Phylogenetische Methoden in der Historischen Linguistik Phylogenetische Inferenz mit den ASJP-Daten

Gerhard Jäger

13. Januar 2015
Forum Scientiarum

Determining distances between word lists

- two steps:
  - compute similarity/distance between individual word forms
  - aggregate word distances to doculect distances

Word distances

- based on string *alignment*
- baseline: Levenshtein alignment ⇒ count matches and mis-matches

```
h a n t      h a n t
| | | |      | | | |
h E n d      m a n o
```

- too crude as it totally ignores sound correspondences

Capturing sound correspondences

- weighted alignment using **P**ointwise **M**utual **I**nformation (PMI, a.k.a. *log-odds*):

$$s(a, b) = \log \frac{p(a, b)}{q(a)q(b)}$$

  - $p(a, b)$: probability of sound a being etymologically related to sound b in a pair of cognates
  - $q(a)$: relative frequency of sound a
- **Needleman-Wunsch algorithm:** given a matrix of pairwise PMI scores between individual symbols and two strings, it returns the alignment that maximizes the aggregate PMI score
- but first we need to estimate $p(a, b)$ and $q(a), q(b)$ for all soundclasses a and b
- $q(a)$: relative frequency of occurence of segment a in all words in ASJP
- $p(a, b)$: that's a bit more complicated...

Computing the weighted alignment score

- Dynamic Programming

	–	m	E	n	S
–	0	−2.5	−4.1	−5.7	−7.3
m	−2.5				
e	−4.1				
n	−5.7				
E	−7.3				
s	−8.9				

Computing the weighted alignment score

- Dynamic Programming

	–	m	E	n	S
–	0	−2.5	−4.1	−5.7	−7.3
m	−2.5				
e	−4.1				
n	−5.7				
E	−7.3				
s	−8.9				

Computing the weighted alignment score

- Dynamic Programming

	–	m	E	n	S
–	0	−2.5	−4.1	−5.7	−7.3
m	−2.5				
e	−4.1				
n	−5.7				
E	−7.3				
s	−8.9				

Computing the weighted alignment score

- Dynamic Programming

	–	m	E	n	S
–	0	−2.5	−4.1	−5.7	−7.3
m	−2.5	4.13			
e	−4.1				
n	−5.7				
E	−7.3				
s	−8.9				

Computing the weighted alignment score

- Dynamic Programming

	–	m	E	n	S
–	0	−2.5	−4.1	−5.7	−7.3
m	−2.5	4.13			
e	−4.1				
n	−5.7				
E	−7.3				
s	−8.9				

Computing the weighted alignment score

- Dynamic Programming

	–	m	E	n	S
–	0	−2.5	−4.1	−5.7	−7.3
m	−2.5	4.13			
e	−4.1				
n	−5.7				
E	−7.3				
s	−8.9				

Computing the weighted alignment score

- Dynamic Programming

	–	m	E	n	S
–	0	−2.5	−4.1	−5.7	−7.3
m	−2.5	4.13	1.53		
e	−4.1				
n	−5.7				
E	−7.3				
s	−8.9				

Computing the weighted alignment score

- Dynamic Programming

	–	m	E	n	S
–	0	−2.5	−4.1	−5.7	−7.3
m	−2.5	4.13	1.53	0.03	
e	−4.1				
n	−5.7				
E	−7.3				
s	−8.9				

Computing the weighted alignment score

- Dynamic Programming

	–	m	E	n	S
–	0	−2.5	−4.1	−5.7	−7.3
m	−2.5	4.13	1.53	0.03	−1.47
e	−4.1				
n	−5.7				
E	−7.3				
s	−8.9				

Computing the weighted alignment score

- Dynamic Programming

	–	m	E	n	S
–	0	−2.5	−4.1	−5.7	−7.3
m	−2.5	4.13	1.53	0.03	−1.47
e	−4.1	1.53			
n	−5.7				
E	−7.3				
s	−8.9				

Computing the weighted alignment score

- Dynamic Programming

	–	m	E	n	S
–	0	−2.5	−4.1	−5.7	−7.3
m	−2.5	4.13	1.53	0.03	−1.47
e	−4.1	1.53	5.65		
n	−5.7				
E	−7.3				
s	−8.9				

Computing the weighted alignment score

- Dynamic Programming

	–	m	E	n	S
–	0	−2.5	−4.1	−5.7	−7.3
m	−2.5	4.13	1.53	0.03	−1.47
e	−4.1	1.53	5.65	3.05	
n	−5.7				
E	−7.3				
s	−8.9				

