Computational Phylogenetics 1

Quantitative Methods in Historical Linguistics

Dr. Henri Kauhanen / University of Konstanz

16 May 2018

First things first

- Questions from last time?
- Did you manage to make the igraph package work and draw trees?

Reminder

Phylogenetic Inference Problem

Given only information about a set of leaves, what is the most likely tree corresponding to the actual historical development from a common ancestor (root)?

- This is a hard problem!
  - The number of possible trees grows rapidly as the number of leaves grows
  - There is little independent evidence against which to evaluate proposed trees
  - The information we have about the leaves may not be very detailed
  - Plus a number of other problems which we will investigate throughout the course
- Can computational methods help?

Distance-based methods

- We will start with **DISTANCE-BASED METHODS**
- These exploit the idea that closely related languages ought to be **SIMILAR** in some sense.
- E.g. Spanish is more similar to Italian than to English → put Spanish and Italian in a group separate from English.
- We need a notion of the **DISTANCE** between languages.
- There is no unique answer to the question “How distant are two given languages?”
  - Choice of features of the languages to focus on
  - Amount and quality of data
  - Different metrics (distance functions)

The Hamming distance

- The **Hamming distance** $H(x, y)$ counts the number of features two languages x and y differ in.

	def. art.	GenN	gender	tone	future	length[1]
English	1	1	1	0	0	0
Spanish	1	0	1	0	1	0
Italian	1	0	1	0	1	1
Cantonese	0	1	0	1	0	1
Finnish	0	1	0	0	0	1

- E.g. $H(\text{English}, \text{Spanish}) = 2$ and $H(\text{Cantonese}, \text{Finnish}) = 1$

[1] def. art. = definite article, GenN = genitive–noun order, gender = grammatical gender, tone = phonological tone, future = morphological future tense, length = contrastive length in phonology

The Hamming distance

	def. art.	GenN	gender	tone	future	length
English	1	1	1	0	0	0
Spanish	1	0	1	0	1	0
Italian	1	0	1	0	1	1
Cantonese	0	1	0	1	0	1
Finnish	0	1	0	0	0	1

- Sometimes the distance is normalized by dividing H by the number of features:
  - $H(\text{English}, \text{Spanish}) = 2/6 \approx 0.333$
  - $H(\text{Cantonese}, \text{Finnish}) = 1/6 \approx 0.167$
- Then $H(x, y)$ ranges between 0 and 1

The Hamming distance

- Assumptions:
  - Each language is described by a vector of the same features (and in the same order) — the vectors are **ALIGNED**
  - Although I've used binary features (0 or 1), they need not be
- Properties:
  - $H(x, x) = 0$
  - $H(x, y) = H(y, x)$
  - $H(x, z) \leq H(x, y) + H(y, z)$
- Notes:
  - H is highly sensitive to the number of features considered!
  - The more features (= more data), the better the distance estimates

The Hamming distance

- In R, you can get the Hamming distance between two vectors as follows:

```
English <- c(1, 1, 1, 0, 0, 0)
Spanish <- c(1, 0, 1, 0, 1, 0)
sum(English != Spanish)
```

- Try this now!
- To get the normalized version:

```
sum(English != Spanish)/length(English)
```

From a feature matrix to a distance matrix

- Recall the feature matrix:

	def. art.	GenN	gender	tone	future	length
English	1	1	1	0	0	0
Spanish	1	0	1	0	1	0
Italian	1	0	1	0	1	1
Cantonese	0	1	0	1	0	1
Finnish	0	1	0	0	0	1

- We can calculate all **PAIRWISE DISTANCES** (between each pair of languages) and put them in another matrix

From a feature matrix to a distance matrix

- We get a **DISTANCE MATRIX**:

	English	Spanish	Italian	Cant.	Finnish
English					
Spanish	*				
Italian	*	*			
Cantonese	*	*	*		
Finnish	*	*	*	*	

- Why do we only need to fill in the cells marked with *?

From a feature matrix to a distance matrix

	English	Spanish	Italian	Cant.	Finnish
English					
Spanish	0.333				
Italian	0.500	0.167			
Cantonese	0.667	1.000	0.833		
Finnish	0.500	0.833	0.667	0.167	

- Given n languages, one needs to calculate

$$\frac{n(n-1)}{2}$$

  distances
- Better let the computer do this for us…

From a feature matrix to a distance matrix

- In R, we can use the rdist package:

```
library(rdist)
English <- c(1, 1, 1, 0, 0, 0)
Spanish <- c(1, 0, 1, 0, 1, 0)
Italian <- c(1, 0, 1, 0, 1, 1)
Cantonese <- c(0, 1, 0, 1, 0, 1)
Finnish <- c(0, 1, 0, 0, 0, 1)
feature_mtx <- rbind(English, Spanish, Italian,
                     Cantonese, Finnish)
dist_mtx <- rdist(feature_mtx, metric="hamming")
```

- Type `feature_mtx` and `dist_mtx` to see the feature and distance matrices.

