A short (?) introduction to phylogenetic networks

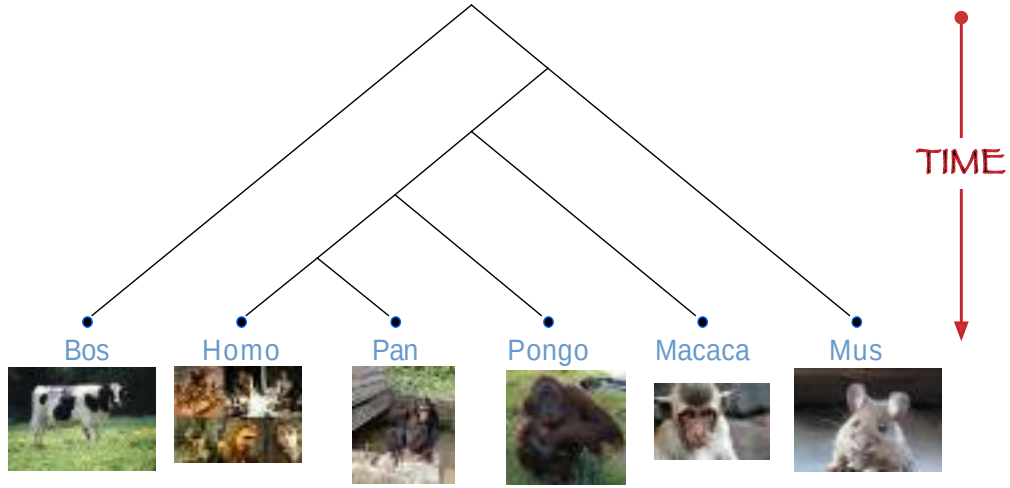
Céline Scornavacca

ISE-M, Equipe Phylogénie & Evolution Moléculaires
Montpellier, France

Rooted species trees ...

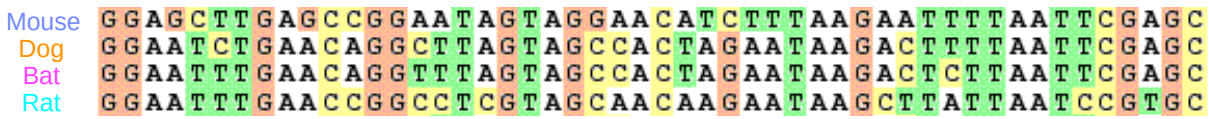
... are oriented connected and acyclic graphs, where terminal nodes are associated to a set of species:

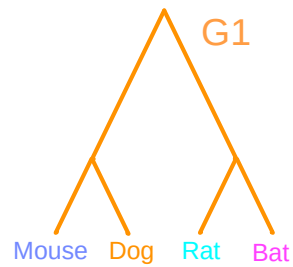
- the leaves or taxa represent extant organisms
- internal nodes represent hypothetical ancestors
- each internal node represents the lowest common ancestor of all taxa below it (clade)
- the only node without ancestor is called root



Gene trees

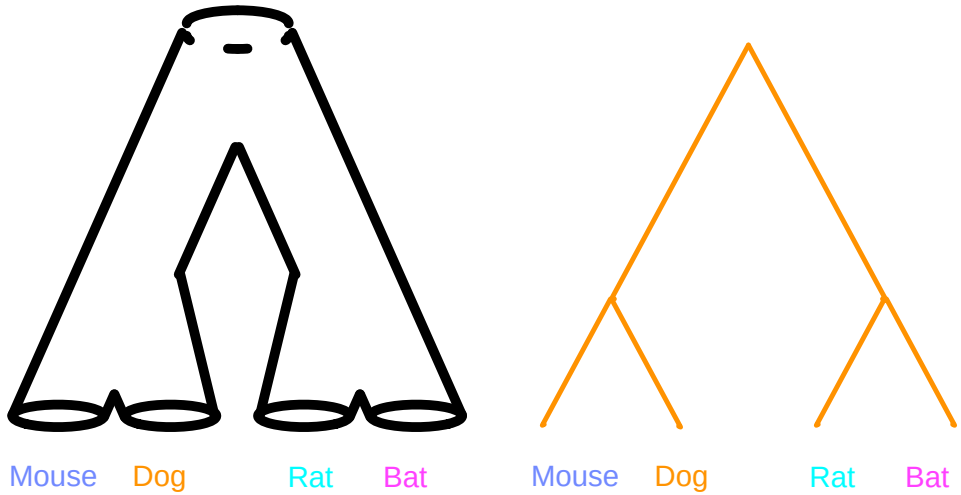
- Gene trees are built by analyzing a gene family, i.e., homologous molecular sequences appearing in the genome of different organisms.





Gene trees

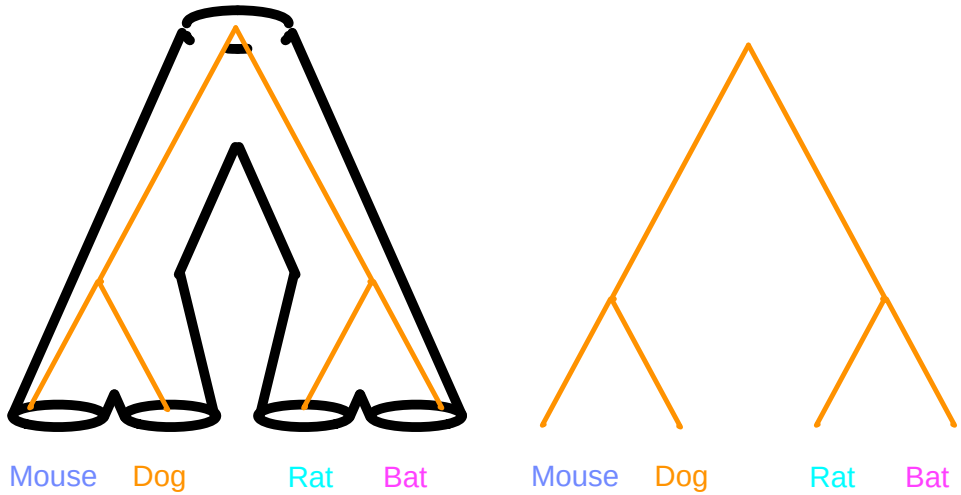
- Gene trees are built by analyzing a gene family.



- Used, among other things, to estimate species trees.

Gene trees

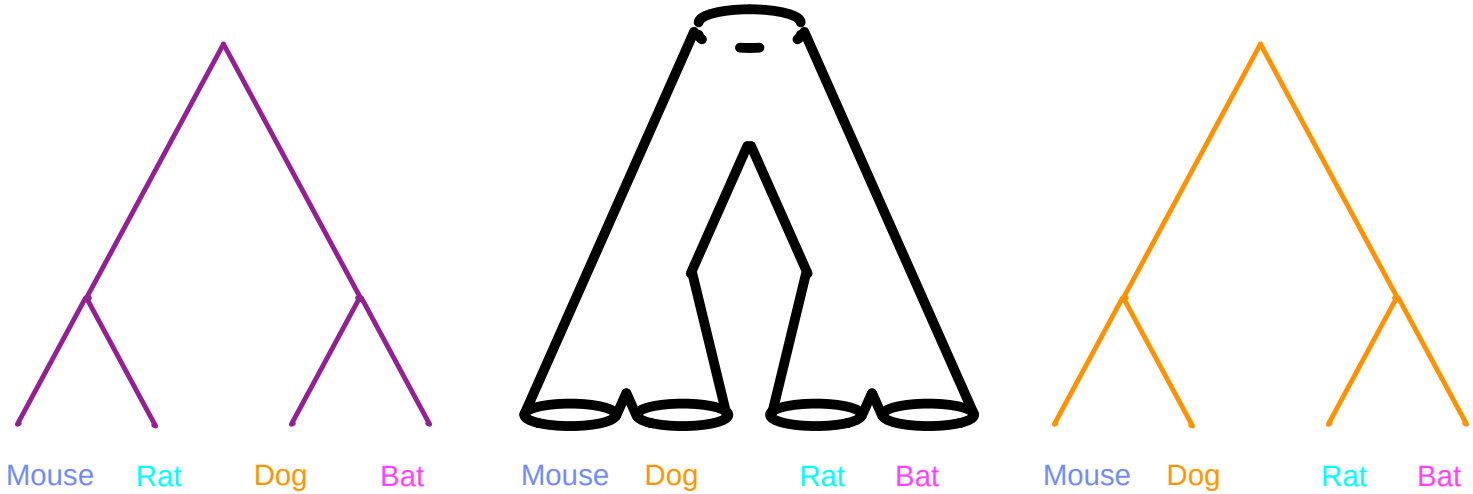
- Gene trees are built by analyzing a gene family.



- Used, among other things, to estimate species trees.

Gene trees

- Gene trees are built by analyzing a gene family.



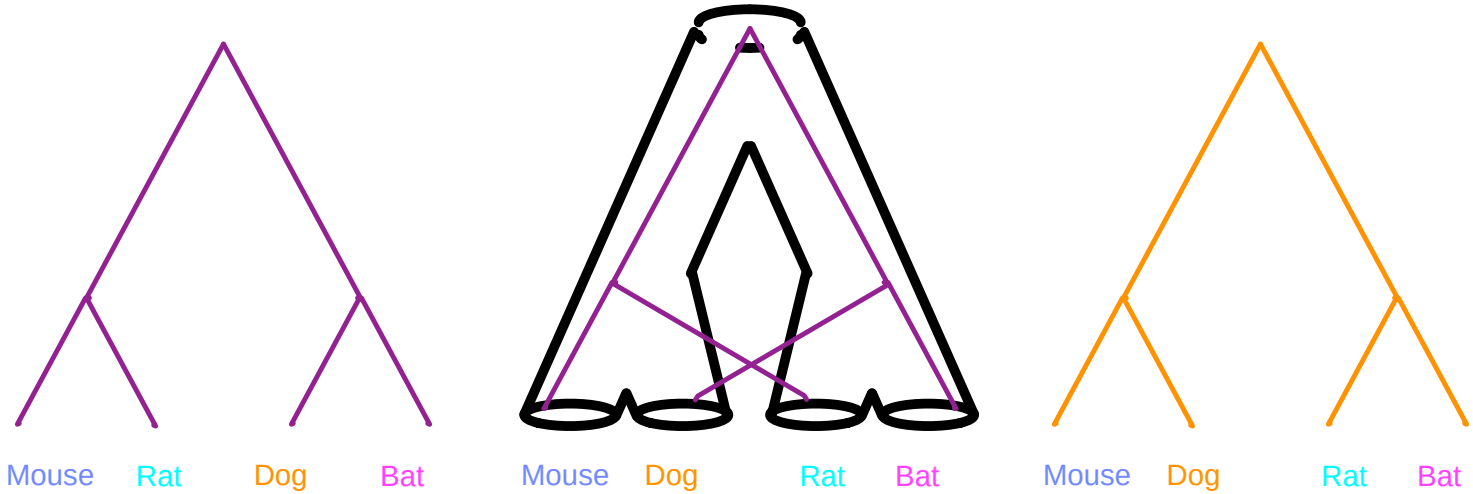
- Used, among other things, to estimate species trees.

Gene trees can significantly differ from the species tree for:

- methodological reasons
- biological reasons

Gene trees

- Gene trees are built by analyzing a gene family.



- Used, among other things, to estimate species trees.

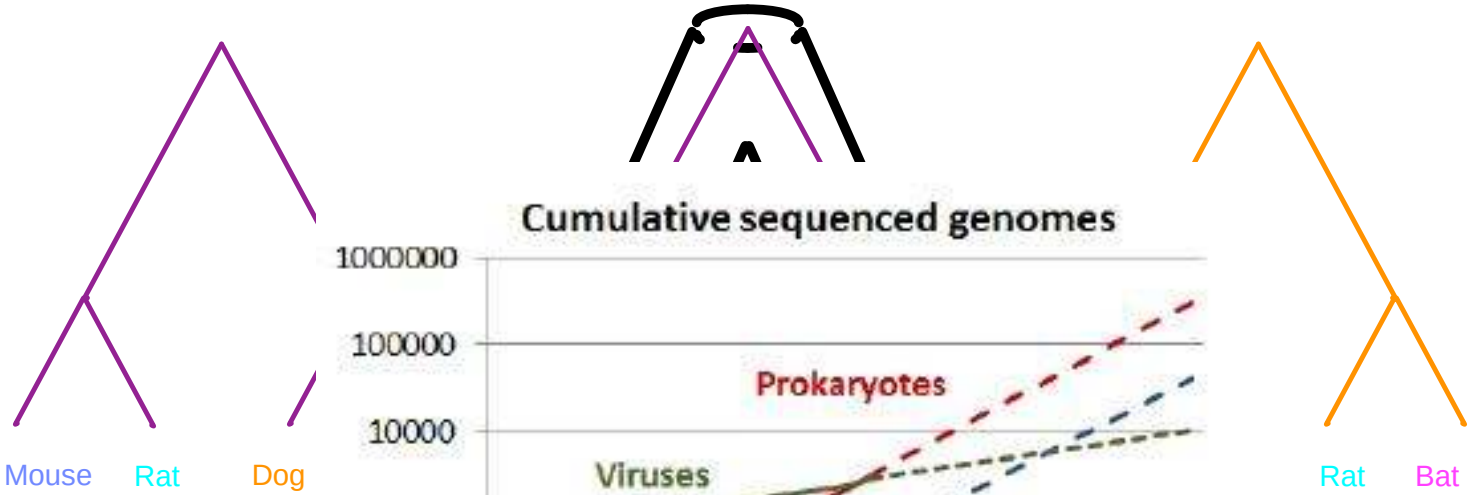
Gene trees can significantly differ from the species tree for:

- methodological reasons
- biological reasons

- We usually use several gene families...

Gene trees

- Gene trees are built by analyzing a gene family.



- Used, among o

Gene trees can sig **ee for:**

- methodological
- biological reasons

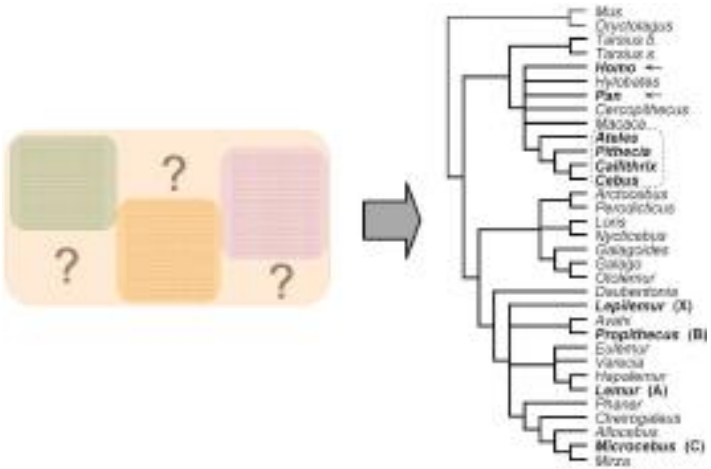
- We usually use several gene families...

http://sulab.org/2013/06/sequenced-genomes-per-year/

Reconstruction of phylogenies for multiple datasets

The two main *classic* approaches:

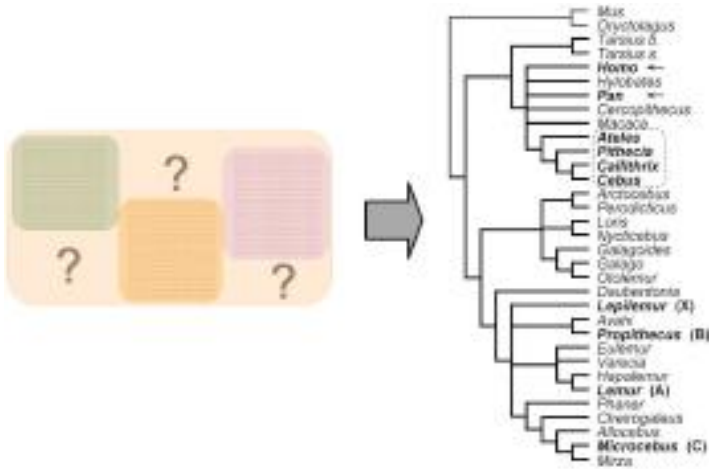
- Supermatrix approach: assembling primary data



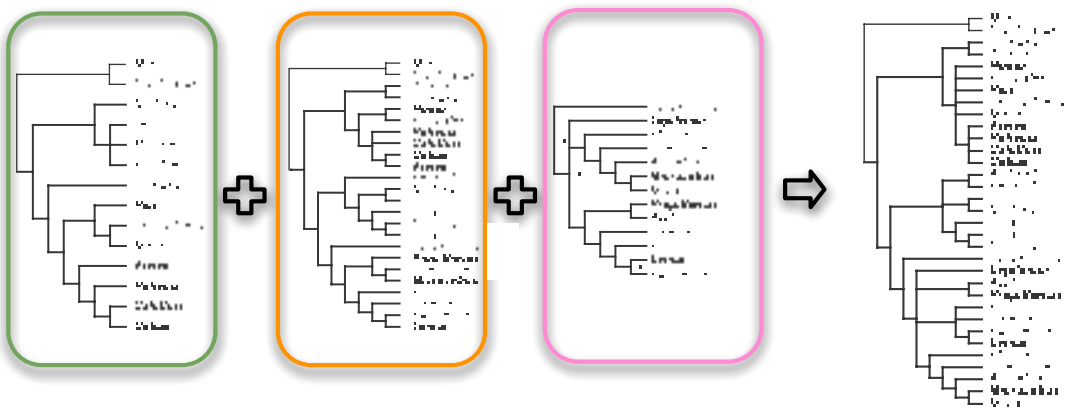
Reconstruction of phylogenies for multiple datasets

The two main *classic* approaches:

- Supermatrix approach: assembling primary data

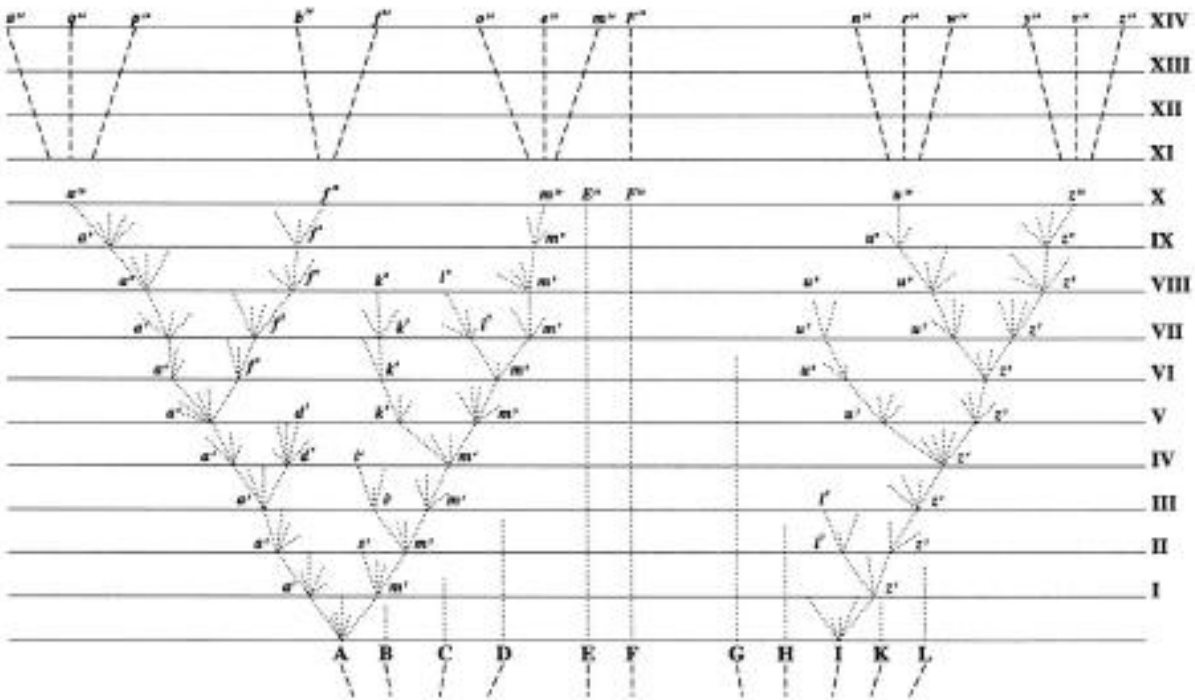


- Supertree approach: assembling trees



An implicit assumption

The implicit assumption of using trees is that, at a macroevolutionary scale, each (current or extinct) species or gene only descends from one ancestor. Darwin described evolution as "descent with modification", a phrase that does not necessarily imply a tree representation...

