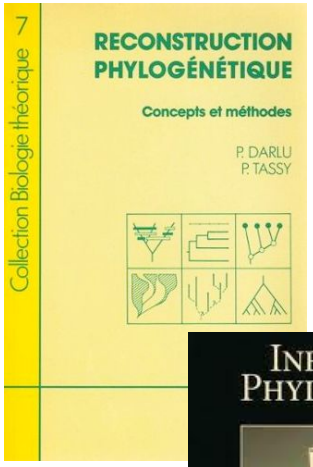
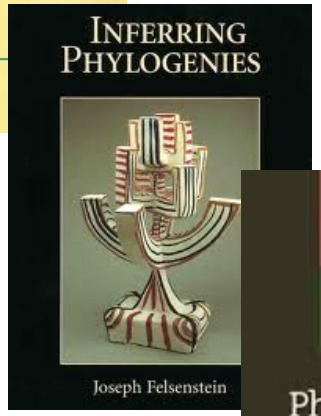
Master Parisien de Recherche en Informatique - c-1-32
Algorithmics for Bioinformatics - 2022

Algorithmics & Phylogenetics

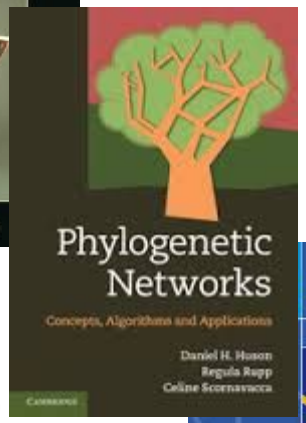
Lecture 1



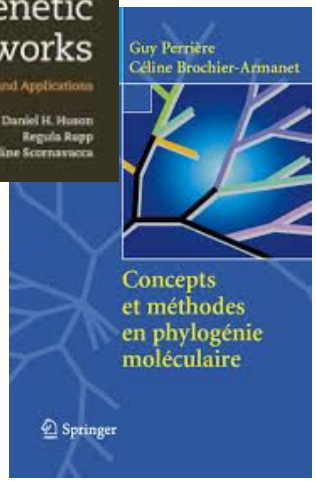
Darlu & Tassy, 1993
http://sfs.snv.jussieu.fr/?q=node/11



Felsenstein, 2002

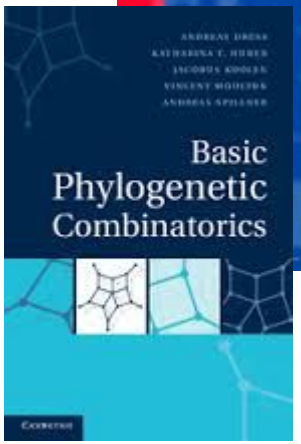


Huson, Rupp & Scornavacca, 2010



Perrière & Brochier-Armanet, 2010

Barthelémy & Guénoche, 1988



Semple & Steel, 2003

Dress, Huber, Koolen, Moulton & Spillner, 2011

Université Gustave Eiffel
GASPARD MONGE LABORATOIRE D'INFORMATIQUE
Philippe Gambette

References

Finding **bibliographic references**:

- Google Scholar: large coverage, including noise (preprints, fake papers...)
- Bibliographic resources with academic access:

  https://ens-paris-saclay.fr/bibliotheque/ressources-documentaires

  → Science Direct for Elsevier journals, Springer, JSTOR, etc.
- University library

Conferences with computer science papers applied to phylogenetics:

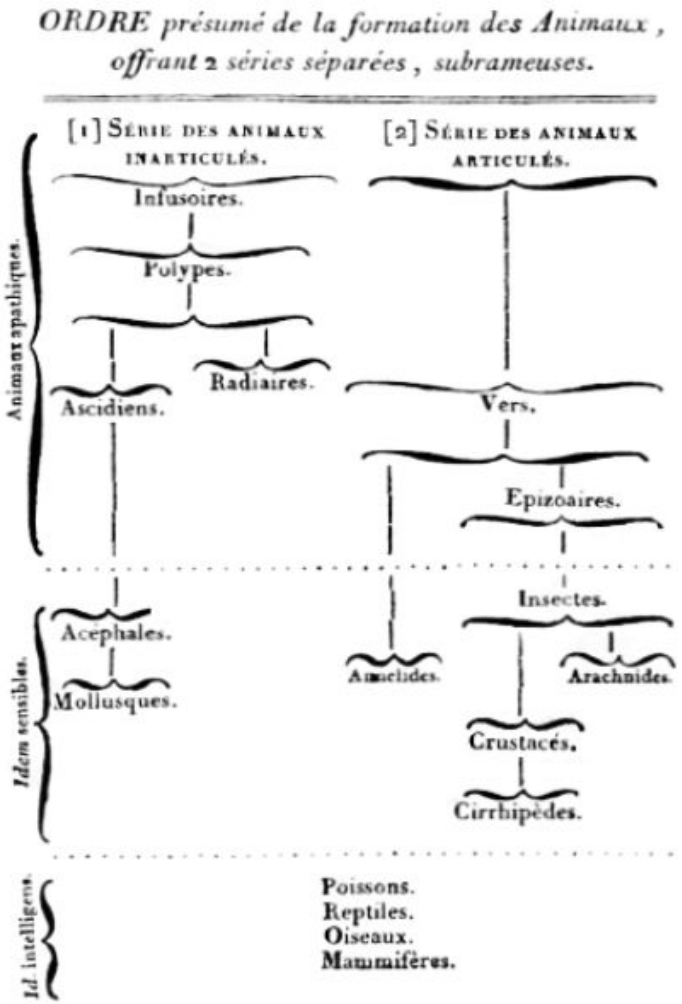
- International: ISMB, RECOMB, WABI, ECCB, ISBRA, etc.
  + algorithmics conferences: SODA, CPM, ISAAC, COCOON, etc.
- In France: JOBIM, Alphy

Scholarly organizations:

Phylogenetic trees

Phylogenetic tree of a set of species:

- **organizing** them according to common characters **classification**
- describing their evolution



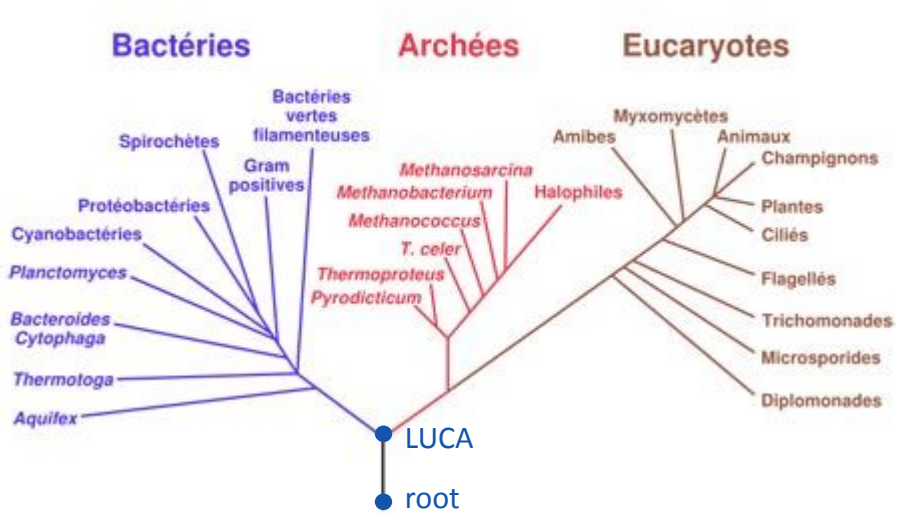
Lamarck : Histoire naturelle des animaux sans vertèbres (1815)

Phylogenetic trees

Phylogenetic tree of a set of species:

- organizing them according to common characters
- **describing** their evolution

modelization



Woese, Kandler, Wheelis : Towards a natural system of organisms: proposal for the domains Archaea, Bacteria, and Eucarya, Proceedings of the National Academy of Sciences, 87(12), 4576–4579 (1990)

Encoding a tree

"Newick" format:

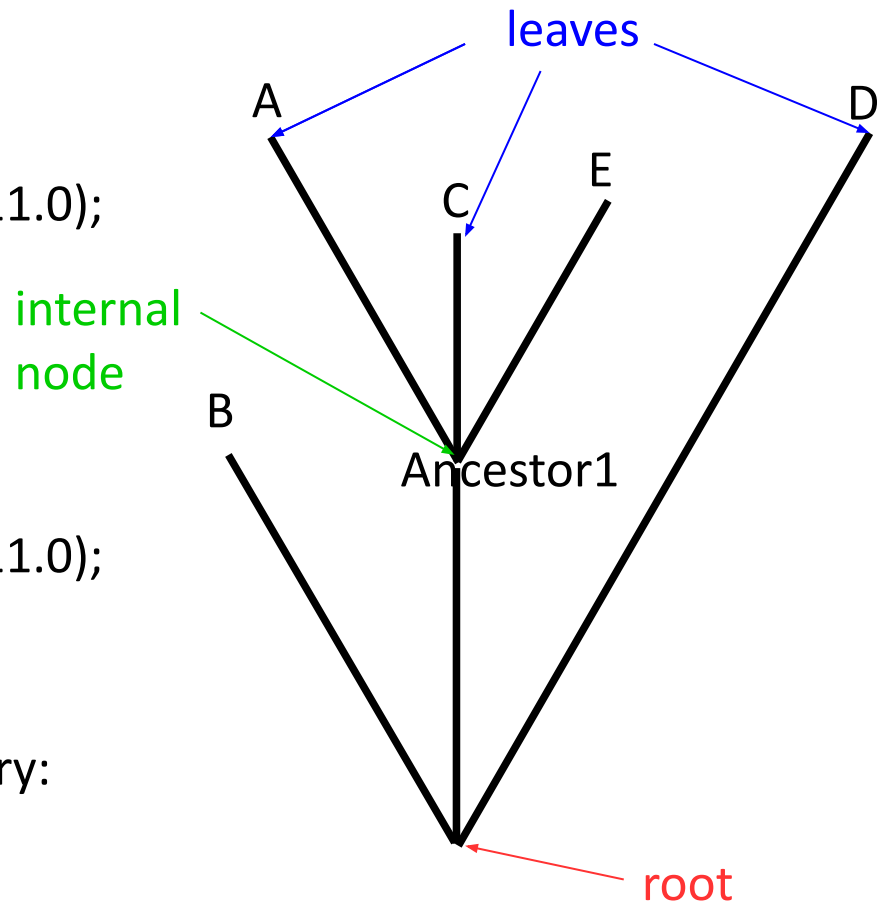
(B:6.0,(A:5.0,C:3.0,E:4.0)Ancestor1:5.0,D:11.0);

Not unique:

((A:5.0,C:3.0,E:4.0)Ancestor1:5.0,B:6.0,D:11.0);

Possible to have nodes which are not binary:

- uncertainty
- or speciation known to be at the same time



http://evolution.genetics.washington.edu/phylip/newicktree.html

Properties of phylogenetic trees

Characterizing a tree with:

- its “clusters”: one cluster of *T* = the set of leaves below one vertex of *T*
- its “triplets”: one triplet of *T* = a tree on 3 leaves contained in *T*
- distances between the leaves

Properties of rooted/unrooted phylogenetic trees

Characterizing a **rooted tree** with:

- its "clusters": one cluster of *T* = the set of leaves below one vertex of *T*
- its "triplets": one triplet of *T* = a tree on 3 leaves contained in *T*
- ~~distances between the leaves~~

Characterizing an **unrooted tree** with:

- its "splits": one split of *T* = bipartition of the leaves induced by one edge of *T*
- its "quartets": one quartet of *T* = an unrooted tree on 4 leaves contained in *T*
- distances between the leaves

Properties of rooted and unrooted trees

Clusters: “laminar family”, i.e. it contains no overlapping sets

$\longrightarrow$ reconstruction from clusters: Hasse Diagram of cluster inclusion

Triplets (binary trees): do not contain {ab|c, b|cd, a|bd} or {ab|c, b|cd, ad|b}

Guillemot & Mnich, Kernel and fast algorithm for Dense Triplet Inconsistency, 2013

Splits: “compatible split system”, i.e. for any pair of splits A1|B1, A2|B2, at least one of the sets $A1 \cap A2$, $A1 \cap B2$, $B1 \cap A2$, $B1 \cap B2$ is empty

Quartets (binary trees): for any leaf *e*, ab|cd $\in Q \Rightarrow$ ab|ce $\in Q$ or ae|cd $\in Q$

Bandelt & Dress, Reconstructing the shape of a tree from observed dissimilarity data, 1986

Properties of rooted and unrooted trees

Tree distances:

Characterized by Buneman's **four-point condition**:

$$\boxed{\text{for all } a, b, c, d,\ d(a,b)+d(c,d) \leq \max\{d(a,c)+d(b,d),\ d(a,d)+d(b,c)\}}$$

$\Leftrightarrow$ for any four points, we can relabel them a, b, c, d such that $d(a,b)+d(c,d) \leq d(a,c)+d(b,d) = d(a,d)+d(b,c)$.