Guessing a tree

- Now we have a distance matrix for a set of languages
- These are the leaves of the tree
- But how to construct the tree from this information?
  - We only have distances between the leaves, but no distances between the leaves and their ancestors!
- First (and simplest) method we're going to try: **UPGMA** (Unweighted Pair Group Method with Arithmetic Mean).

UPGMA: Step 1

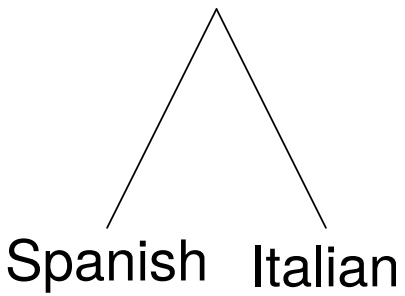
- First, we locate the language pair that has the smallest distance
- If this pair is not unique (as is the case here), we pick one pair at random (it doesn't matter which one)

	English	Spanish	Italian	Cant.	Finnish
English					
Spanish	0.333				
Italian	0.500	0.167			
Cantonese	0.667	1.000	0.833		
Finnish	0.500	0.833	0.667	0.167	

- In this case, we pick Spanish and Italian

UPGMA: Step 1

- We then form the following subgrouping:



- Next, we combine Spanish and Italian in our distance matrix, as if it was just one language. Let's call it "SI".

UPGMA: Step 1

	English	SI	Cantonese	Finnish
English				
SI	???			
Cantonese	0.667	???		
Finnish	0.500	???	0.167	

- But how do we know what the distance between the subgroup SI and the other languages is?
- Assume it is the **AVERAGE** distance to S and I

UPGMA: Step 1

- Since $H(E,S) = 0.333$ and $H(E,I) = 0.500$, the distance of English (E) to the combined group SI will be

$$H(E,SI) = \frac{0.333 + 0.500}{2} = 0.4165$$

- We get:

	English	SI	Cantonese	Finnish
English				
SI	0.4165			
Cantonese	0.667	???		
Finnish	0.500	???	0.167	

UPGMA: Step 1

- Exercise: do the same with Cantonese and Finnish

	English	Spanish	Italian	Cant.	Finnish
English					
Spanish	0.333				
Italian	0.500	0.167			
Cantonese	0.667	1.000	0.833		
Finnish	0.500	0.833	0.667	0.167	

	English	SI	Cantonese	Finnish
English				
SI	0.4165			
Cantonese	0.667	???		
Finnish	0.500	???	0.167	

UPGMA: Step 1

- You should get:

	English	SI	Cantonese	Finnish
English				
SI	0.4165			
Cantonese	0.667	0.9165		
Finnish	0.500	0.750	0.167	

UPGMA: Step 2

- We now continue in exactly the same manner
- Using the **NEW** distance matrix, pick the pair with the smallest distance:

	English	SI	Cantonese	Finnish
English				
SI	0.4165			
Cantonese	0.667	0.9165		
Finnish	0.500	0.750	0.167	

UPGMA: Step 2

- Now we have two little subtrees:

- Next (just as before) we combine Cantonese and Finnish together in our distance matrix; call the combined group "CF".
- And calculate average distances to this group.

UPGMA: Step 2

- We get the distance of English to CF just as before:

$$H(E, CF) = \frac{H(E, C) + H(E, F)}{2} = \frac{0.667 + 0.500}{2} = 0.5835$$

- We now have:

	English	SI	CF
English			
SI	0.4165		
CF	0.5835	???	

UPGMA: Step 2

- To calculate the distance between the subgroups SI and CF, we average over **ALL** distances $H(x, y)$ where x is from SI and y is from CF:

$$H(SI, CF) = \frac{H(S, C) + H(S, F) + H(I, C) + H(I, F)}{4}$$
$$= \frac{1.000 + 0.833 + 0.833 + 0.667}{4} = 0.83325$$

- We now have the distance matrix:

	English	SI	CF
English			
SI	0.4165		
CF	0.5835	0.83325	

UPGMA: Step 3

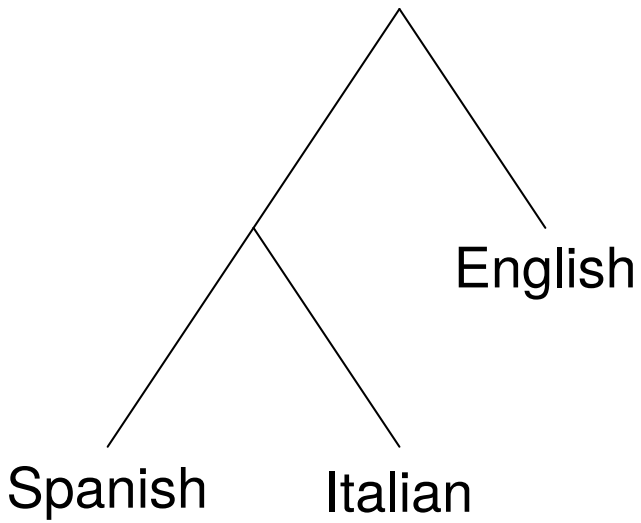
- Continue in exactly the same manner
- Using the newest matrix, pick the pair with the smallest distance:

	English	SI	CF
English			
SI	0.4165		
CF	0.5835	0.83325	

- This means we will next join English with the SI group.

UPGMA: Step 3

- English joined with SI:



- We now form the group ESI and calculate its distance to CF.

UPGMA: Step 4

- To calculate $H(ESI, CF)$, we again take all distances $H(x, y)$, where x is from ESI and y from CF, and take their average. In other words:

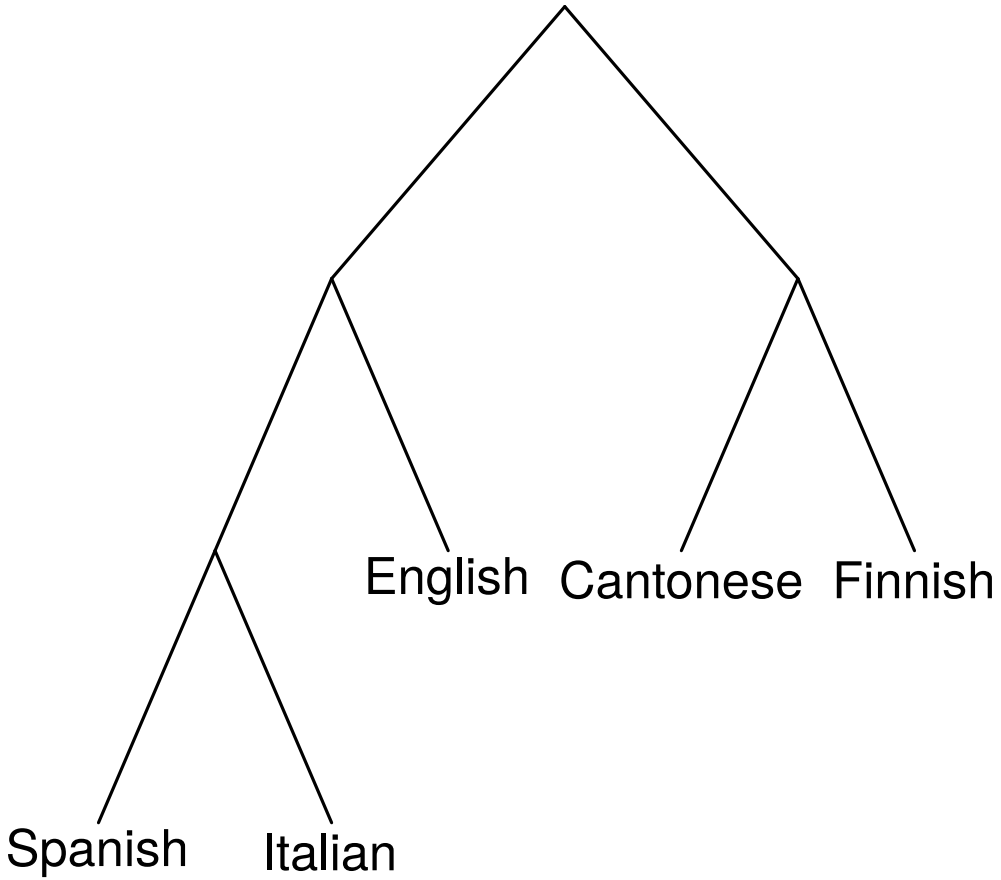
$$\begin{aligned} H(ESI, CF) &= \frac{H(E,C) + H(E,F) + H(S,C) + H(S,F) + H(I,C) + H(I,F)}{6} \\ &= \frac{0.667 + 0.500 + 1.000 + 0.833 + 0.833 + 0.667}{6} \\ &= 0.75 \end{aligned}$$

- Our distance matrix now looks like this:

