Subgrouping and trees

Quantitative Methods in Historical Linguistics

Dr. Henri Kauhanen / University of Konstanz

9 May 2018

First things first

- Questions?
- Let’s go through the IPA drill from last time
- Let’s go through the Quechuan exercise from last time

IPA drill

Exercise

Transcribe the following English and German words into IPA:

1. shoe
2. socks
3. trousers
4. pullover
5. stockings
6. Raum
7. Seminar
8. Veranstaltung
9. lernen
10. schreiben

IPA drill

Exercise

Transcribe the following English and German words into IPA:

1. shoe /ˈʃuː/
2. socks /ˈsɒks/
3. trousers /ˈtraʊzə(r)z/
4. pullover /ˈpʊləʊvə(r)/
5. stockings /ˈstɒkɪŋz/
6. Raum /ˈraʊm/
7. Seminar /zemiˈnaːr/
8. Veranstaltung /feranˈʃtaltʊŋ/
9. lernen /ˈlernən/
10. schreiben /ˈʃraɪbən/

- N.B. Exact transcription depends on variety! (British vs. American, Manchester vs. Liverpool, Standard German vs. Austrian German, etc.)
- e.g. realizations of the rhotic /r/: [r], [ɹ], [ɾ], [ʀ], [ʁ], ∅, …

Quechua

- Three varieties under inspection: Junín, Cajamarca and Ayacucho
- How many sound correspondences (for the consonants) did you find?
- How many consonants did you reconstruct?
- Which subgrouping is supported best?
- Let's go through this on the whiteboard…

Subgrouping

- We have seen how the comparative method (CM) may be used to propose language families
- We've also seen how we can take a step further to propose an internal structure for the family (subgrouping)
- Subgroupings are established based on **SHARED INNOVATIONS**

Shared innovation

A shared innovation is a characteristic exhibited by a subset of the languages in the family but not exhibited by the reconstructed proto-language.

Subgrouping

- Examples of shared innovations:
  - (from previous lecture:) Spanish and French display lenition in some contexts (/kabra/, /ʃɛvʁ/), which sets them apart from Italian (/kapra/) which retains the proto-phoneme
  - Junín and Ayacucho Quechua display lenition of /q/ to /χ/ and ∅; Cajamarca doesn't
  - Proto-Quechua had a post-nasal lenition rule, which Junín and Ayacucho (but not Cajamarca) lost

Lenition

- What exactly is lenition?
- **Sonority hierarchy**:

  voiceless stop > voiced stop > voiceless fricative > voiced fricative > nasal > approximant > close vowel > open vowel > ∅
- As a sound moves to the right along this hierarchy, it becomes more sonorous i.e. more vowel-like
- Ultimately, it may be lost altogether (∅)
- There is a physiological basis for lenition in sonorous contexts (e.g. if the sound occurs between two vowels)
- Latin *lenis* ‘soft’

Some common sound changes

- Reconstructions with the CM often involve the principle of directionality
- We need to know which kinds of sound changes (apart from lenition) are common
- Here's a rough overview…

Apocope

- **Apocope** refers to the loss of a sound at the end of a word
  - Vulgar Latin /pane/ > Spanish /pan/ (‘bread’)
  - Old English /sunu/ > Modern English /sɒn/
  - Proto-Finno-Ugric /jalkɑ/ > Estonian /jalk/ (‘foot’)

Epenthesis

- **Epenthesis**: insertion of a sound (often a vowel)
  - Old English /θunor/ > Modern English /θʌndər/
  - Middle Persian /bræːdær/ > Modern Persian /bæræːdær/ (‘brother’)
  - Latin /skoːla/ > Spanish /eskuela/ (‘school’)
    - (This is an example of **prothesis**, a special kind of epenthesis where a sound is inserted at the beginning of a word)

Diphthongization and monophthongization

- **Diphthongization**: the development of a monophthong into a diphthong
  - Latin /skoːla/ > Spanish /eskuela/ (‘school’)
  - Middle High German /huːs/ > Modern German /haʊs/ (‘house’)
- **Monophthongization**: the reverse process
  - Latin /aurum/ > Spanish /oro/ (‘gold’)

