Phylogenetic trees I
Foundations, Distance-based inference

Gerhard Jäger

ESSLLI 2016

Background readings for this lecture

- Ewens and Grant (2005), sections 15.1–15.4
- Nunn (2011), chapter 2

Why trees?

- tree diagrams have long history in linguistics and life sciences:
  - taxonomies (from Aristotle to Linné)
  - tree of life (Darwin)
  - language family trees (Schleicher)
- commonalities between biological and language family trees:
  - tree diagram represents a historical hypothesis
  - internal nodes represent a historical reality, not just a taxonomic category
- technical term for this kind of tree: **phylogenetic tree** (aka *phylogeny*)

Some definitions

Definition (Tree)

An *unrooted tree* is a connected undirected acyclic weighted graph with positives weights. In other words, an unrooted tree $\mathcal{T}$ is a triple (V, E, l) with

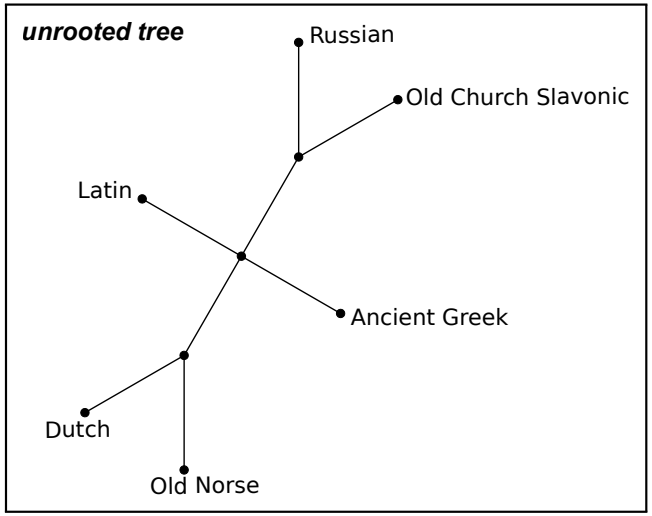
- V is a finite set, the *nodes* or *vertices*,
- $E \subset V \times V$, the set of *edges*, is symmetric,
- E^+ (E's transitive closure) is irreflexive,
- $E^* = V \times V$, and
- $l : E \mapsto \mathbb{R}^+$ is a function assigning each edge a non-negative *length*.

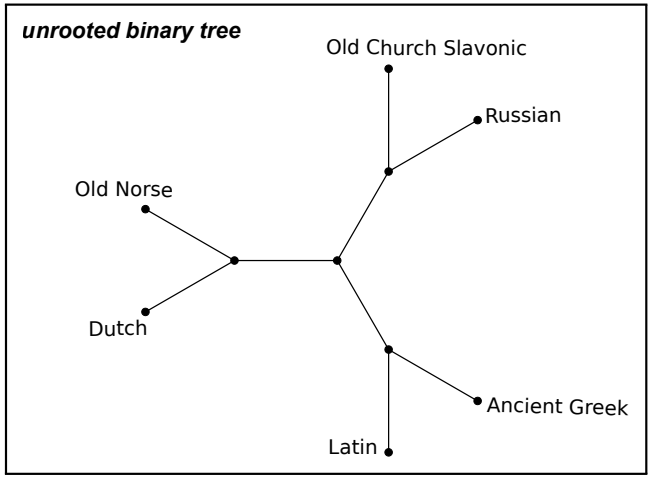
Remark: Unrooted trees might seem to be unintuitive data structures. Later on we will see though that often, estimating the unrooted version of a phylogeny is a quite different task from estimating the location of the root. So it makes sense to separate the two problems.

Some more definitions

Definition

- The *degree* of node v is the number of edges containing v as a component.
- Nodes with degree 1 are called *tips* or *leaves*.
- An *unrooted binary tree* is an unrooted tree with all nodes having degree 3 or 1.

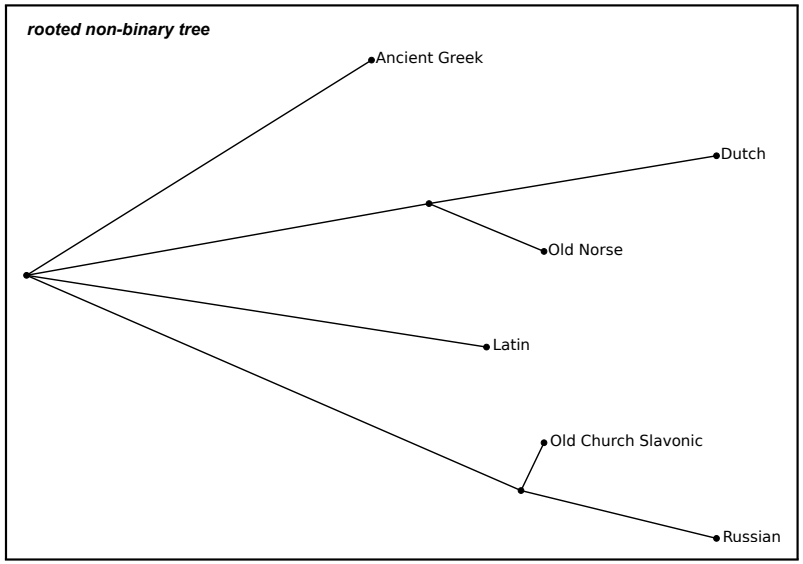


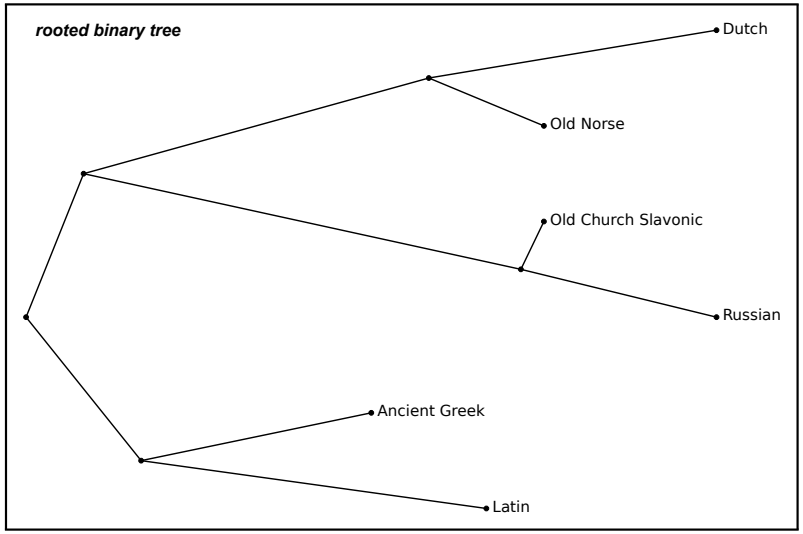


Even more definitions

Definition (Rooted trees)

- A *rooted tree* is a pair $(\mathcal{T}, v)$, where $\mathcal{T}$ is an unrooted tree and v is a designated vertex in $\mathcal{T}$ (its *root*).
- A *rooted binary tree* is an unrooted tree where exactly one node (the root) has degree 2 and all other nodes have degrees 1 or 3.





Distances

Definition (Distances)

Let $\mathcal{T} = (V, E, l)$ be a tree. Let $d : V \times V \mapsto \mathbb{R}$ be the unique function such that for all $a, b \in V$:

- If $(a, b) \in E$, then $d(a, b) = l(a, b)$.
- $l(a, a) = 0$.
- $d(a, b) = d(b, a)$.
- $l(a, b) = min_c(l(a, c) + l(c, b))$

Vulgo: $d(a, b)$ is the length of the unique path between a and b.

Ultrametric trees

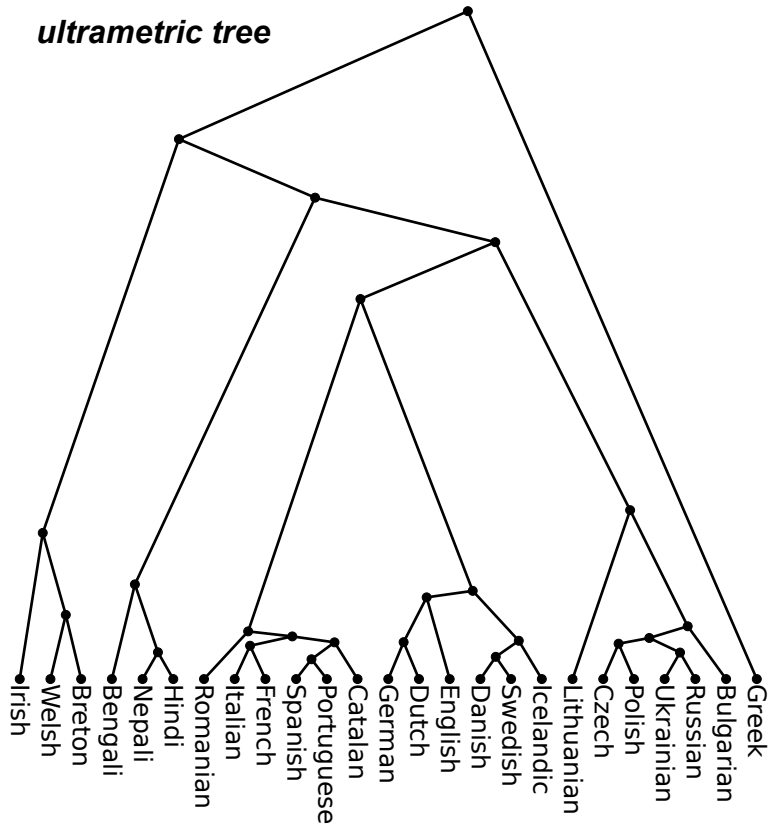
Definition (Ultrametric distance)

d is an *ultrametric distance* if it is a metric ($d(a,a) = 0, d(a,b) = d(b,a) \geq 0, d(a,b) + d(b,c) \geq d(a,c)$) with

$$d(a,b) \leq max\{d(a,c), d(b,c)\}$$

Definition (Ultrametric tree)

A rooted tree is *ultrametric* iff all tips have the same distance from the root.