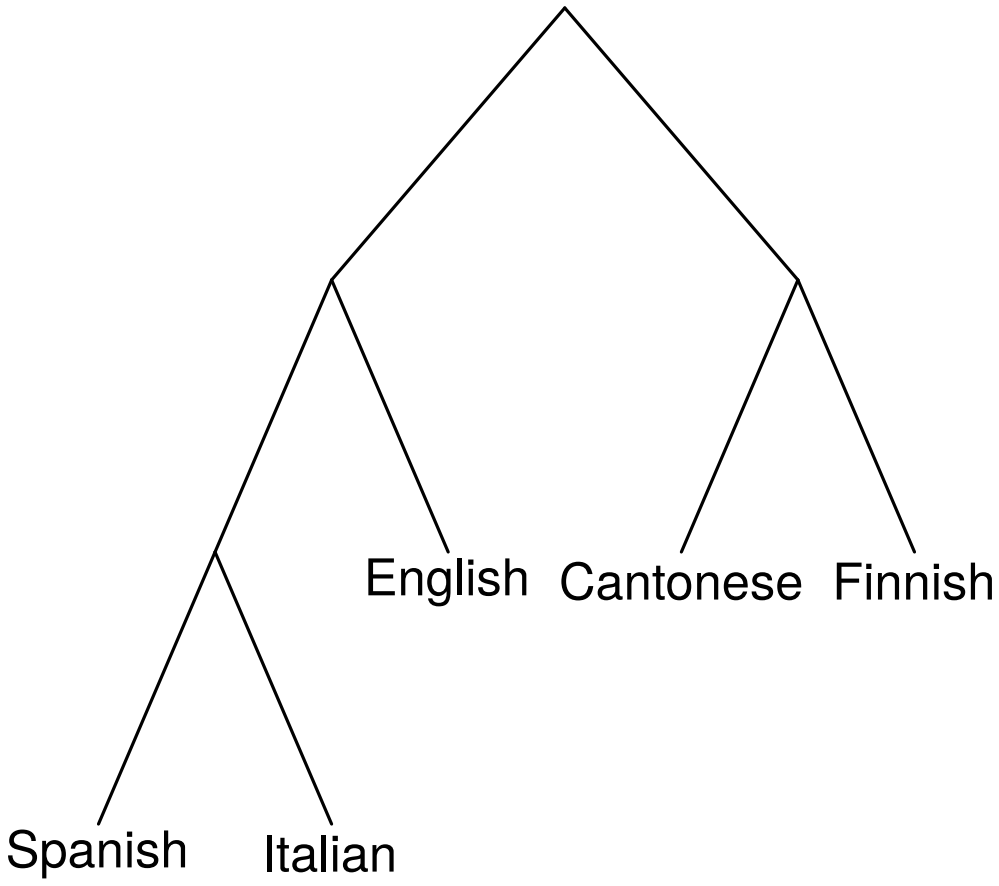
	ESI	CF
ESI		
CF	0.75	

UPGMA: Step 4

- Trivially ESI and CF now have the shortest distance (since they are the only remaining groups), so we join them:



UPGMA: Completion



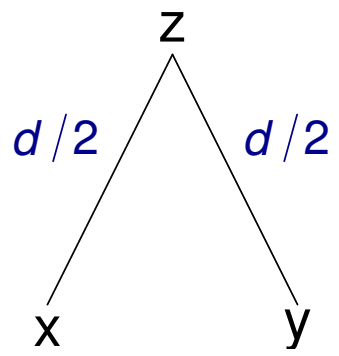
- This completes the construction. The result is not entirely unreasonable! Basically, we have found groupings for Romance, Indo-European and “Other”.

Metric trees

- UPGMA classifies languages into **HIERARCHICAL CLUSTERS**. Each cluster has two elements (either leaves or internal nodes).
- But it does more: the algorithm actually implicitly assigns a distance to each edge of the tree.
- This makes it possible to talk about how much a language changes as we move from one node of the tree to another.
- A tree whose edges bear such distance information is known as a **METRIC TREE**.
- Non-metric trees are **TOPOLOGICAL**. (All the trees we've seen so far have been topological trees.)

