Introduction to phylogenetics

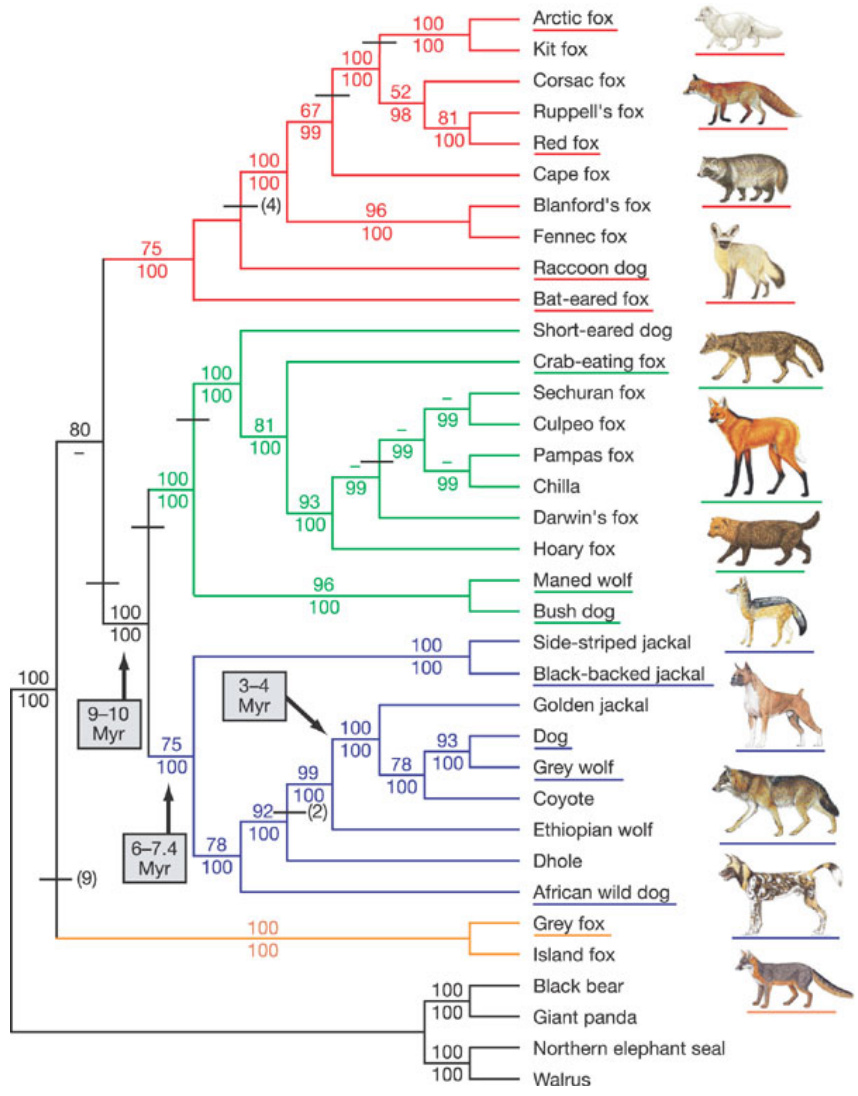
Francesc Rosselló (UIB)

ISBBC'13. Madrid, September 2013

Introduction

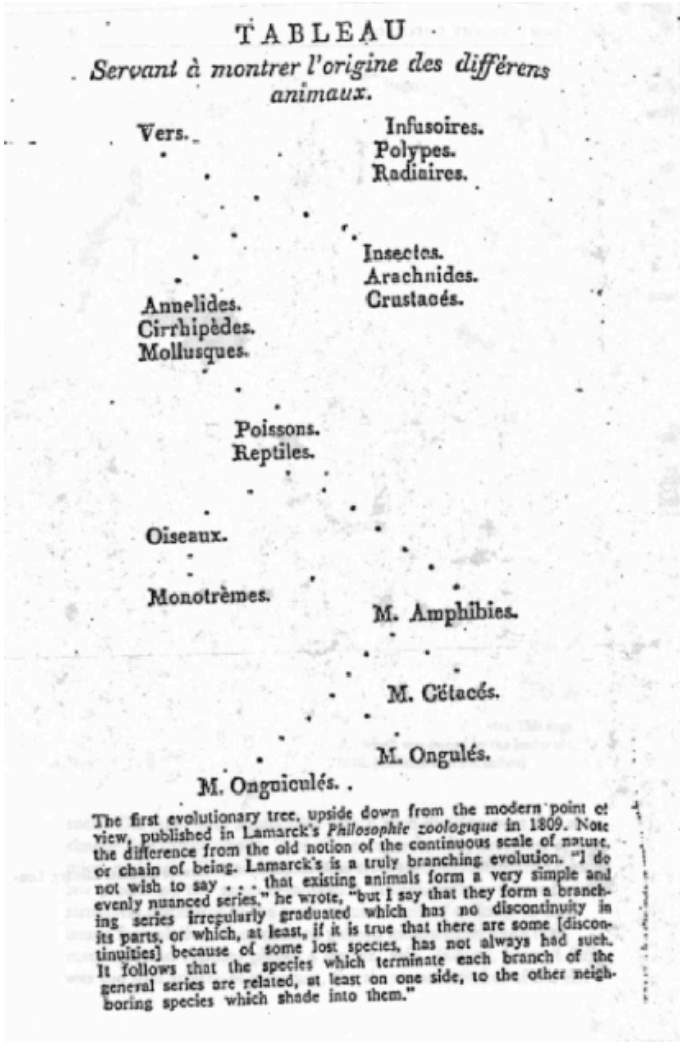
Phylogenetics is the study of the evolutionary histories (phylogenies) of groups of organisms (or proteins, genes, ...)

Phylogenies are usually (but not always) represented by means of phylogenetic trees



Source: K. Lindblad-Toh *et al*, *Nature* 438, 803-819 (2005)

Introduction

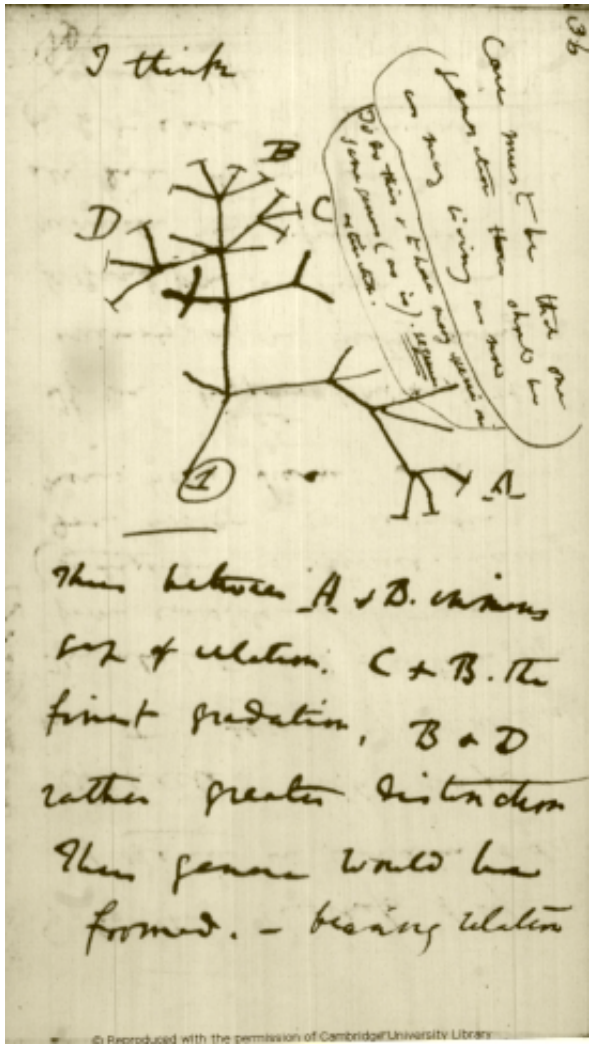


TABLEAU

Servant à montrer l'origine des différens animaux.

Vers.
Infusoires.
Polypes.
Radiaires.
Insectes.
Arachnides.
Crustacés.
Annelides.
Cirrhipèdes.
Mollusques.
Poissons.
Reptiles.
Oiseaux.
Monotrèmes.
M. Amphibies.
M. Cétacés.
M. Ongulés.
M. Onguiculés.

The first evolutionary tree, upside down from the modern point of view, published in Lamarck's *Philosophie zoologique* in 1809. Note the difference from the old notion of the continuous scale of nature, or chain of being. Lamarck's is a truly branching evolution. "I do not wish to say . . . that existing animals form a very simple and evenly nuanced series," he wrote, "but I say that they form a branching series irregularly graduated which has no discontinuity in its parts, or which, at least, if it is true that there are some [discontinuities] because of some lost species, has not always had such. It follows that the species which terminate each branch of the general series are related, at least on one side, to the other neighboring species which shade into them."



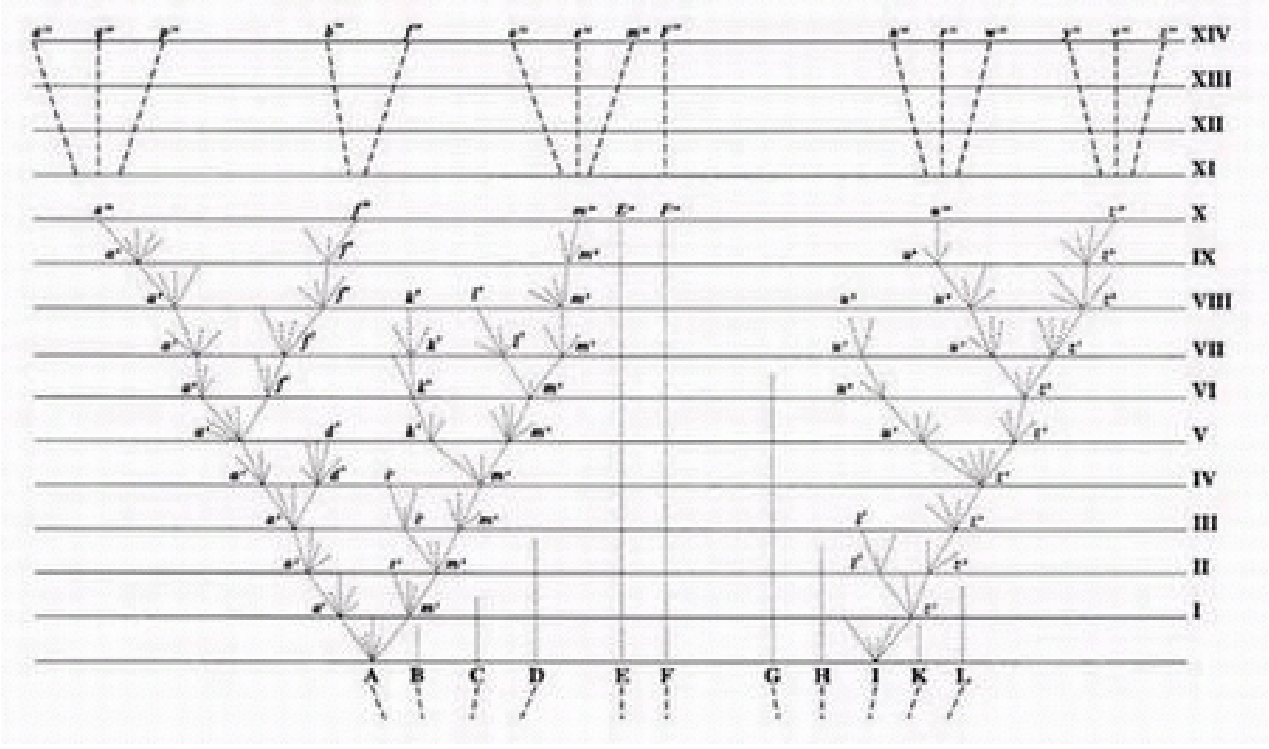
Lamarck presented an evolutionary tree of animals in 1809

Darwin drew an evolutionary tree in 1837 in his notebook

Introduction

Introduction

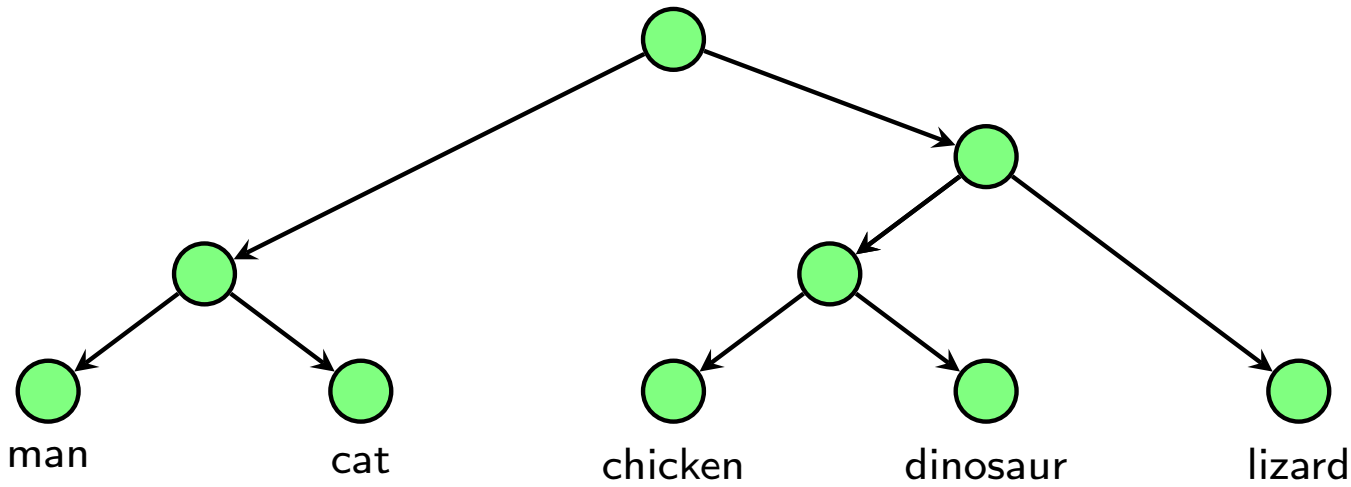
The tree of life according to *The origin of species* (1859)



Phylogenetic trees for mathematicians

Phylogenetic trees

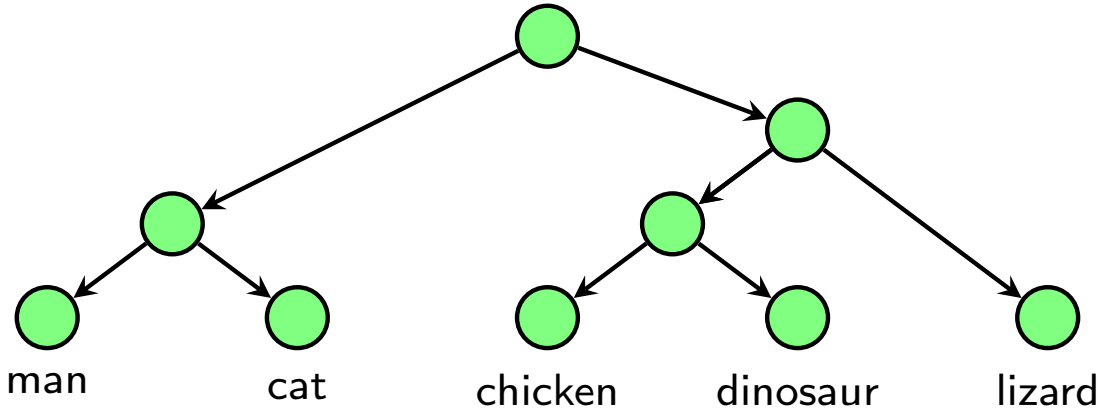
A phylogenetic tree on a set S (of OTU, *Operational Taxonomic Units*: species, organisms, proteins, genes, ...) is a rooted tree without elementary nodes and with its leaves bijectively labeled in S



Phylogenetic trees

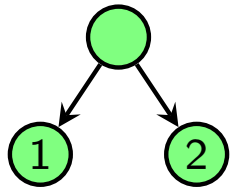
A phylogenetic tree is a description of a (hypothetical) evolutionary history of a set of OTU:

- The leaves represent the OTU under study
- The root represents their last common ancestor
- The internal nodes represent ancestors of the OTU under study that are descendants of the root
- The edges represent the direct descendance
- Only speciation events given by mutations are taken into account: every species has only one parent