Ultrametric trees

Theorem

The pairwise distances between a set of taxa are ultrametric if and only if there is an ultrametric tree with the taxa as tips representing those distances.

Proof: By induction over number of taxa.

Unweighted Pair Group Method Using Arithmetic Averages (UPGMA) algorithm constructs ultrametric tree from pairwise distances.

UPGMA

Cluster distances

Led A and B be two non-empty sets of taxa.

$$d(A,B) \doteq \frac{1}{|A| \times |B|} \sum_{x \in A, y \in B} d(x,y)$$

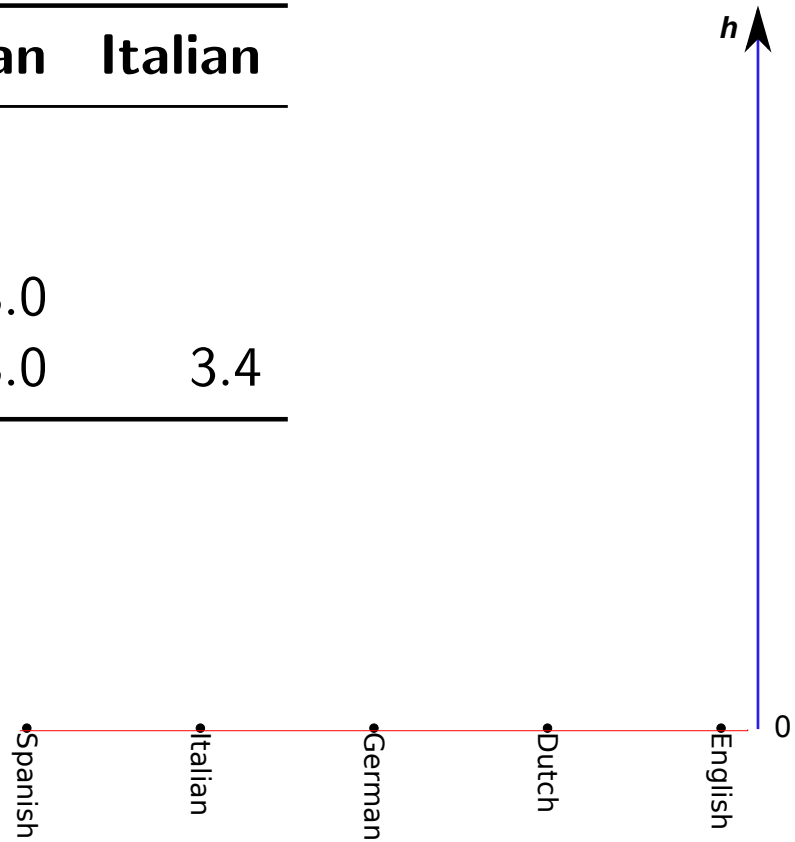
UPGMA

UPGMA algorithm

- **Initialization:**
  - $X \leftarrow$ the set of taxa.
  - $V \leftarrow X$
  - $E \leftarrow \emptyset$
  - $h(x) = 0 \; \forall x \in X$
- **Iteration:**
  - **while** $|X| > 1$
    - $\{i, j\} \leftarrow \arg_{x \in X, y \in X, x \neq y} \min d(x, y)$
    - $X \leftarrow X \setminus \{i, j\} \cup \{\{i, j\}\}$
    - $V \leftarrow V \cup \{\{i, j\}\}$
    - $E \leftarrow E \cup \{(\{i, j\}, i), (\{i, j\}, j)\}$
    - $h(\{i, j\}) = {}^{d(i,j)}/_{2}$
    - $l(\{i, j\}, i) = h(\{i, j\}) - h(i)$
    - $l(\{i, j\}, j) = h(\{i, j\}) - h(j)$
    - $d(\{i, j\}, k) = {}^{d(i,k)+d(j,k)}/_{2}$

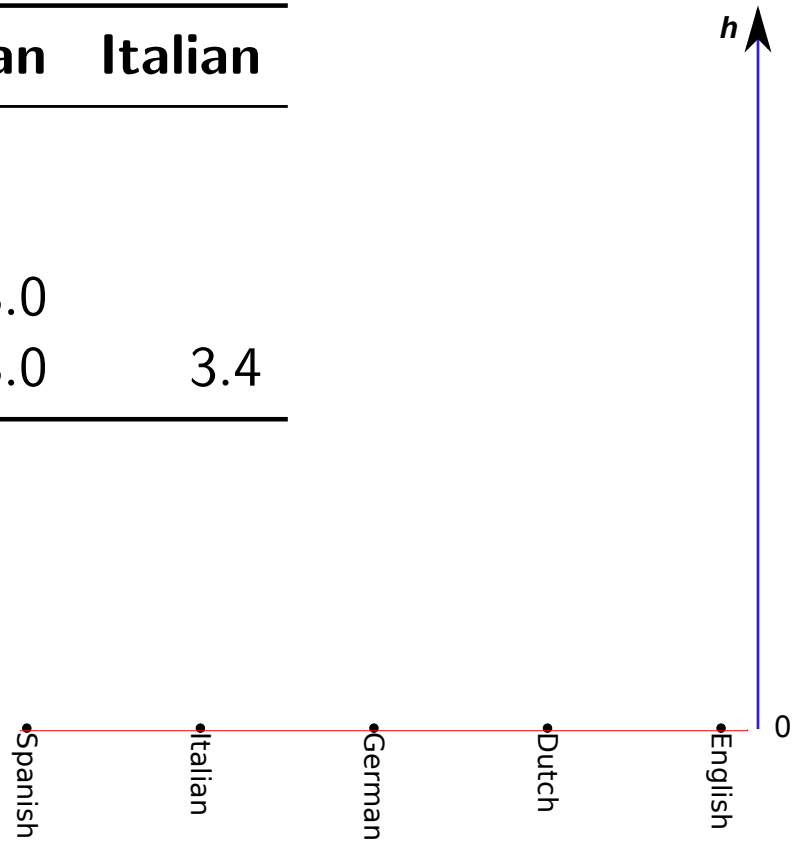
Example

	English	Dutch	German	Italian
Dutch	3.0			
German	3.0	2.0		
Italian	8.0	8.0	8.0	
Spanish	8.0	8.0	8.0	3.4



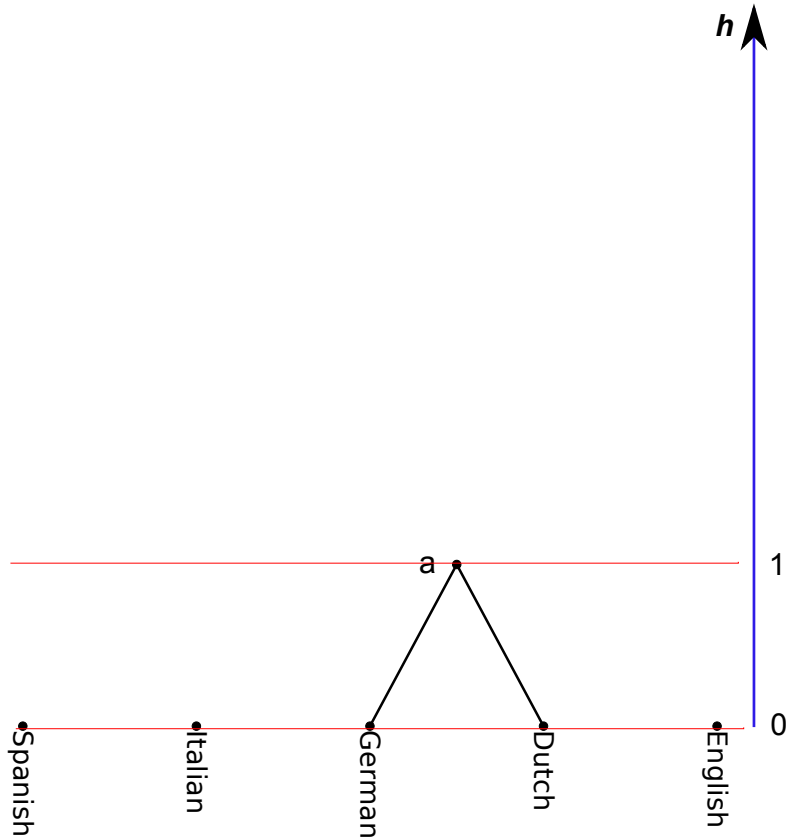
Example

	English	Dutch	German	Italian
Dutch	3.0			
German	3.0	**2.0**		
Italian	8.0	8.0	8.0	
Spanish	8.0	8.0	8.0	3.4



