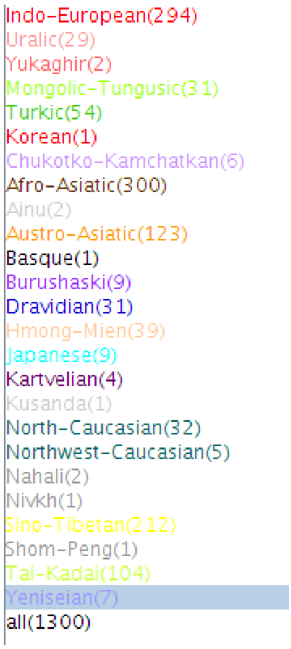
x:-62.38984680175781 to 79.121337890625
y:-75.66326904296875 to 79.94876861572266

Round: 20200

Use Attraction values better ... 0.36 0.0

□ show names □ show numbers □ show connections

Figure 5: The languages of Eurasia



Indo-European(294)
Uralic(29)
Yukaghir(2)
Mongolic-Tungusic(31)
Turkic(54)
Korean(1)
Chukotko-Kamchatkan(6)
Afro-Asiatic(300)
Ainu(2)
Austro-Asiatic(123)
Basque(1)
Burushaski(9)
Dravidian(31)
Hmong-Mien(39)
Japanese(9)
Kartvelian(4)
Kusanda(1)
North-Caucasian(32)
Northwest-Caucasian(5)
Nahali(2)
Nivkh(1)
Sino-Tibetan(212)
Shom-Peng(1)
Tai-Kadai(104)
Yeniseian(7)
all(1300)

Figure 6: Legend for Figure 5

ian languages (dark blue) are located at the center. Afro-Asiatic (brown), Uralic (pink), Indo-European (red), Sino-Tibetan (yellow), Hmong-Mien (light orange), Austro-Asiatic (orange), and Tai-Kadai (yellowish light green) are arranged around the center. Japanese (light blue) is located further to the periphery outside Sino-Tibetan. Outside Indo-European the families Chukotko-Kamchatkan (light purple), Mongolic-Tungusic (lighter green), Turkic (darker green)[3] Kartvelian (dark purple) and Yukaghir (pinkish) are further towards the periphery beyond the Turkic languages. The Caucasian languages (both the North Caucasian languages such as Lezgic and the Northwest-Caucasian languages such as Abkhaz) are located at the periphery somewhere between Indo-European and Sino-Tibetan. Burushaski (purple) is located near to the Afro-Asiatic languages.

Some of these pattern coincide with proposals about macro-families that have been made in the literature. For instance the relative proximity of

[3] According to the categorization used in ASJP, the Mongolic, Tungusic, and Turkic languages form the genus Altaic. This classification is controversial in the literature. In CLANS, Mongolic/Tungusic consistently forms a single cluster, and likewise does Turkic, but there is no indication that there is a closer relation between these two groups.

Indo-European, Uralic, Chukotko-Kamchatkan, Mongolic-Tungusic, the Turkic languages, and Kartvelian is reminiscent of the hypothetical Nostratic super-family. Other patterns, such as the consistent proximity of Japanese to Sino-Tibetan, is at odds with the findings of historical linguistics and might be due to language contact. Other patterns, such as the affinity of Burushaski to the Afro-Asiatic languages, appear entirely puzzling.

5 Conclusion

CLANS is a useful tool to aid automatic language classification. An important advantage of this software is its computational efficiency. Producing a cluster map for a 5,000 × 5,000 similarity matrix hardly takes more than an hour on a regular laptop, while it is forbidding to run a phylogenetic tree algorithm with this hardware and this amount of data. Next to this practical advantage, CLANS presents information in a format that facilitates the discovery of macroscopic patterns that are not easily discernible with alternative methods. Therefore it is apt to be a useful addition to the computational toolbox of modern data-oriented historical and typological language research.

Acknowledgments

I would like to thank Andrei Lupas, David Erschler, and Søren Wichmann for many inspiring discussions. Furthermore I am grateful for the comments I received from the reviewers for this workshop.

References

Dik Bakker, André Müller, Viveka Velupillai, Søren Wichmann, Cecil H. Brown, Pamela Brown, Dmitry Egorov, Robert Mailhammer, Anthony Grant, and Eric W. Holman. 2009. Adding typology to lexicostatistics: a combined approach to language classification. *Linguistic Typology*, 13:167–179.

Tancred Frickey and Andrei N. Lupas. 2004. Clans: a java application for visualizing protein families based on pairwise similarity. *Bioinformatics*, 20(18):3702–3704.

Steven Henikoff and Jorja G. Henikoff. 1992. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–9.

Saul B. Needleman and Christian D. Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48:443453.

Jelena Prokic. 2010. *Families and Resemblances*. Ph.D. thesis, Rijksuniversiteit Groningen.

Søren Wichmann, André Müller, Viveka Velupillai, Cecil H. Brown, Eric W. Holman, Pamela Brown, Sebastian Sauppe, Oleg Belyaev, Matthias Urban, Zarina Molochieva, Annkathrin Wett, Dik Bakker, Johann-Mattis List, Dmitry Egorov, Robert Mailhammer, David Beck, and Helen Geyer. 2010. The ASJP Database (version 13). http://email.eva.mpg.de/~wichmann/ASJPHomePage.htm.