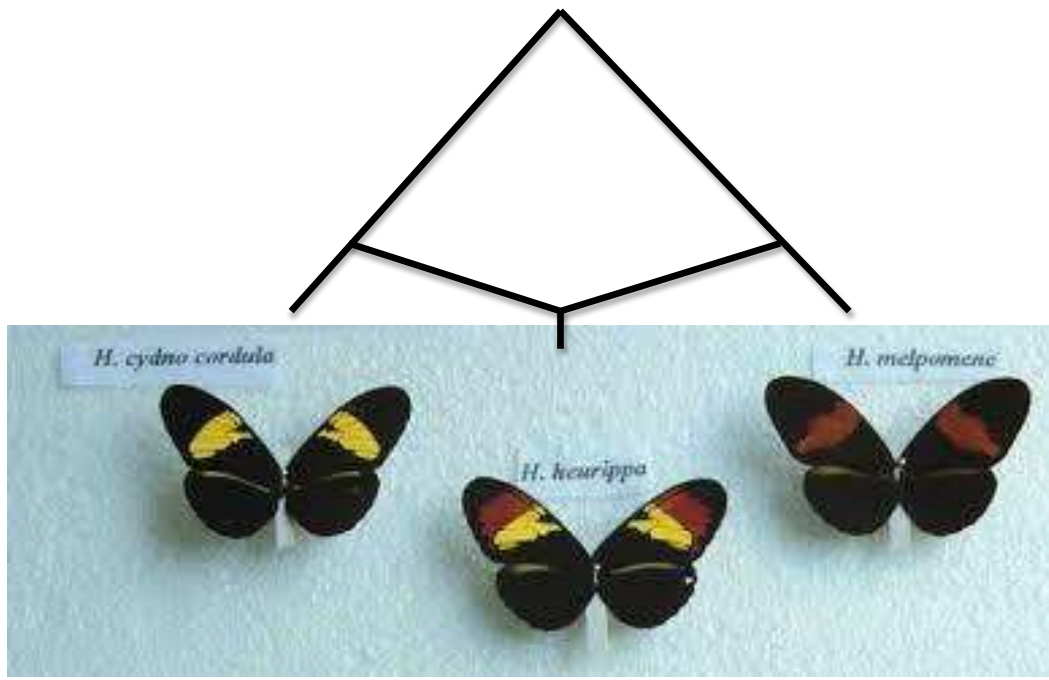


A *new* approach: building phylogenic networks

Why do we need them? Due to reticulate evolutionary phenomena (hybridization, recombination, horizontal gene transfer) the evolution of a set of species sometimes cannot be described using phylogenetic trees.

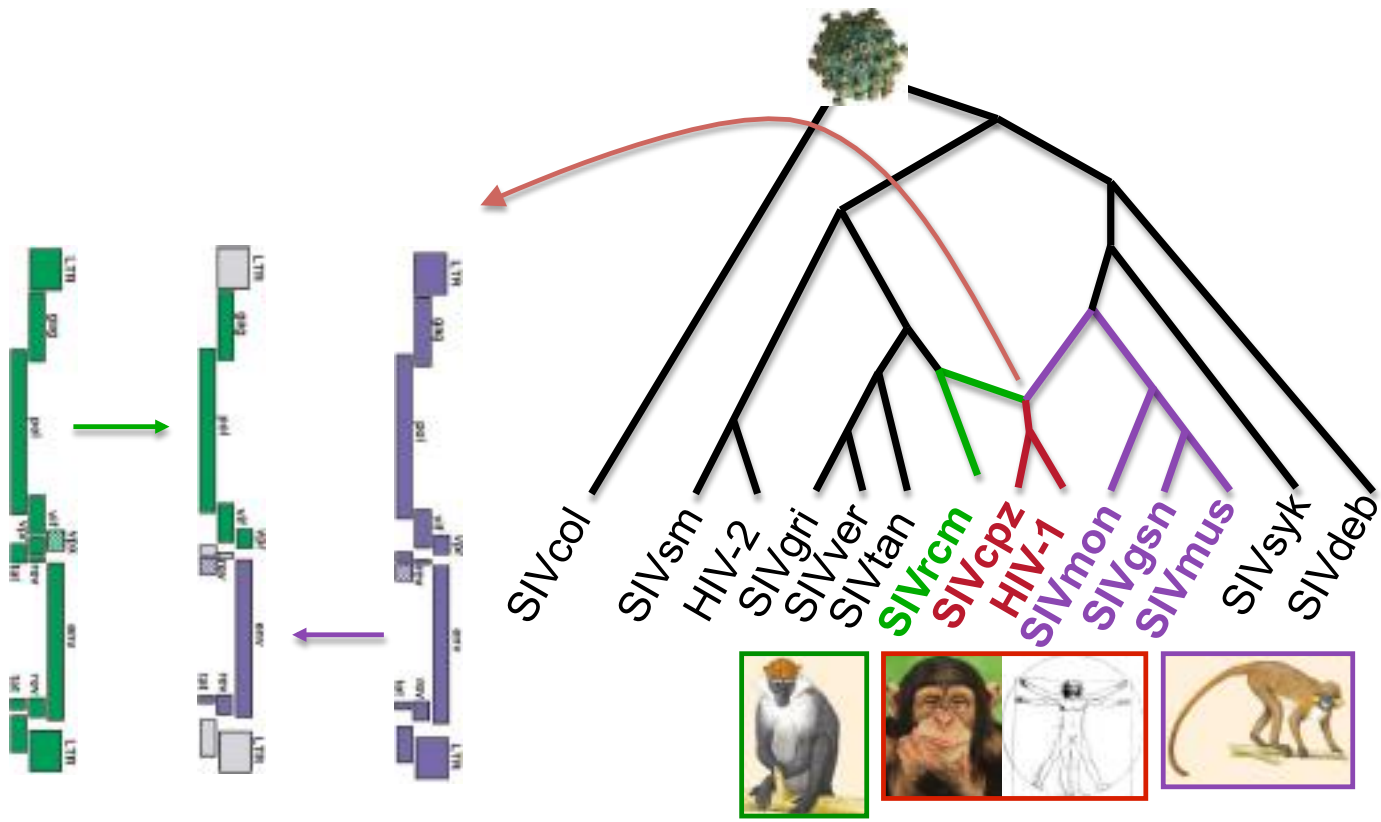
A *new* approach: building phylogenic networks

Why do we need them? Due to reticulate evolutionary phenomena (hybridization, recombination, horizontal gene transfer) the evolution of a set of species sometimes cannot be described using phylogenetic trees.



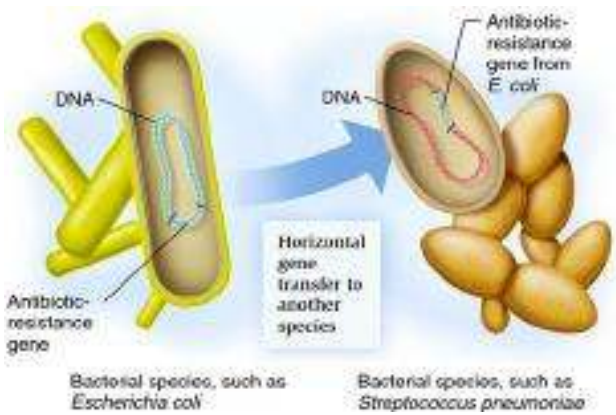
A *new* approach: building phylogenic networks

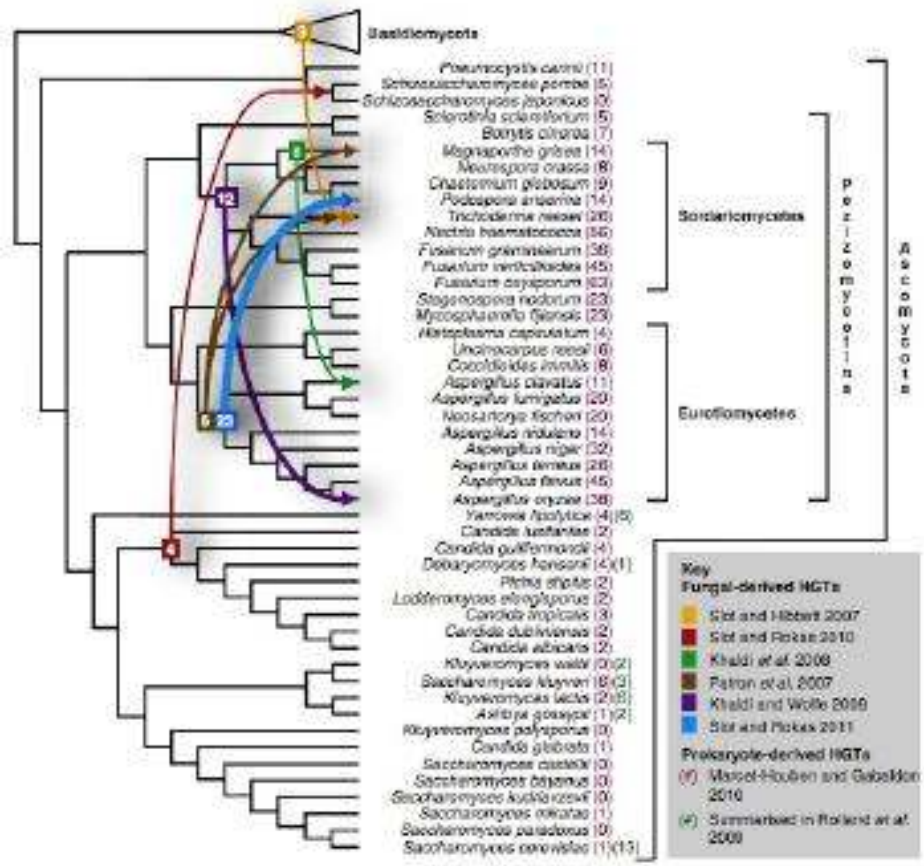
Why do we need them? Due to reticulate evolutionary phenomena (hybridization, recombination, horizontal gene transfer) the evolution of a set of species sometimes cannot be described using phylogenetic trees.



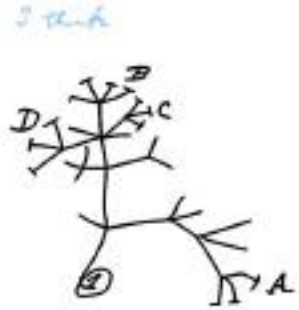
A *new* approach: building phylogenic networks

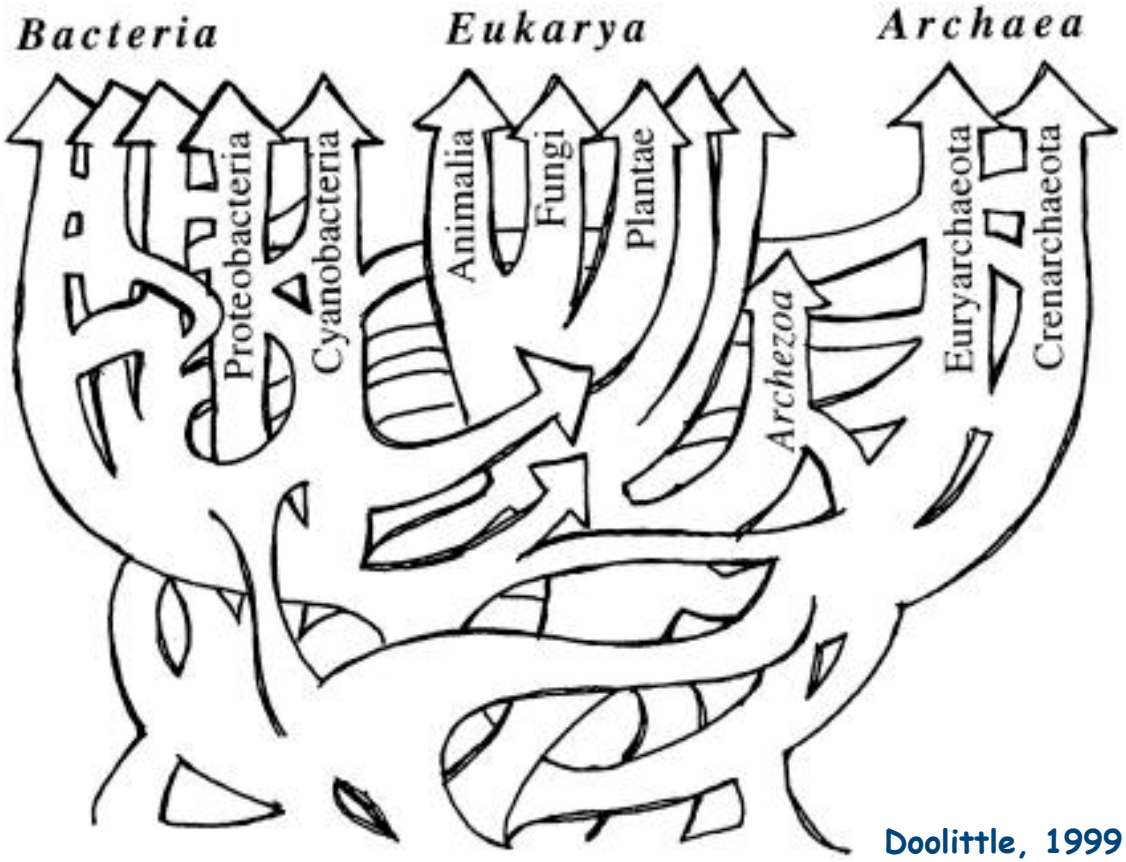
Why do we need them? Due to reticulate evolutionary phenomena (hybridization, recombination, horizontal gene transfer) the evolution of a set of species sometimes cannot be described using phylogenetic trees.