Given a tree distance, only one possible tree.

Buneman, A Note on the Metric Properties of Trees, 1974

Tree distances when the tree contains a center at equal distance from all leaves:

Characterized by the **ultrametric inequality**:

$$\boxed{\text{For all } a, b, c,\ d(a,b) \leq \max\{d(a,c),\ d(b,c)\}}$$

molecular clock hypothesis!

Reconstructing a tree from an ultrametric

UPGMA algorithm (Unweighted Pair Group Method with Arithmetic Mean):

- Initialize all clusters with leaf singletons
- While there are more than 2 clusters:
  - pick the nearest two clusters
  - combine them and update the distance matrix with average values (average weighted by the size of each of the two clusters)

Sokal & Michener, A statistical method for evaluating systematic relationships, 1958

→ Correctly reconstructs ultrametric distances, but not all tree distances

→ Neighbor-Joining...

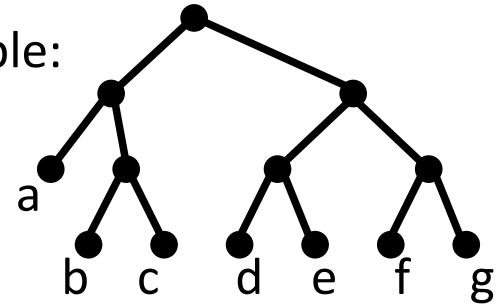
Reconstructing a tree from its triplets

BUILD algorithm:

- Build the following graph: leaves as vertices; for each triplet a|bc, add edge bc.
- While there is more than one connected component:
  - each connected component corresponds to one subtree
  - recursively apply the algorithm on the leaf set of each connected component

Aho, Sagiv, Szymanski & Ullman, Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions, 1981.

Example:



{a|bc, a|de, a|df, b|dg, b|ef, c|df, d|ac, d|fg, e|ab, f|de, g|ab, g|ac}

→ When missing triplets, efficient implementation in $O(|T|+n^2 \log n)$

Henzinger, King & Warnow, Constructing a tree from homeomorphic subtrees, with applications to computational evolutionary biology, 1999.

Reconstructing a tree from its triplets

BUILD algorithm:

- Build the following graph: leaves as vertices; for each triplet a|bc, add edge bc.
- While there is more than one connected component:
  - each connected component corresponds to one subtree
  - recursively apply the algorithm on the leaf set of each connected component

Aho, Sagiv, Szymanski & Ullman, Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions, 1981.

Example:

{a|bc, a|de, a|df, b|dg, b|ef, c|df, d|ac, d|fg, e|ab, f|de, g|ab, g|ac}

→ When missing triplets, efficient implementation in $O(|T|+n^2 \log n)$

Henzinger, King & Warnow, Constructing a tree from homeomorphic subtrees, with applications to computational evolutionary biology, 1999.