Computing the weighted alignment score

- Dynamic Programming

	–	m	E	n	S
–	0	−2.5	−4.1	−5.7	−7.3
m	−2.5	4.13	1.53	0.03	−1.47
e	−4.1	1.53	5.65	3.05	1.55
n	−5.7				
E	−7.3				
s	−8.9				

Computing the weighted alignment score

- Dynamic Programming

	–	m	E	n	S
–	0	−2.5	−4.1	−5.7	−7.3
m	−2.5	4.13	1.53	0.03	−1.47
e	−4.1	1.53	5.65	3.05	1.55
n	−5.7	0.03			
E	−7.3				
s	−8.9				

Computing the weighted alignment score

- Dynamic Programming

	–	m	E	n	S
–	0	−2.5	−4.1	−5.7	−7.3
m	−2.5	4.13	1.53	0.03	−1.47
e	−4.1	1.53	5.65	3.05	1.55
n	−5.7	0.03	3.05		
E	−7.3				
s	−8.9				

Computing the weighted alignment score

- Dynamic Programming

	–	m	E	n	S
–	0	−2.5	−4.1	−5.7	−7.3
m	−2.5	4.13	1.53	0.03	−1.47
e	−4.1	1.53	5.65	3.05	1.55
n	−5.7	0.03	3.05	9.2	
E	−7.3				
s	−8.9				

Computing the weighted alignment score

- Dynamic Programming

	–	m	E	n	S
–	0	−2.5	−4.1	−5.7	−7.3
m	−2.5	4.13	1.53	0.03	−1.47
e	−4.1	1.53	5.65	3.05	1.55
n	−5.7	0.03	3.05	9.2	6.6
E	−7.3				
s	−8.9				

Computing the weighted alignment score

- Dynamic Programming

	–	m	E	n	S
–	0	−2.5	−4.1	−5.7	−7.3
m	−2.5	4.13	1.53	0.03	−1.47
e	−4.1	1.53	5.65	3.05	1.55
n	−5.7	0.03	3.05	9.2	6.6
E	−7.3	−1.47			
s	−8.9				

Computing the weighted alignment score

- Dynamic Programming

	–	m	E	n	S
–	0	−2.5	−4.1	−5.7	−7.3
m	−2.5	4.13	1.53	0.03	−1.47
e	−4.1	1.53	5.65	3.05	1.55
n	−5.7	0.03	3.05	9.2	6.6
E	−7.3	−1.47	4.75		
s	−8.9				

Computing the weighted alignment score

- Dynamic Programming

	–	m	E	n	S
–	0	−2.5	−4.1	−5.7	−7.3
m	−2.5	4.13	1.53	0.03	−1.47
e	−4.1	1.53	5.65	3.05	1.55
n	−5.7	0.03	3.05	9.2	6.6
E	−7.3	−1.47	4.75	6.6	
s	−8.9				

Computing the weighted alignment score

- Dynamic Programming

	–	m	E	n	S
–	0	−2.5	−4.1	−5.7	−7.3
m	−2.5	4.13	1.53	0.03	−1.47
e	−4.1	1.53	5.65	3.05	1.55
n	−5.7	0.03	3.05	9.2	6.6
E	−7.3	−1.47	4.75	6.6	7.62
s	−8.9				

Computing the weighted alignment score

- Dynamic Programming

	–	m	E	n	S
–	0	−2.5	−4.1	−5.7	−7.3
m	−2.5	4.13	1.53	0.03	−1.47
e	−4.1	1.53	5.65	3.05	1.55
n	−5.7	0.03	3.05	9.2	6.6
E	−7.3	−1.47	4.75	6.6	7.62
s	−8.9	−2.97			

Computing the weighted alignment score

- Dynamic Programming

	–	m	E	n	S
–	0	−2.5	−4.1	−5.7	−7.3
m	−2.5	4.13	1.53	0.03	−1.47
e	−4.1	1.53	5.65	3.05	1.55
n	−5.7	0.03	3.05	9.2	6.6
E	−7.3	−1.47	4.75	6.6	7.62
s	−8.9	−2.97	2.15		

Computing the weighted alignment score

- Dynamic Programming

	–	m	E	n	S
–	0	−2.5	−4.1	−5.7	−7.3
m	−2.5	4.13	1.53	0.03	−1.47
e	−4.1	1.53	5.65	3.05	1.55
n	−5.7	0.03	3.05	9.2	6.6
E	−7.3	−1.47	4.75	6.6	7.62
s	−8.9	−2.97	2.15	5.1	