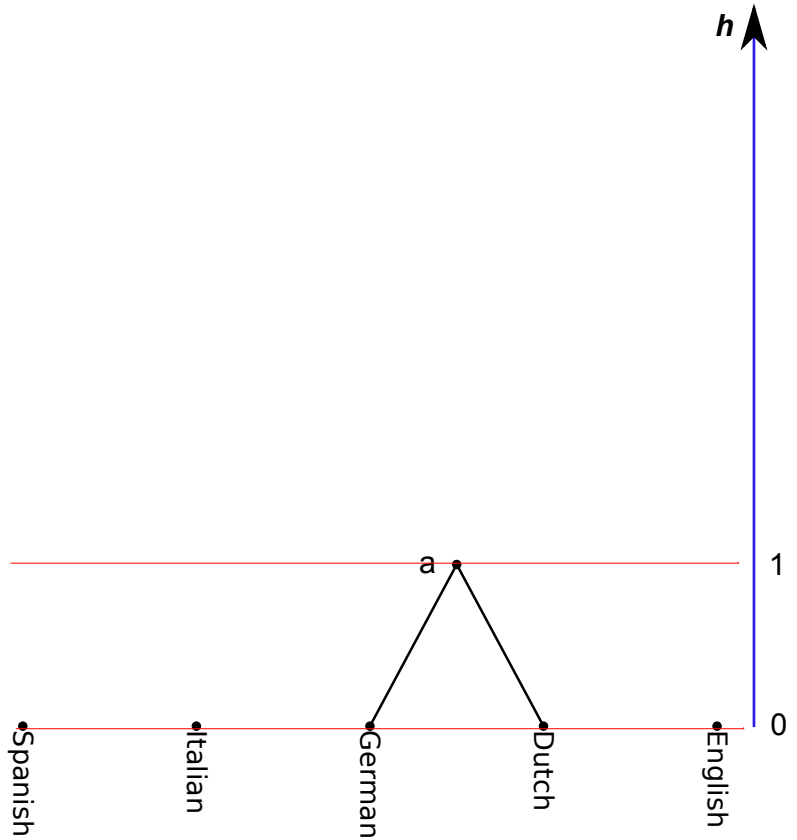
Example

	English	a	Italian
a	3.0		
Italian	8.0	8.0	
Spanish	8.0	8.0	3.4



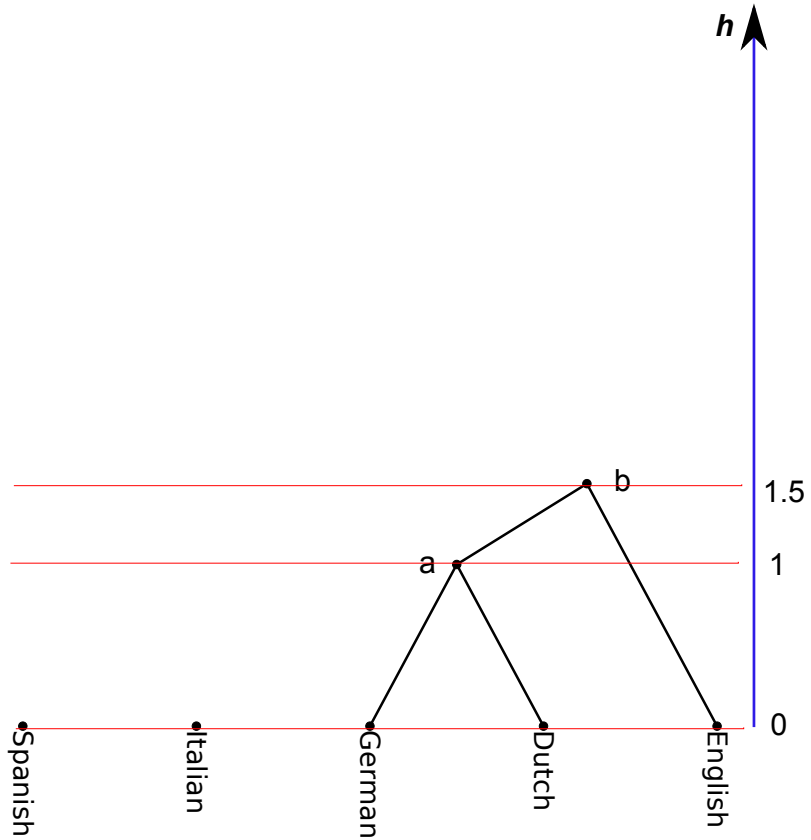
Example

	English	a	Italian
a	**3.0**		
Italian	8.0	8.0	
Spanish	8.0	8.0	3.4



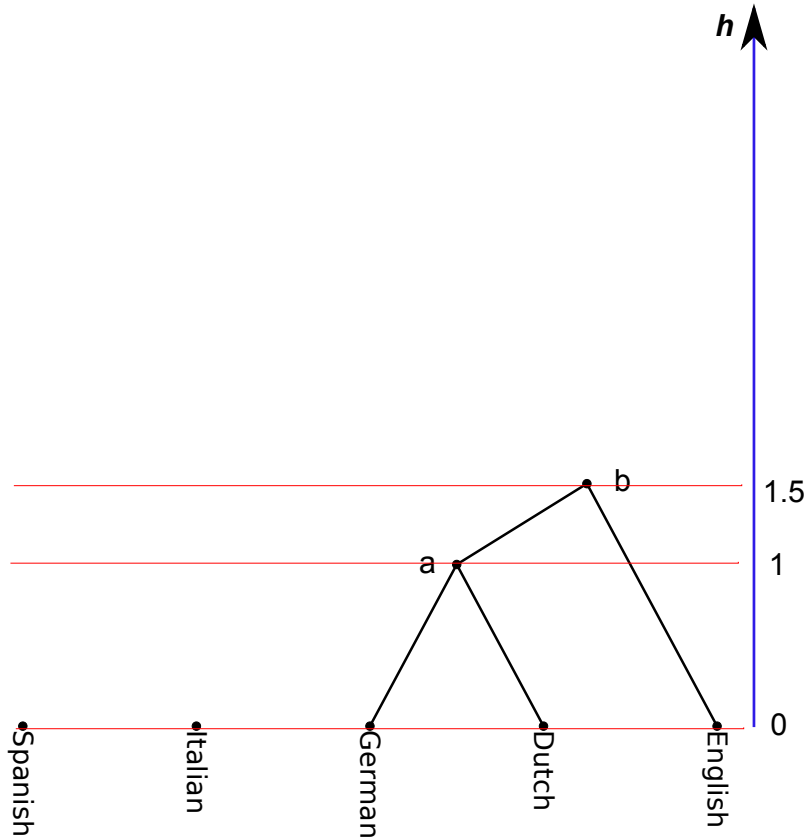
Example

	b	Italian
Italian	8.0	
Spanish	8.0	3.4



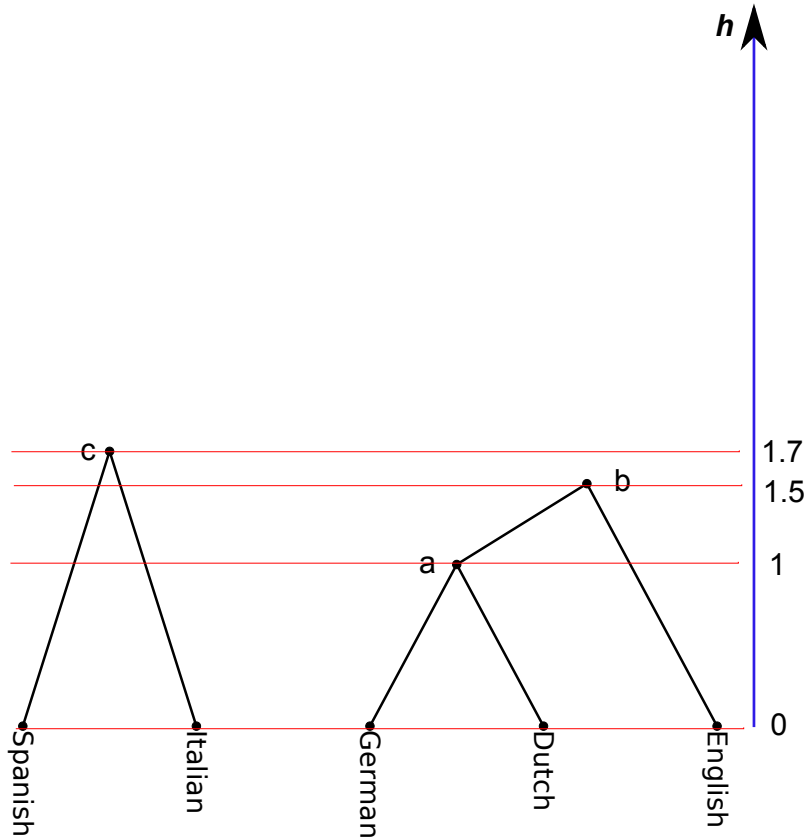
Example

	b	Italian
Italian	8.0	
Spanish	8.0	**3.4**



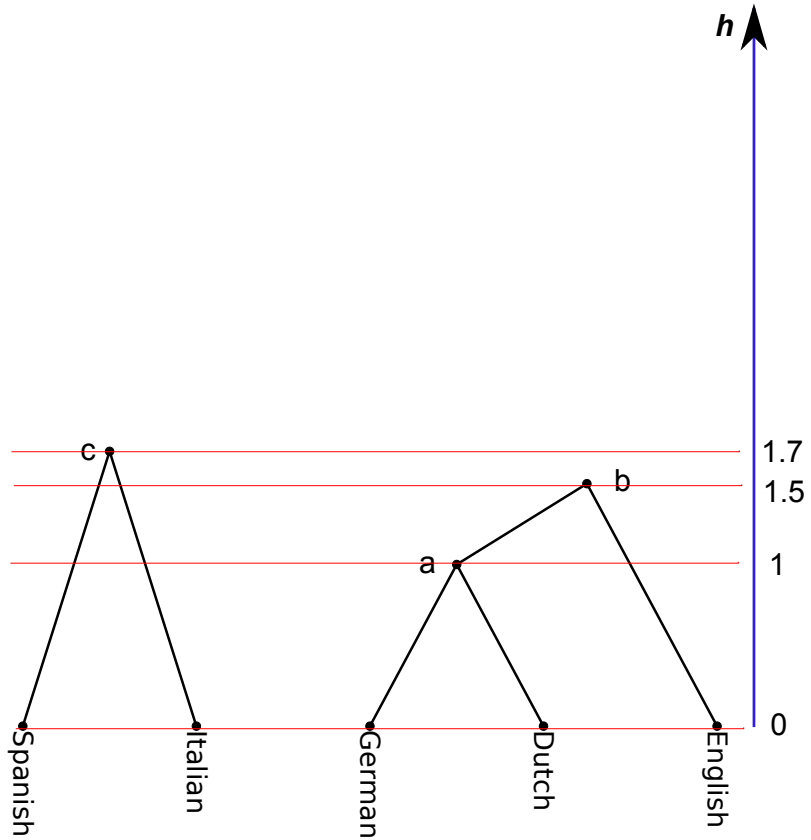
Example

	b
c	8.0

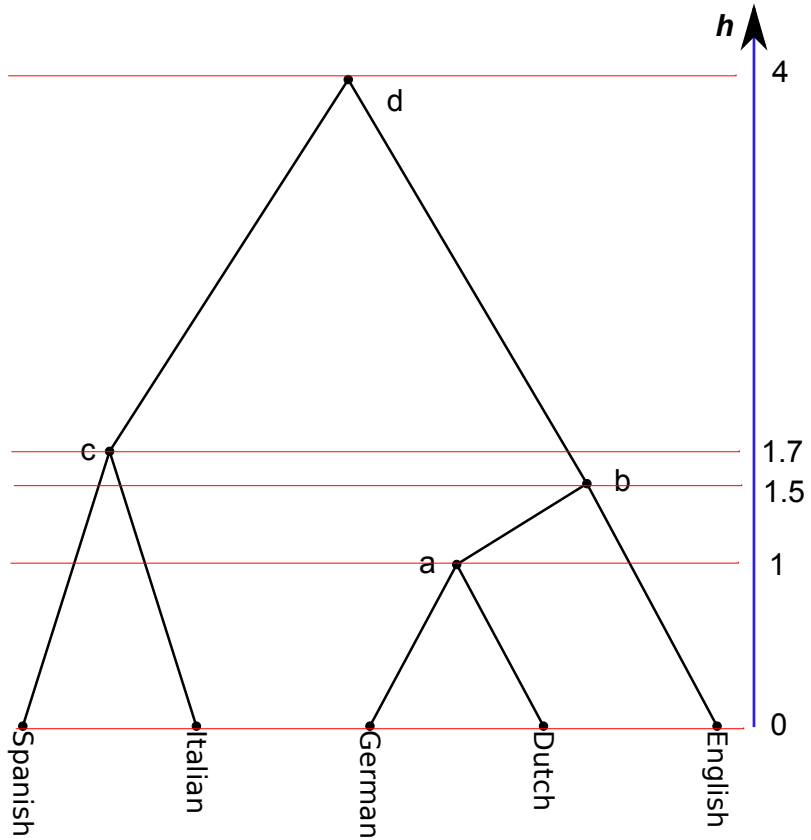


Example

	b
c	8.0



Example



Doing it in R

load library

```
library(phangorn)
```

Doing it in R

define distance matrix

```
taxa <- c('English','Dutch','German','Italian','Spanish')

d <- as.dist(matrix(c(0.0,3.0,3.0,8.0,8.0,
                      3.0,0.0,2.0,8.0,8.0,
                      3.0,2.0,0.0,8.0,8.0,
                      8.0,8.0,8.0,0.0,3.4,
                      8.0,8.0,8.0,3.4,0.0  ),
                    byrow=T,nrow=5,
                    dimnames=list(taxa,taxa)))
```

Doing it in R

```
print(d)
```

```
##         English Dutch German Italian
## Dutch       3.0
## German      3.0   2.0
## Italian     8.0   8.0    8.0
## Spanish     8.0   8.0    8.0     3.4
```

