Computational stemmatology and the biblical text

Using artificial intelligence to reconstruct manuscript history

Sophie Robert-Hayek

Université de Lorraine
Écritures - *Maison des Sciences de l'Homme*

13/12/2024 - Lyon - HISOMA

The SHERBET project

Outline

The SHERBET project

SHERBET (Stemmatology for the HEbRew BiblE Transmission)

4 years funding (French national grant, starting in Sept. 2023) to reconstruct the genealogical linkage of Qumran and Cairo Genizah manuscripts using **computational tools**.

Consortium of biblical studies laboratory (Ecritures, Université de Lorraine), computer science laboratory (LORIA, Université de Lorraine) and applied mathematics laboratories (LJK, Université de Grenoble).

The SHERBET project

3 anticipated work packages:

1. **Benchmarking** and **calibrating stemmatology algorithms** to the Qumran and Cairo Genizah textual traditions;

The SHERBET project

3 anticipated work packages:

1. **Benchmarking** and **calibrating stemmatology algorithms** to the Qumran and Cairo Genizah textual traditions;
2. Development of **novel** computational stemmatology algorithms:
   - Using a precise probability transition model;
   - Leveraging recent advances in Natural Language Processing;
   - Outperforming current algorithms;

The SHERBET project

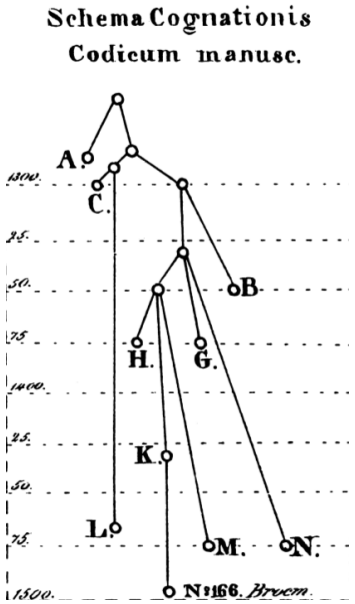
3 anticipated work packages:

1. **Benchmarking** and **calibrating stemmatology algorithms** to the Qumran and Cairo Genizah textual traditions;
2. Development of **novel** computational stemmatology algorithms:
   - Using a precise probability transition model;
   - Leveraging recent advances in Natural Language Processing;
   - Outperforming current algorithms;
3. Applications of these algorithms to **build the genealogical lineage of several traditions**, starting with Hebrew manuscripts of Ben Sira.

Computational stemmatology

Outline

Stemmatology



Stemmatology

Stemmatology consists in building the **genealogical lineage** of a set of textual witnesses by analyzing the textual **variants**, to better understand textual transformations and scribal behavior.

Usual method rely on **manual variant analysis**:

- Paul Maas conjunctive/separative errors (Maas 1958)

Computational stemmatology

The improvements in computational algorithms and computer speed has led to:

- the development of **automatic algorithms**;
- inspired from **biology**;

Computational stemmatology

The improvements in computational algorithms and computer speed has led to:

- the development of **automatic algorithms**;
- inspired from **biology**;

Computational stemmatology

Computational stemmatology uses **computational techniques and algorithms** to reconstruct the evolutionary relationships between the witnesses.

Computational stemmatology

Many algorithms have been designed over the last 50 years:

- Encoding "standard" stemmatology algorithms: Poole's algorithm (Camps 2015), RHM algorithm (Roos and Heikkila 2009) ...

Computational stemmatology

Many algorithms have been designed over the last 50 years:

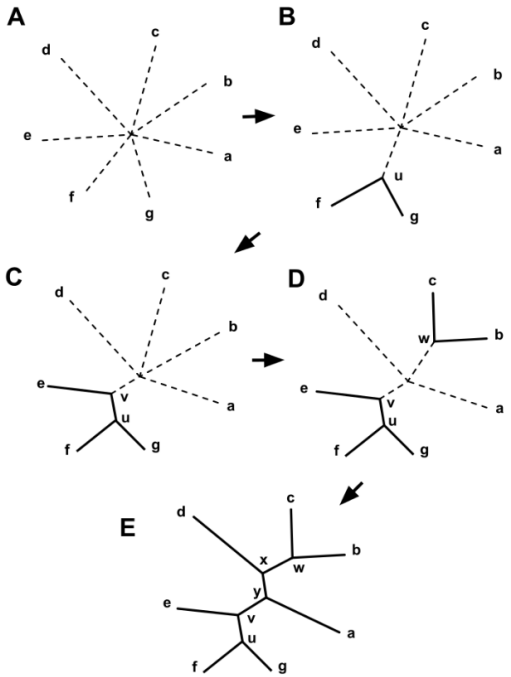
- Encoding "standard" stemmatology algorithms: Poole's algorithm (Camps 2015), RHM algorithm (Roos and Heikkila 2009) ...

- Borrowing from **philogeny** (study of the evolutionary history among organisms):
  - Distance based algorithms;
  - Probabilistic based algorithms.

Distance based algorithms

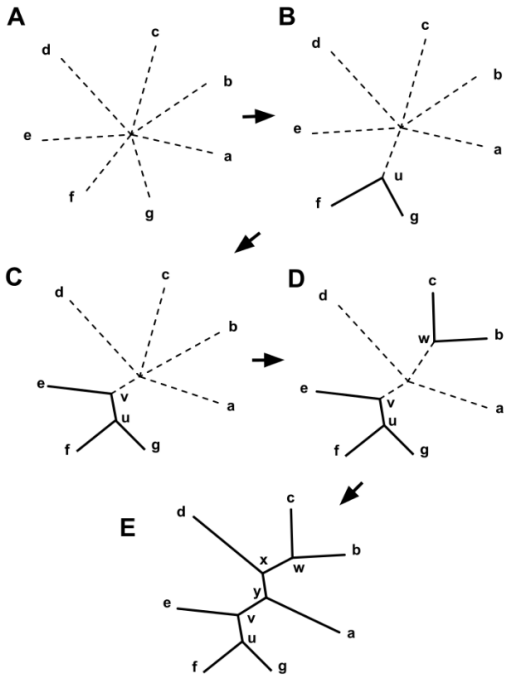
Computational stemmatology: distance based algorithms

- Define a distance **matrix** between manuscripts;
- Iteratively group together the closest manuscripts;



Computational stemmatology: distance based algorithms

- Define a distance **matrix** between manuscripts;
- Iteratively group together the closest manuscripts;



Example algorithms

UPGMA (Sokal and Michener 1958) and Neighbor Joining (Saitou N 1987).

Example of distance based approach

Manuscript	Text Variant
A	The quick brown fox jumps over the lazy dog.
B	The quick brown fox leaped over the lazy dog.
C	The quick fox jumps over the dog.
D	The brown fox jumps over the lazy dog.

