# The world tree of languages: How to infer it from data, and what it is good for

Gerhard Jäger

Tübingen University

Workshop *Evolutionary Theory in the Humanities*, Torun

*April 14, 2018*

Words Bones Genes Tools
Tracking Linguistic, Cultural, and Biological Trajectories of the Human Past

Eberhard Karls Universität Tübingen

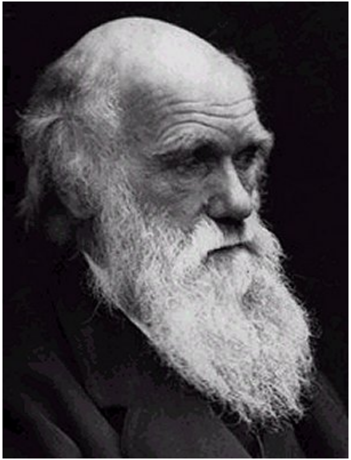
DFG

erc
European Research Council
Established by the European Commission

# Introduction

# Language change and evolution

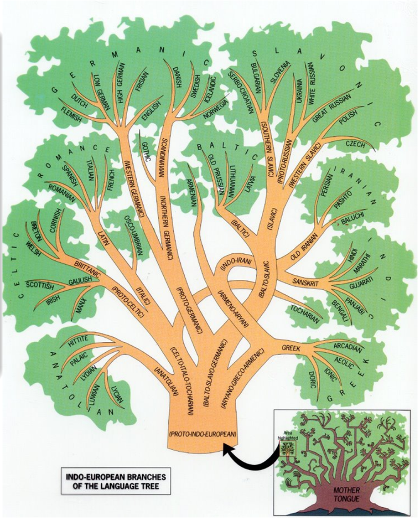
*"If we possessed a perfect pedigree of mankind, a genealogical arrangement of the races of man would afford the best classification of the various languages now spoken throughout the world; and if all extinct languages, and all intermediate and slowly changing dialects, had to be included, such an arrangement would, I think, be the only possible one. Yet it might be that some very ancient language had altered little, and had given rise to few new languages, whilst others (owing to the spreading and subsequent isolation and states of civilisation of the several races, descended from a common race) had altered much, and had given rise to many new languages and dialects. The various degrees of difference in the languages from the same stock, would have to be expressed by groups subordinate to groups; but the proper or even only possible arrangement would still be genealogical; and this would be strictly natural, as it would connect together all languages, extinct and modern, by the closest affinities, and would give the filiation and origin of each tongue."* (Darwin, The Origin of Species)



# Language phylogeny

## Comparative method

1. identifying *cognates*, i.e. obviously related morphemes in different languages, such as *new*/*nowy*, *two*/*dwa*, or *water*/*voda*
2. reconstruction of *common ancestor* and *sound laws* that explain the change from reconstructed to observed forms
3. applying this iteratively leads to phylogenetic language trees



# Language phylogeny

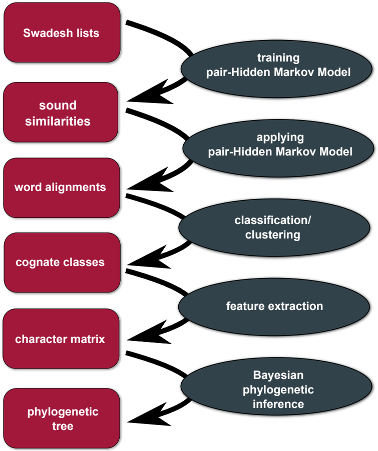
## Scope of the method

- reconstructed vocabulary shrinks with growing time depth
- maximal time horizon seems to be about 8,000 years
- grammatical morphemes and categories arguably more stable and less apt to borrowing
- problem here: limited number of features, cross-linguistic variation constrained by language universals, frequently convergent evolution
- comparative method is hard to apply in regions with high linguistic diversity and without written documents (Paleo-America, Papua)
- tree structure might be inappropriate if there is a significant effect of language contact (cf. Australia)

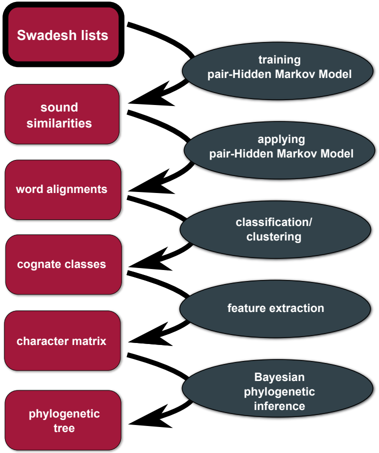
# Computational Methods

- both cognate detection and tree construction lend themselves to algorithmic implementation
- Advantages:
  - easy to scale up
  - comparability of results
  - affords statistical evaluation
- Disadvantages:
  - cognacy judgments require lots of linguistic insight and experience
  - tree construction should be subject to historical (including archeological) and geographical plausibility

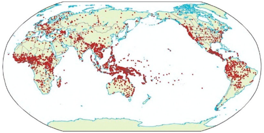
# From words to trees



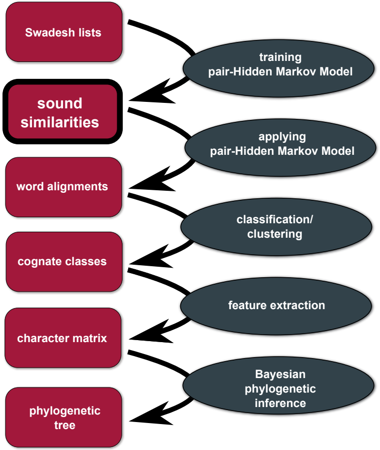
# From words to trees

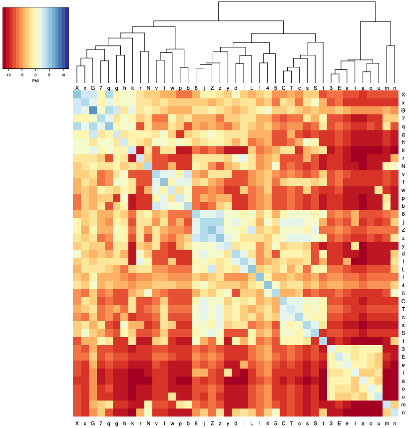



| *concept* | Latin | English |
|---|---|---|
| *I* | ego | Ei |
| *you* | tu | yu |
| *we* | nos | wi |
| *one* | unus | w3n |
| *two* | duo | tu |
| *person* | persona, homo | pers3n |
| *fish* | piskis | fiS |
| *dog* | kanis | dag |
| *louse* | pedikulus | laus |
| *tree* | arbor | tri |
| *leaf* | foly∼u* | lif |
| *skin* | kutis | skin |
| *blood* | saNgw∼is | bl3d |
| *bone* | os | bon |
| *horn* | kornu | horn |
| *ear* | auris | ir |
| *eye* | okulus | Ei |

