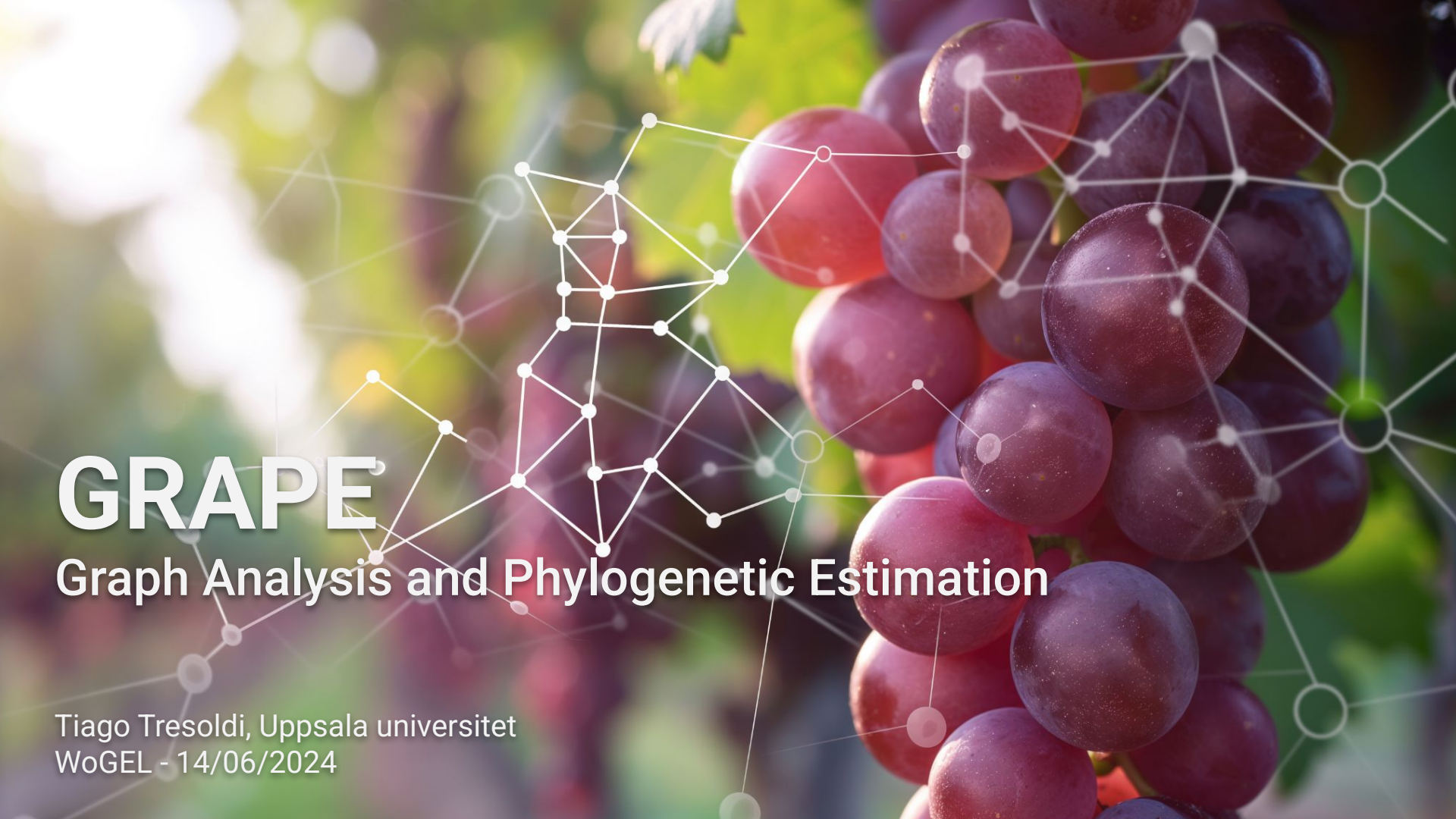


GRAPE

Graph Analysis and Phylogenetic Estimation

Tiago Tresoldi, Uppsala universitet
WoGEL - 14/06/2024

Introduction

- A novel method for phylogenetic inference
- Really “novel”
  - Based on community-detection in graphs
  - “Novel” does not mean it will change the world
- Explore how it works
- Give some linguistic examples
- Quickly discuss publication and alternatives

Methods for phylogenetic inference

- Manual construction
- Distance-based methods
  - NJ and UPGMA
- Character-based methods
  - Parsimony
  - Maximum Likelihood and Bayesian

Methods for phylogenetic inference

- Manual construction
- Distance-based methods
  - NJ and UPGMA
- Character-based methods
  - Parsimony
  - Maximum Likelihood and Bayesian

	Swedish	Danish	Italian	Hittite
Swedish	0.0	0.1	0.4	0.9
Danish	0.1	0.0	0.4	0.9
Italian	0.4	0.4	0.0	0.9
Hittite	0.9	0.9	0.9	0.0

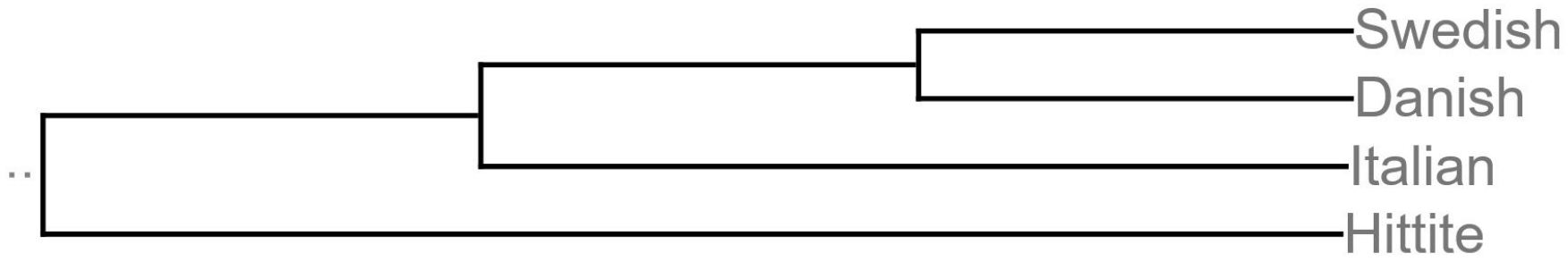
Note: Made-up numbers!

	Swedish	Danish	Italian	Hittite
Swedish	0.0	0.1	0.4	0.9
Danish	0.1	0.0	0.4	0.9
Italian	0.4	0.4	0.0	0.9
Hittite	0.9	0.9	0.9	0.0

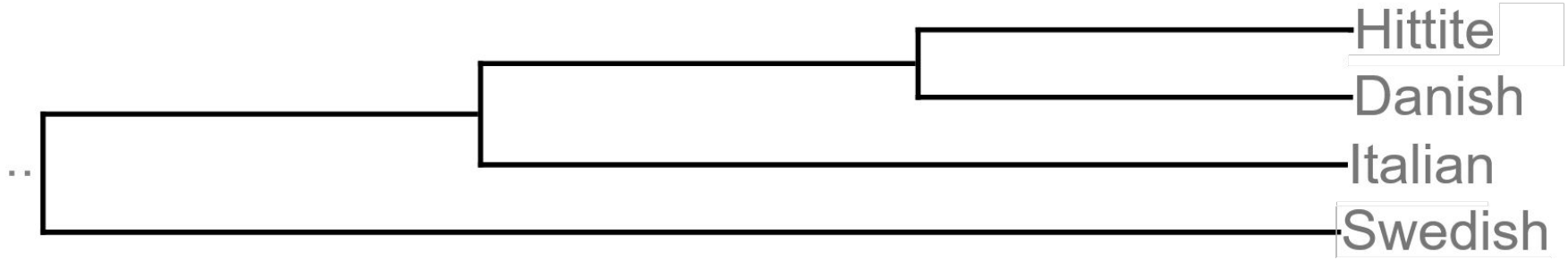
Note: Made-up numbers!

Methods for phylogenetic inference

- Manual construction
- Distance-based methods
  - NJ and UPGMA
- Character-based methods
  - Parsimony
  - Maximum Likelihood and Bayesian

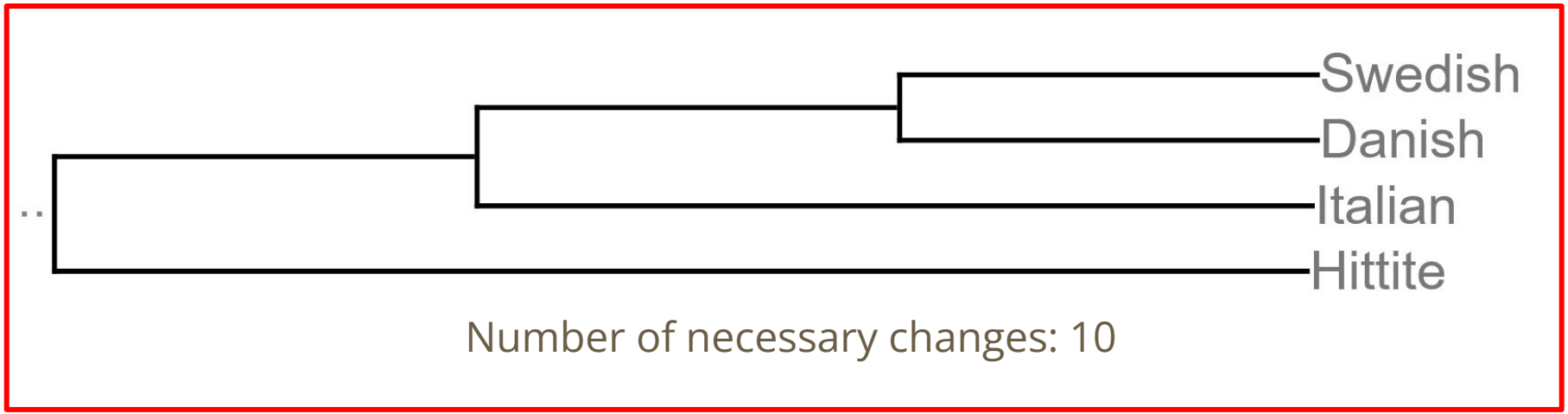


Number of necessary changes: 10

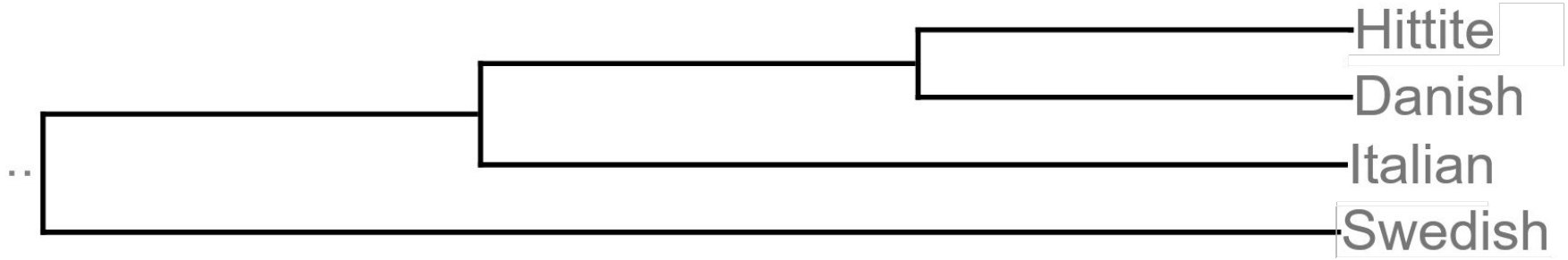


Number of necessary changes: 80

Note: Made-up numbers!



Number of necessary changes: 10



Number of necessary changes: 80

Note: Made-up numbers!

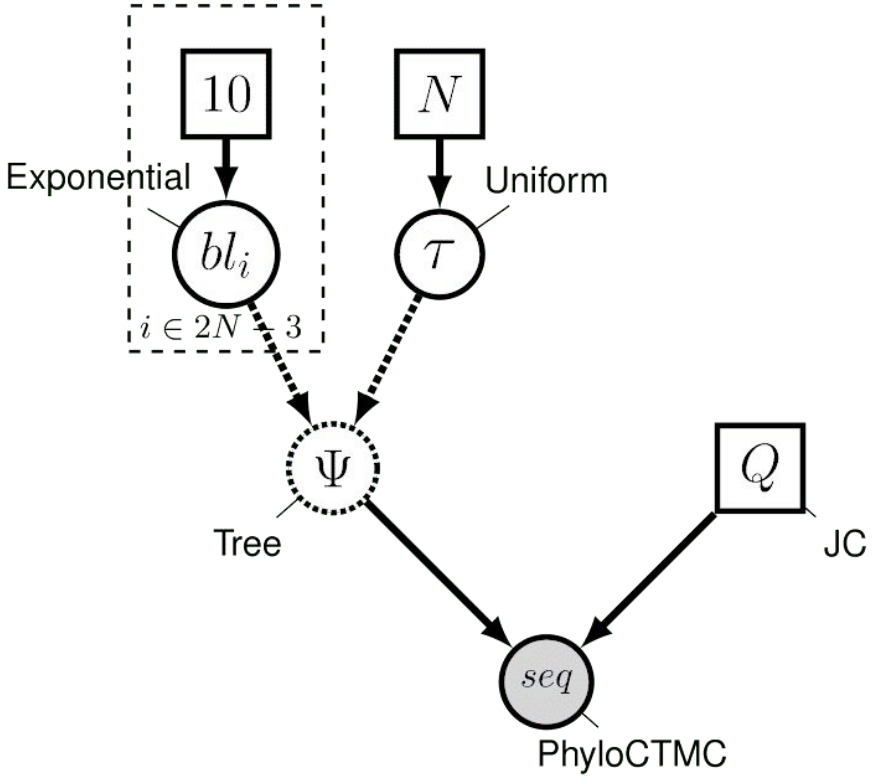
$$\mathbf{Q} = \mathbf{R}\boldsymbol{\Pi} = \begin{array}{c} \\ A \\ C \\ G \\ T \end{array} \begin{array}{c} \begin{array}{cccc} A & C & G & T \end{array} \\ \begin{pmatrix} - & \mu a & \mu b & \mu c \\ \mu a & - & \mu d & \mu e \\ \mu b & \mu d & - & \mu f \\ \mu c & \mu e & \mu f & - \end{pmatrix} \end{array} \begin{pmatrix} \pi_A & 0 & 0 & 0 \\ 0 & \pi_C & 0 & 0 \\ 0 & 0 & \pi_G & 0 \\ 0 & 0 & 0 & \pi_T \end{pmatrix}$$

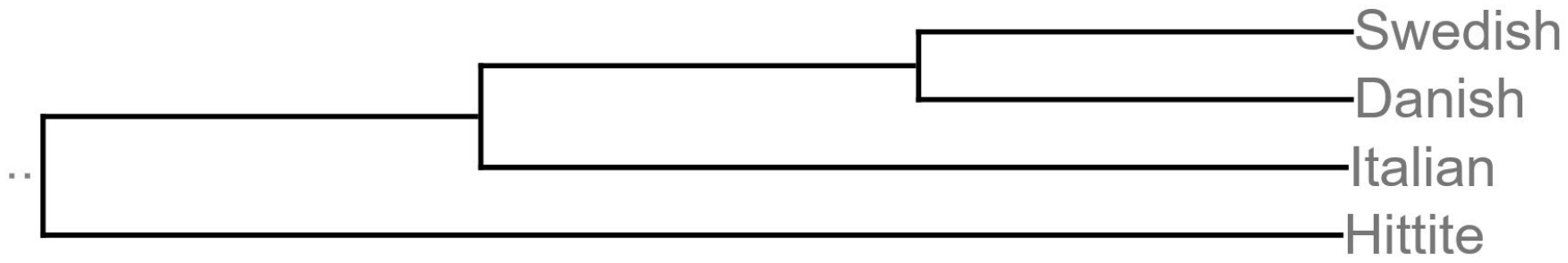
$b + e + a + c + d + f = 6$ $\qquad$ $\pi_A + \pi_C + \pi_G + \pi_T = 1$