Computing the weighted alignment score

- Dynamic Programming

	–	m	E	n	S
–	0	−2.5	−4.1	−5.7	−7.3
m	−2.5	4.13	1.53	0.03	−1.47
e	−4.1	1.53	5.65	3.05	1.55
n	−5.7	0.03	3.05	9.2	6.6
E	−7.3	−1.47	4.75	6.6	7.62
s	−8.9	−2.97	2.15	5.1	8.84

Computing the weighted alignment score

- Dynamic Programming

	–	m	E	n	S
–	0	−2.5	−4.1	−5.7	−7.3
m	−2.5	4.13	1.53	0.03	−1.47
e	−4.1	1.53	5.65	3.05	1.55
n	−5.7	0.03	3.05	9.2	6.6
E	−7.3	−1.47	4.75	6.6	7.62
s	−8.9	−2.97	2.15	5.1	8.84

- memorizing in each step which of the three cells to the left and above gave rise to the current entry lets us recover the corresponing optimal alignment

Computing the weighted alignment score

- Dynamic Programming

	–	m	E	n	S
–	0	−2.5	−4.1	−5.7	−7.3
m	−2.5	4.13	1.53	0.03	−1.47
e	−4.1	1.53	5.65	3.05	1.55
n	−5.7	0.03	3.05	9.2	6.6
E	−7.3	−1.47	4.75	6.6	7.62
s	−8.9	−2.97	2.15	5.1	8.84

- memorizing in each step which of the three cells to the left and above gave rise to the current entry lets us recover the corresponing optimal alignment

Computing the weighted alignment score

- Dynamic Programming

	–	m	E	n	S
–	0	−2.5	−4.1	−5.7	−7.3
m	−2.5	4.13	1.53	0.03	−1.47
e	−4.1	1.53	5.65	3.05	1.55
n	−5.7	0.03	3.05	9.2	6.6
E	−7.3	−1.47	4.75	6.6	7.62
s	−8.9	−2.97	2.15	5.1	8.84

- memorizing in each step which of the three cells to the left and above gave rise to the current entry lets us recover the corresponing optimal alignment

Computing the weighted alignment score

- Dynamic Programming

	—	m	E	n	S
—	0	−2.5	−4.1	−5.7	−7.3
m	−2.5	4.13	1.53	0.03	−1.47
e	−4.1	1.53	↖ 5.65	3.05	1.55
n	−5.7	0.03	3.05	↖ 9.2	6.6
E	−7.3	−1.47	4.75	6.6 ↑	7.62
s	−8.9	−2.97	2.15	5.1	↖ 8.84

- memorizing in each step which of the three cells to the left and above gave rise to the current entry lets us recover the corresponing optimal alignment

Computing the weighted alignment score

- Dynamic Programming

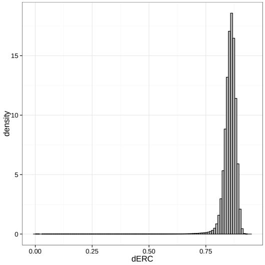
	–	m	E	n	S
–	0	−2.5	−4.1	−5.7	−7.3
m	−2.5	4.13	1.53	0.03	−1.47
e	−4.1	1.53	5.65	3.05	1.55
n	−5.7	0.03	3.05	9.2	6.6
E	−7.3	−1.47	4.75	6.6	7.62
s	−8.9	−2.97	2.15	5.1	8.84

- memorizing in each step which of the three cells to the left and above gave rise to the current entry lets us recover the corresponing optimal alignment

```
m E n - S
m e n E s
```

Capturing sound correspondences

- **First step:** automatically compile a list of language pairs that are (fairly) certain to be related
- start with a measure for language dissimilarity based on Levenshtein alignment



- all language pairs with dissimilarity ≤ 0.7 (ca. 1% of all pairs) qualify as *probably related*

Capturing sound correspondences

- doculects *probably related* (in this sense) to English:

AFRIKAANS, ALSATIAN, BERNESE_GERMAN, BRABANTIC, CIMBRIAN, DANISH, DUTCH, EASTERN_FRISIAN, FAROESE, FRANS_VLAAMS, FRISIAN_WESTERN, GJESTAL_NORWEGIAN, ICELANDIC, JAMTLANDIC, LIMBURGISH, LUXEMBOURGISH, NORTH_FRISIAN_AMRUM, NORTHERN_LOW_SAXON, NORWEGIAN_BOKMAAL, NORWEGIAN_NYNORSK_TOTEN, NORWEGIAN_RIKSMAL, PLAUTDIETSCH, SANDNES_NORWEGIAN, SAXON_UPPER, SCOTS, STANDARD_GERMAN, STELLINGWERFS, SWABIAN, SWEDISH, WESTVLAAMS, YIDDISH_EASTERN, YIDDISH_WESTERN, ZEEUWS

- these are all and only the Germanic languages
- 99.9% of all probably related pairs belong to the same family, and 60% to the same genus

Capturing sound correspondences

- **Second step:**
  - let L_1 and L_2 be *probably related*
  - every pair of words w_1/w_2 from L_1/L_2 sharing the same meaning are considered *potentially cognate*
  - all potential cognate pairs are (Levenshtein-)aligned
  - relative frequency of a being aligned with b is used as estimate of $s(a, b)$
  - all potential cognate pairs are Needleman-Wunsch aligned using PMI scores obtained in the previous step
  - all potential cognate pairs with an aggregate PMI score ≥ 5.0 are considered *probable cognates*
  - $s(a, b)$ is re-estimated using only probable cognate pairs
  - this is repeated ten times

Capturing sound correspondences

- only probabe cognate between English and Latin: `pers3n/persona`
- probable cognates English/German:

fiS	fiS
laus	laus
bl3d	blut
horn	horn
brest	brust
liv3r	leb3r
star	StErn
wat3r	vas3r
ful	fol

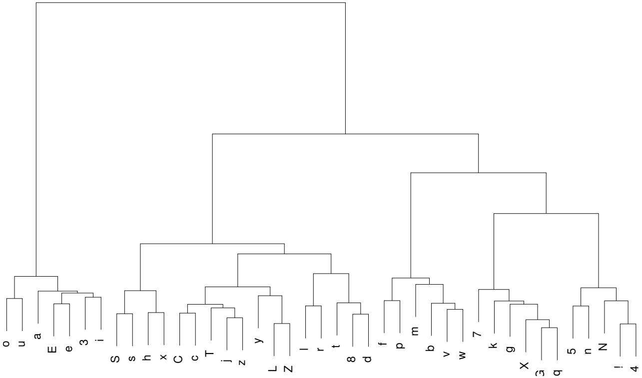
Capturing sound correspondences

- procedures results in pairwise PMI scores for each pair from the 41 ASJP sound classes
- positive PMI-score between a and b: evidence for etymological relatedness
- negative PMI-score between a and b: evidence against etymological relatedness

	a	**e**	**i**	**o**	**u**	**p**	**b**	**d**	**t**	**8**	**s**	**h**
a	**1.88**	−1.35	−2.35	−1.66	−2.54	−8.49	−8.82	−7.07	−7.03	−4.64	−8.78	−8.40
e	−1.35	**2.40**	−0.48	−1.52	−2.88	−7.47	−7.80	−7.66	−6.01	−5.01	−7.76	−7.38
i	−2.35	−0.48	**2.37**	−2.81	−1.32	−6.75	−8.46	−8.33	−8.98	−3.48	−7.04	−6.66
o	−1.66	−1.52	−2.81	**2.48**	−0.27	−7.08	−8.10	−7.96	−8.61	−5.31	−8.06	−7.68
u	−2.54	−2.88	−1.32	−0.27	**2.76**	−6.62	−8.05	−7.91	−8.56	−5.26	−8.01	−7.63
p	−8.49	−7.47	−6.75	−7.08	−6.62	**3.69**	**0.36**	−6.59	−4.30	−3.94	−2.70	−0.49
b	−8.82	−7.80	−8.46	−8.10	−8.05	**0.36**	**3.62**	−4.84	−5.09	−3.58	−5.63	−3.24
d	−7.07	−7.66	−8.33	−7.96	−7.91	−6.59	−4.84	**3.41**	−0.10	**2.52**	−2.29	−2.81
t	−7.03	−6.01	−8.98	−8.61	−8.56	−4.30	−5.09	−0.10	**3.15**	**2.11**	−1.67	−1.76
8	−4.64	−5.01	−3.48	−5.31	−5.26	−3.94	−3.58	**2.52**	**2.11**	**5.49**	**1.92**	−0.85
s	−8.78	−7.76	−7.04	−8.06	−8.01	−2.70	−5.63	−2.29	−1.67	**1.92**	**3.50**	**0.26**
h	−8.40	−7.38	−6.66	−7.68	−7.63	−0.49	−3.24	−2.81	−1.76	−0.85	**0.26**	**3.50**

