Phylogenetic trees II
Estimating distances, estimating trees from distances

Gerhard Jäger

ESSLLI 2016

Background

- ideally, we could infer the historical time since the latest common ancestor for any pair of languages
- not possible — at least not in a purely data-driven way
- best we can hope for: estimate *amount of linguistics change* since latest common ancestor
- following the lead of bioinformatics, estimation is based on *continuous time Markov process* model
- basic idea:
  - time is continuous
  - language change involves mutations of discrete characters
  - mutations can occur at any point in time
  - mutations in different branches are stochastically independent

Markov processes

Discrete time Markov chains

Ewens and Grant (2005), 4.5–4.9, 11

Definition

A *discrete time Markov chain* over a countable state space S is a function from $\mathbb{N}$ into random variables X over S with the *Markov property*

$$\mathbb{P}(X_{n+1} = x | X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) = \mathbb{P}(X_{n+1} = x | X_n = x_n)$$

which is *stationary*:

$$\forall m, n : \mathbb{P}(X_{n+1} = x_i | X_n = x_j) = \mathbb{P}(X_{m+1} = x_i | X_m = x_j)$$

Discrete time Markov chains

A dt Markov chain with finite state space is characterized by

- its *initial distribution* X_0, and
- its *transition Matrix* P, where

$$p_{ij} \quad = \quad \mathbb{P}(X_{n+1} = x_j | X_n = x_i)$$

P is a *stochastic matrix*, i.e. $\forall i \sum_j p_{i,j} = 1$.

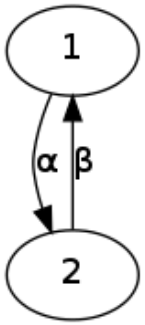
Definition

"Markov(λ, P)" is the dt Markov chain with initial distribution λ and transition matrix P.

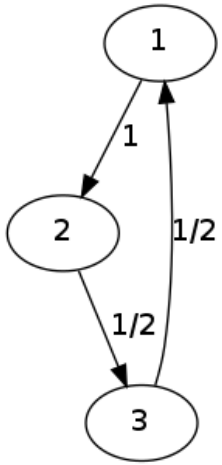
Discrete time Markov chains

Transition matrices over a finite state space can conveniently be represented as weighted graphs.

$$P = \begin{pmatrix} 1-\alpha, \alpha \\ \beta, 1-\beta \end{pmatrix}$$



$$P = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1/2 & 1/2 \\ 1/2 & 0 & 1/2 \end{pmatrix}$$



Discrete time Markov chains

- We say $i \rightarrow j$ if there is a path (with positive probabilities in each step) from x_i to x_j.
- The symmetric closure of this relation, $i \leftrightarrow j$, is an equivalence relation. It partitions a Markov chain into *communicating classes*.
- A Markov chain is *irreducible* iff it consists of a single communicating class.
- A state x_i is *recurrent* iff

$$\forall n \exists m : \mathbb{P}(X_{n+m} = x_i) > 0$$

- A state is *transient* iff it is not recurrent.

Discrete time Markov chains

- For each communicating class C: Either all of its states are transient or all of its states are recurrent.

Discrete time Markov chains

By convention, we assume that λ is a row vector. The distribution at time n is given by

$$\mathbb{P}(X_t = x_i) = (\lambda P^n)_i$$

Discrete time Markov chains

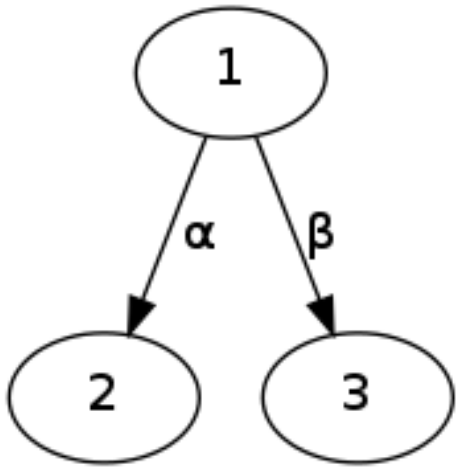
For each stochastic matrix P there is at least one distribution π with

$$\pi P = P$$

(π is a left eigenvector for P.) π is called an **invariant distribution.**

π need not be unique:

$$P = \begin{pmatrix} 1-\alpha-\beta & \alpha & \beta \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



$\pi = (0, \gamma, \delta)$ is a left eigenvector for P for each $\gamma, \delta \in [0, 1]$.

Discrete time Markov chains

If an irreducible Markov chain converges, then it converges to an invariant distribution:

If $\lim_{n\to\infty} P^n = A$, then

- there is a distribution π with $A_i = \pi$ for all i, and
- π is invariant.

π is called the **equilibrium distribution**. Not every Markov chain has an equilibrium:

$$P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Discrete time Markov chains

Definition

- The **period** k of state x_i is defined as

$$k = \text{gcd}\{n : \mathbb{P}(X_n = i | X_0 = i) > 0\}$$

- A state is **aperiodic** iff its period $= 1$.
- A Markov chain is **aperiodic** iff each of its states is aperiodic.

Theorem

If a finite Markov chain is irreducible and aperiodic, then

- *it has exactly one invariant distribution, π, and*
- *π is its equilibrium.*

Discrete time Markov chains

Theorem

If a finite Markov chain is irreducible and aperiodic, with equilibrium distribution π, then

$$\lim_{n \to \infty} \frac{|\{k < n | X_k = x_i\}|}{n} = \pi_i$$

Intuitively: the relative frequency of times spent in a state converges to the equilibrium probability of that state.

Continuous time Markov chains

- If P is the transition matrix of a discrete time Markov process, then so is P^n.
- In other words, P^n give the transition probabilities for a time interval n.
- Generalization:
  - $P(t)$ is transition matrix as a function of time t.
  - For discrete time: $P(t) = P(1)^t$.
  - How can this be generalized to continuous time?

Matrix exponentials

Definition

$$e^A \doteq \sum_{k=0}^{\infty} \frac{A^k}{k!}$$

Some properties:

- $e^0 = I$
- If $AB = BA$, then $e^{A+B} = e^A e^B$
- $e^{nA} = (e^A)^n$
- If Y is invertible, then $e^{YAY^{-1}} = Ye^AY^{-1}$
- $e^{\mathsf{diag}(x_1,\ldots,x_n)} = \mathsf{diag}(e^{x_1}, \ldots, e^{x_n})$

Continuous time Markov chains

Definition (Q-matrix)

A square matrix Q is a **Q-matrix** or **rate matrix** iff

- $q_{ii} \leq 0$ for all i,
- $q_{ij} \geq 0$ iff $i \neq j$, and
- $\sum_j q_{ij} = 0$ for all i.

Theorem

If P is a stochastic matrix, then there is exactly one Q-matrix Q with

$$e^Q = P.$$

Continuous time Markov chains

Definition

Let Q be a Q-matrix and λ the initial probability distribution. Then

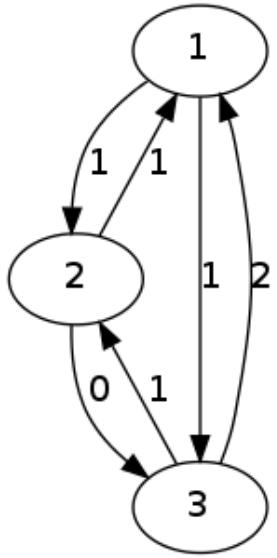
$$X(t) \doteq \lambda e^{tQ}$$

is a **continuous time Markov chain**.