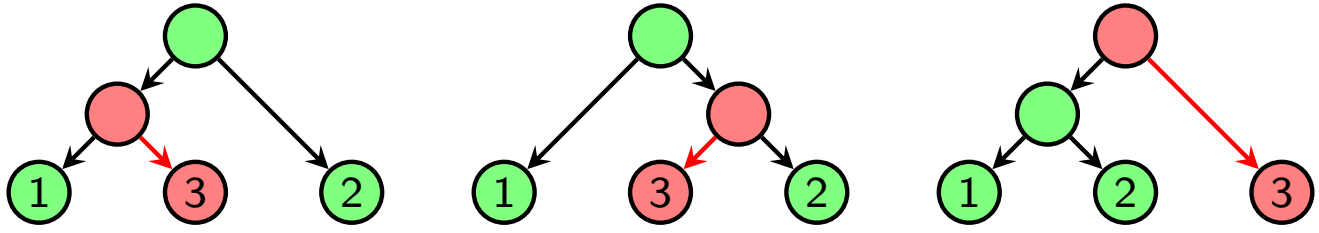


Counting binary trees

2 leaves: 1 tree

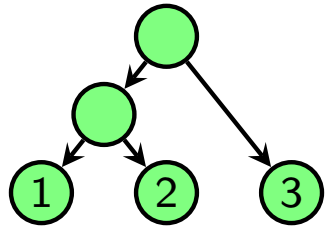


3 leaves: 3 trees

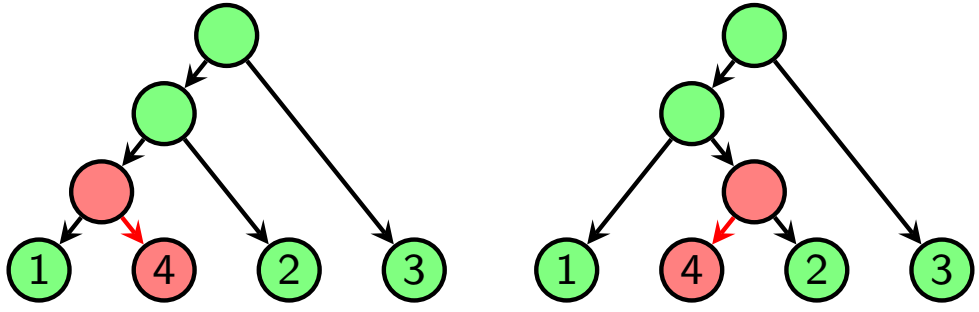


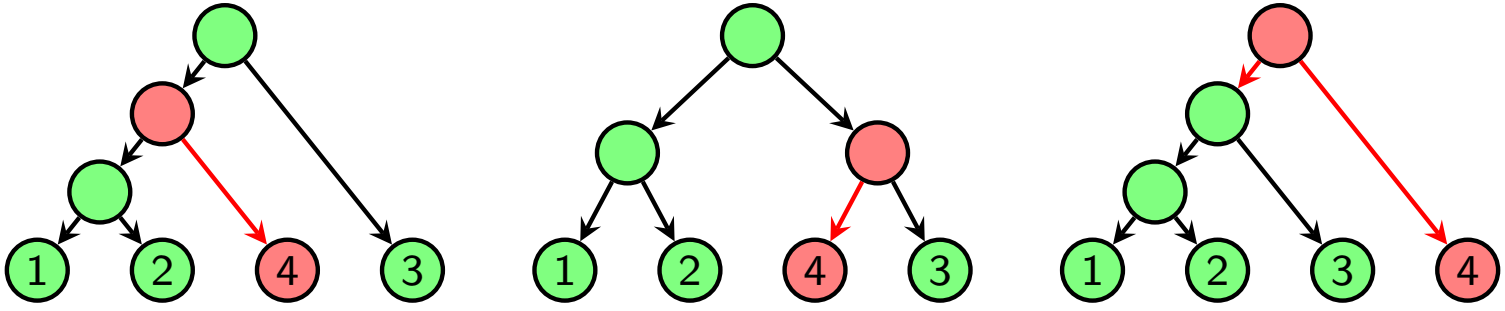
Counting binary trees

4 leaves: 15, because for each tree with 3 leaves



we can perform





Counting binary trees

Theorem

The number of binary (rooted) phylogenetic trees with n leaves is

$$(2n-3)!! := (2n-3)(2n-5)(2n-7)\cdots 5\cdot 3\cdot 1$$

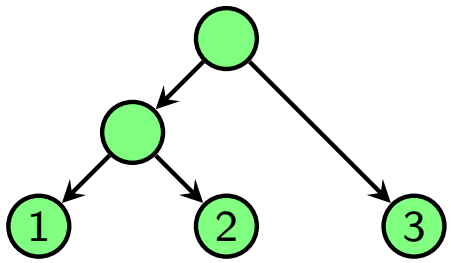
$(2\cdot 10-3)!! = 34\,459\,425$

$(2\cdot 20-3)!! \sim 8.2\cdot 10^{21}$

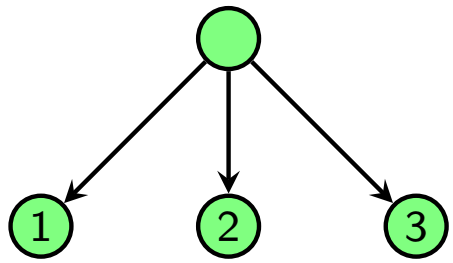
$(2\cdot 53-3)!! \sim 2.67\cdot 10^{82}$

Counting trees: the general case

We cannot use the previous argument to count all phylogenetic trees with n leaves, because the number of places (nodes or in the interior of edges) where we can add the new leaf n varies from tree to tree



7 places



5 places

Counting trees: the general case

$T_{n,m}$: phylogenetic trees with n leaves and m internal nodes

T_n: phylogenetic trees with n leaves $|T_n| = \sum_{m=1}^{n-1} |T_{n,m}|$

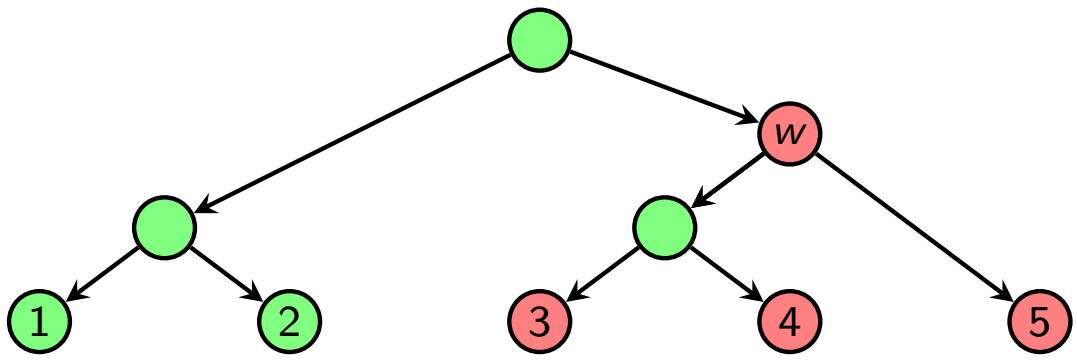
Theorem

$$|T_{n,m}| = \begin{cases} m|T_{n-1,m}| + (n+m-2)|T_{n-1,m-1}| & \text{if } m > 1 \\ 1 & \text{if } m = 1 \end{cases}$$

Closed formulas for $|T_{n,m}|$ or $|T_n|$ are not known, only recurrences and generating functions

Clusters

The cluster $C(v)$ of a node v is the set of the labels of its descendant leaves



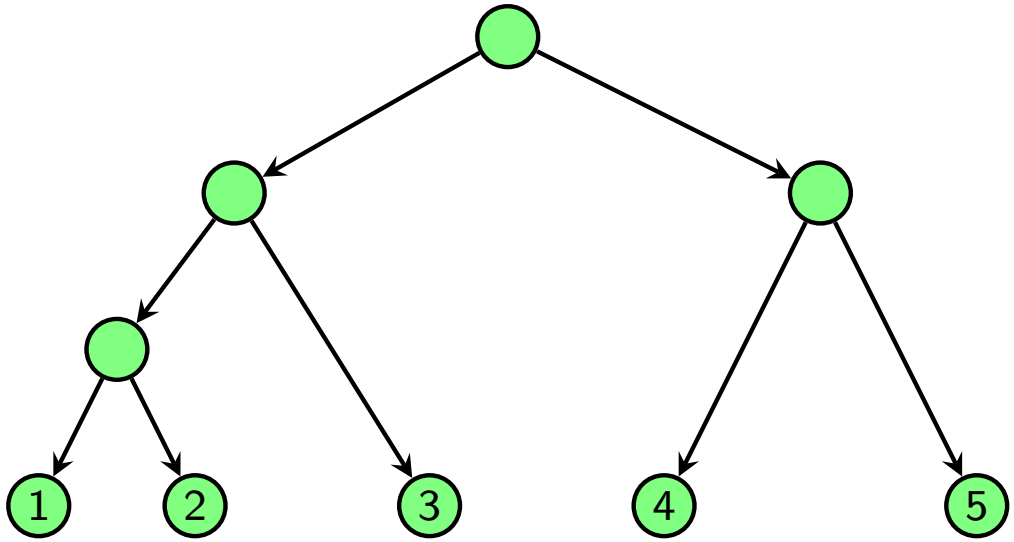
$$C(w) = \{3, 4, 5\}$$

In 'aulde phylogenetic', cluster=clade

Clusters

The family of clusters displayed by $T = (V, E)$ is

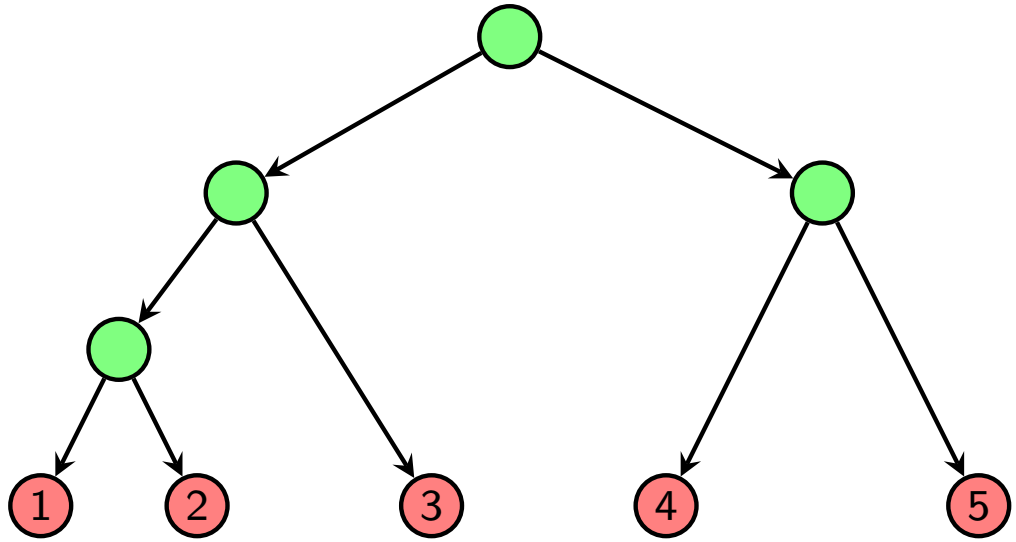
$$\mathcal{C}(T) = \{C(v) \mid v \in V\}$$



Clusters

The family of clusters displayed by $T = (V, E)$ is

$$\mathcal{C}(T) = \{C(e) \mid e \in E\}$$

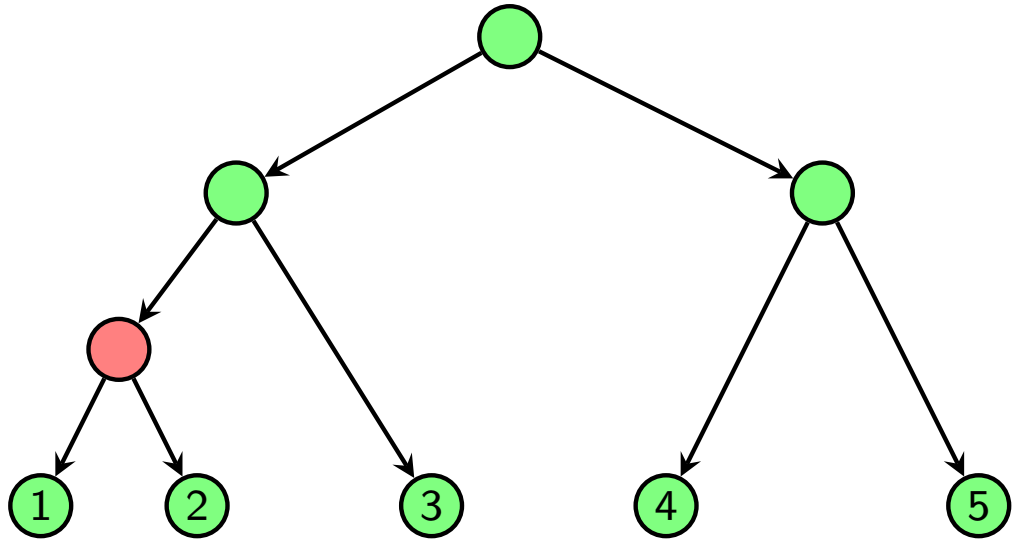


$$C(T) = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}$$

Clusters

The family of clusters displayed by $T = (V, E)$ is

$$\mathcal{C}(T) = \{C(e) \mid e \in E\}$$

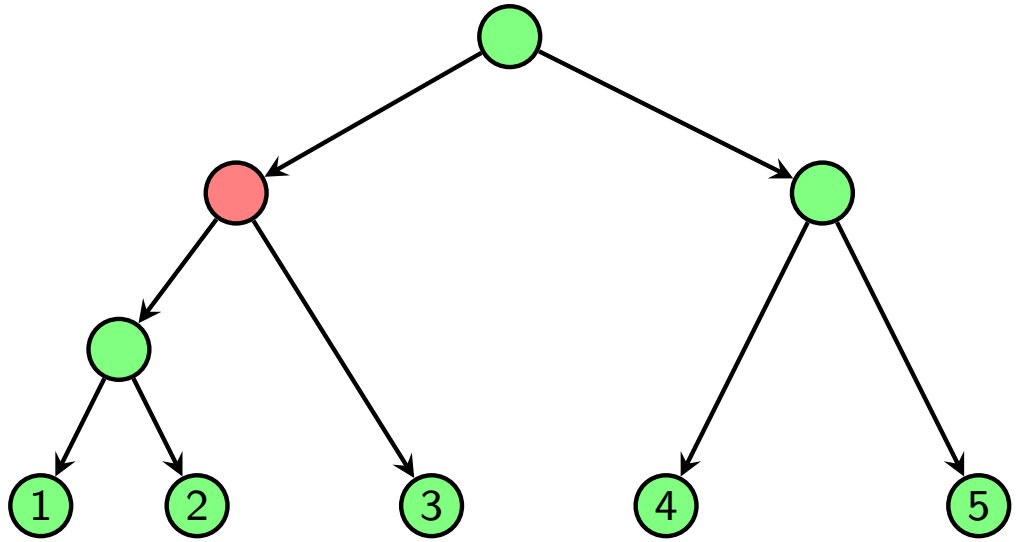


$$C(T) = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{1, 2\}$$

Clusters

The family of clusters displayed by $T = (V, E)$ is

$$\mathcal{C}(T) = \{C(e) \mid e \in E\}$$

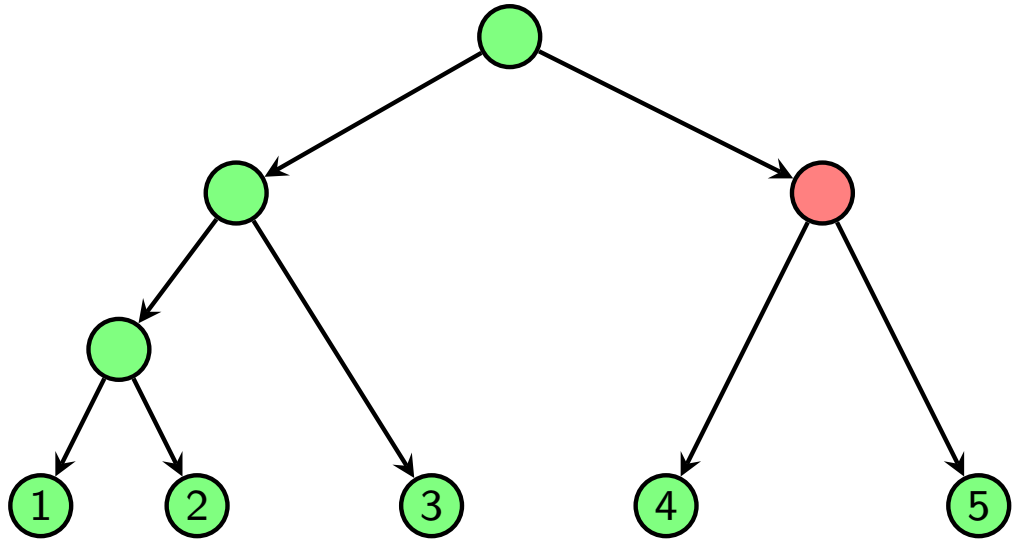


$$C(T) = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{1, 2\}, \{1, 2, 3\}$$

Clusters

The family of clusters displayed by $T = (V, E)$ is

$$\mathcal{C}(T) = \{C(e) \mid e \in E\}$$

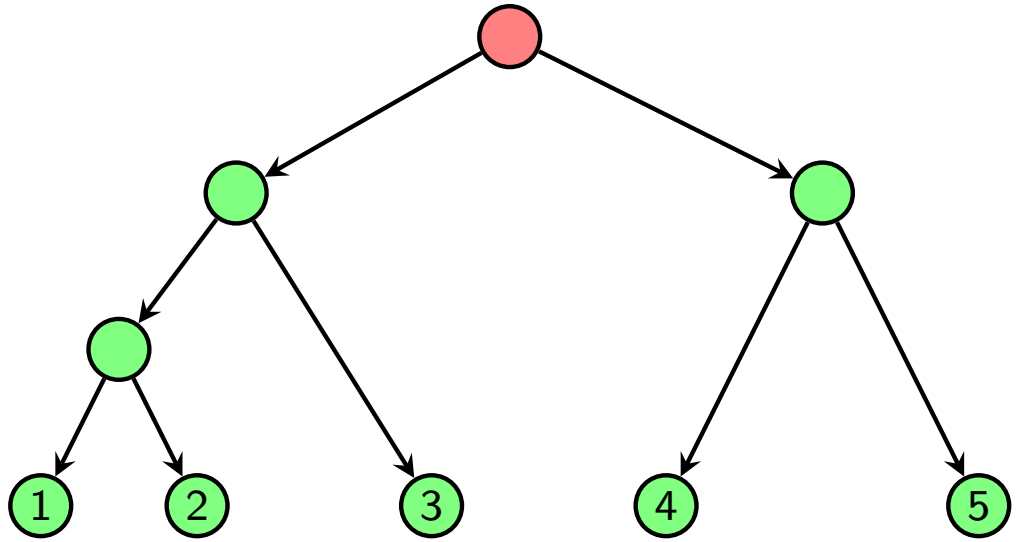


$$C(T) = \big\{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{1,2\}, \{1,2,3\}, \{4,5\}$$

Clusters

The family of clusters displayed by $T = (V, E)$ is

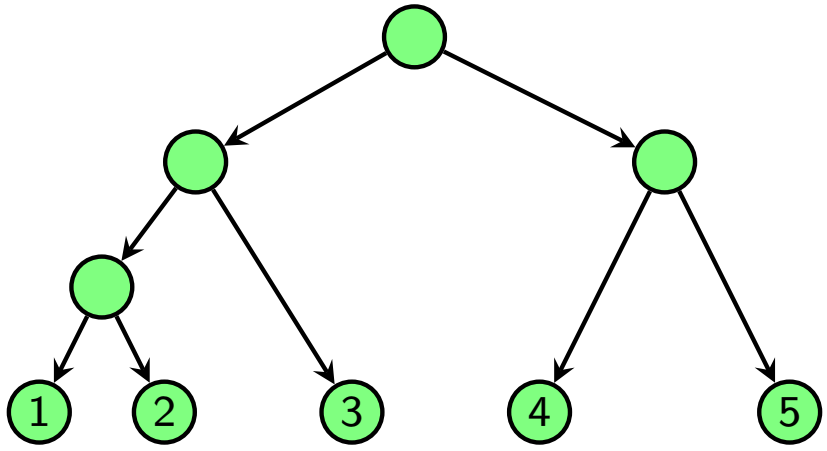
$$\mathcal{C}(T) = \{C(e) \mid e \in E\}$$



$C(T) =$

$$\big\{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{1,2\}, \{1,2,3\}, \{4,5\}, \{1,2,3,4,5\}\big\}$$

Clusters



- $v \rightsquigarrow w$ if, and only if, $C(w) \subseteq C(v)$
- Each pair $C(v)$, $C(w)$ are compatible: If $C(v) \cap C(w) \neq \emptyset$, then $C(v) \subseteq C(w)$ or $C(v) \subseteq C(w)$

Compatible clusters

Family of clusters of S: subset of $\mathcal{P}(S)$ containing S and all singletons

A family of clusters $\mathcal{C}$ is compatible if its members are pairwise compatible

Theorem

$\mathcal{C} = \mathcal{C}(T)$ for some phylogenetic tree T over S iff $\mathcal{C}$ is a compatible family of clusters of S