Doing it in R

perform UPGMA

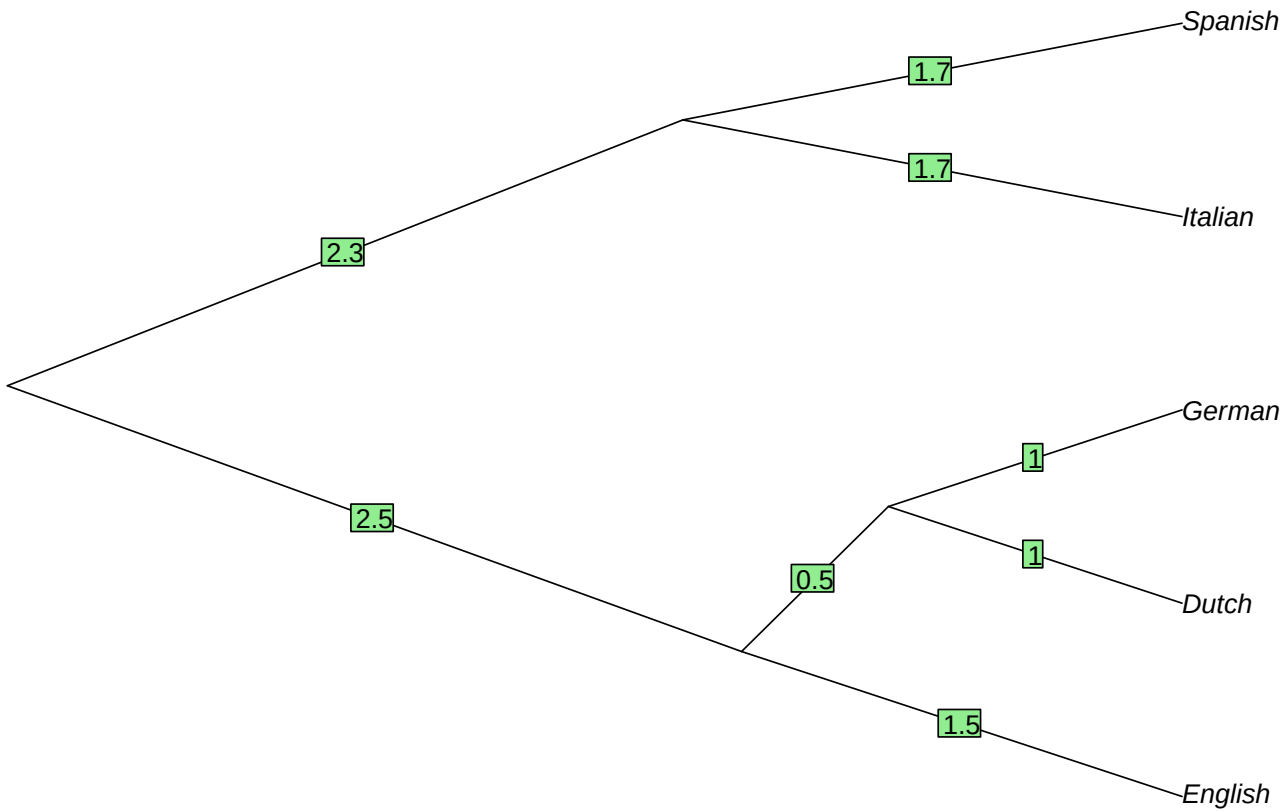
```
upgma.tree <- upgma(d)

cophenetic(upgma.tree)-as.matrix(d)
```

```
##         English Dutch German Italian Spanish
## English       0     0      0       0       0
## Dutch         0     0      0       0       0
## German        0     0      0       0       0
## Italian       0     0      0       0       0
## Spanish       0     0      0       0       0
```

Doing it in R

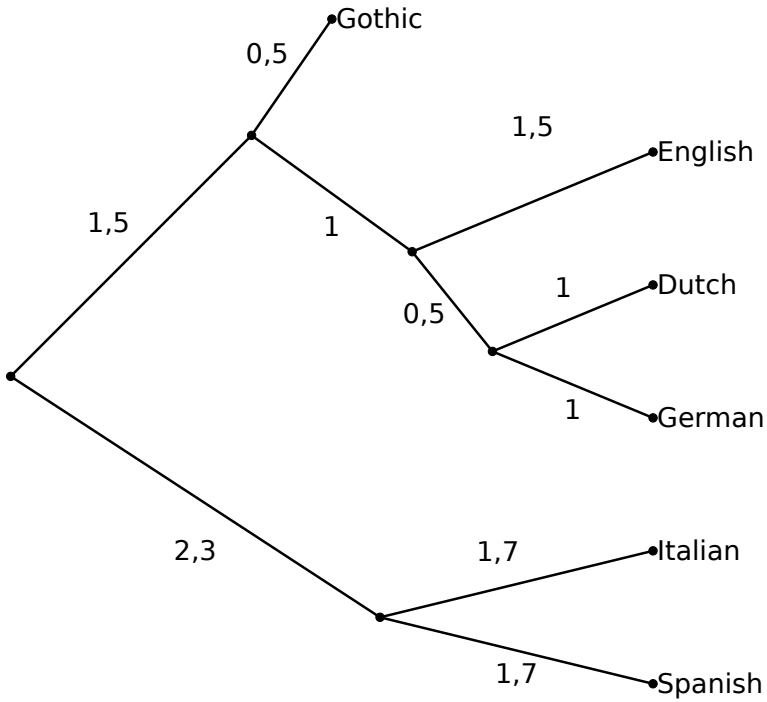
visualize result

```
plot(upgma.tree,type='cladogram')
edgelabels(upgma.tree$edge.length)
```



If distances are not ultra-metric

- UPGMA algorithm also works with distances which are not ultra-metric
- in this case it will **not** recover the correct distances
- tree topology may or may not be recovered



If distances are not ultra-metric

```
taxa <- c('German','Dutch','English',
          'Spanish','Italian','Gothic')
d <- as.dist(matrix(c(0,2,3,8,8,3,
                      2,0,3,8,8,3,
                      3,3,0,8,8,3,
                      8,8,8,0,3.4,6,
                      8,8,8,3.4,0,6,
                      3,3,3,6,6,0),
                    byrow=T,nrow=6,
                    dimnames=list(taxa,taxa)))
```