Continuous time Markov chains

Q-matrices can be represented as graphs in the straightforward way (with loops being omitted).

$$Q = \begin{pmatrix} -2 & 1 & 1 \\ 1 & -1 & 0 \\ 2 & 1 & -3 \end{pmatrix}$$



Description in terms of jump chain/holding times

Let Q be a Q-matrix. The corresponding **jump matrix** Π is defined as

$$\pi_{ij} = \begin{cases} -q_{ij}/q_{ii} & \text{if } j \neq i \text{ and } q_{ii} \neq 0 \\ 0 & \text{if } j \neq i \text{ and } q_{ii} = 0 \end{cases}$$

$$\pi_{ii} = \begin{cases} 0 & \text{if } q_{ii} \neq 0 \\ 1 & \text{if } q_{ii} = 0 \end{cases}$$

$$Q = \begin{pmatrix} -2 & 1 & 1 \\ 1 & -1 & 0 \\ 2 & 1 & -3 \end{pmatrix} \quad \Pi = \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1 & 0 & 0 \\ 2/3 & 1/3 & 0 \end{pmatrix}$$

Description in terms of jump chain/holding times

Let Q be a Q-matrix and Π the corresponding jump matrix. The Markov process described by $\langle \lambda, Q \rangle$ can be conceived as:

1. Choose an initial state according to distribution λ.
2. If in state i, wait a time t that is exponentially distributed with parameter $-q_{ii}$.
3. Then jump into a new state j chosen according to the distribution $\Pi_{i.}$.
4. Goto 2.

Continuous time Markov chains

Let $M = \langle \lambda, Q \rangle$ be a continuous time Markov chain and Π be the corresponding jump matrix.

- A state is recurrent (transient) for M if it is recurrent (transient) for a discrete time Markov chain with transition matrix Π.
- The communicating classes of M are those defined by Π.
- M is irreducible iff Π is irreducible.

Continuous time Markov chains

Theorem

If Q is irreducible and recurrent. Then there is a unique distribution π with

- $\pi Q = 0$
- $\pi e^{tQ} = \pi$
- $\lim_{t \to \infty} (e^{tQ})_{ij} = \pi_j$

Time reversibility

- Does **not** mean that $a \rightarrow b$ and $b \rightarrow a$ are equally likely.
- Rather, the condition is

$$\begin{aligned} \pi_a p(t)_{ab} &= \pi_b p(t)_{ba} \\ \pi_a q_{ab} &= \pi_b q_{ba} \end{aligned}$$

- This means that sampling an a from the equilibrium distribution and observe a mutation to b in some interval t is as likely as sampling a b in equilibrium and see it mutate into a after time t.

Time reversibility

- Practical advantages of time reversibility:
  - If Q is time reversible, the lower triangle can be computed from the upper triangle, so we need only half the number of parameters.
  - The likelihood of a tree does not depend on the location of the root.

The Jukes-Cantor model

The **Jukes-Cantor model** of DNA evolution is defined by the rate matrix

$$Q = \begin{pmatrix} -3/4\mu & \mu/4 & \mu/4 & \mu/4 \\ \mu/4 & -3/4\mu & \mu/4 & \mu/4 \\ \mu/4 & \mu/4 & -3/4\mu & \mu/4 \\ \mu/4 & \mu/4 & \mu/4 & -3/4\mu \end{pmatrix}$$

$$\Pi = \begin{pmatrix} 0 & 1/3 & 1/3 & 1/3 \\ 1/3 & 0 & 1/3 & 1/3 \\ 1/3 & 1/3 & 0 & 1/3 \\ 1/3 & 1/3 & 1/3 & 0 \end{pmatrix}$$

The Jukes-Cantor model

- $\pi = (1/4, 1/4, 1/4, 1/4)$

$$P(t) = \begin{pmatrix} 1/4 + 3/4e^{-t\mu} & 1/4 - 1/4e^{-t\mu} & 1/4 - 1/4e^{-t\mu} & 1/4 - 1/4e^{-t\mu} \\ 1/4 - 1/4e^{-t\mu} & 1/4 + 3/4e^{-t\mu} & 1/4 - 1/4e^{-t\mu} & 1/4 - 1/4e^{-t\mu} \\ 1/4 - 1/4e^{-t\mu} & 1/4 - 1/4e^{-t\mu} & 1/4 + 3/4e^{-t\mu} & 1/4 - 1/4e^{-t\mu} \\ 1/4 - 1/4e^{-t\mu} & 1/4 - 1/4e^{-t\mu} & 1/4 - 1/4e^{-t\mu} & 1/4 + 3/4e^{-t\mu} \end{pmatrix}$$

Two-states model, equal rates

$$Q = \begin{pmatrix} -r & r \\ r & -r \end{pmatrix} \qquad P(t) = \frac{1}{2}\begin{pmatrix} 1+e^{-2rt} & 1-e^{-2rt} \\ 1-e^{-2rt} & 1+e^{-2rt} \end{pmatrix}$$

$$\pi = (1/2, 1/2)$$

Two-states model, different rates

$$Q = \begin{pmatrix} -r & r \\ s & -s \end{pmatrix} \quad P(t) = \frac{1}{r+s} \begin{pmatrix} s + re^{-(r+s)t} & r - re^{-(r+s)t} \\ s - se^{-(r+s)t} & r + se^{-(r+s)t} \end{pmatrix}$$

$$\pi = \left(s/r+s, r/r+s\right)$$

Two-states model, different rates

- if we measure time in expected number of mutations, we have

$$r + s = 1$$

- therefore:

Two-state model

$$Q = \begin{pmatrix} -r & r \\ s & -s \end{pmatrix} \quad P(t) = \begin{pmatrix} s + re^{-t} & r - re^{-t} \\ s - se^{-t} & r + se^{-t} \end{pmatrix}$$

$$\pi = (s, r)$$

The two-state model is always time reversible.

Estimating distances

Back to the running example

language	*iso_code*	*gloss*	*global_id*	*local_id*	*transcription*	*cognate_class*
ELFDALIAN	qov	woman	962	woman	ˈkɛ̀lɪŋg	woman:Ag
DUTCH	nld	woman	962	woman	vrɑu	woman:B
GERMAN	deu	woman	962	woman	fraŭ	woman:B
DANISH	dan	woman	962	woman	ˈgʰvenə	woman:D
DANISH_FJOLDE		woman	962	woman	k̊vinʲ	woman:D
GUTNISH_LAU		woman	962	woman	ˈkvɪnːˌfolk	woman:D
LATIN	lat	woman	962	woman	ˈmulier	woman:E
LATIN	lat	woman	962	woman	ˈfeːmina	woman:G
ENGLISH	eng	woman	962	woman	wʊmən	woman:H
GERMAN	deu	woman	962	woman	vaĭp	woman:H
DANISH	dan	woman	962	woman	ˈd̥ɛːmə	woman:K

- Let's focus on cognate classes for now.
- We transform the cognacy information into a **binary character matrix**

Binary character matrices

language	woman:Ag	woman:B	woman:D	woman:E	woman:G	woman:H	woman:K	···
DANISH	0	0	1	0	0	0	1	···
DANISH_FJOLDE	0	0	1	0	0	0	0	···
DUTCH	0	1	0	0	0	0	0	···
ELFDALIAN	1	0	0	0	0	0	0	···
ENGLISH	0	0	0	0	0	1	0	···
GERMAN	0	1	0	0	0	1	0	···
GUTNISH_LAU	0	0	1	0	0	0	0	···
LATIN	0	0	0	1	1	0	0	···

Binary character matrices

- We assume that gain/loss of cognate classes follows continuous time Markov process, and that characters a stochastically independent.
- Both assumptions are clearly false:
  - Markov assumption is violated due to language contact → borrowings constitute mutations, but their probability depends on the state of the borrowing and the receiving language
  - gaining a cognate class for a given concept increases likelihood for loss of different class and vice versa (avoidance of lexical gaps and synonymy)
  - ...
- For the time being, we will also assume that all cognate classes have the same mutation rate. (OMG!!!)
- Justification: Let's start with the simplest model possible and refine it step by step when necessary.