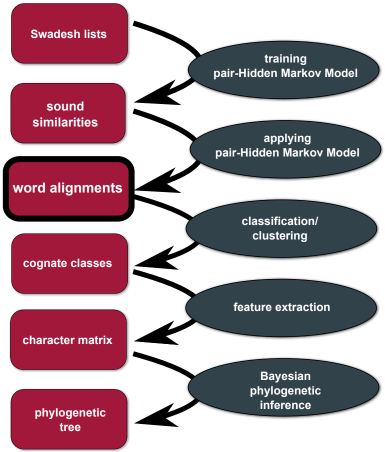


# From words to trees



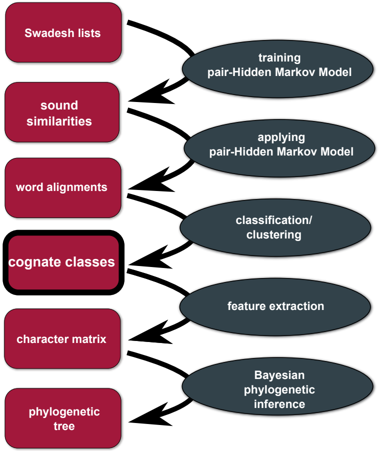



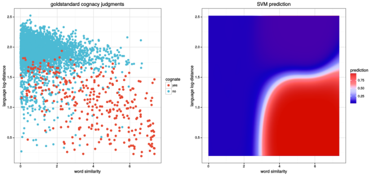
# From words to trees




| *Language* | *fish:z* | *tongue:1* | *smoke:1* |
|---|---|---|---|
| Abui-Atangmelang | -af-u | | |
| Abui-Fuimelang | -af-u | tal-i-fi-- | |
| Adang | aab-- | tal-E-b--- | awai--b-a-n-o-7o- |
| Blagar-Bakalang | -ab-- | --j-e-bur- | --ad--b-a-n-aNka- |
| Blagar-Bama | aab-- | teg-e-bur- | ------b-e-n-a-xa- |
| Blagar-Kulijahi | -ab-- | tej-e-bur- | ------b-e-n-aNka- |
| Blagar-Nule | aab-- | tej-e-bur- | --ad--b-e-n-aNka- |
| Blagar-Tuntuli | aab-- | tej-e-bur- | a-adgeb-a-n-a-q-- |
| Blagar-Warsalelang | -ab-- | tel-e-bur- | a-ad--b-a-n-a-x-- |
| Bunaq | | | ------b-o-t-o-h-- |
| Deing | haf-- | | ------buu-n------ |
| Hamap | 7ab-- | nar-ø-buN- | ------b-a-n-o-7-- |
| Kabola | hab-- | tal-e-b--- | awal--b-e-n-e-7o- |
| Kaera-Padangsul | -ab-- | talee-b--- | a-ad--b-e-naa-x-- |
| Kafoa | -afUi | tal-i-p--- | ------f-o-n-a---- |
| Kamang | -ap-i | nal---pu-- | ------p-u-n----a- |
| Kiraman | -Eb-- | nal-i-bar- | --ar--b-a-n-o-kan |
| Klon | -eb-i | gel-E-b--- | --ed-ab-o-n------ |
| Kui | -eb-- | tal-i-ber- | --ar--b-o-n-o-k-- |
| Kula | -ap-i | -il-I-p--- | ------p---n-ekka- |
| Nedebang | aaf-i | gel-e-fu-- | --ar-ab-u-n------ |
| Reta | aab-- | nal-e-bul- | a-ad--b-o-n-a---- |
| Sar-Adiabang | haf-- | --p-e-fal- | --ar--buu-n------ |
| Sar-Nule | haf-- | nal-e-faj- | |
| Sawila | -ap-i | gal-impuru | ------p-u-n-a-ka- |
| Teiwa-Madar | xaf-- | gel-i-vi-- | ------buu-n------ |
| Wersing | -ap-i | nej-e-bur- | --ad-ap-u-n-a-k-- |
| Wpantar | hap-- | nal-e-bu-- | ------b-unn-a---- |

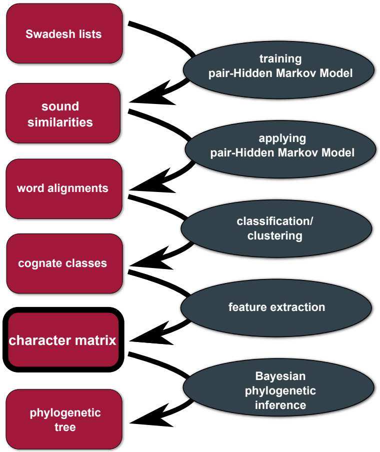
# From words to trees



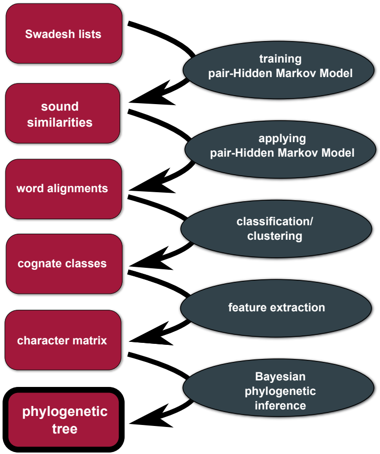

| | English | Spanish | Modern Greek | Standard German |
|---|---|---|---|---|
| *I* | Ei:A | yo:B | exo:C | iX:D |
| *you* | yu:A | ustet:B, tu:C | esi:D | du:E |
| *we* | wi:A | nosotros:B | emis:C | vir:A |
| *one* | w3n:A | uno:B | enas:C, ena:C | ains:D |
| *two* | tu:A | dos:B | 8y∼o:C, 8io:D | cvai:E |
| *person* | pers3n:A | persona:A | an8∼ropos:B | mEnS:C |
| *fish* | fiS:A | peskado:A, pes:A | psari:B | fiS:A |
| *dog* | dag:A | pero:B | sTili:C, sTilos:C | hunt:D |
| *come* | k3m:A | veni:B | erx∼o:C | kh∼om3n:A |
| *sun* | s3n:A | sol:B | ily∼os:C, iLos:C | zon3:A |
| *star* | star:A | estreya:A | asteri:A, astro:A | StErn:A |
| *water* | wat3r:A | agw∼a:B | nero:C | vas3r:A |
| *stone* | ston:A | piedra:B | petra:B | Stain:A |
| *fire* | fEir:A | fuego:B | foty∼a:C | foia:D |
| *path* | pE8:A | senda:B | 8romos:C | pf∼at:A, vek:D |
| *mountain* | maunt3n:A | sero:B, monta5a:A | vuno:C, oros:D | bErk:E |
| *full* | ful:A | yeno:B | yematos:C, pliris:D | fol:A |
| *new* | nu:A | nuevo:A | neos:A, Tenury∼os:B | noi:A |
| *name* | nem:A | nombre:A | onoma:A | nam3:A |

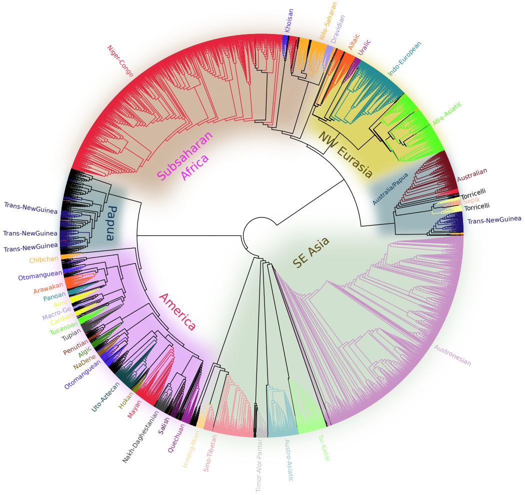
# From words to trees




```
TNG.ENGAN.MAIBI                 10000000000000000000000000000000000000000000
TNG.ENGAN.POLE                  00000000000000000000000000000000000010000000
TNG.ENGAN.SAU                   00000000000000000000000000000000000000000000
TNG.ENGAN.YARIBA                10000000000000000000000000000000000000000000
TNG.FASU.FASU                   00000000000000000000000000000000000010000000
TNG.FASU.NAMUMI                 00000000000000000000000000000000000000001000
TNG.FINISTERRE-HUON.AWARA       00000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.BORONG      00000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.BURUM       00000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.BURUM_MIND  00000000000000000000000000000000001010000000
TNG.FINISTERRE-HUON.DEDUA       00000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.HUBE        00000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.KATE        00000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.KOMBA       00000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.KOSORONG    00000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.MAPE        00000000000000000000000000000000001000000000
TNG.FINISTERRE-HUON.MAPE_2      00000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.MIGABAC     00000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.MINDIK      00000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.MOMOLILI    00000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.NABAK       00000000000000000000000000000000001000000000
TNG.FINISTERRE-HUON.NANKINA     00000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.NEK         00000000000000000000000000000000000001000000
TNG.FINISTERRE-HUON.NUKNA       00000000000000000000000000000000000001000000
TNG.FINISTERRE-HUON.ONO         00000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.SELEPET     00000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.TIMBE       00000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.TOBO        00000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.WANTOAT     00000000000000000000000000000000000110000000
TNG.FINISTERRE-HUON.YOPNO       00000000000000000000000000000000000001000000
TNG.GOILALAN.AFOA               00000000000000000000000000000000001100000000
TNG.GOILALAN.KUNIMAIPA          00000000000000000000000000000000000010000000
TNG.GOILALAN.MAFULU             00000000000000000000000000000000000010000000
```

# From words to trees



# From word lists to distances

# The Automated Similarity Judgment Program

- Started as project at MPI EVA in Leipzig around Søren Wichmann
- covers more than 7,000 languages and dialects
- basic vocabulary of 40 words for each language, in uniform phonetic transcription
- freely available

**used concepts:** *I, you, we, one, two, person, fish, dog, louse, tree, leaf, skin, blood, bone, horn, ear, eye, nose, tooth, tongue, knee, hand, breast, liver, drink, see, hear, die, come, sun, star, water, stone, fire, path, mountain, night, full, new, name*

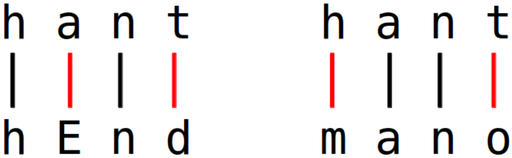
# Automated Similarity Judgment Project

| concept | Latin | English |
|---|---|---|
| *I* | ego | Ei |
| *you* | tu | yu |
| *we* | nos | wi |
| *one* | unus | w3n |
| *two* | duo | tu |
| *person* | persona, homo | pers3n |
| *fish* | piskis | fiS |
| *dog* | kanis | dag |
| *louse* | pedikulus | laus |
| *tree* | arbor | tri |
| *leaf* | foly∼u* | lif |
| *skin* | kutis | skin |
| *blood* | saNgw∼is | bl3d |
| *bone* | os | bon |
| *horn* | kornu | horn |
| *ear* | auris | ir |
| *eye* | okulus | Ei |

| concept | Latin | English |
|---|---|---|
| *nose* | nasus | nos |
| *tooth* | dens | tu8 |
| *tongue* | liNgw∼E | t3N |
| *knee* | genu | ni |
| *hand* | manus | hEnd |
| *breast* | pektus, mama | brest |
| *liver* | yekur | liv3r |
| *drink* | bibere | drink |
| *see* | widere | si |
| *hear* | audire | hir |
| *die* | mori | dEi |
| *come* | wenire | k3m |
| *sun* | sol | s3n |
| *star* | stela | star |
| *water* | akw∼a | wat3r |
| *stone* | lapis | ston |
| *fire* | iNnis | fEir |