Model	Structure	Degrees of freedom
GTR	1: μ ; b e a c d f ; π_C π_G π_T π_A	8
HKY85	1: μ ; $b = e$; $a = c = d = f$; π_C π_G π_T π_A	4
F81	1: μ b e a c d f ; π_C π_G π_T π_A	3
JC	1: μ b e a c d f ; 0.25: π_C π_G π_T π_A	0

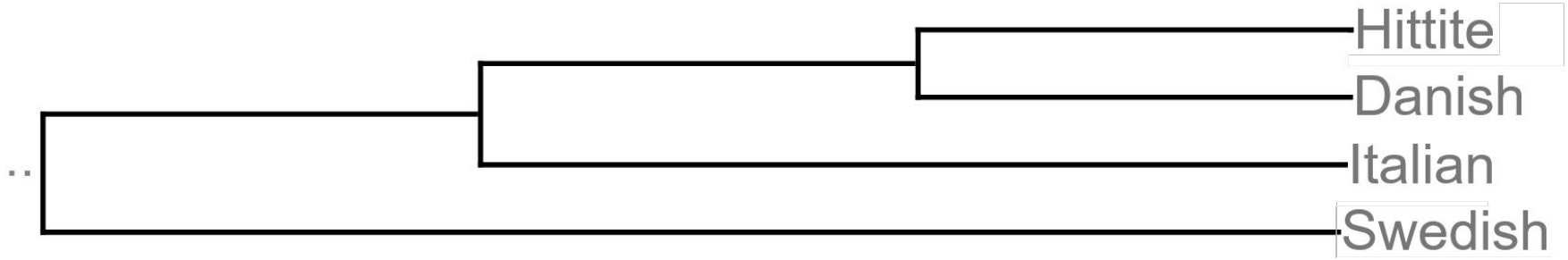
Legend

- 1: x — Constant. Set to the value above the box.
- (x) — Stochastic. Free to vary.
- (x) — Deterministic. Value depends on other values.



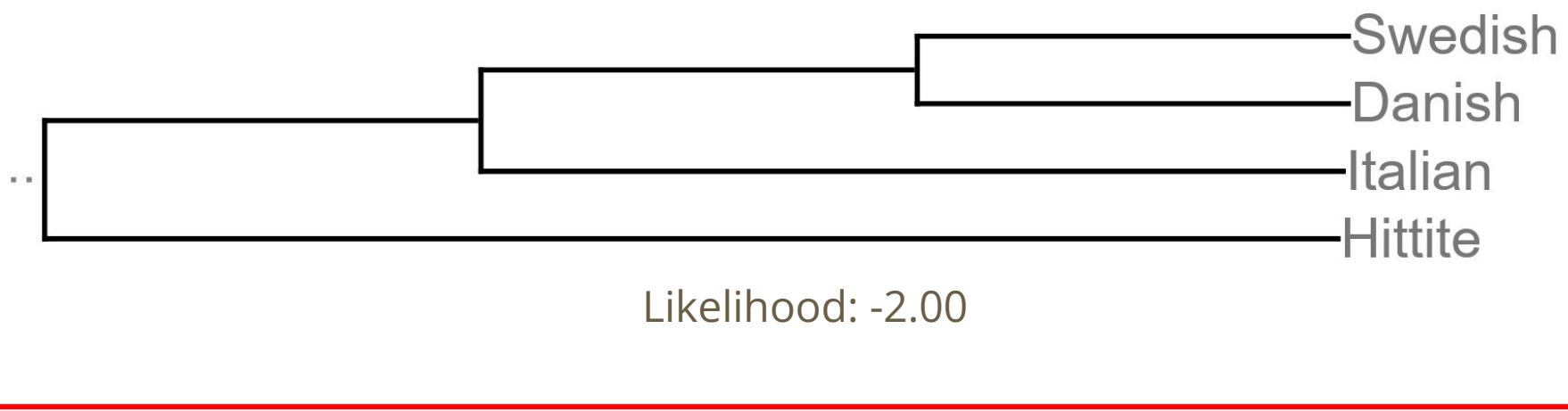


Likelihood: -2.00

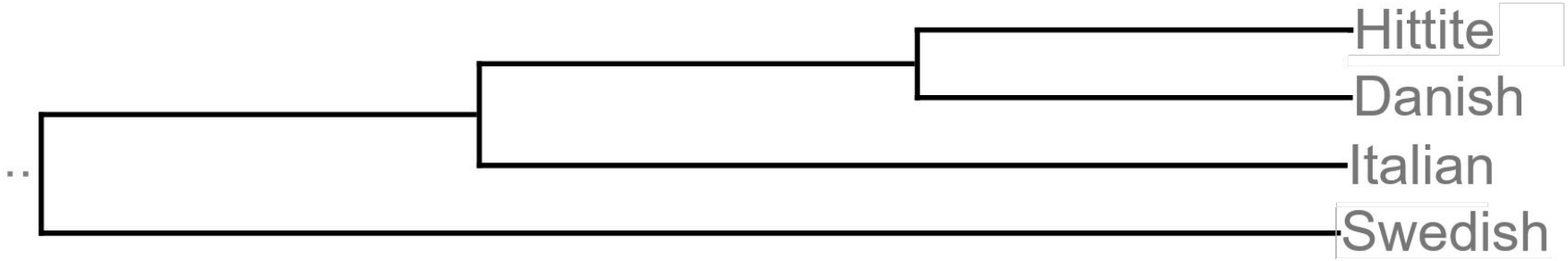


Likelihood: -10.00

Note: Made-up numbers!



Likelihood: -2.00

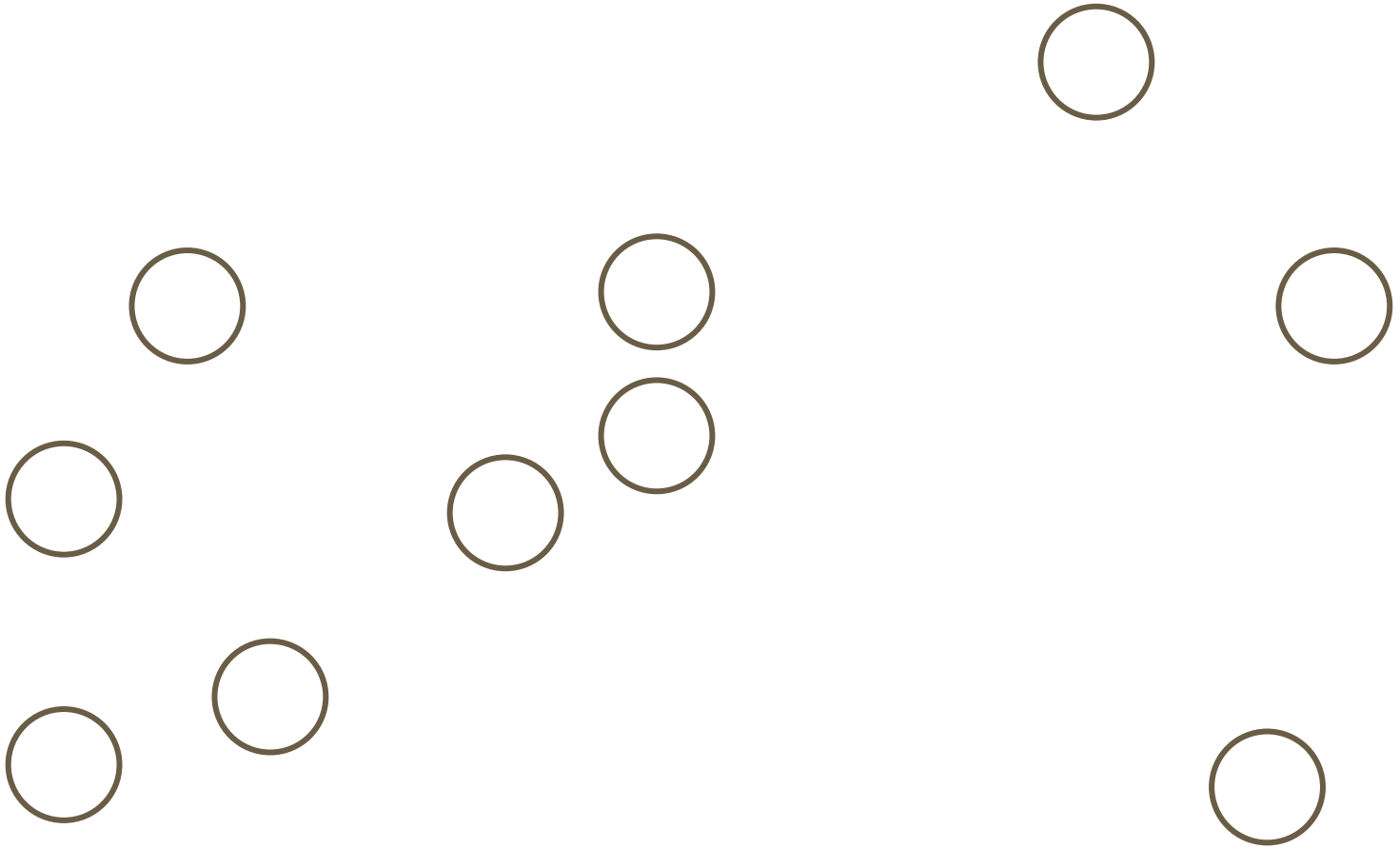


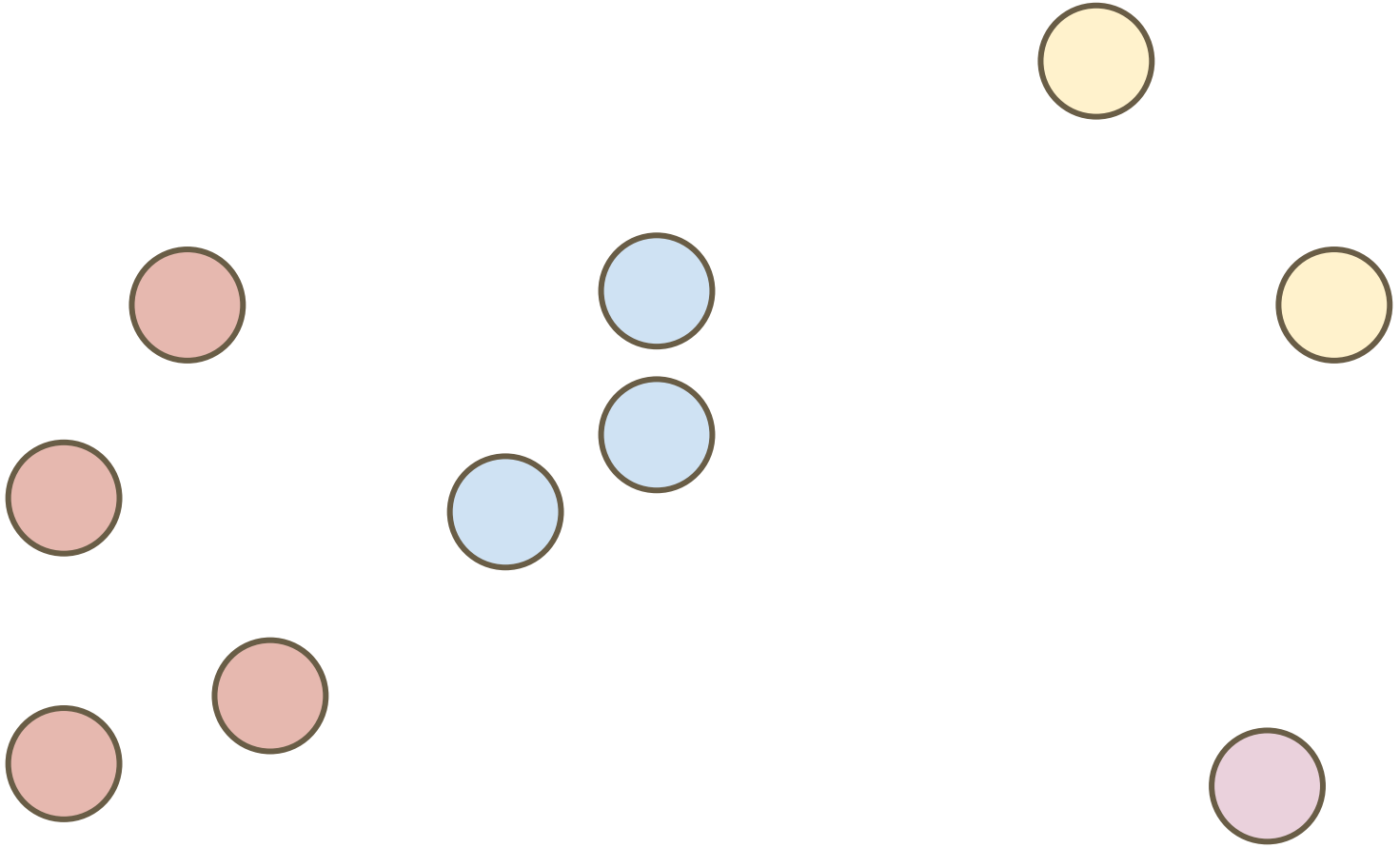
Likelihood: -10.00

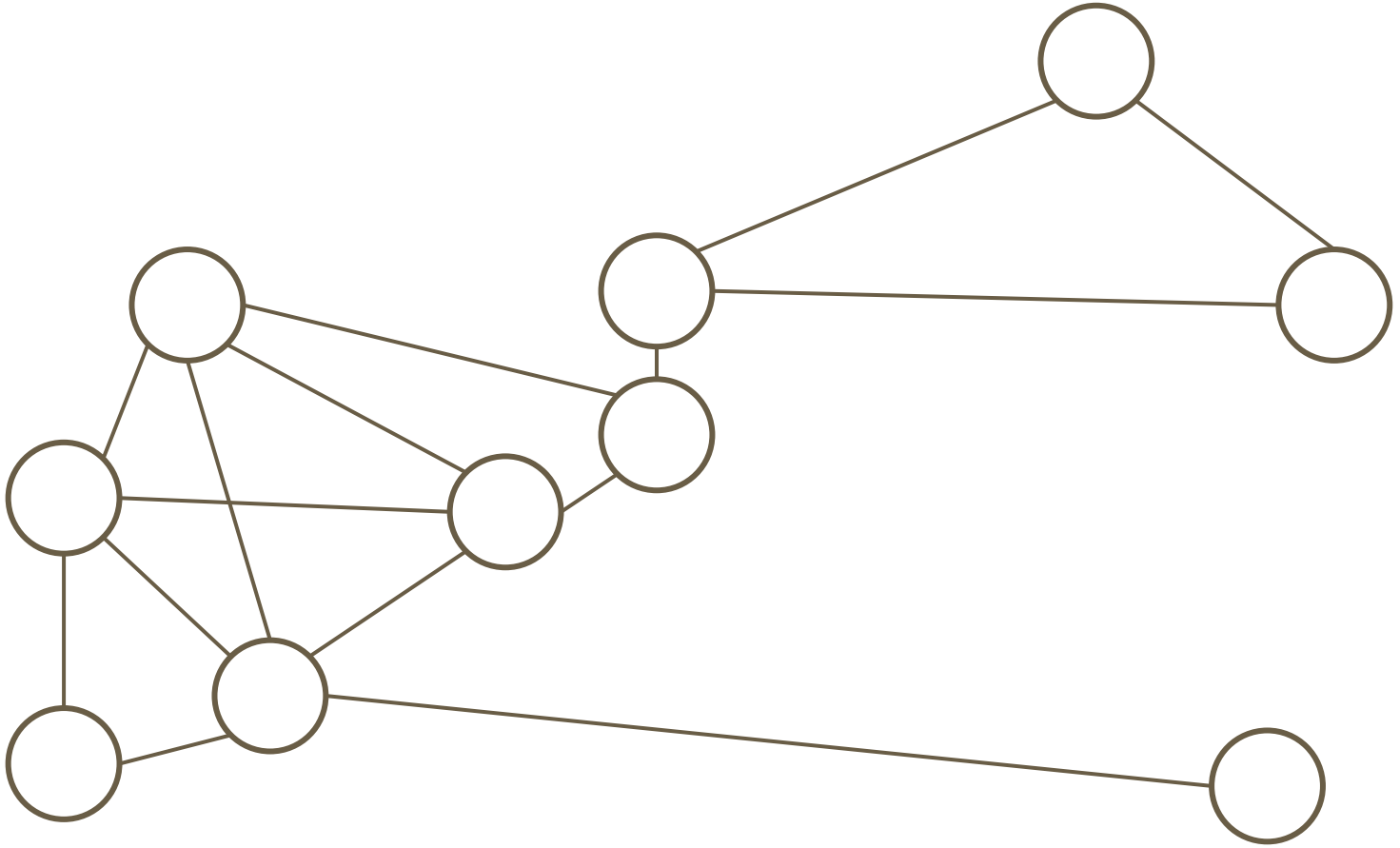
Note: Made-up numbers!