Dollo model

- Ideally, each cognate class can be lost multiple times, but it can be gained only once.
- This amounts to a model with

$$\begin{aligned} r &\approx 0 \\ s &\approx 1 \end{aligned}$$

- This goes by the name of **Dollo model** in theoretical biology.

Dollo model

Why the Dollo model is wrong

- Borrowings have the effect of introducing a cognate class into a lineage which originated elsewhere $\rightarrow$ multiple mutations $0 \rightarrow 1$
- Parallel semantic change:
  - IELex cognate class *leg:Q* derived from *foot:B* independently in Greek, Indo-Iranian, Romanian, Swabian...

- Dollo model is still a good approximation

Estimating distances

- Let's consider Italian and English
- contingeny matrix (ignoring all characters where one of the two languages is undefined)

	English : 0	English : 1
Italian : 0	1021	144
Italian : 1	129	62

- normalized

	English : 0	English : 1
Italian : 0	0.753	0.106
Italian : 1	0.095	0.046

Estimating distances

- model is time-reversible, so we can safely pretend that English is a direct descendant of Italian
- we also assume that Italian is in equilibrium
- note though: there are virtually infinitely possible cognate classes not covered, so the true frequency of 0s is much higher than our counts
- expected values of normalized contingency table (t is the distance between Italian and English)

$$P(t) \begin{pmatrix} s & 0 \\ 0 & r \end{pmatrix} = \begin{pmatrix} s^2 + rse^{-t} & rs - rse^{-t} \\ rs - rse^{-t} & r^2 + rse^{-t} \end{pmatrix}$$

Dice distance

Definition (Dice distance)

$$\mathsf{dice}(A, B) \quad = \quad \frac{|A - B| + |B - A|}{|A| + |B|}$$

- If time t has passed between initial and final state, we expect the Dice distance between initial and final state to be (for positive r)

$$\mathsf{dice}(x, y) \quad = \quad s(1 - e^{-t})$$

- If we have an estimate of $\mathsf{dice}(\mathsf{x}, \mathsf{y})$, we can estimate t as

$$t = -\log(1 - \frac{\mathsf{dice}(x, y)}{s})$$

Dice distance

- According to Dollo assumption, r converges to 0 and s to 1

$$
\begin{aligned}
t &= -\log(1 - \text{dice}(x, y)) \\
\text{dice}(\textit{Italian}, \textit{English}) &= 0.688 \\
t &= 1.164
\end{aligned}
$$

Estimated distances

	Bengali	Breton	Bulgarian	Catalan	Czech	Danish	Dutch	English	French
Bengali	–	2.16	1.64	1.39	1.81	1.41	1.24	1.33	1.28
Breton	2.16	–	1.81	1.67	1.77	1.82	1.86	1.80	1.64
Bulgarian	1.64	1.81	–	1.55	0.34	1.44	1.52	1.31	1.56
Catalan	1.39	1.67	1.55	–	1.53	1.40	1.37	1.17	0.29
Czech	1.81	1.77	0.34	1.53	–	1.40	1.44	1.34	1.53
Danish	1.41	1.82	1.44	1.40	1.40	–	0.45	0.48	1.38
Dutch	1.24	1.86	1.52	1.37	1.44	0.45	–	0.51	1.31
English	1.33	1.80	1.31	1.17	1.34	0.48	0.51	–	1.09
French	1.28	1.64	1.56	0.29	1.53	1.38	1.31	1.09	–
German	1.25	1.72	1.45	1.39	1.40	0.43	0.27	0.49	1.28
Greek	1.57	2.09	1.74	1.72	1.85	1.64	1.69	1.64	1.71
Hindi	0.54	1.89	1.33	1.24	1.34	1.53	1.56	1.41	1.22
Icelandic	1.29	1.85	1.50	1.48	1.51	0.25	0.60	0.58	1.44
Irish	1.87	0.85	1.44	1.58	1.37	1.38	1.38	1.31	1.35
Italian	1.40	1.52	1.51	0.24	1.52	1.32	1.30	1.16	0.26
Lithuanian	2.22	1.66	0.84	1.22	0.83	1.34	1.41	1.25	1.19
Nepali	0.56	0.18	0.20	0.13	0.30	0.20	0.30	0.20	0.20
Polish	1.65	1.86	0.43	1.56	0.28	1.44	1.42	1.32	1.51
Portuguese	1.34	1.57	1.49	0.30	1.44	1.39	1.39	1.16	0.36
Romanian	1.32	1.05	1.19	0.32	1.19	1.12	1.09	1.00	0.28
Russian	1.64	1.73	0.34	1.49	0.29	1.38	1.45	1.26	1.44
Spanish	1.36	1.55	1.47	0.21	1.45	1.42	1.38	1.15	0.30
Swedish	1.43	1.87	1.49	1.41	1.44	0.15	0.49	0.57	1.43
Ukrainian	1.67	1.82	0.40	1.53	0.32	1.45	1.46	1.32	1.51
Welsh	2.08	0.38	1.39	1.19	1.41	1.00	1.08	1.15	1.02