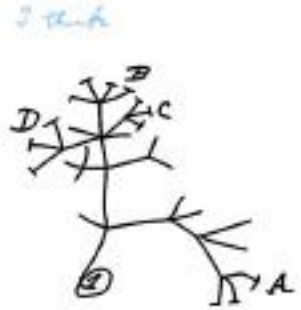
The network of life



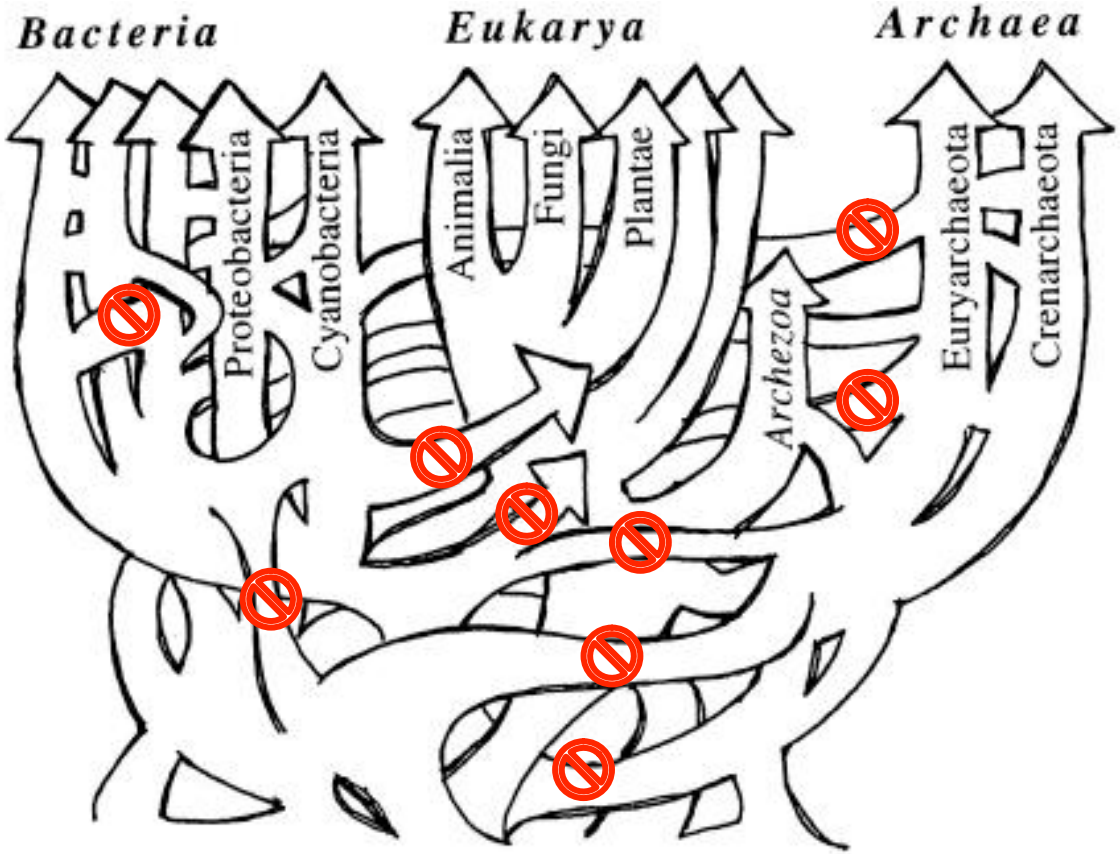


Doolittle, 1999

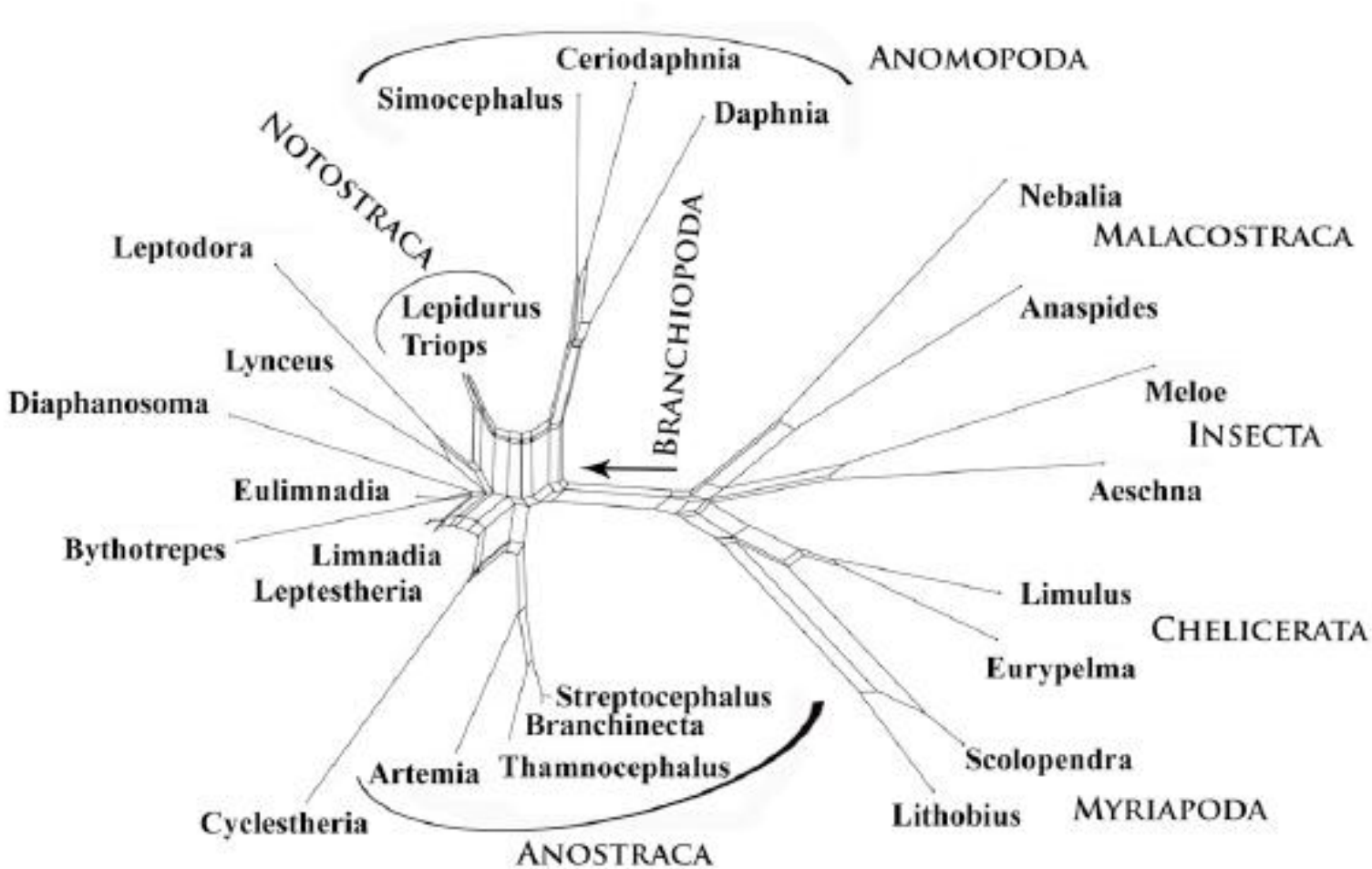
Three different paradigms



We (want to) see only the tree

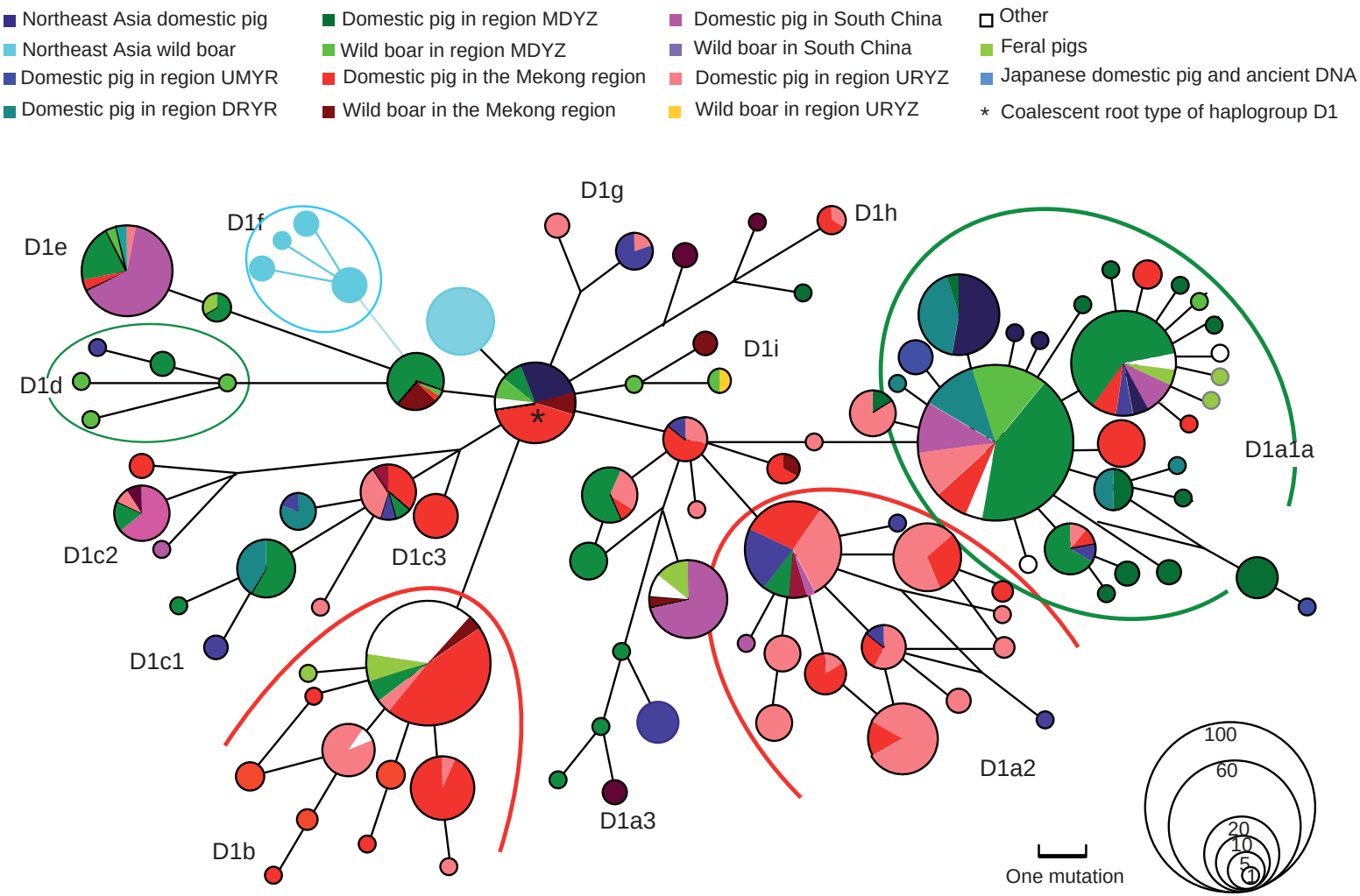


An example - a split network



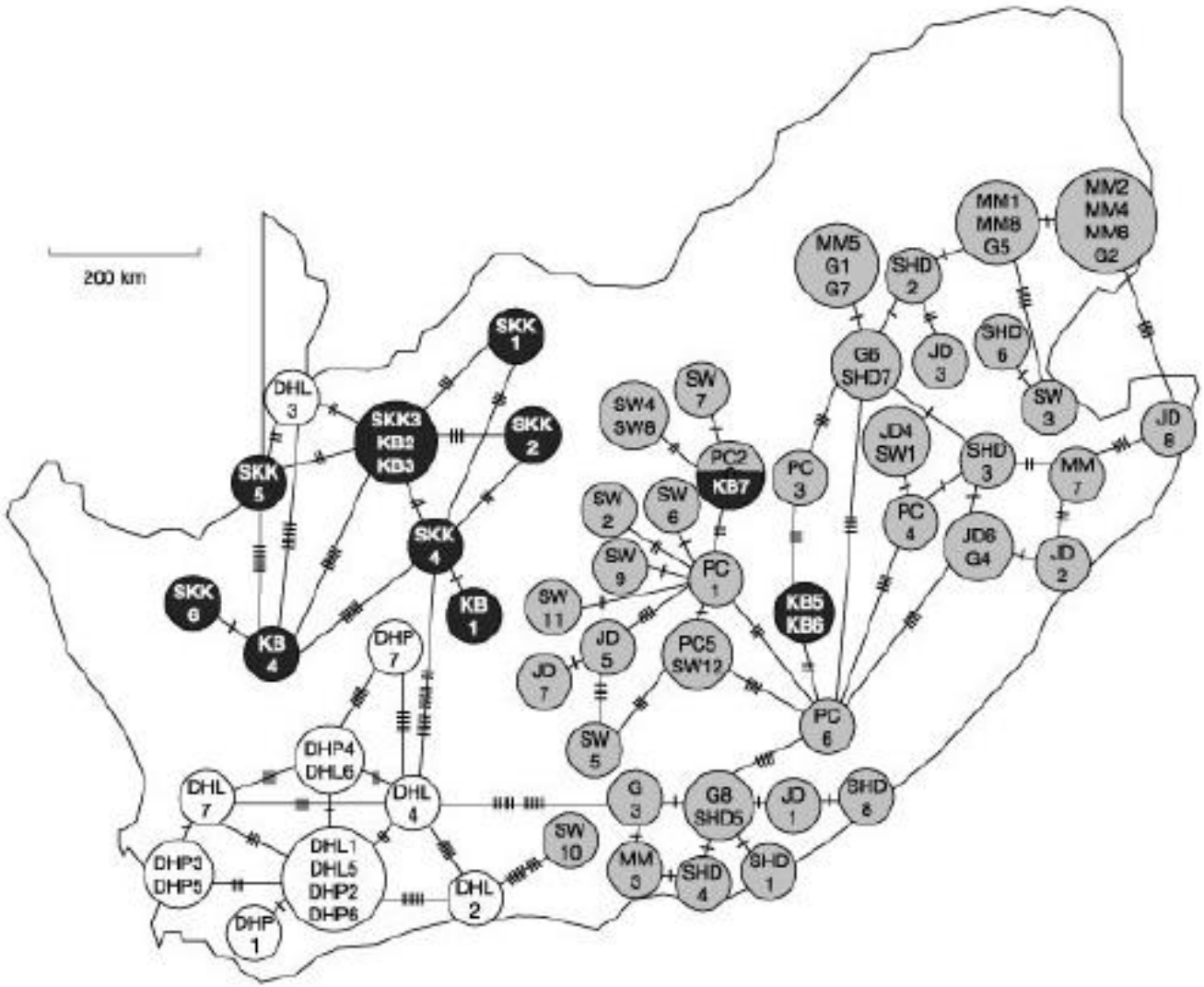
J. Wagele and C. Mayer. Visualizing differences in phylogenetic information content of alignments and distinction of three classes of long-branch effects. BMC Evolutionary Biology, 7(1):147, 2007

An example - a reduced median network



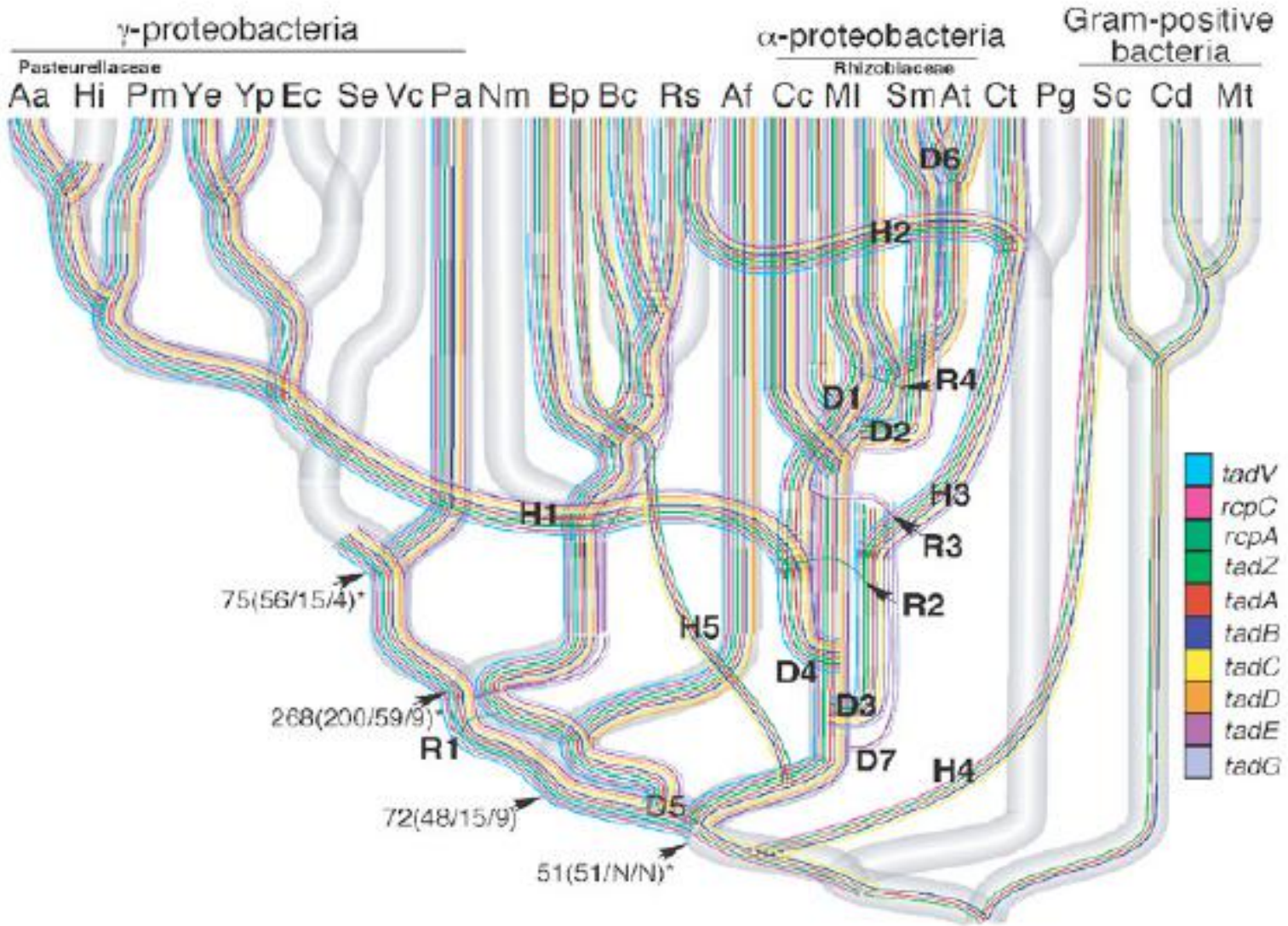
G.-S. Wu, Y.-G. Yao, K.-X. Qu, Z.-L. Ding, H. Li, M. Palanichamy, Z.-Y. Duan, N. Li, Y.-S. Chen, and Y.-P. Zhang. Population phylogenomic analysis of mitochondrial DNA in wild boars and domestic pigs revealed multiple domestication events in East Asia. Genome Biology, 8(11):R245, 2007

An example - a minimum spanning network



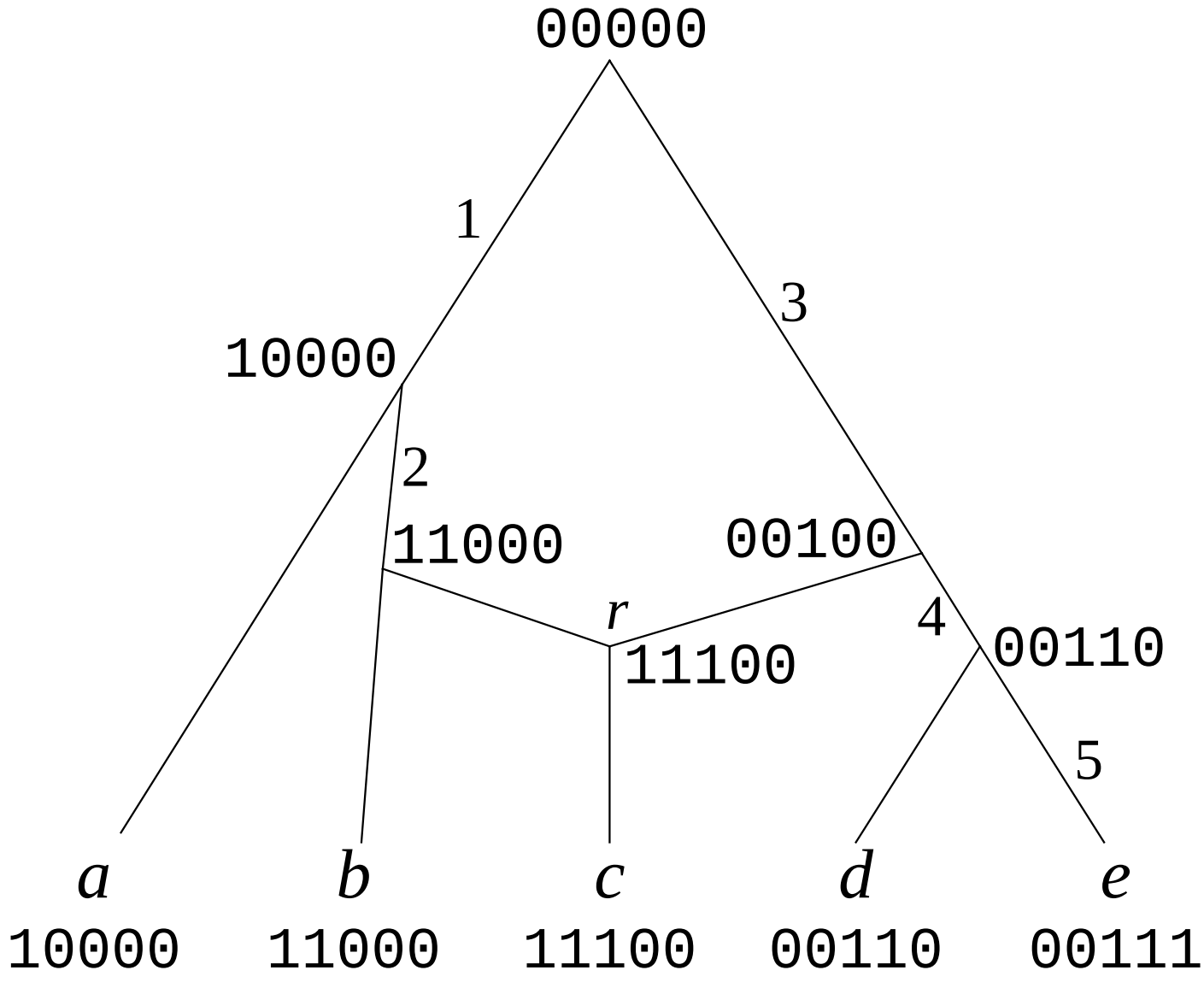
C. M. Miller-Butterworth, D. S. Jacobs, and E. H. Harley. Strong population sub- structure is correlated with morphology and ecology in a migratory bat. Nature, 424(6945):187-191, 2003

An example - a DTLR network



P.J. Planet, S.C. Kachlany, D.H. Fine, R. DeSalle, and D.H. Figurski. The wide spread colonization island of actinobacillus actinomycetemcomitans. Nature Genetics, 34:193–198, 2003.

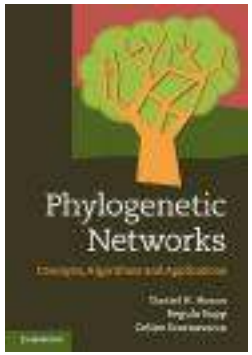
An example - a recombination network

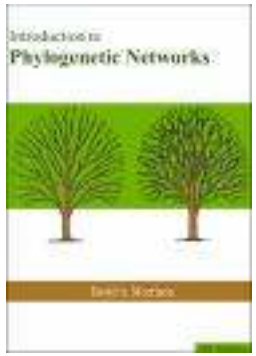


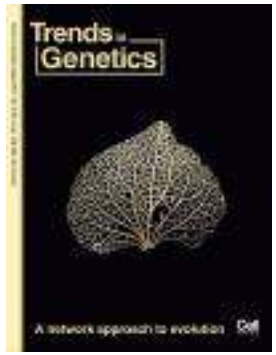
Daniel H. Huson, Regula Rupp, Celine Scornavacca. Phylogenetic Networks. Cambridge University Press. 2011