Limitations

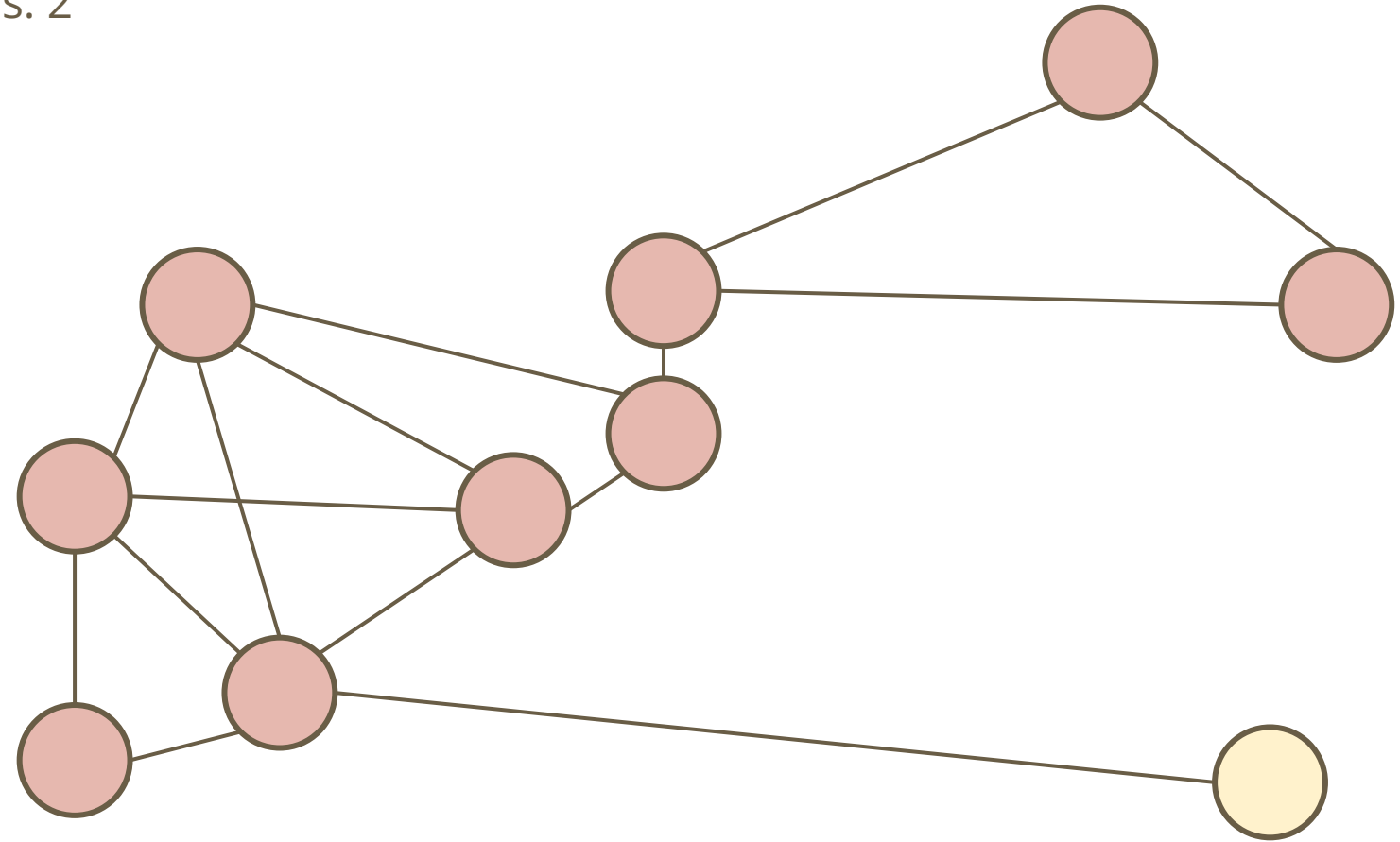
- Some methods are hard to introspect and understand
- Some methods can take *extremely* long time to compute
- No method really addresses issues like borrowings
  - But, truth be told, there are some extensions
- No method really addresses issues like informativity
  - Shared innovations vs. shared retentions
  - This can be specified by the researcher (good!)