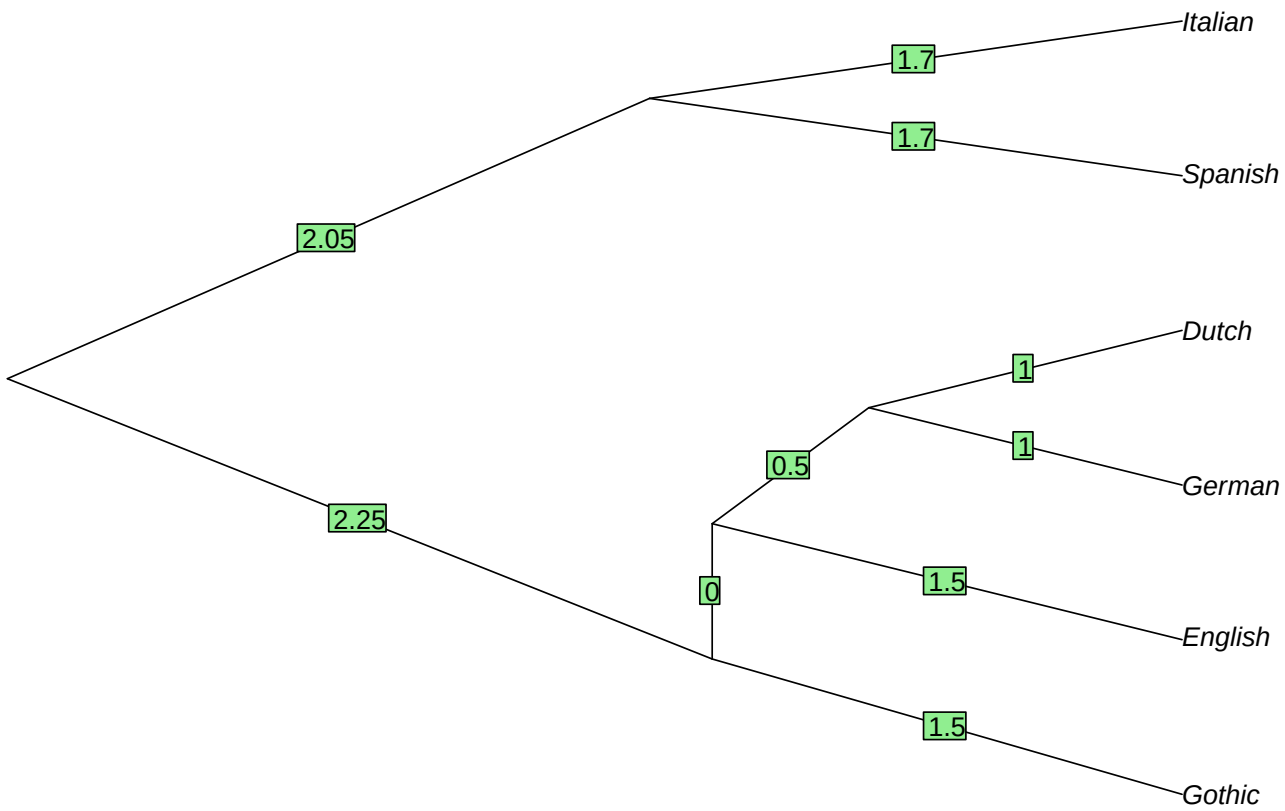
If distances are not ultra-metric

```
upgma.tree <- upgma(d)
cophenetic(upgma.tree)-as.matrix(d)
```

```
##          German Dutch English Spanish Italian Gothic
## German      0.0   0.0     0.0    -0.5    -0.5    0.0
## Dutch       0.0   0.0     0.0    -0.5    -0.5    0.0
## English     0.0   0.0     0.0    -0.5    -0.5    0.0
## Spanish    -0.5  -0.5    -0.5     0.0     0.0    1.5
## Italian    -0.5  -0.5    -0.5     0.0     0.0    1.5
## Gothic      0.0   0.0     0.0     1.5     1.5    0.0
```

If distances are not ultra-metric

```
plot(upgma.tree,type='cladogram')
edgelabels(round(upgma.tree$edge.length,2))
```



Neighbor Joining

- If distances are derived from non-ultrametric distances, we can recover the correct unrooted tree.
- Most commonly used method: **Neighbor Joining** (NJ) (Saitou and Nei, 1987)
- **Neighbors:** Two tips are *neighbors* if the path between them consists of only one node.

Neighbor Joining

- suppose distances between N vertices are given
- auxiliary quantity:

$$\delta(x,y) = (N-4)d(x,y) - \sum_{z \notin \{x,y\}} (d(x,z) + d(y,z))$$

Theorem

If d is derived from a tree $\mathcal{T}$ and $\delta(x,y)$ is minimal, then x and y are neighbors in $\mathcal{T}$.

Proof: See Ewens and Grant (2005), 15.4.

Neigbor Joining

Neighbor Joining algorithm

- **Initialization:**
  - $X \leftarrow$ the set of taxa
  - $V \leftarrow X$
  - $E \leftarrow \emptyset$
- **Iteration:**
  - **while** $|X| > 1$
    - $\delta(x,y) = (|X|-4)d(x,y) - \sum_{z \notin \{x,y\}}(d(x,z)+d(y,z))$
    - $\{i,j\} \leftarrow \arg_{x \in X, y \in X, x \neq y} \min \delta(x,y)$
    - $V \leftarrow V \cup \{\{i,j\}\}$
    - $E \leftarrow E \cup \{(\{i,j\},i),(\{i,j\},j)\}$
    - $l(\{i,j\},i) = \frac{1}{2}d(i,j) + \frac{1}{2(|X|-2)}\sum_{k \in X}(d(i,k)-d(j,k))$
    - $l(\{i,j\},j) = \frac{1}{2}d(i,j) + \frac{1}{2(|X|-2)}\sum_{k \in X}(d(j,k)-d(i,k))$
    - $d(\{i,j\},k) = \frac{1}{2}(d(i,k)+d(j,k)-d(i,j))$
    - $X \leftarrow X \setminus \{i,j\} \cup \{\{i,j\}\}$

Example

d	German	Dutch	English	Spanish	Italian
Dutch	2.0				
English	3.0	3.0			
Spanish	8.0	8.0	8.0		
Italian	8.0	8.0	8.0	3.4	
Gothic	3.0	3.0	3.0	6.0	6.0

Example

d	German	Dutch	English	Spanish	Italian
Dutch	2.0				
English	3.0	3.0			
Spanish	8.0	8.0	8.0		
Italian	8.0	8.0	8.0	3.4	
Gothic	3.0	3.0	3.0	6.0	6.0

δ	German	Dutch	English	Spanish	Italian
Dutch	−40.0				
English	−37.0	−37.0			
Spanish	−25.4	−25.4	−26.4		
Italian	−25.4	−25.4	−26.4	−53.2	
Gothic	−33.0	−33.0	−34.0	−30.4	−30.4

Example

d	German	Dutch	English	Spanish	Italian
Dutch	2.0				
English	3.0	3.0			
Spanish	8.0	8.0	8.0		
Italian	8.0	8.0	8.0	3.4	
Gothic	3.0	3.0	3.0	6.0	6.0

δ	German	Dutch	English	Spanish	Italian
Dutch	−40.0				
English	−37.0	−37.0			
Spanish	−25.4	−25.4	−26.4		
Italian	−25.4	−25.4	−26.4	**−53.2**	
Gothic	−33.0	−33.0	−34.0	−30.4	−30.4

Example

d	German	Dutch	English	Gothic
Dutch	2.0			
English	3.0	3.0		
Gothic	3.0	3.0	3.0	
a	6.3	6.3	6.3	4.3

Example

d	German	Dutch	English	Gothic
Dutch	2.0			
English	3.0	3.0		
Gothic	3.0	3.0	3.0	
a	6.3	6.3	6.3	4.3

δ	German	Dutch	English	Gothic
Dutch	−22.6			
English	−20.6	−20.6		
Gothic	−18.6	−18.6	−19.6	
a	−18.6	−18.6	−19.6	−23.6

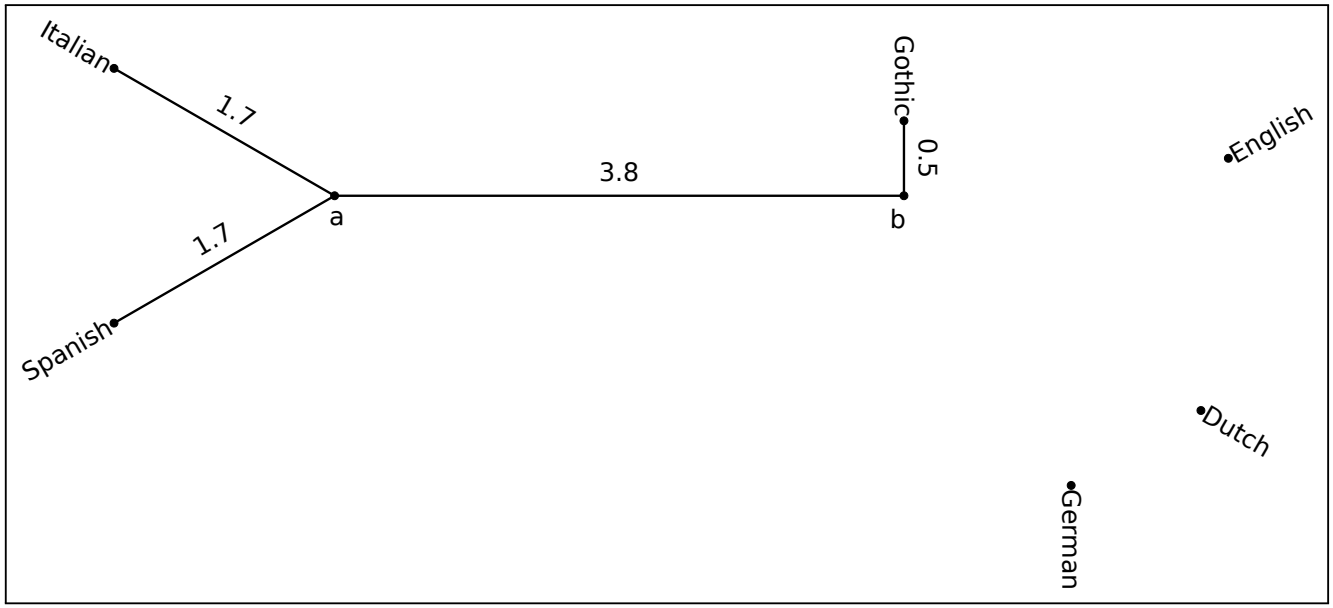
Example

d	German	Dutch	English	Gothic
Dutch	2.0			
English	3.0	3.0		
Gothic	3.0	3.0	3.0	
a	6.3	6.3	6.3	4.3

δ	German	Dutch	English	Gothic
Dutch	−22.6			
English	−20.6	−20.6		
Gothic	−18.6	−18.6	−19.6	
a	−18.6	−18.6	−19.6	**−23.6**