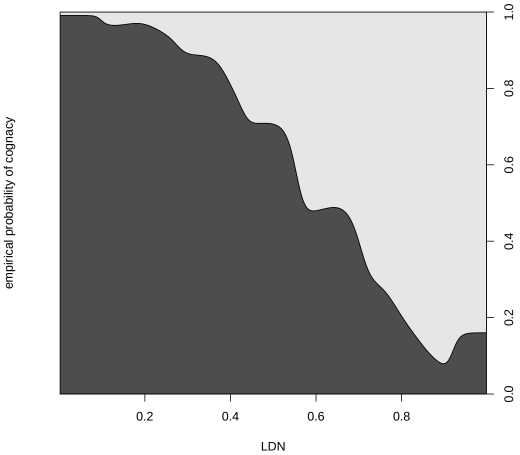
# Word distances

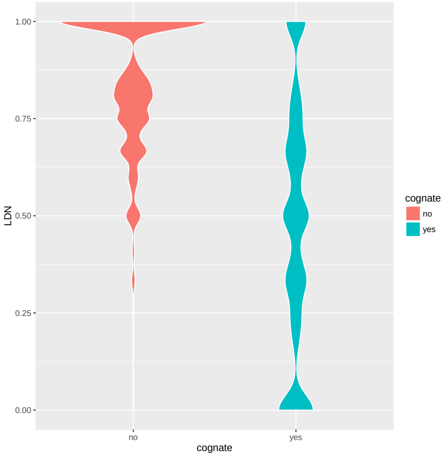
- based on string *alignment*
- baseline: Levenshtein alignment ⇒ count matches and mis-matches




- too crude as it totally ignores sound correspondences

# How well does normalized Levenshtein distance predict cognacy?



# Problems

- binary distinction: match vs. non-match
- frequently genuin sound correspondences in cognates are missed:

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **c** | v | a | i | | n | a | **z** | 3 | | - | - | - | **f** | i | S |
| - | - | **t** | u | | n | - | o | **s** | | **p** | i | s | k | i | s |

- corresponding sounds count as mismatches even if they are aligend correctly

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

- substantial amount of chance similarities

# 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...

# Substitution matrix for the ASJP data

1. identify large sample of pairs of closely related languages (using expert information or heuristics based on aggregated Levenshtein distance)

```
An.NORTHERN_PHILIPPINES.CENTRAL_BONTOC
An.MESO-PHILIPPINE.NORTHERN_SORSOGON

WF.WESTERN_FLY.IAMEGA
WF.WESTERN_FLY.GAMAEWE

Pan.PANOAN.KASHIBO_BAJO_AGUAYTIA
Pan.PANOAN.KASHIBO_SAN_ALEJANDRO

AA.EASTERN_CUSHITIC.KAMBAATA_2
AA.EASTERN_CUSHITIC.HADIYYA_2

ST.BAI.QILIQIAO_BAI_2
ST.BAI.YUNLONG_BAI

An.SULAWESI.MANDAR
An.OCEANIC.RAGA

An.SULAWESI.TANETE
An.SAMA-BAJAW.BOEPINANG_BAJAU

An.SOUTHERN_PHILIPPINES.KAGAYANEN
An.NORTHERN_PHILIPPINES.LIMOS_KALINGA

An.MESO-PHILIPPINE.CANIPAAN_PALAWAN
An.NORTHWEST_MALAYO-POLYNESIAN.LAHANAN

NC.BANTOID.LIFONGA
NC.BANTOID.BOMBOMA_2

IE.INDIC.WAD_PAGGA
IE.INDIC.TALAGANG_HINDKO

NC.BANTOID.LINGALA
NC.BANTOID.LIFONGA

An.CENTRAL_MALAYO-POLYNESIAN.BALILEDO
An.CENTRAL_MALAYO-POLYNESIAN.PALUE

AuA.MUNDA.HO
AuA.MUNDA.KORKU
```

# Substitution matrix for the ASJP data

2. pick a concept and a pair of related languages at random
   - languages: `Pen.MAIDUAN.MAIDU_KONKAU`, `Pen.MAIDUAN.NE_MAIDU`
   - concept: *one*
3. find corresponding words from the two languages:
   - `nisam`, `niSem`
4. do Levenshtein alignment

| | | | | |
|---|---|---|---|---|
| n | i | s | a | m |
| n | i | S | e | m |

5. for each sound pair, count number of correspondences
   - nn: 1; ii: 1; sS; 1; ae: 1; mm: 1

# Finding the best alignment

- 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 | | | | |

# Finding the best alignment

- 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 | $-8.9$ | | | | |

# Finding the best alignment

- 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 | $-8.9$ | | | | |

# Finding the best alignment

- 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 | | | | |

# Finding the best alignment

- 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 | | | | |

# Finding the best alignment

- 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 | $-8.9$ | | | | |

# Finding the best alignment

- 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 | $-8.9$ | | | | |

# Finding the best alignment

- 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 | $-8.9$ | | | | |

# Finding the best alignment

- 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 | | | | |

# Finding the best alignment

- 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 | $-8.9$ | | | | |

# Finding the best alignment

- 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 | | | | |

# Finding the best alignment

- 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 | | | | |

# Finding the best alignment

- 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 | | | | |

# Finding the best alignment

- 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 | | | | |

# Finding the best alignment

- 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 | | | | |

# Finding the best alignment

- 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 | | | | |

# Finding the best alignment

- 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 | | | | |

# Finding the best alignment

- 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 | $-8.9$ | | | | |

# Finding the best alignment

- 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 | $-8.9$ | | | | |

# Finding the best alignment

- 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 | | | | |

# Finding the best alignment

- 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 | $-8.9$ | | | | |

# Finding the best alignment

- 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 | $-8.9$ | $-2.97$ | | | |

# Finding the best alignment

- 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 | | |

# Finding the best alignment

- 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 | |

# Finding the best alignment

- 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 |

# Finding the best alignment

- 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

# Finding the best alignment

- 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

# Finding the best alignment

- 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

# Finding the best alignment

- 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

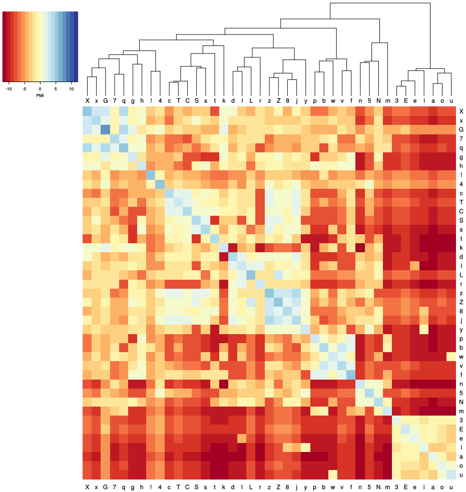
# Finding the best alignment

- Dynamic Programming

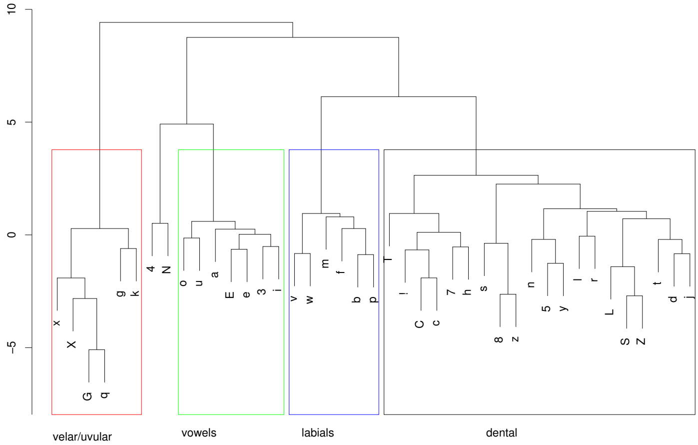
| | – | m | E | n | S |
|---|---|---|---|---|---|
| – | 0 | −2.5 | −4.1 | −5.7 | −7.3 |
| m | −2.5 | 4.13 | 1.53 | 0.03 | −1.47 |
| e | −4.1 | 1.53 | 5.65 | 3.05 | 1.55 |
| n | −5.7 | 0.03 | 3.05 | 9.2 | 6.6 |
| E | −7.3 | −1.47 | 4.75 | 6.6 | 7.62 |
| s | −8.9 | −2.97 | 2.15 | 5.1 | 8.84 |