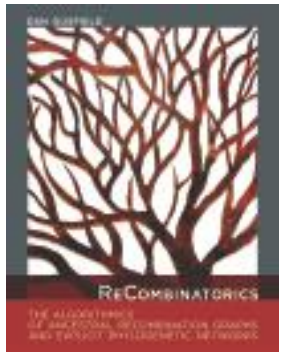
Phylogenetic networks

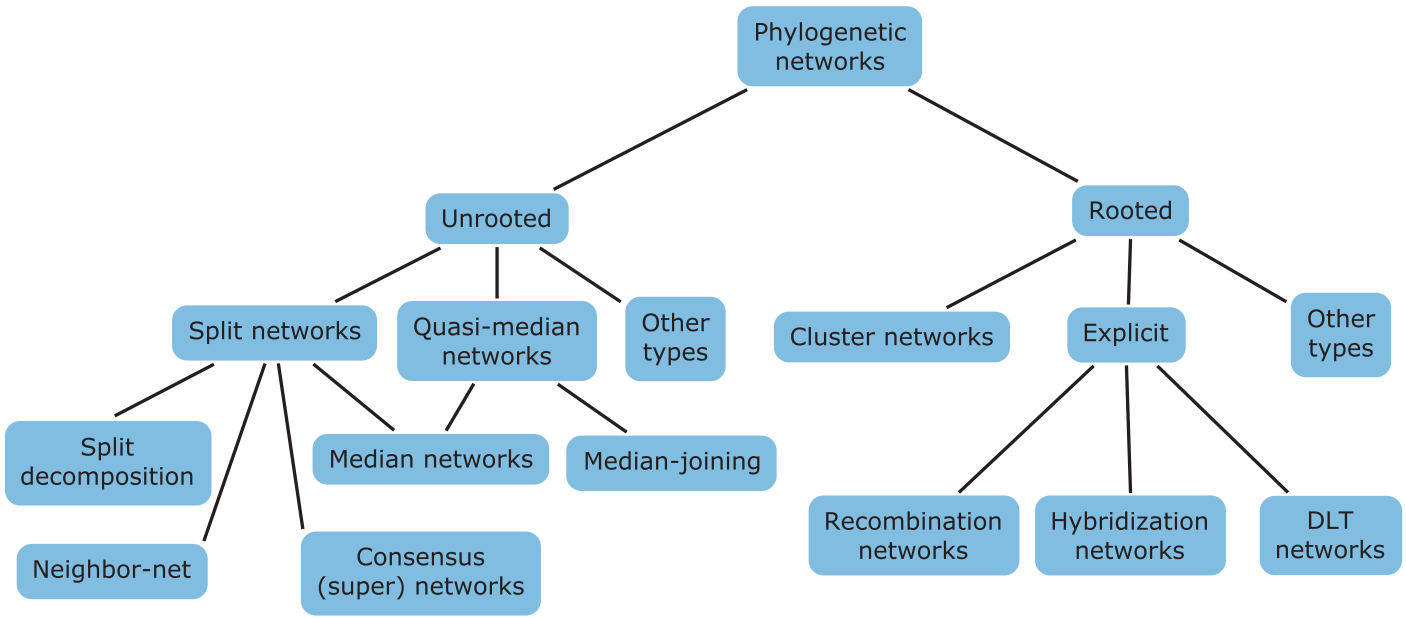
2008 2010 2011 2013 2014





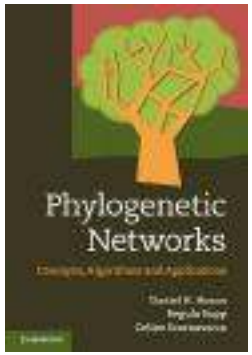


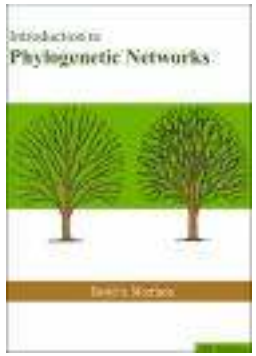


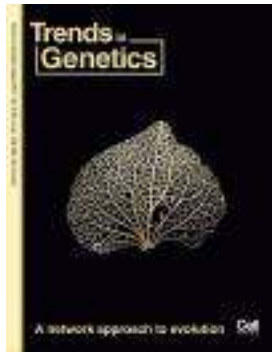


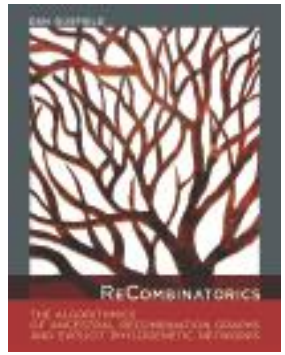
Phylogenetic networks

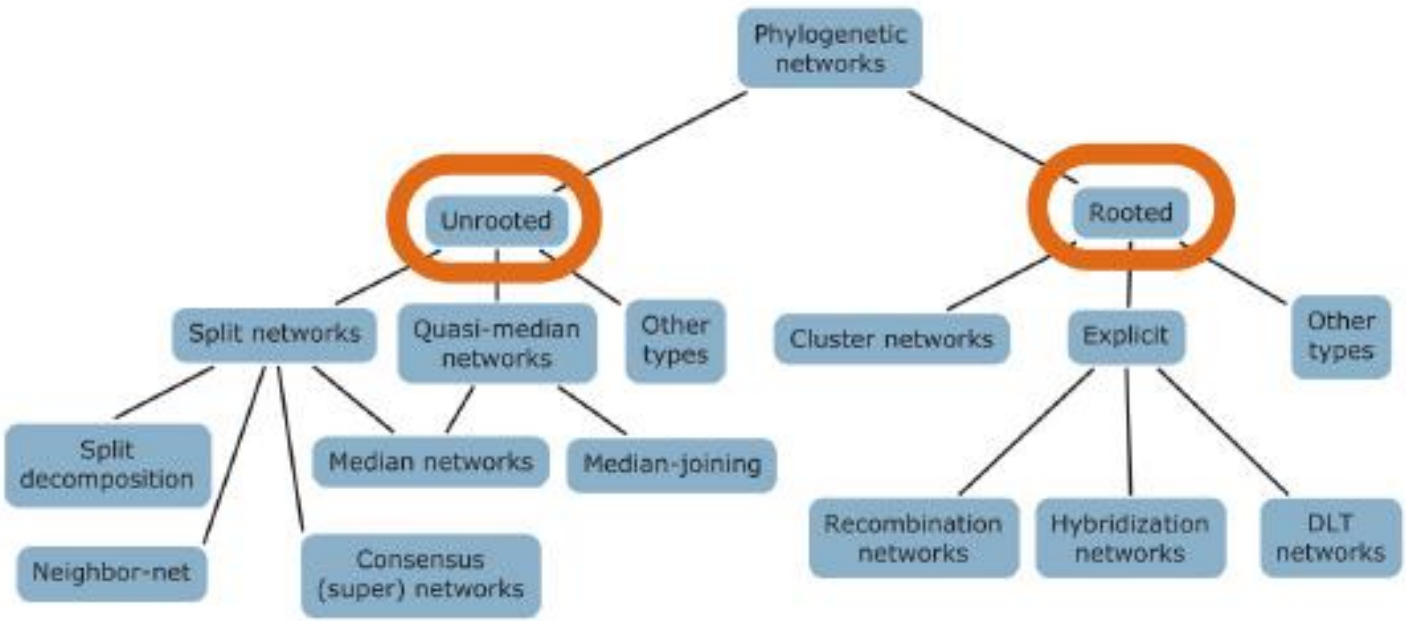
2008 2010 2011 2013 2014





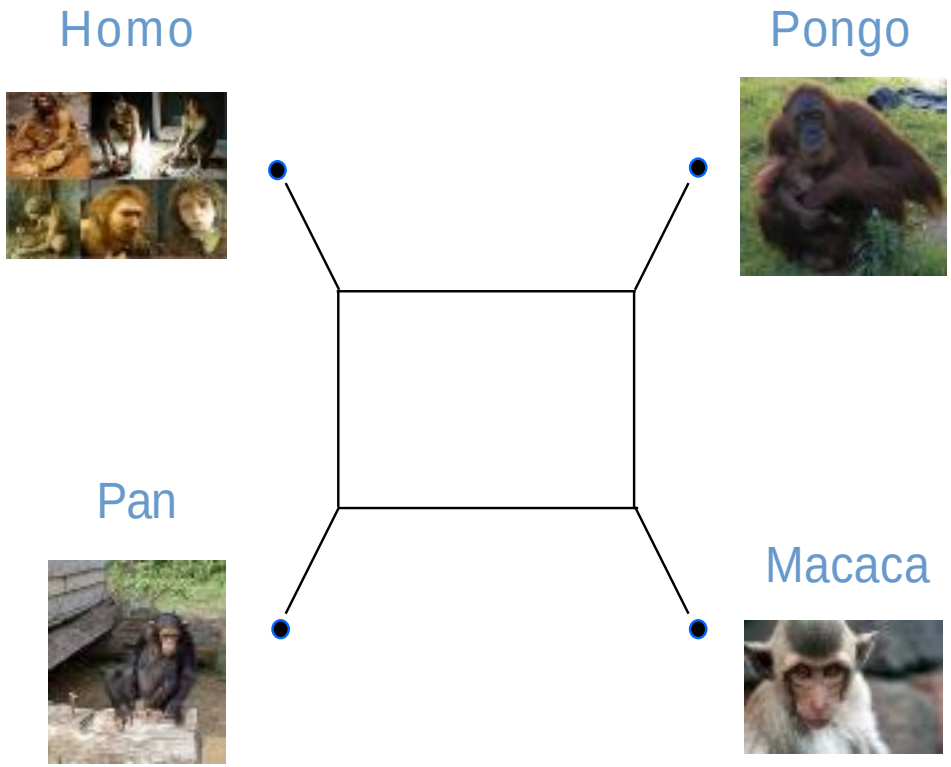






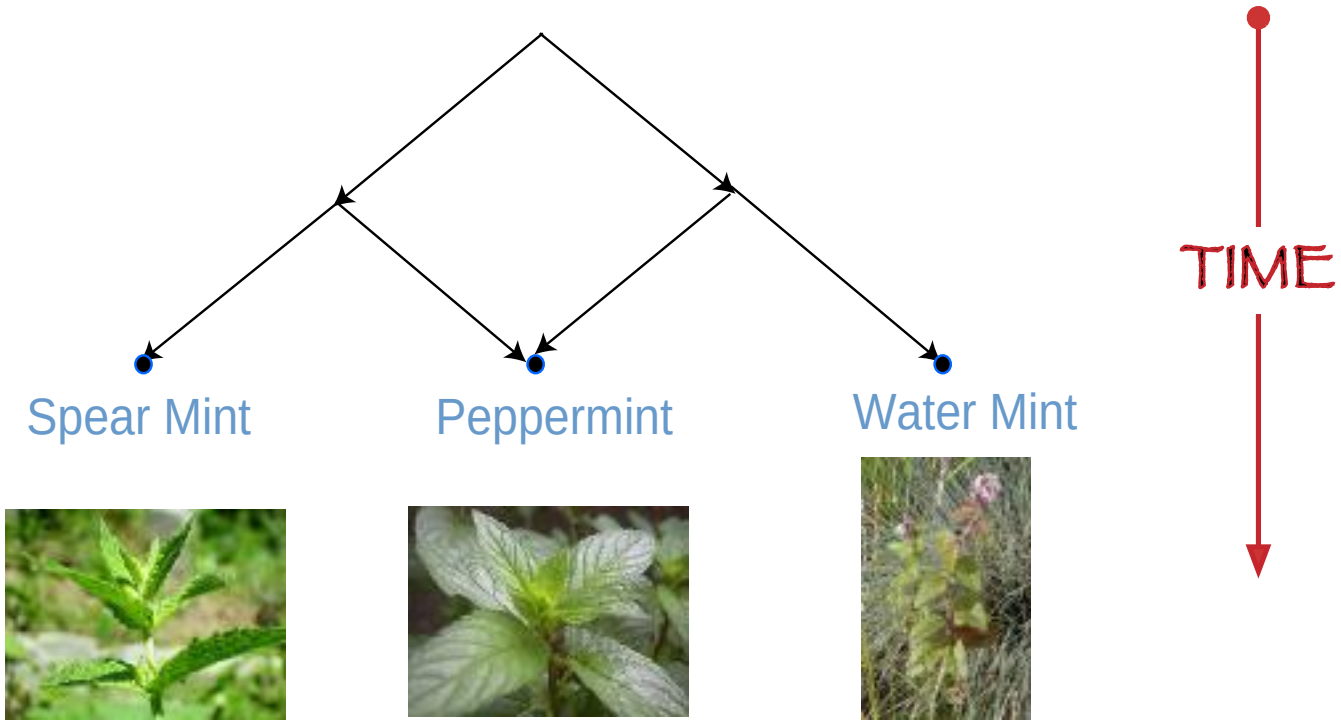
A phylogenetic network ...

... is any connected graph, where terminal nodes are associated to a set of species.



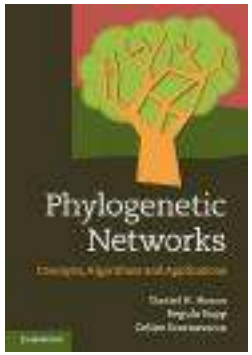
A rooted phylogenetic network ...

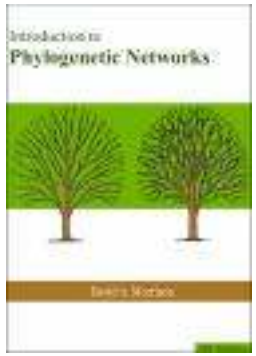
... is any single-rooted directed acyclic graph, where terminal nodes are associated to a set of species.

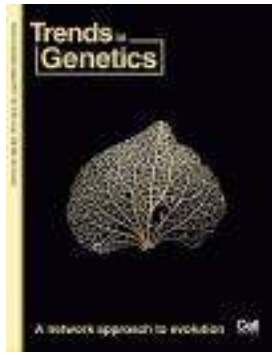


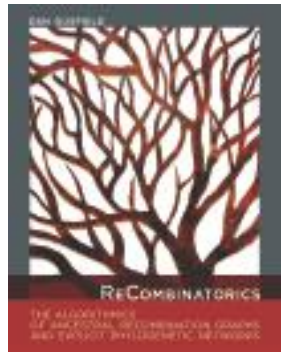
Phylogenetic networks

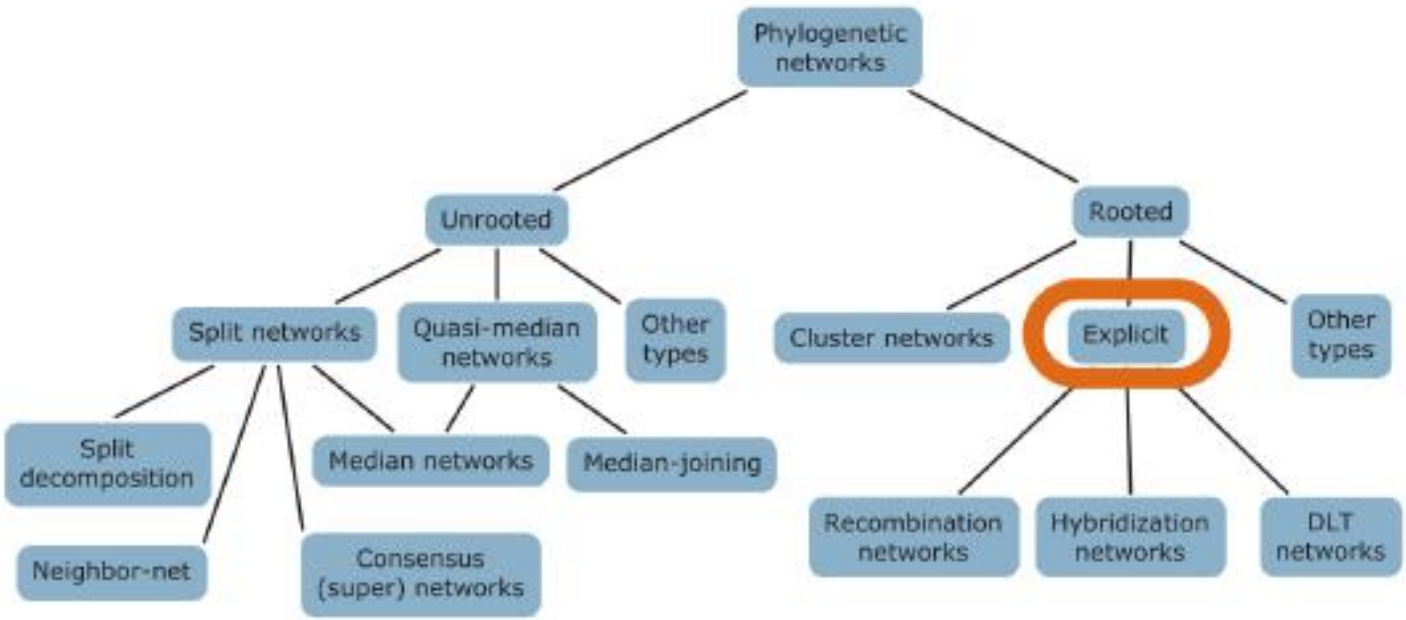
2008 2010 2011 2013 2014





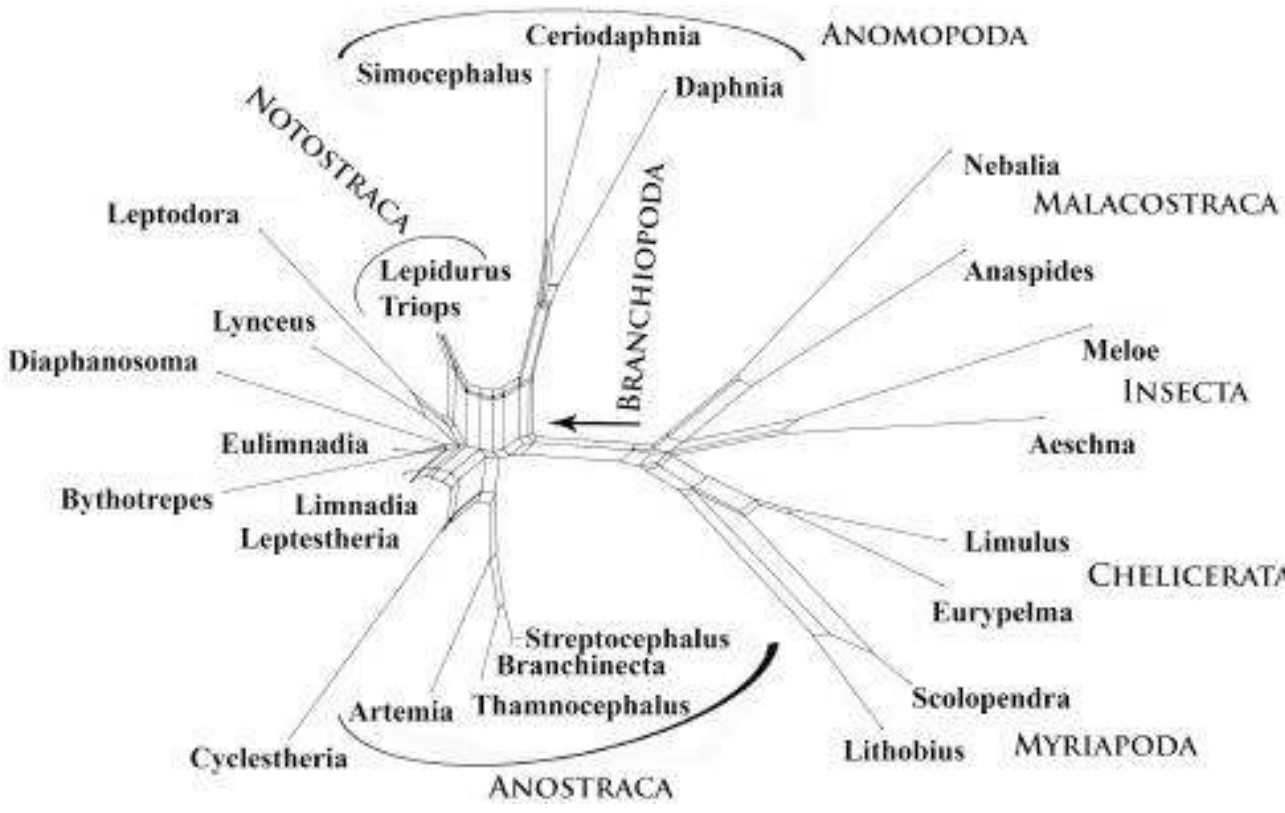






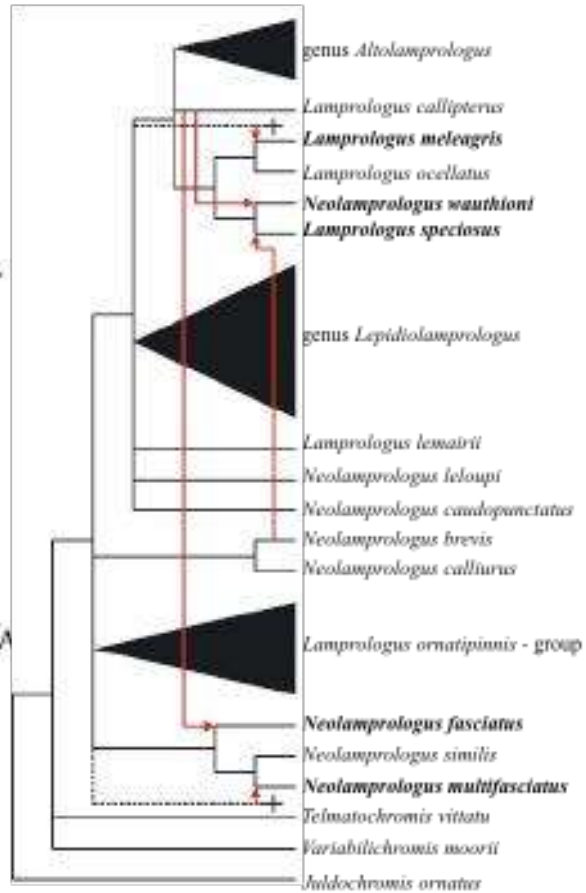
Abstract VS explicit phylogenetic networks

Split network:



Shows conflicting placement of taxa

Hybridization network:



Shows putative hybridization history

The plan of the survey

1. combinatorial and distance methods not accounting for ILS
   - unrooted networks
   - rooted networks (explicit or not)
2. methods accounting for ILS (always explicit)

Unrooted phylogenetic networks

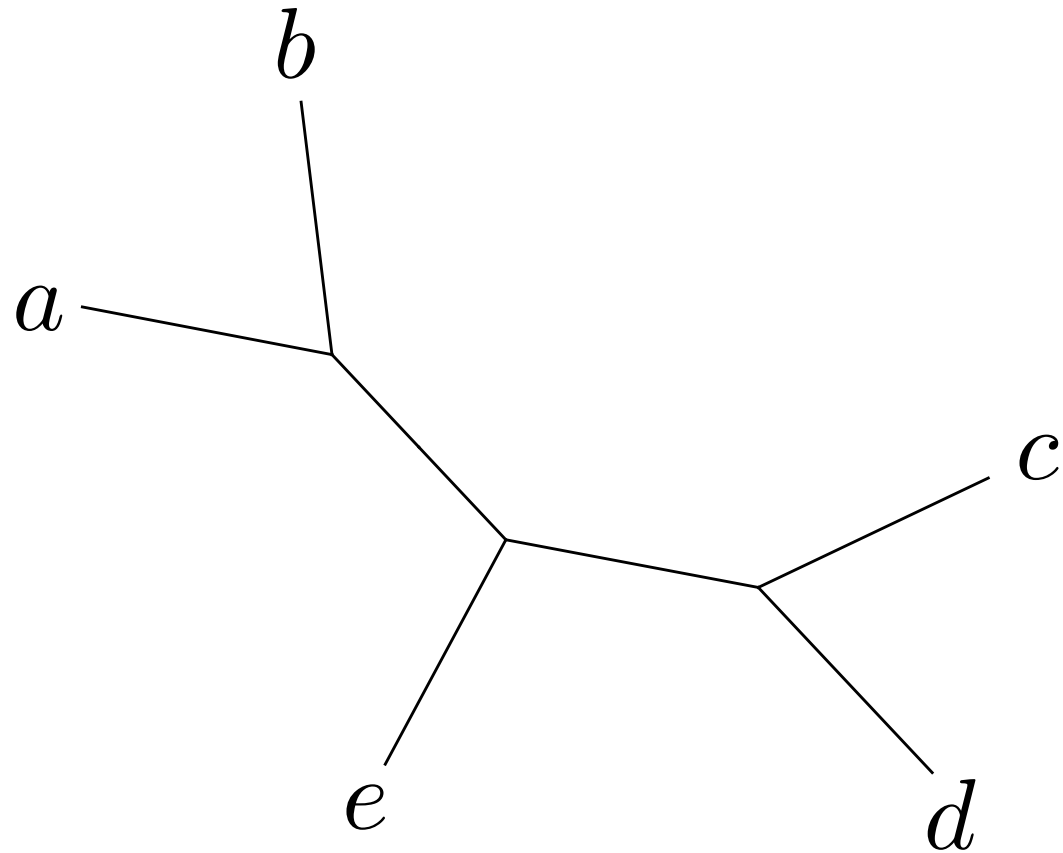

SplitsTree4
by Daniel Huson and David Bryant
with contributions from Markus Franz, Migüel Jette',
Tobias Kloepper and Michael Schröder
www.splitstree.org

Reconstruction of unrooted phylogenetic networks

- from splits
- from distances (via splits or not)
- from trees (via splits)
- from sequences (via splits or not)

Splits

A *split* $A \mid B$ on $\mathcal{X}$ is a partition of a taxon set $\mathcal{X}$ into two non-empty sets.



Compatible splits

Definition (Compatible splits)

Two splits $S_1 = A_1|B_1$ and $S_2 = A_2|B_2$ on $\mathcal{X}$ are called compatible, if one of the following four possible intersections of their split parts is empty: $A_1 \cap A_2$, $A_1 \cap B_2$, $B_1 \cap A_2$ or $B_1 \cap B_2$. Otherwise, the two splits are called incompatible. A set of splits $\mathcal{S}$ is called compatible if all pairs of splits in $\mathcal{S}$ are compatible.

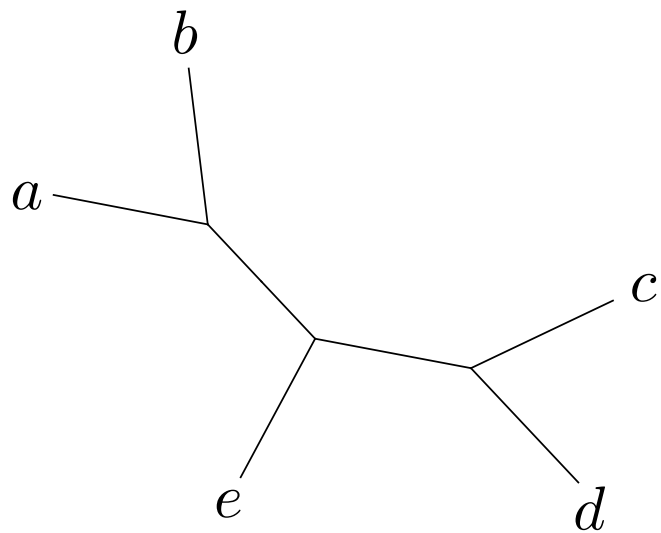
Example

$$\mathcal{S}_1 = \begin{array}{l} \{a\}|\{b,c,d,e\} \\ \{b\}|\{a,c,d,e\} \\ \{c\}|\{a,b,d,e\} \\ \{d\}|\{a,b,c,e\} \\ \{e\}|\{a,b,c,d\} \\ \{a,b\}|\{c,d,e\} \\ \{a,b,e\}|\{c,d\} \end{array} \qquad \mathcal{S}_2 = \begin{array}{l} \{a,b,d,e,h\} \mid \{c,f,g\} \\ \{a,c,d,e,g,h\} \mid \{b,f\} \\ \{a,c,e,g\} \mid \{b,d,f,h\} \\ \{a,c,g\} \mid \{b,d,e,f,h\} \\ \{a,c,e,f,g\} \mid \{b,d,h\} \\ \{a,e,h\} \mid \{b,c,d,f,g\} \end{array}$$

Compatible splits

Theorem (Compatible splits)

Let $\mathcal{S}$ be a set of splits on $\mathcal{X}$ and assume that $\mathcal{S}$ contains all trivial splits on $\mathcal{X}$. There exists a unique unrooted phylogenetic tree T that realizes $\mathcal{S}$,that is, with $\mathcal{S}(T) = \mathcal{S}$, if and only if $\mathcal{S}$ is compatible.