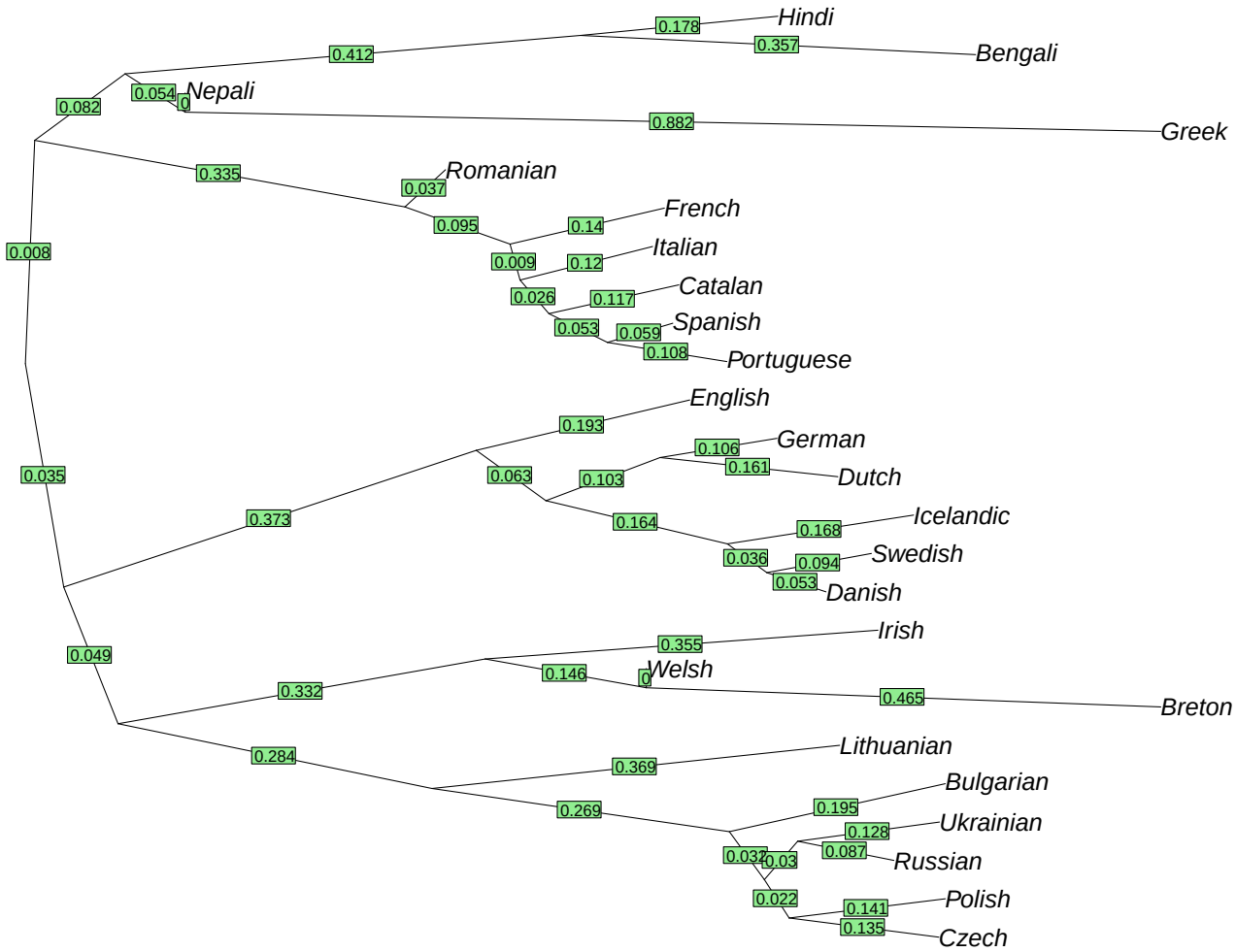
Estimated distances

	German	Greek	Hindi	Icelandic	Irish	Italian	Lithuanian	Nepali	Polish
Bengali	1.25	1.57	0.54	1.29	1.87	1.40	2.22	0.56	1.65
Breton	1.72	2.09	1.89	1.85	0.85	1.52	1.66	0.18	1.86
Bulgarian	1.45	1.74	1.33	1.50	1.44	1.51	0.84	0.20	0.43
Catalan	1.39	1.72	1.24	1.48	1.58	0.24	1.22	0.13	1.56
Czech	1.40	1.85	1.34	1.51	1.37	1.52	0.83	0.30	0.28
Danish	0.43	1.64	1.53	0.25	1.38	1.32	1.34	0.20	1.44
Dutch	0.27	1.69	1.56	0.60	1.38	1.30	1.41	0.30	1.42
English	0.49	1.64	1.41	0.58	1.31	1.16	1.25	0.20	1.32
French	1.28	1.71	1.22	1.44	1.35	0.26	1.19	0.20	1.51
German	–	1.65	1.46	0.61	1.30	1.28	1.30	0.20	1.38
Greek	1.65	–	1.53	1.68	1.70	1.60	1.74	0.41	1.85
Hindi	1.46	1.53	–	1.64	1.40	1.28	1.37	0.08	1.35
Icelandic	0.61	1.68	1.64	–	1.43	1.44	1.34	0.30	1.55
Irish	1.30	1.70	1.40	1.43	–	1.30	1.32	0.46	1.41
Italian	1.28	1.60	1.28	1.44	1.30	–	1.18	0.24	1.55
Lithuanian	1.30	1.74	1.37	1.34	1.32	1.18	–	0.81	0.78
Nepali	0.20	0.41	0.08	0.30	0.46	0.24	0.81	–	0.30
Polish	1.38	1.85	1.35	1.55	1.41	1.55	0.78	0.30	–
Portuguese	1.30	1.63	1.27	1.44	1.47	0.32	1.25	0.20	1.44
Romanian	1.00	1.36	0.96	1.18	1.00	0.26	1.20	0.22	1.19
Russian	1.36	1.78	1.34	1.46	1.41	1.48	0.84	0.20	0.32
Spanish	1.32	1.67	1.21	1.50	1.37	0.28	1.18	0.20	1.46
Swedish	0.50	1.68	1.60	0.30	1.38	1.36	1.41	0.20	1.46
Ukrainian	1.42	1.88	1.31	1.51	1.41	1.52	0.79	0.30	0.27
Welsh	0.94	1.12	0.96	1.20	0.54	1.02	0.69	0.69	1.39

Estimated distances

	Portuguese	Romanian	Russian	Spanish	Swedish	Ukrainian	Welsh
Bengali	1.34	1.32	1.64	1.36	1.43	1.67	2.08
Breton	1.57	1.05	1.73	1.55	1.87	1.82	0.38
Bulgarian	1.49	1.19	0.34	1.47	1.49	0.40	1.39
Catalan	0.30	0.32	1.49	0.21	1.41	1.53	1.19
Czech	1.44	1.19	0.29	1.45	1.44	0.32	1.41
Danish	1.39	1.12	1.38	1.42	0.15	1.45	1.00
Dutch	1.39	1.09	1.45	1.38	0.49	1.46	1.08
English	1.16	1.00	1.26	1.15	0.57	1.32	1.15
French	0.36	0.28	1.44	0.30	1.43	1.51	1.02
German	1.30	1.00	1.36	1.32	0.50	1.42	0.94
Greek	1.63	1.36	1.78	1.67	1.68	1.88	1.12
Hindi	1.27	0.96	1.34	1.21	1.60	1.31	0.96
Icelandic	1.44	1.18	1.46	1.50	0.30	1.51	1.20
Irish	1.47	1.00	1.41	1.37	1.38	1.41	0.54
Italian	0.32	0.26	1.48	0.28	1.36	1.52	1.02
Lithuanian	1.25	1.20	0.84	1.18	1.41	0.79	0.69
Nepali	0.20	0.22	0.20	0.20	0.20	0.30	0.69
Polish	1.44	1.19	0.32	1.46	1.46	0.27	1.39
Portuguese	–	0.28	1.39	0.17	1.43	1.44	0.96
Romanian	0.28	–	1.13	0.24	1.13	1.20	0.69
Russian	1.39	1.13	–	1.41	1.43	0.22	1.23
Spanish	0.17	0.24	1.41	–	1.45	1.48	1.03
Swedish	1.43	1.13	1.43	1.45	–	1.46	1.06
Ukrainian	1.44	1.20	0.22	1.48	1.46	–	1.25
Welsh	0.96	0.69	1.23	1.03	1.06	1.25	–