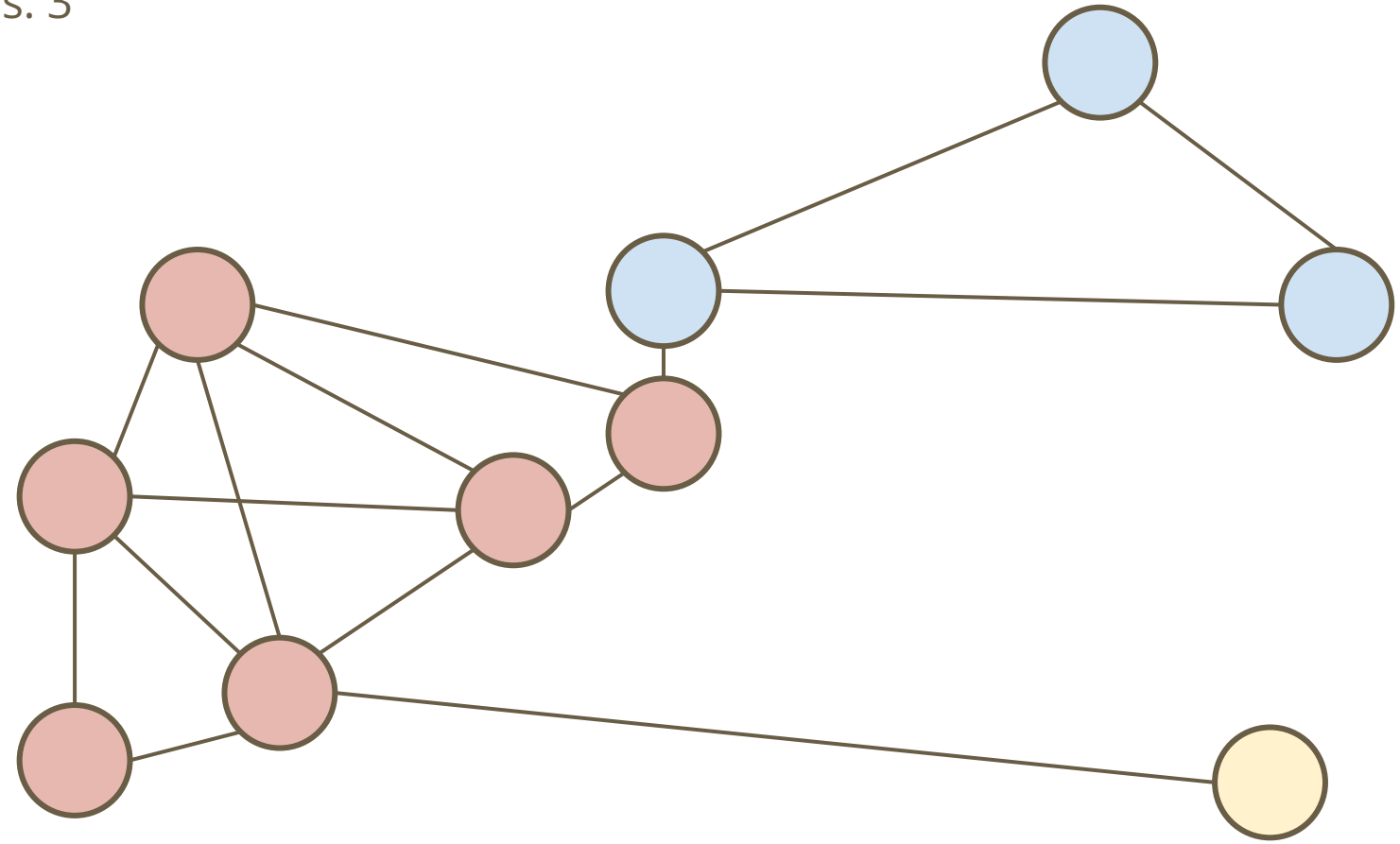




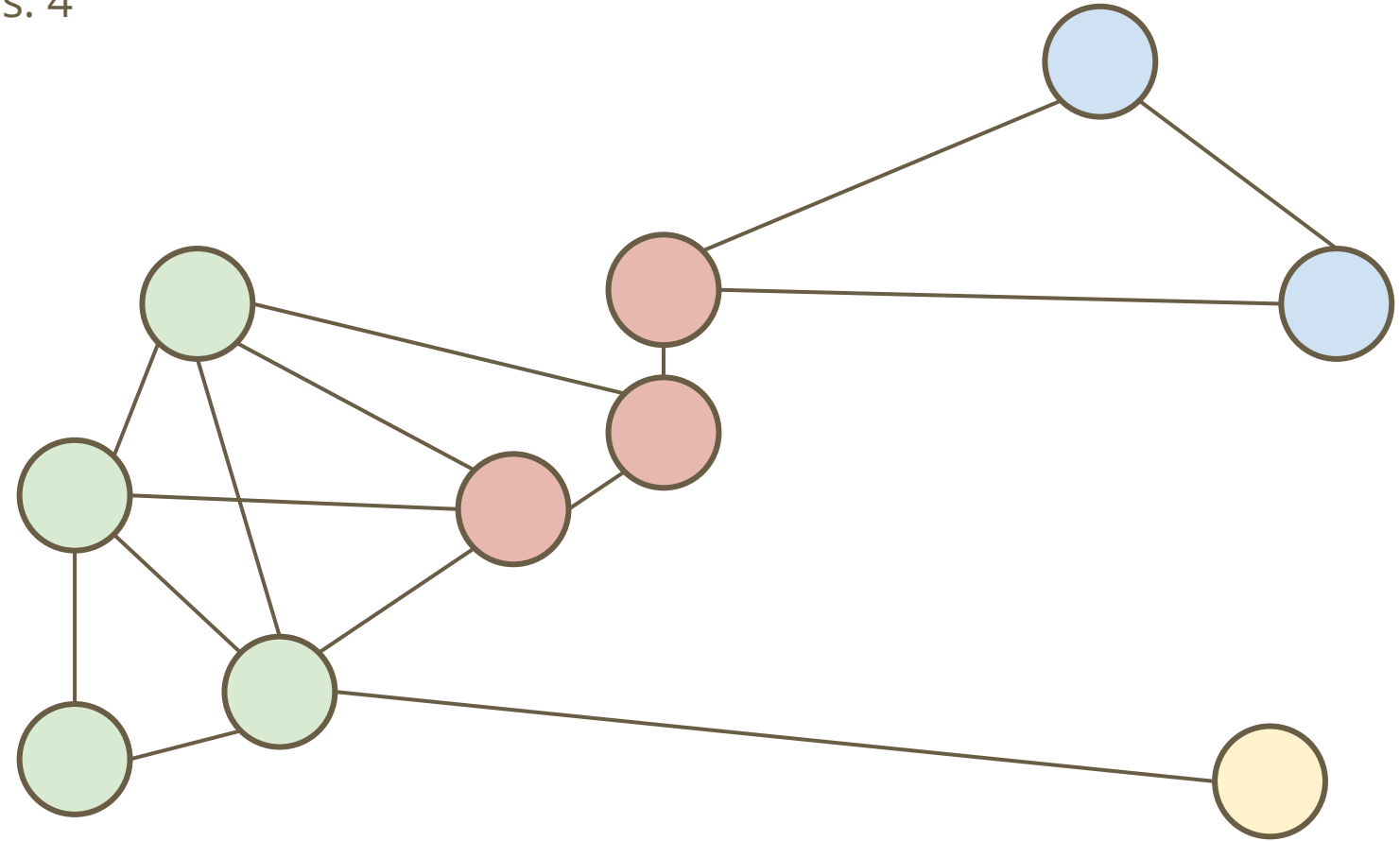
Clusters: 2



Clusters: 3

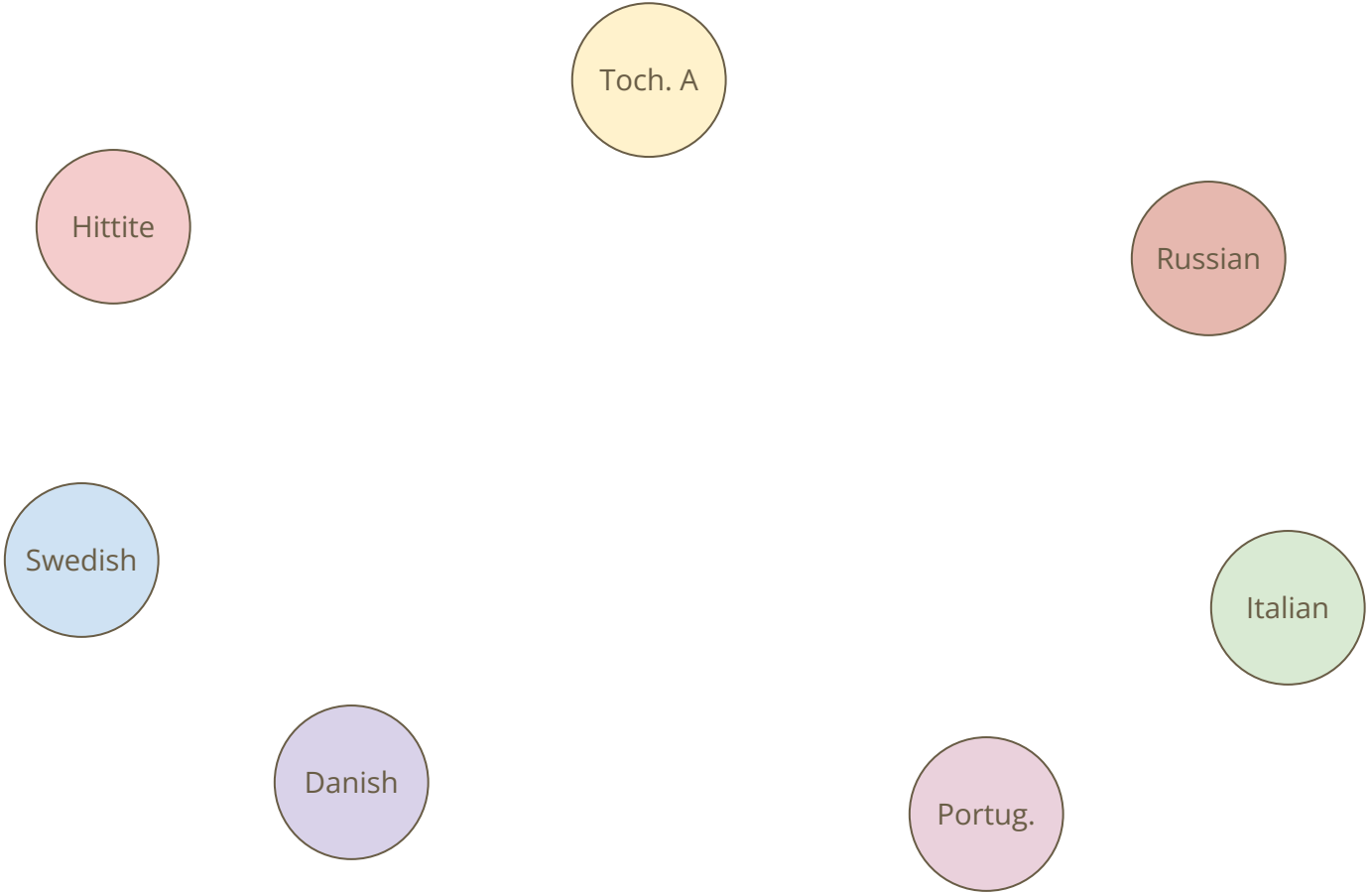


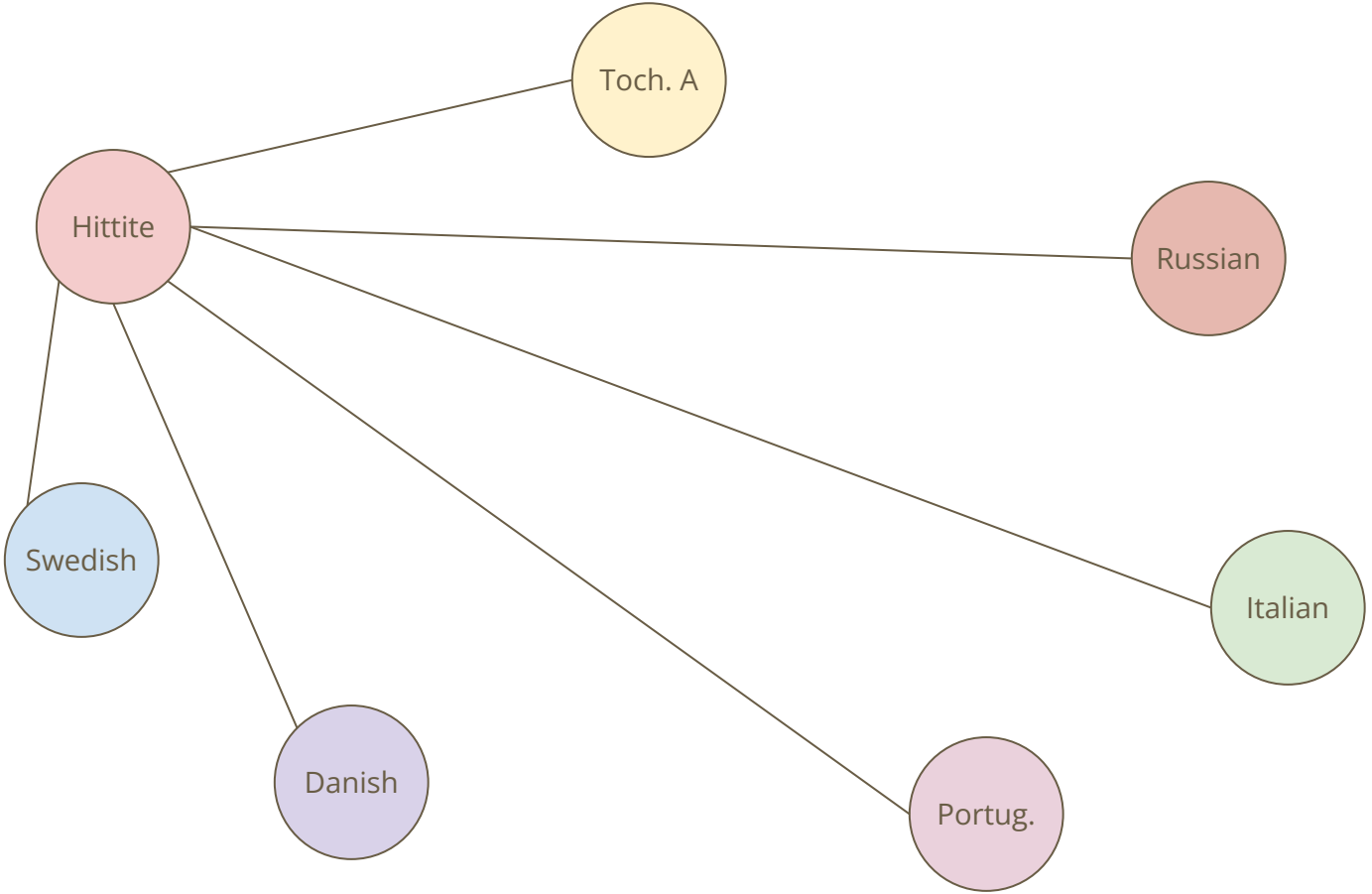
Clusters: 4

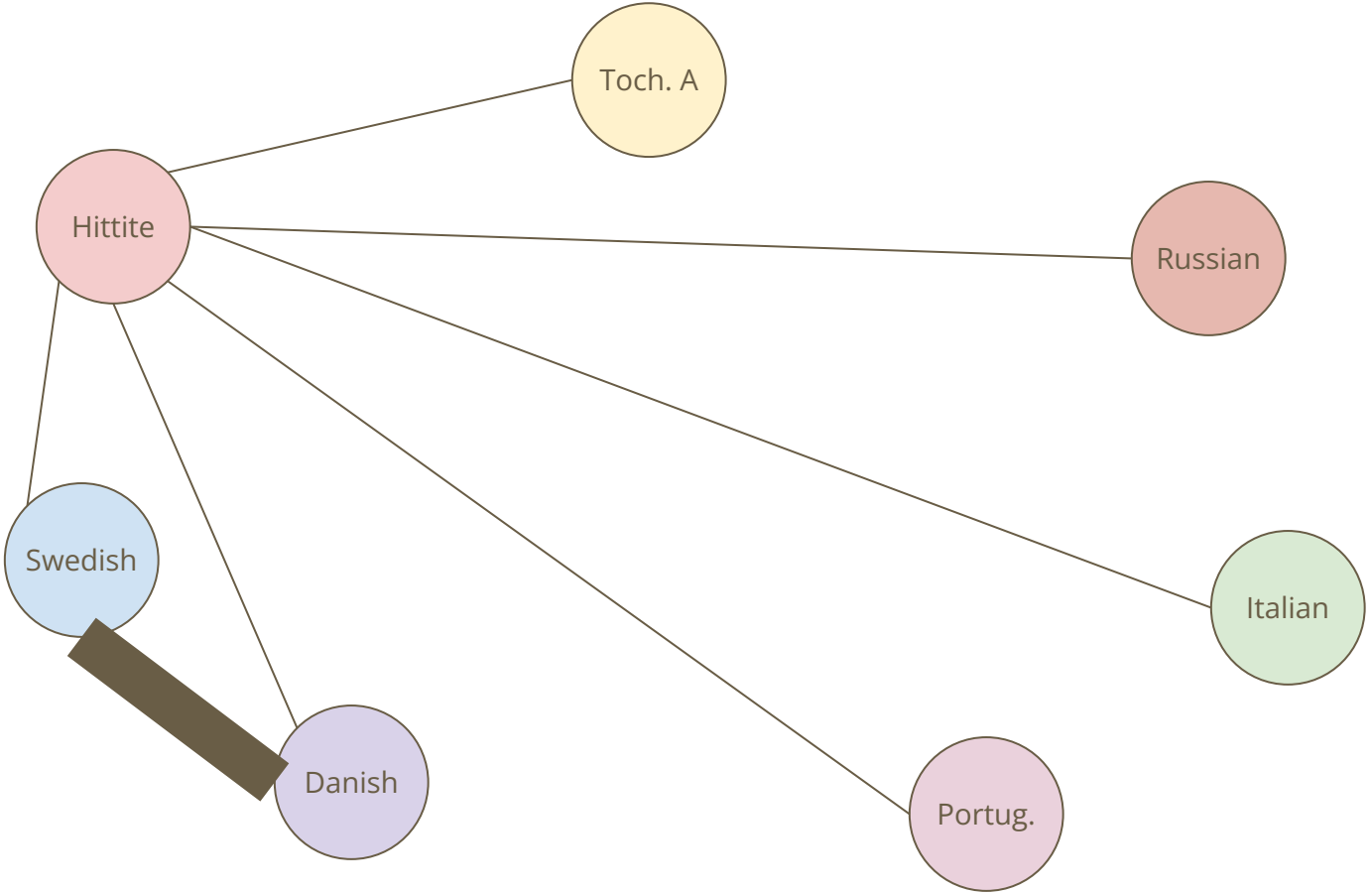


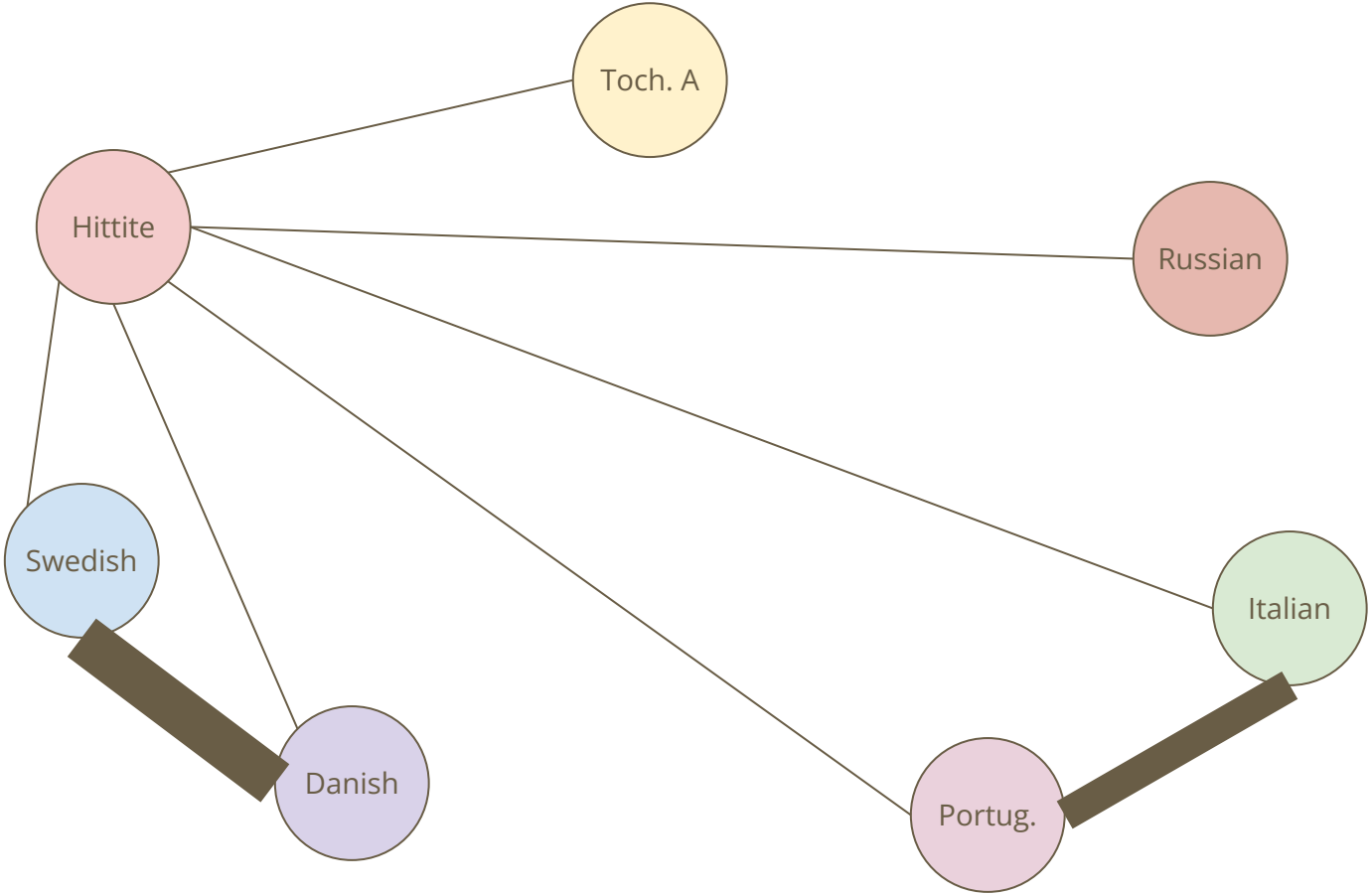
Community detection

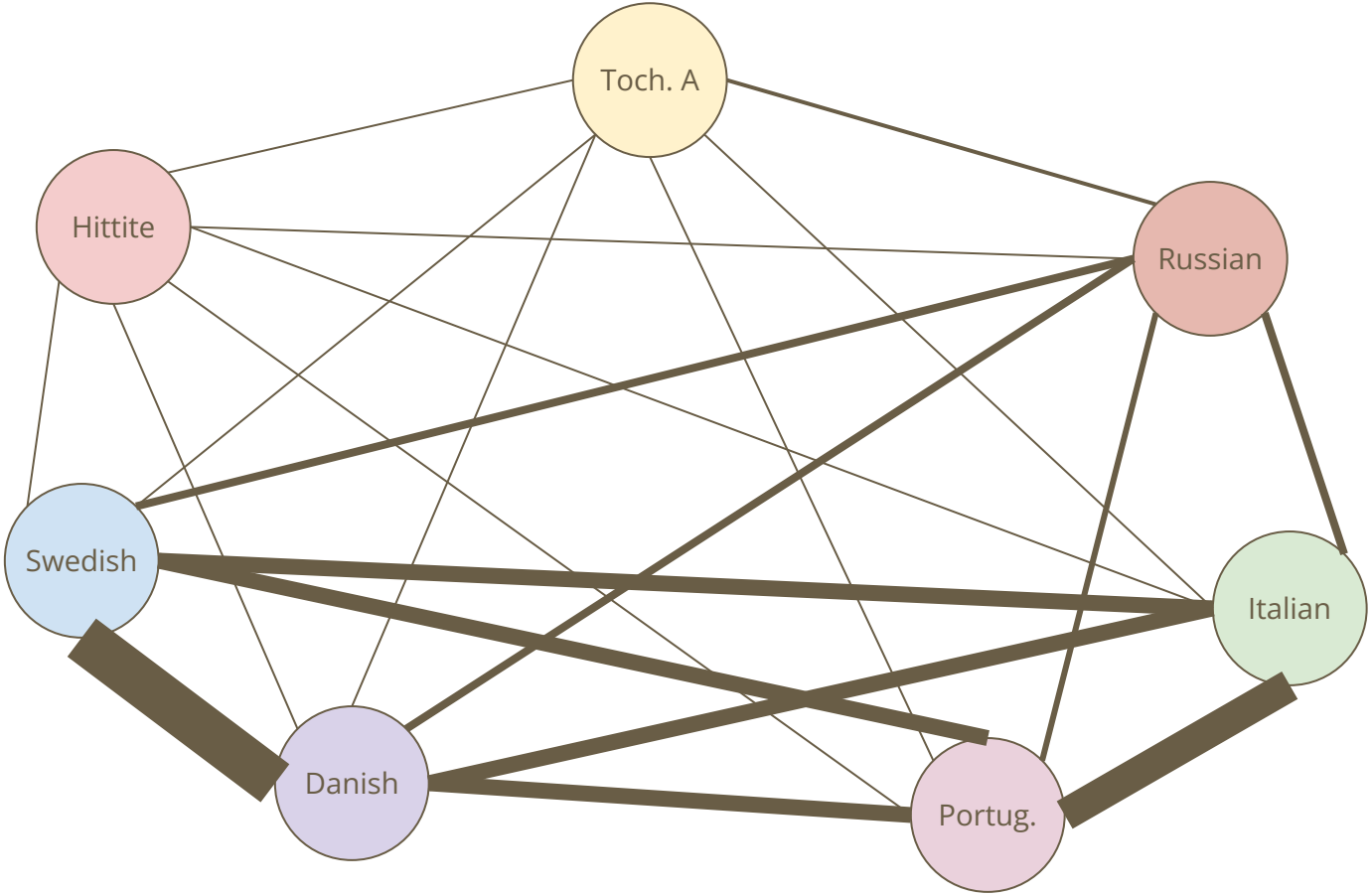
- Very active area of research
  - Already used in CHL: Lexstat+Infomap, for example, is the best method in `lingpy` because of Infomap, not Lexstat
- Many methods allow to specify either the number the clusters (*k*) or the resolution (*r*)