(a) Unrooted tree T

$\{a\}|\{b,c,d,e\}$
$\{b\}|\{a,c,d,e\}$
$\{c\}|\{a,b,d,e\}$
$\{d\}|\{a,b,c,e\}$
$\{e\}|\{a,b,c,d\}$
$\{a,b\}|\{c,d,e\}$
$\{a,b,e\}|\{c,d\}$

(b) Split encoding of T

Circular splits

Definition (Circular splits)

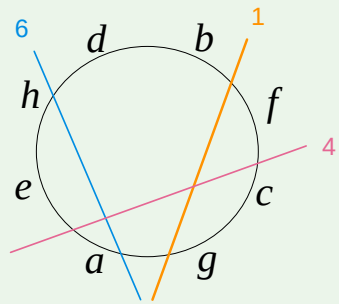
A set of splits $\mathcal{S}$ on $\mathcal{X}$ is called *circular*, if there exists a linear ordering $\pi = (x_1, \ldots, x_n)$ of the elements of $\mathcal{X}$ *for* $\mathcal{S}$ such that each split $S \in \mathcal{S}$ is *interval-realizable*, that is, has the form

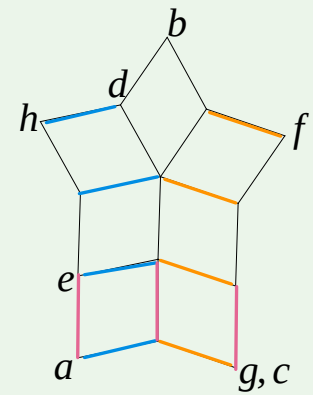
$$S = \{x_p, x_{p+1}, \ldots, x_q\} \mid (\mathcal{X} \setminus \{x_p, x_{p+1}, \ldots, x_q\}),$$

for appropriately chosen $1 < p \leq q \leq n$.

Example

$\{a, b, d, e, h\} \mid \{c, f, g\}$
$\{a, c, d, e, g, h\} \mid \{b, f\}$
$\{a, c, e, g\} \mid \{b, d, f, h\}$
$\{a, c, g\} \mid \{b, d, e, f, h\}$
$\{a, c, e, f, g\} \mid \{b, d, h\}$
$\{a, e, h\} \mid \{b, c, d, f, g\}$





Circular splits

Problem (Consecutive Ones problem)

Let M be a binary matrix. Does there exist a permutation of the columns of the matrix M such that in each row, all ones in the row occur in a single consecutive run?

Example

$\{a, b, d, e, h\} \mid \{c, f, g\}$
$\{a, c, d, e, g, h\} \mid \{b, f\}$
$\{a, c, e, g\} \mid \{b, d, f, h\}$
$\{a, c, g\} \mid \{b, d, e, f, h\}$
$\{a, c, e, f, g\} \mid \{b, d, h\}$
$\{a, e, h\} \mid \{b, c, d, f, g\}$

$$\begin{array}{c} a\ b\ c\ d\ e\ f\ g\ h \\ \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

(a) Input matrix

$$\begin{array}{c} a\ e\ h\ d\ b\ f\ c\ g \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \end{array}$$

(b) Permuted matrix

Circular splits

Problem (Consecutive Ones problem)

Let M be a binary matrix. Does there exist a permutation of the columns of the matrix M such that in each row, all ones in the row occur in a single consecutive run?

Example

$\{a, b, d, e, h\} \mid \{c, f, g\}$
$\{a, c, d, e, g, h\} \mid \{b, f\}$
$\{a, c, e, g\} \mid \{b, d, f, h\}$
$\{a, c, g\} \mid \{b, d, e, f, h\}$
$\{a, c, e, f, g\} \mid \{b, d, h\}$
$\{a, e, h\} \mid \{b, c, d, f, g\}$

$$\begin{array}{cccccccc} a & b & c & d & e & f & g & h \\ \left(\begin{array}{cccccccc} 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \end{array}\right) \end{array}$$

(a) Input matrix

$$\begin{array}{cccccccc} a & e & h & d & b & f & c & g \\ \left(\begin{array}{cccccccc} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{array}\right) \end{array}$$

(b) Permuted matrix

Note

decision problem is polynomial solvable
finding an ordering of X that minimizes the number of runs of ones in the matrix M (Optimal Consecutive Ones problem) is NP-hard.

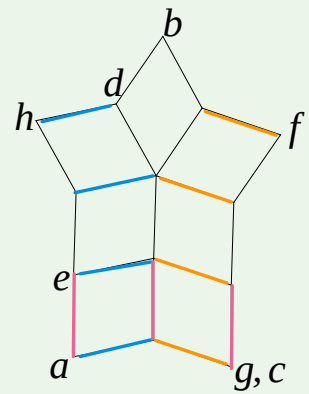
Circular splits

Theorem (Circular implies outer-labeled planar)

A set of splits $\mathcal{S}$ on $\mathcal{X} = \{x_1, ..., x_n\}$ is circular if and only if it can be represented by a split network N that is outer-labeled planar.

Example

$\{a, b, d, e, h\} \mid \{c, f, g\}$
$\{a, c, d, e, g, h\} \mid \{b, f\}$
$\{a, c, e, g\} \mid \{b, d, f, h\}$
$\{a, c, g\} \mid \{b, d, e, f, h\}$
$\{a, c, e, f, g\} \mid \{b, d, h\}$
$\{a, e, h\} \mid \{b, c, d, f, g\}$



Weakly compatible splits

Definition (Weakly compatible splits)

A set of splits $\mathcal{S}$ on $\mathcal{X}$ is called weakly compatible, if any *three* distinct splits in $\mathcal{S}$ are weakly compatible. Three such splits $S_1 = \frac{A_1}{B_1}$, $S_2 = \frac{A_2}{B_2}$, and $S_3 = \frac{A_3}{B_3}$ are called *weakly compatible*, if

1. at least one of the following four intersections is empty:
   $A_1 \cap A_2 \cap A_3$, $A_1 \cap B_2 \cap B_3$, $B_1 \cap A_2 \cap B_3$ and $B_1 \cap B_2 \cap A_3$,
2. at least one of the following four intersections is empty:
   $B_1 \cap B_2 \cap B_3$, $B_1 \cap A_2 \cap A_3$, $A_1 \cap B_2 \cap A_3$ and $A_1 \cap A_2 \cap B_3$.

Example

$$\mathcal{S}_1 = \begin{array}{c} \{a,b,d,e,h\} \mid \{c,f,g\} \\ \{a,c,d,e,g,h\} \mid \{b,f\} \\ \{a,c,e,g\} \mid \{b,d,f,h\} \\ \{a,c,g\} \mid \{b,d,e,f,h\} \\ \{a,c,e,f,g\} \mid \{b,d,h\} \\ \{a,e,h\} \mid \{b,c,d,f,g\} \end{array} \qquad \mathcal{S}_2 = \begin{array}{c} \{a,b,d,e,h\} \mid \{c,f,g\} \\ \{a,c,d,e,g,h\} \mid \{b,f\} \\ \{a,c,e,g\} \mid \{b,d,f,h\} \\ \{a,c,d,e\} \mid \{b,f,g\} \\ \{a,b\} \mid \{c,d,e,f,g\} \\ \{a,e,f\} \mid \{b,c,d,g\} \end{array}$$

Weakly compatible splits

Definition (Weakly compatible splits)

A set of splits $\mathcal{S}$ on $\mathcal{X}$ is called weakly compatible, if any *three* distinct splits in $\mathcal{S}$ are weakly compatible. Three such splits $S_1 = \frac{A_1}{B_1}$, $S_2 = \frac{A_2}{B_2}$, and $S_3 = \frac{A_3}{B_3}$ are called *weakly compatible*, if

1. at least one of the following four intersections is empty:
2. [illegible]

3,

3.

Why all this?!?

Phylogenetic networks reconstructed from weakly compatible are easier than the ones reconstructed from generic splits

Ex

$$\mathcal{S}_1 = \begin{array}{c} \{a,b,d,e,h\} \mid \{c,f,g\} \\ \{a,c,d,e,g,h\} \mid \{b,f\} \\ \{a,c,e,g\} \mid \{b,d,f,h\} \\ \{a,c,g\} \mid \{b,d,e,f,h\} \\ \{a,c,e,f,g\} \mid \{b,d,h\} \\ \{a,e,h\} \mid \{b,c,d,f,g\} \end{array} \qquad \mathcal{S}_2 = \begin{array}{c} \{a,b,d,e,h\} \mid \{c,f,g\} \\ \{a,c,d,e,g,h\} \mid \{b,f\} \\ \{a,c,e,g\} \mid \{b,d,f,h\} \\ \{a,c,d,e\} \mid \{b,f,g\} \\ \{a,b\} \mid \{c,d,e,f,g\} \\ \{a,e,f\} \mid \{b,c,d,g\} \end{array}$$

UPN from distances

or “how to get the splits from distances”

From weighted splits to distances

Any split $S \in \mathcal{S}$ can be used to define a distance matrix D_S on $\mathcal{X}$, by setting:

$$d_S(x, y) = \begin{cases} 1 & \text{if } S \text{ separates } x \text{ and } y, \\ 0 & \text{else.} \end{cases}$$

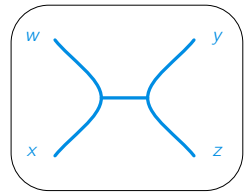
for all taxa x and y in $\mathcal{X}$.
Assume we are given a weighting of a set of splits, $\omega : \mathcal{S} \to \mathbb{R}^{\geq 0}$, then we define the weighted split metric $D_{(\mathcal{S},\omega)}$ as:

$$d_{(\mathcal{S},\omega)}(x, y) = \sum_{S \in \mathcal{S}} \omega(S) \times d_S(x, y)$$

for all taxa x and y in $\mathcal{X}$.

PN from distances: the split decomposition

The split decomposition algorithm [Bandelt and Dress, 1992]: Given a distance matrix D on $\mathcal{X} = \{x_1, \ldots, x_n\}$ we start by computing the isolation index for quartets and splits :



- for any four taxa w, x, y and z with $\{w, x\} \cap \{y, z\} = \emptyset$, but not necessarily $w \neq x$ or $y \neq z$:
  $\hat{\alpha}_D(\frac{\{w,x\}}{\{y,z\}}) = \frac{1}{2}(\max\{d(w,x) + d(y,z), d(w,y) + d(x,z), d(w,z) + d(x,y)\} - d(w,x) - d(y,z))$.
- for any (partial) split S: $\alpha_D(S) = \min\{\hat{\alpha}_D(\frac{\{w,x\}}{\{y,z\}}) \mid w, x \in A,\ y, z \in B\} \geq 0$.

Then, we set $X_0 = \emptyset$ and $\mathcal{S}_0 = \emptyset$. Now, assume that the set of splits $\mathcal{S}_i$ on the first i taxa $\mathcal{X}_i = \{x_1, \ldots, x_i\}$. To obtain $\mathcal{S}_{i+1}$ on $\mathcal{X}_{i+1} = \{x_1, \ldots, x_{i+1}\}$, for each split $\frac{A}{B} \in \mathcal{S}_i$ do:

1. Consider $S = \frac{A \cup \{x_{i+1}\}}{B}$. If $\alpha_D(S) > 0$, set $\omega(S) = \alpha_D(S)$ and add S to $\mathcal{S}_{i+1}$.
2. Consider $S = \frac{A}{B \cup \{x_{i+1}\}}$. If $\alpha_D(S) > 0$, set $\omega(S) = \alpha_D(S)$ and add S to $\mathcal{S}_{i+1}$.
3. Consider $S = \frac{\mathcal{X}_i}{\{x_{i+1}\}}$. If $\alpha_D(S) > 0$, set $\omega(S) = \alpha_D(S)$ and add S to $\mathcal{S}_{i+1}$.

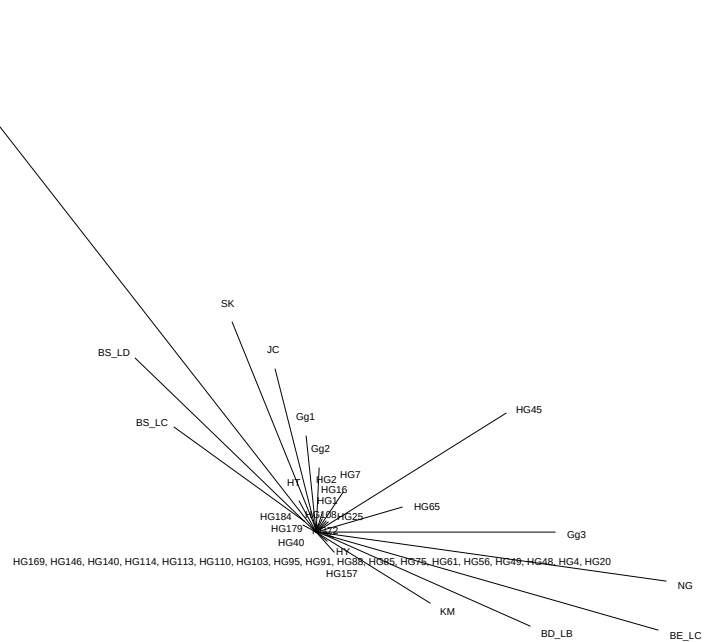
The result is given by $\mathcal{S}_n$.

PN from distances: the split decomposition

- A split S whose isolation index $\alpha_D(S)$ is greater than 0 is called a D-split. D-splits are always weakly compatible.
- It follows from this that the split decomposition always computes a set of weakly compatible splits

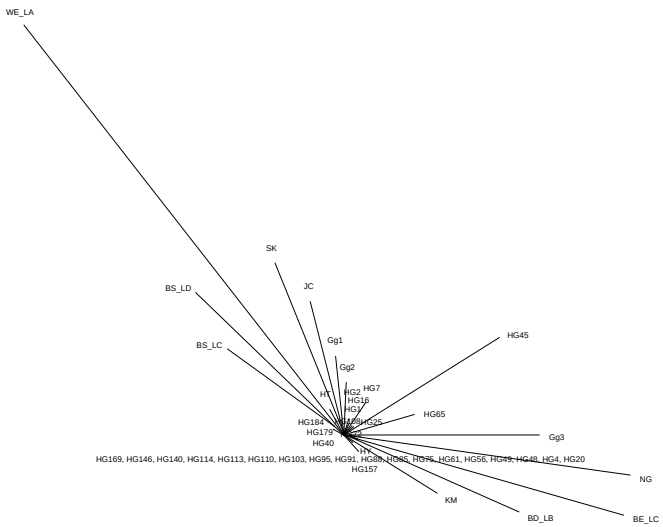
PN from distances: the split decomposition

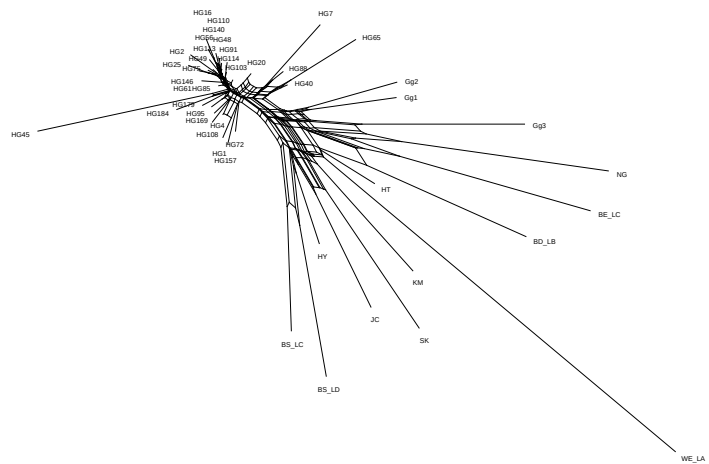
- A split S whose isolation index $\alpha_D(S)$ is greater than 0 is called a D-split. D-splits are always weakly compatible.
- It follows from this that the split decomposition always computes a set of weakly compatible splits
- The SD is a conservative method
- It can be used for small number of taxa or low divergence



PN from distances: Neighbor-Net

- Given a distance matrix D on $\mathcal{X}$, the Neighbor-Net algorithm [Bryant and Moulton, 2004] computes a circular ordering π of $\mathcal{X}$ from D and then a set of weighted splits $\mathcal{S}$ that are interval-realizable with respect to π:
  - produces circular splits
  - uses together with circular network algorithm to get planar networks
  - can be used for large number of taxa and high divergence





PN from distances: Neighbor-Net

- Given a distance matrix D on $\mathcal{X}$, the Neighbor-Net algorithm [Bryant and Moulton, 2004] computes a circular ordering π of $\mathcal{X}$ from D and then a set of weighted splits $\mathcal{S}$ that are interval-realizable with respect to π:
  - produces circular splits
  - uses together with circular network algorithm to get planar networks
  - can be used for large number of taxa and high divergence

Theorem (Neighbor-Net consistency)

Given a circular distance matrix D on $\mathcal{X}$, the Neighbor-Net algorithm produces a circular ordering π and a set of weighted splits $\mathcal{S}$ that are interval-realizable with respect to π such that $D = D(\mathcal{S})$.

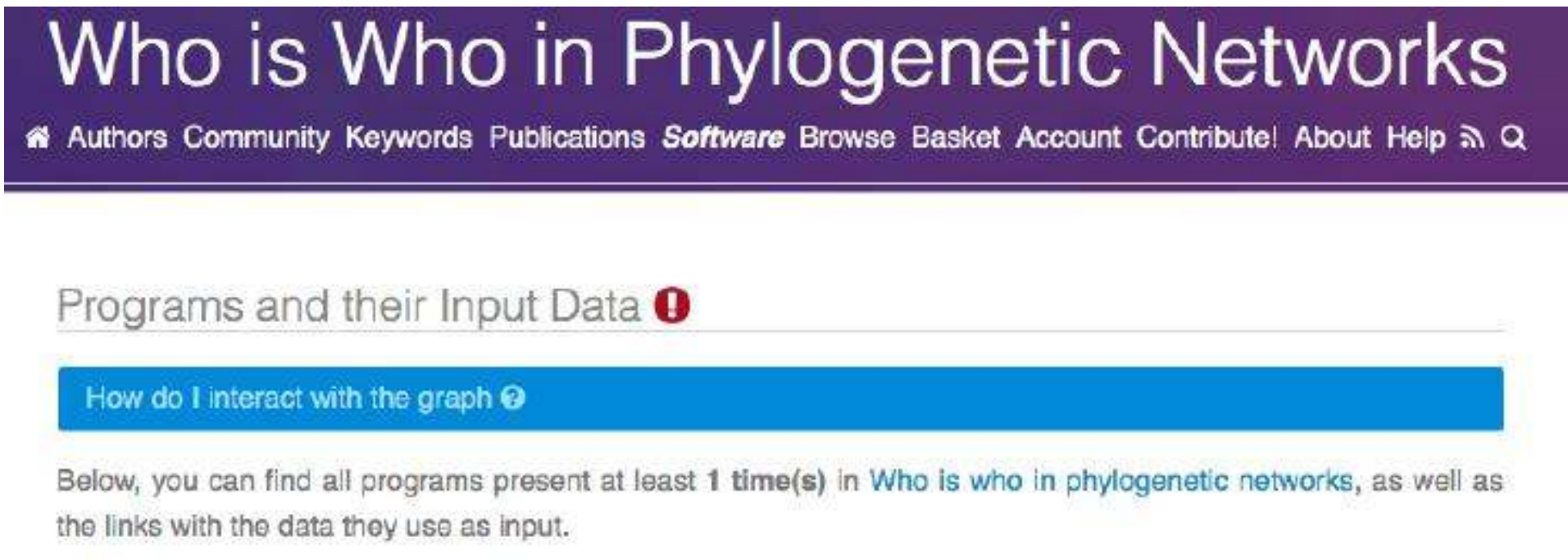
PN from distances

Other algorithms from distances:

- Minimum spanning network
- T-Rex
- ...

A great source of information:

`http://phylnet.univ-mlv.fr/`



UPN from splits

or “what to do with the splits?”