- 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

# Evaluation

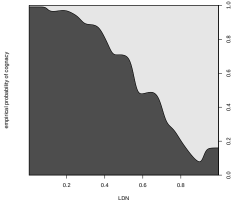

-10
-5
0
5
10
PMI
X x G 7 q g h ! 4 c T C S s t k d l L r z Z 8 j y p b w v f n 5 N m 3 E e i a o u

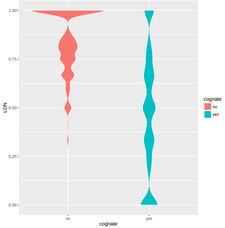
# Evaluation

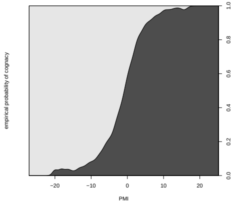

10
5
0
−5
x X G q g k
4 N
o u a E e 3 i
v w m f b p
T ! C c 7 h s 8 z n 5 y l r L S Z t d j
velar/uvular
vowels
labials
dental

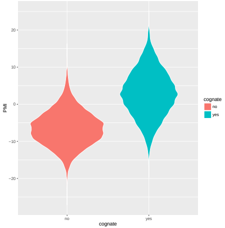
# How well does PMI similarity predict cognacy?

expert cognacy judgments used as gold standard



# Calibrated PMI similarity

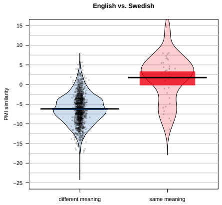
## English / Swedish

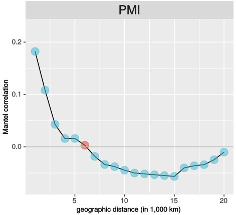
| | Ei | yu | wi | w3n | tu | fiS | ... |
|---|---|---|---|---|---|---|---|
| **yog** | **−7.77** | 0.75 | −7.68 | −7.90 | −8.57 | −10.50 | |
| **du** | −7.62 | **0.33** | −5.71 | −7.41 | 2.66 | −8.57 | |
| **vi** | −2.72 | −2.83 | **4.04** | −1.34 | −6.45 | 0.70 | |
| **et** | −5.47 | −7.87 | −5.47 | **−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

# Calibrated PMI similarity

- let $s$ be the PMI-similarity between the English and Swedish word for concept $c$
- **calibrated string similarity**: $-\log$(probability that random word pairs are more similar than $s$)
- **language similarity:** average word similarity for all concepts



# Cognate clustering

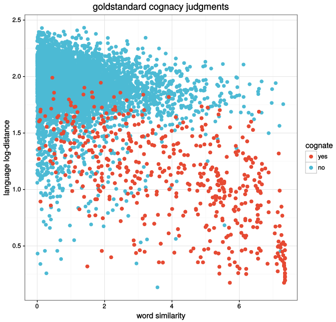
# Cognate clustering

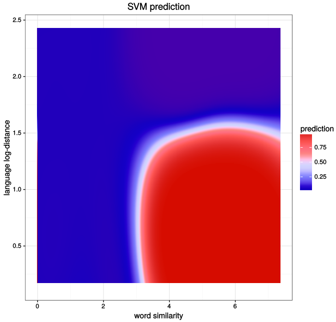
- clustering of ASJP strings into *automatically inferred cognate classes* (Jäger and Sofroniev, 2016; Jäger et al., 2017) (take "cognate" with a grain of salt)
- supervised learning, based on expert cognacy judgments as goldstandard
- sources (only the 40 ASJP concepts were used)

| Dataset | Source | Words | Concepts | Languages | Families | Cognate classes |
|---|---|---|---|---|---|---|
| ABVD | Greenhill et al. (2008) | 2,306 | 34 | 100 | Austronesian | 409 |
| Afrasian | Militarev (2000) | 770 | 39 | 21 | Afro-Asiatic | 351 |
| Chinese | Běijng Dàxué (1964) | 422 | 20 | 18 | Sino-Tibetan | 126 |
| Huon | McElhanon (1967) | 441 | 32 | 14 | Trans-New Guinea | 183 |
| IELex | Dunn (2012) | 2,089 | 40 | 52 | Indo-European | 318 |
| Japanese | Hattori (1973) | 387 | 39 | 10 | Japonic | 74 |
| Kadai | Peiros (1998) | 399 | 40 | 12 | Tai-Kadai | 102 |
| Kamasau | Sanders and Sanders (1980) | 270 | 36 | 8 | Torricelli | 59 |
| Mayan | Brown et al. (2008) | 1,113 | 40 | 30 | Mayan | 241 |
| Miao-Yao | Peiros (1998) | 206 | 36 | 6 | Hmong-Mien | 69 |
| Mixe-Zoque | Cysouw et al. (2006) | 355 | 39 | 10 | Mixe-Zoque | 79 |
| Mon-Khmer | Peiros (1998) | 579 | 40 | 16 | Austroasiatic | 232 |
| ObUgrian | Zhivlov (2011) | 769 | 39 | 21 | Uralic | 68 |
| total | | 10,106 | 40 | 318 | 13 | 2,311 |

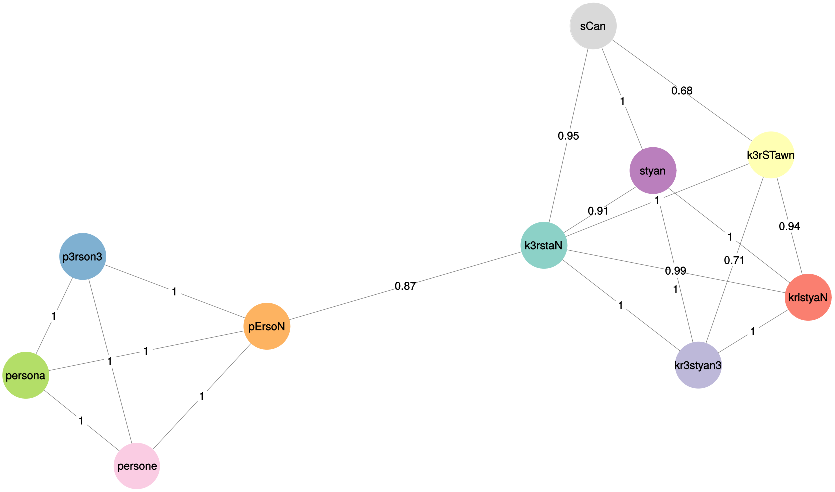
# Cognate clustering

- calibrated word similarity and language similarity were used as predictors to train a *Support Vector Machine* $\rightarrow$ probability of being cognate for each pair of synonymous ASJP entries
- *Label Propagation* (Raghavan et al., 2007) for clustering
- $0.84$ B-cubed F-score with cross-validation on goldstandard data