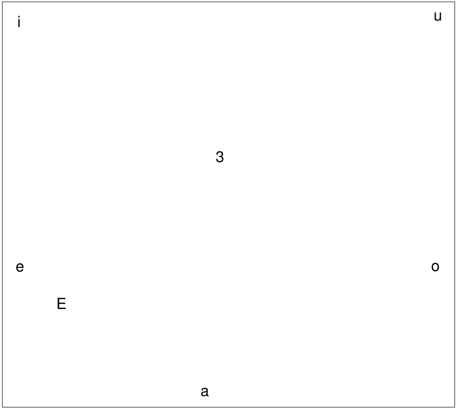
Capturing sound correspondences

- hierarchical clustering of sound classes according to PMI scores:

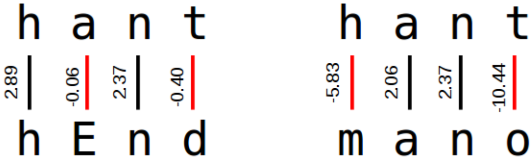


Capturing sound correspondences

- multidimensional scaling of vowel classes according to PMI scores:



Weighted alignment



$\Sigma = 4.80$ $\Sigma = -11.85$

Weighted alignment

- alignments German/Latin:

```
iX-
ego

du
tu

vir--
--nos

ain-s
-unus

cvai
d-uo

--mEnS
homo--

fiS---
piskis

hun-t
kanis

--la-u--s
pedikulus
```

```
--baum
arb-or

b-lat
folu-

haut--
k-utis

--blut
saNgis

knoX3n
--os--

-or--
auris

a-ug3-
okulus

naz3-
nasus

can-
dens
```

```
cuN-3
liNgE

k-ni
genu

han-t
manus

b--rust
pektus-

leb3r
yekur

triNk3n-
b-i-bere

--ze-3n
widere-

--her3n
audire-

Sterb3n
-mor-i-
```

```
kom3n---
w--enire

zon3
sol-

StErn-
ste-la

vas3r
-aka-

Sta-in
-lapis

foi--a-
--iNnis

p--at
viya-

bErk
mons

naxt
noks
```

```
f---ol
plenus

no-i-
nowus

nam3-
nomen
```

Weighted alignment

- alignments German/Cimbrian:

```
iX
ix

du
dE

vir
bar

cvai-
sb-en

mEn-S
menEs

hunt
hunt

laus
laus

baum
p-om

blat
-lop

blut
plut

knoX3n
-po-an

horn
horn

o-r
oar

aug3
-ogE

--n--az3
kanipa--

cuN3-----
--gaprext

hant
hant

brus---t
p-uzamEn

leb3r-
lEbara

triNk3n
trink--

ze3n
ze-g

her3n
hor--

Sterb3n
sterb--

kom3n
kEm--

zon3
zuna

StE-rn
stEarn

vas3r
basar

St-ain
stoa-n

foia-
bo-ar

vek---
bEgale

bErk
perg

naxt
naxt

--fol--
gabasEt

noi
noy

nam3
namo
```

Aggregating word similarites

- Needleman-Wunsch alignment returns a *similarity score* for each word pair
- not too reliable to identify cognates:
  - often low scores for genuine cognate pairs (‘false negatives’):
    - lat. *genu*/eng. *knee*: -3.39
    - lat. *unus*/eng. *one*: -5.00
  - occasionally high scores for non-cognates (‘chance similarities’/‘false positives’):
    - grm. *Blatt* ('leaf')/Tilquiapan *bldag* ('leaf'): 0.22
    - lat. *oculus* ('eye)/Lachixio *ikulu* ('eye'): 6.72
- approach pursued here:
  - for each language pair, estimate amount of chance similarities
  - quantify to what degree the observed similarities exceed expected chance similarities