- As seen, it is not guaranteed the higher resolutions will respect the lower-resolution groupings
- A few algorithms (greedy, Louvain, Infomap) allow us to use **weighted** edges


Toch. A
Hittite
Russian
Swedish
Italian
Danish
Portug.

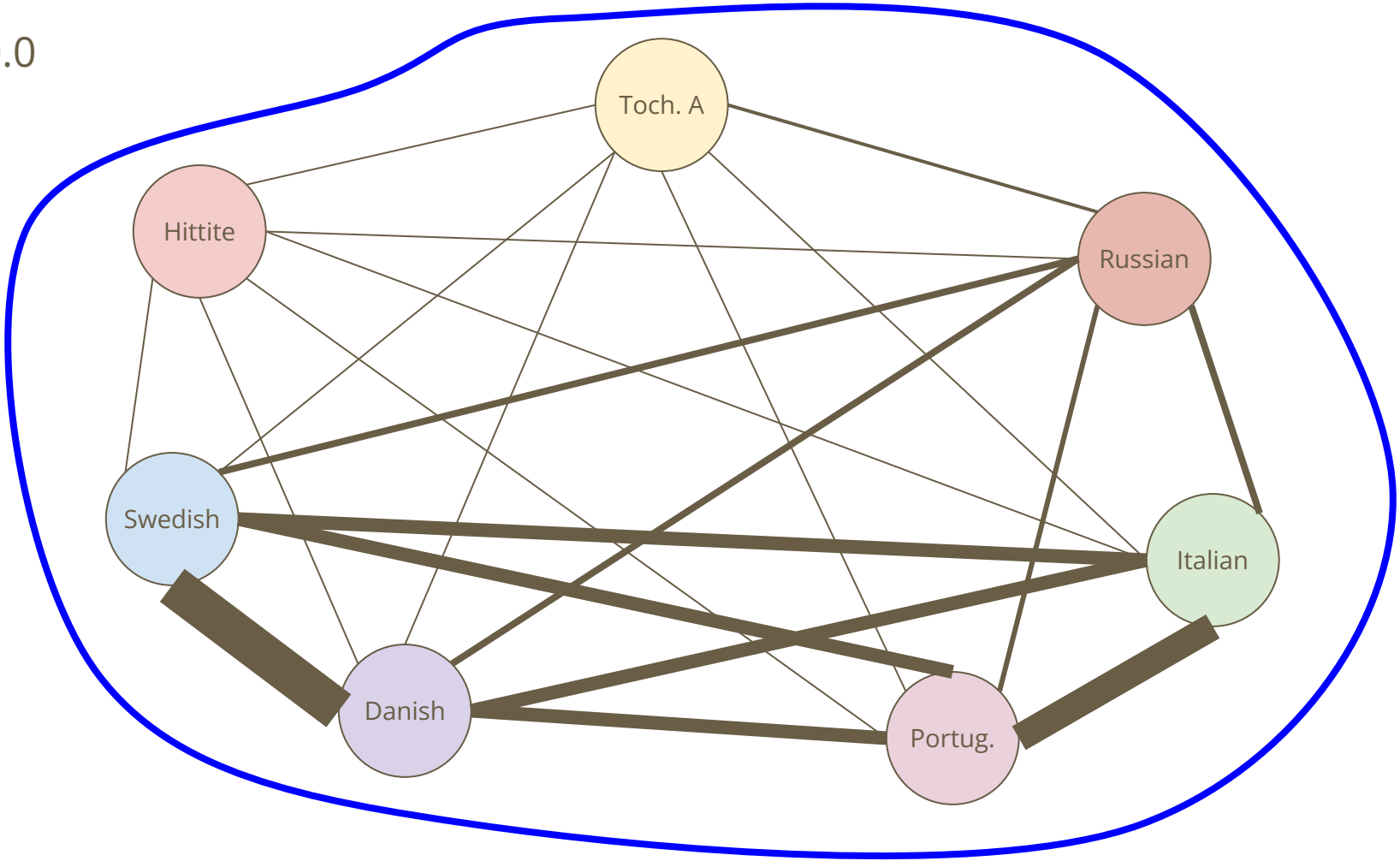

Toch. A
Hittite
Russian
Swedish
Italian
Danish
Portug.


Toch. A
Hittite
Russian
Swedish
Italian
Danish
Portug.

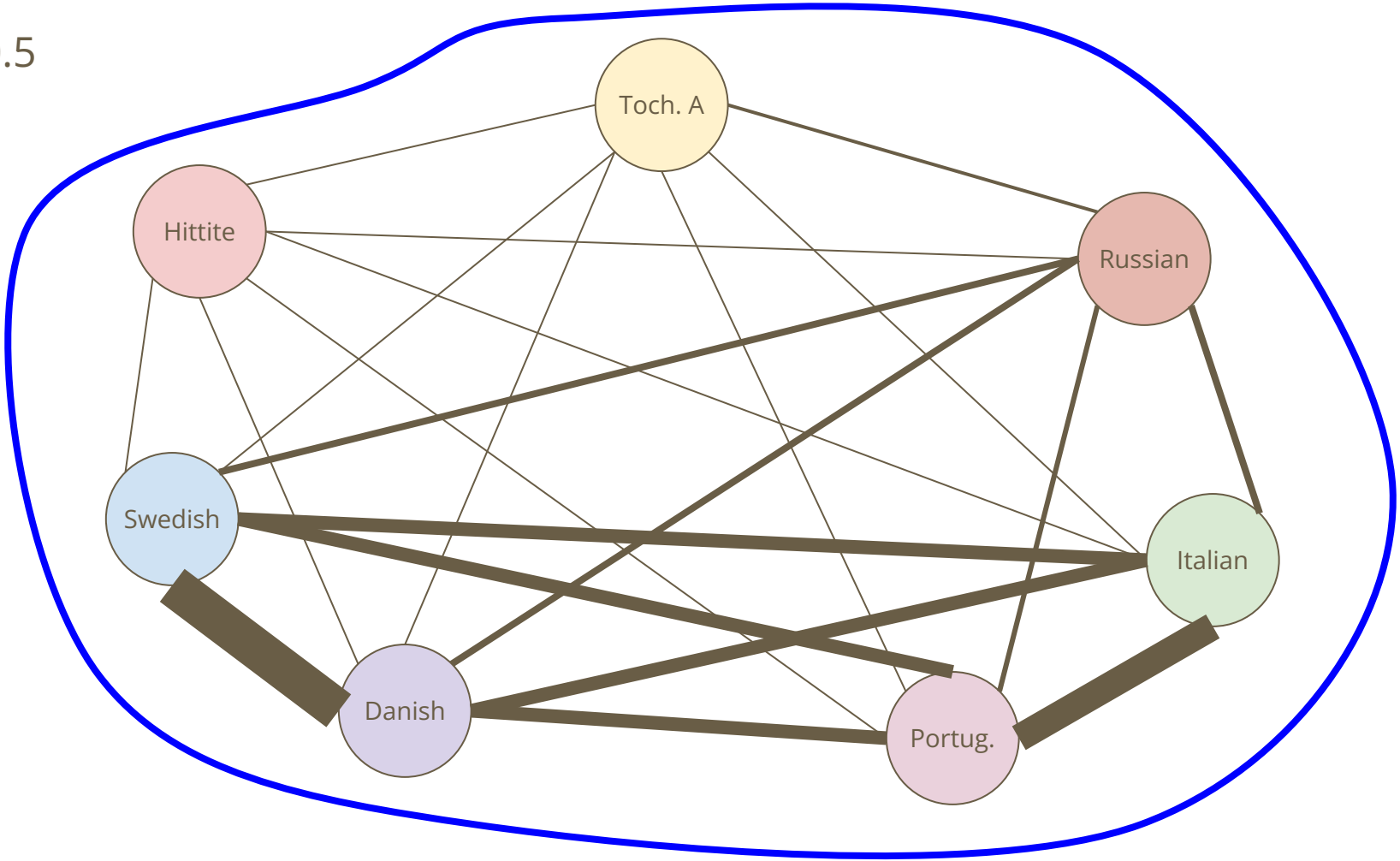

Toch. A
Hittite
Russian
Swedish
Italian
Danish
Portug.


Toch. A
Hittite
Russian
Swedish
Italian
Danish
Portug.