Proof: $\Leftarrow$) Draw the Hasse diagram of $(\mathcal{C}, \subseteq)$, root it at S, and label each $\{a\}$ with a

Trees from compatible clusters

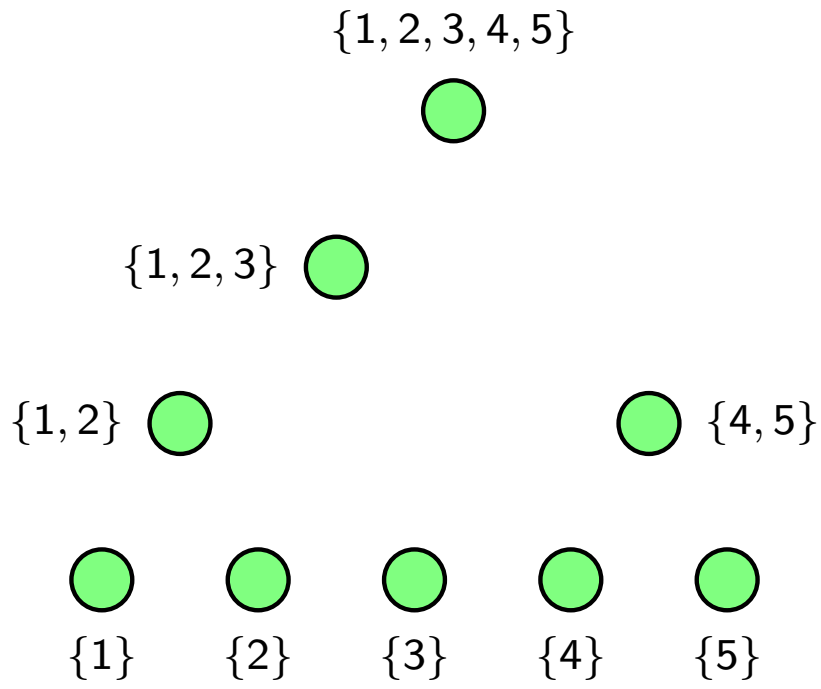
Example: $S = \{1, 2, 3, 4, 5\}$

$$\mathcal{C} = \big\{\{1, 2, 3\}, \{4, 5\}, \{1, 2\}, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{1, 2, 3, 4, 5\}\big\}$$

Trees from compatible clusters

Example: $S = \{1, 2, 3, 4, 5\}$

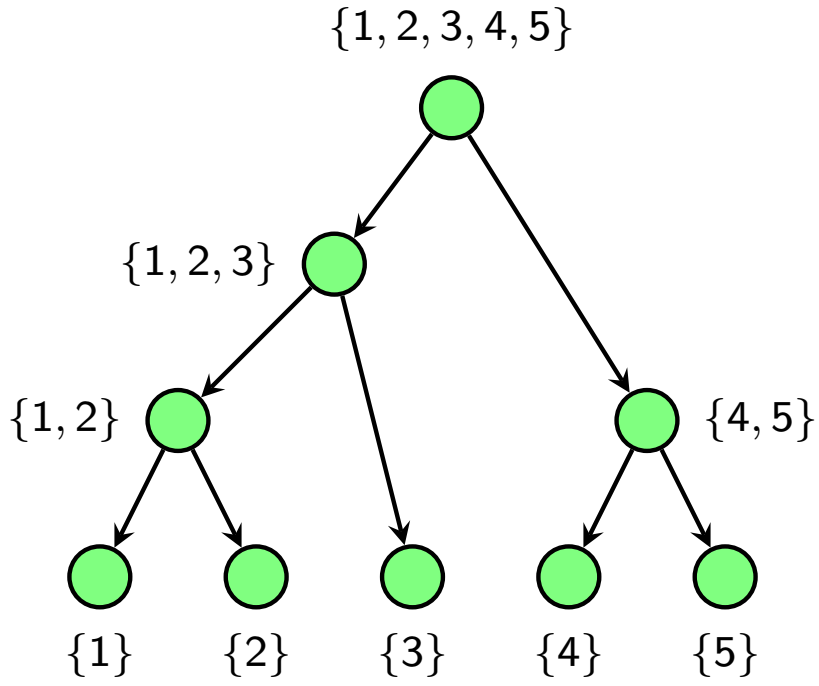
$\mathcal{C} = \big\{\{1, 2, 3\}, \{4, 5\}, \{1, 2\}, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{1, 2, 3, 4, 5\}\big\}$



Trees from compatible clusters

Example: $S = \{1, 2, 3, 4, 5\}$

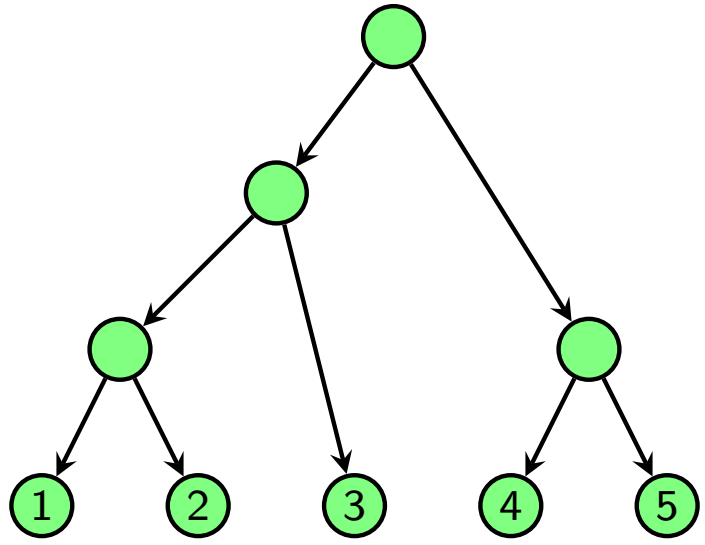
$\mathcal{C} = \big\{\{1, 2, 3\}, \{4, 5\}, \{1, 2\}, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{1, 2, 3, 4, 5\}\big\}$



Trees from compatible clusters

Example: $S = \{1, 2, 3, 4, 5\}$

$\mathcal{C} = \big\{\{1, 2, 3\}, \{4, 5\}, \{1, 2\}, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{1, 2, 3, 4, 5\}\big\}$

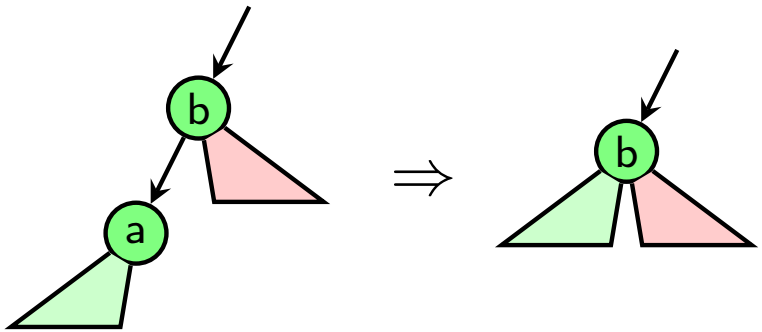


Clusters

Theorem

Let T, T' be phylogenetic trees over S. If $\mathcal{C}(T) = \mathcal{C}(T')$, then $T \cong T'$.

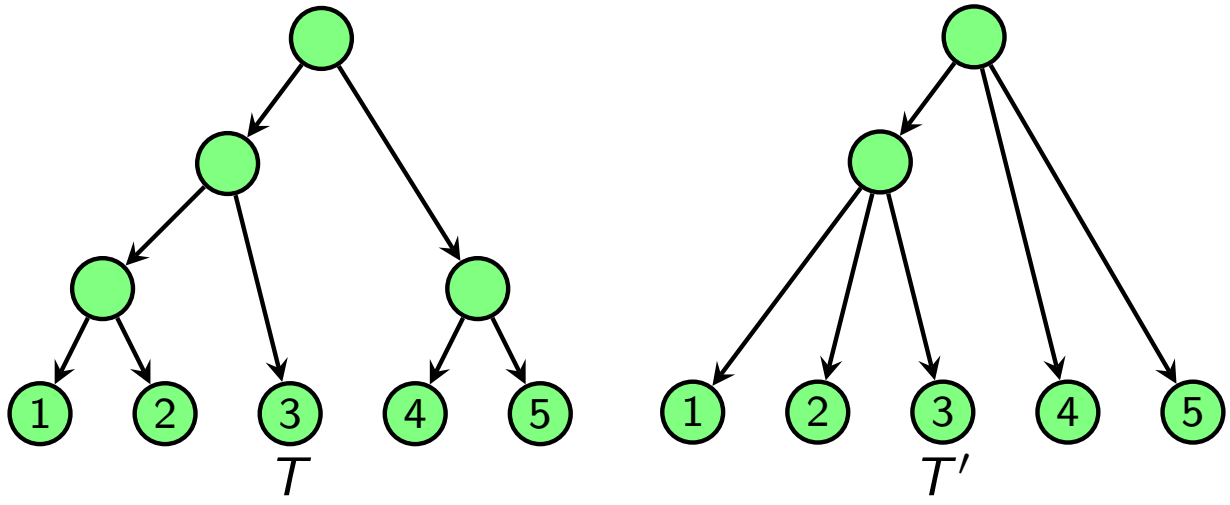
$T' \leqslant T$ (T refines T') when T' is obtained from T by contracting edges



Theorem

Let T, T' be phylogenetic trees over S. Then, $\mathcal{C}(T') \subseteq \mathcal{C}(T)$ iff $T' \leqslant T$.

Clusters



$$\mathcal{C}(T) = \{\{1,2,3\},\{4,5\},\{1,2\},\{1\},\{2\},\{3\},\{4\},\{5\},\{1,2,3,4,5\}\}$$

$$\mathcal{C}(T') = \{\{1,2,3\},\{1\},\{2\},\{3\},\{4\},\{5\},\{1,2,3,4,5\}\}$$

Incompatible clusters

What to do when $\mathcal{C}$ is incompatible?

- Remove a minimal subset of taxa such that $\mathcal{C}$ becomes compatible: NP-hard (M. Steel, A. Hamel, *Appl. Math. Lett.* 9 (1996), 55–60)
- Remove a minimal subset of clusters such that $\mathcal{C}$ becomes compatible: NP-complete (finding maximal cliques)
- Forget about trees, look for multi-labeled trees or phylogenetic networks

Incompatible clusters

What to do when $\mathcal{C}$ is incompatible?

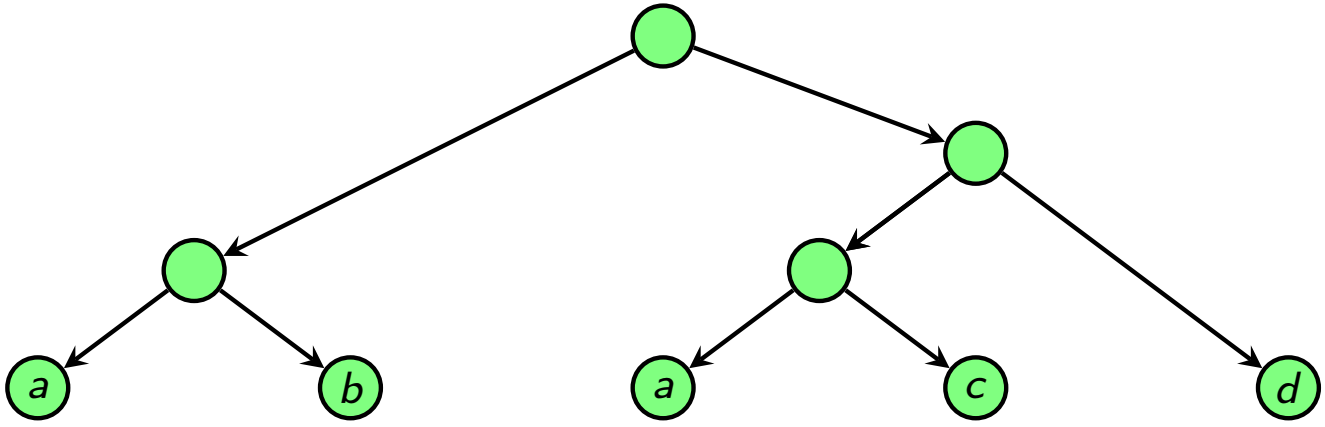
- Remove a minimal subset of taxa such that $\mathcal{C}$ becomes compatible: NP-hard (M. Steel, A. Hamel, *Appl. Math. Lett.* 9 (1996), 55–60)
- Remove a minimal subset of clusters such that $\mathcal{C}$ becomes compatible: NP-complete (finding maximal cliques)
- Forget about trees, look for multi-labeled trees or phylogenetic networks

Where do incompatible clusters come?

- Taxonomic information from different sources
- Trying to reconcile a family of trees

Mul-trees

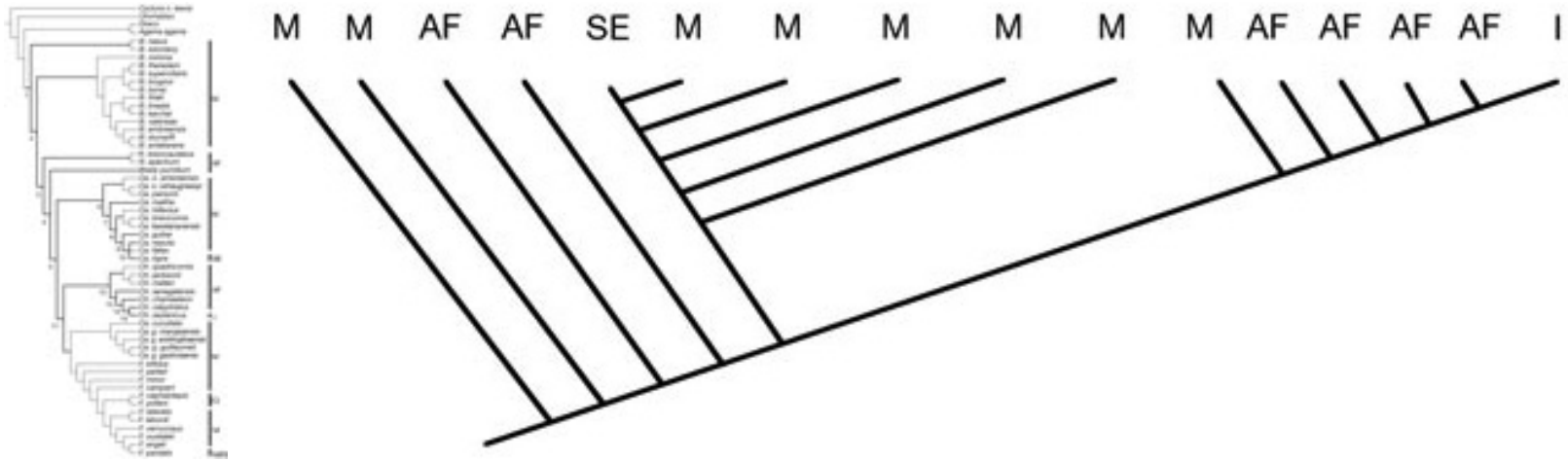
A multi-labeled tree (mul-tree) on a set S is as a phylogenetic tree, but with possibly repeated leaves



$$\mathcal{C}(T) = \{\{a, b\}, \{a, c\}, \{a, c, d\}, \dots\}$$

Mul-trees

Example: Area cladograms, phylogenetic trees where species at the leaves are replaced by regions

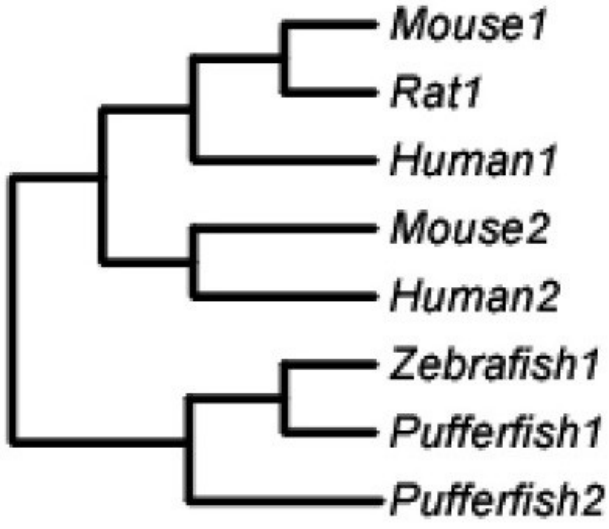


AF, Africa; AN, Antarctica; AU, Australia; M, Madagascar; SA, South America; SE, the Seychelles; I, India and Sri Lanka

Source: C. Raxworthy *et al.* *Nature* 415 (2002), 784–787

Mul-trees

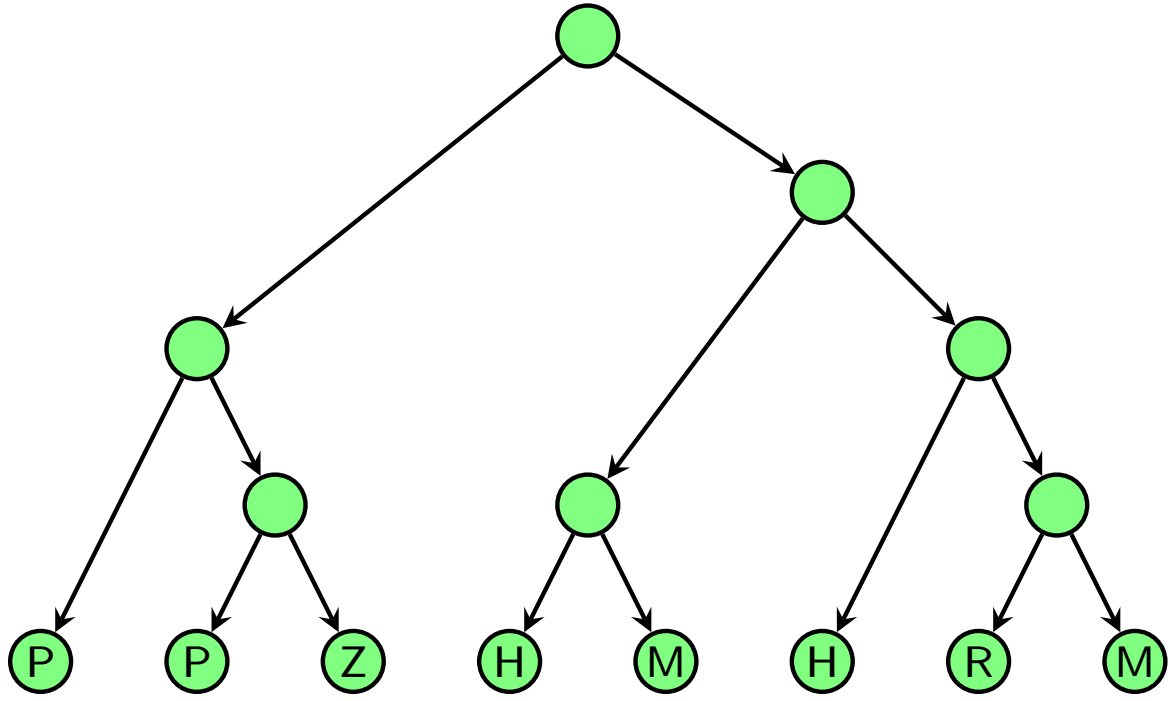
Example: Gene trees, which describe how genes have evolved through duplications and mutations