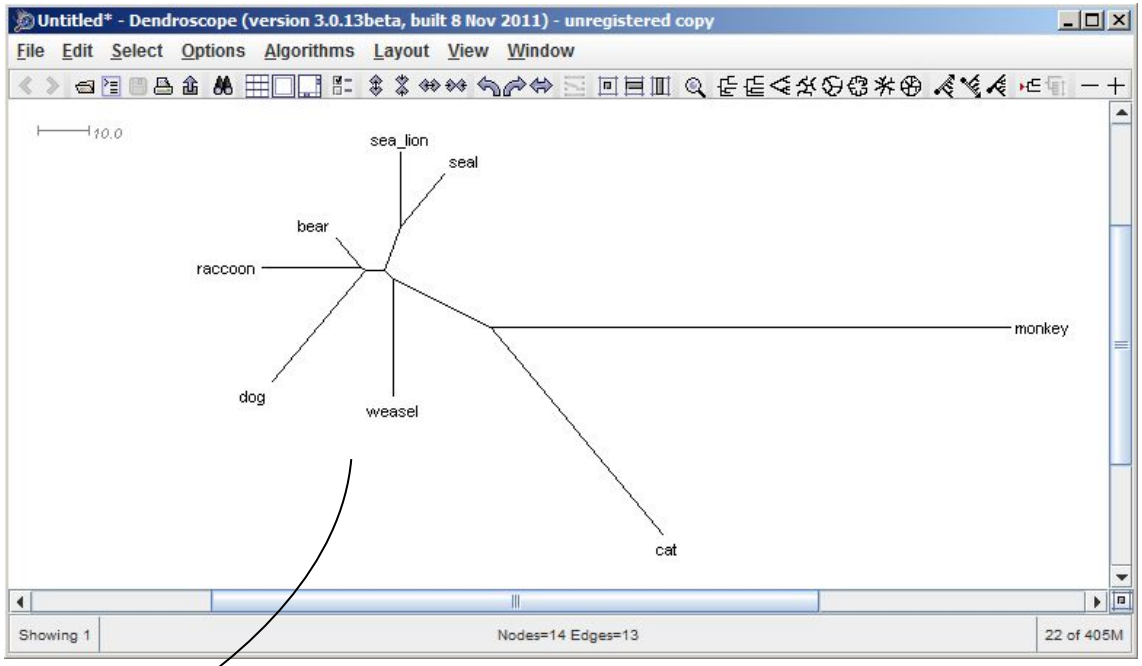
Visualizing phylogenetic trees

Visualize branch lengths or not

Several kinds of visualizations:

- rectangular phylogram
- rectangular cladogram
- slanted cladogram
- circular phyogram
- circular cladogram
- inner circular cladogram
- radial phylogram
- radial cladogram

((raccoon:19.19959,bear:6.80041):0.84600,((sea_lion:11.99700, seal:12.00300):7.52973,((monkey:100.85930,cat:47.14069):20.59201, weasel:18.87953):2.09460):3.87382,dog:25.46154);



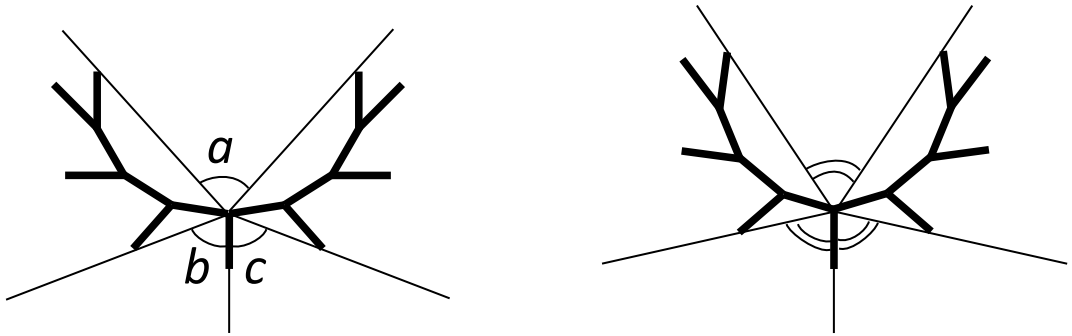
http://www.dendroscope.org

Visualizing phylogenetic trees: radial phylogram

"Equal angle" algorithm to draw a radial phylogram on *n* leaves:

- Compute the angles "bottom-up" starting with angle $2i\pi/n$ for leaf *i*
- Locate the nodes "top-down" using:
  - the angles
  - the edge lengths
- Add the labels (avoiding overlap)

"Equal daylight" algorithm to optimize used space:



Felsenstein, Inferring phylogenies, 2002, 579-583

Comparing trees

- **Maximum Agreement Subtree** (MAST): Finden & Gordon, Obtaining common pruned trees, 1985

  - Given T1 and T2 on the set X of leaves, an *agreement subtree* T of T1 and T2, on a subset X' of leaves, is such that T1 and T2 restricted to X' are equal to T.

  - a *maximum agreement subtree* is an agreement subtree of maximum size

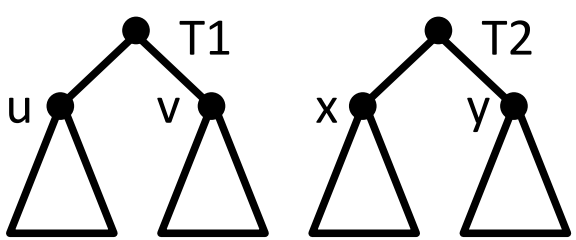
→ Polynomial-time algorithm for 2 trees (rooted or not), with fixed degree (or not)
Steel & Warnow, Kaikoura tree theorems: Computing the maximum agreement subtree, 1993

→ $O(n \log n)$ algorithm for 2 rooted binary trees
Cole & Hariharan, An $O(n \log n)$ algorithm for the maximum agreement subtree problem for binary trees, 1996