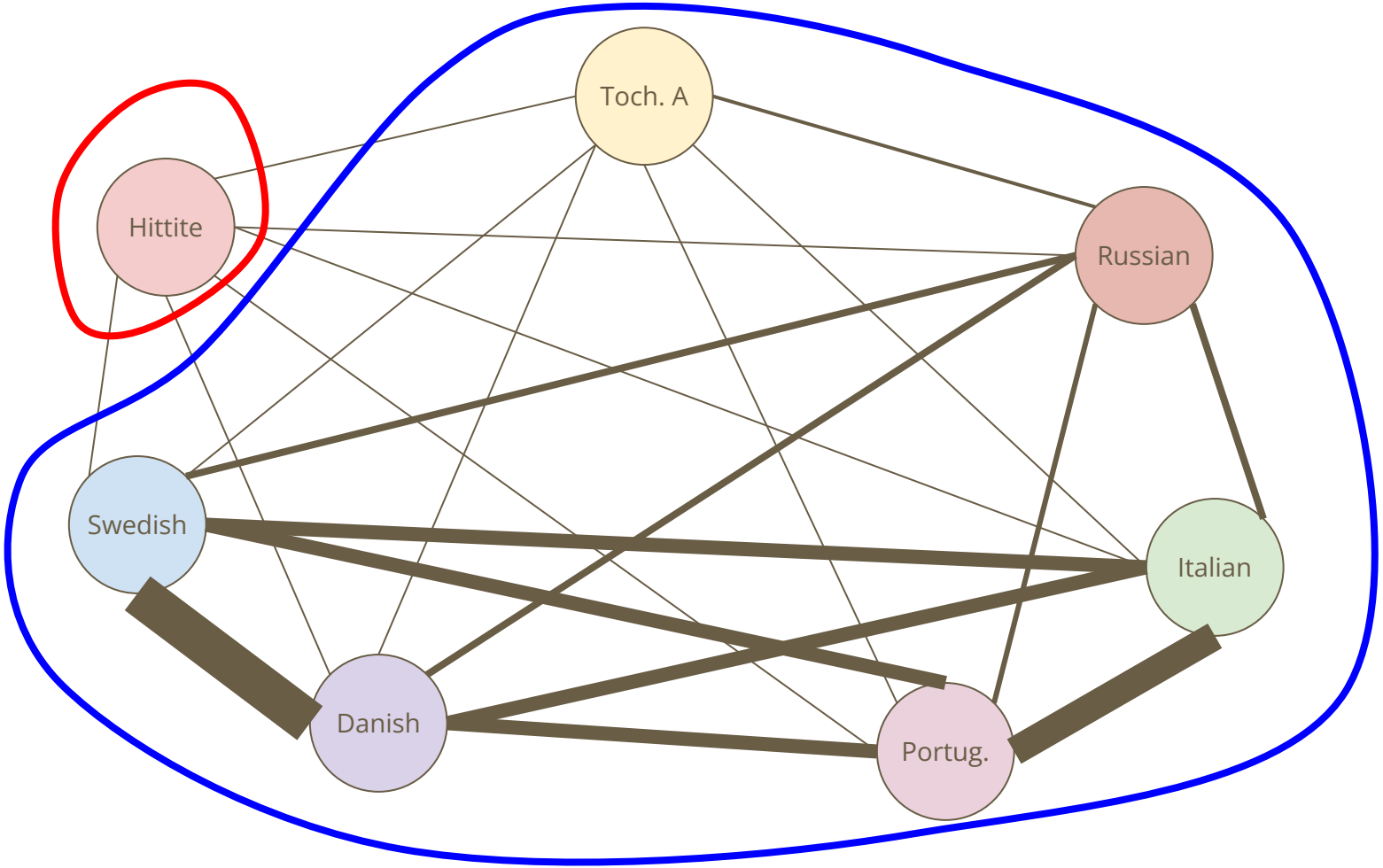
r = 0.0



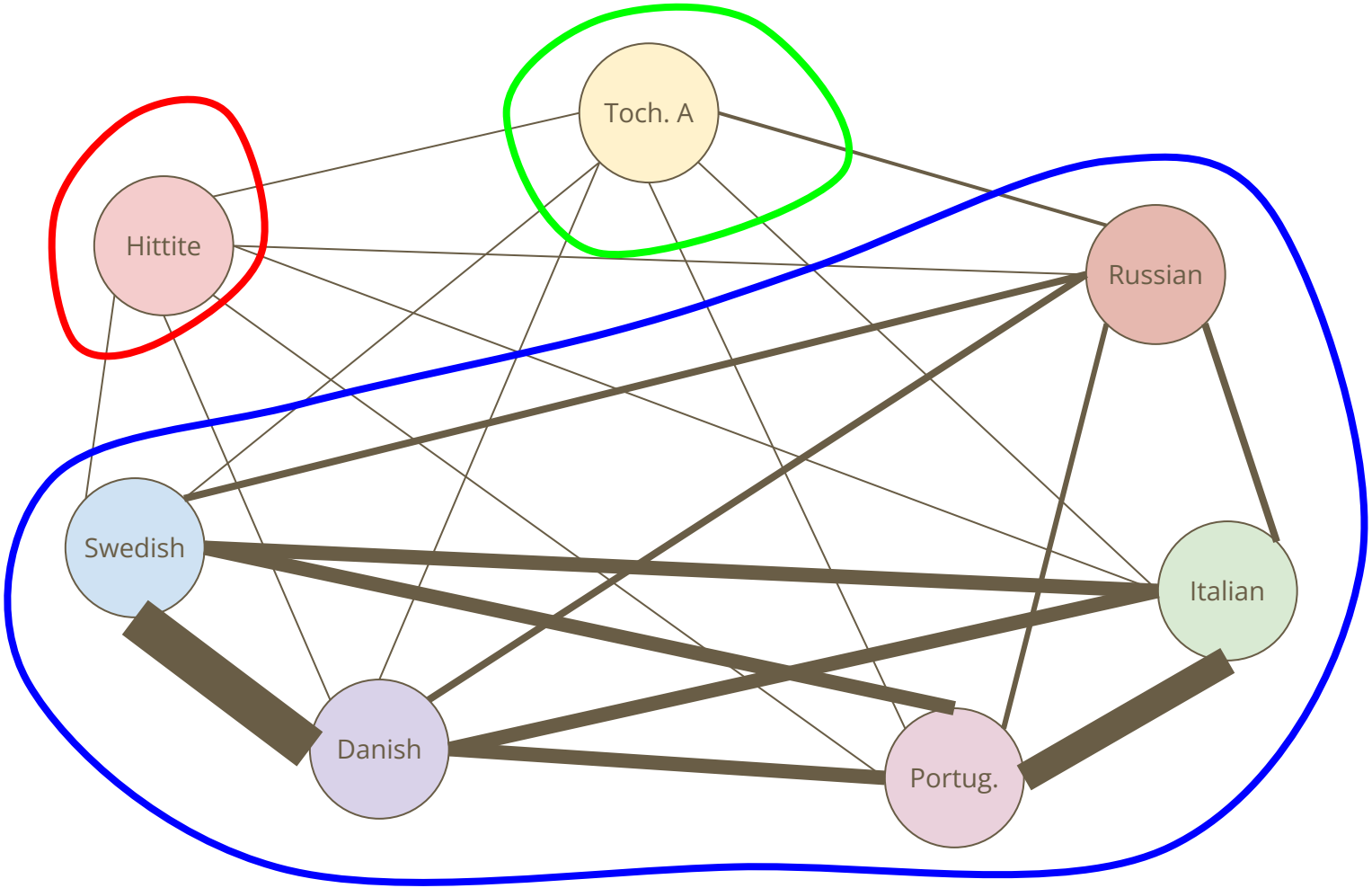
r = 0.5



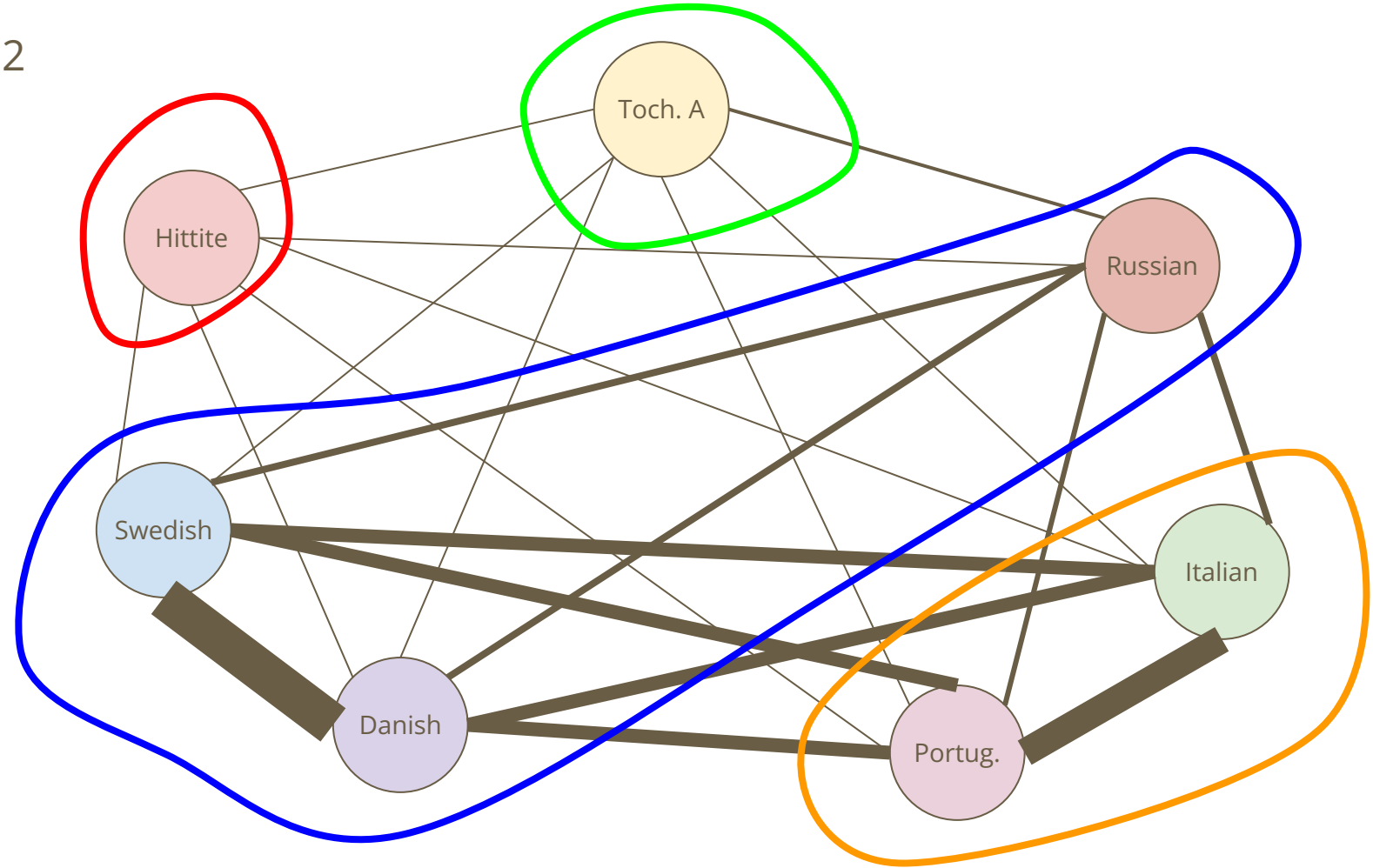
r = 0.6



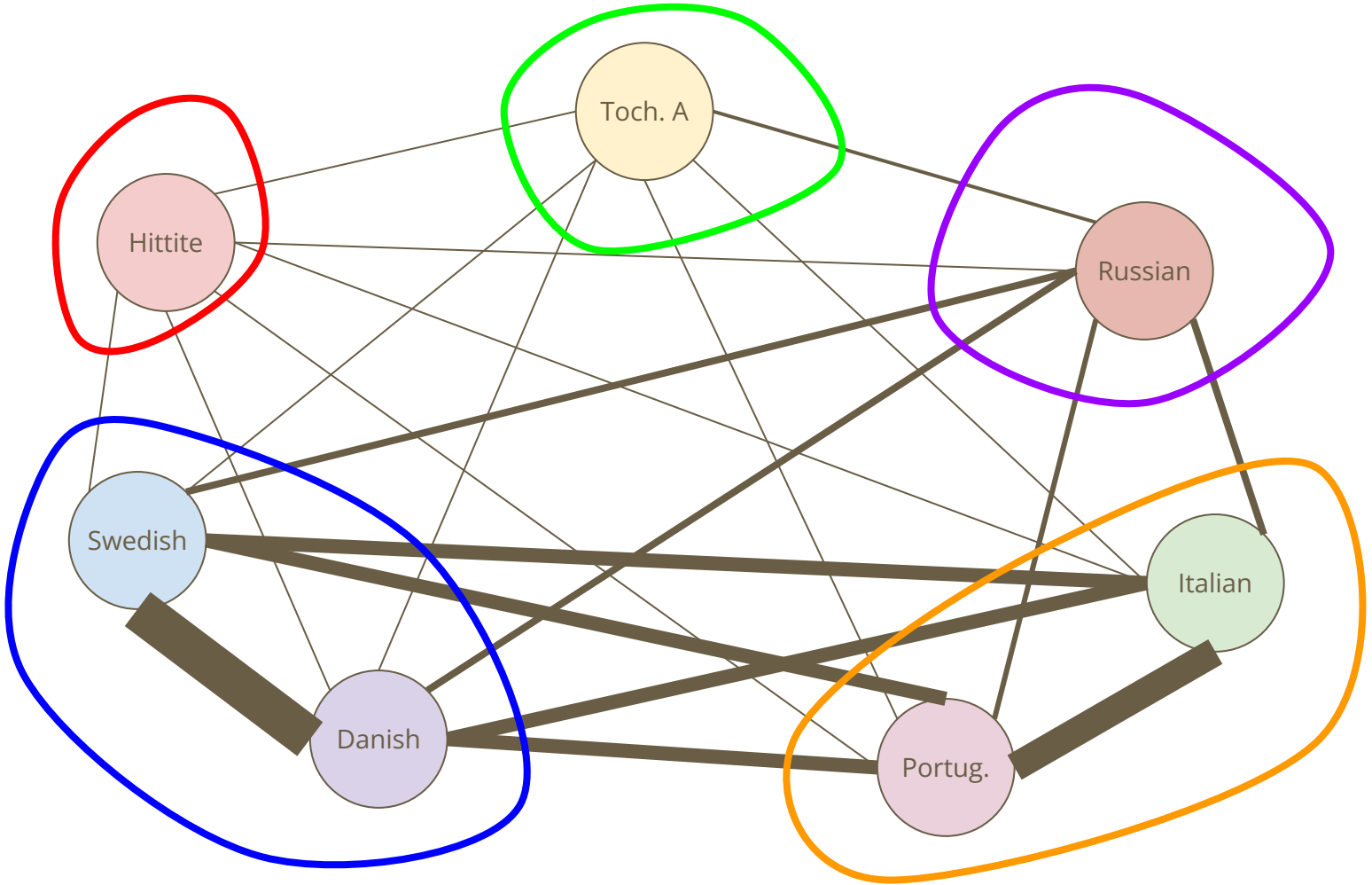
r = 0.8



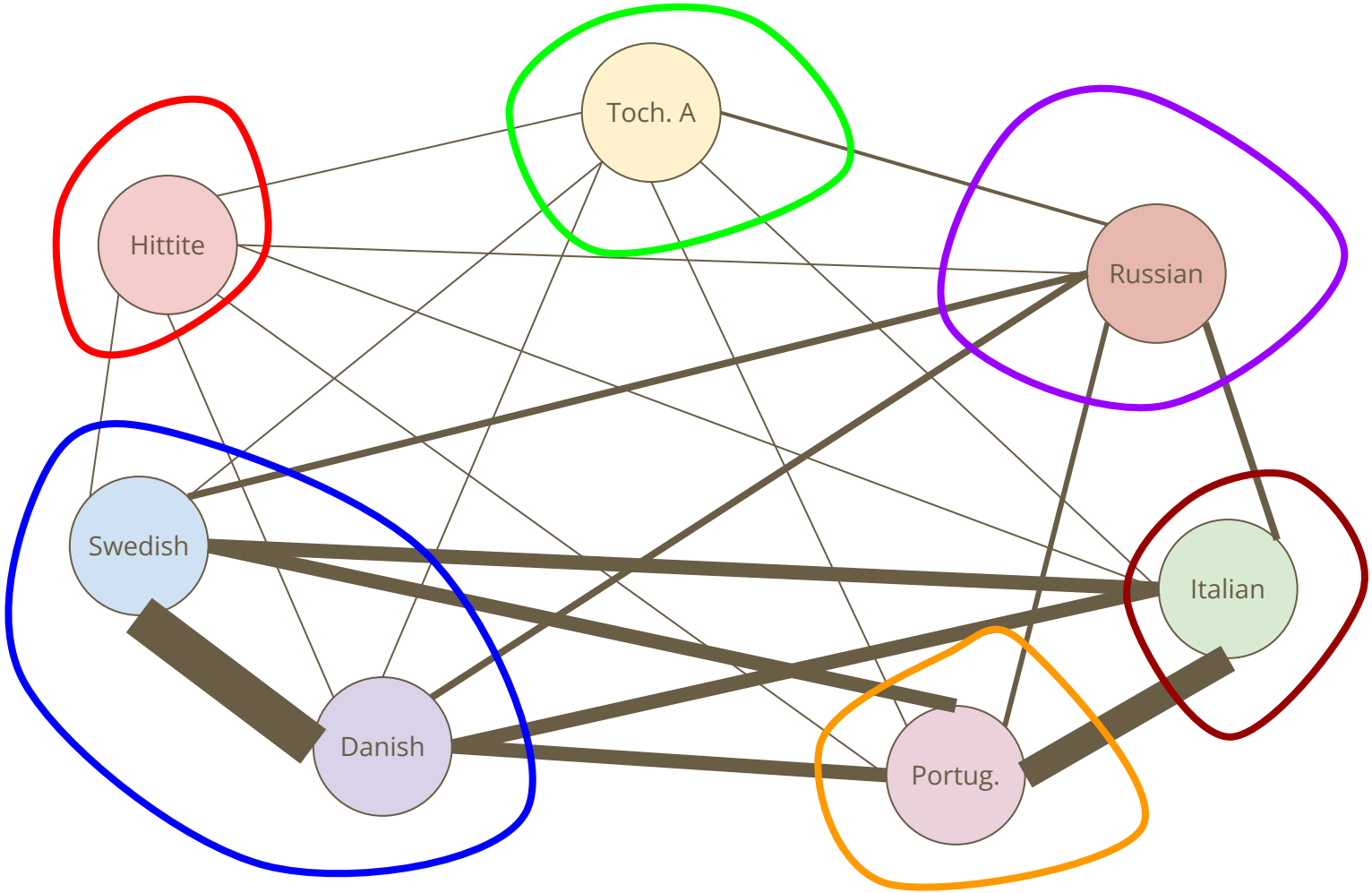
r = 1.2



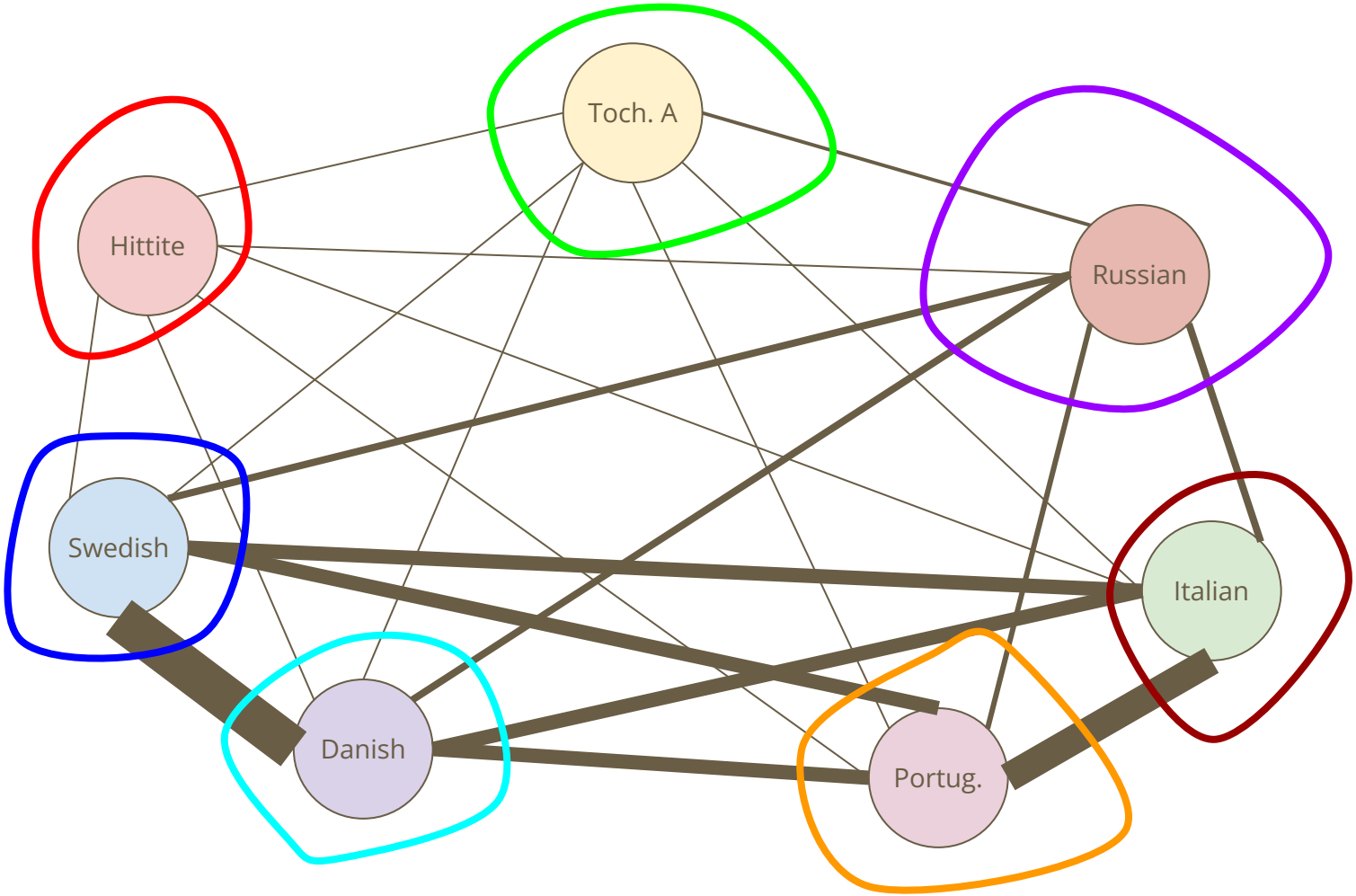
r = 2.4



r = 4.5



r = 4.9



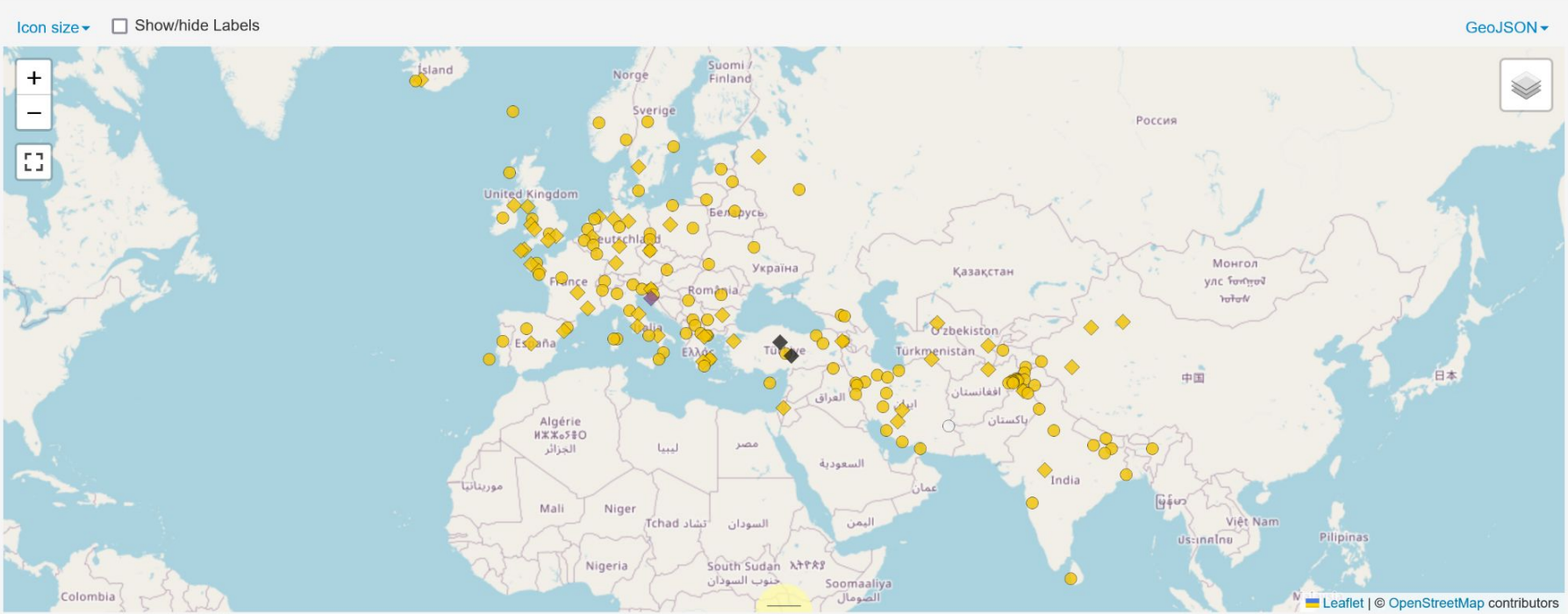
Tree building

- We can build a tree by gradually increasing the resolution and recording when the number of communities (i.e., clades) increases
  - The difference in resolution is used as a branch length
  - But not directly! It is (inversely) proportional and, depending on the method for community detection, **not** linear
- There are a number of technical difficulties