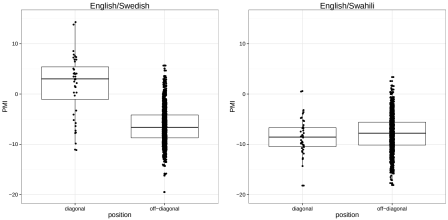
Aggregating word distances

English / Swedish

	Ei	**yu**	**wi**	**w3n**	**tu**	**fiS**	...
yog	**−7.77**	0.75	−7.68	−7.90	−8.57	−10.50	
du	−7.62	**0.33**	−5.71	−7.41	2.66	−8.57	
vi	−2.72	−2.83	**4.04**	−1.34	−6.45	0.70	
et	−5.47	−7.87	−5.47	**−6.43**	−1.83	−4.70	
tvo	−7.91	−4.27	−3.64	−4.57	**0.39**	−6.98	
fisk	−7.45	−11.2	−3.07	−9.97	−8.66	**7.58**	
⋮							

- values along diagonal give similarity between candidates for cognacy (possibility of meaning change is disregarded)
- values off diagonal provide sample of similarity distribution between non-cognates

Aggregating word distances



- distance between two word lists is a measure for how much the distribution along the diagonal differs from the distribution off the diagonal

Aggregating word distances

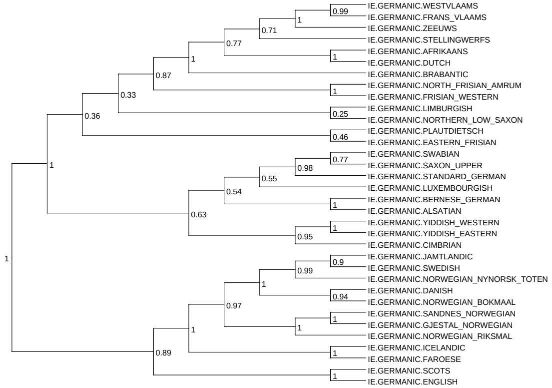
- some examples

A	B	$d(A, B)$
English	Scots	0.2139
Danish	Swedish	0.2773
English	Swedish	0.3981
English	Frisian	0.4215
English	Dutch	0.4040
Hindi	Farsi	0.6231
English	French	0.7720
English	Hindi	0.7735
Amharic	Vietnamese	0.8566
Swahili	Warlpiri	0.8573
Navajo	Dyirbal	0.8436
Japanese	Haida	0.8504
English	Swahili	0.8901

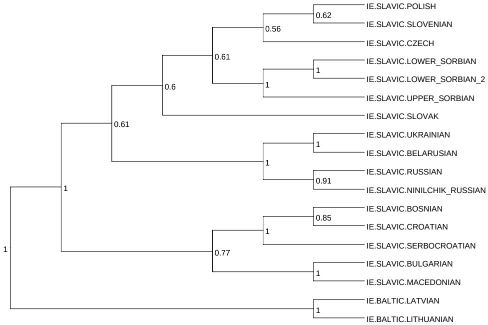
Phylogenetic inference

- pairwise distances for all (extant) languages present in ASJP are computed
- resulting distance matrix is fed into distance-based phylogenetic algorithm (*Neighbor Joining* + *Ordinary Least Square Nearest Neighbor Interchange Optimization*)
- outcome recognizes language families and their internal structure remarkably well