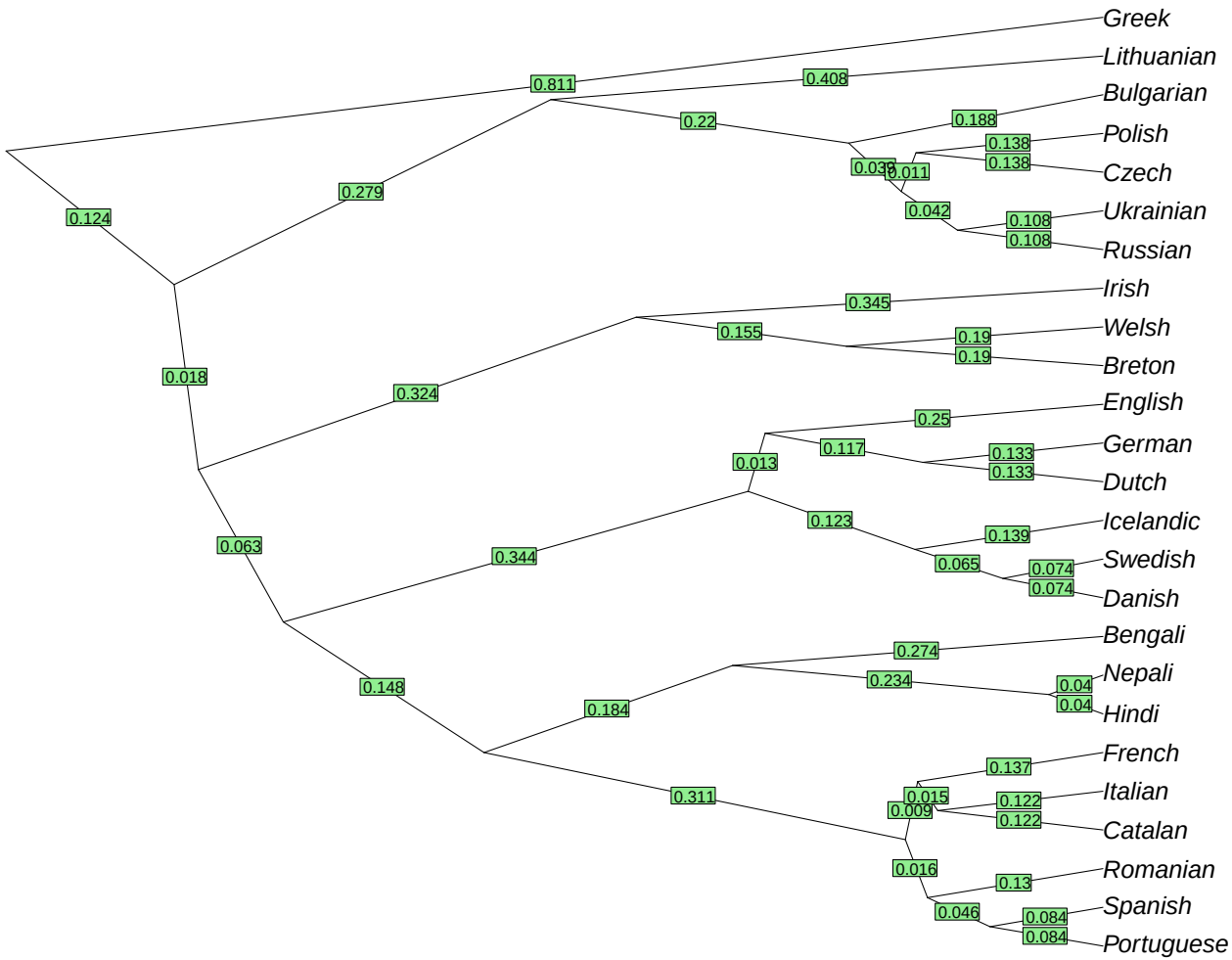
Neighbor Joining tree



Neighbor Joining tree

- data sparseness for *Nepali* (only 31 characters are defined) → all distances come out as way too small
- note that root was determined by *midpoint rooting* to make it look nicer
- Neighbor Joining does not tell us anything about the location of the root
- tree structure is largely consistent with received opinion (except that Italian and French should swap places, and English is too high within Germanic)

UPGMA tree


Greek
Lithuanian
Bulgarian
Polish
Czech
Ukrainian
Russian
Irish
Welsh
Breton
English
German
Dutch
Icelandic
Swedish
Danish
Bengali
Nepali
Hindi
French
Italian
Catalan
Romanian
Spanish
Portuguese
0.811
0.408
0.22
0.188
0.138
0.138
0.039
0.011
0.042
0.108
0.108
0.279
0.124
0.345
0.155
0.19
0.19
0.018
0.324
0.25
0.117
0.133
0.133
0.013
0.123
0.139
0.065
0.074
0.074
0.063
0.344
0.274
0.234
0.04
0.04
0.148
0.184
0.137
0.015
0.122
0.122
0.009
0.311
0.016
0.13
0.046
0.084
0.084

UPGMA tree

- tree structure largely recognizes the major sub-groupings
- fine structure of Romance is a bit of a mess

WALS features

- WALS features are binarized → binary character matrix

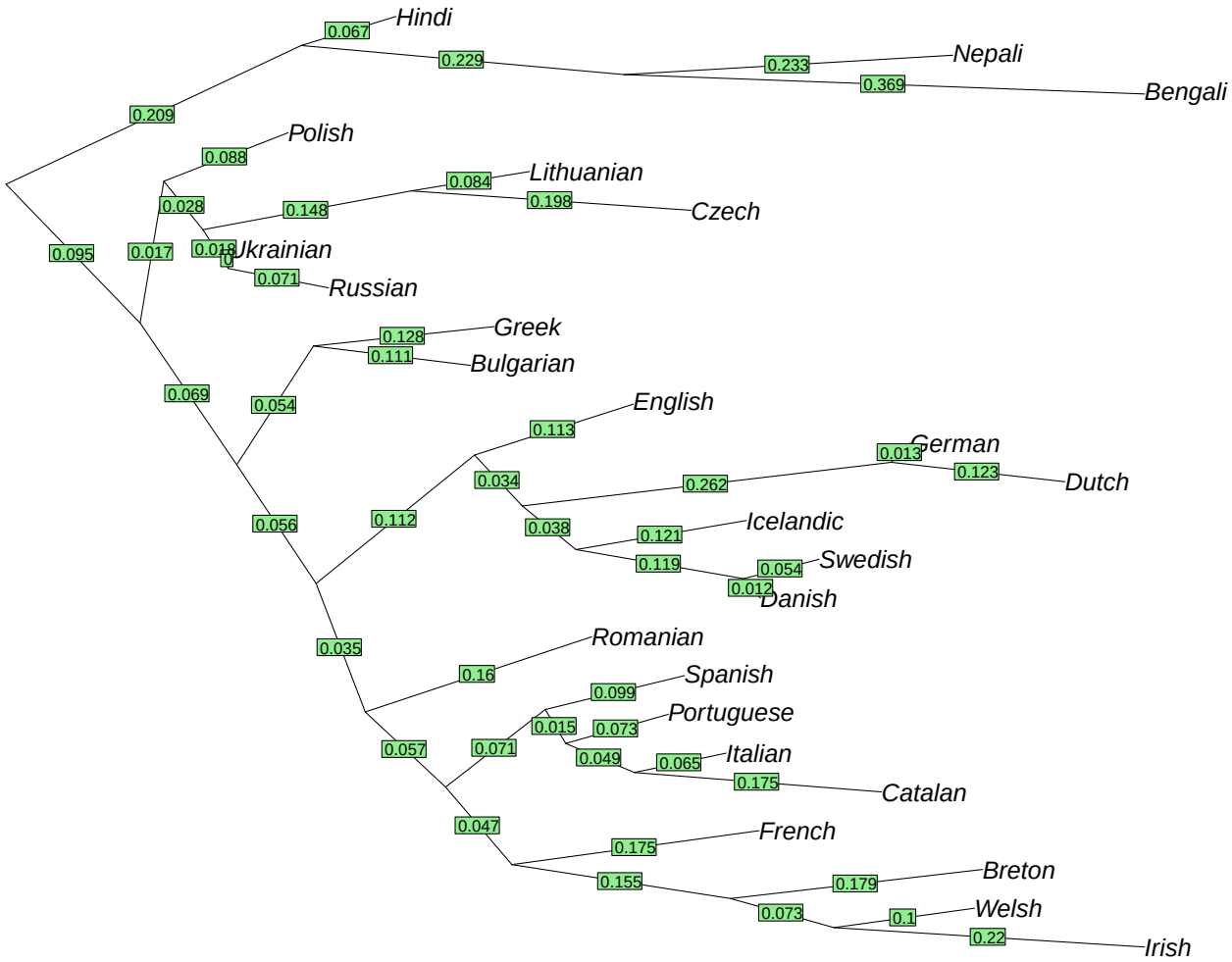
language	SVO	SOV	VSO	no dominant order	···
DANISH	1	0	0	0	···
DUTCH	0	0	0	1	···
ENGLISH	1	0	0	0	···
GERMAN	0	0	0	1	···
GREEK	0	0	0	1	···
HINDI	0	1	0	0	···
ICELANDIC	1	0	0	0	···
WELCH	0	0	1	0	···