Example of distance based approach

We rely on Jaccard (dis)similarity to compute the distance between manuscripts:

$$J(A,B) = 1 - \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

	A	B	C	D
A	0.00	0.20	0.22	0.11
B	0.20	0.00	0.40	0.30
C	0.22	0.40	0.00	0.22
D	0.11	0.30	0.22	0.00

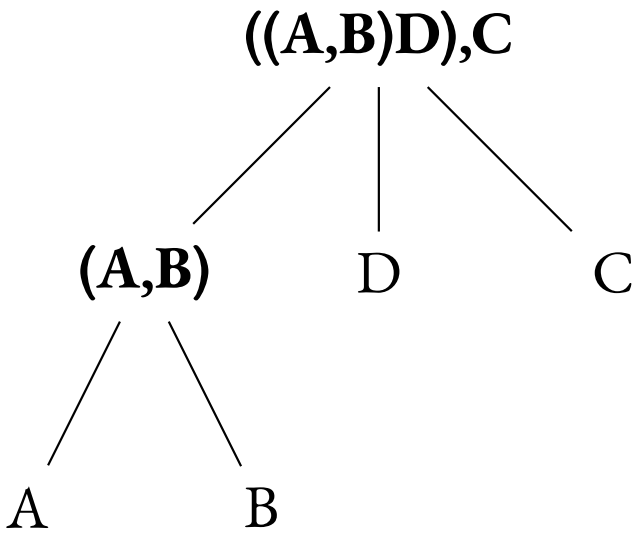
Example of distance based approach

- **Step 1**: Calculate Q-matrix from the distance matrix, with the formula:

$$Q(i,j) = (N-2) \cdot d(i,j) - \sum_k d(i,k) - \sum_k d(j,k)$$

  With:
  - $d(i,j)$: Distance between nodes i and j.
  - N: Number of nodes in the current matrix.
  - $\sum_k$: Sum of distances from nodes i and j to all other nodes.
- **Step 2**: Identify closest pair based on minimum Q-value.
- **Step 3**: Join the closest pair and recalculate distances.

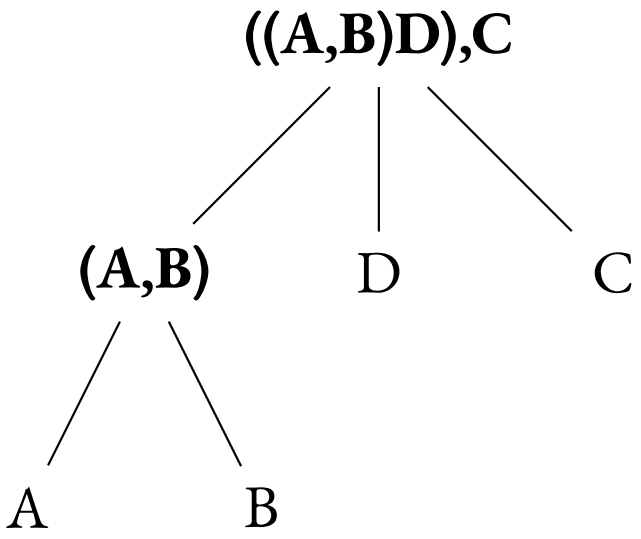
Example of distance based approach



Advantages:

- Very easy to compute;
- Gave good results on benchmarking suite.

Example of distance based approach



Advantages:

- Very easy to compute;
- Gave good results on benchmarking suite.

Main drawbacks:

- Tree is unrooted !
- Little information regarding the process of evolution.

Probability algorithms

Computational stemmatology: probability based algorithms

- Define a probability model of transition between manuscripts : probability of going from manuscript P_i to manuscript P_j given the **variants**;
- Select the tree that is the **most likely true** given the data.

The likelihood, $L(T)$ of observing the given manuscript data D, under the tree T and the tradition parameters Θ, can be calculated as:

$$L(T) = \mathbb{P}(D|T, \Theta)$$

Computational stemmatology: probability based algorithms

- Define a probability model of transition between manuscripts : probability of going from manuscript P_i to manuscript P_j given the **variants**;
- Select the tree that is the **most likely true** given the data.

The likelihood, $L(T)$ of observing the given manuscript data D, under the tree T and the tradition parameters Θ, can be calculated as:

$$L(T) = \mathbb{P}(D|T, \Theta)$$

Example algorithms

Bayesian Inference (Drummond and Bouckaert 2015), Maximum Likelihood trees (Felsenstein 1981) ...

Computational stemmatology: probability based algorithms

Advantages:

- State of the art algorithms in phylogeny;
- Allows for estimation of parameters.

Main drawbacks:

- Very long to compute.

Computational stemmatology: probability based algorithms

Advantages:

- State of the art algorithms in phylogeny;
- Allows for estimation of parameters.

Main drawbacks:

- Very long to compute.

To be added to the benchmarking studies!

Computational stemmatology: Natural Language Processing

A novel approach will consider **the probability using variant classification**.

The probability of a text transforming into another will **depend on the modelization of the scribal behavior**.

Computational stemmatology: Natural Language Processing

A novel approach will consider **the probability using variant classification**.

The probability of a text transforming into another will **depend on the modelization of the scribal behavior**.

Variant rates must be estimated (**Mathematically tricky!**):

- Before fitting the model;
- From the estimation performed by probabilistic approaches.

Computational stemmatology: Natural Language Processing

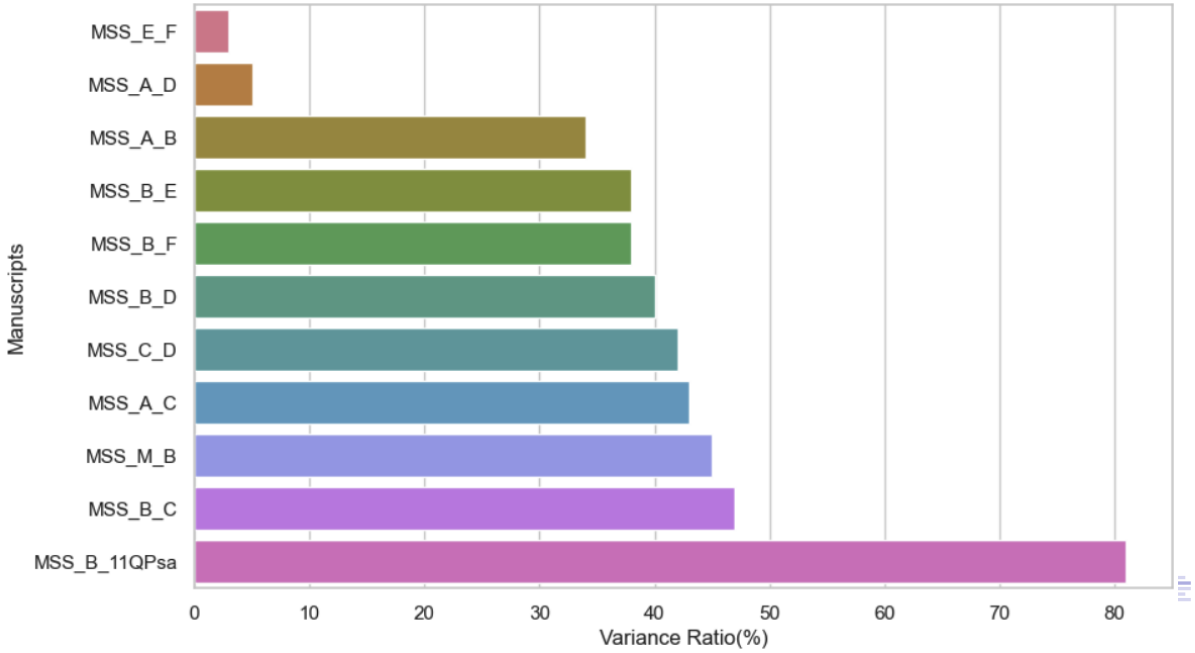
6 months of effort for tagging of all the variants observed between 8 manuscripts of Ben Sira, according to the following categories:

- Lexical
- Morphological
- Plus/minus
- Inversion
- Unclassifiable

Computational stemmatology: Natural Language Processing

We have calculated the rate of variance between two manuscripts according to the following ratio:

$$\frac{variant\ locations}{variant\ locations + equivalences}$$



How to select the right algorithm?