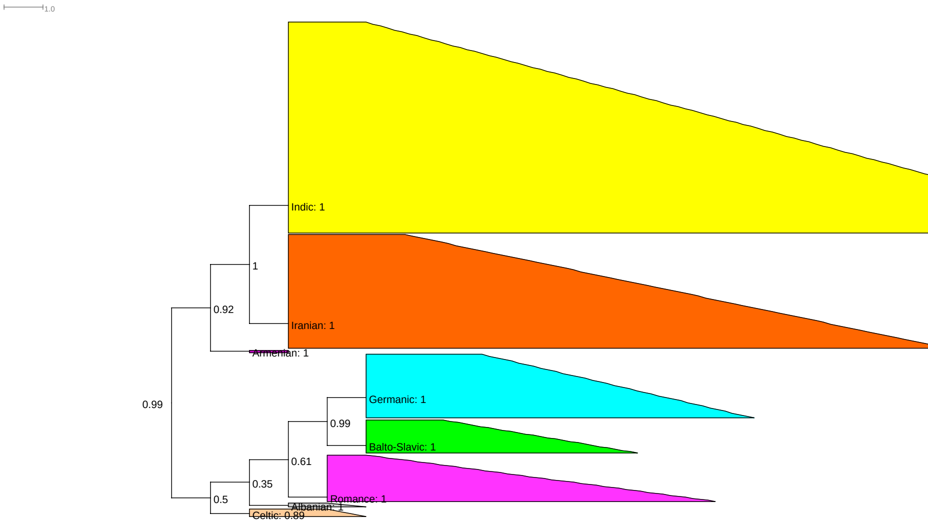
Phylogenetic inference



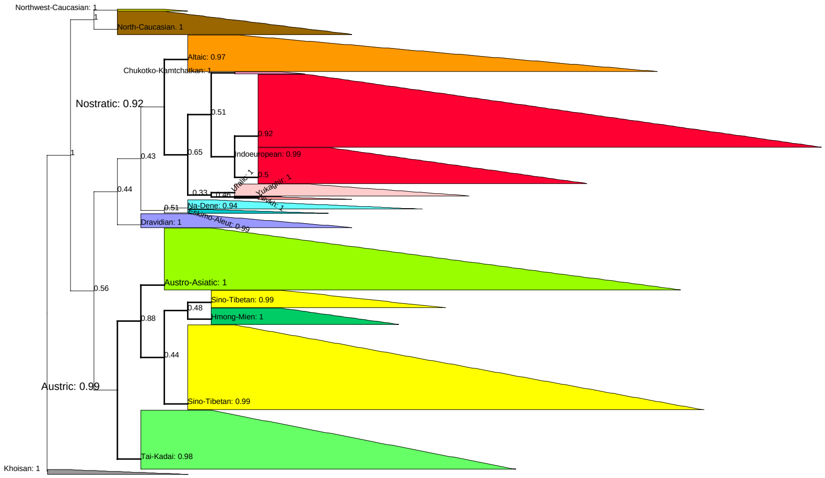
Phylogenetic inference



Phylogenetic inference



Phylogenetic inference

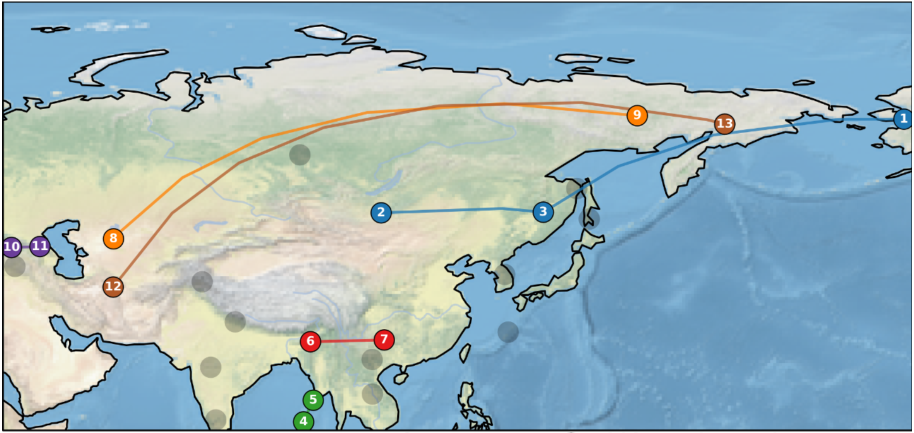


Distant relationships

(joint work with Cecil Brown, Eric Holman, Johann-Mattis List and Søren Wichmann)

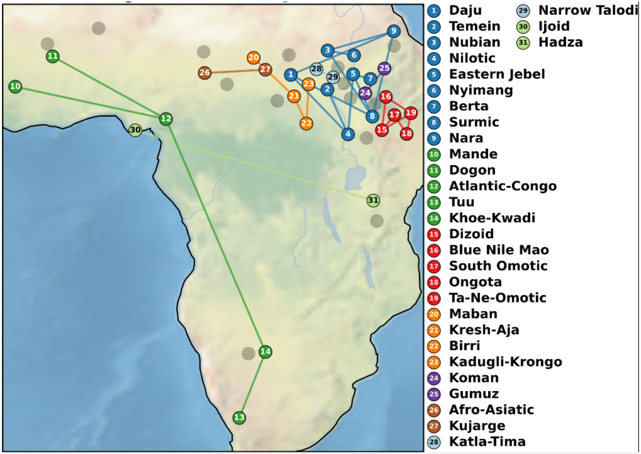
- compute aggregate distances between language families
- find threshold with *false discovery rate* of 5%: all families pairs with a distance below this threshold are genuinely related (due to common descent or contact) with a confidence or 95%

Distant relationships



1 Eskimo-Aleut
2 Mongolic
3 Tungusic
4 Jarawa-Onge
5 Great Andamanese
6 Sino-Tibetan
7 Hmong-Mien
8 Turkic
9 Yukaghir
10 Abkhaz-Adyge
11 Nakh-Daghestanian
12 Indo-European
13 Chukotko-Kamchatkan