WALS features

- Dollo assumption is too far off the mark here to apply it
- We need an estimate for (r, s)!
- Null assumption: for each WALS feature, all values are equally likely in equilibrium
- leads to estimate

$$\begin{aligned} r &= \frac{\text{number of WALS features}}{\text{number of binary characters}} \\ &\approx 0.14 \\ s &= 1 - r \approx 0.86 \end{aligned}$$

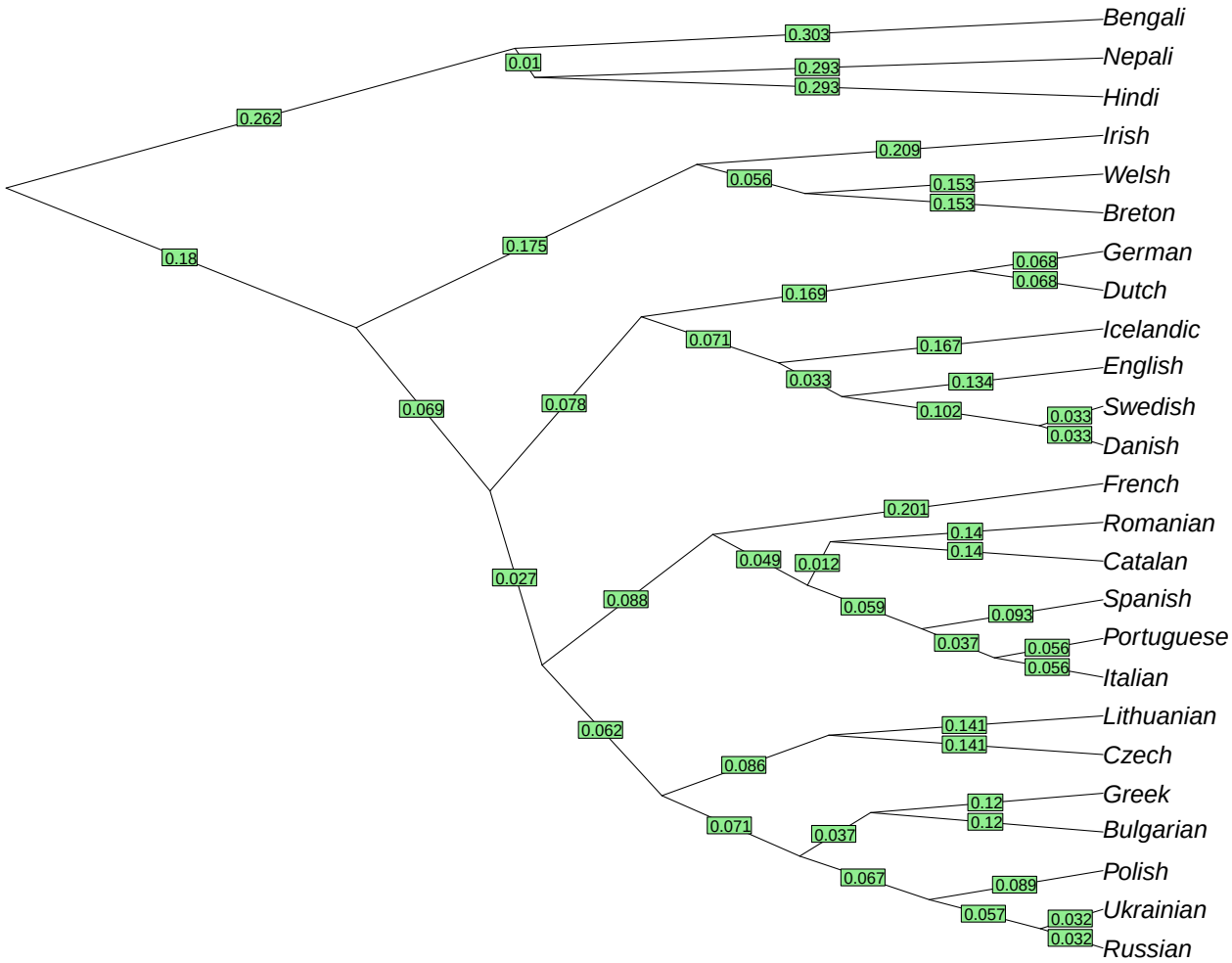
Neighbor Joining tree



Neighbor Joining tree

- clearly worse than cognacy tree
- some oddities
  - Polish and Lithuanian have swapped places
  - Celtic comes out as sub-group of Romance
  - Bulgarian far removed from the rest of Slavic; it is sister-taxon of Greek

UPGMA tree


Bengali
Nepali
Hindi
Irish
Welsh
Breton
German
Dutch
Icelandic
English
Swedish
Danish
French
Romanian
Catalan
Spanish
Portuguese
Italian
Lithuanian
Czech
Greek
Bulgarian
Polish
Ukrainian
Russian
0.262
0.01
0.303
0.293
0.293
0.18
0.175
0.209
0.056
0.153
0.153
0.069
0.078
0.169
0.068
0.068
0.071
0.167
0.033
0.134
0.102
0.033
0.033
0.027
0.088
0.201
0.14
0.14
0.049
0.012
0.059
0.093
0.037
0.056
0.056
0.062
0.086
0.141
0.141
0.071
0.037
0.12
0.12
0.067
0.089
0.057
0.032
0.032

UPGMA tree

- somewhat better, but still pretty bad
- some oddities
  - Greek as Slavic language
  - Czech as Baltic language
  - Romanian and Catalan are much too close

$\Rightarrow$ typological features are ill-suited for phylogenetic estimation

  - strong influence of language contact
  - non-independence of features
  - data sparseness

Working with phonetic strings

Phonetic characters

- cognacy data and grammatical/typological classifications rely on expert judgments:
  - labor intensive
  - subjective, hard to replicate
- sound change, a very conspicuous aspect of language change, is ignored
- information on sound change does not come in nicely packaged discrete characters though