Metric trees

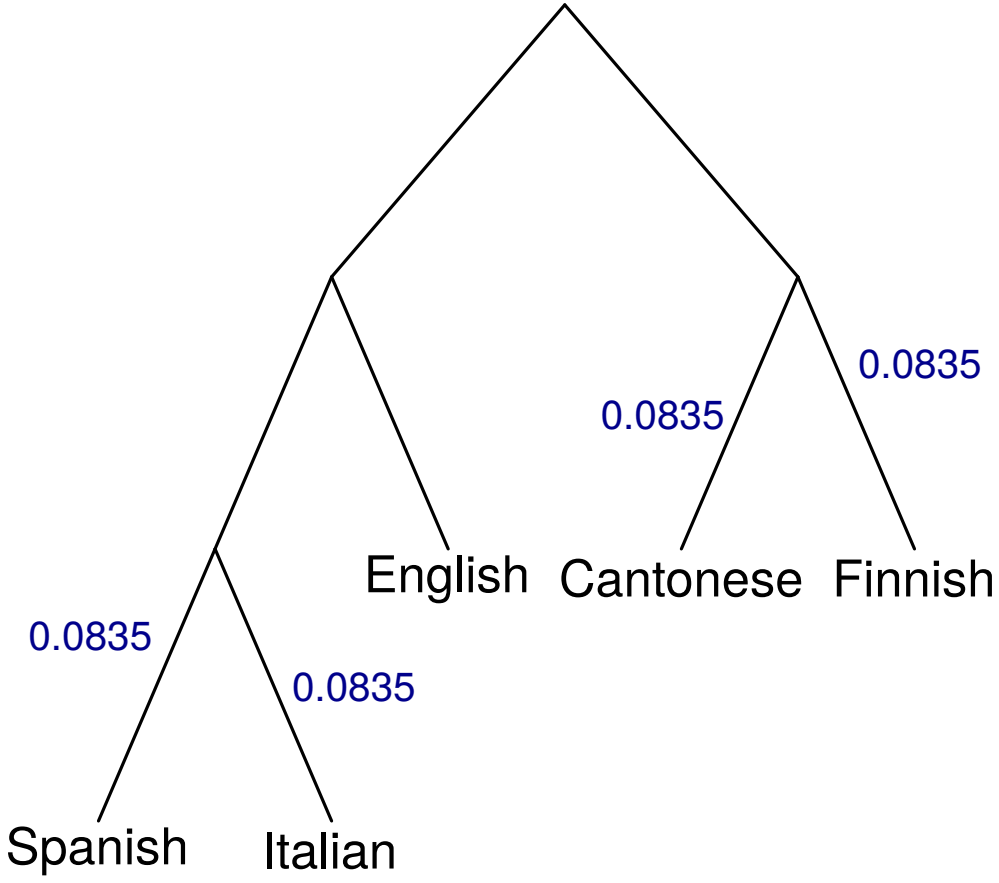
- Here's how UPGMA assigns distances to edges.
- The basic idea: if two leaves x and y with a common ancestor z are at a distance $H(x, y) = d$, then the ancestor's distance to each leaf ought to be $d/2$:



- Why? Because in order to pass from x to y, you first have to "undo" the changes that produced x from z, and then do the changes that produced y from z.
- The overall distance adds up:
  $H(x, y) = H(x, z) + H(z, y) = d/2 + d/2 = d$.