Example

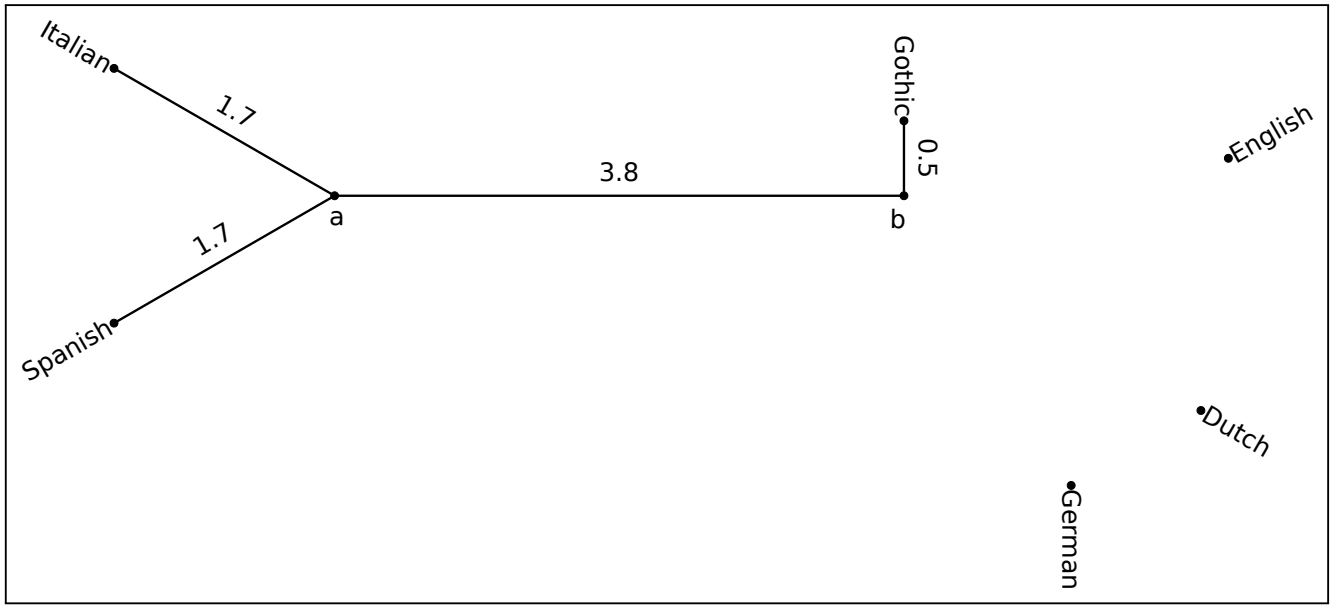
d	German	Dutch	English
Dutch	2.0		
English	3.0	3.0	
b	3.5	2.5	2.5



Example

d	German	Dutch	English
Dutch	2.0		
English	3.0	3.0	
b	3.5	2.5	2.5

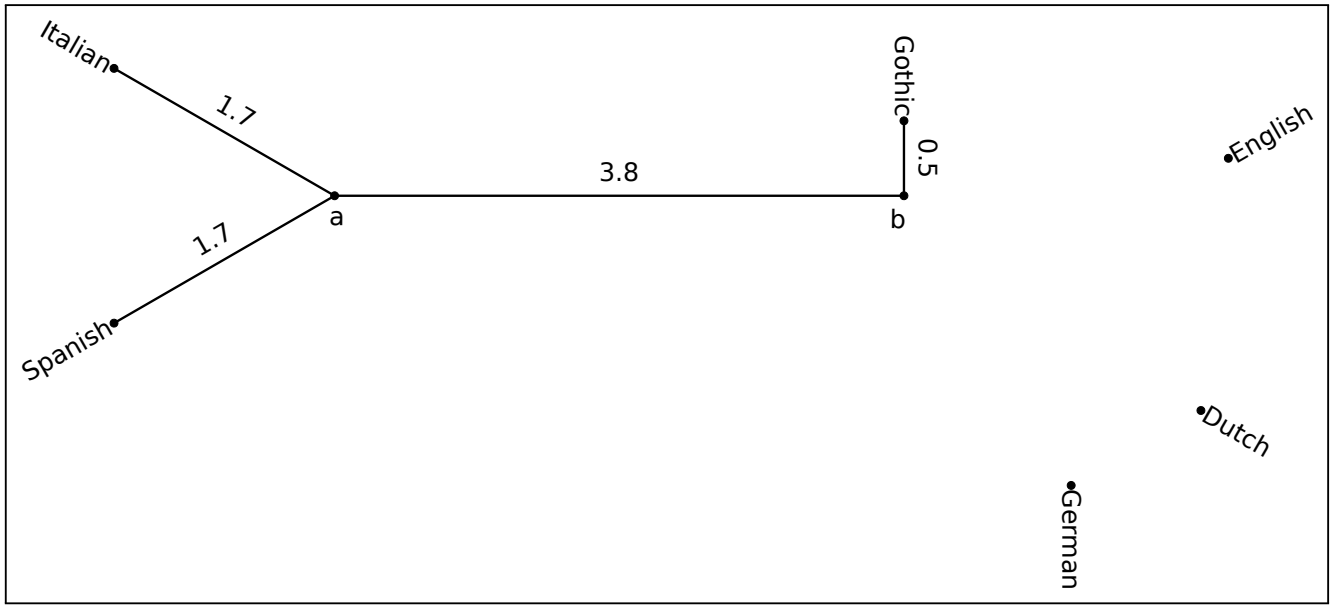
δ	German	Dutch	English
Dutch	−11.0		
English	−10.0	−10.0	
b	−10.0	−10.0	−11.0



Example

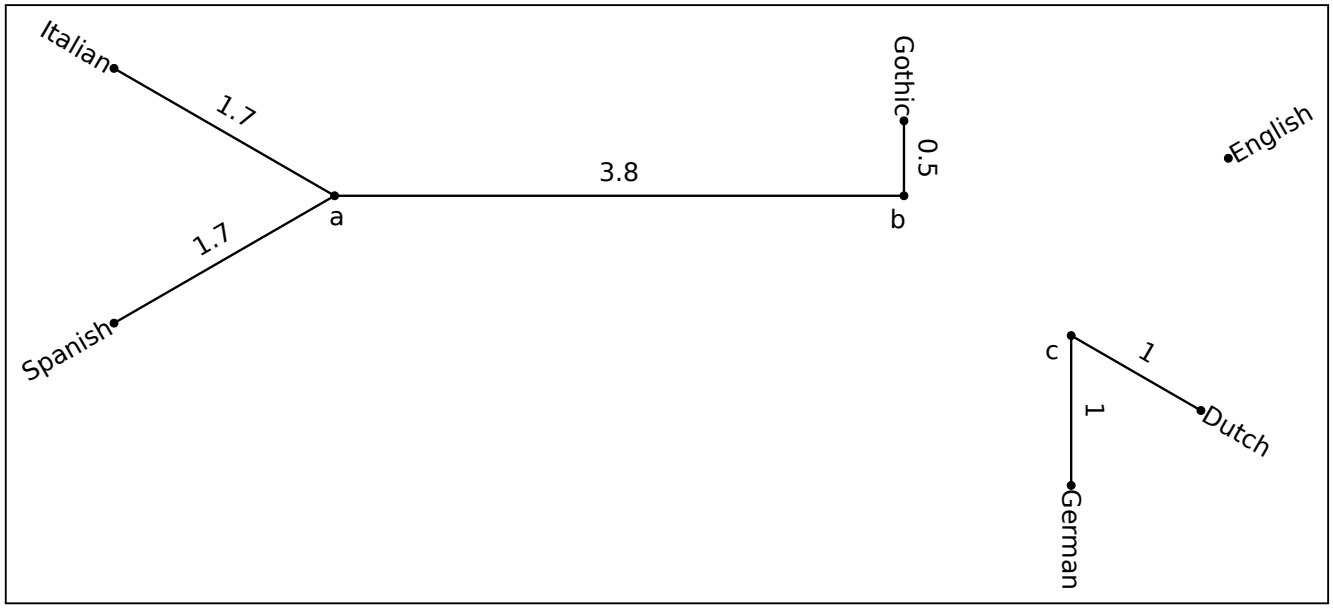
d	German	Dutch	English
Dutch	2.0		
English	3.0	3.0	
b	3.5	2.5	2.5

δ	German	Dutch	English
Dutch	$\mathbf{-11.0}$		
English	-10.0	-10.0	
b	-10.0	-10.0	-11.0