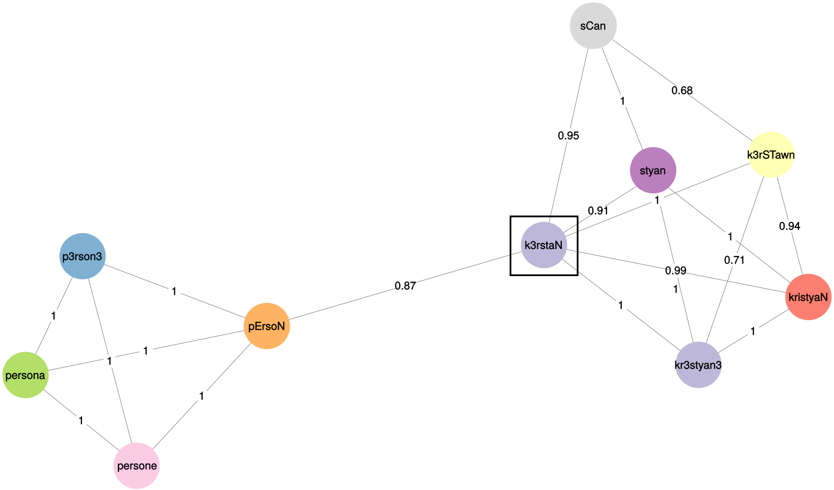


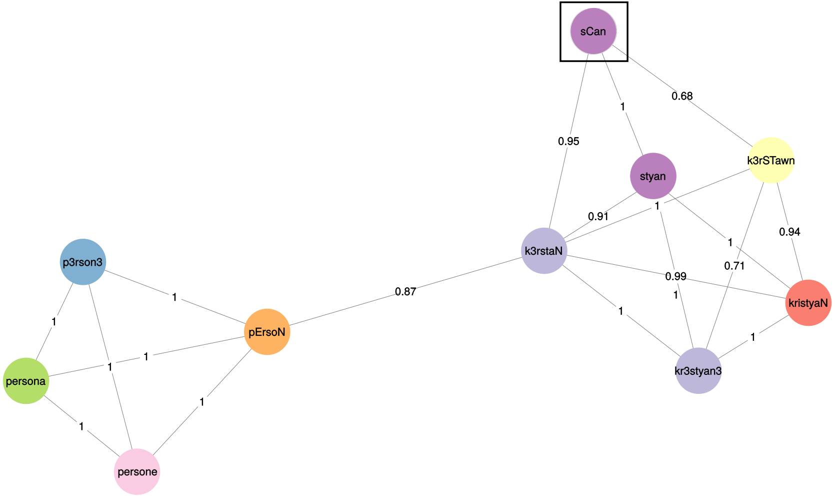
# Clustering via Label Propagation



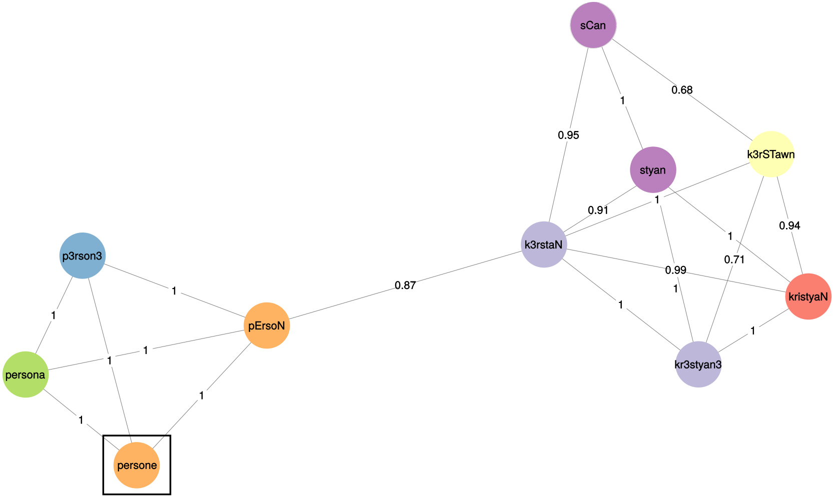
# Clustering via Label Propagation



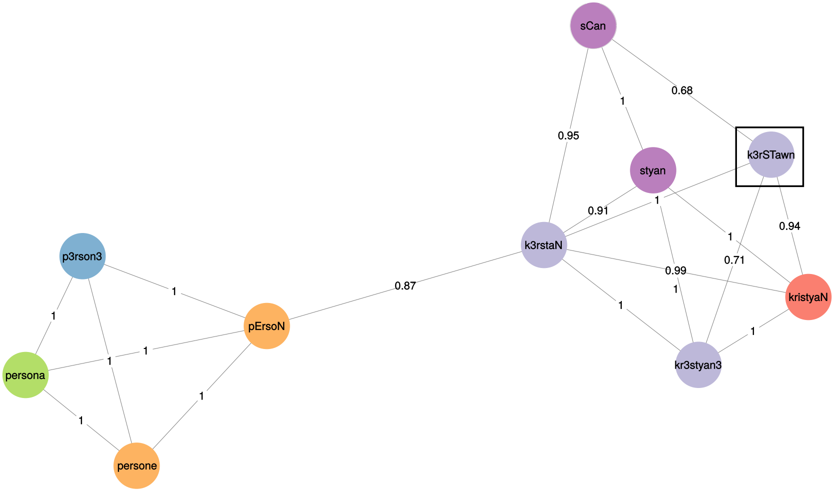
# Clustering via Label Propagation



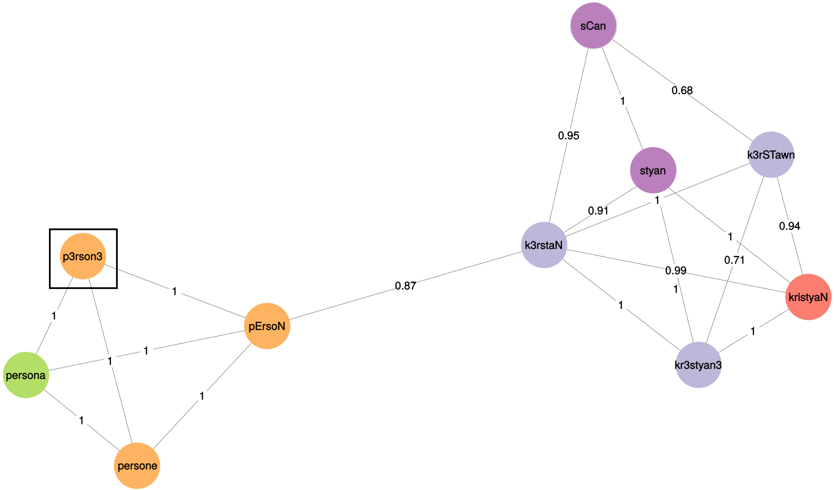
# Clustering via Label Propagation



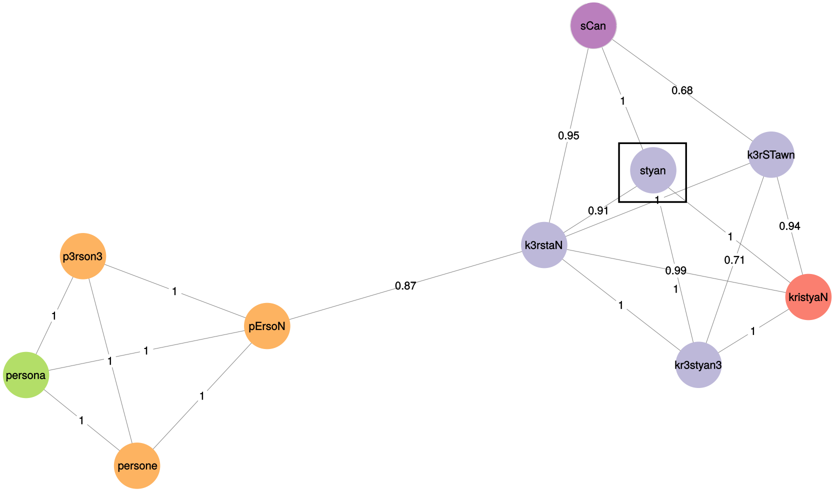
# Clustering via Label Propagation



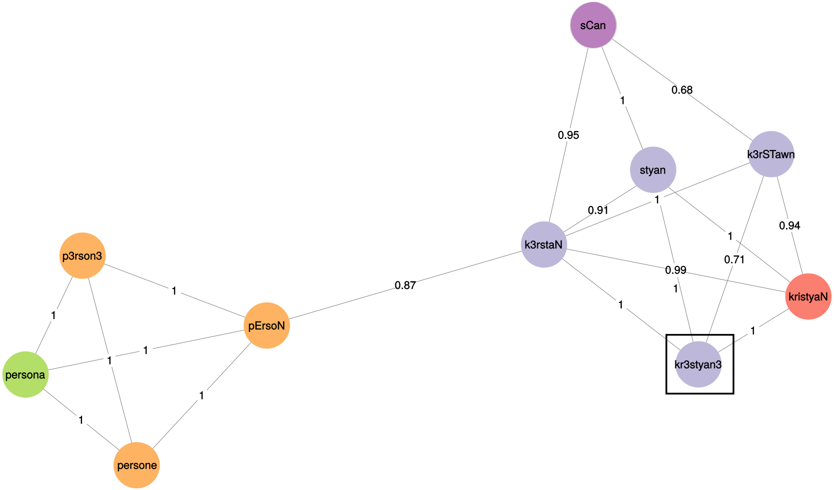
# Clustering via Label Propagation



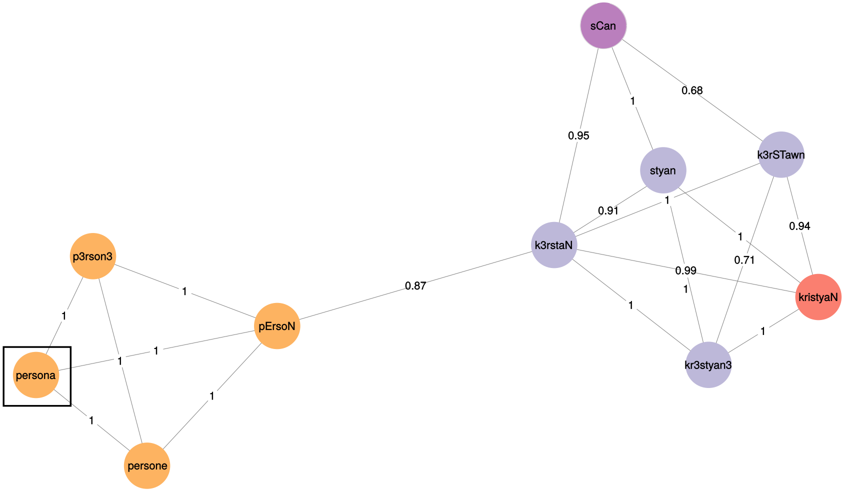
# Clustering via Label Propagation



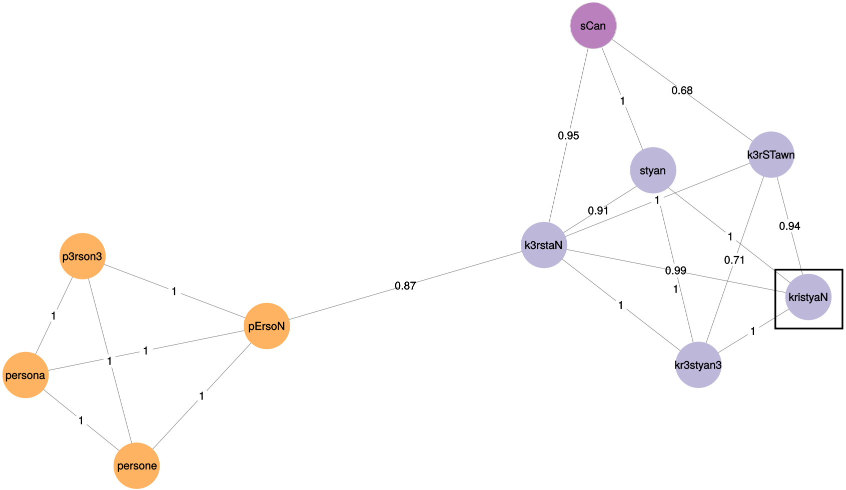
# Clustering via Label Propagation



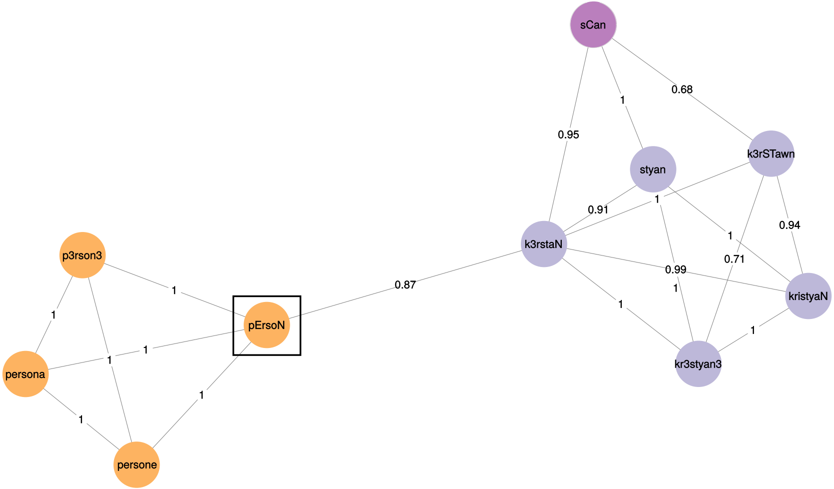
# Clustering via Label Propagation



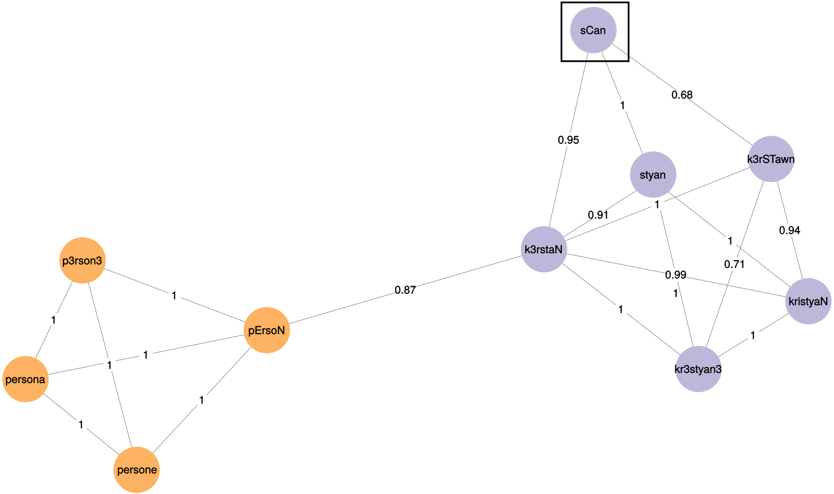
# Clustering via Label Propagation



# Clustering via Label Propagation



# Clustering via Label Propagation



# Cognate clustering

| doculect | word | class label |
|---|---|---|
| ALBANIAN | vet3 | 0 |
| ALBANIAN_TOSK | vEt3 | 0 |
| ARAGONESE | ombre | 1 |
| ITALIAN_GROSSETO_TUSCAN | omo | 2 |
| ROMANIAN_MEGLENO | wom | 2 |
| VLACH | omu | 2 |
| ASTURIAN | persona | 3 |
| BALEAR_CATALAN | p3rson3 | 3 |
| CATALAN | p3rson3 | 3 |
| FRIULIAN | pErsoN | 3 |
| ITALIAN | persona | 3 |
| SPANISH | persona | 3 |
| VALENCIAN | persone | 3 |
| CORSICAN | nimu | 4 |
| DALMATIAN | om | 5 |
| EMILIANO_CARPIGIANO | om | 5 |
| ROMANIAN_2 | om | 5 |
| TURIA_AROMANIAN | om | 5 |
| EMILIANO_FERRARESE | styan | 6 |
| LIGURIAN_STELLA | kristyaN | 6 |
| NEAPOLITAN_CALABRESE | kr3styan3 | 6 |
| ROMAGNOL_RAVENNATE | sCan | 6 |
| ROMANSH_GRISHUN | k3rSTawn | 6 |
| ROMANSH_SURMIRAN | k3rstaN | 6 |
| GALICIAN | ome | 7 |
| GASCON | omi | 7 |
| PIEMONTESE_VERCELLESE | omaN | 8 |
| ROMANSH_VALLADER | uman | 8 |
| ALBANIAN_GHEG | 5eri | 9 |
| SARDINIAN_CAMPIDANESE | omini | 9 |
| SARDINIAN_LOGUDARESE | omine | 9 |

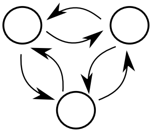