Working with phonetic strings

- quick-and-dirty method to extract binary characters from phonetic strings:
  1. convert phonetic entries into ASJP format
  2. presence-absence characters for each sound class/concept combination
  3. character changes can represent sound shift or lexical replacement
     Latin *puer* → Italian *bambino*
     child/p:1 → child/p:0
     Latin *oculus* → Italian *occhio*
     eye/u:1 → eye/u:0

language	*phonological form* (IELex)	*ASJP representation*
Bengali	-	-
Breton	-	-
Bulgarian	muˈrɛ	murE
Catalan	mar; mar; ma	mar; mar; ma
Czech	ˈmɔɻɛ	morE
Danish	hɑw;søˀ	how; se
Dutch	ze	ze
English	si:	si
French	mɛʁ	mEr
German	ze:;ʾo:t͡sea:n;me:ɐ̯	ze; otsean; mea
Greek	ˈθalaˌsa	8alasa
Hindi	-	-
Icelandic	haːv/sjouːr	hav; syour
Irish	ˈfʷærʷʲɪ	fErCi
Italian	ˈmare	mare
Lithuanian	ˈjuːrɛ	yura
Nepali	-	-
Polish	ˈmɔʐɛ	moZE
Portuguese	mar	mar
Romanian	ˈmare	mare
Russian	ˈmɔrʲɛ	morE
Spanish	mar	mar
Swedish	hɑːv; ɧøː	hov; Se
Ukrainian	ˈmɔrɛ	morE
Welsh	-	-

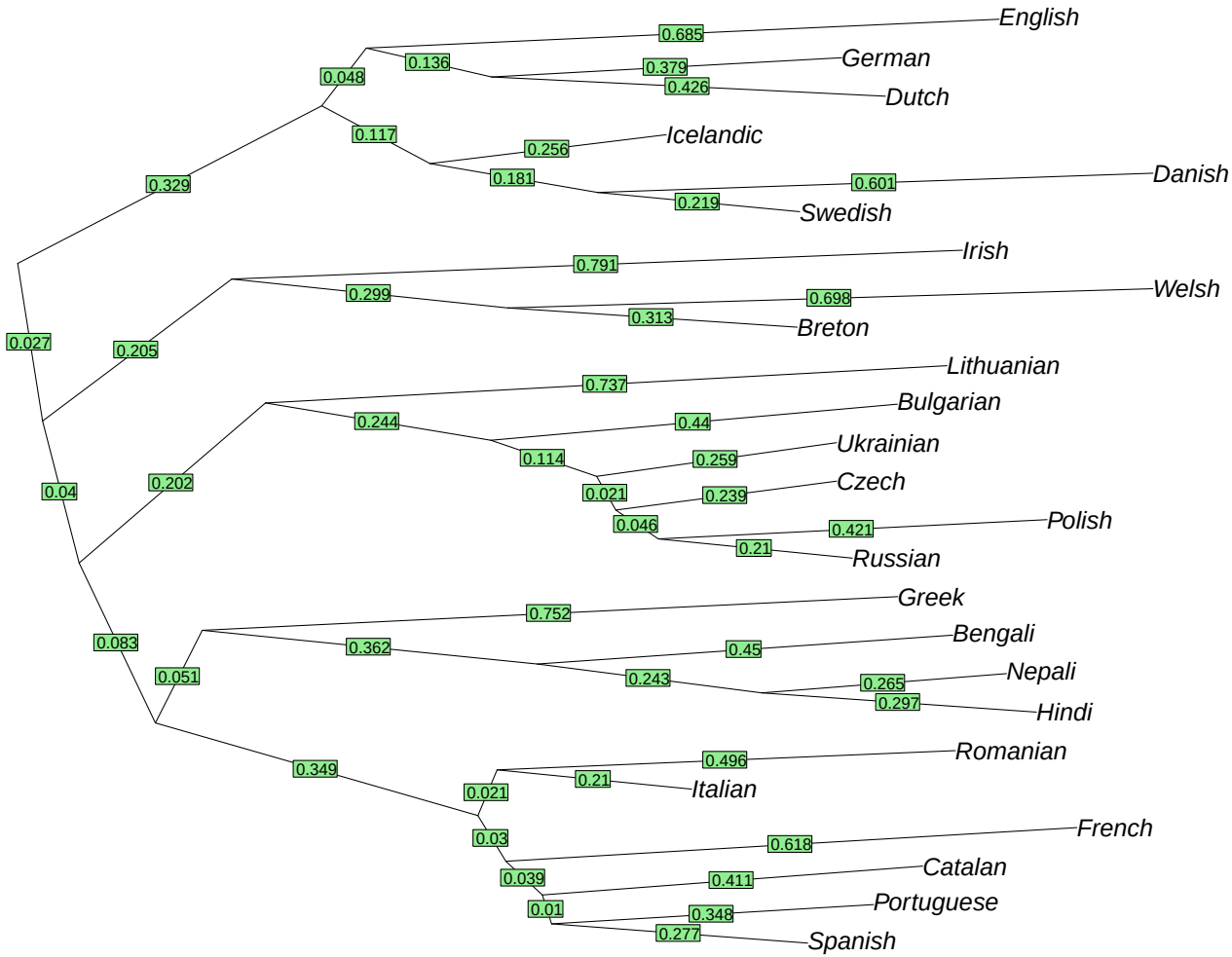
Working with phonetic strings

	see:m	see:r	see:a	see:s	···	see:Z
Bengali	-	-	-	-	···	-
Bulgarian	1	1	0	0	···	0
Catalan	1	1	1	0	···	0
Czech	1	1	0	0	···	0
Danish	0	0	0	1	···	0
Italian	1	1	1	0	···	0
Ukrainian	1	1	0	0	···	0
⋮	⋮	⋮	⋮	⋮	⋱	⋮

- estimating r as

$$\frac{\sum_{s \in \text{sound classes}} \frac{|\{w \in \text{words} | s \in w\}|}{|\text{words}|}}{|\text{sound classes}|} \approx 0.105$$