Outline

Selecting the right algorithms

Faced with as many possible choices...

Selecting the right algorithms

Faced with as many possible choices...

What algorithm should we select given a textual tradition ?

Selecting the right algorithms

Faced with as many possible choices...

What algorithm should we select given a textual tradition ?

There is no single optimum algorithms that **will outperform all others** and the algorithms **should be selected given the particularity of each tradition**.

(Machine Learning/Deep Learning community refers to this as the "no free lunch" theorem)

Benchmarking of stemmatology algorithms

Benchmarking

Benchmarking refers **to the process of evaluating the performance** of a new model, algorithm, or technique by comparing it against established and standardized datasets, metrics, or existing models.

Benchmarking of stemmatology algorithms

Benchmarking

Benchmarking refers **to the process of evaluating the performance** of a new model, algorithm, or technique by comparing it against established and standardized datasets, metrics, or existing models.

Benchmarking is required to:

- Suggest new algorithms and compare them to the state of the art;
- Select the optimum algorithm given the particularity of a tradition.

Why should we benchmark stemmatology algorithms ?

Suggesting a new algorithm: A new variation of algorithms should **perform at least as well** on at least **one case study** to be an acceptable.

Why should we benchmark stemmatology algorithms ?

Suggesting a new algorithm: A new variation of algorithms should **perform at least as well** on at least **one case study** to be an acceptable.

Example

When suggesting to use a **new distance** in a distance based stemmatology algorithms such as Neighbor Joining, we should show that in practice it outperforms other textual distances to be **suggested as an alternative**.

Why should we benchmark stemmatology algorithms ?

Selecting the algorithm:

Select the best performing algorithms given **the characteristics of the variants within the studied textual tradition**.

Why should we benchmark stemmatology algorithms ?

Selecting the algorithm:

Select the best performing algorithms given **the characteristics of the variants within the studied textual tradition**.

Example

Given a tradition with a :

- 1% plus/minus rate;
- 5% morphological change;
- 2% word inversion;
- 10% of missing manuscripts;

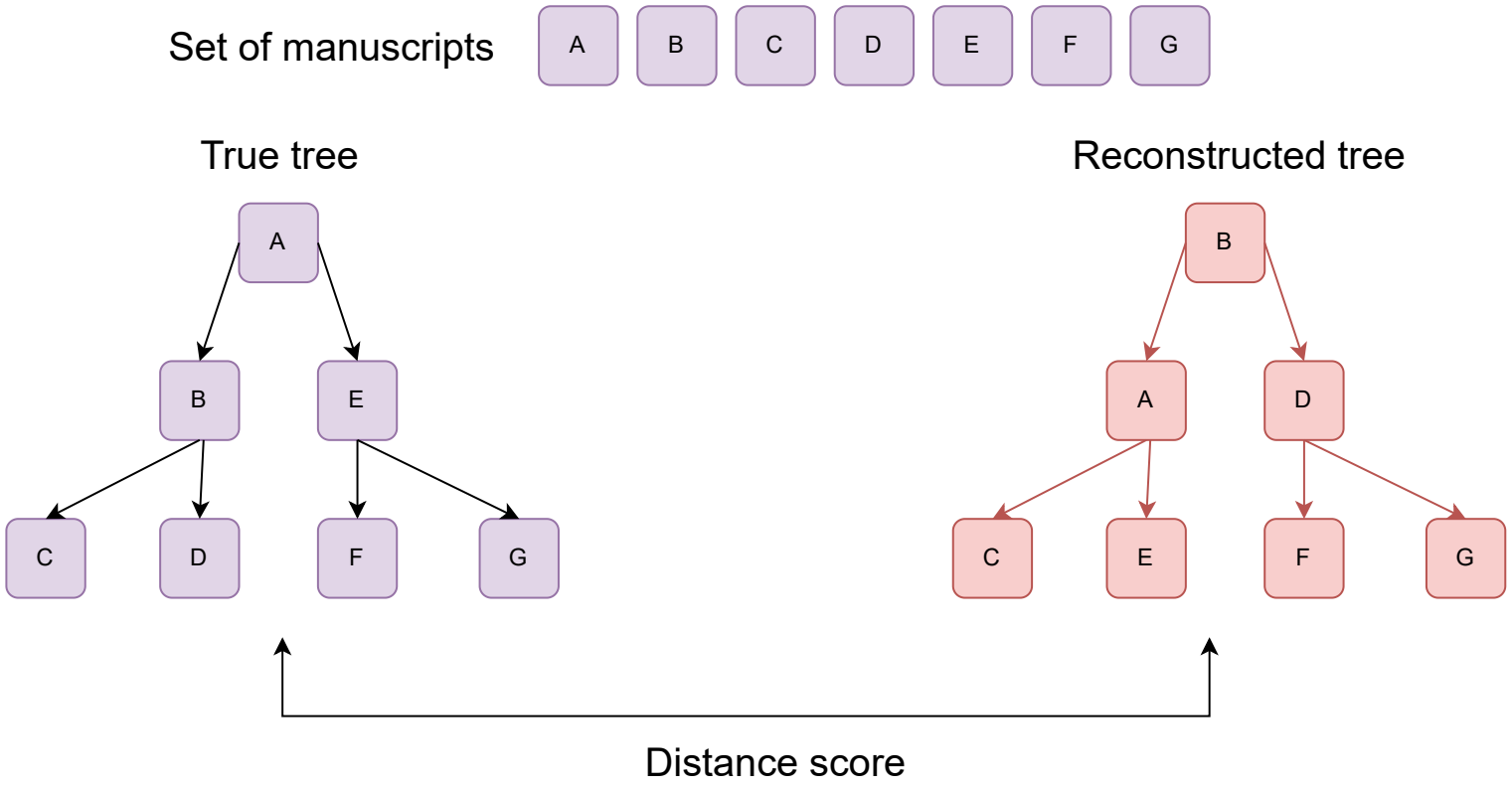
probability at each generation,
What algorithm should I select given these characteristics ?