# Cognate clustering

| concept | doculect | glot_fam | transcription |
| --- | --- | --- | --- |
| eye | DORASQUE | Chibchan | oko |
| eye | NORTHERN_LOW_SAXON | Indo-European | ok |
| eye | NORTH_FRISIAN_AMRUM | Indo-European | uk |
| eye | STELLINGWERFS | Indo-European | ok |
| eye | ASSAMESE | Indo-European | soku |
| eye | CHAKMA_UnnamedInSource | Indo-European | sog |
| eye | DALMATIAN | Indo-European | vaklo |
| eye | FRIULIAN | Indo-European | voli |
| eye | ITALIAN | Indo-European | okkyo |
| eye | ITALIAN_GROSSETO_TUSCAN | Indo-European | okyo |
| eye | JUDEO_ESPAGNOL | Indo-European | oxo |
| eye | LATIN | Indo-European | okulus |
| eye | NEAPOLITAN_CALABRESE | Indo-European | woky3 |
| eye | ROMANIAN_2 | Indo-European | oky |
| eye | ROMANIAN_MEGLENO | Indo-European | wokLu |
| eye | SARDINIAN | Indo-European | ogu |
| eye | SARDINIAN_CAMPIDANESE | Indo-European | oxu |
| eye | SARDINIAN_LOGUDARESE | Indo-European | okru |
| eye | SICILIAN_UnnamedInSource | Indo-European | okiu |
| eye | SPANISH | Indo-European | oho |
| eye | TURIA_AROMANIAN | Indo-European | okLu |
| eye | VLACH | Indo-European | okklu |
| eye | BELARUSIAN | Indo-European | voka |
| eye | BOSNIAN | Indo-European | oko |
| eye | BULGARIAN | Indo-European | oko |
| eye | CROATIAN | Indo-European | oko |
| eye | CZECH | Indo-European | oko |
| eye | KASHUBIAN | Indo-European | wokwo |
| eye | LOWER_SORBIAN | Indo-European | voko |
| eye | LOWER_SORBIAN_2 | Indo-European | woko |
| eye | MACEDONIAN | Indo-European | oko |
| eye | OLD_CHURCH_SLAVONIC | Indo-European | oko |
| eye | POLISH | Indo-European | oko |
| eye | SERBOCROATIAN | Indo-European | oko |
| eye | SLOVAK | Indo-European | oko |
| eye | SLOVENIAN | Indo-European | oko |
| eye | UKRAINIAN | Indo-European | oko |
| eye | UPPER_SORBIAN | Indo-European | voCko |
| eye | UPPER_SORBIAN | Indo-European | voko |
| eye | BAINOUK_GUNYAAMOLO | Atlantic-Congo | g3li |
| eye | USINO | Nuclear_Trans_New_Guinea | ogo |