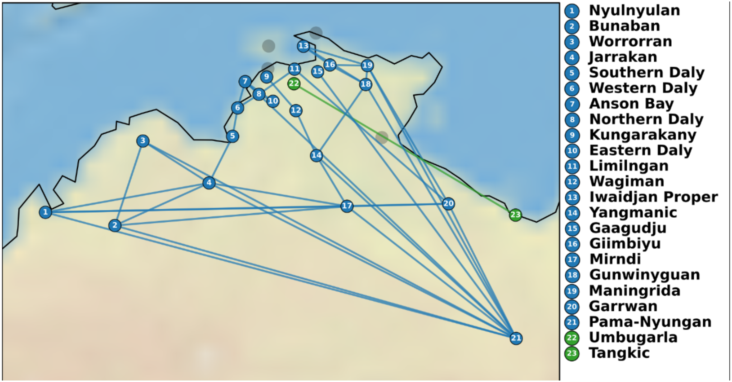
Distant relationships

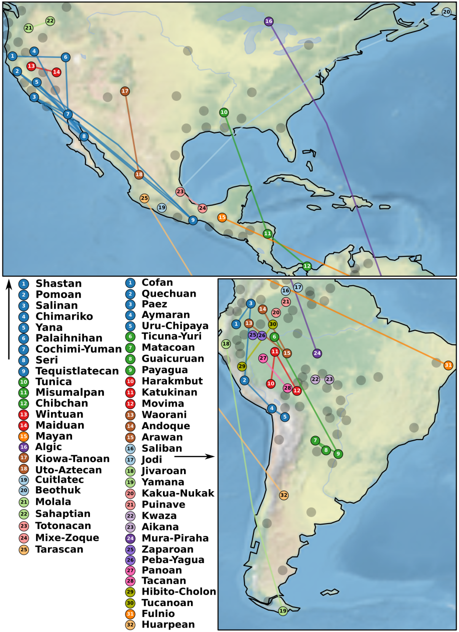


Distant relationships

1. Nimboran
2. Kosare
3. Elseng
4. Border
5. Suki-Gogodala
6. Nuclear Torricelli
7. Tirio
8. Waia
9. Kiwaian
10. Taiap
11. Nuclear Trans-New Guinea
12. Lepki-Murkim
13. Namla-Tofanma
14. Kimki
15. Biksi
16. Pauwasi
17. Koiarian
18. East Strickland
19. Dibiyaso
20. Bosavi
21. Fasu
22. East Kutubu
23. Turama-Kikori
24. Austronesian
25. Bilua
26. Touo
27. Lavukaleve
28. Savosavo
29. Walio
30. Sepik
31. Ndu
32. Morehead-Wasur
33. Pahoturi
34. Eastern Trans-Fly
35. Alor-Pantar
36. East Timor-Bunaq
37. West Bomberai
38. Marindic
39. Awin-Pa
40. Kamula
41. Bogaya
42. Duna
43. Amto-Musan
44. Left May
45. Greater Kwerba
46. Kapauri
47. Maybrat
48. Anem
49. Mpur
50. Yawa
51. Kolopom
52. Bulaka River
53. Kaure-Narau
54. Yale

Distant relationships




1 Shastan
2 Pomoan
3 Salinan
4 Chimariko
5 Yana
6 Palaihnihan
7 Cochimi-Yuman
8 Seri
9 Tequistlatecan
10 Tunica
11 Misumalpan
12 Chibchan
13 Wintuan
14 Maiduan
15 Mayan
16 Algic
17 Kiowa-Tanoan
18 Uto-Aztecan
19 Cuitlatec
20 Beothuk
21 Molala
22 Sahaptian
23 Totonacan
24 Mixe-Zoque
25 Tarascan
1 Cofan
2 Quechuan
3 Paez
4 Aymaran
5 Uru-Chipaya
6 Ticuna-Yuri
7 Matacoan
8 Guaicuruan
9 Payagua
10 Harakmbut
11 Katukinan
12 Movima
13 Waorani
14 Andoque
15 Arawan
16 Saliban
17 Jodi
18 Jivaroan
19 Yamana
20 Kakua-Nukak
21 Puinave
22 Kwaza
23 Aikana
24 Mura-Piraha
25 Zaparoan
26 Peba-Yagua
27 Panoan
28 Tacanan
29 Hibito-Cholon
30 Tucanoan
31 Fulnio
32 Huarpean