Benchmarking of stemmatology algorithms

To perform benchmarking we need:

1. A golden standard (**ground truth**): a tradition where we know the true stemma;
2. A set of stemmatology algorithms to compare;
3. A metric to compare the different results between them.

Benchmarking of stemmatology algorithms



Benchmarking of stemmatology algorithms

Two possible approaches:

Benchmarking of stemmatology algorithms

Two possible approaches:

- Use "real" **handwritten benchmarking data** or traditions with known ground truth.

Benchmarking of stemmatology algorithms

Two possible approaches:

- Use "real" **handwritten benchmarking data** or traditions with known ground truth.
- Use **computer generated traditions** that imitates observed traditions and simulates variants over time.

Using handwritten data

Landmark study of Roos et al. (Roos and Heikkila 2009) that compare 22 different variations of stemmatology algorithms on 4 traditions (3 synthetic, 1 "real"):

Data	**Number of manuscripts**
Heinrichi	67
Parzival	21
Notre besoin	14
Legend	52*

Using handwritten data (Roos and Heikkila 2009)

	Data		
Method	***Heinrichi* (%)**	***Parzival* (%)**	***Notre besoin* (%)**
RHM	**76.0**	79.9	76.9
PAUP*			
Parsimony	74.4	77.8	74.5
Parsimony BS[b]	73.6	85.4	77.3
Neighbour Joining	64.4	81.5	76.2
Neighbour Joining BS[b]	62.9	**87.1**	77.4
Least squares	64.2	81.5	70.2
Least squares BS[b]	62.6	79.8	73.0
n-Gram clustering	64.4	79.3	66.4
SplitsTree4			
NeighborNet	59.1	77.8	70.2
SplitDecomp.	53.1	74.5	73.1
ParsimonySplits	56.8	83.7	71.6
CompLearn	52.7	81.5	70.6
Hierarchical clustering	51.4	72.6	60.2
'Classical' method A[a]			74.4
'Classical' method B[a]			**85.1**
Weighted support method			66.3
Neighbour joining A			76.0
Neighbour joining B			75.0
Parsimony			74.4
Data compression			62.0

Limits

Handwritten texts in conditions that resemble the working conditions of scribes, but:

Limits

Handwritten texts in conditions that resemble the working conditions of scribes, but:

1. Very expensive;
2. No guarantee of being representative;
3. Very dependent on experimental parameters;
4. Hard to fine tune.

Benchmarking of stemmatology algorithms

Suggestion of a complementary approach **based on simulation** to generate **representative textual traditions**.

Benchmarking of stemmatology algorithms

Suggestion of a complementary approach **based on simulation** to generate **representative textual traditions**.

We present the **StemmaBench** Python library, a set of utilities to **generate ground truth traditions for benchmarking of stemmatology algorithms**.

Computer generated traditions

Outline

Purpose

StemmaBench is a Python library that allows you to quickly generate an artificial textual traditions given:

- An **input text**;
- A **configuration file**.

Example

Input text: Extract from *the Lion, The Witch and The Wardrobe*, by C.S. Lewis

> *Once there were four children whose names were Peter, Susan, Edmund and Lucy.*

Configuration file:

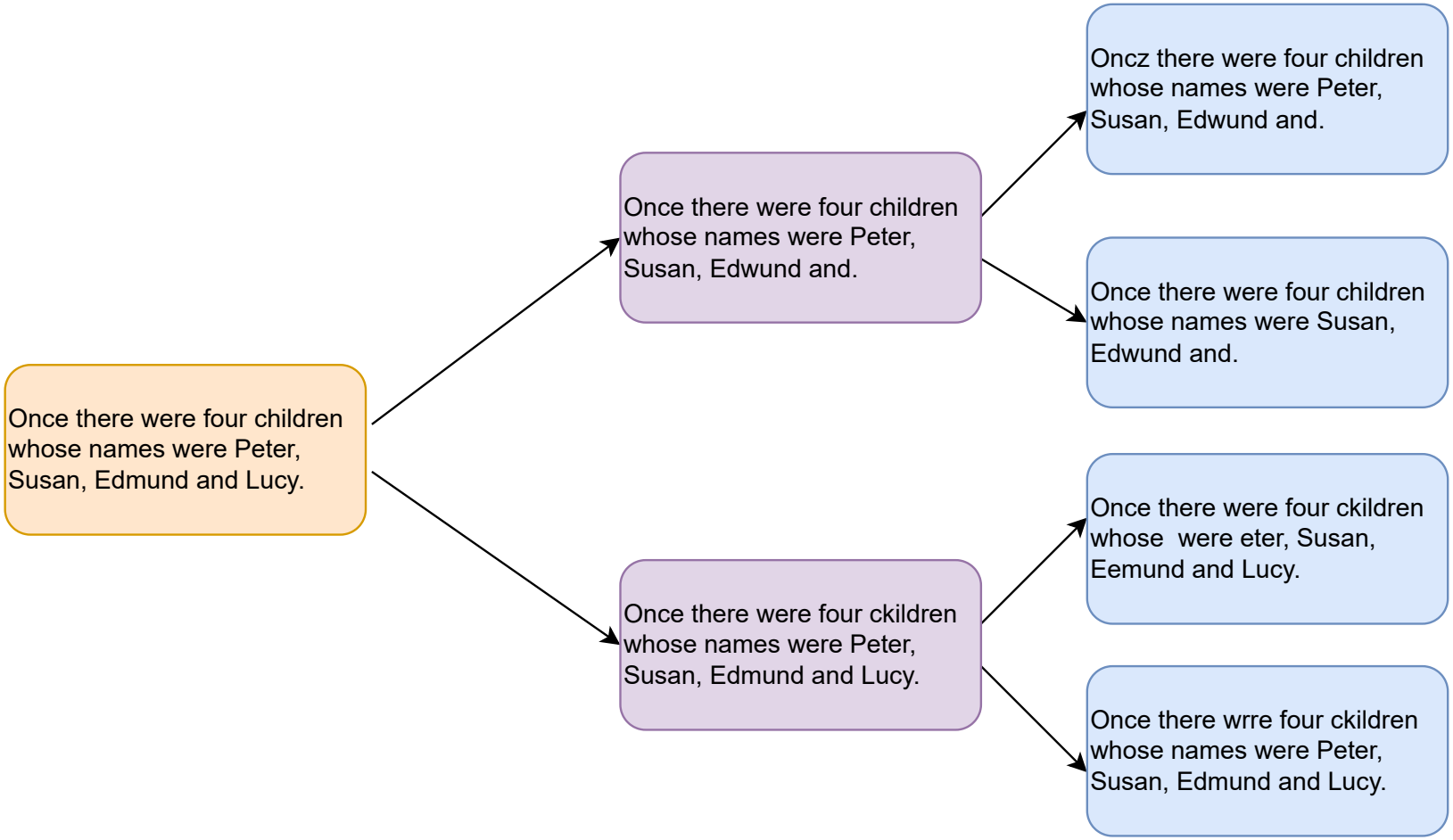
```yaml
meta:
  language: eng

variants:
  words:
    synonym:
      law: Bernouilli
      rate: 0.01
    mispell:
      law: Bernouilli
      rate: 0.1
    omit:
      law: Bernouilli
      rate: 0.05
  sentences:
    duplicate:
      args:
        nbr_words: 2
      law: Bernouilli
      rate: 0.1

stemma:
  depth: 2
  width:
    law: Uniform
    min: 2
    max: 4
```

Example

StemmaBench will generate the following artificial tradition:



Using stemmabench

Input:

- **Define the text**: `input_text.txt`
- **Define the wanted configuration**: `config.yml`

```
generate narnia.txt output_folder config.yaml
```

Using stemmabench

Input:

- **Define the text**: `input_text.txt`
- **Define the wanted configuration**: `config.yml`

```
generate narnia.txt output_folder config.yaml
```

Output:

- `edges.txt`: A display of the generated tree.
- A text file per generated manuscript within the tradition.