Example

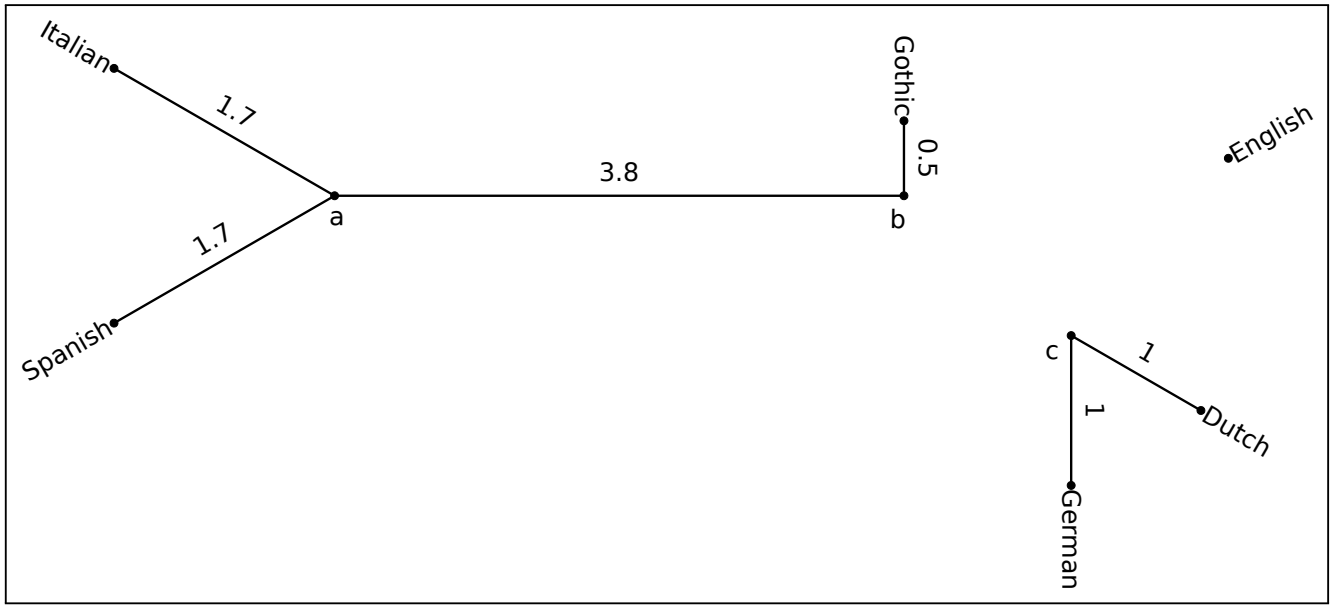
d	English	b
b	2.5	
c	2.0	1.5



Example

d	English	b
b	2.5	
c	2.0	1.5

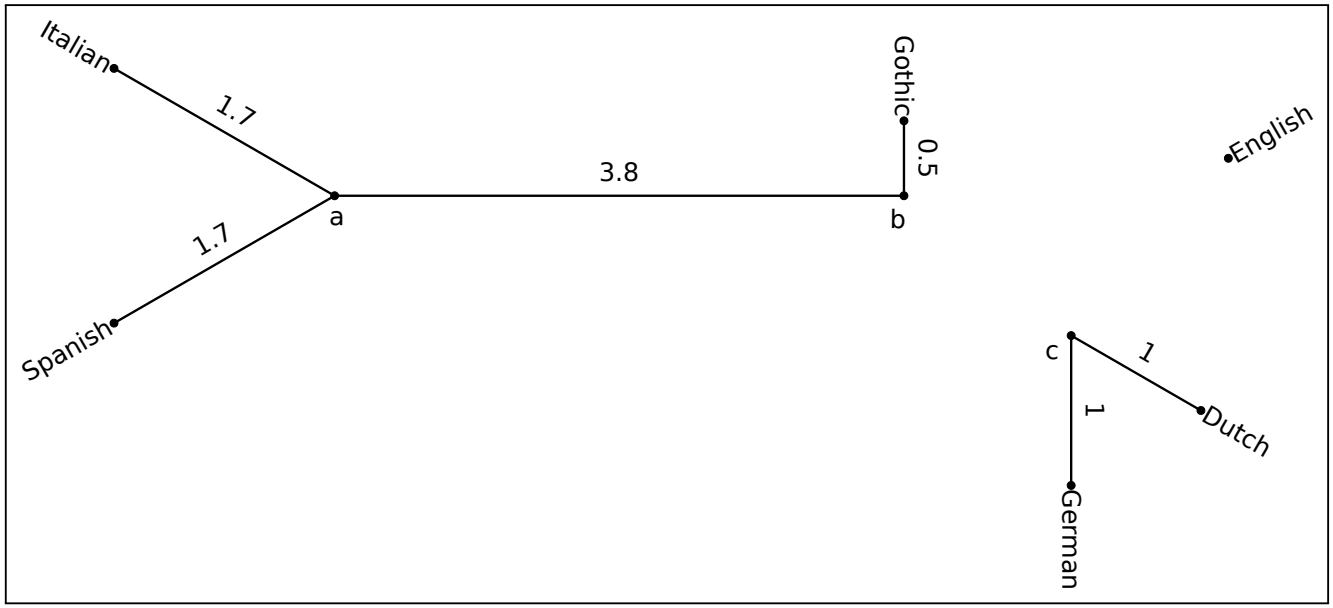
δ	English	b
b	−6.0	
c	−6.0	−6.0



Example

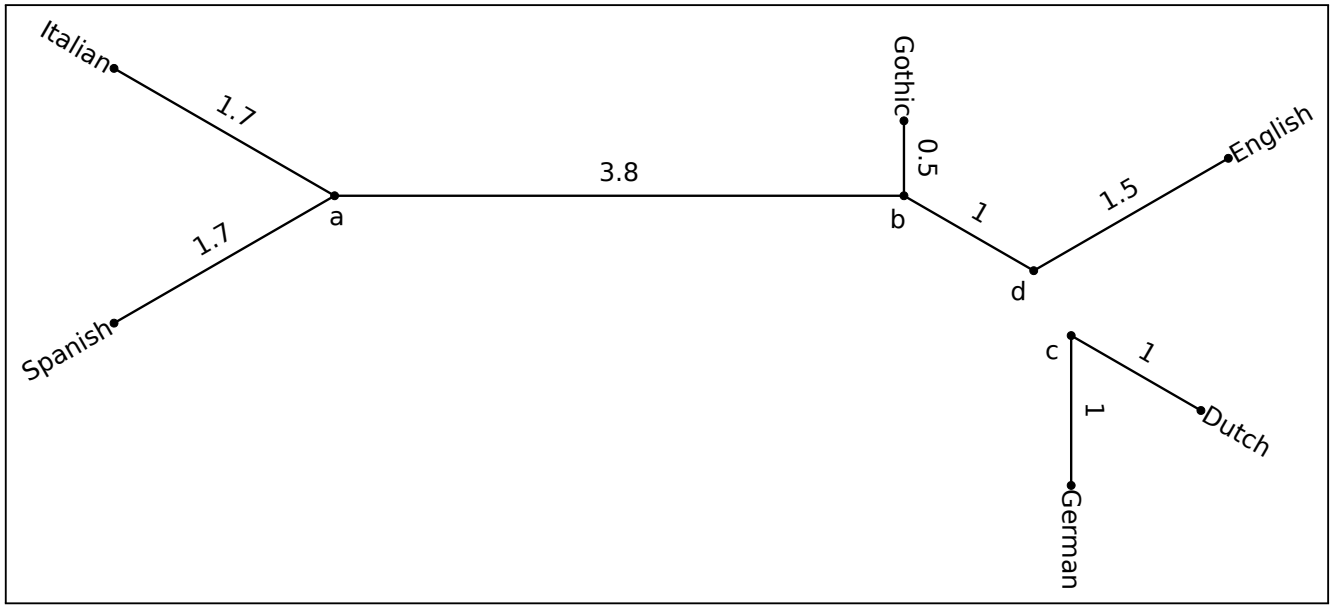
d	English	b
b	2.5	
c	2.0	1.5

δ	English	b
b	**−6.0**	
c	−6.0	−6.0



Example

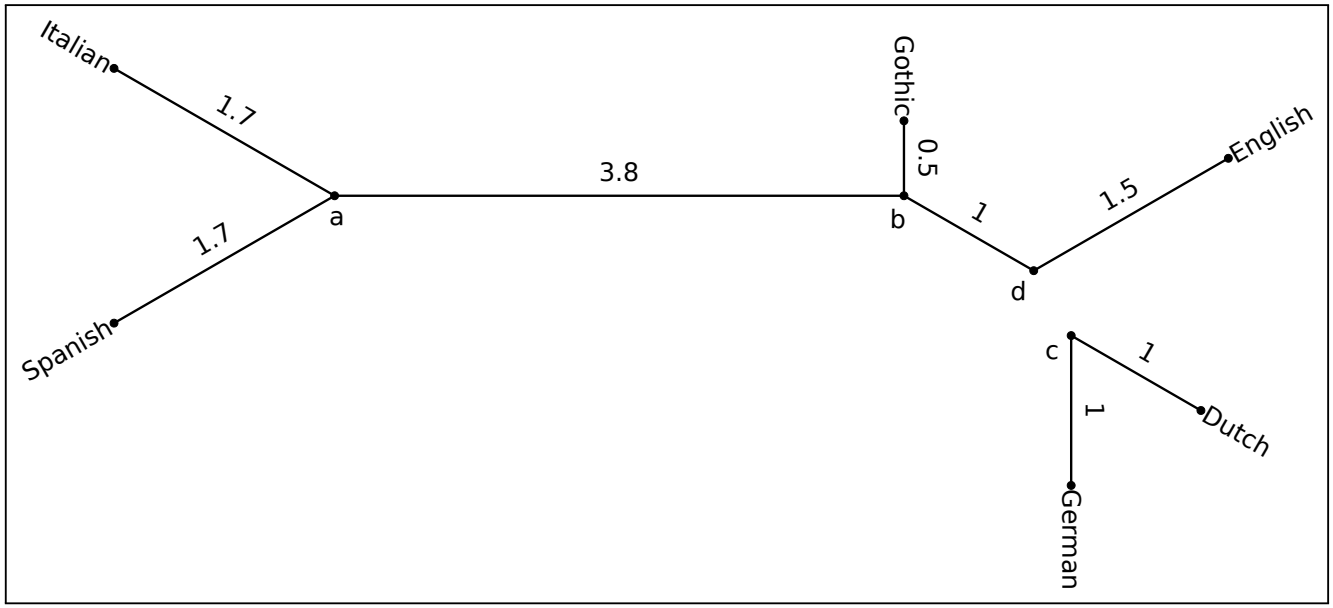
d	**c**
d	0.5



Example

d	**c**
d	0.5

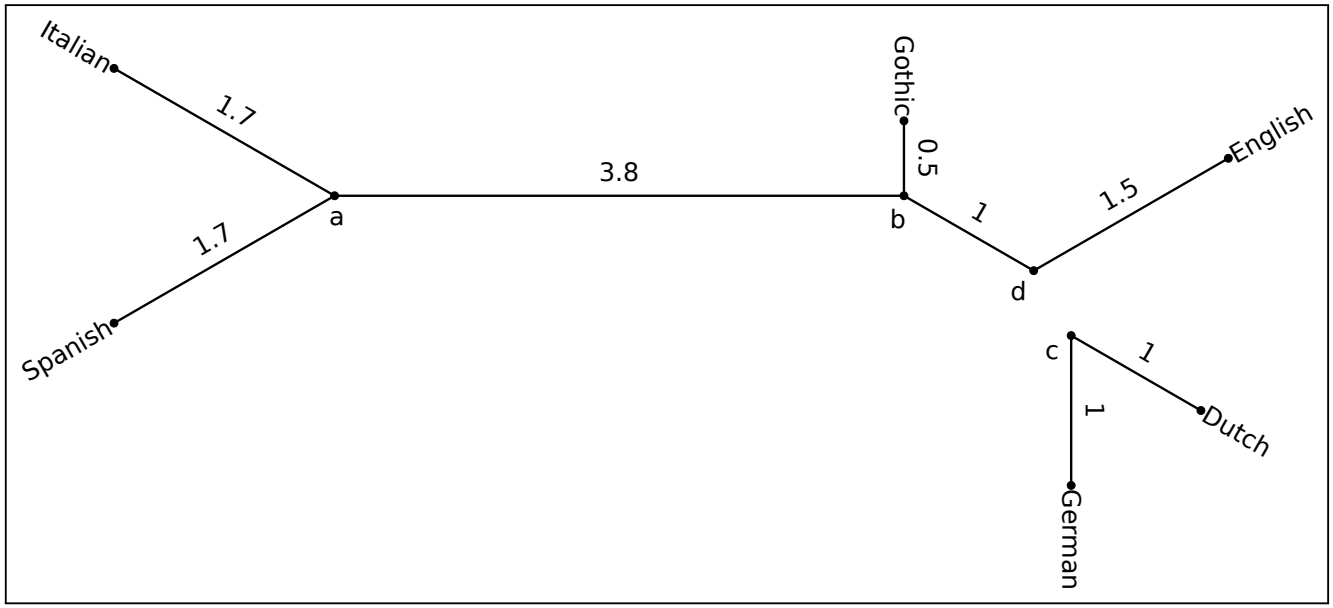
δ	**c**
d	-1.0



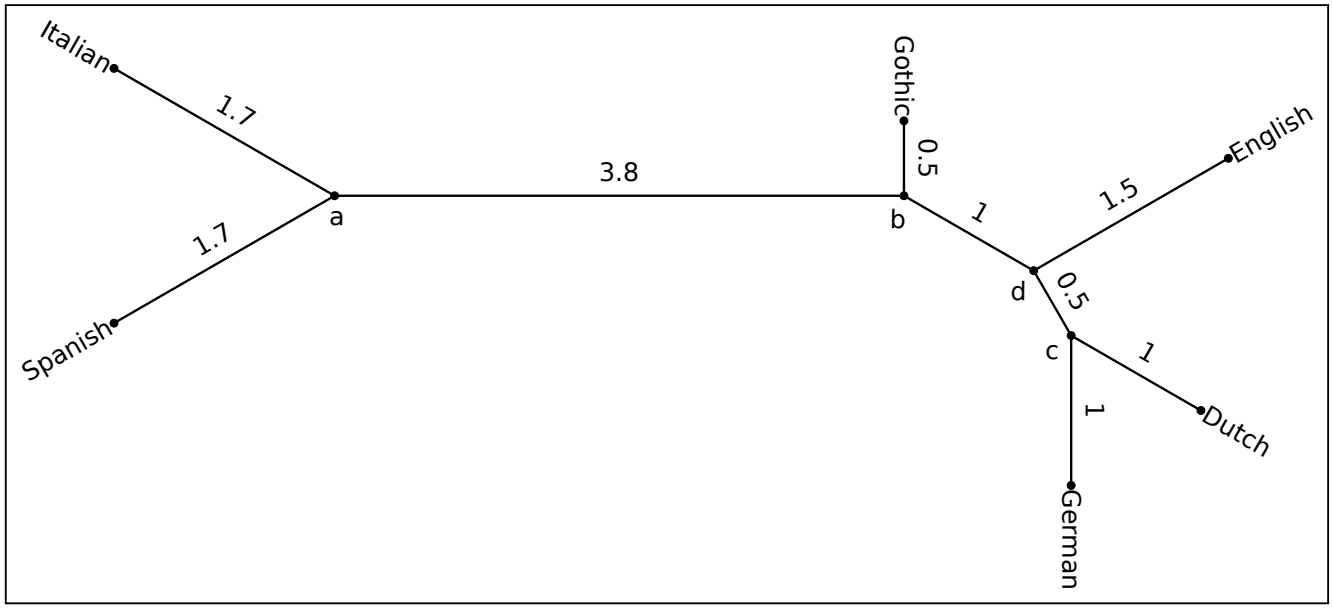
Example

d	**c**
d	0.5

δ	**c**
d	**−1.0**



Example


Italian
1.7
1.7
Spanish
a
3.8
b
Gothic
0.5
1
1.5
English
d
0.5
c
1
Dutch
1
German

Example

- This is the correct **unrooted** tree.
- There is no way to locate the root just from the distance information.
- Generally, NJ will recover the correct unrooted tree if the distances are derived from a tree.

Doing it in R

```
library(phangorn)



taxa <- c('German','Dutch','English',
          'Spanish','Italian','Gothic')
distMatrix <- matrix(c(0,2,3,8,8,3,
                       2,0,3,8,8,3,
                       3,3,0,8,8,3,
                       8,8,8,0,3.4,6,
                       8,8,8,3.4,0,6,
                       3,3,3,6,6,0),
                     byrow=T,nrow=6,
                     dimnames=list(taxa,taxa))
```

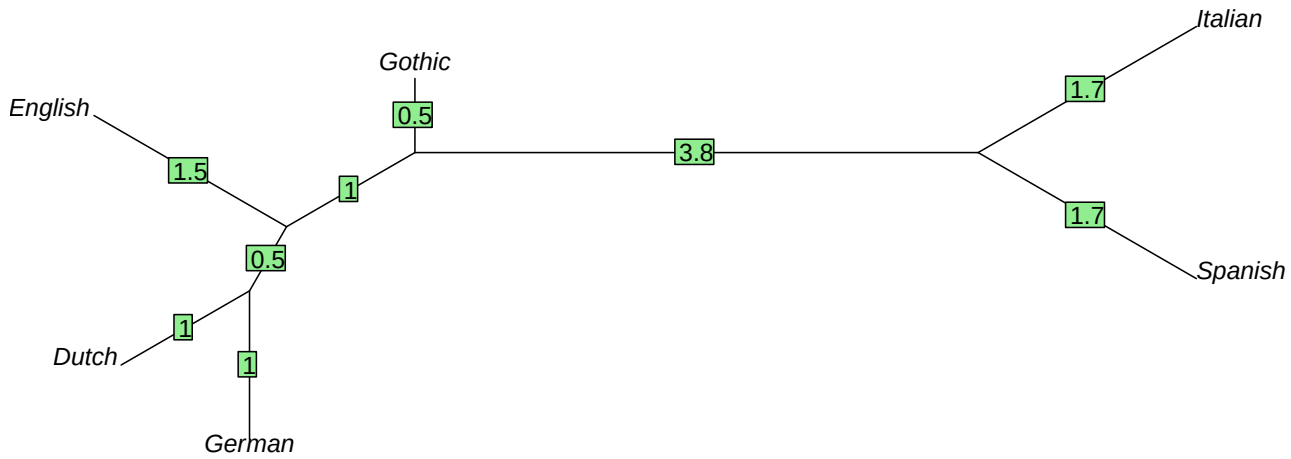
Doing it in R

```
d <- as.dist(distMatrix)

nj.tree <- nj(d)
cophenetic(nj.tree)[taxa,taxa]-as.matrix(d)
```

```
##                German         Dutch       English Spanis
## German   0.000000e+00  0.000000e+00 -4.440892e-16
## Dutch    0.000000e+00  0.000000e+00 -4.440892e-16
## English -4.440892e-16 -4.440892e-16  0.000000e+00
## Spanish  0.000000e+00  0.000000e+00  0.000000e+00
## Italian  0.000000e+00  0.000000e+00  0.000000e+00
## Gothic   8.881784e-16  8.881784e-16  8.881784e-16
##                Gothic
## German  8.881784e-16
## Dutch   8.881784e-16
## English 8.881784e-16
## Spanish 0.000000e+00
## Italian 0.000000e+00
## Gothic  0.000000e+00
```