Nasalization

- **Nasalization**: a vowel turns from oral to nasal, sometimes becoming contrastive (=phonemic)
  - Latin /bonum/ > Early French /bɔ̃n/ > Modern French /bɔ̃/ (‘good’)
- Usually (as above) triggered by a following nasal consonant

Assimilation

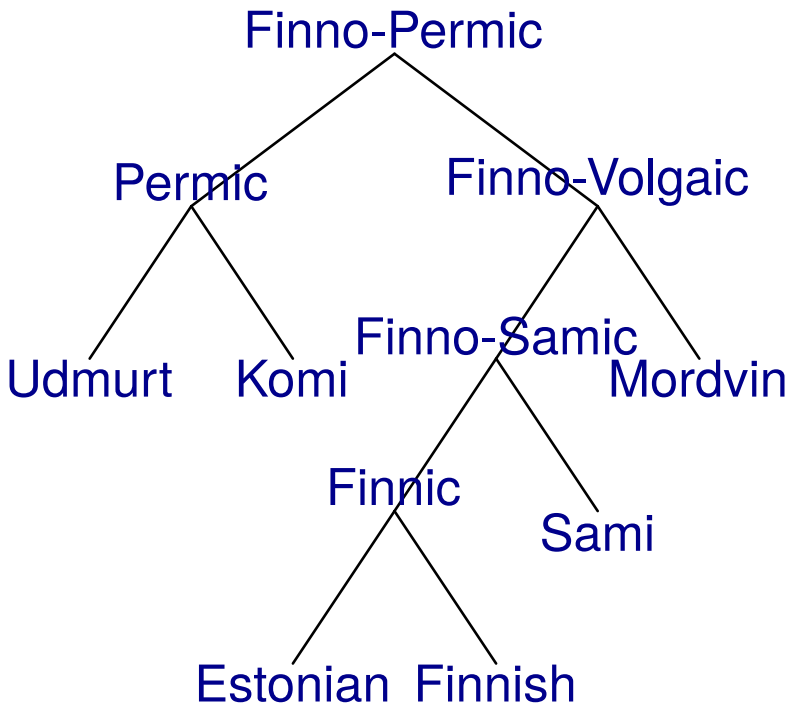
- **Assimilation**: a sound becomes more similar (or even identical) to an adjacent sound
  - Latin /septem/ > Italian /sɛtte/ (‘seven’)
  - Latin /lektus/ > Italian /letto/ (‘bed’)
  - Latin /bonum/ > Early French /bɔ̃n/

Gemination and degemination

- **Gemination**: the doubling (or lengthening) of consonants
  - Early Finnish /kukaːn/ > Savo Finnish /kukkaːn/ (‘nobody’)
- **Degemination**: the loss of a geminate
  - Latin /pekkaːtum/ > Spanish /pekado/ (‘sin’)

Phylogenetic trees

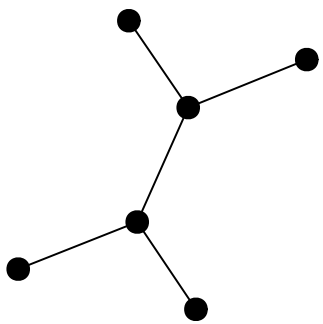
- The CM and subgrouping produce **PHYLOGENETIC TREES** (< Greek *phûlon* 'species', *génesis* 'origin')
- Example:



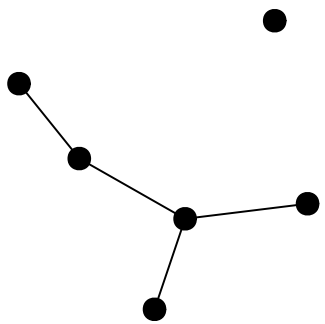
- Let's learn a bit more about trees (in general)…

Trees

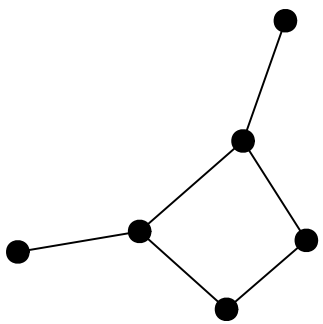
- A **TREE** is a set of **NODES** (dots) and **EDGES** (lines) with the following property: each pair of nodes is connected by exactly one path along the edges.