Database: `http://www.ebi.ac.uk/biomodels-main/`

Mul-trees

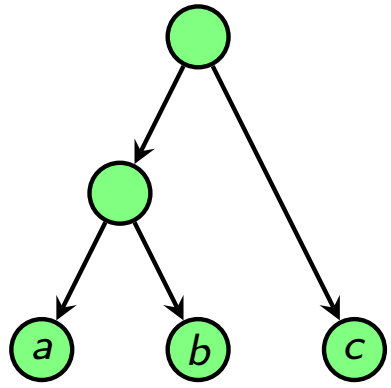
Example: Gene trees, which describe how genes have evolved through duplications and mutations



Incompatible clusters and mul-trees

- Every family of clusters $\mathcal{C}$ is displayed by (possibly) many mul-trees, even if we take into account multiplicities
- Deciding whether there exists some mul-tree displaying $\mathcal{C}$ with at most $k \geqslant 1$ duplications (repetitions of leaves) is NP-hard
- Finding a mul-tree displaying $\mathcal{C}$ with the least number of duplications is NP-hard. A few algorithms have been proposed recently.
- Finding a minimal mul-tree displaying $\mathcal{C}$ is an open problem

Triples

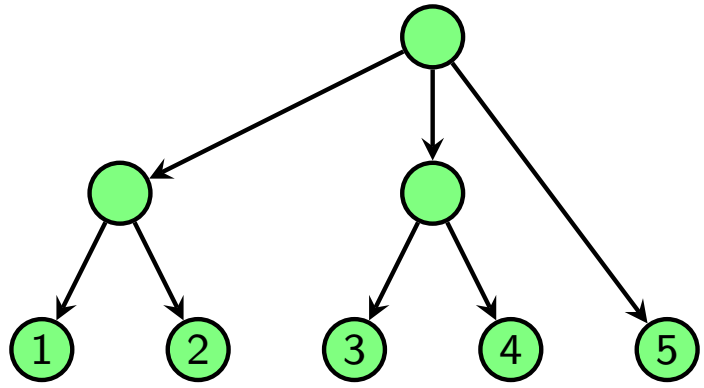


$ab|c$

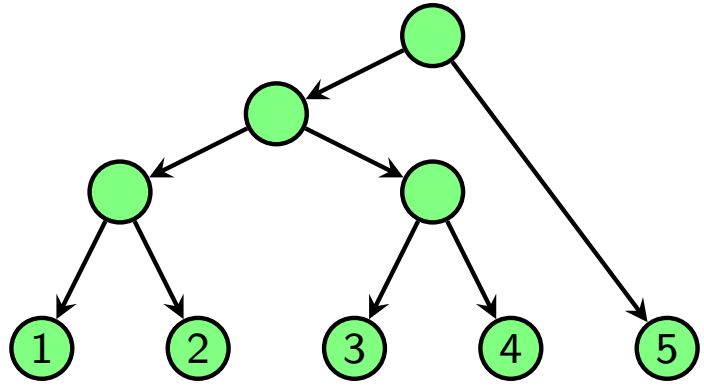
T contains $ab|c$ when $LCA(a,b) < LCA(a,c) = LCA(b,c)$

$\Gamma(T)$ = set of all triples contained in T

Triples



$$\Gamma(T) = \{12|3, 12|4, 12|5, 34|1, 34|2, 34|5\}$$



$$\Gamma(T) = \{12|3, 12|4, 12|5, 34|1, 34|2, 34|5, 13|5, 14|5, 23|5, 24|5\}$$

Triples

Proposition

The information in $\mathcal{C}(T)$ and $\Gamma(T)$ are equivalent.

- $ab|c \in \Gamma(T)$ iff $\exists C \in \mathcal{C}(T)$ such that $a, b \in C$ and $c \notin C$
- $C \in \mathcal{C}(T)$ iff $ab|c \in \Gamma(T)$ for every $a, b \in C$ and $c \notin C$

Corollary

$T' \leqslant T$ iff $\Gamma(T') \subseteq \Gamma(T)$. In particular, $\Gamma(T)$ singles out T among all trees over S.

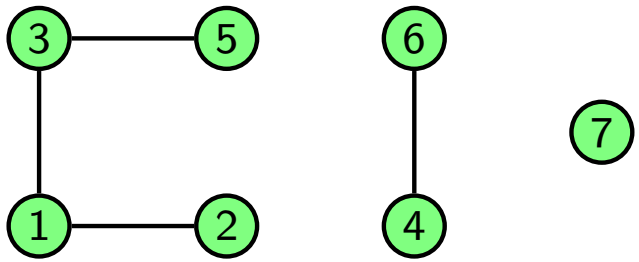
Trees from triples: Aho's algorithm

Given a set Γ of triples over S, $AG(\Gamma) = (V, E)$, where:

- $V = S$
- $\{a, b\} \in E$ iff there exists some $ab|c$ in Γ

Example:

$$S = \{1, 2, 3, 4, 5, 6, 7\}, \Gamma = \{12|3, 12|5, 13|4, 35|4, 46|3\}$$



$AG(\Gamma)$

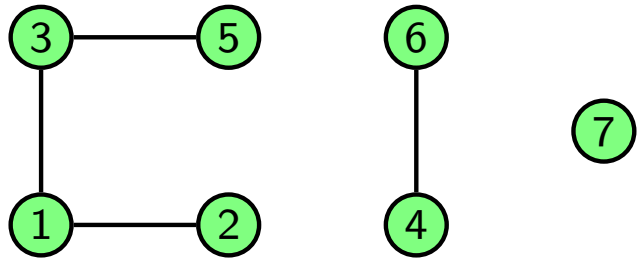
Trees from triples: Aho's algorithm

Given a set Γ of triples over S, let's compute a tree A_Γ over S such that $\Gamma = \Gamma(A_\Gamma)$, if some exist:

- If $|S| \leqslant 2$, A_Γ is the tree with the elements of S as leaves
- If $|S| \geqslant 3$, compute $AG(\Gamma)$
  - If $AG(\Gamma)$ is connected, output *Fail*
  - If $AG(\Gamma)$ is not connected, for each node set U of a connected component, recursively apply the algorithm to $\Gamma|_U$
  - Create a root node r and make it the parent of the roots of all $A_{\Gamma|_U}$

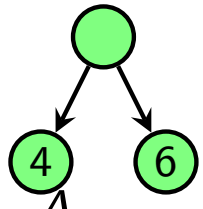
Trees from triples: Aho's algorithm

$\Gamma = \{12|3, 12|5, 13|4, 35|4, 46|3\}$



$AG(\Gamma)$

$U_1 = \{1, 2, 3, 5\}, U_2 = \{4, 6\}, U_3 = \{7\}$

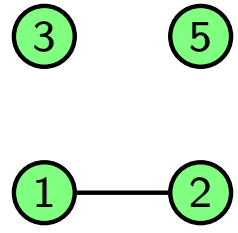


$A_{\Gamma|U_2}$

$A_{\Gamma|U_3}$

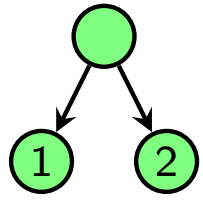
Trees from triples: Aho's algorithm

$\Gamma|_{U_1} = \{12|3, 12|5\}$



$AG(\Gamma|_{U_1})$

$U_{1,1} = \{1, 2\}, U_{1,2} = \{3\}, U_{1,3} = \{5\}$

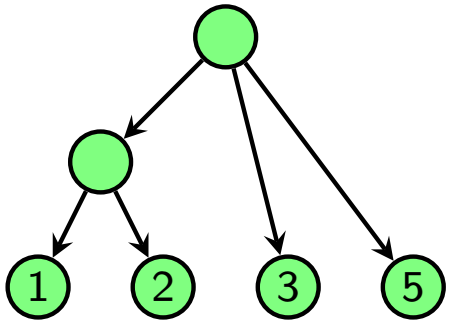


$A_{\Gamma|U_{1,1}}$

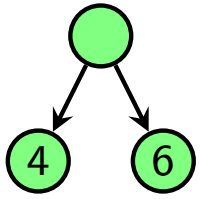
$A_{\Gamma|U_{1,2}}$

$A_{\Gamma|U_{1,3}}$

Trees from triples: Aho's algorithm

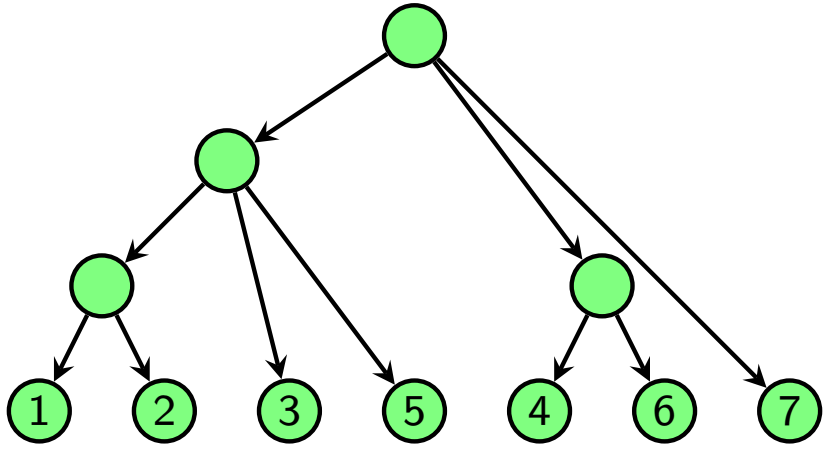


$A_{\Gamma|U_1}$



$A_{\Gamma|U_2}$

$A_{\Gamma|U_3}$

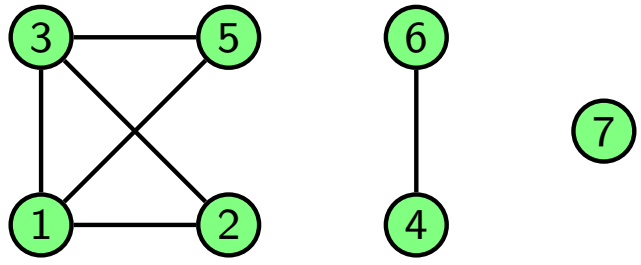


A_Γ

$$\Gamma = \{12|3, 12|5, 13|4, 35|4, 46|3\}$$

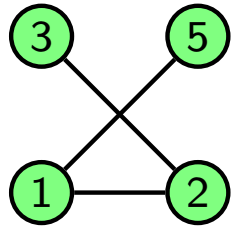
Trees from triples: Aho's algorithm

$\Gamma = \{12|3, 12|5, 13|4, 35|4, 46|3, 23|5, 15|3\}$



$AG(\Gamma)$

$U_1 = \{1, 2, 3, 5\}$, $\Gamma|U_1 = \{12|3, 12|5, 23|5, 15|3\}$

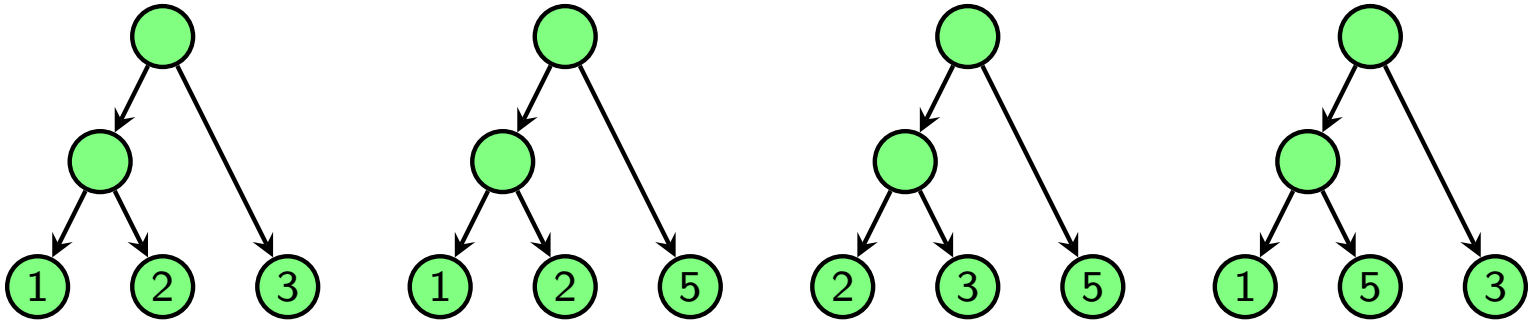


$AG(\Gamma|U_1)$

Fail

Example

We found an obstruction:



$$LCA(1,5) < LCA(3,5) = LCA(2,5) = LCA(1,5)$$

Trees from triples: Aho's algorithm

If Γ is compatible, A_Γ is minimal containing Γ with this property (if we contract any edge, the resulting tree does'nt contain Γ)

If Γ is incompatible, the Aho algorithm reports *fail*

Incompatible triples

What to do when Γ is incompatible?

- Remove a minimal subset of triples such that Γ becomes compatible: NP-hard (D. Bryant, PhD Thesis (1997))
- Usual heuristic: Determine a small cut set of edges in $AG(\Gamma)$, remove the corresponding triples, and continue
- Forget about trees, look for multi-labeled trees or phylogenetic networks

Building phylogenetic trees

The reconstruction problem

Problem

Given information about a set of OTU, find a phylogenetic tree representing an evolutionary history that best explains them

There are hundreds of algorithms and programs 'solving' this problem in its different versions

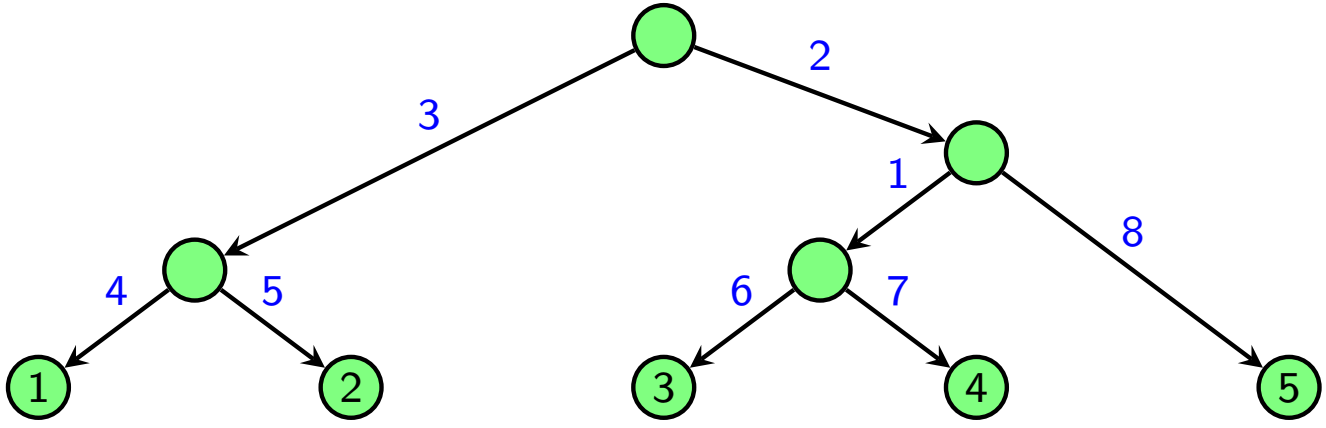
A complete collection:
http://evolution.genetics.washington.edu/phylip/software.html

Daily new contributions to the field, anyone is welcome

From distances

Additive tree: weighted phylogenetic tree, whose weights represent a quantitative measure of evolutionary divergence (e.g., number of mutations, evolutionary time, etc.)

An additive tree defines an additive distance on the set of OTU



d_T	1	2	3	4	5
1	0	9	16	17	17
2		0	17	18	18
3			0	13	15
4				0	16
5					0

From distances

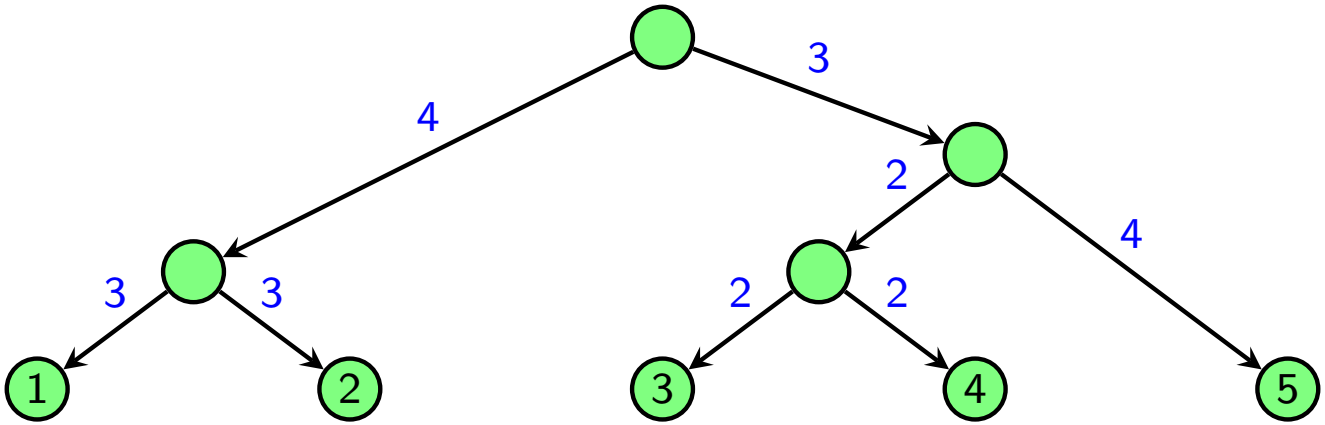
Problem: Given a matrix of distances between OTU, find an additive tree that defines an additive distance closest to the input distance

NP-hard in most cases

Several popular heuristic "solutions" (yielding rooted or unrooted trees)

From distances

Ultrametric tree: additive tree where all leaves are equidistant from the root



Models molecular clock hypothesis (L. Pauling *et al*, 1960s): The 'speed' of evolution is constant in all evolutionary histories.