- Important question: where do the weights come from?

Weights

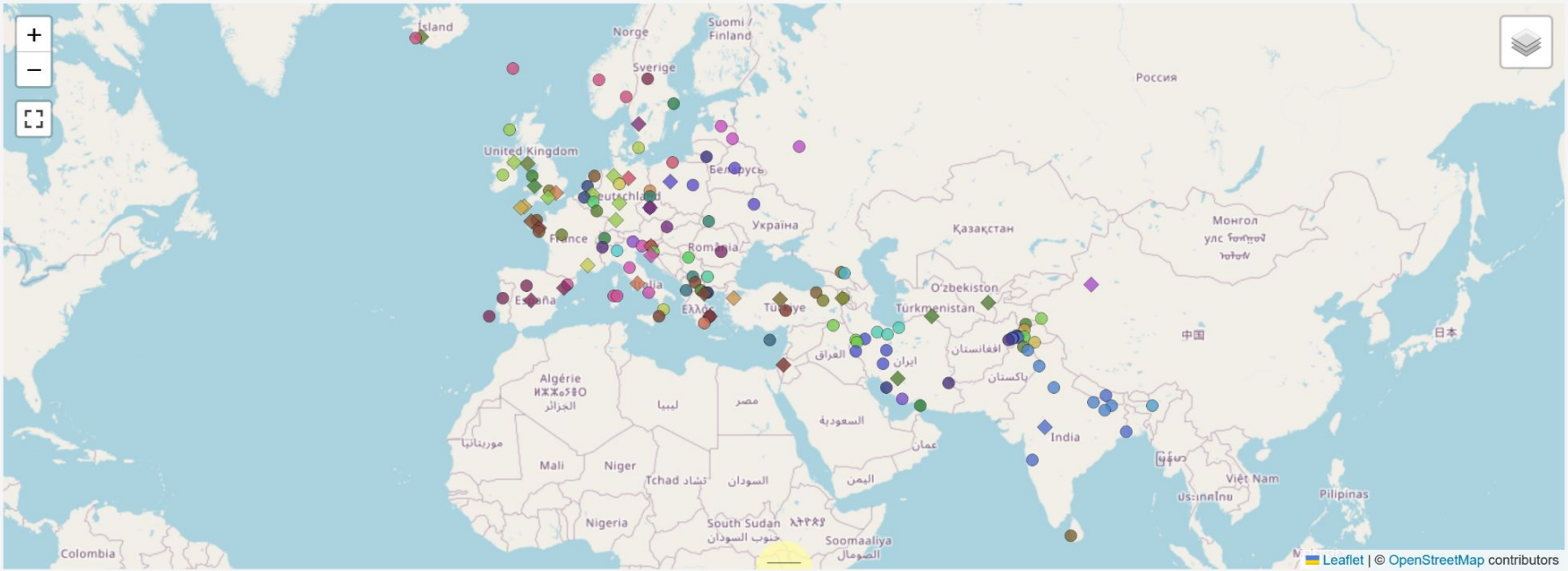
- Different strategies are possible, and I have been experimenting
- The easiest is to just add 1.0 whenever there is a shared trait (e.g. a cognate set)
  - This effectively makes the graph similar in spirit to a neighbornet
- We can also adjust the scoring by language proximity and entropy
  - This effectively addresses (*to a minimal extent!*) issues like borrowings and parallel innovations, with higher weights to shared innovations

Meaning: four



Represented in 159 languages with 4 cognate sets.

Meaning: dirty



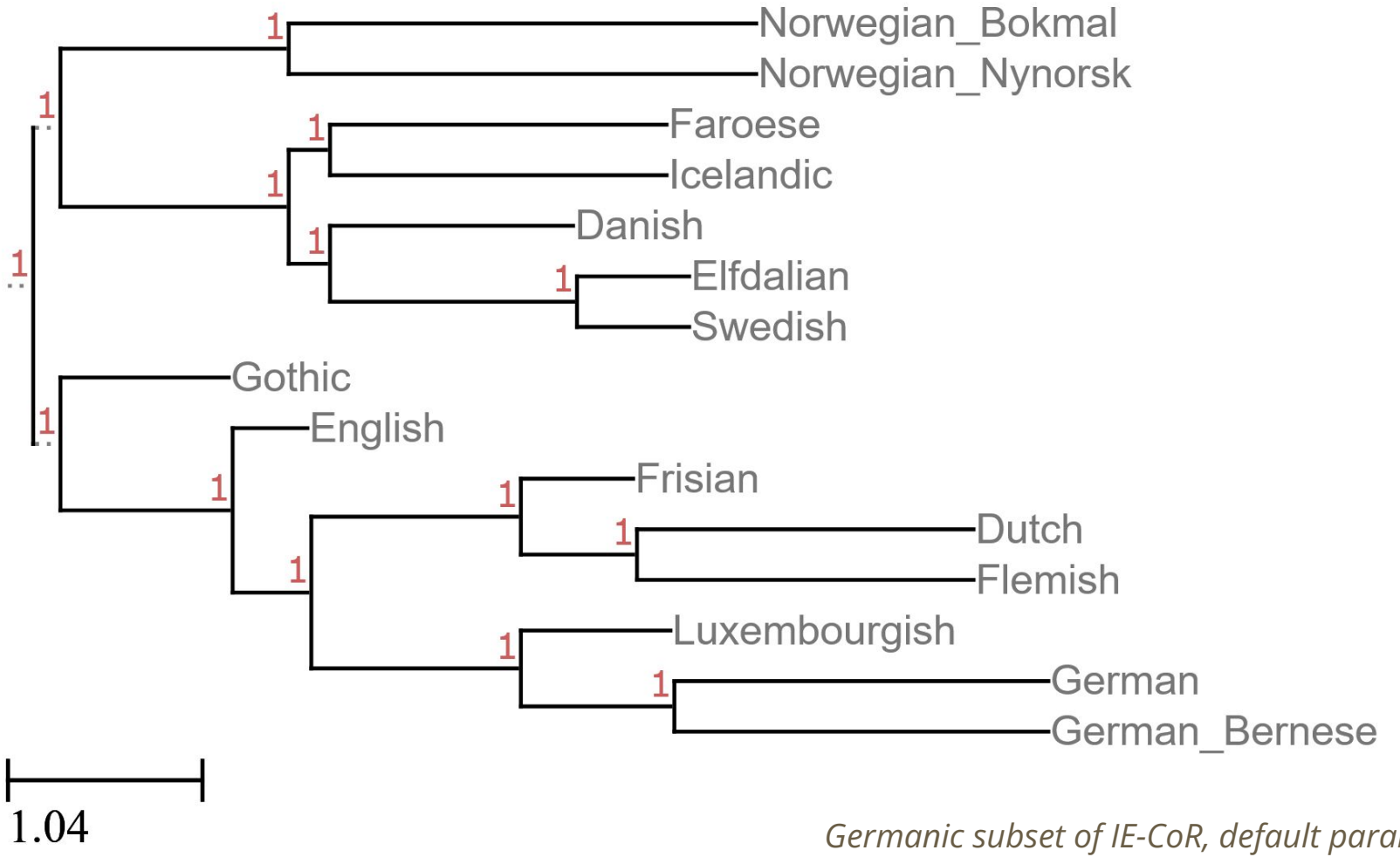
Represented in 142 languages with 83 cognate sets.