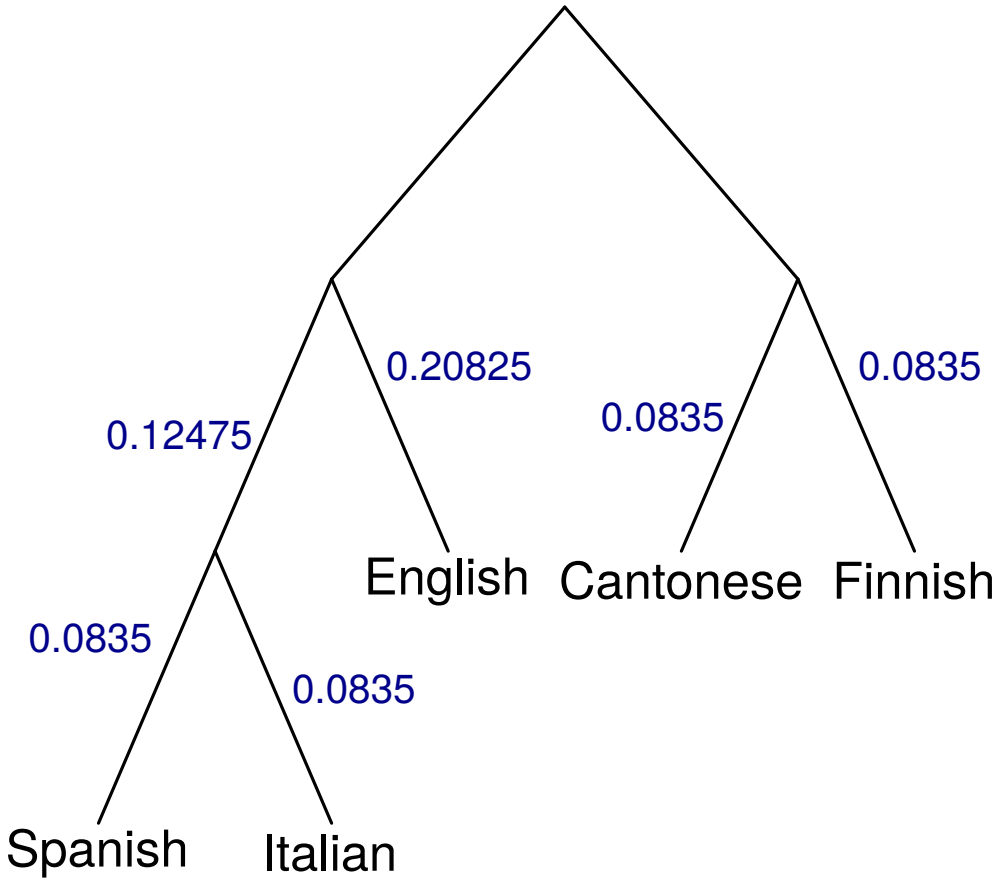
Metric trees

- Since $H(S, I) = 0.167$ and $H(C, F) = 0.167$, and since $0.167/2 = 0.0835$, we have:



Metric trees

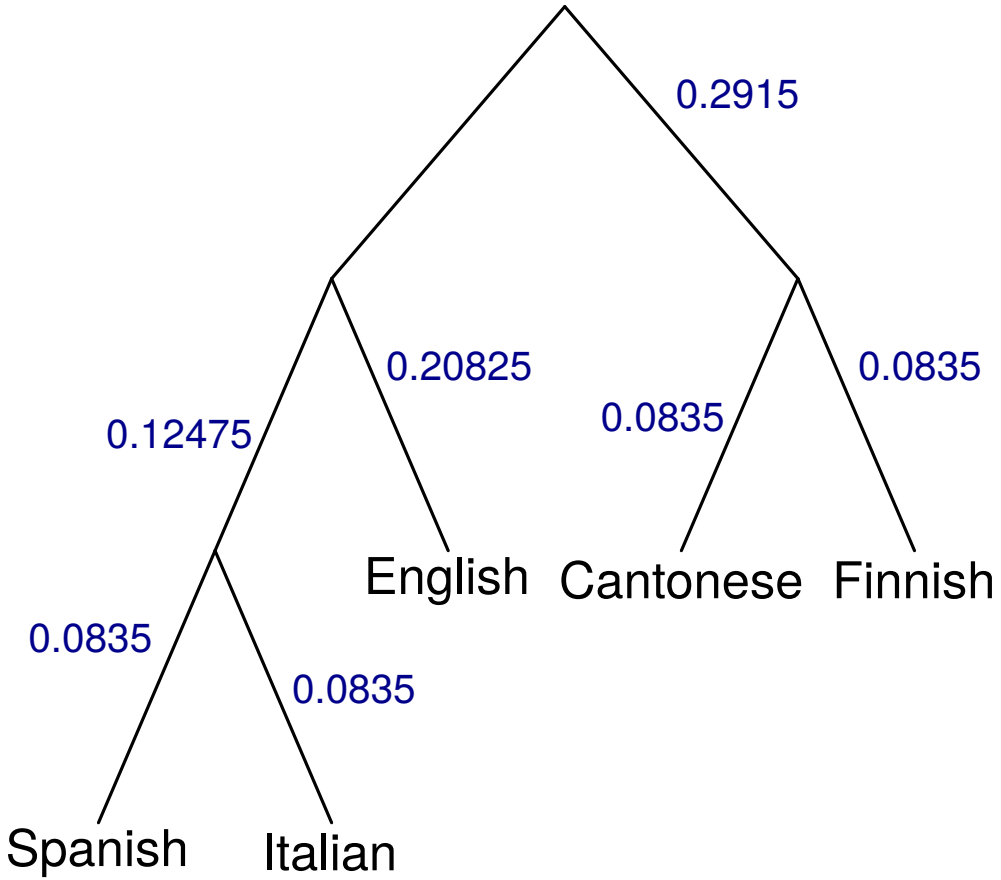
- Since $H(E, SI) = 0.4165$ and $0.4165/2 = 0.20825$, we have:



- Note that $0.0835 + 0.12475 = 0.20825$.