UPGMA (aka simple-linkage hierarchical clustering algorithm) produces an ultrametric tree that is the closest sub-dominant solution for $\| \ \|_\infty$

From characters

Problem: Given descriptions of the OTU as vectors of characters, find a simplest tree explaining them

These vectors of characters can be:

- Discrete, usually dichotomic, properties of organisms

	Hair	Lungs	Oviparous	Milk
Dog	1	1	0	1
Frog	0	1	1	0
Chicken	0	1	1	0
Salmon	0	0	1	0

From characters

Problem: Given descriptions of the OTU as vectors of characters, find a simplest tree explaining them

These vectors of characters can be:

- Letters at aligned positions (by a multiple alignment) in biomolecular sequences

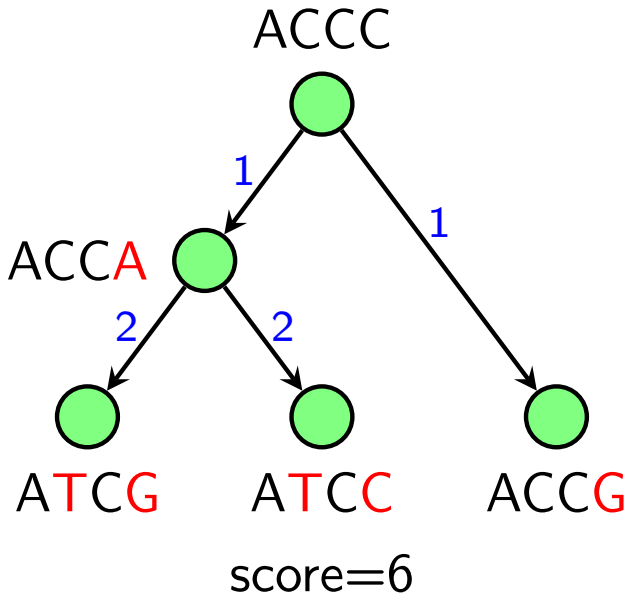
Dog	`ACTTTAACTACT`
Frog	`ACATTGACTGGT`
Chicken	`AACGTACTTACT`
Salmon	`AATTTCACTAAC`

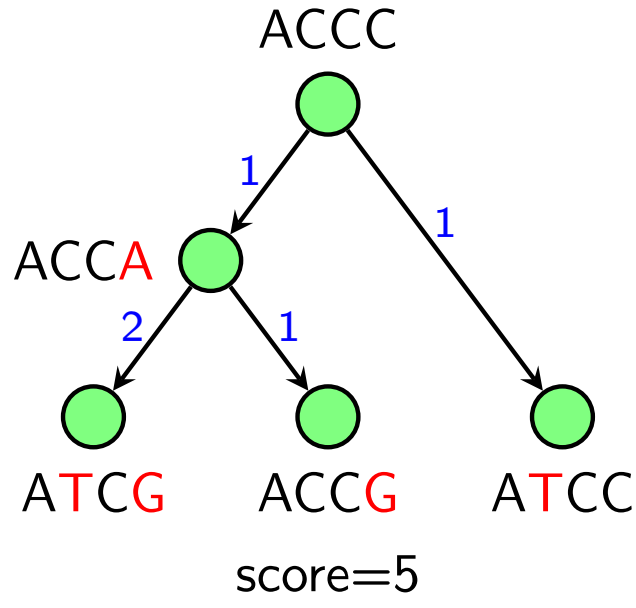
From sequences: Parsimony methods

Problem: Given biomolecular sequences, find a tree that produces them from a single sequence through minimum amount of evolution

Assigning sequences to internal nodes and weights to the mutations represented by branches, we look for the tree with smallest total weight (parsimony score)

Example: sequences ATCG, ATCC, ACCG



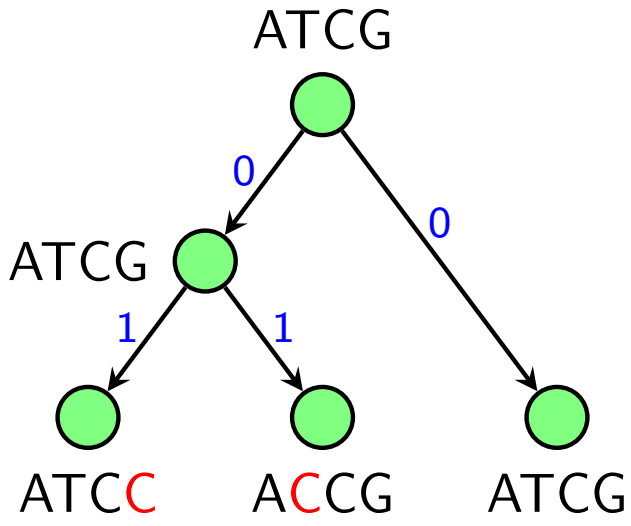


From sequences: Parsimony methods

Problem: Given biomolecular sequences, find a tree that produces them from a single sequence through minimum amount of evolution

Assigning sequences to internal nodes and weights to the mutations represented by branches, we look for the tree with smallest total weight (parsimony score)

Example: sequences ATCG, ATCC, ACCG



The most parsimonious: score=2

From sequences: Parsimony methods

Problem: Given biomolecular sequences, find a tree that produces them from a single sequence through minimum amount of evolution

Given a fixed tree topology, sequences at the leaves, and a matrix of mutation scores, the sequences at the internal nodes minimizing the score can be computed in polynomial time through dynamic programming (Sankoff algorithm,1983)

Finding the most parsimonious tree topology is NP-complete

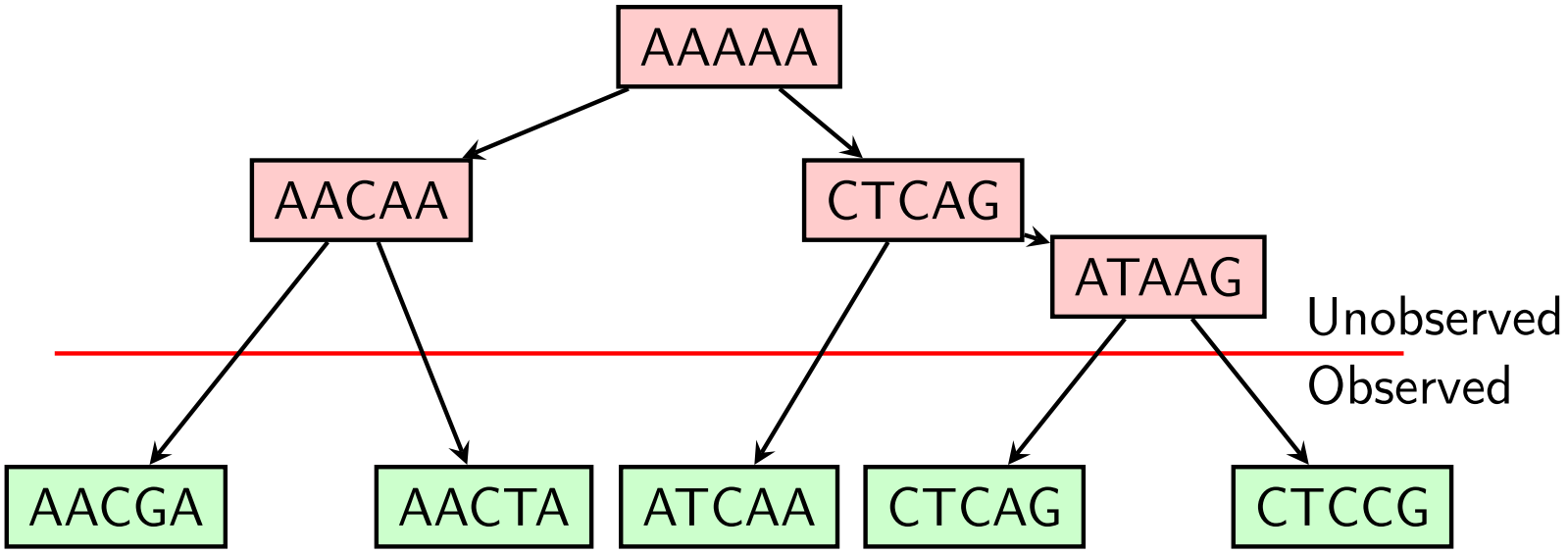
From sequences: Parsimony methods

Solutions:

- For small n ($\leqslant 10$), exhaustive search in the space of all binary trees with n leaves, computing for each one of them its optimal score
- For large n:
  - generate randomly many trees, compute their optimal score, keep the most parsimonious
  - modify them randomly through edit operations and keep the modified trees if their score is smaller
  - iterate this procedure several times
- Other heuristics ...

From sequences: Likelihood methods

A phylogenetic tree can be considered as an stochastic process: mutations are randomly applied to the sequences along the edges, and speciation events occurr randomly at the internal nodes



From sequences: Likelihood methods

Additive phylogenetic trees as (simple) Markov models:

- Nodes are labeled with DNA sequences of fixed length m
- Only substitutions occur along the evolutionary process (the length of the sequences remains constant)
- Each site(=nucleotide position) evolves independently of the others
- Evolution along edges are independent of each other
- Each edge (u, v) has a weight $t_{u,v}$ measuring the evolutionary time between the species associated with nodes u and v

From sequences: Likelihood methods

The rate of substitution $\theta_{y|x}$ of x with y measures the rate at which x changes into y per unit time

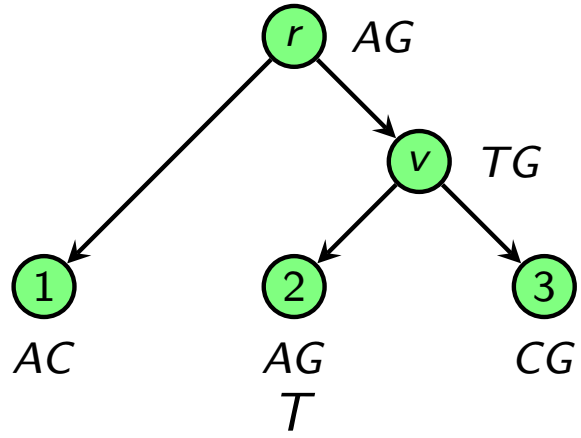
$$\theta = \begin{pmatrix} \theta_{A|A} & \theta_{C|A} & \theta_{G|A} & \theta_{T|A} \\ \theta_{A|C} & \theta_{C|C} & \theta_{G|C} & \theta_{T|C} \\ \theta_{A|G} & \theta_{C|G} & \theta_{G|G} & \theta_{T|G} \\ \theta_{A|T} & \theta_{C|T} & \theta_{G|T} & \theta_{T|T} \end{pmatrix}$$

The probability $P(y|x,t)$ that x changes into y within the time t is obtained from θ

The model of evolution (Jukes-Cantor, Kimura, ...) provides a distribution of probabilities (q_A, q_C, q_G, q_T) at the root of the tree and the probabilities $P(y|x,t)$

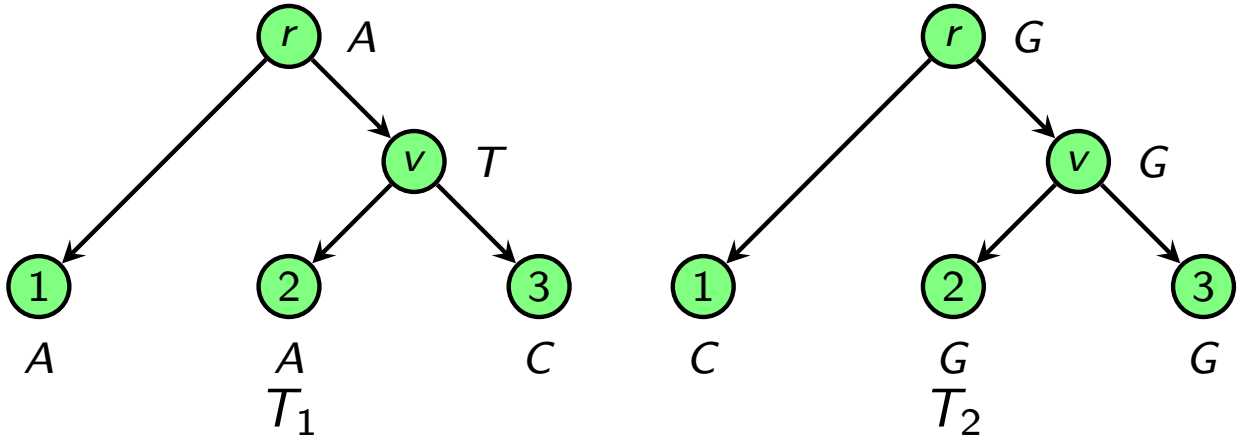
From sequences: Likelihood methods

Given a phylogenetic tree with all its nodes labeled with sequences of length m, from these parameters we can compute the probability of the tree



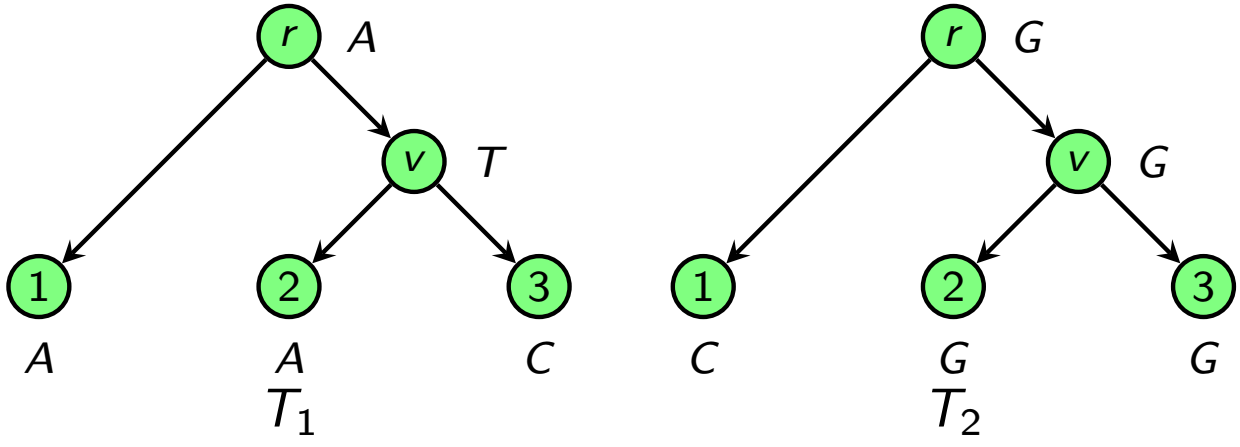
From sequences: Likelihood methods

Given a phylogenetic tree with all its nodes labeled with sequences of length m, from these parameters we can compute the probability of the tree



From sequences: Likelihood methods

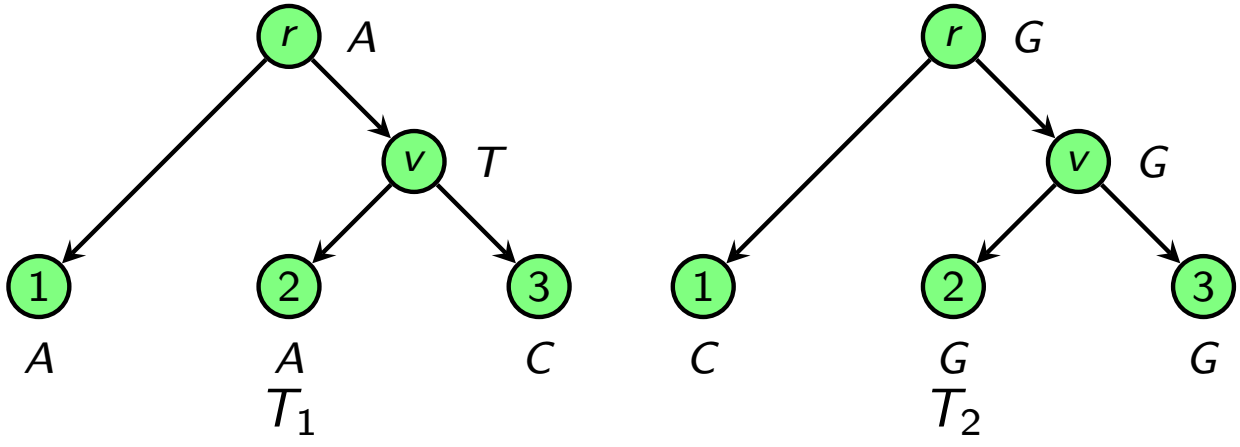
Given a phylogenetic tree with all its nodes labeled with sequences of length m, from these parameters we can compute the probability of the tree



$$P(T_1) = q_A \cdot P(A|A, t_{r,1}) \cdot P(T|A, t_{r,v}) \cdot P(A|T, t_{v,2}) \cdot P(C|T, t_{v,3})$$

From sequences: Likelihood methods

Given a phylogenetic tree with all its nodes labeled with sequences of length m, from these parameters we can compute the probability of the tree

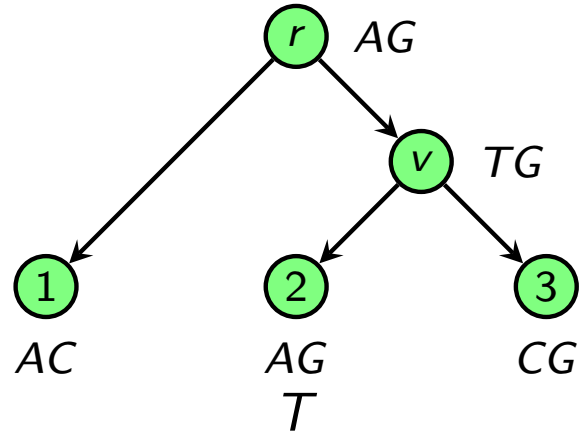


$$P(T_1) = q_A \cdot P(A|A, t_{r,1}) \cdot P(T|A, t_{r,v}) \cdot P(A|T, t_{v,2}) \cdot P(C|T, t_{v,3})$$

$$P(T_2) = q_G \cdot P(C|G, t_{r,1}) \cdot P(G|G, t_{r,v}) \cdot P(G|G, t_{v,2}) \cdot P(G|G, t_{v,3})$$

From sequences: Likelihood methods

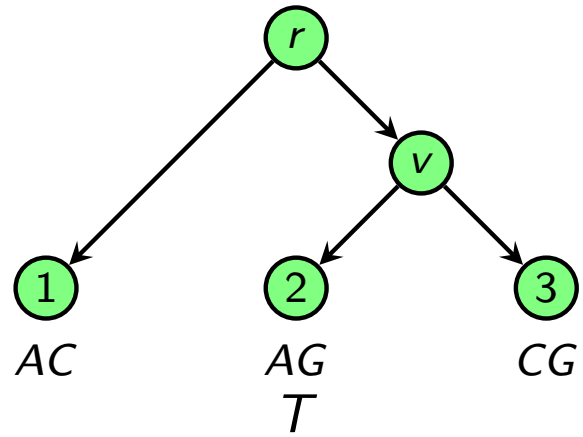
Given a phylogenetic tree with all its nodes labeled with sequences of length m, from these parameters we can compute the probability of the tree



$$P(T) = P(T_1) \cdot P(T_2) = \ldots$$

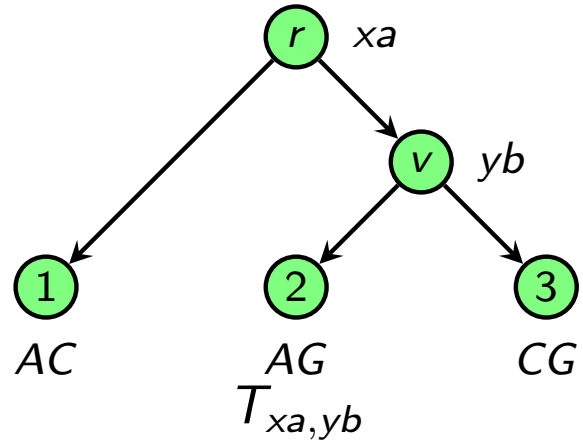
From sequences: Likelihood methods

Given a phylogenetic tree with only its leaves labeled with sequences of length m, we can compute the probability of observing these sequences at the leaves, by adding up the probabilities of all trees obtained labeling the internal nodes



From sequences: Likelihood methods

Given a phylogenetic tree with only its leaves labeled with sequences of length m, we can compute the probability of observing these sequences at the leaves, by adding up the probabilities of all trees obtained labeling the internal nodes



$$P(T) = \sum_{xa,yb \in \{A,C,G,T\}^2} P(T_{xa,yb})$$

From sequences: Likelihood methods

Problem: Given biomolecular sequences and an evolution model, find a most probable additive tree that produces them from a single sequence

Given a fixed additive tree and sequences at the leaves, the most likely sequences at the internal nodes, maximizing the probability of the tree, are computed in polynomial time by means of dynamic programming (Felsenstein's algorithm, 1981)

Finding an additive tree maximizing the probability is, of course, NP-complete

From sequences: Likelihood methods

Solution:

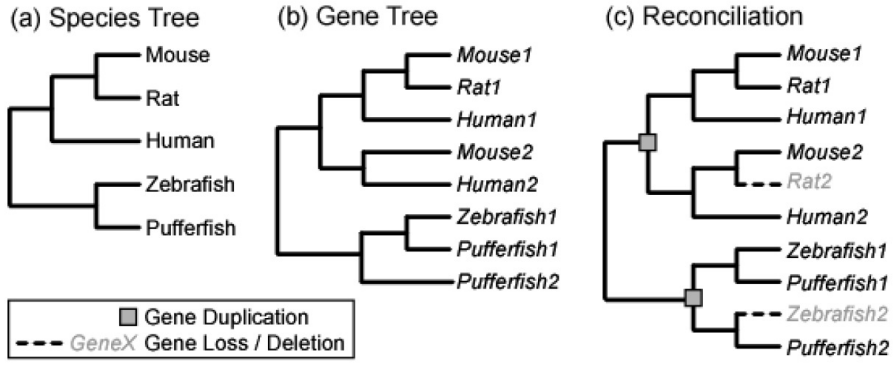
- For small n ($\leqslant 10$),
  - exhaustive search in the space of all trees with n leaves
  - computing for each one of them the probability as a function of the weights $(t_{u,v})_{(u,v)\in E}$
  - maximizing this function

- For large n, heuristic methods as in the maximum parsimony problem.