a tree



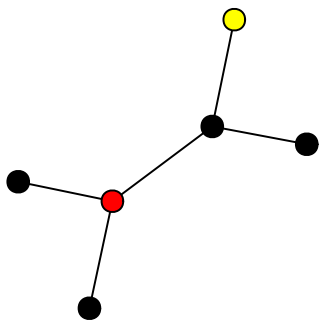
not a tree



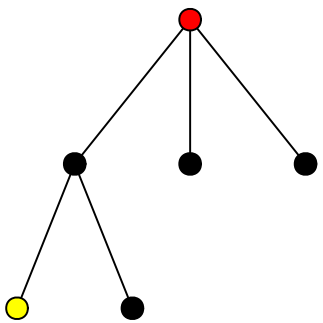
not a tree

Trees

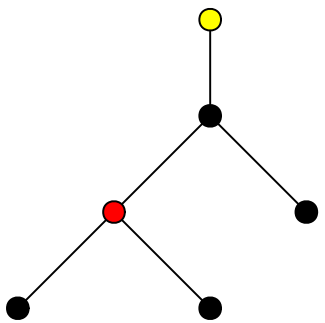
- A tree can be **ROOTED** by choosing a particular node as the root.



unrooted tree



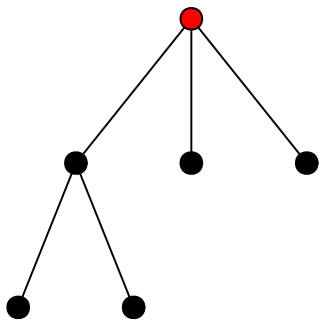
rooted at red



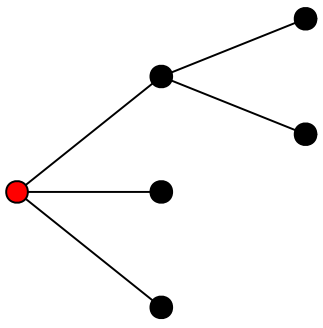
rooted at yellow

Trees

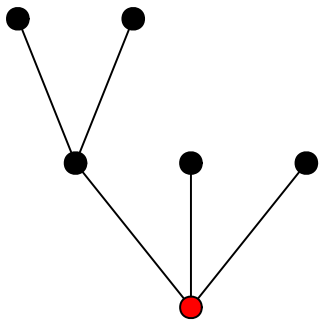
- A rooted tree can be “grown” in any direction.



top-down



left-to-right

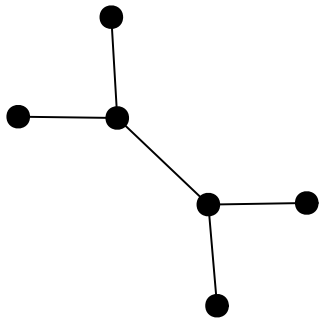


bottom-up

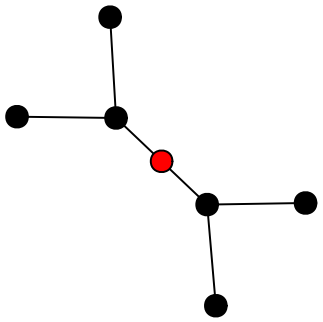
- Usually in this course (and elsewhere) trees are grown either top-down or left-to-right.

Trees

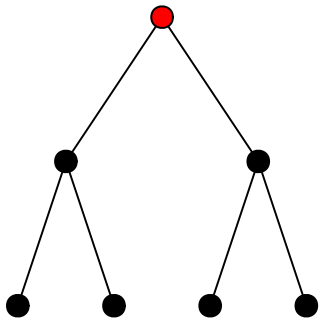
- A tree can also be **ROOTED AT AN EDGE** by placing an extra node at one edge and choosing that node as the root.