Supported languages

For now, supported languages for synonym generation are:

- English;
- Koiné Greek.

Incoming supported language: **biblical hebrew**.

3 improvement direction

- **Improving** scribal modelization;
- **Estimating** the parameters of the tradition;
- Dealing with **contamination** / horizontal transmission.

Benchmarking of stemmatology algorithms

To perform benchmarking we need:

1. A golden standard (ground truth): a tradition where we know the true stemma;
2. A set of stemmatology algorithms to compare;
3. **A metric to compare the different results between them.**

Many metrics exist in computer science and network based analysis.

Comparing metrics from a philologist point of view

Outline

Metric for comparison

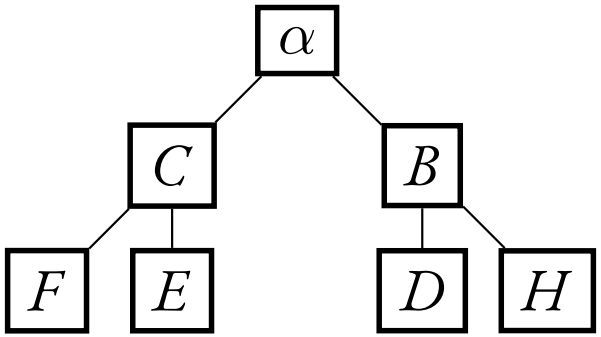
Just like in **classification** problems, **the selected metric for benchmarking** depends on the task and the context:

- Accuracy (capacity to accurately predict the right class);
- Recall (conservative towards false negative);
- Precision (conservative towards false positive);

Depending on the **wanted focus** of the *stemma codicum*, **different scoring metrics can be used**.

Reference stemma

Let's consider the **following stemma**:



The Role of the Stemma Codicum in Lachmannian Textual Criticism

Lachmannian Approach

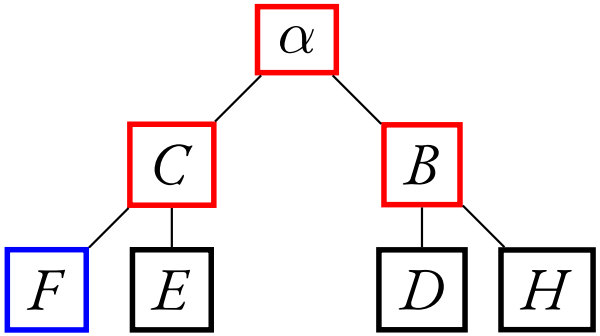
In the **Lachmannian/Maas** method, the *stemma codicum*:

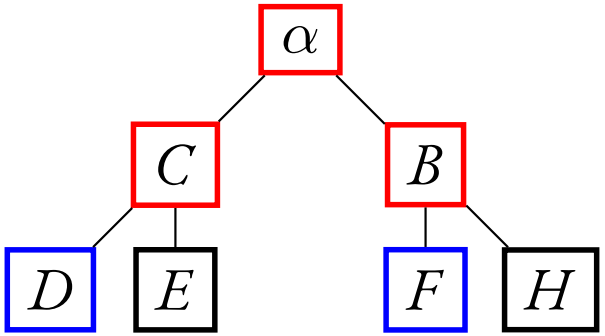
- **Reconstructs the textual history**: traces the relationships among surviving manuscripts to deduce the structure of the transmission history.
- **Identifies common ancestors**: reveals hypothetical ancestors (archetypes) that served as sources for groups of copies.
- **Eliminates secondary witnesses**: Once relationships are clear, **manuscripts that do not contribute unique information can be excluded from the critical edition**.

The Role of the Stemma Codicum in Lachmannian Textual Criticism

The *stemma codicum* works backwards to recover the **closest possible approximation** of the **original text**, for its **reconstruction**.

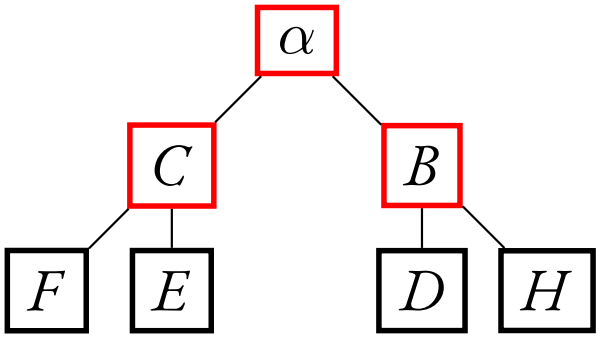
Misplacing secondary witnesses (=leaves) should lead to ***a lower penalty****::*

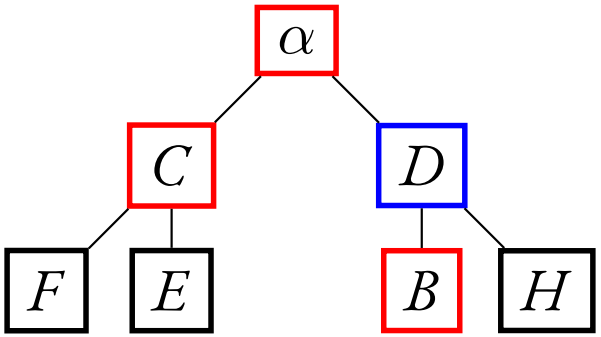




The Role of the Stemma Codicum in Lachmannian Textual Criticism

Misplacing primary witnesses (=nodes) should lead to ***a higher penalty****:*





The Role of the Stemma Codicum in Philology

New Philology Approach

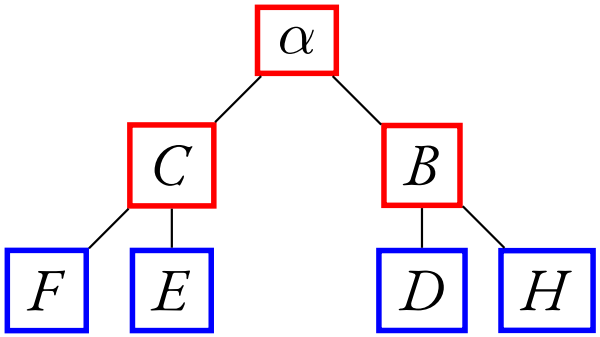
Another possibility is to shift the focus of the *stemma codicum*:

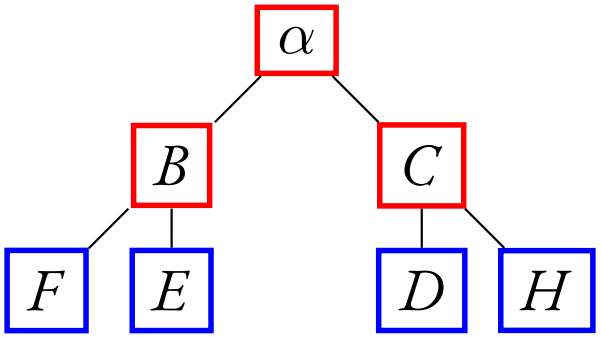
- **Emphasizes variation over hierarchy**: Value variations across manuscripts as meaningful evidence of cultural and historical contexts, rather than single "original" text.
- **Highlights textual plurality**: each manuscript is seen as a unique witness.
- **De-emphasizes a singular archetype**: not trace all copies to a single ancestor but to **explore the transmission and diversity of textual forms over time**.

The Role of the Stemma Codicum in Philology

The *stemma codicum* allows to appreciate the **dynamic and fluid nature** of textual transmission, acknowledging each manuscript's role in the understanding of the evolution of the life of the text.

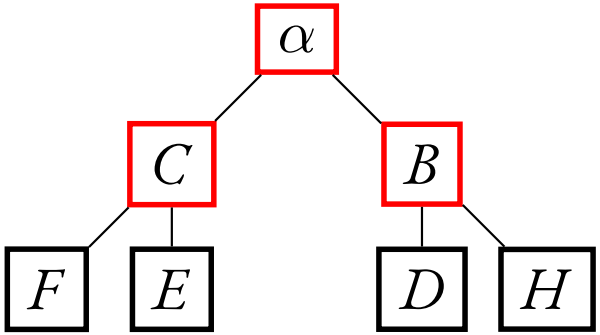
Misplacing secondary witnesses (=leaves) should lead to ***a stronger penalty****:*

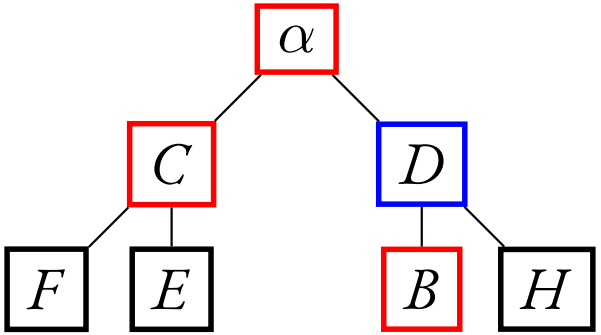




The Role of the Stemma Codicum in New Philology

Misplacing secondary witnesses (=nodes) should be as ***penalized*** *as misplacing primary witnesses*:





Comparing metrics from a philologist point of view

Selected metrics for benchmarking should take into account these considerations ...
Let's see how these affect Roos et al. study!

Metrics for stemma comparison

Outline

Mathematical Representation of a Stemma

Directed Acyclic Graph (DAG) Structure

We consider a stemma as a **Directed Acyclic Graph (DAG)**:

- Let $S = (V, E)$ be a stemma, where:
  - V is the set of vertices (or nodes), representing individual manuscripts or textual witnesses.
  - $E \subseteq V \times V$ is the set of **directed edges**, representing transmission relationships.
- Let $\mathcal{S}$ the space of considered *stemmata*.

Distances between stemmata

- **Goal:** Define a metric $d(G_1, G_2)$ for graphs G_1 and G_2 in the space of stemmata (or graph space).
- **Metric definition:** A function $d : \mathcal{S} \times \mathcal{S} \to \mathbb{R}_+$ is a distance metric on stemma space $\mathcal{S}$ if:
  1. $d(S_1, S_2) \geq 0$ (non-negativity),
  2. $d(S_1, S_2) = 0$ if and only if $S_1 = S_2$ (identity of indiscernibles),
  3. $d(S_1, S_2) = d(S_2, S_1)$ (symmetry),
  4. $d(S_1, S_3) \leq d(S_1, S_2) + d(S_2, S_3)$ (triangle inequality).
- If the **triangle inequality** is relaxed, the function becomes a **similarity**.

Roos and Heikkila's similarity

Major study uses the **conservation of tree structure**:

- $d(A, B)$: Distance (shortest path) between nodes A and B in a given stemma.
- $d'(A, B)$: Distance between A and B in the correct reference stemma.

For each triplet (A, B, C), calculate $u(A, B, C)$:

$$u(A, B, C) = 1 - \frac{1}{2} \left| sign(d(A, B) - d(A, C)) - sign(d'(A, B) - d'(A, C)) \right|$$

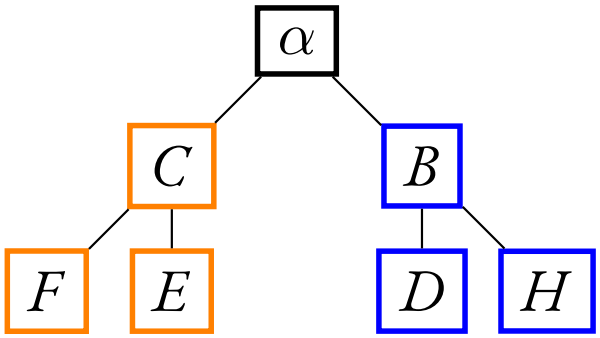
Roos and Heikkila's similarity

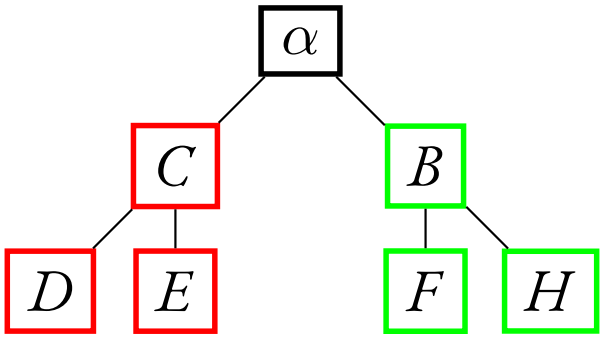
The total score S is the sum over all possible triplets:

$$S = \sum_{i=1}^{N} u(A_i, B_i, C_i)$$

where N is the number of triplets.

Roos and Heikkila's similarity





$$Sim(S_1, S_2) = 0.34$$

Robinson-Foulds (RF) metric

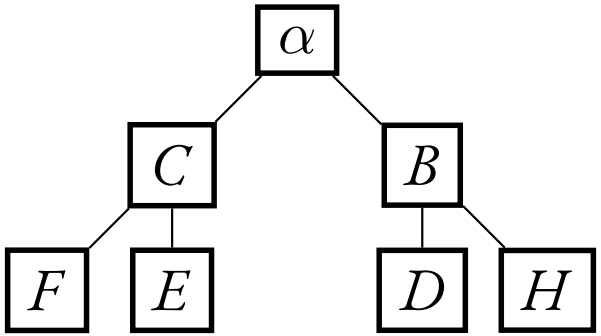
A **split** of an undirected graph is a cut whose cut-set forms a **complete bipartite graph**.