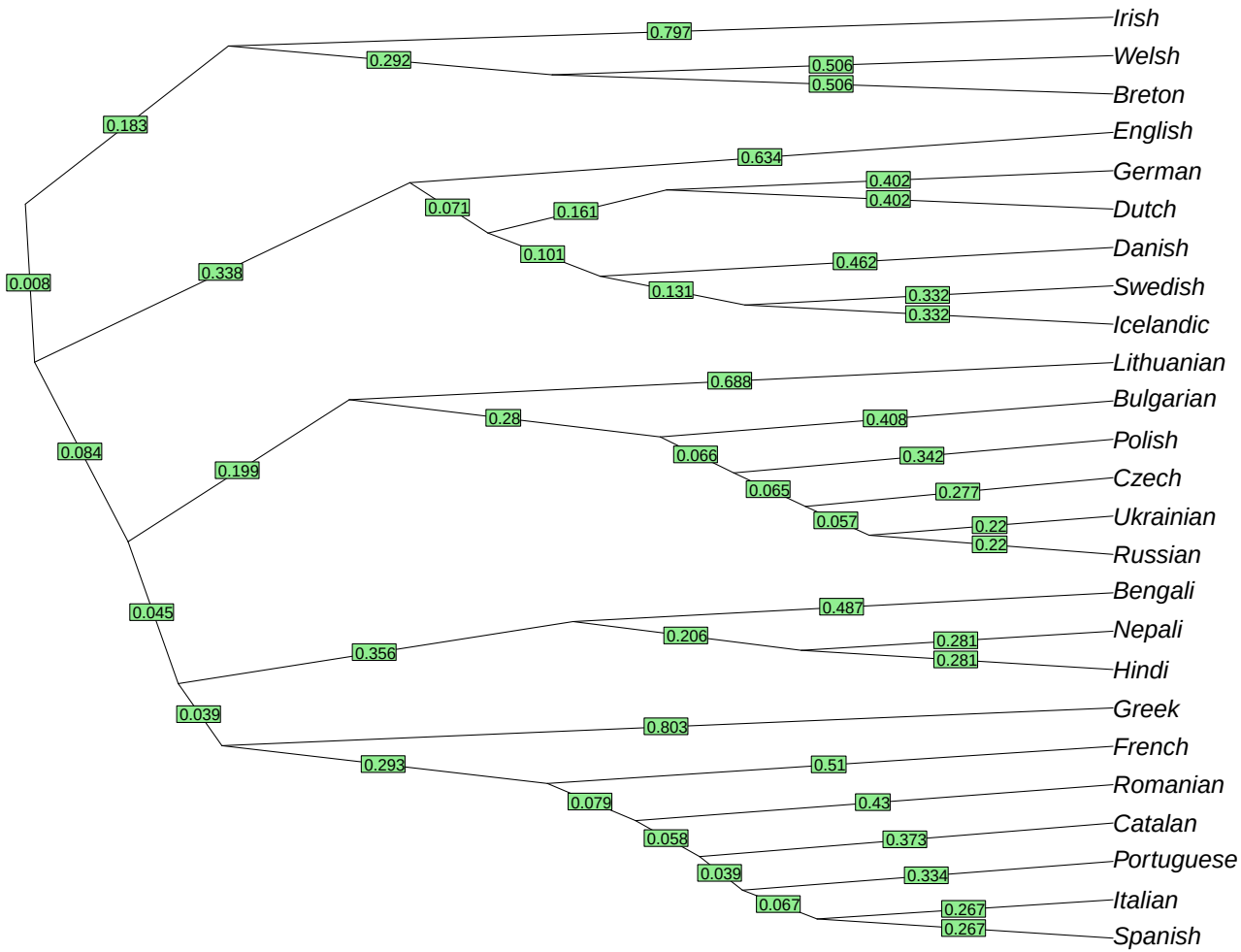
Neighbor Joining tree



Neighbor Joining tree

- almost fully consistent with expert opinion
- two deviations
  - Russian should be next two Ukrainian rather than next to Polish (language contact?)
  - Italian and Romanian shouldn't be neighbors

UPGMA tree


0.797
0.292
0.506
0.506
0.183
0.634
0.402
0.402
0.071
0.161
0.101
0.462
0.008
0.338
0.131
0.332
0.332
0.688
0.28
0.408
0.084
0.199
0.066
0.342
0.065
0.277
0.057
0.22
0.22
0.045
0.487
0.206
0.281
0.281
0.356
0.039
0.803
0.293
0.51
0.079
0.43
0.058
0.373
0.039
0.334
0.067
0.267
0.267
Irish
Welsh
Breton
English
German
Dutch
Danish
Swedish
Icelandic
Lithuanian
Bulgarian
Polish
Czech
Ukrainian
Russian
Bengali
Nepali
Hindi
Greek
French
Romanian
Catalan
Portuguese
Italian
Spanish

UPGMA tree

- somewhat worse than NJ tree
- some oddities
  - English too high within Germanic
  - position of Russian is correct, but Czech comes out as East Slavic
  - Italian and French at wrong positions within Romance

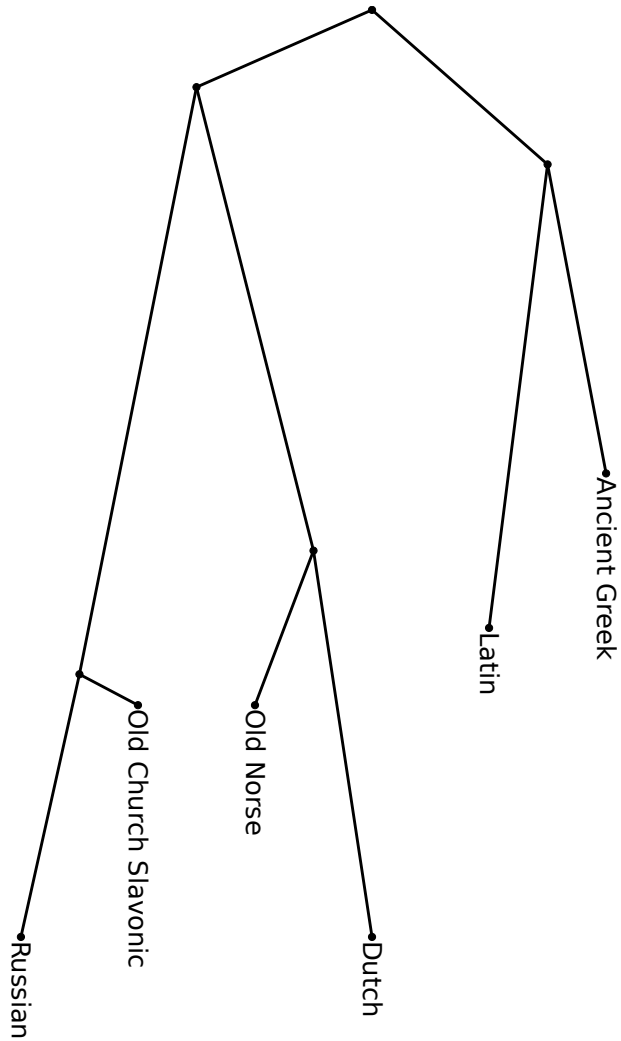
Hands-on

Data formats

Newick format for trees

- see Wikipedia entry for details
- bracketed string
- labels of internal nodes (optional) after closing bracket
- edge lengths (optional) after node name, separated by ":"
- example:

```
(("Ancient Greek":2,Latin:3):1,
((Dutch:2.5, "Old Norse":1):3,
("Old Church Slavonic":0.2,
Russian:1.7):3.8):0.5);
```



Data formats

Character matrices as Nexus files

- **Nexus** (suffix `.nex`): versatile file format for phylogenetic information
- Structure of a Nexus file for a binary character matrix:
  1. header (`ntax` = number of rows, `nchar`=number of columns):

```
#NEXUS

BEGIN DATA;
DIMENSIONS ntax=25 NCHAR=1481;
FORMAT DATATYPE=STANDARD GAP=? MISSING=- interleave=yes;
MATRIX
```

Data formats

Character matrices as Nexus files

2. matrix: each row consists of the taxon name, followed by white space, followed by matrix entries; undefined values are represented by "–"

```
Greek       000100001000000000...
Bulgarian   001000001000000010...
Russian     001000001000000010...
Romanian    -----010000--------...
⋮                       ⋮
```

3. footer:

```
;
END;
```

Loading Nexus files into R

- *phangorn* is geared towards biomolecular data
- some workaround needed to handle binary matrices

```
library(ape)
library(phangorn)

contrasts <- matrix(data=c(1,0,
                           0,1,
                           1,1),
                    ncol=2,byrow=T)

dimnames(contrasts) <- list(c('0','1','-'),
                            c('0','1'))

cognacy.data <- phyDat(read.nexus.data('ielex.bin.nex'),
                       'USER',
                       levels=c('0','1','-'),
                       contrast=contrasts,
                       ambiguity='-')

cognacy.matrix <- as.character(cognacy.data)
```

Exercise

- run the script `loadNexusFiles.r` in an interactive session
- implement the Dice distance. Note that all characters with value "-" in either of the vectors compared have to be ignored
- computed the distance matrices for the three Nexus files, using the estimates for s from the slides
- compute the Neighbor Joining trees, using the function `nj()`
- display the tree with the `plot()` command
- experiment with different values for s to get a feel for how sensitive the result is for this parameter

Ewens, W. and G. Grant (2005). *Statistical Methods in Bioinformatics: An Introduction*. Springer, New York.