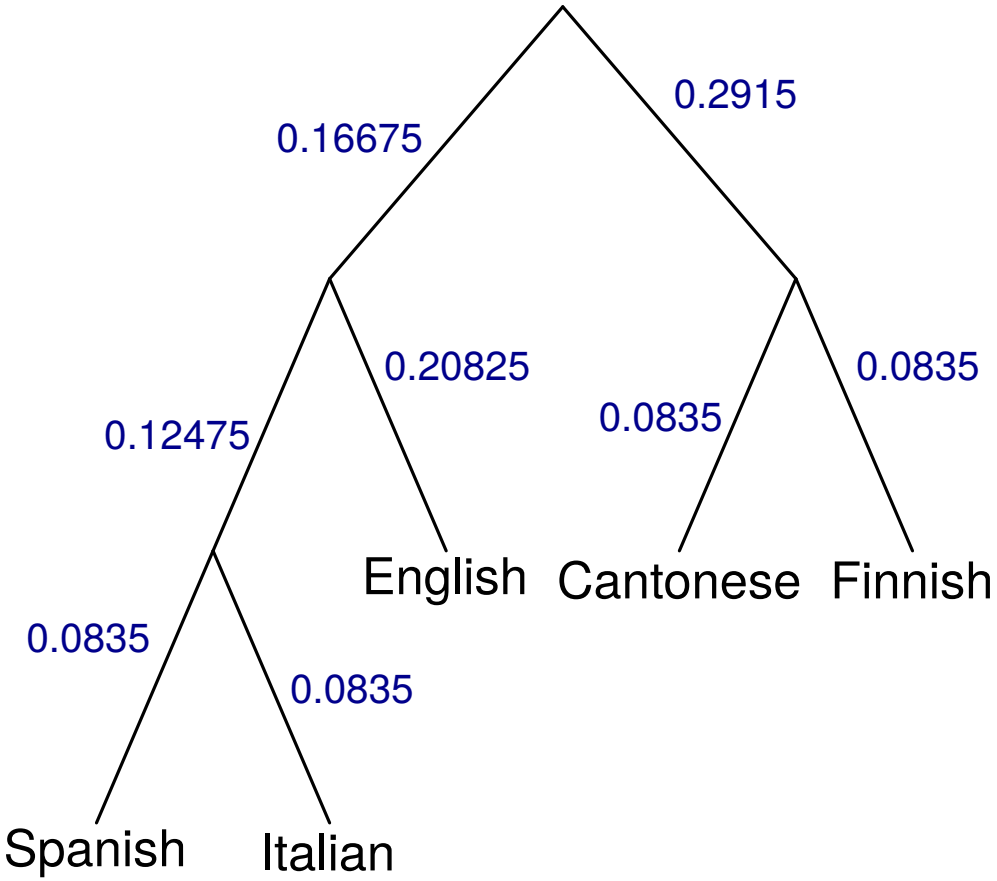
Metric trees

- Since $H(ESI, CF) = 0.75$ and $0.75/2 = 0.375$ and $0.375 - 0.0835 = 0.2915$, we get:



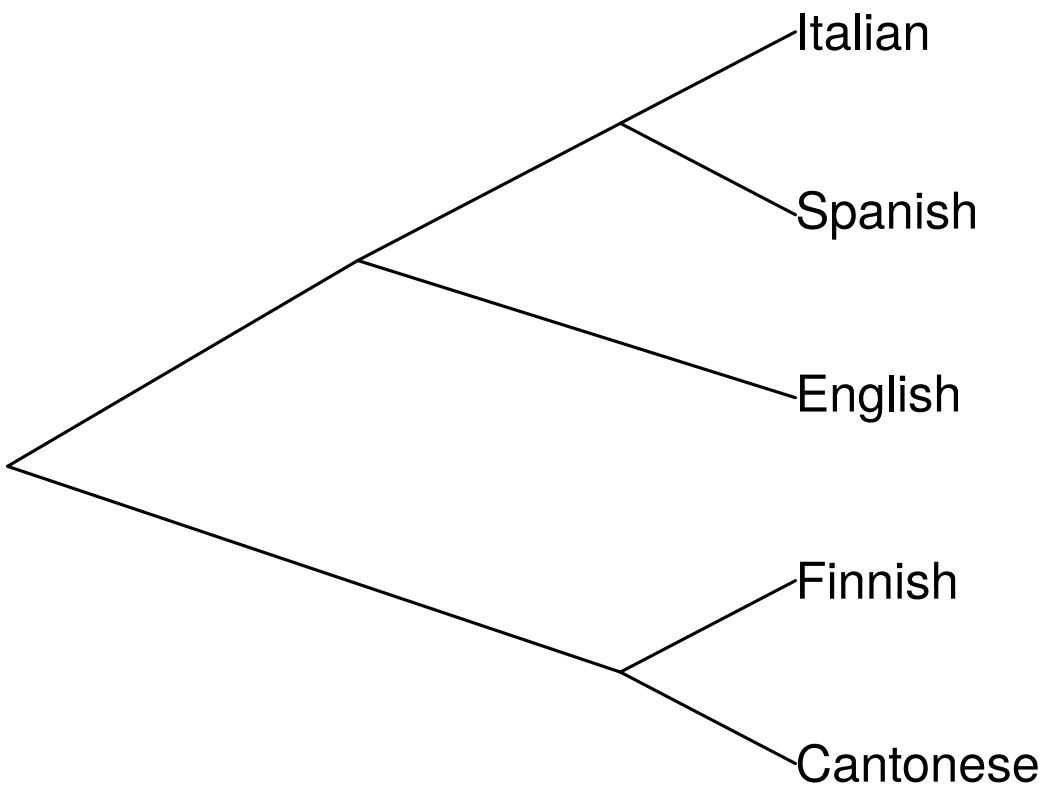
Metric trees

- And since 0.375 – 0.20825 = 0.16675, we finally have:



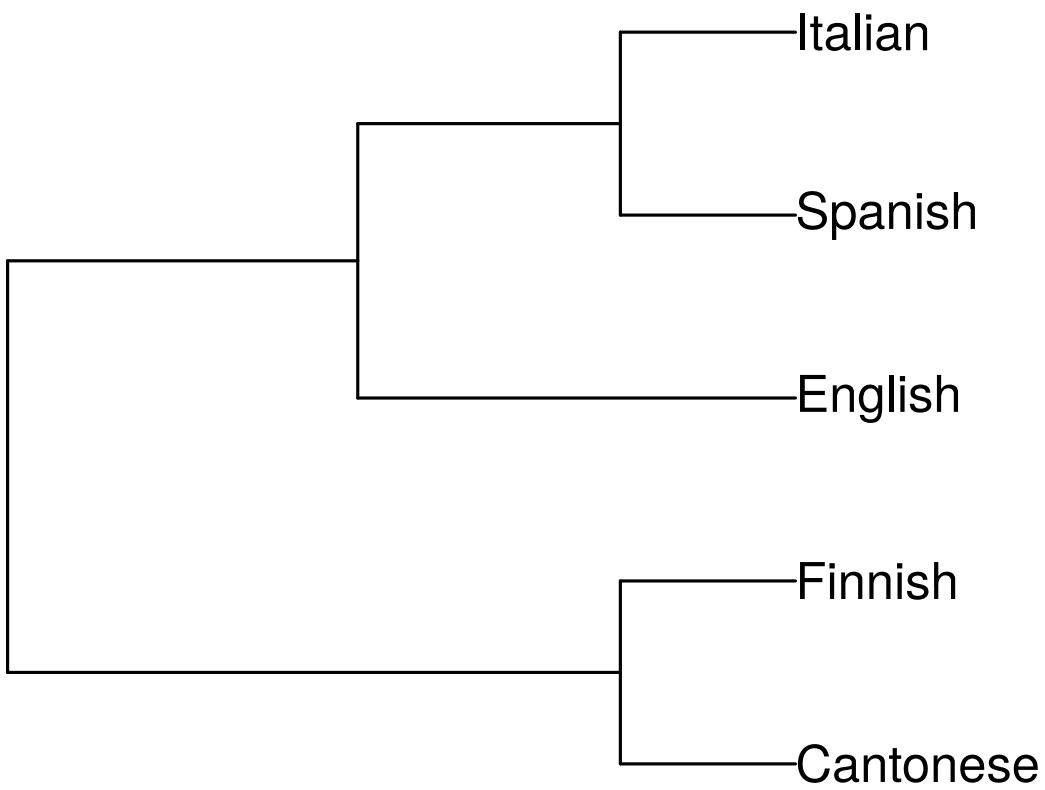
Metric trees

- Alternative representation of the same tree, where the physical length of each edge corresponds to the distance on that edge:



Metric trees

- Another alternative representation, a **PHYLOGRAM**:



The molecular clock

- It can be proved that every tree generated by UPGMA has the following property:

Ultrametricity

The distance from the root to a leaf is the same regardless of the leaf.

- It means that each leaf has changed just as much in comparison to the root as any other leaf.
- In biology, ultrametricity is interpreted as meaning that the rate of molecular evolution is constant and independent of the specific lineage (there is a "**MOLECULAR CLOCK**").
- Is this a reasonable assumption in linguistics?

Exercise

- Here's a feature matrix for four imaginary languages A, B, C and D and four arbitrary binary features F1–F4:

	F1	F2	F3	F4
A	1	1	1	1
B	1	1	1	0
C	1	0	0	0
D	0	0	0	0

1. Form the distance matrix
2. Use UPGMA to figure out the topological tree
3. Use the ultrametricity property to figure out distances on the edges of the corresponding metric tree

UPGMA in R

- In practice, no one does the UPGMA by hand (even though it is important to do it at least once so that you really understand what the method is doing!)
- To run the UPGMA in R, we use the phangorn package which provides the `upgma` function

```
library(phangorn)
mytree <- upgma(dist_mtx)
plot(mytree)
plot(mytree, type="phylogram")
```

- Try this now!