PN from splits: the Convex hull algorithm

Let $\mathcal{S}$ be a set of splits on $\mathcal{X}$ comprising all trivial ones. Assume that we have already computed a split network N for $\mathcal{S}$. To obtain a split network for $\mathcal{S} \cup S$, where $S = \frac{A}{B}$ is a new non-trivial split, modify N as follows:

1. Compute the two convex hulls $H(A)$ and $H(B)$ in N and let M be the split graph induced in N by the set of nodes $H(A) \cap H(B) \neq \emptyset$.
2. Create a copy M' of M and for each node v and edge e in M let v' and e' denote their copies in M', respectively.
3. For every edge f that leads from some node u in $H(B) \setminus H(A) \neq \emptyset$ to a node v in M, redirect the edge f so that it leads from u to v'.
4. Connect each pair of nodes v in M and v' in M' by a new edge $f = (v, v')$ and set $\sigma(f) = S$

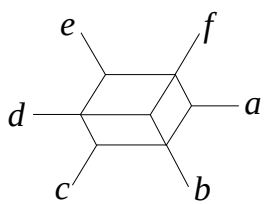
PN from splits: the circular network algorithm

Let $\mathcal{S}$ be a set of splits on $\mathcal{X}$ comprising all trivial ones. Assume that we have already computed a split network N for $\mathcal{S}$. To obtain a split network for $\mathcal{S} \cup S$, where $S = \frac{\{x_p,\ldots,x_q\}}{\mathcal{X}\setminus\{x_p,\ldots,x_q\}}$ is a new non-trivial split, modify N as follows: (splits have to be considered in a certain order)

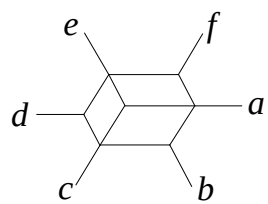
1. Determine the path $M(x_p, x_q)$ and let $\dot{M}$ denote the path obtained by removing the first and last (leaf) edges from $M(x_p, x_q)$.
2. Create a copy $\dot{M}'$ of $\dot{M}$ and for each node v and edge e in $\dot{M}$ let v' and e' denote their copies, respectively.
3. Redirect every edge f that leads from some node $u = \lambda(x_i)$ to some node v in $\dot{M}$ so that it leads from u to v', for all $i = p, \ldots, q$.
4. Connect each pair of nodes v in $\dot{M}$ and v' in $\dot{M}'$ by a new edge $f = (v, v')$ and set $\sigma(f) = S_{t+1}$.

PN from splits: attention!!!

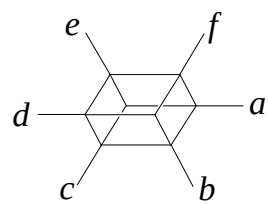
All four different split networks shown below represent the same set of splits. The split networks shown in (a) and (b) were computed using the circular network algorithm processing the splits and taxa in two different orders. The one shown in (c) was constructed using the convex hull algorithm. The split network shown in (d) can be obtained by deleting superfluous edges in any of the first three.



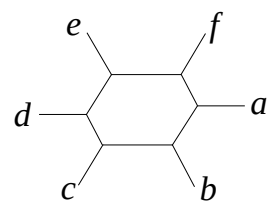
(a)



(b)



(c)



(d)

UPN from trees

or “how to get splits from a bunch of trees”

PN from trees: Consensus split networks

Consensus splits [Holland et al, 2004]

- Input: Trees on identical taxon sets
- Determine splits in more than X% of trees
- For >50%, the result is compatible

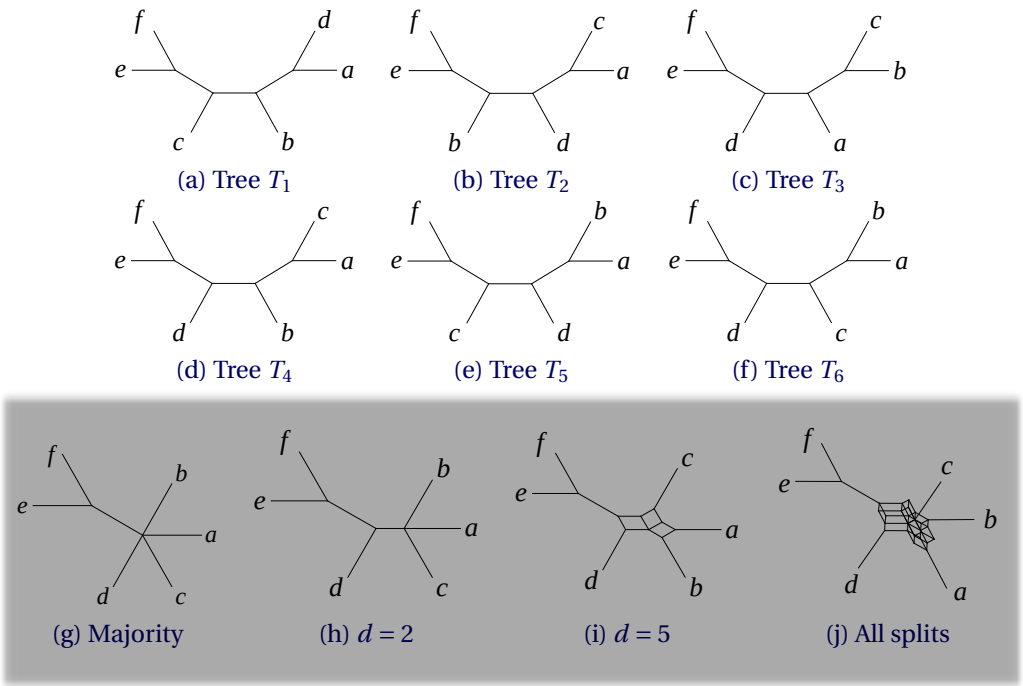


Figure 11.1 (a)– (f) Six different phylogenetic trees $T_1, \ldots, T_6$ on $\mathcal{X} = \{a, b, c, d, e, f\}$. (g) Their majority consensus tree and (h) their consensus split network for $d = 2$, representing all splits that are present in more than $\frac{1}{3}$ of the trees. Note that in this case the network is still a tree, but more resolved than the majority consensus tree. (i) The consensus split network for $d = 5$ and (j) the split network representing all splits present in the six trees.

PN from trees: Consensus split networks

Consensus splits [Holland et al, 2004]

- Input: Trees on identical taxon sets
- Determine splits in more than X% of trees
- For >50%, the result is compatible

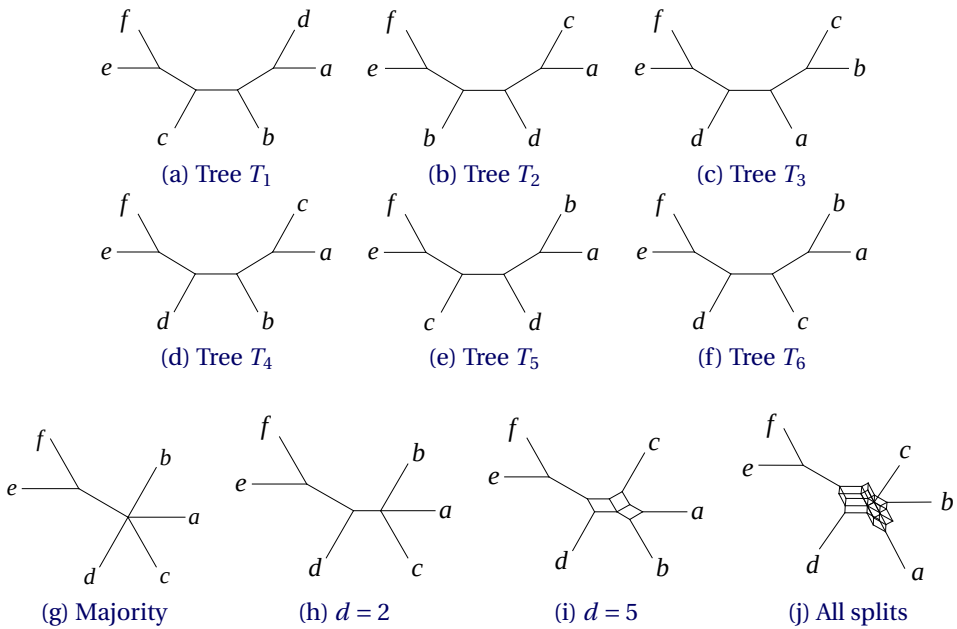


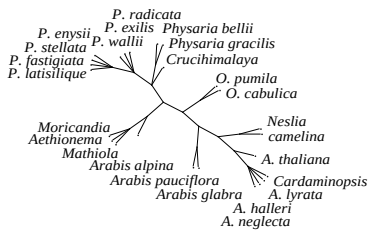
Figure 11.1 (a)– (f) Six different phylogenetic trees $T_1, \ldots, T_6$ on $\mathcal{X} = \{a, b, c, d, e, f\}$. (g) Their majority consensus tree and (h) their consensus split network for $d = 2$, representing all splits that are present in more than $\frac{1}{3}$ of the trees. Note that in this case the network is still a tree, but more resolved than the majority consensus tree. (i) The consensus split network for $d = 5$ and (j) the split network representing all splits present in the six trees.

PN from trees: Consensus super splits networks

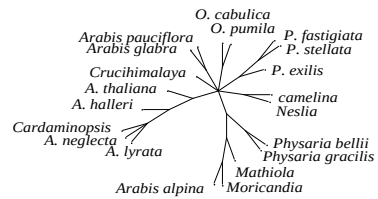
Consensus super splits [Huson et al, 2004, Whitfield et al 2008].

Input: Trees on overlapping taxon sets

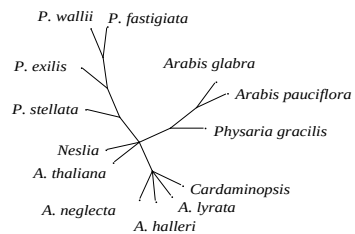
- Use the Z–closure to complete partial splits
- Use the "distortion" values to filter splits



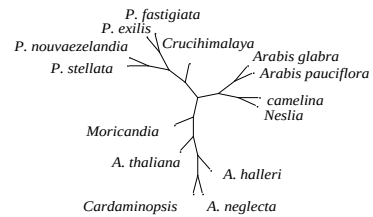
(a) Tree T_1



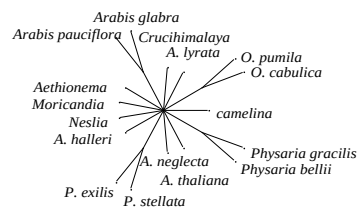
(b) Tree T_2



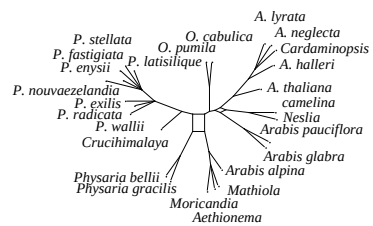
(c) Tree T_3



(d) Tree T_4



(e) Tree T_5



(f) Super network N

Figure 11.3 (a)–(e) Five partial gene trees $T_1, \ldots, T_5$ on 13–25 plant taxa. (f) The corresponding super split network N on all 26 taxa, computed using the Z-closure method. The edges in N are scaled to represent the number of input trees that contain the edge. The network N shows that the placement of the pair of taxa *Physaria bellii* and *Physaria gracilis* differs in the five trees.

The Z-closure

The *Z-closure* method takes a set of partial splits on $\mathcal{X}$ as input and infers a set of complete splits on $\mathcal{X}$ as output.

- Two partial splits $S_1 = \frac{A_1}{B_1} \in \mathcal{S}$ and $S_2 = \frac{A_2}{B_2} \in \mathcal{S}$ are said to be in *Z-relation* to each other, if exactly one of the four intersections $A_1 \cap A_2$, $A_1 \cap B_2$, $B_1 \cap A_2$ or $B_1 \cap B_2$ is empty. In this case, if the empty intersection is $A_1 \cap B_2$, say, then we write $\frac{A_1}{B_1} \, \mathsf{Z} \, \frac{A_2}{B_2}$.
- The *Z-operation* applied to S_1 and S_2 is defined as the creation of two new splits

$$S_1' = \frac{A_1}{B_1 \cup B_2} \text{ and } S_2' = \frac{A_1 \cup A_2}{B_2}.$$

- If at least one of the two new splits contains more taxa than its predecessor, the pair of splits is called *productive*.

We obtain a set of complete splits from a set partial splits $\mathcal{S} = \{S_1, \ldots, S_m\}$ on $\mathcal{X}$ as follows: While $\mathcal{S}$ contains a productive pair of splits $\{S_i, S_j\}$, apply the Z-operation to obtain two new splits $\{S_i', S_j'\}$ and then replace the former pair by the latter pair in $\mathcal{S}$. Finally, add all trivial splits on $\mathcal{X}$.

The distortion values

- Let $\mathcal{T} = (T_1, ..., Tk_)$ be a set of partial trees on $\mathcal{X}$. For any complete split S on $\mathcal{X}$ we define the distortion of S (with respect to $\mathcal{T}$) as $\sigma(S) = \sum_{i=1}^{k} \sigma(T_i, S)$
- $\sigma(T_i, S)$ denotes the minimal homoplasy score for S on the input tree T_i, i.e. the smallest number of (SPR or TBR) branch-swapping operations required to transform some refinement of T_i into a tree that contains the split S
- The distortion of a split can be efficiently computed using dynamic programming

UPN from sequences

Median networks

For a condensed[1] multiple alignment M of binary sequences on $\mathcal{X}$, its median network is a phylogenetic network $N = (V, E, \sigma, \lambda)$ whose node set is given by the median closure $V = \bar{M}$ and in which any two nodes a and b are connected by an edge e of color $\sigma(e) = i \in E$, if any only if they differ in exactly in their i-th position (as haplotypes). An associated taxon labeling $\lambda : X \to V$ maps each taxon x onto the node $\lambda(x)$ that represents the corresponding sequence.

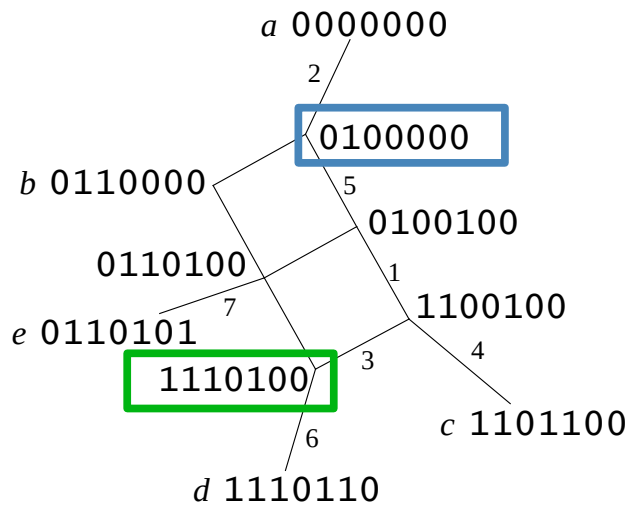
[1] each set of identical sequences is pooled into a single haplotype, then all constant columns are removed and finally, every set of columns that have the same pattern is replaced by a single column i that is assigned a weight $\omega(i)$ that equals the number of columns in the represented set.

Median networks

For a condensed[1] multiple alignment M of binary sequences on $\mathcal{X}$, its median network is a phylogenetic network $N = (V, E, \sigma, \lambda)$ whose node set is given by the median closure $V = \bar{M}$ and in which any two nodes a and b are connected by an edge e of color $\sigma(e) = i \in E$, if any only if they differ in exactly in their i-th position (as haplotypes). An associated taxon labeling $\lambda : X \to V$ maps each taxon x onto the node $\lambda(x)$ that represents the corresponding sequence.

a	0000000
b	0110000
c	1101100
d	1110110
e	0110101

(a) Alignment M



(b) Median network N

[1] each set of identical sequences is pooled into a single haplotype, then all constant columns are removed and finally, every set of columns that have the same pattern is replaced by a single column i that is assigned a weight $\omega(i)$ that equals the number of columns in the represented set.

Quasi median networks

a	A A A A A
b	B B A A A
c	A B A B B
d	A A B B C
e	A A C B C

(a) Input M

a	0 0 0 0 0 0 0 0
b	1 1 0 0 0 0 0 0
c	0 1 0 0 0 1 1 0
d	0 0 1 1 0 1 1 0
e	0 0 1 0 1 1 1 0

(b) Binary expansion M_1

a	0 0 0 0 0 0 0
b	1 1 0 0 0 0 0
c	0 1 0 0 0 1 1
d	0 0 1 1 0 1 0
e	0 0 1 0 1 1 0

(c) Condensed M_1

0 0 0 0 0 0 0
1 1 0 0 0 0 0
0 1 0 0 0 1 1
0 0 1 1 0 1 0
0 0 1 0 1 1 0
0 0 1 0 0 1 0
0 0 0 0 0 1 0
0 1 0 0 0 0 0
0 1 0 0 0 1 0

(d) Median closure M_2

0 0 0 0 0 0 0 0 0
1 1 0 0 0 0 0 0 0
0 1 0 0 0 1 1 1 0
0 0 1 1 0 1 1 0 1
0 0 1 0 1 1 1 0 1
0 0 1 0 0 1 1 0 1
0 0 0 0 0 1 1 0 0
0 1 0 0 0 0 0 0 0
0 1 0 0 0 1 1 0 0

(e) Expanded M_2

A A A A A
B B A A A
A B A B B
A A C B C
A A B B C
A A ⋆ B C
A A A B ⋆
A B A A A
A B A B ⋆

(f) Multi-states M_3

A	0 0 0
B	1 1 0
C	1 0 1

AA⋆BC =
A A A B C
A A B B C
A A C B C

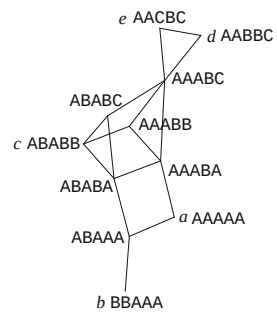
AAAB⋆ =
A A A B A
A A A B B
A A A B C

ABAB⋆ =
A B A B A
A B A B B
A B A B C

(g) Expansion of virtual medians

A A A A A
B B A A A
A B A B B
A A C B C
A A B B C
A A A B C
A A A B A
A A A B B
A B A A A
A B A B A
A B A B C

(h) Final matrix M_4



(i) Quasi-median network N

How to keep the complexity of the network down...

The number of nodes of the quasi-median network can be very large, even for a small number of short sequences. Thus, the quasi-median network is rarely useful in practice. There exist two alternative methods:

- median-joining algorithm, which aims at computing an UPN that is as informative as a quasi-median network, but usually much smaller. The algorithm has a parameter Δ that is used to control how complex the resulting phylogenetic network will be.
- geodesically-pruned quasi-median networks: a method that aims at computing a pruned version of the full quasi-median network by considering only those sequences that lie on a geodesic between two of the original input sequences.

How to keep the complexity of the network down...

UPN from ...

quartets ... QNet

`http://www2.cmp.uea.ac.uk/~vlm/qnet/`

`http://phylnet.univ-mlv.fr/`

Rooted phylogenetic networks

Dendroscope 3

by Daniel H. Huson

with contributions from Benjamin Albrecht.
Philippe Gambette, Leo van Iersel,
Celine Scornavacca and others.

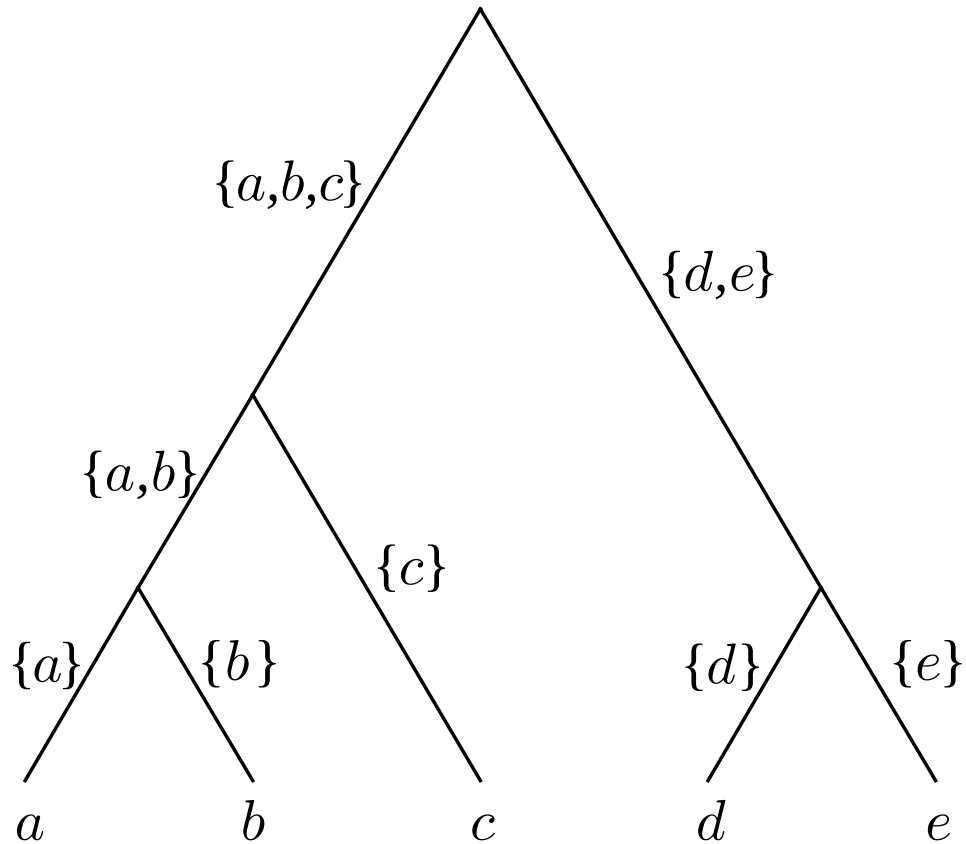
www-ab.informatik.uni-tuebingen.de/software/dendroscope

Reconstruction of rooted phylogenetic networks

- from clusters
- from trees (via clusters or not)
- from sequences
- ~~from distances~~

Clusters

A *cluster* C on $\mathcal{X}$ is a subset of the taxon set $\mathcal{X}$.