# Phylogenetic inference

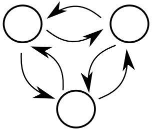
# Modeling language change

**Markov process**

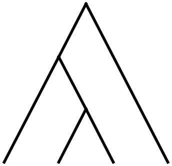


# Modeling language change

**Markov process**

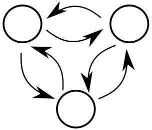


**Phylogeny**

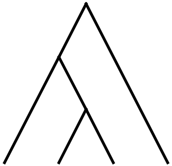


# Modeling language change

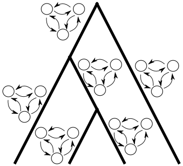
**Markov process**



**Phylogeny**

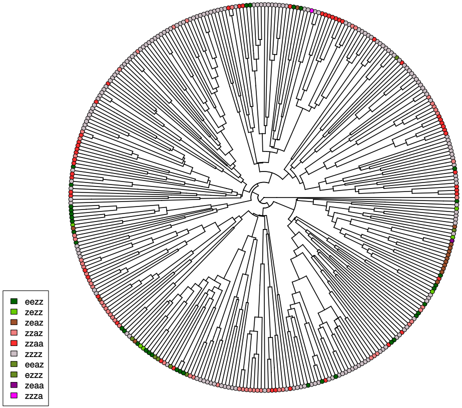


**Branching process**



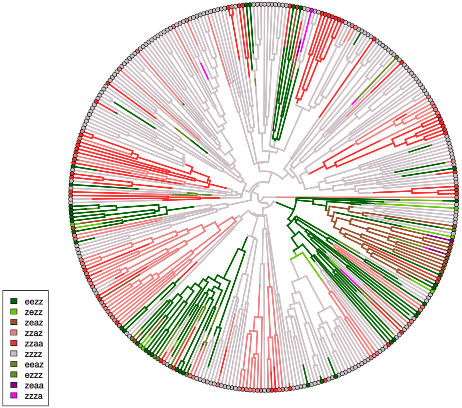
# Estimating rates of change

- if phylogeny and states of extant languages are known...



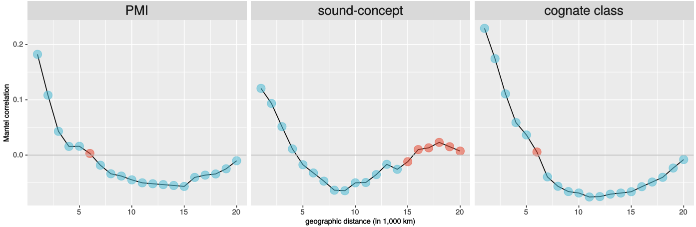
# Estimating rates of change

- if phylogeny and states of extant languages are known...
- ... transition rates and ancestral states can be estimated based on Markov model



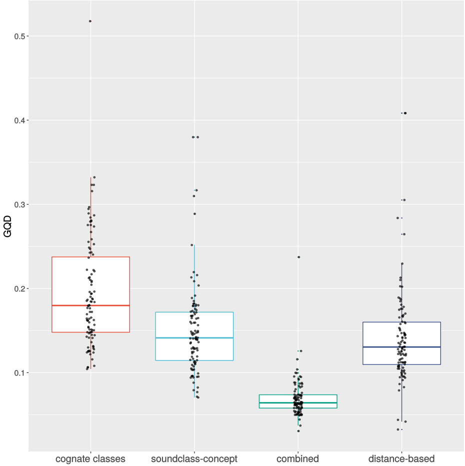
# ASJP word lists $\rightarrow$ character matrix

1. **Automatically inferred cognate classes**
   - each cluster $cc$ defines one character
   - doculect $l$ has value 1 if its word list contains an element of $cc$, undefined if the slot of the corresponding concept is undefined, and 0 else
2. **Soundclass-concept characters**
   - each combination $(c, s)$ of an ASJP concept $c$ and an ASJP sound class $s$ is a character
   - doculect $l$ has value 1 if one of its entries for $c$ contains $s$, 0 if not, and undefined if there is no entry for $c$

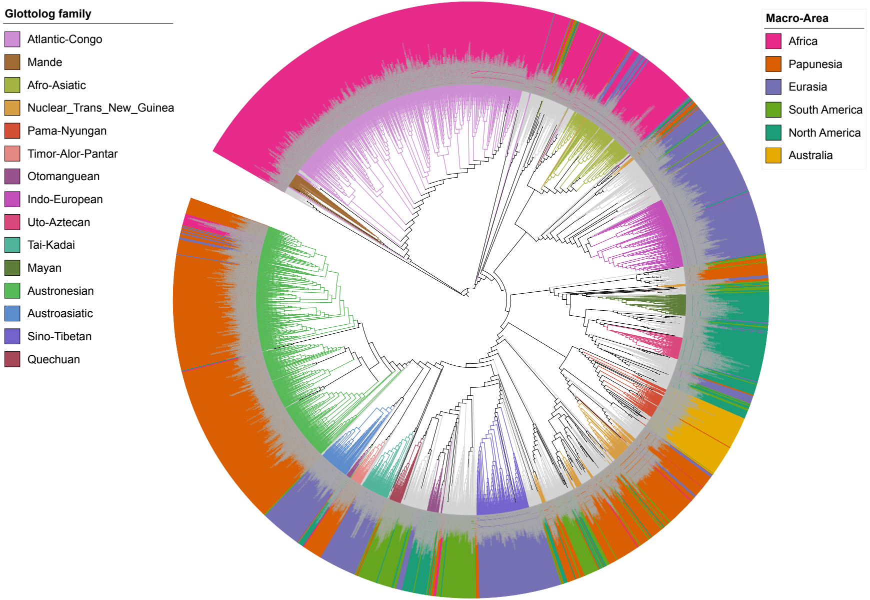


# Character matrix → trees

- validation
  - correlation with geographic distance
  - phylogenetic inference (Maximum Likelihood) + comparison to Glottolog expert tree on 100 random sample of ASJP doculects, containing between 20 and 400 doculects
  - using Stamatakis' **RAxML** (which is great)
- partitioned character-based inference seems to work best