→ NP-hard for 3 (or more) trees of unbounded degree
Amir & Keselman, Maximum agreement subtree in a set of evolutionary trees, 1997

→ Polynomial-time algorithm if one of the trees has bounded maximum degree
Farach, Przytycka & Thorup, On the agreement of many trees, 1995

Dynamic programming for 2 rooted binary trees:



$$n_{\text{MAST}}(T1,T2)=\max\left\{\right.$$

Comparing trees

- **Maximum Agreement Subtree** (MAST): Finden & Gordon, Obtaining common pruned trees, 1985

  - Given T1 and T2 on the set X of leaves, an *agreement subtree* T of T1 and T2, on a subset X' of leaves, is such that T1 and T2 restricted to X' are equal to T.

  - a *maximum agreement subtree* is an agreement subtree of maximum size

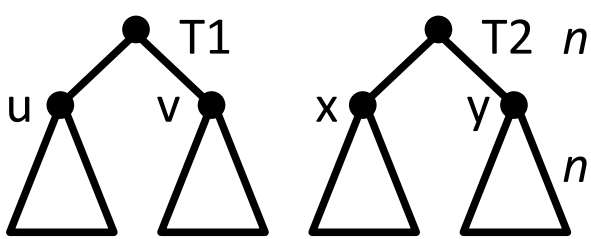
→ Polynomial-time algorithm for 2 trees (rooted or not), with fixed degree (or not)
Steel & Warnow, Kaikoura tree theorems: Computing the maximum agreement subtree, 1993

→ $O(n \log n)$ algorithm for 2 rooted binary trees
Cole & Hariharan, An $O(n \log n)$ algorithm for the maximum agreement subtree problem for binary trees, 1996

→ NP-hard for 3 (or more) trees of unbounded degree
Amir & Keselman, Maximum agreement subtree in a set of evolutionary trees, 1997

→ Polynomial-time algorithm if one of the trees has bounded maximum degree
Farach, Przytycka & Thorup, On the agreement of many trees, 1995

Dynamic programming for 2 rooted binary trees:



$$n_{MAST}(T1,T2)=\max\begin{cases} n_{MAST}(T1[u],T2[x]) + n_{MAST}(T1[v],T2[y]), \\ n_{MAST}(T1[v],T2[x]) + n_{MAST}(T1[u],T2[y]), \\ n_{MAST}(T1,T2[x]),\ n_{MAST}(T1,T2[y]), \end{cases}$$

Comparing trees

- **Maximum Compatible Tree** (MCT):

  - if a tree is not binary, several binary trees *refine* it

  - a compatible tree on a subset X' of leaves is a binary tree which refines the trees T1 and T2 restricted to the leaves of X'.

→ Polynomial-time algorithm for 2 trees of bounded degree
→ NP-hard if one tree can have arbitrarily large degree

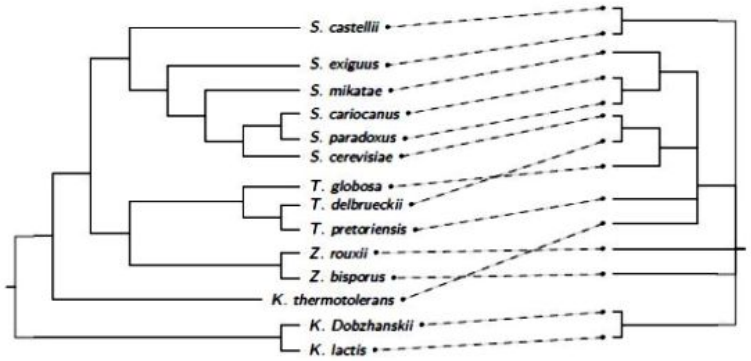
Hein, Jiang, Wang & Zhang, On the complexity of comparing evolutionary trees, 1996

→ $O(\min\{3^p kn, 2.27^p + kn^3\})$ for k trees, where p is the number of leaves to remove

Berry & Nicolas, Improved parametrized complexity of the maximum agreement subtree and maximum compatible tree problems, 2006

- **Tanglegrams**:

  - display both trees for visual comparison, linking their leaves with edges, minimizing edge crossings.
  → general problem NP-complete
  → planar embedding in linear time
  → if one tree is fixed, $O(n \log n)$



Venkatachalam, Apple, St John, Gusfield, Untangling tanglegrams: comparing trees by their drawings, 2010