Compatible clusters

Definition (Compatible splits)

Two clusters C_1 and C_2 on X are called compatible, if $C_2 \subseteq C_1$ or $C_1 \subseteq C_2$ or $C_1 \cap C_2 = \emptyset$. Otherwise, the two clusters are called incompatible. A set of clusters $\mathcal{C}$ is called compatible if all pairs of clusters in $\mathcal{C}$ are compatible.

Example

$$\mathcal{C}_1 = \begin{matrix} \{a\} \\ \{b\} \\ \{c\} \\ \{d\} \\ \{e\} \\ \{a,b\} \\ \{d,e\} \\ \{a,b,c\} \\ \{a,b,c,d,e\} \end{matrix} \qquad \mathcal{C}_2 = \begin{matrix} \{a\} \\ \{b\} \\ \{c\} \\ \{d\} \\ \{e\} \\ \{a,b\} \\ \{d,e\} \\ \{a,b,e\} \\ \{a,b,c,d,e\} \end{matrix}$$

Compatible clusters

Definition (Compatible splits)

Two clusters C_1 and C_2 on X are called compatible, if $C_2 \subseteq C_1$ or $C_1 \subseteq C_2$ or $C_1 \cap C_2 = \emptyset$. Otherwise, the two clusters are called incompatible. A set of clusters $\mathcal{C}$ is called compatible if all pairs of clusters in $\mathcal{C}$ are compatible.

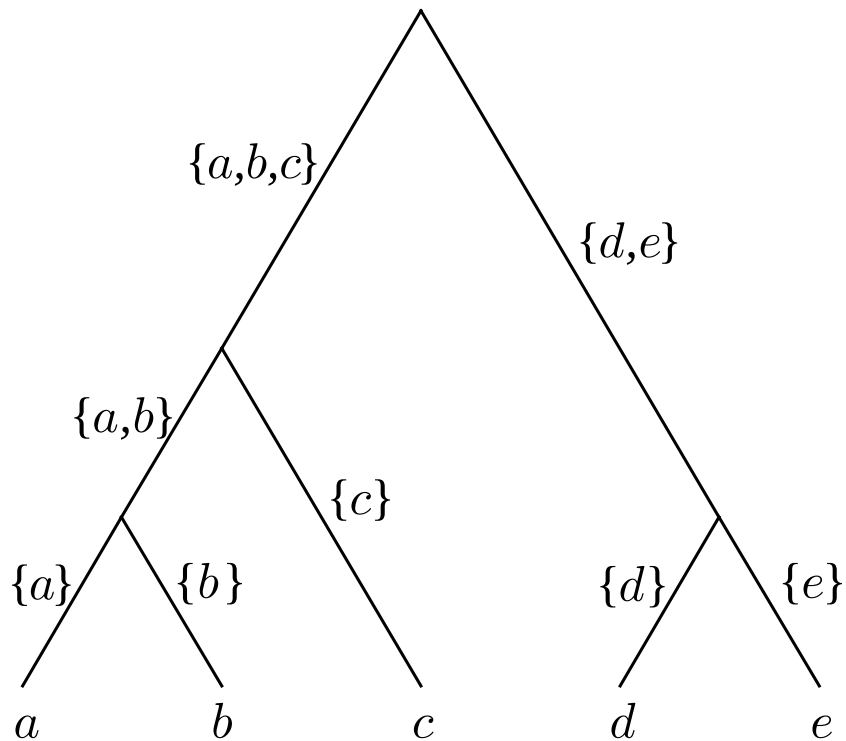
Example

$$\mathcal{C}_1 = \begin{matrix} \{a\} \\ \{b\} \\ \{c\} \\ \{d\} \\ \{e\} \\ \{a,b\} \\ \{d,e\} \\ \{a,b,c\} \\ \{a,b,c,d,e\} \end{matrix} \qquad \mathcal{C}_2 = \begin{matrix} \{a\} \\ \{b\} \\ \{c\} \\ \{d\} \\ \{e\} \\ \{a,b\} \\ \{d,e\} \\ \{a,b,e\} \\ \{a,b,c,d,e\} \end{matrix}$$

Compatible clusters

Theorem (Compatible clusters)

Let $\mathcal{C}$ be a set of clusters on $\mathcal{X}$ and assume that $\mathcal{C}$ contains all trivial splits on $\mathcal{X}$. There exists a unique rooted phylogenetic tree T that realizes $\mathcal{C}$,that is, with $\mathcal{C}(T) = \mathcal{C}$, if and only if $\mathcal{C}$ is compatible.



RPN from clusters

When a phyl. network N represents a cluster C?

HARDWIRED SENSE : if there exists a tree edge of N such that the set of all taxa below the edge equals C

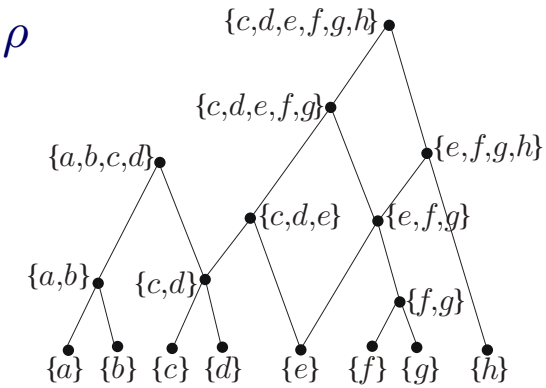
$e_1 : \{a, b, c\}$

$e_2 : \{c, d, e\}$

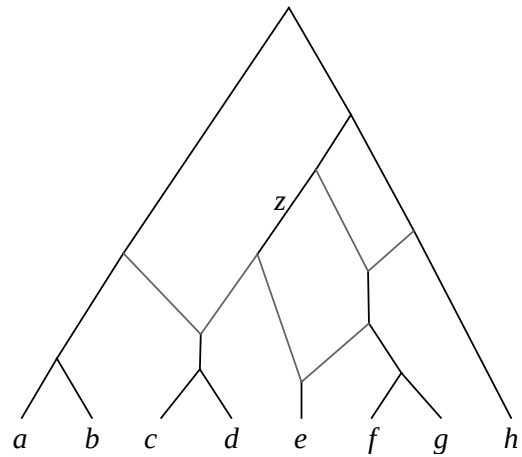
RPN from clusters - Cluster networks

The Cluster-popping algorithm [Huson & Rupp, 2008]:

for each cluster $C \in \mathcal{C}$ create a node u and define $\nu(C) = u$ and $\nu^{-1}(u) = C$.
Create a root node ρ and set $\nu^{-1}(\rho) = \mathcal{X}$
for each $C \in \mathcal{C}$ in order of non-increasing cardinality **do**
    Unmark all nodes
    Push ρ onto a stack S and mark ρ
    while S is not empty **do**
        Pop v off S
        Set *isBelow* = *false*
        for each child w of v **do**
            if $C \subset \nu^{-1}(w)$ **then**
                Set *isBelow* = *true*
                if w is unmarked **then** Mark w and push w onto S
        if *isBelow* = *false* **then** Create a new edge $(v, \nu(C))$
for each node v with indegree ≥ 2 and outdegree $\neq 1$ **do**
    Create a new node v'
    Redirect all in-edges of v to v'
    Create a new edge (v', v)
for each taxon $a \in \mathcal{X}$ **do**
    Set $\lambda(a) = \nu^{-1}(\{a\})$
end



(a) Hasse diagram

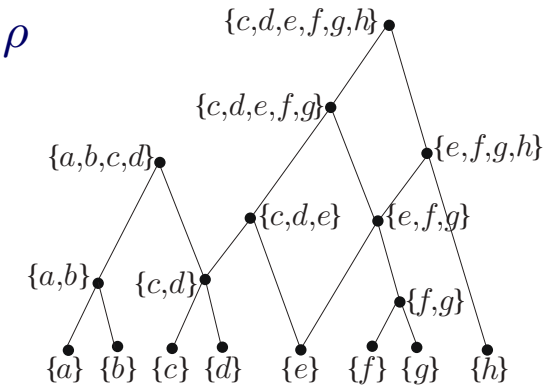


(b) Cluster network

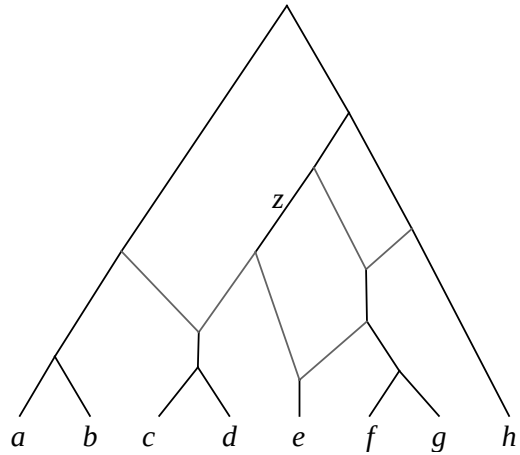
RPN from clusters - Hardwired sense

The Cluster-popping algorithm [Huson & Rupp, 2008]:

for each cluster $C \in \mathcal{C}$ create a node u and define $\nu(C) = u$ and $\nu^{-1}(u) = C$.
Create a root node ρ and set $\nu^{-1}(\rho) = \mathcal{X}$
for each $C \in \mathcal{C}$ in order of non-increasing cardinality **do**
    Unmark all nodes
    Push ρ onto a stack S and mark ρ
    while S is not empty **do**
        Pop v off S
        Set *isBelow* = *false*
        for each child w of v **do**
            if $C \subset \nu^{-1}(w)$ **then**
                Set *isBelow* = *true*
                if w is unmarked **then** Mark w and push w onto S
        if *isBelow* = *false* **then** Create a new edge $(v, \nu(C))$
for each node v with indegree ≥ 2 and outdegree $\neq 1$ **do**
    Create a new node v'
    Redirect all in-edges of v to v'
    Create a new edge (v', v)
for each taxon $a \in \mathcal{X}$ **do**
    Set $\lambda(a) = \nu^{-1}(\{a\})$
end



(a) Hasse diagram



(b) Cluster network

Problem

The networks obtained in this way are too complex

When a phyl. network N represents a cluster C?

SOFTWIRED SENSE : if there exists a tree edge of N such that the set of all taxa below the edge equals C (with one edge per reticulation node "switched on")

$e_1 : \{a, b, c\}$
$\{a, b\}$

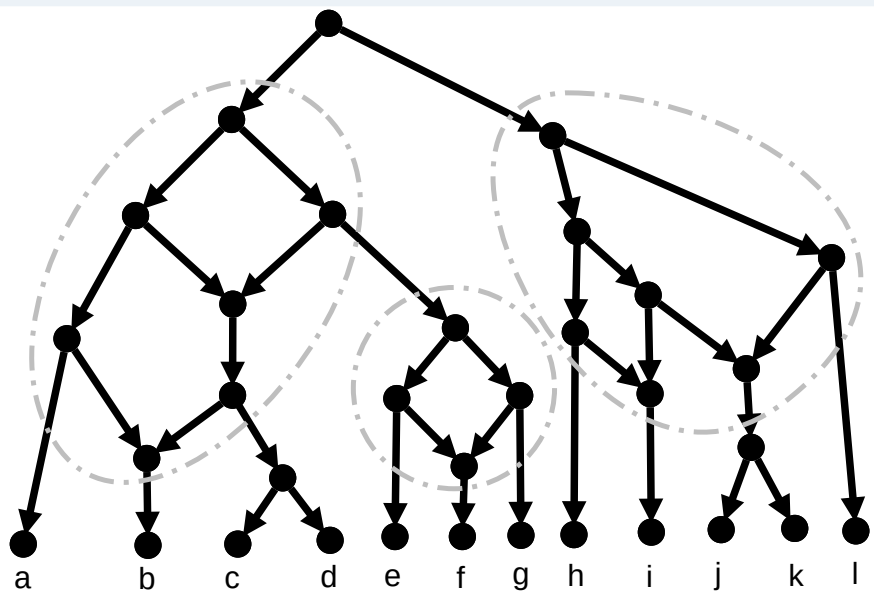
$e_2 : \{c, d, e\}$
$\{d, e\}$

Constructing minimal softwired networks

- cluster containment: NP-hard
- minimization NP-hard, APX-hard

A possible solution ... topological constraints:

- galled trees (if every non-trivial biconnected component of N properly contains exactly one reticulation) (it does not always exist)
- galled networks (if every reticulation in N has a tree cycle) (still NP-hard)
- level-k networks (maximum reticulation number among biconnected components of N is k) (still NP-hard)



RPN from clusters - Softwired sense

CASS algorithm [van Iersel et al., 2010], guaranteed to construct minimal networks if level is 1 or 2:

Algorithm 8.5.1 (Level-k networks for tangled clusters) *Let $k \geq 0$ be a fixed number. Input is a set of clusters $\mathscr{C}$ on $\mathscr{X}$, which is assumed to be tangled in the initial call to the algorithm. Recursively construct a set of level-k networks N for $\mathscr{C}$ in the following steps (if any such network exists):*

Set $\mathscr{N} = \emptyset$
if $k = 0$ **then**
  if *$\mathscr{C}$ is compatible* **then**
    Compute the rooted phylogenetic tree T for $\mathscr{C}$ (and all trivial clusters on $\mathscr{X}$)
    Insert an auxiliary edge that separates the root from the rest of the tree T
    Set $\mathscr{N} = \{T\}$
else *(comment: $k > 0$)*
  Create a new auxiliary taxon z
  for each *taxon $x \in \mathscr{X} \cup \{z\}$* **do**
    Set $\mathscr{C}' = \mathscr{C}|_{\mathscr{X}-\{x\}}$, remove all trivial clusters and then collapse to obtain $\mathscr{C}''$ on $\mathscr{X}''$
    Recursively compute the set $\mathscr{N}''$ of level-$(k-1)$ networks for $\mathscr{C}''$ on $\mathscr{X}''$
    for each *network N'' in $\mathscr{N}''$* **do**
      Replace each leaf of N'', which is labeled by a composite taxon $\bar{A} \in \mathscr{X}''$, by the rooted phylogenetic tree $T(\mathscr{C}'|_A)$ that represents $\mathscr{C}'|_A$
      Let N' be the resulting network
      for each *pair of (not necessarily distinct) edges e_1 and e_2 in a new copy of N'* **do**
        Create two nodes r and u, add an edge from r to u and set $\lambda(x) = u$
        Insert a new node v_1 into e_1 and connect v_1 to r
        Insert a new node v_2 into e_2 and connect v_2 to r
        if *the resulting network N represents $\mathscr{C}$ (disregarding all auxiliary taxa)* **then**
          add N to $\mathscr{N}$
return $\mathscr{N}$

If the returned set $\mathscr{N}$ is empty, then the algorithm reports fail.

$\{a,g\}, \{b,c\}, \{d,e\}, \{a,b,f\}, \{b,c,f\}, \{c,d,e\}, \{b,c,d,f\}, \{a,b,f,g\}$

(a) Set of tangled input clusters $\mathscr{C}$

$\{a,g\}, \{d,e\}, \{a,b,f\}, \{b,f\}, \{b,d,f\}, \{a,b,f,g\}$

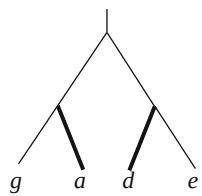
(b) Set $\mathscr{C}' = \mathscr{C}|_{\mathscr{X}-\{c\}}$

$\{a,g\}, \{d,e\}, \{a,\{b,f\}\}, \{\{b,f,\}d\}, \{a,\{b,f\},g\}$

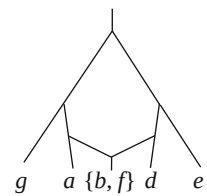
(c) Set $\mathscr{C}''$ obtained by collapsing $\mathscr{C}'$

$\{a,g\}, \{d,e\}$

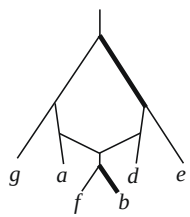
(d) Result of removing $\{b,f\}$ from $\mathscr{C}''$



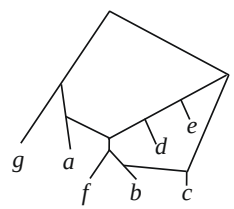
(e) Rooted tree T for set of clusters in (d)



(f) Level-1 network for set of clusters $\mathscr{C}''$ in (c)



(g) Level-1 network for set of clusters $\mathscr{C}'$ in (b)



(h) Final level-2 network for set $\mathscr{C}$ in (a)

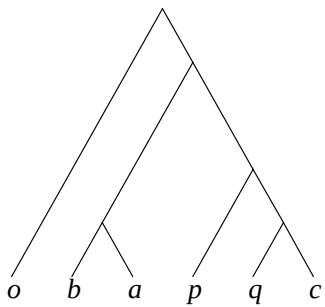
RPN from trees (via clusters)

RPN from trees - option 1

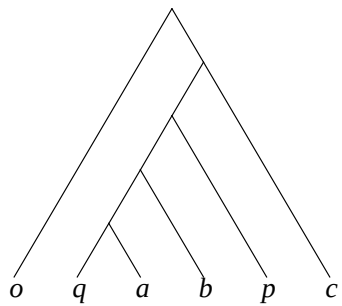
- decompose the trees in clusters
- apply the cluster methods to (a part of) the clusters

RPN from trees - option 1

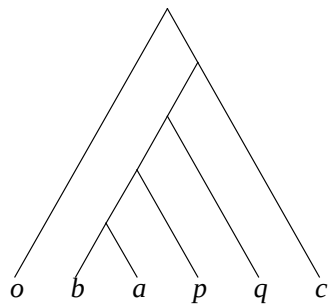
- decompose the trees in clusters
- apply the cluster methods to (a part of) the clusters



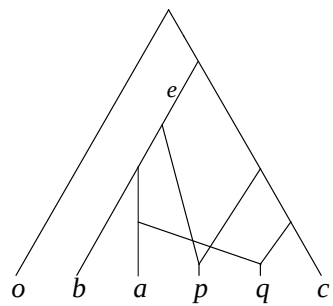
(a) Tree T_1



(b) Tree T_2



(c) Tree T_3

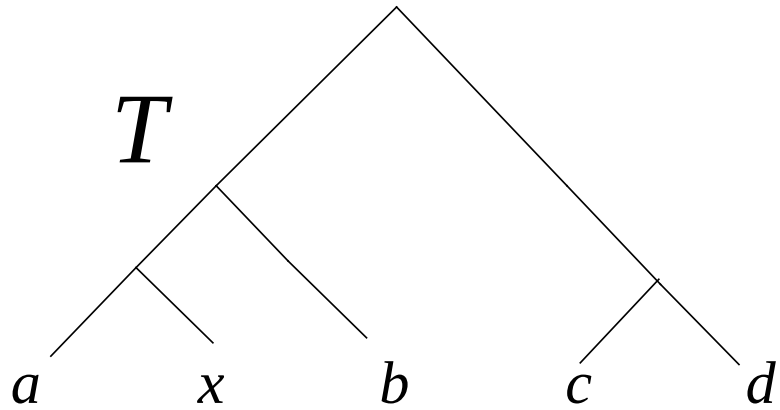


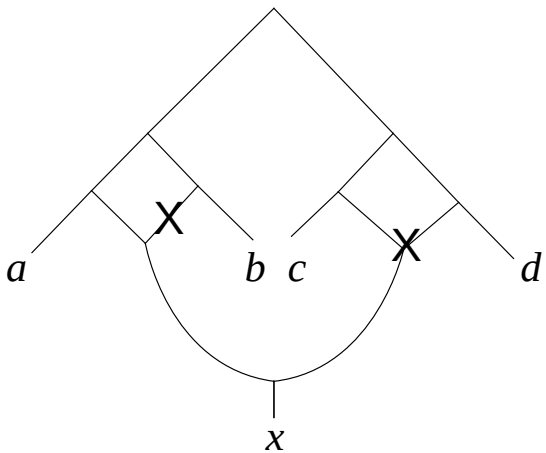
(d) Network N

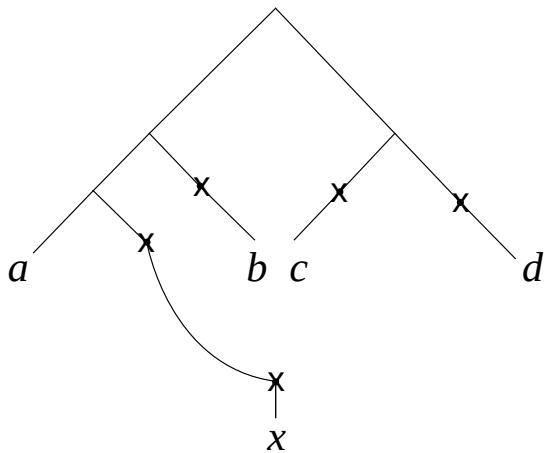
RPN from trees (NOT via clusters)

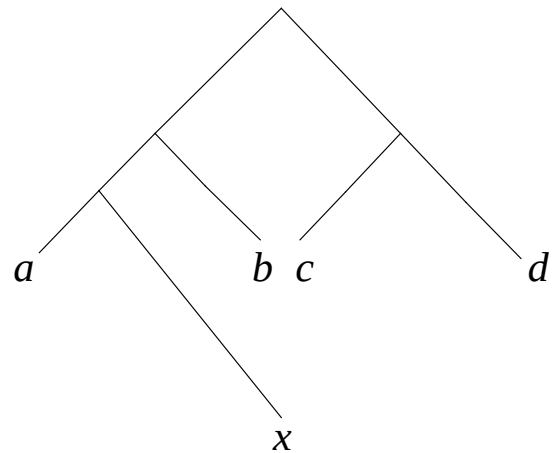
When a phyl. network N embeds a tree T?