unrooted tree



extra node

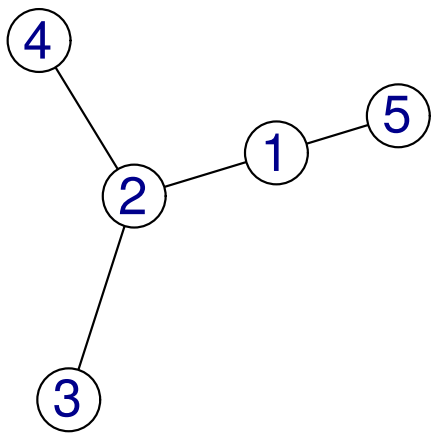


rooted tree

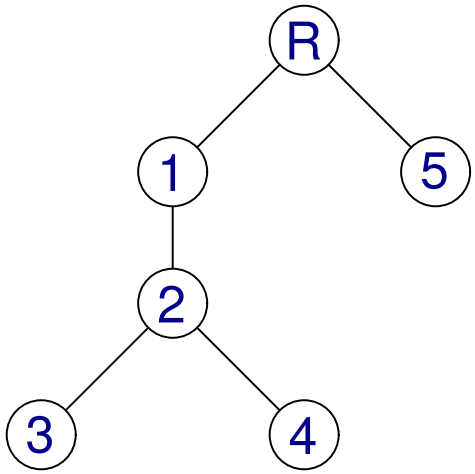
- We will usually root at an edge rather than at a node.

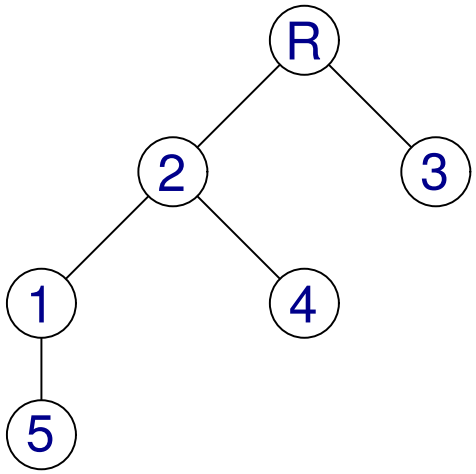
Trees

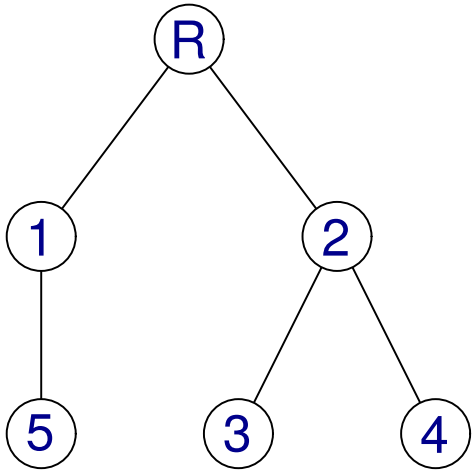
- Exercise: Root the following tree at each of its edges.

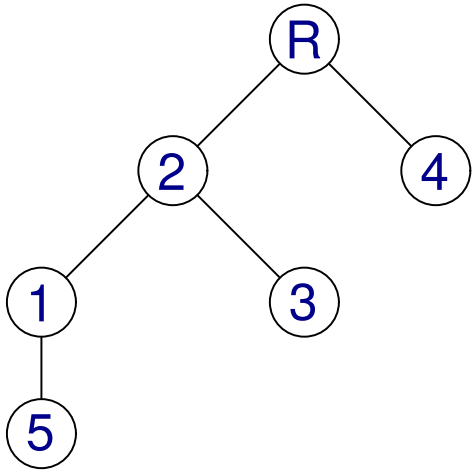


Trees







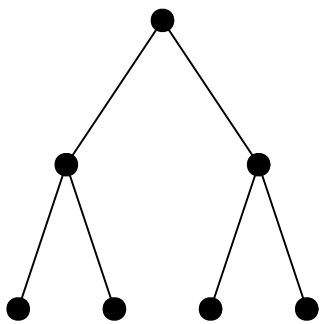


Trees

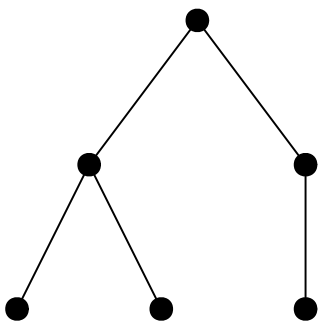
- A **TERMINAL NODE** (a.k.a. **LEAF**) is a node which has one edge only.
- All other nodes are **INTERNAL NODES**.
- Here, terminal nodes in yellow and internal nodes in green:

Trees

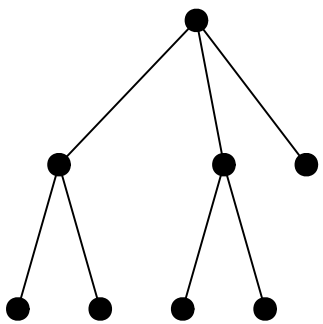
- A rooted tree is **BINARY-BRANCHING** if:
  1. each internal node apart from the root has three edges; and
  2. the root has two edges.



binary-branching



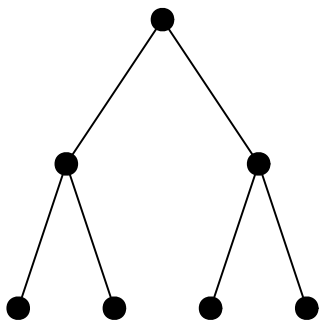
non-binary-branching

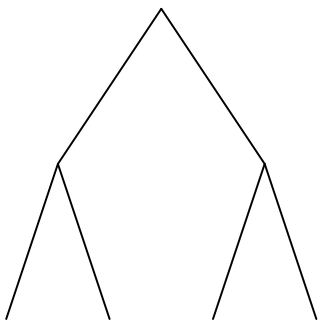


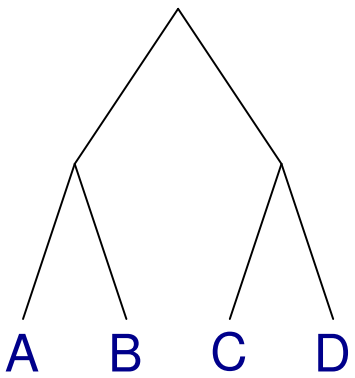
non-binary-branching

Trees

- Sometimes the nodes are not shown explicitly, or only the leaves are labelled.
- So these three represent one and the same tree:







Phylogenetic trees

- In a phylogenetic tree, the leaves usually correspond to existing languages
- Internal nodes correspond to reconstructed proto-languages
- These internal nodes also correspond to linguistic **SPLITS**
  - e.g. Finno-Samic split into Finnic and Sami
  - Finnic split into Estonian and Finnish
  - Finnish split into Western Finnish and Eastern Finnish
  - and so on
- In brief: a phylogenetic tree is a hypothesis about how the leaves are related through historical developments

Phylogenetic trees

- We can now state the

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?

- This is a hard problem!
- Methods such as CM are time-consuming and error-prone, and cannot make use of too much data (because that would take too much time)
- Can computational methods help?
- Find out next time!

Drawing trees with R

- We can use R to draw trees, using e.g. the `igraph` package
- First, let's load the package:

```
library(igraph)
```

- (If that doesn't work, you'll have to install the package first:)

```
install.packages("igraph")
```

Drawing trees with R

- A tree is constructed using the `graph` function
- It takes an `edges` argument which specifies the nodes and edges of the tree
- We also set `directed=FALSE`

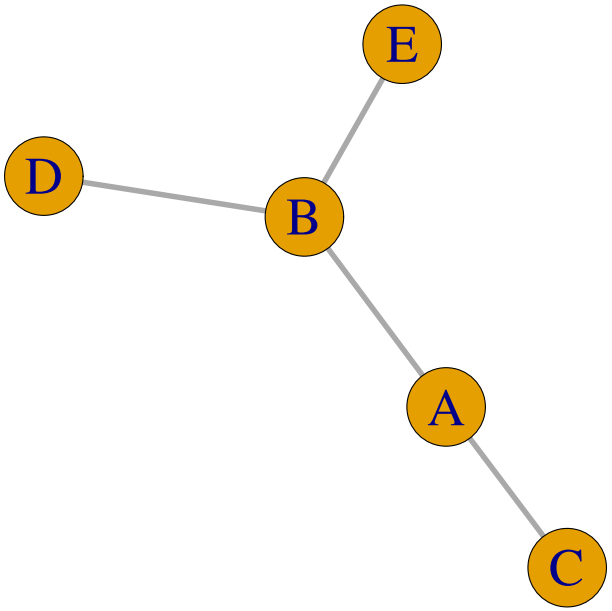
```
t <- graph(edges=c("A", "B", "A", "C", "B", "D", "B", "E"),
directed=FALSE)
```