In the case of a tree, divides the set of taxa into two disjoint groups A and B by cutting a branch, so that:

- $A \cap B = \emptyset$
- $A \cup B = V(G)$
- Both A and B are non-empty sets.

Robinson-Foulds (RF) metric

Considering the previous stemma,



splits (considering internal nodes) are:

- (C, F, E);
- (B, D, H);

Robinson-Foulds' metric

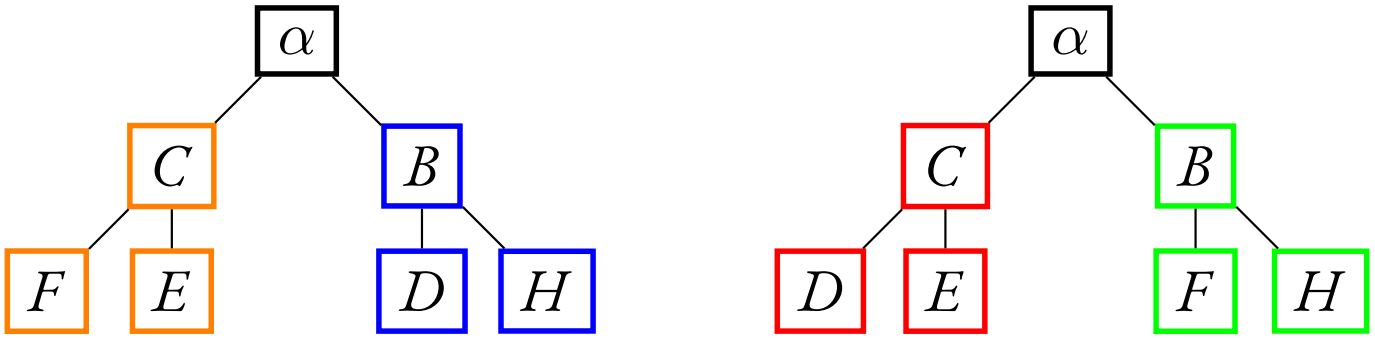
The **Robinson-Foulds (RF) distance** is a measure of distance between two trees based on the splits they contain, by counting the cardinality of the symmetric difference of the splits (=bipartitions) of trees:

$$\mathrm{RF}(S_1, S_2) = \frac{1}{2}(|S_1 - S_2| + |S_2 - S_1|)$$

where S_1 and S_2 are the sets of splits in each tree, and $|S_1 - S_2|$ represents splits unique to S_1 and vice versa.

- **Common splits**: Splits that appear in both trees contribute to the similarity between the trees.
- **Unique splits**: Splits that appear in only one of the two trees increase the RF distance.

Robinson-Foulds' metric



$$RF(S_1, S_2) = 2$$

Graph edit distance

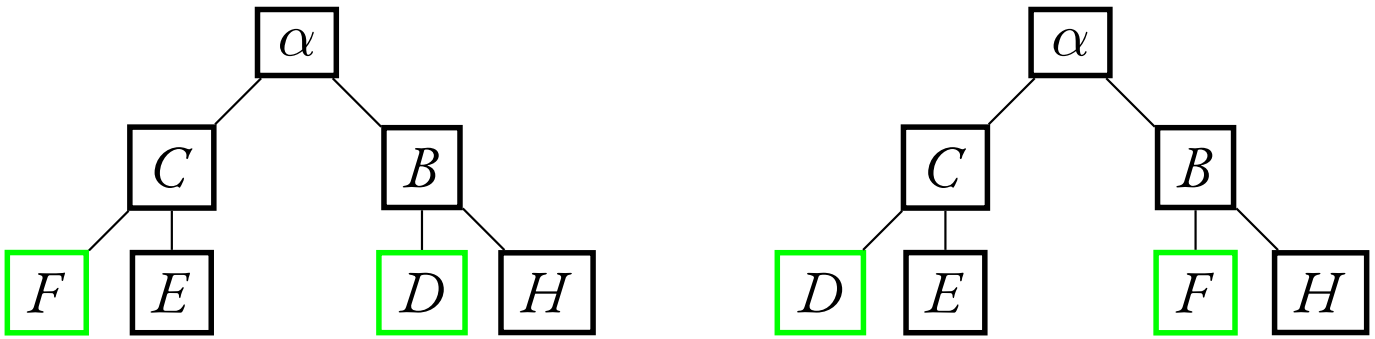
The **graph edit distance** considers the distance between two trees as **the minimum number of operations** (insertion, deletion, substitution) required to **transform a tree into another** (**tree_to_tree_editing**; **ged**).

If $c(e)$ is the cost of the operation e, the Graph Edit Distance is:

$$d(S_T, S_B) = \min \{c(e) \mid e \text{ sequence of operations to transform } S_1 \text{ into } S_2\}$$

where $c(e) = \sum_i c(e_i)$ is the cost of this sequence.

Graph edit distance



Operations: 1 substitution (F → D)

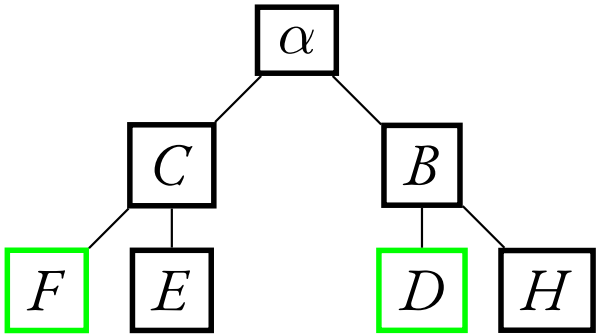
$$GE(S_1, S_2) = 1$$

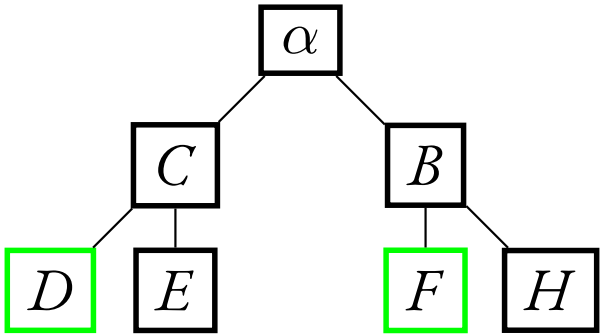
Adjacency based similarity

An **adjacency matrix** is a square matrix used to represent a finite graph.

- Each entry A_{ij} in the matrix indicates whether there is an edge between nodes i and j.
- Typically, $A_{ij} = 1$ if an edge exists and $A_{ij} = 0$ if there is no edge.
- The ℓ^1 or ℓ^2 matrix norm provides a distance between adjacency matrices.