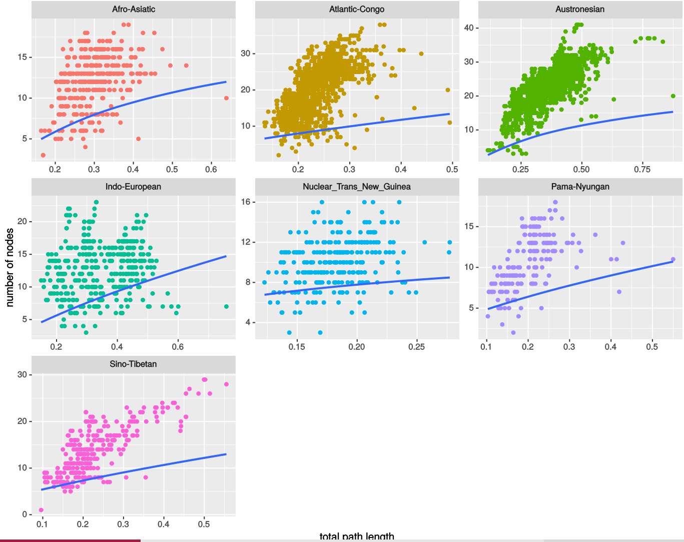


# The world tree

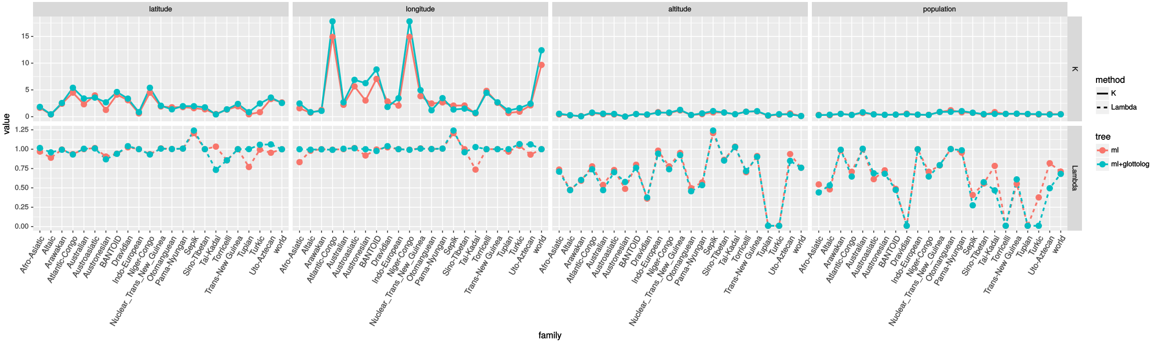


# Applications

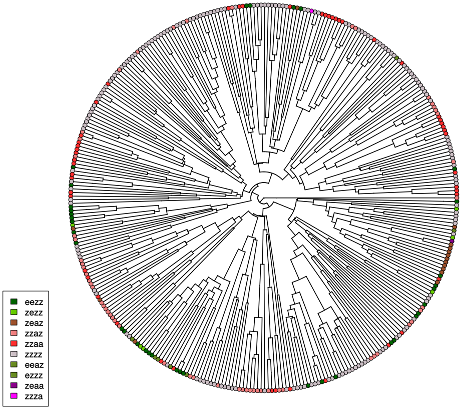
# Punctuated language evolution



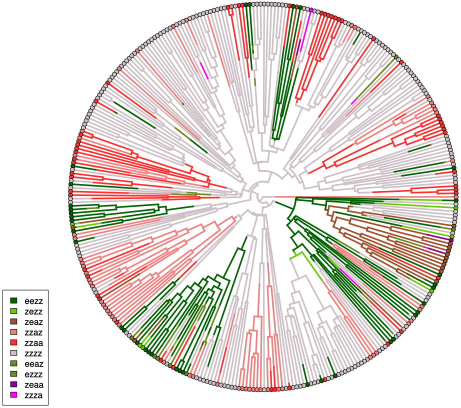
# Phylogenetic signal

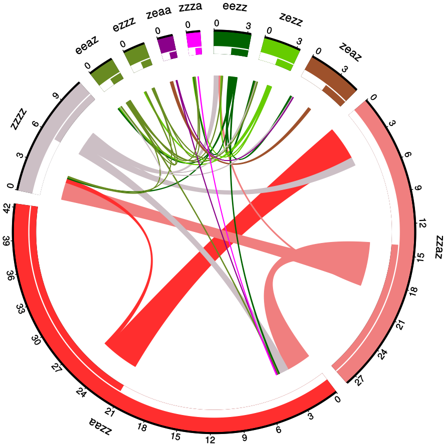


# Estimating Markov processes



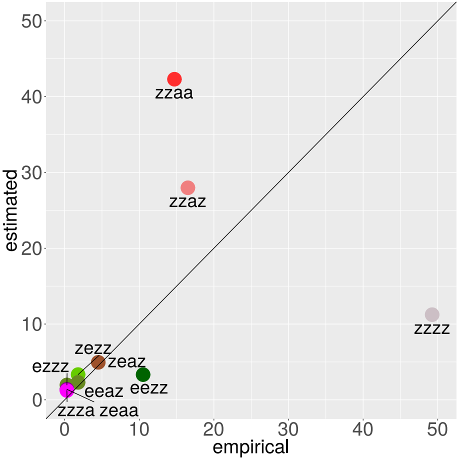
# Estimating Markov processes



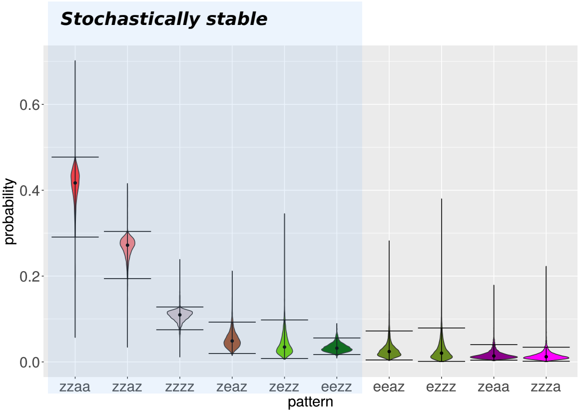



# Equilibrium probabilities

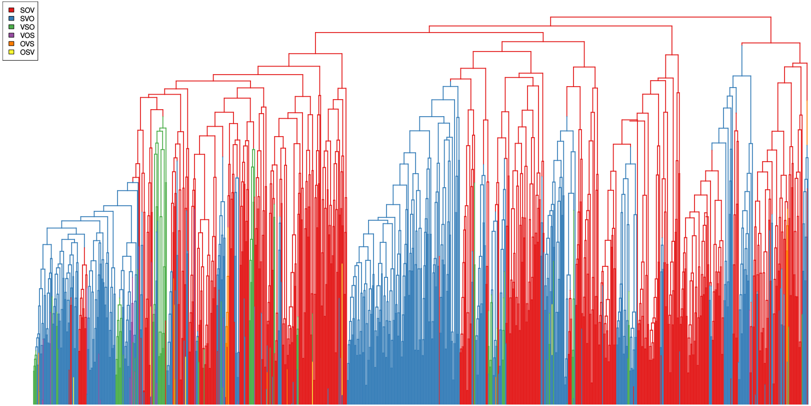
## Empirical vs. estimated percentages

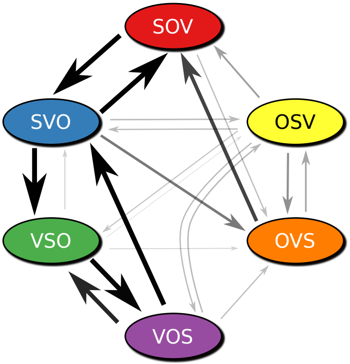



## Posterior distribution

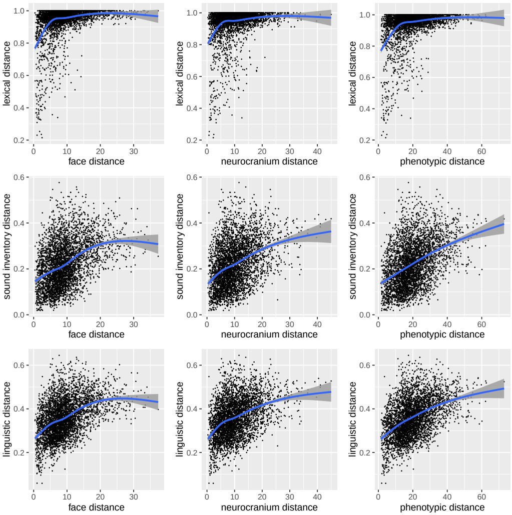


# Clines of typological change



# Correlation between linguistic and phenotypic distances



# Conclusion

- two areas where this might be useful:
  - prehistory/anthropology: identifying deep patterns of common ancestry and contact
  - historical linguistics and typology: statistical model for phylogenetically controlled inference
- data and code are available on arXiv/OSF: Jäger (2018)

Cecil H. Brown, Eric W. Holman, Søren Wichmann, and Viveka Velupillai. Automated classification of the world's languages: A description of the method and preliminary results. *STUF — Language Typology and Universals*, 4:285–308, 2008.

Běijng Dàxué. *Hànyǔ fngyán cíhuì* [Chinese dialect vocabularies]. Wénzì Gǎigé, 1964.

Michael Cysouw, Søren Wichmann, and David Kamholz. A critique of the separation base method for genealogical subgrouping. *Journal of Quantitative Linguistics*, 13(2-3):225–264, 2006.

Michael Dunn. Indo-European lexical cognacy database (IELex). URL: http://ielex.mpi.nl/, 2012.

Simon J. Greenhill, Robert Blust, and Russell D. Gray. The Austronesian Basic Vocabulary Database: From bioinformatics to lexomics. *Evolutionary Bioinformatics*, 4:271–283, 2008.

Shirō Hattori. Japanese dialects. In Henry M. Hoenigswald and Robert H. Langacre, editors, *Diachronic, areal and typological linguistics*, pages 368–400. Mouton, The Hague and Paris, 1973.

Gerhard Jäger. Global-scale phylogenetic linguistic inference from lexical resources. arXiv:1802.06079, 2018.

Gerhard Jäger and Pavel Sofroniev. Automatic cognate classification with a Support Vector Machine. In Stefanie Dipper, Friedrich Neubarth, and Heike Zinsmeister, editors, *Proceedings of the 13th Conference on Natural Language Processing*, volume 16 of *Bochumer Linguistische Arbeitsberichte*, pages 128–134. Ruhr Universität Bochum, 2016.

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

Kenneth A. McElhanon. Preliminary observations on Huon Peninsula languages. *Oceanic Linguistics*, 6(1):1–45, 1967. ISSN 00298115, 15279421. URL http://www.jstor.org/stable/3622923.

A IU Militarev. *Towards the chronology of Afrasian (Afroasiatic) and its daughter families*. McDonald Institute for Archaelogical Research, Cambridge, 2000.

Ilia Peiros. Comparative linguistics in Southeast Asia. *Pacific Linguistics*, 142, 1998.

Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3):036106, 2007.

Joy Sanders and Arden G Sanders. Dialect survey of the Kamasau language. *Pacific Linguistics. Series A. Occasional Papers*, 56:137, 1980.

Alexandros Stamatakis. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, 30(9):1312–1313, 2014.

Søren Wichmann, Eric W. Holman, and Cecil H. Brown. The ASJP database (version 17). http://asjp.clld.org/, 2016.

Mikhail Zhivlov. Annotated Swadesh wordlists for the Ob-Ugrian group. In George S. Starostin, editor, *The Global Lexicostatistical Database*. RGGU, Moscow, 2011. URL: http://starling.rinet.ru.