- The function returns a “graph object” which we store in the variable `t`
- In this case, the following edges are formed:
  - A–B
  - A–C
  - B–D
  - B–E

Drawing trees with R

- We can draw a picture of the tree using the `plot` function:

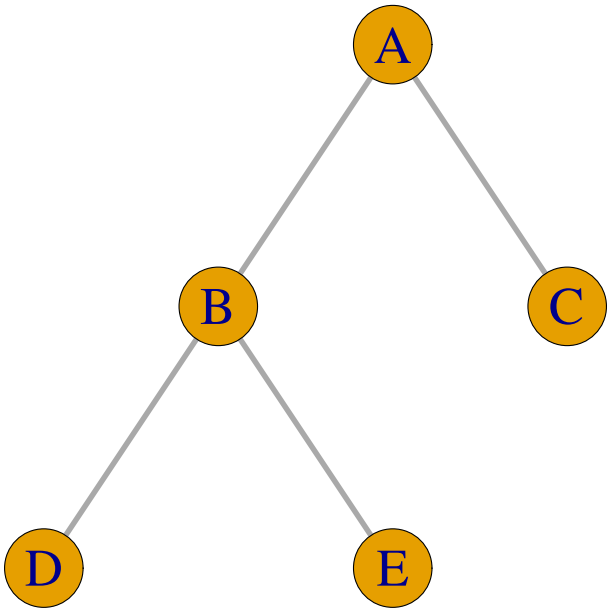
```
plot(t)
```



Drawing trees with R

- By default we get an unrooted tree. Here's how to root it at a particular node:

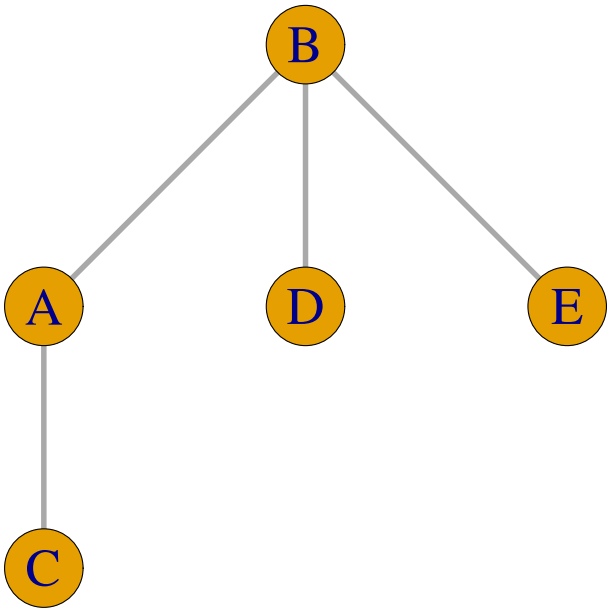
```
plot(t, layout=layout_as_tree(t, root="A"))
```



Drawing trees with R

- By default we get an unrooted tree. Here's how to root it at a particular node:

```
plot(t, layout=layout_as_tree(t, root="B"))
```



Drawing trees with R

- You can pass many other arguments to `plot` to control the way your tree is drawn
- Try the following (and their combinations):

```
plot(t, vertex.color="red")
plot(t, vertex.size=40)
plot(t, vertex.label=c("node1", "node2", "node3", "node4",
"node5"))
plot(t, vertex.label.cex=4.0)
plot(t, edge.color="blue")
plot(t, edge.width=5)
plot(t, edge.lty=2)
```

Drawing trees with R

- More info about the plotting parameters:

```
?igraph.plotting
```

- and at http://kateto.net/networks-r-igraph (more info than we will ever need)

Porfolio Exercise 2

- Available now on ILIAS
- Extends our work on applying the CM to Quechua