Adjacency based similarity





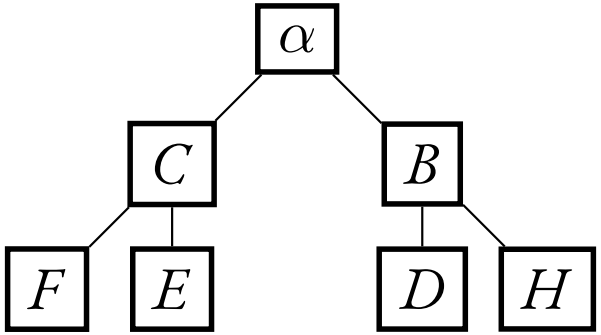
	α	C	B	F	E	D	H
α	0	1	1	0	0	0	0
C	1	0	0	1	1	0	0
B	1	0	0	0	0	1	1
F	0	1	0	0	0	0	0
E	0	1	0	0	0	0	0
D	0	0	1	0	0	0	0
H	0	0	1	0	0	0	0

	α	C	B	D	E	F	H
α	0	1	1	0	0	0	0
C	1	0	0	1	1	0	0
B	1	0	0	0	0	1	1
D	0	1	0	0	0	0	0
E	0	1	0	0	0	0	0
F	0	0	1	0	0	0	0
H	0	0	1	0	0	0	0

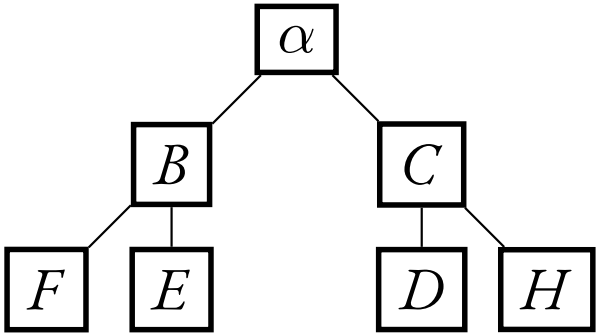
$$d(S_1, S_2) = 0.40$$

Ranking of each metric

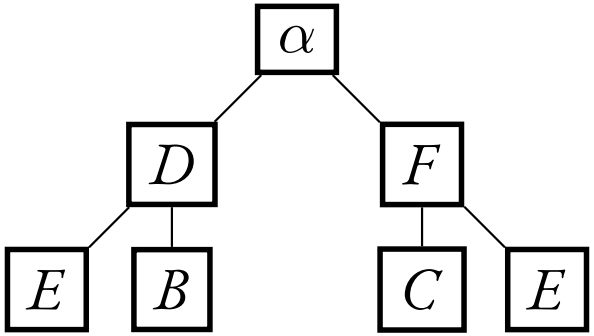
Reference stemma:



S_1



S_2



Ranking of each metric

Ranking of stemma closeness depending on metrics:

	RF	Adj	GED	Roos
S_1	**2**	**1**	**1**	**1**
S_2	**1**	**1**	**2**	**1**

- RF is very affected by **the move of primary witnesses**;
- **Graph edit** is very suitable for reconstructionist approaches, and a **possible extension is the augmentation of the penalty depending on depth of the node**;
- Roos and Adjacency, considering structure, are **equilibrated regarding the structure**.

Results on the Notre Besoin and Parzival tradition

Outline

Experimental setup

Artificial traditions:

- Parzival;
- Notre Besoin.

Metrics

- Roo's similarity;
- Robinson's Foulds;
- Graph Edit Distance;
- Adjacency matrix distance;

Tested algorithms (PAUP):

- Neighbor joining (with and without bootstrapping);
- Least squares (with and without bootstrapping);
- Parsimony (with and without bootstrapping).

Implementations

Introducing the **stemmadist** package:

- Only Python package centralizing the implementation of tree metrics (combining ETE3 and networkX);
- Perform distance computation on command line;

```
compute --distance rf
--tree1 "((F,(X, Y)E)C,(D,(W, Z)H)B)A;"
--tree2 "(((X, Y)C,E)F,((W, Z)B,H)D)A;"
```

GitHub Repository: github.com/metz-theolab/stemmadist

Results

- ALl distances strongly emphasize Neighbor Joining instead of.

Results

- ALl distances strongly emphasize Neighbor Joining instead of.
- Depending on the **selected metric**, results are altered **radically** with no single **outperforming algorithm**;

Results

- ALl distances strongly emphasize Neighbor Joining instead of.
- Depending on the **selected metric**, results are altered **radically** with no single **outperforming algorithm**;
- The task wanted for the stemma needs to **be taken into account** whenever interpreting the results:
  - **Existing benchmarking** favorite structure of the tree, and the **RHM algorithm**;
  - Neighbor Joining, in spite of its simplicity, **performs very well**;
  - New benchmarking studies should take into account the **specificity of different approaches**, using *Graph Edit Distance* (GED).

Choosing the metric of benchmarking is almost as delicate as finding the ground truth!

Conclusion

Outline

Using the mentioned softwares

Stemmabench: **Link to GitHub project**:
`https://github.com/metz-theolab/stemmabench`.
Link to project's website:
`https://metz-theolab.github.io/stemmabench/`.
Link to PyPi module:
`https://pypi.org/project/stemmabench/`.

Graph distances:
`https://github.com/metz-theolab/stemmadist`

Further works

- Implement **contamination** in StemmaBench by studying real examples;
- Generate **new handwritten traditions** (exciting experiment with Helsinki University!) to study:
  - Study **variant generation**;
  - Validate **current modelization**;
  - Generate new data for **validation** of models.

Questions

Questions ?

Bibliography I

- Camps, Jean-Baptiste (Jan. 2015). "Copie, authenticité, originalité dans la philologie et son histoire". fr. In: *Questes* 29, pp. 35–67. ISSN: 2102-7188, 2109-9472. DOI: 10.4000/questes.3535.
- Drummond, Alexei J. and Remco R. Bouckaert (2015). *Bayesian evolutionary analysis with BEAST*. Cambridge: Cambridge University Press.
- Felsenstein, Joseph (1981). "Evolutionary trees from DNA sequences: A maximum likelihood approach.". In: *Journal of Molecular Evolution* 17, pp. 168–376.
- Maas, Paul (Jan. 1958). *Textual criticism*. London: Oxford University Press. ISBN: 978-0-19-814318-5.
- Roos, Teemu and T. Heikkila (2009). "Evaluating methods for computer-assisted stemmatology using artificial benchmark datasets". In: *Literary and Linguistic Computing* 24, pp. 417–433.

Bibliography II

- Saitou N, Nei M. (July 1987). "The neighbor-joining method: a new method for reconstructing phylogenetic trees.". In: *Mol Biol Evol.* 4, pp. 406–425.
- Sokal, Robert R. and Charles Duncan Michener (1958). "A statistical method for evaluating systematic relationships". In: *University of Kansas science bulletin* 38, pp. 1409–1438. URL: https://api.semanticscholar.org/CorpusID:61950873.