From trees

Problem: Given a family of phylogenetic trees, find a phylogenetic tree that represents as much evolutionary information contained in them as possible

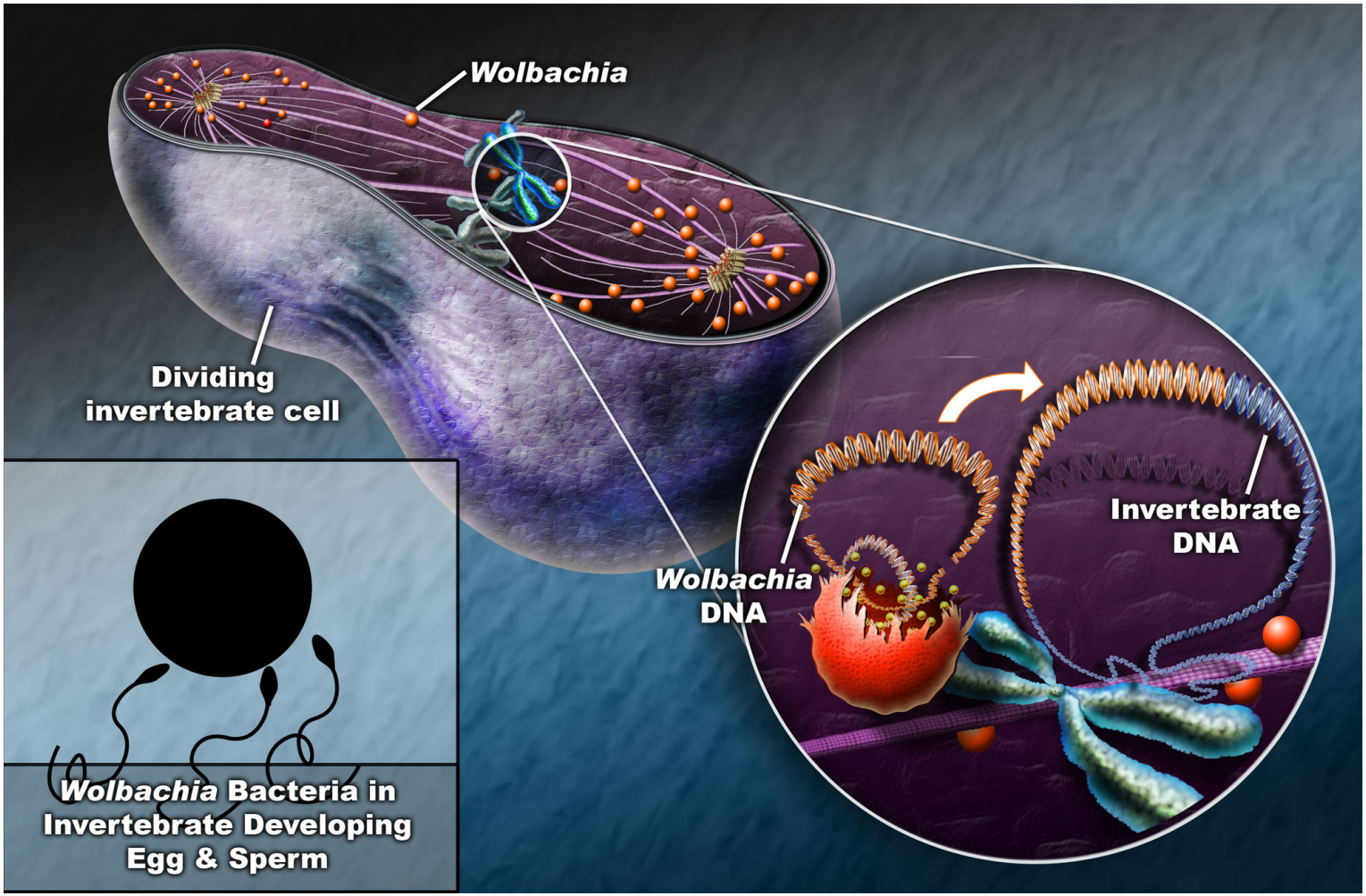
- Translate the trees into clusters or triples, and build a phylogenetic tree from their union
- Several consensus supertrees heuristic methods
- Lots of recent work on: reconcile several gene trees into a "species" tree



Phylogenetic networks

Lateral gene transfers

The whole genome of the *Wolbachia* bacterium is contained in the genome of the fly *D. Melanogaster*



Source: J. C. D. Hotopp, M. E. Clark *et al*, *Science* 317 (2007), 1753–1756

Lateral gene transfers

- Insertion of a snake gene in the genome of ruminants 50 million years ago

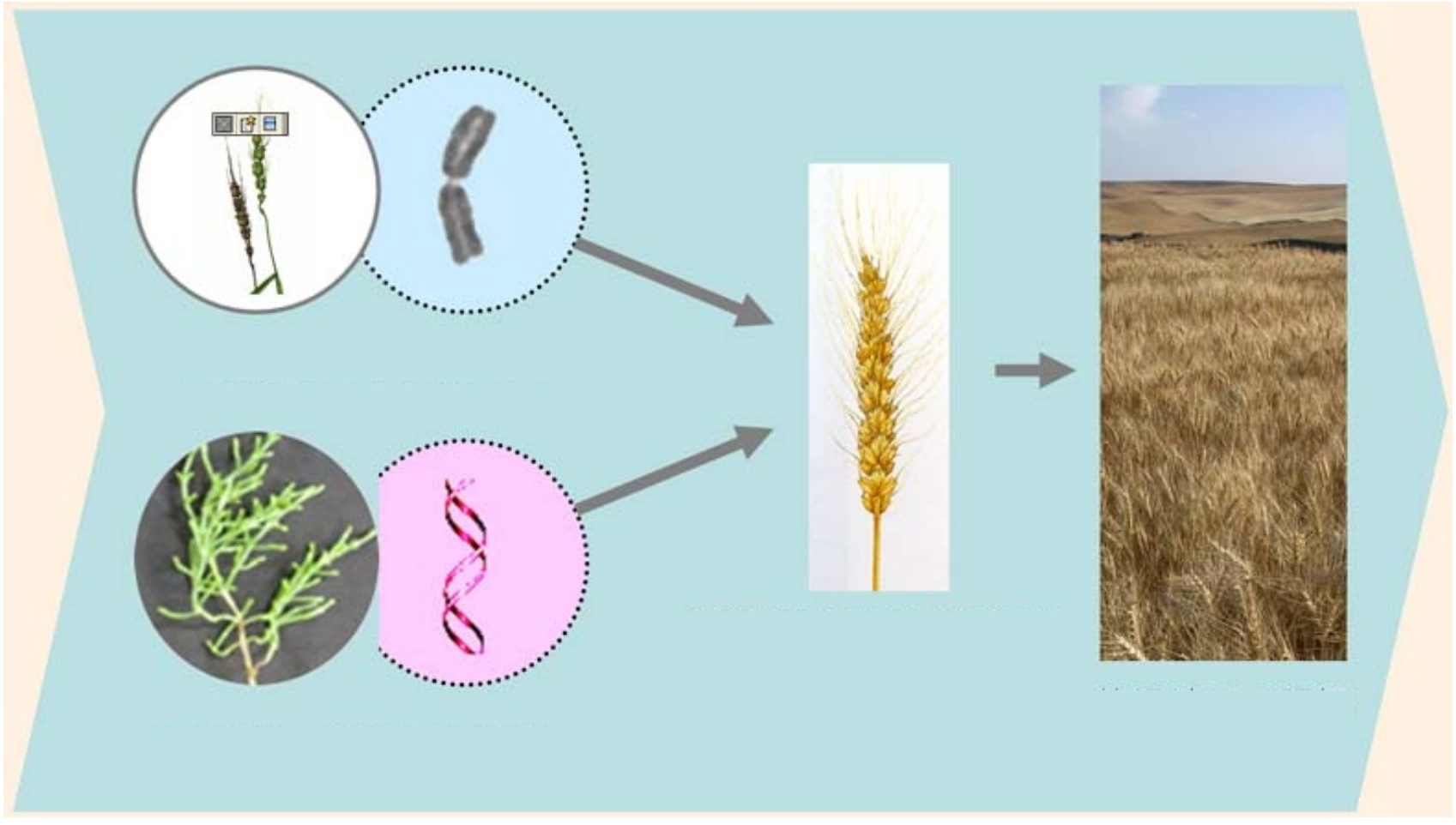
  D. Kordis, F. Gubensek, *Eur. J. Biochem.* 246 (1997), 772–779
- The mamal gene *syncytin*, key in the development of the placenta, comes from a virus

  J. P. Stoye, *PNAS* 106 (2009), 11827–1828
- The current distribution of genes seems to be a consequence of copious horizontal gene transfers in early evolutionary eras

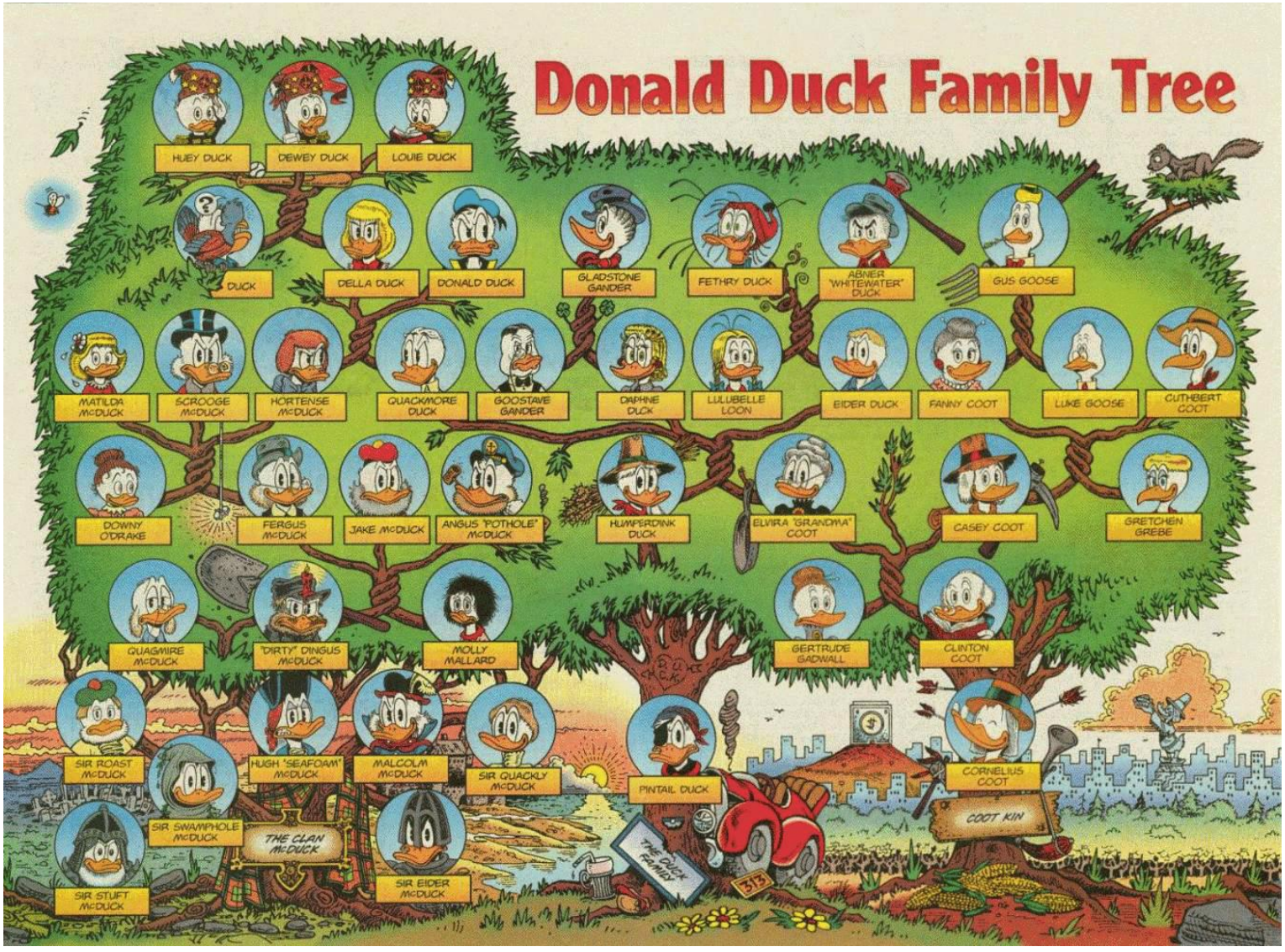
  T. Dagan, W. Martin, *PNAS* 104 (2007), 870–875

Database: HGT-DB (`http://genomes.urv.cat/HGT-DB/`)