if T can be obtained from N by performing a series of node deletions, edge deletions and node suppressions









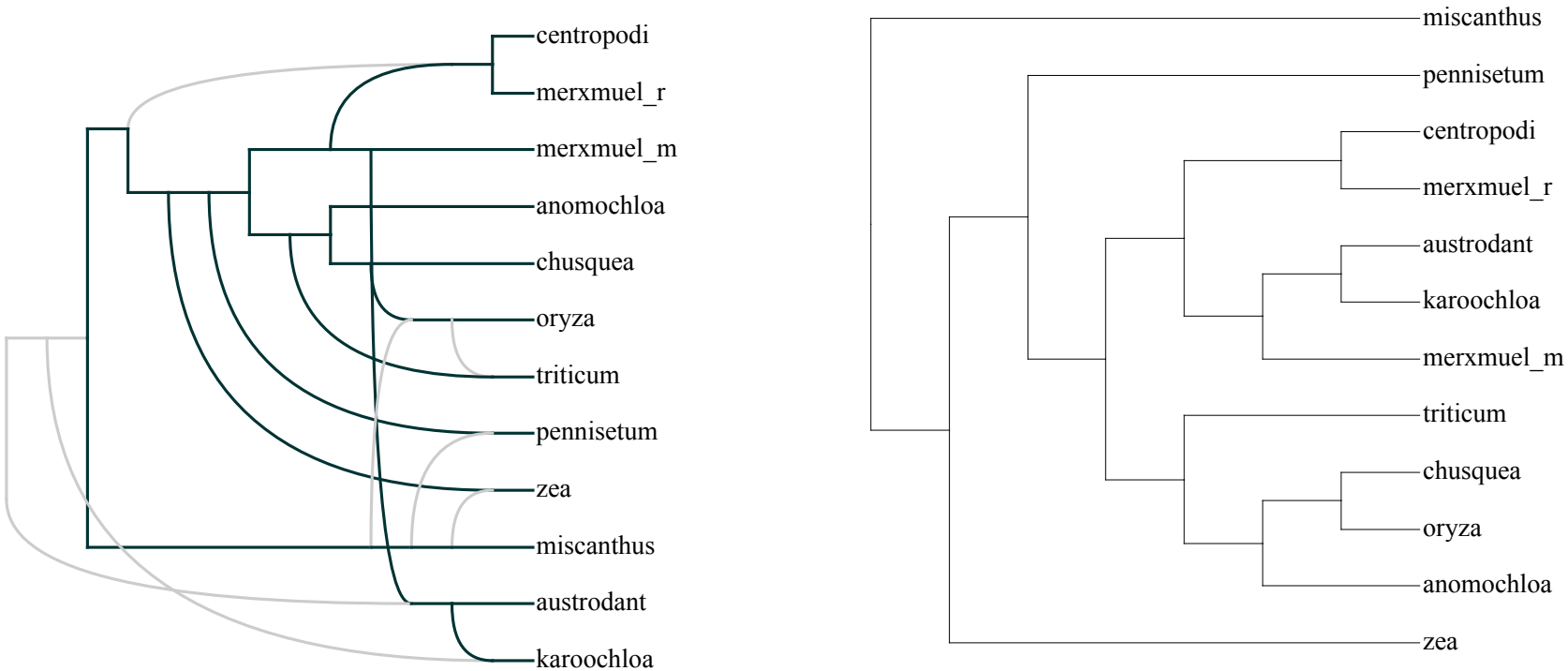
Problem

Given:
A set of (binary) trees (with same taxa set) and **different** topology.

Question:
What is the **most probable** evolutionary history?
Assumptions: Difference is caused by hybridizations, parsimony

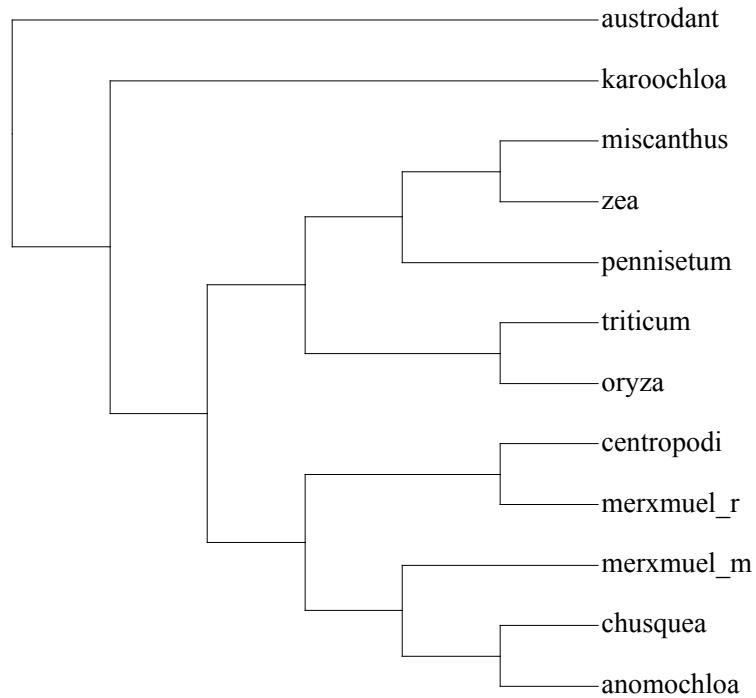
Answer:
Network embedding the trees with a **minimal** number of hybridization nodes.

Tree embedding



Hybridization network (left) highlighting the embedding of the phylogenetic tree based on gene *rbcL* (right).

Tree embedding



Hybridization network (left) highlighting the embedding of the phylogenetic tree based on gene *waxy* (right).

Agreement forests

An agreement forest for two rooted bifurcating phylogenetic trees T_1 and T_2 on $\mathcal{X} \cup \rho$ is a set of components $\mathcal{F} = \{F_\rho, F_1, \ldots, F_n\}$ on $\mathcal{X} \cup \rho$ such that...

1. ... the taxon ρ is contained in F_ρ.
2. ... each component F_i is a restricted subtree of T_1 and T_2.
3. ... the trees in $\{T_1(\mathcal{X}_i | i = \rho, 1, \ldots, n)\}$ and $\{T_2(\mathcal{X}_i | i = \rho, 1, \ldots, n)\}$ are node disjoint subtrees of T_1 and T_2, respectively.

Acyclic agreement forests

A **maximal acyclic** agreement forest, denoted by *MAAF*, is any acyclic agreement forest of **minimal** size.

Using MAAFs to construct hybridization networks

Using MAAFs to construct hybridization networks

Compute acyclic ordering.

Using MAAFs to construct hybridization networks

Using MAAFs to construct hybridization networks

Using MAAFs to construct hybridization networks

Using MAAFs to construct hybridization networks

Using MAAFs to construct hybridization networks

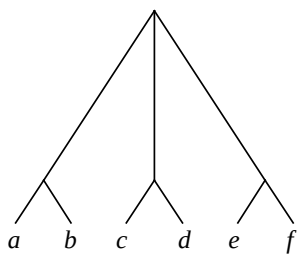
Software

1. HybridNet : `http://www.cs.cityu.edu.hk/~lwang/software/Hn/treeComp.html`
2. UltraNet: `http://rnc.r.dendai.ac.jp/ultraNet.html`
3. HybridInterleave: `http://www.math.canterbury.ac.nz/~c.semple/software.shtml`
4. NonbinaryCycleKiller: the two trees are not necessarily binary `http://homepages.cwi.nl/~iersel/cyclekiller/`
5. Dendroscope: the two trees are not necessarily binary and not necessarily on the same taxon set.
6. Hybroscale: multiply trees, not necessarily binary and not necessarily on the same taxon set `www.bio.ifi.lmu.de/softwareservices/hybroscale`
7. ...

RPN from triplets

Triplets

A rooted triple is given by a bifurcating, rooted phylogenetic tree on a set of three taxa x,y,z.



$$\frac{ab}{c} \quad \frac{ab}{d} \quad \frac{ab}{e} \quad \frac{ab}{f}$$

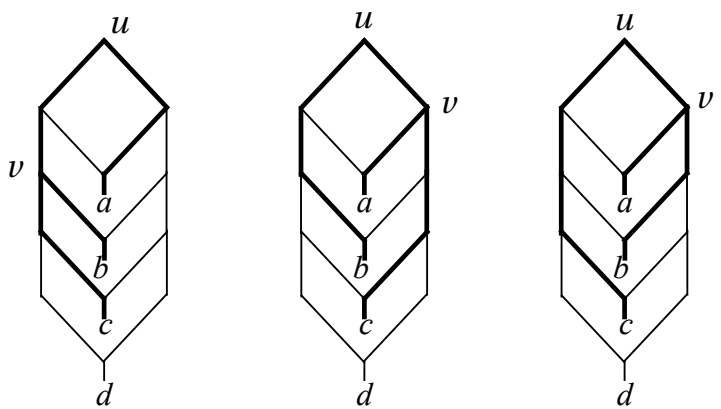
$$\frac{cd}{a} \quad \frac{cd}{b} \quad \frac{cd}{e} \quad \frac{cd}{f}$$

$$\frac{ef}{a} \quad \frac{ef}{b} \quad \frac{ef}{c} \quad \frac{ef}{d}$$

Triplets

A rooted triple $yz|x$ is said to be contained in a rooted phylogenetic network N , if there exist two nodes u and v in N such that:

1. There exists a directed path from u to the node labeled x.
2. There exists a directed path from u to v.
3. There exists a directed path from v to the node labeled y.
4. There exists a directed path from v to the node labeled z.
5. All four paths are node-disjoint (except at their end-points).



$\frac{bc}{a}$ $\frac{ac}{b}$ $\frac{ab}{c}$

Software for networks from triplets

https://leovaniersel.wordpress.com/software/

LEV1ATHAN: A Practical Algorithm for Reconstructing Level-1 Phylogenetic Networks. Combines any set of phylogenetic trees into a level-1 phylogenetic network (a galled tree) that is consistent with a large number of the triplet topologies of the input trees. Paper. **Download**.

SIMPLISTIC: Constructs level-k phylogenetic networks from triplets. This program always returns a phylogenetic network consistent with all input triplets. Partly based on the SL-k and MINPITS algorithms in this paper. **Download**.

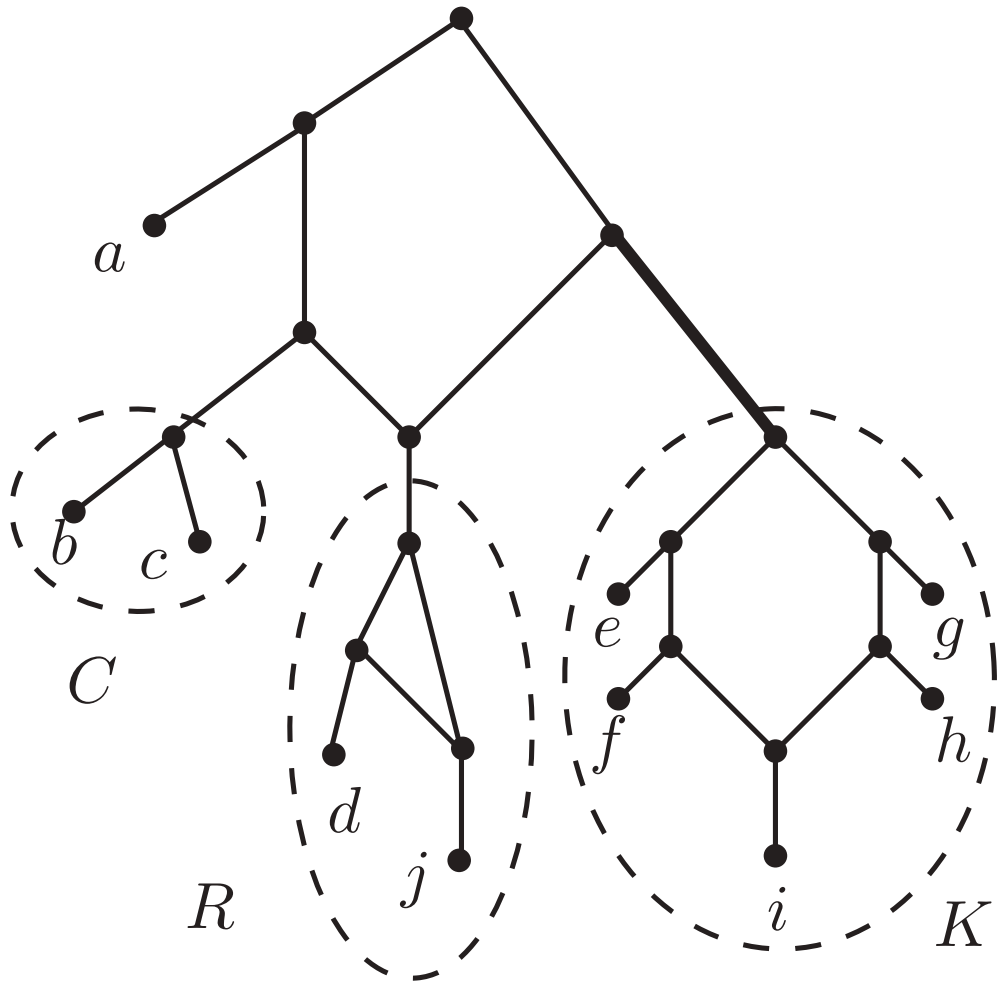
MARLON: Constructs a level-1 phylogenetic network with a minimum number of reticulations consistent with a dense set of triplets, if such a network exists. Paper. **Download**.

LEVEL2: Constructs a level-2 phylogenetic network consistent with a dense set of triplets, if such a network exists. Paper. **Download**.

Software for networks from binets and trinets

TriLoNet

https://www.uea.ac.uk/computing/TriLoNet



RPN from sequences
– ARGs –

Recombination networks

Definition (Recombination network)

Let M be a multiple alignment of binary sequences of length L, on $\mathcal{X}$. A *recombination network* N representing M is given by a bicombining rooted phylogenetic network on $\mathcal{X}$, together with two additional labellings:

1. Each node v of N is labeled by a binary sequence $\sigma(v)$ of length L.
2. Each tree edge e is labeled by a set of positions $\delta(e) \subseteq \{1, \ldots, L\}$.

These two labellings must fulfill the following compatibility conditions:

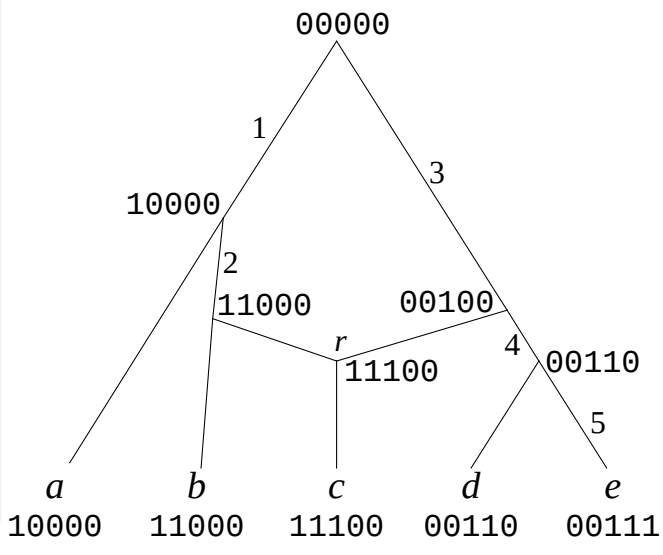
- A The sequence $\sigma(v)$ assigned to any leaf v must equal the sequence in M that is given for the taxon associated with v.
- B If r is a reticulate node (often called a recombination node in this context) with parents v and w, then the sequence $\sigma(r)$ must be obtainable from the two sequences $\sigma(v)$ and $\sigma(w)$ by a crossover.
- C If $e = (v, w)$ is a tree edge, then the set of positions at which the two sequences $\sigma(v)$ and $\sigma(w)$ differ must equal $\delta(e)$.

For computational reasons, the following condition is usually also required

- D *Any given position may mutate at most once in the network. In other words, for any given position i there exists at most one edge e with $i \in \delta(e)$.*

This condition is usually referred to as the infinite sites assumption because for sequences of infinite length it holds that the probability of the same site being hit by a mutation more than once is zero, under a uniform distribution.

a 10000
b 11000
c 11100
d 00110
e 00111



Recombination networks

Definition (Recombination network)

Let M be a multiple alignment of binary sequences of length L, on $\mathcal{X}$. A *recombination network* N representing M is given by a bicombining rooted phylogenetic network on $\mathcal{X}$, together with two additional labellings:

1. Each node v of N is labeled by a binary sequence $\sigma(v)$ of length L.
2. Each tree edge e is labeled by a set of positions $\delta(e) \subseteq \{1, \ldots, L\}$.

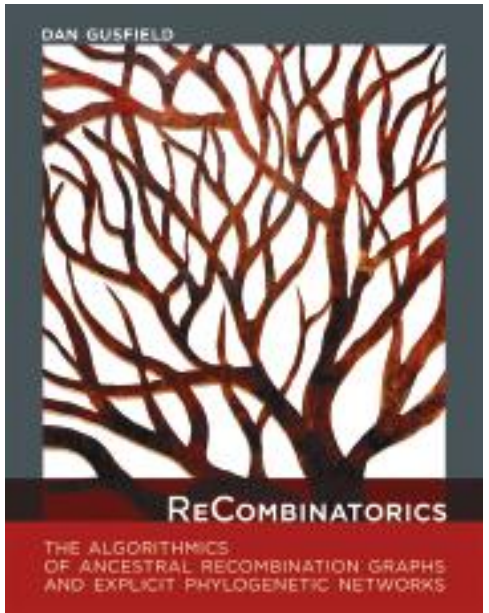
These two labellings must fulfill the following compatibility conditions:

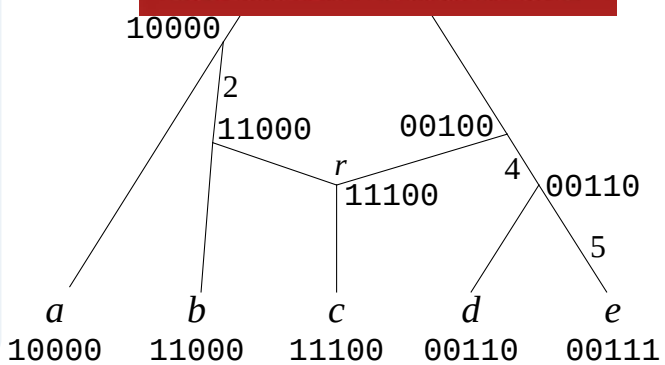
- A The sequence $\sigma(v)$ assigned to any leaf v must equal the sequence in M that is given for the taxon associated with v.
- B If r is a reticulate node (often called a recombination node in this context) with parents v and w, then the sequence $\sigma(r)$ must be obtainable from the two sequences $\sigma(v)$ and $\sigma(w)$ by a crossover.
- C If $e = (v, w)$ is a tree edge, then the set of positions at which the two sequences $\sigma(v)$ and $\sigma(w)$ differ must equal $\delta(e)$.

For computational reasons, the following condition is usually also required

- D *Any given position may mutate at most once in the network. In other words, for any given position i there exists at most one edge e with $i \in \delta(e)$.*

This condition is usually referred to as the infinite sites assumption because for sequences of infinite length it holds that the probability of the same site being hit by a mutation more than once is zero, under a uniform distribution.





Recombination networks

Definition (Recombination network)

Let M be a multiple alignment of binary sequences of length L, on $\mathcal{X}$. A *recombination network* N representing M is given by a bicombining rooted phylogenetic network on $\mathcal{X}$, together with two additional labellings:

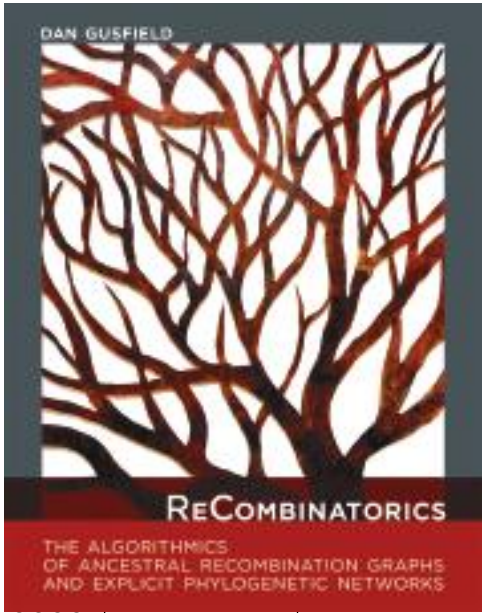
1. Each node v of N is labeled by a binary sequence $\sigma(v)$ of length L.
2. Each tree edge e is labeled by a set of positions $\delta(e) \subseteq \{1, \ldots, L\}$.

These two labellings must fulfill the following compatibility conditions:

- A The sequence $\sigma(v)$ assigned to any leaf v must equal the sequence in M that is given for the taxon associated with v.
- B If r is a reticulate node (often called a recombination node in this context) with parents v and w, then the sequence $\sigma(r)$ must be obtainable from the two sequences $\sigma(v)$ and $\sigma(w)$ by a crossover.
- C If $e = (v, w)$ is a tree edge, then the set of positions at which the two sequences $\sigma(v)$ and $\sigma(w)$ differ must equal $\delta(e)$.

For computational reasons, the following condition is usually also required

- D *Any given position may mutate at most once in the network. In other words, for any given position i there exists at most one edge e with $i \in \delta(e)$.*



the approach has to solve two NP-hard problems

Bacter: `http://tgvaughan.github.io/bacter/`, a package of BEAST 2

ARGweaver: http://mdrasmus.github.io/argweaver/doc/

RPN from sequences
– Other approaches –