On this approach - I

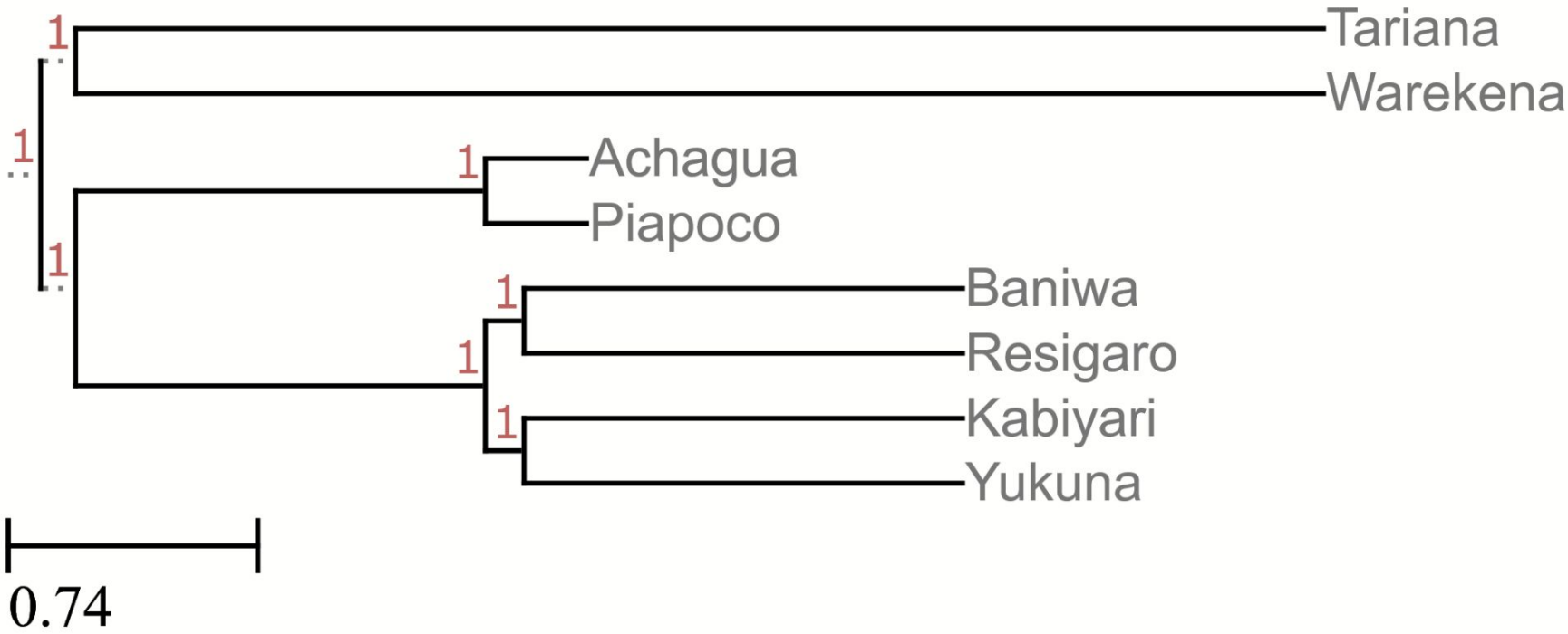
- While using characters, this approach is more like distance-methods
  - No implied evolutionary model
  - Decision ultimately based on shared material
- However, we can correct edge weights by substitution model expectancy (i.e., as in Bayesian), without mandatory symmetry, or expert judgement (e.g., Billing and Elgh [forth.])
  - This essentially incorporates an evolutionary model!
- I argue that this is still worth in our phylogenetic toolbox
  - No model is better than a bad model
  - It combines some advantages of distance- and character-methods
- The trees are rooted, and have branch length
  - We can force them to be bifurcating, but it will be more natural to have polytomies

On this approach - II

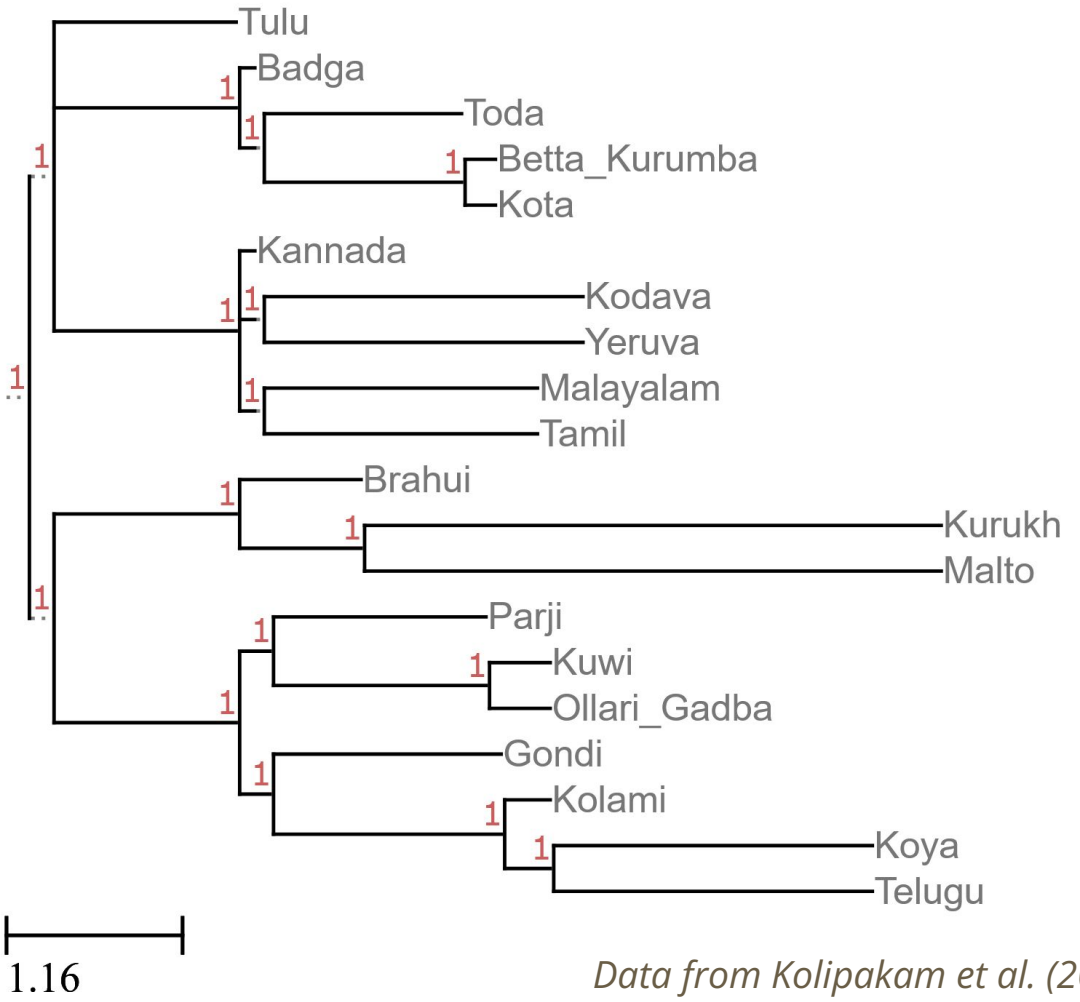
- It’s more a framework than a method
  - Different community detection methods
  - Different graph construction strategies
  - Different tree building strategies
- The following examples all use the simplest (and “less correct”) methods, all with default parameters, no calibrations, no monophyletic restrictions
  - Don’t mind branch lengths too much - I’ll explain why



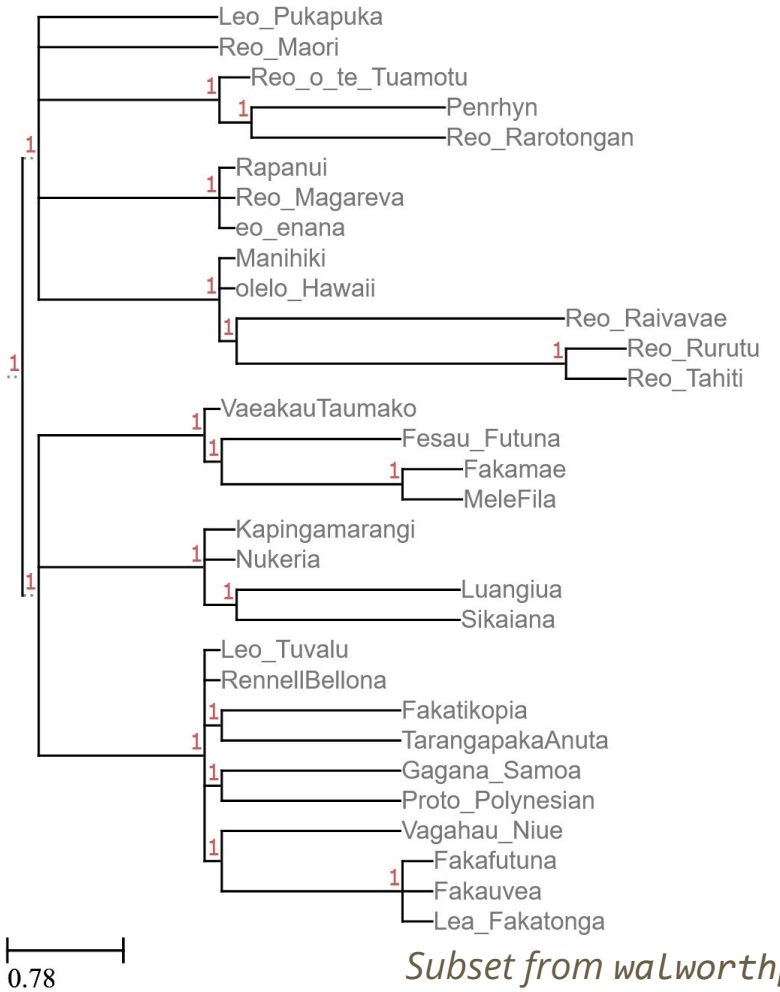
Germanic subset of IE-CoR, default parameters



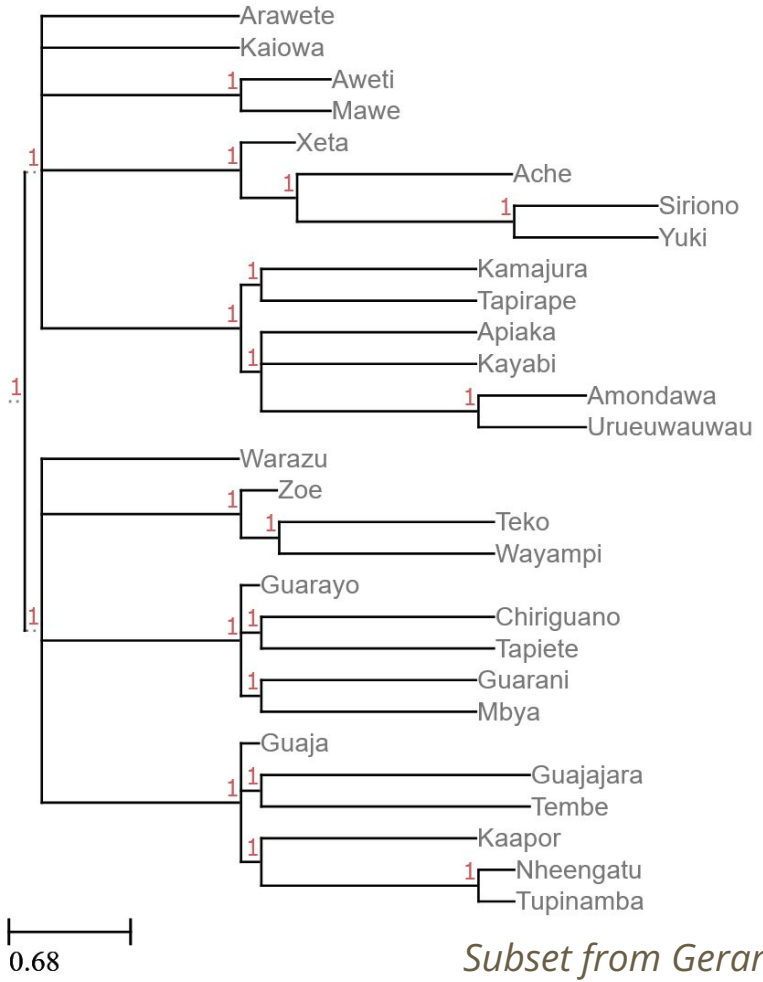
Subset of chaconarawakan, default parameters



Data from Kolipakam et al. (2018), default parameters



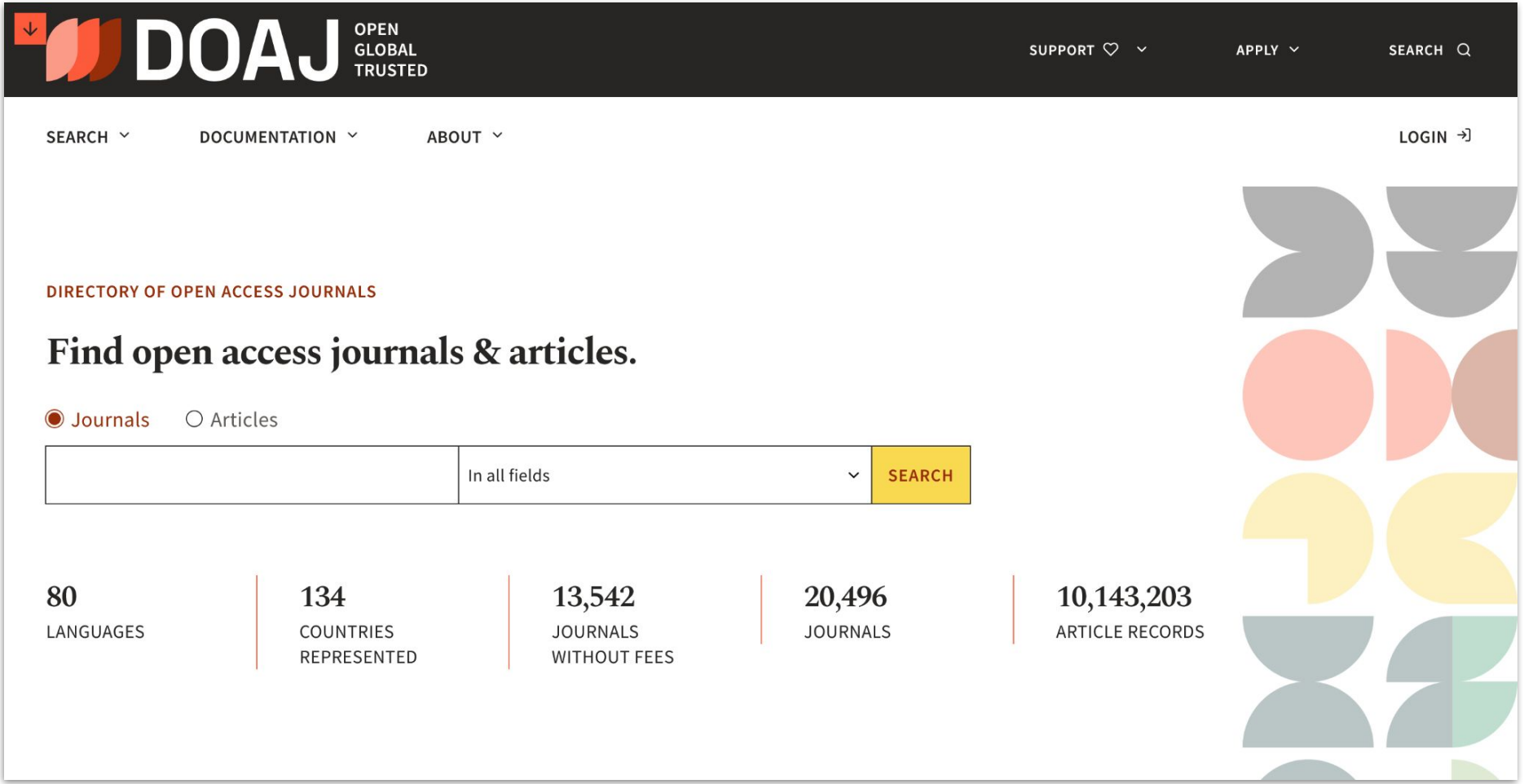
Subset from `walworthpolynesian`, default parameters



Subset from Gerardi et al. (2023), default parameters

How to proceed?

- Code on GitHub is essentially ready
  - More methods could be implemented, especially Infomap and new strategies for tree-building
- Independent researcher…
- Some free-to-publish options
  - Journal of Language Modelling (appropriate?)
  - Journal of Open Source Software (more a method than software)
  - Look for a different journal?
  - Just release with a DOI on Zenodo and be happy?


DOAJ
OPEN
GLOBAL
TRUSTED
SUPPORT
APPLY
SEARCH
SEARCH
DOCUMENTATION
ABOUT
LOGIN
DIRECTORY OF OPEN ACCESS JOURNALS
Find open access journals & articles.
Journals
Articles
In all fields
SEARCH
80
LANGUAGES
134
COUNTRIES
REPRESENTED
13,542
JOURNALS
WITHOUT FEES
20,496
JOURNALS
10,143,203
ARTICLE RECORDS

Thank you!

tresoldi@gmail.com
tiago@tresoldi.org