Doing it in R

```
plot(nj.tree,type='unrooted',use.edge.length=T)
edgelabels(nj.tree$edge.length)
```



Where do we go from there?

- in practice, "true" distances are never known → must be *estimated*
- ideally, we want to know/estimate distances in terms of historical time
- in practice, the best we can hope for are estimates of the *amount of change*
- whether or not historical and evolutionary time are proportional depends in how much *rate of change* varies across lineages

Where do we go from there?

- ultrametric trees only make sense if
  - rate of change is (approximately) constant ("molecular clock assumption")
  - all taxa exist at the same point in time
- as the first condition is rarely fulfilled, this NJ is usually superior to UPGMA
- However: for n taxa,
  - branch lengths in ultrametric tree have $n - 1$ degrees of freedom
  - branch lengths in unrooted (non-ultrametric) tree have $2n - 3$ degrees of freedom

  $\Rightarrow$ UPGMA is less prone to overfitting than NJ
- both UPGMA and NJ are computationally efficient — $\mathcal{O}(n^3)$ for naive implementations

Exercises: Theory

Exercises 15.1–15.5 (pages 535/536) from Ewens and Grant (2005)

Exercises: Programming

- Install the R-packages ape and `phangorn`.
- Type in and run the R-code shown in these slides. Play around with modified distance matrices and different options of the `plot.phylo` command for trees.
- Implement UPGMA and NJ yourself.

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

Nunn, C. L. (2011). *The Comparative Approach in Evolutionary Anthropology and Biology*. The University of Chicago Press, Chicago.

Saitou, N. and M. Nei (1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular biology and evolution*, **4**(4):406–425.