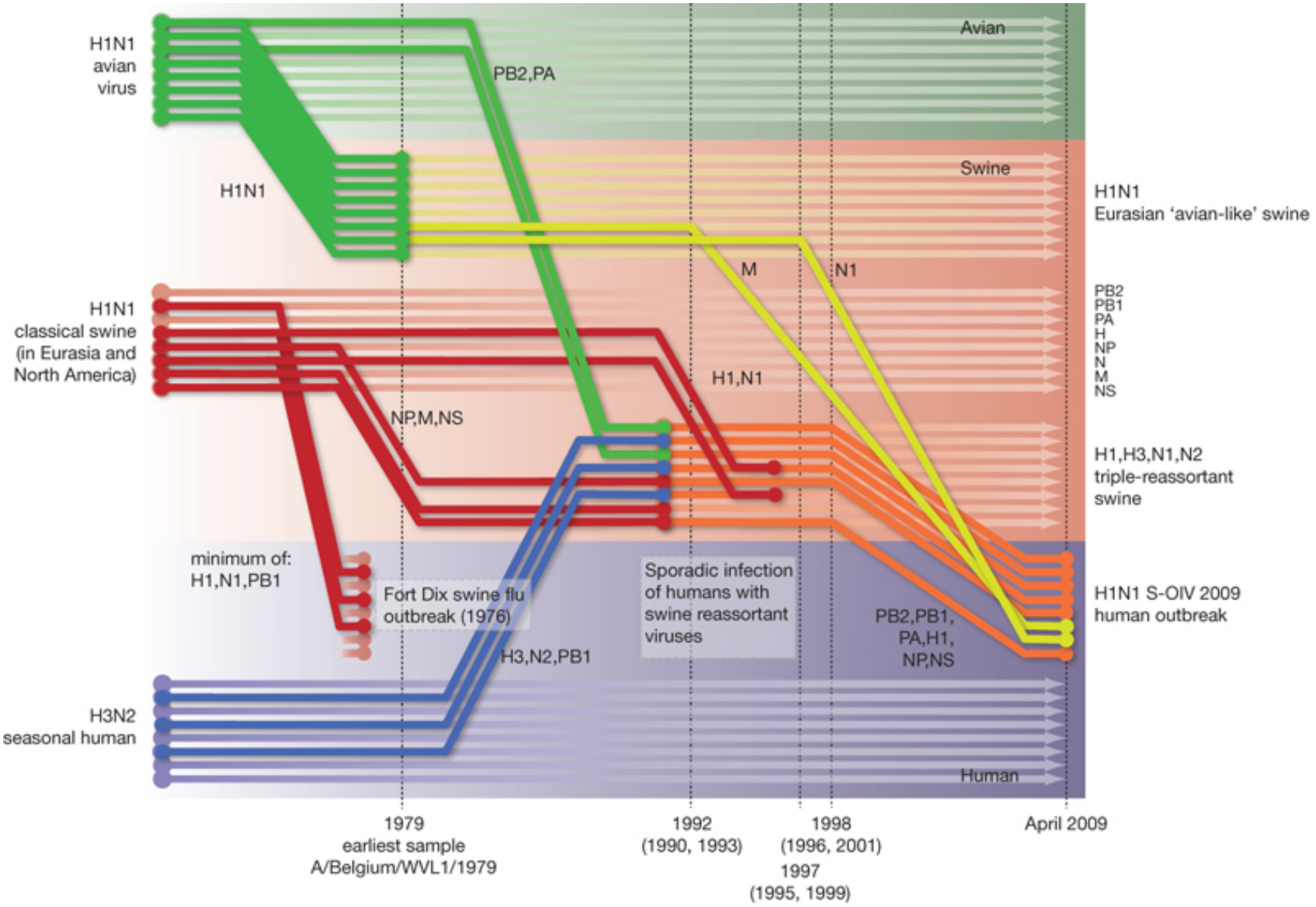
Hybridizations



Hybridizations

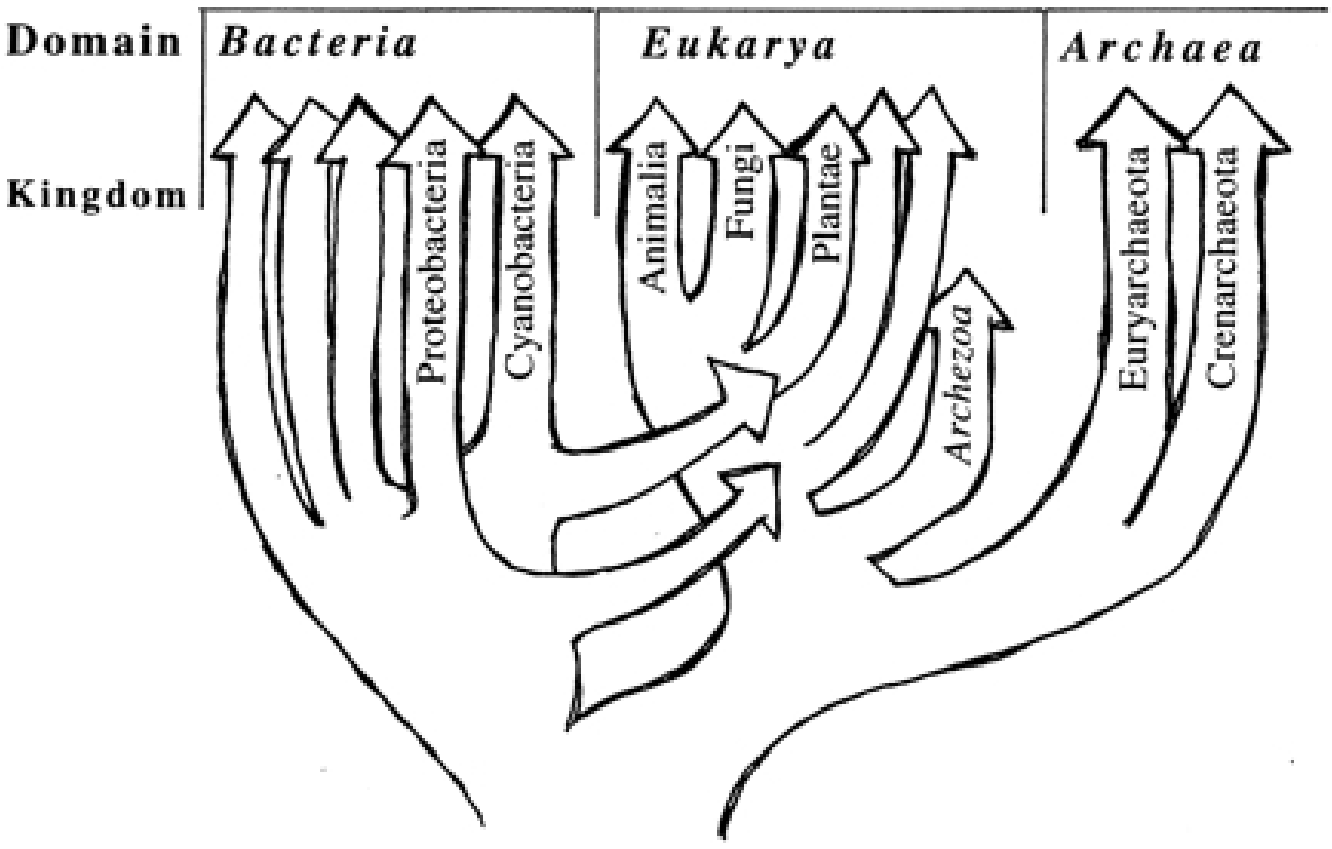


Recombinations



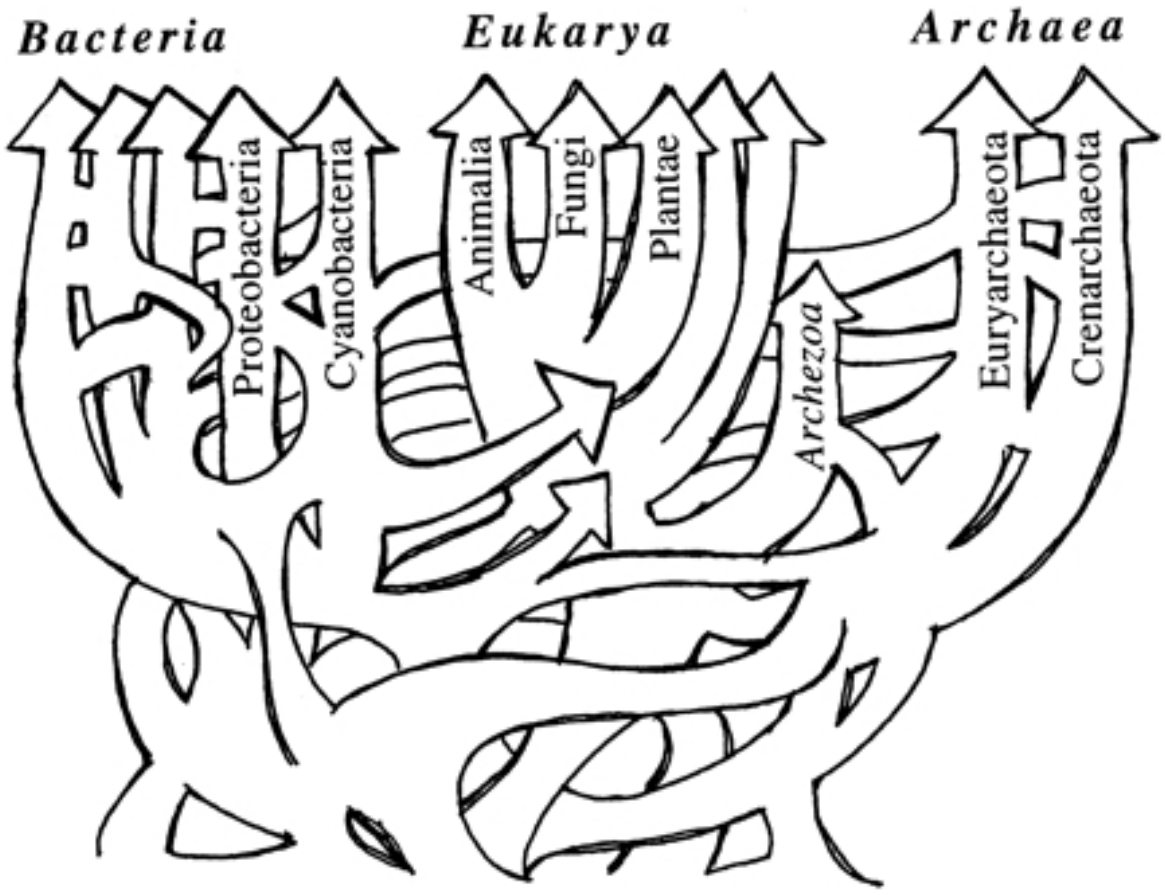
Source: G. J. D. Smith, *et al*, *Nature* 459 (2009), 1122–1125

The tree of life. . .



Source: W. F. Doolittle, *Science* 284 (1999), 2124–2128

The tree of life is not a tree

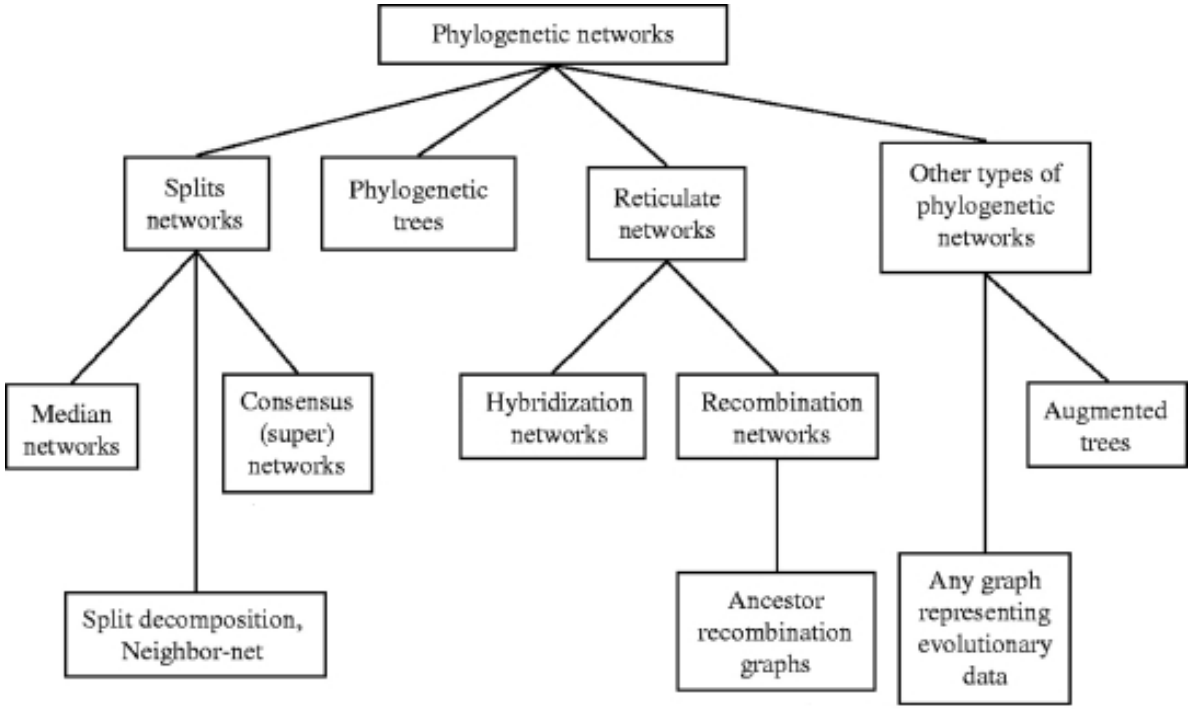


Source: W. F. Doolittle, *Science* 284 (1999), 2124–2128

Phylogenetic network

A phylogenetic network is, roughly, any graph that represents an evolutionary history (directed) or evolutionary closeness (undirected)

There are many specific definitions, imposing further conditions on the graph



Source: D. H. Huson, D. Bryant, *Molecular Biology and Evolution* 23 (2006), 254–267

Reticulate networks

Definition

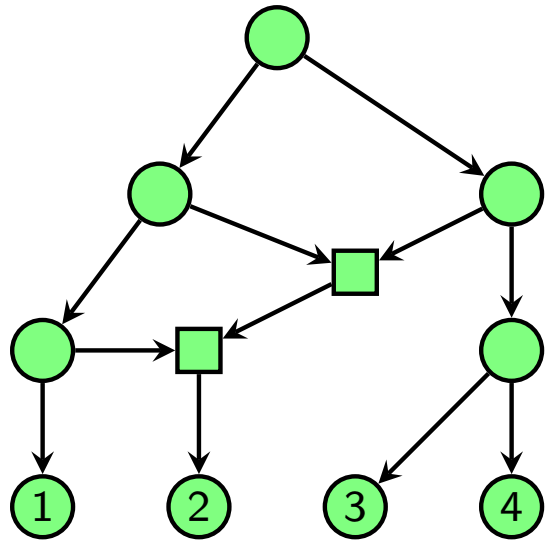
A reticulate network over S is an rDAG $N = (V, E)$ without elementary nodes and with its leaves bijectively labelled in S

- tree nodes ●: $d_{in} \leqslant 1$ y $d_{out} \neq 1$

  Represent species or mutations

- reticulations ■: $d_{in} > 1$

  Represent species obtained through reticulate events, or the reticulate events themselves



Clusters in reticulate networks

Let $N = (V, E)$ be a phyl. network over S

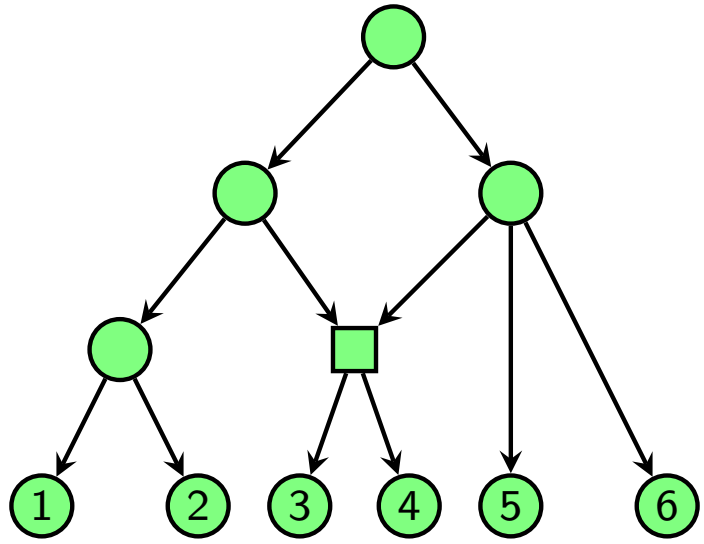
For every node v, let

$$C(v) = \text{labels of descendant leaves of } x$$

The family of clusters displayed (in the hardwired sense) by N is

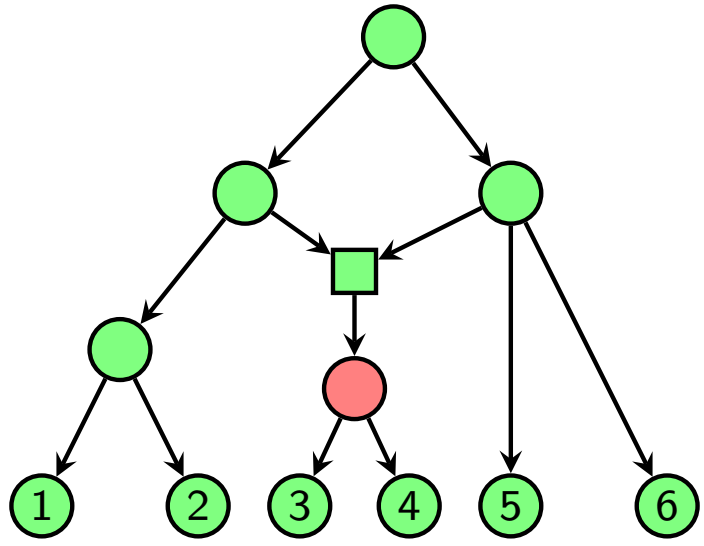
$$\mathcal{C}(N) = \{C(v) \mid v \in V \text{ is a tree node}\}$$

Clusters in reticulate networks



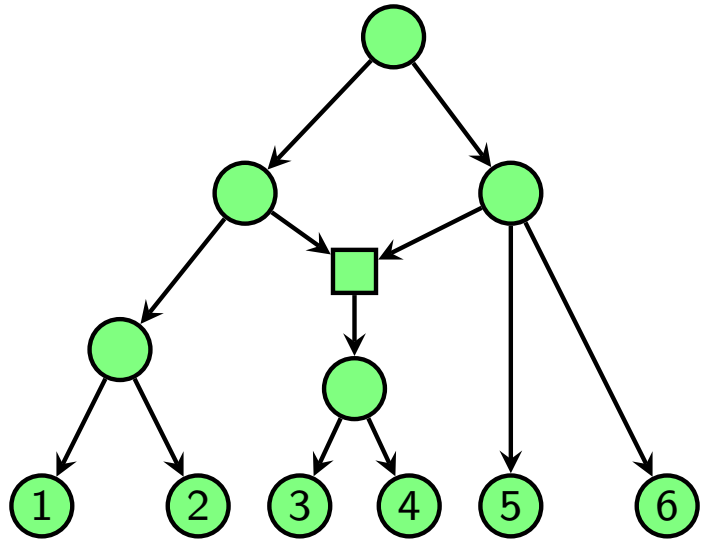
$$C(N) = \big\{\{1\},\{2\},\{3\},\{4\},\{5\},\{6\},\{1,2\},$$
$$\{1,2,3,4\},\{3,4,5,6\},\{1,2,3,4,5,6\}\big\}$$

Clusters in reticulate networks



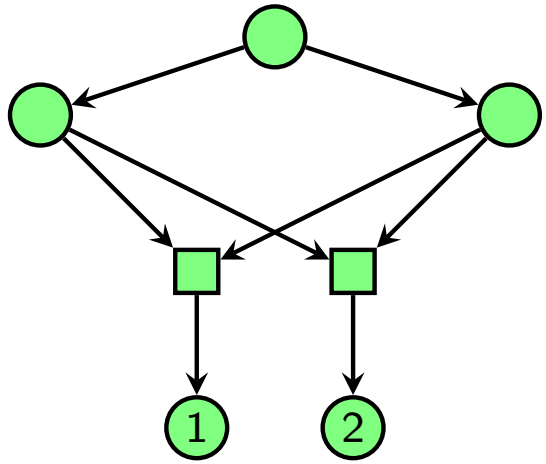
$$C(N) = \big\{\{1\},\{2\},\{3\},\{4\},\{5\},\{6\},\{1,2\},\{3,4\},$$
$$\{1,2,3,4\},\{3,4,5,6\},\{1,2,3,4,5,6\}\big\}$$

Clusters in reticulate networks



- If $v \rightsquigarrow w$, then $C(w) \subseteq C(v)$
- $C(v) \cap C(w) \neq \emptyset$ does not imply $C(v) \subseteq C(w)$ or $C(w) \subseteq C(v)$

Clusters in reticulate networks

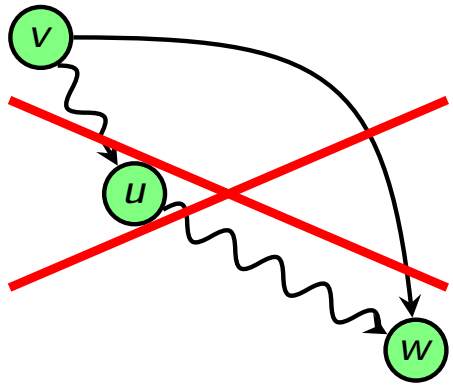


- $C(w) \subseteq C(v)$ does not imply $v \rightsquigarrow w$

Cluster networks

A cluster network on S is an S-rDAG G such that:

1. Every reticulation has exactly one child, and it is a tree node
2. If $C(w) \subsetneq C(v)$, then $v \rightsquigarrow w$
3. $C(v) = C(w)$ iff $v = w$ or they are a reticulation and its only child
4. If $(v, w) \in E$, then there exists no $u \in V$ such that $C(w) \subsetneq C(u) \subsetneq C(v)$



Computing cluster networks

Given a family $\mathcal{C}$ of clusters of S containing all singletons, a cluster network $N_C(\mathcal{C})$ such that $\mathcal{C}(N_C(\mathcal{C})) = \mathcal{C}$ can be obtained as follows:

Cluster-popping algorithm:

1. Draw the Hasse diagram of $(\mathcal{C} \cup \{S\}, \subseteq)$ and root it at S
2. Insert additional tree edges with source reticulations to ensure (1)
3. Label leaves with the corresponding taxa

Computing cluster networks

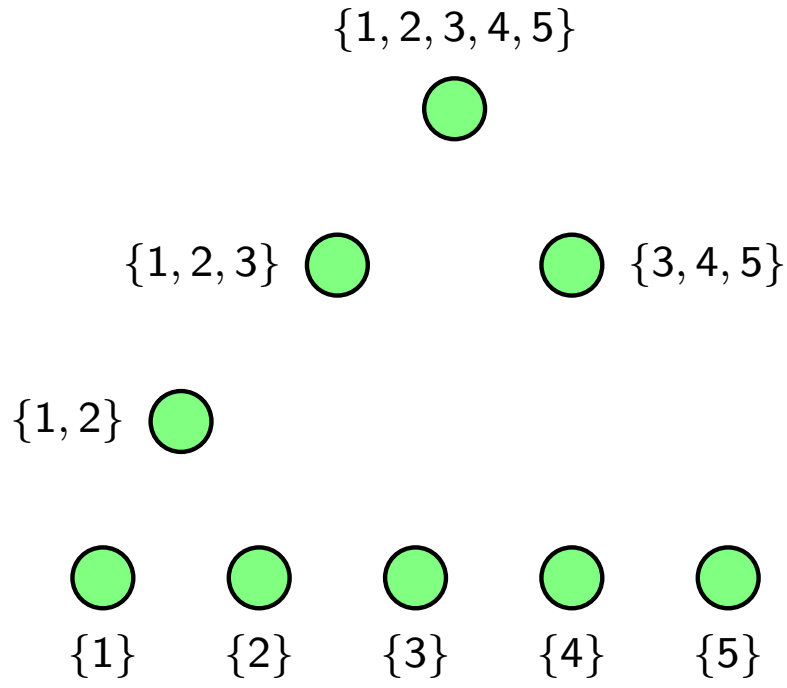
Example: $S = \{1, 2, 3, 4, 5\}$

$\mathcal{C} = \big\{\{1, 2, 3\}, \{3, 4, 5\}, \{1, 2\}, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}\big\}$

Computing cluster networks

Example: $S = \{1, 2, 3, 4, 5\}$

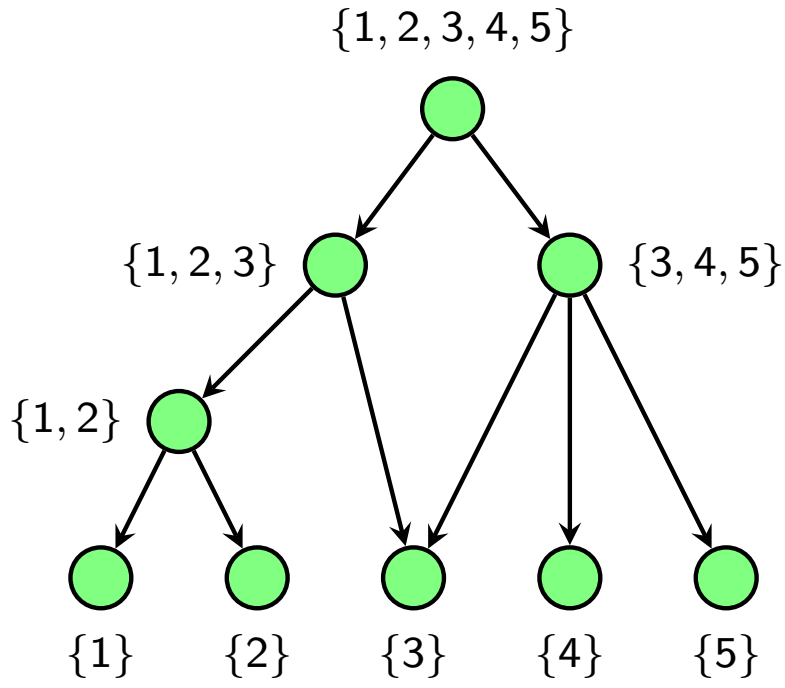
$\mathcal{C} = \big\{\{1, 2, 3\}, \{3, 4, 5\}, \{1, 2\}, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}\big\}$



Computing cluster networks

Example: $S = \{1, 2, 3, 4, 5\}$

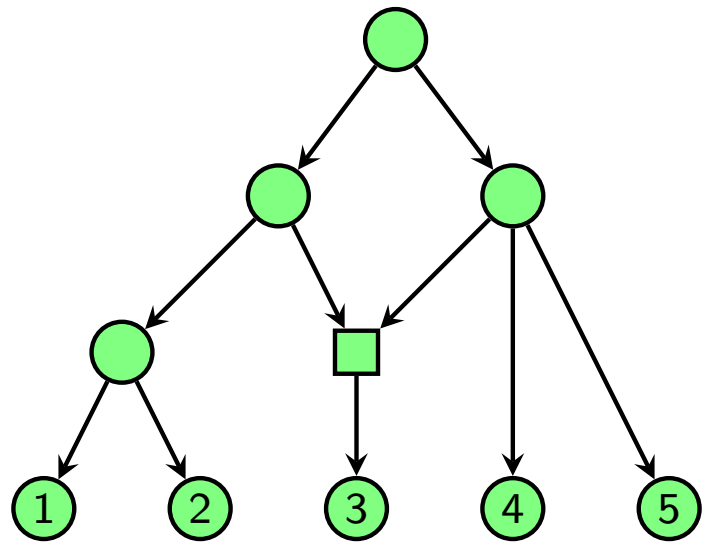
$\mathcal{C} = \big\{\{1, 2, 3\}, \{3, 4, 5\}, \{1, 2\}, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}\big\}$



Computing cluster networks

Example: $S = \{1, 2, 3, 4, 5\}$

$\mathcal{C} = \big\{\{1, 2, 3\}, \{3, 4, 5\}, \{1, 2\}, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}\big\}$



Computing cluster networks

Theorem

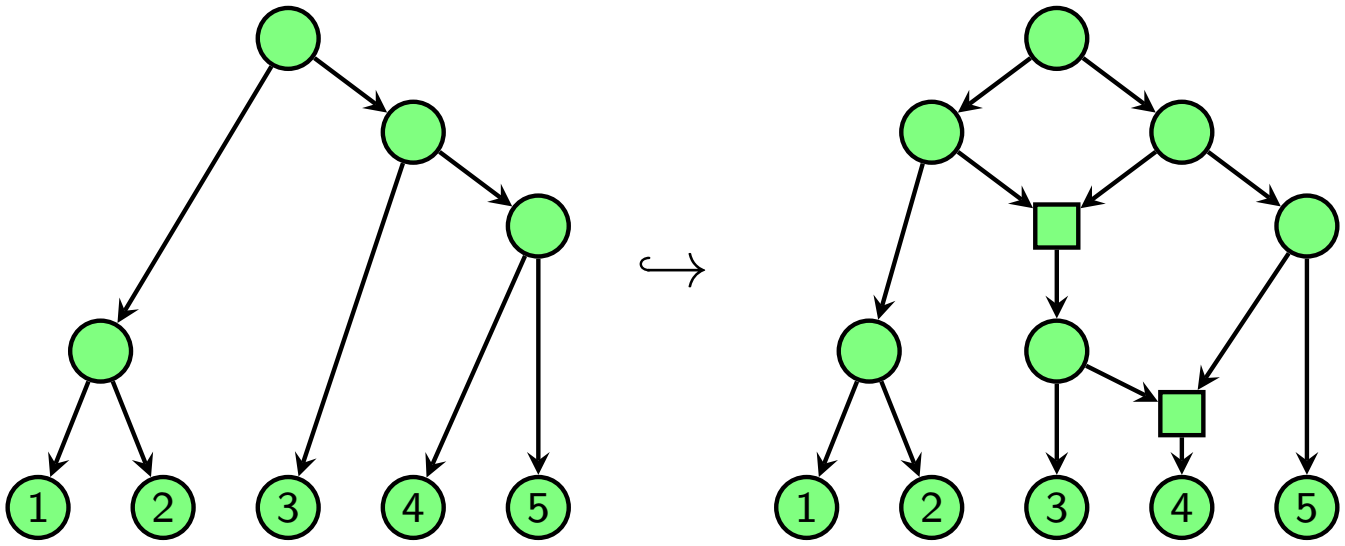
$N_C(\mathcal{C})$ is a cluster network and $N_C(\mathcal{C}) = \mathcal{C}$.

Theorem

Let N, N' be cluster networks. If $\mathcal{C}(N) = \mathcal{C}(N')$, then $N \cong N'$.

Embeddings

A phylogenetic tree T is represented by a reticulate network N when it can be obtained from N by deleting, in every reticulation, all incoming edges but one, and then suppressing elementary nodes



Cluster network as consensus

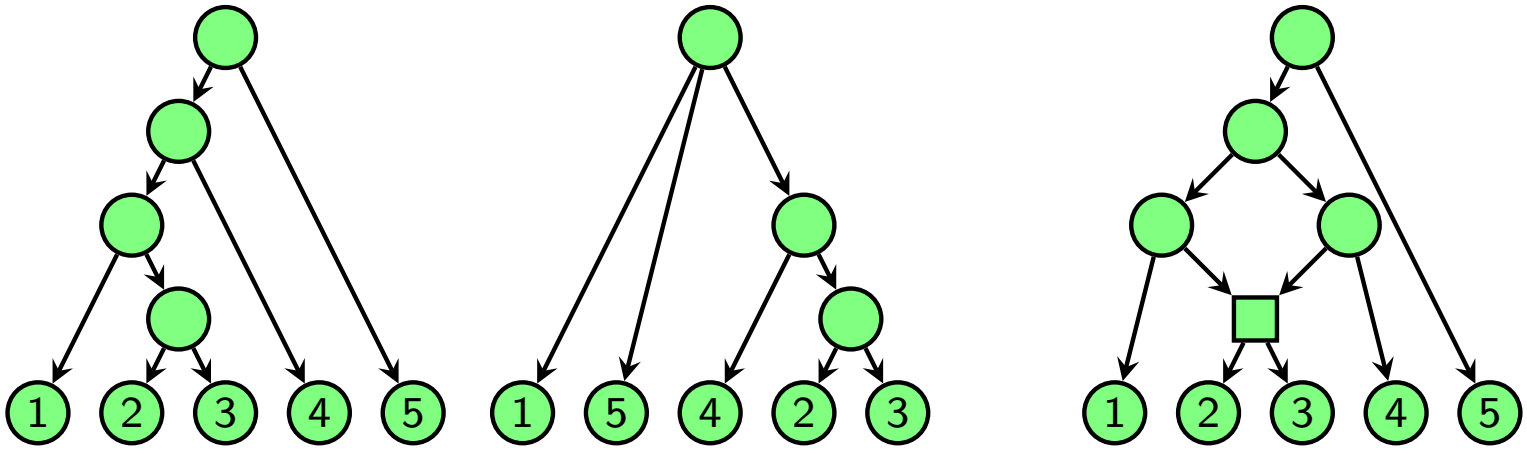
Theorem

Let $\mathcal{T}$ be a family of trees over S. Then, $N_C(\mathcal{C}(\mathcal{T}))$ represents a refinement of every tree in $\mathcal{T}$.

Cluster network as consensus

Theorem

Let $\mathcal{T}$ be a family of trees over S. Then, $N_C(\mathcal{C}(\mathcal{T}))$ represents a refinement of every tree in $\mathcal{T}$.



Refinement cannot be avoided in the statement

Active problem: Find a reticulate network (possibly with extra properties) that represents a family of trees

Hardwired and softwired clusters

Let N be a reticulate network over S, and $C \subseteq S$.

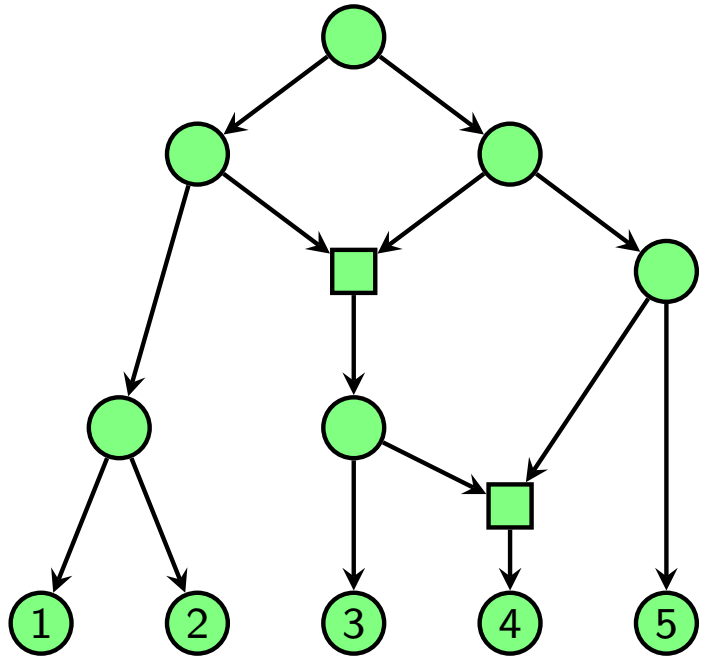
- $C \in \mathcal{C}(N)$ iff $C = C_N(v)$ for some tree node v
- $C \in \mathcal{C}_{soft}(N)$ iff $C = C_T(v)$ for some node v in a tree T represented by N

N represents $\mathcal{C}$ in the hardwired sense if $\mathcal{C} \subseteq \mathcal{C}(N)$

N represents $\mathcal{C}$ in the softwired sense if $\mathcal{C} \subseteq \mathcal{C}_{soft}(N)$

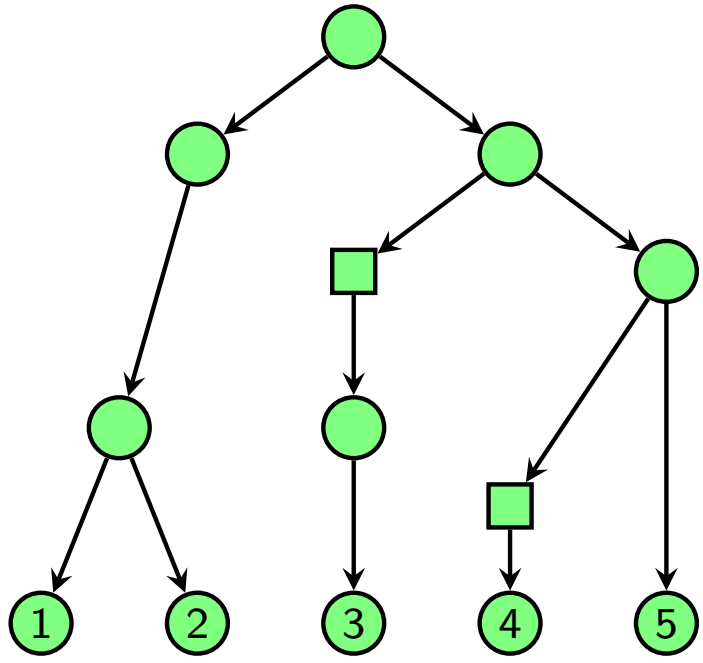
Every tree node in N represents only one cluster in the hardwired sense, but may represent several clusters in the softwired sense

Hardwired and softwired clusters



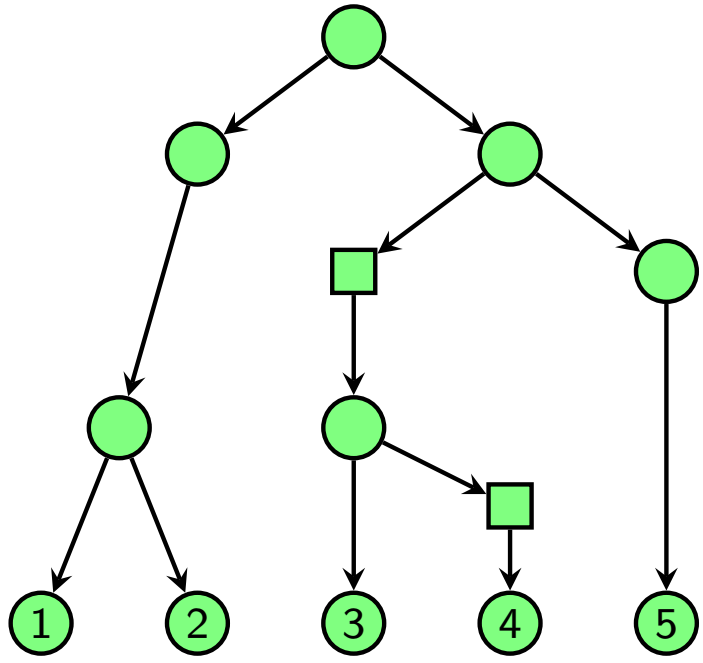
$$\mathcal{C}(N) = \big\{\{1\},\{2\},\{3\},\{4\},\{5\},\{1,2\},\{3,4\},\{4,5\},\{1,2,3,4\},\{3,4,5\}\big\}$$

Hardwired and softwired clusters



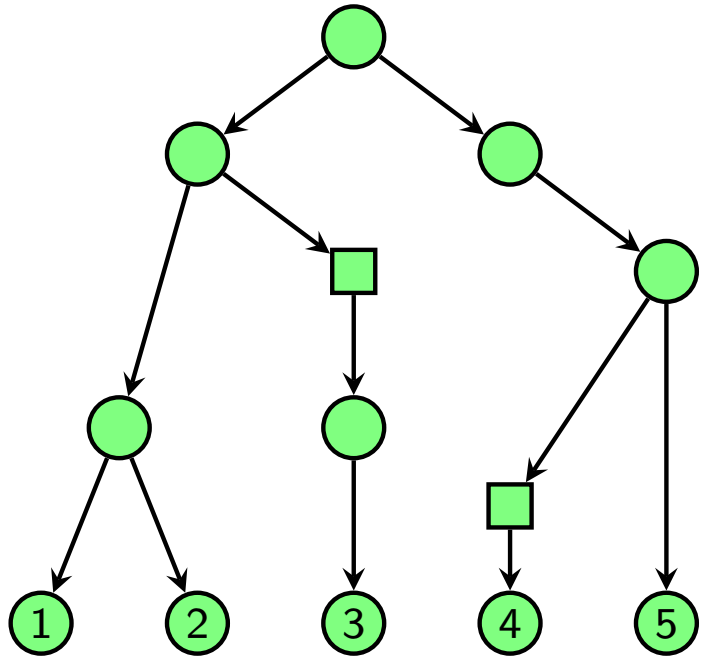
$$\mathcal{C}_{soft}(N) = \{\{1\},\{2\},\{3\},\{4\},\{5\},\{1,2\},\{4,5\},\{3,4,5\}$$

Hardwired and softwired clusters



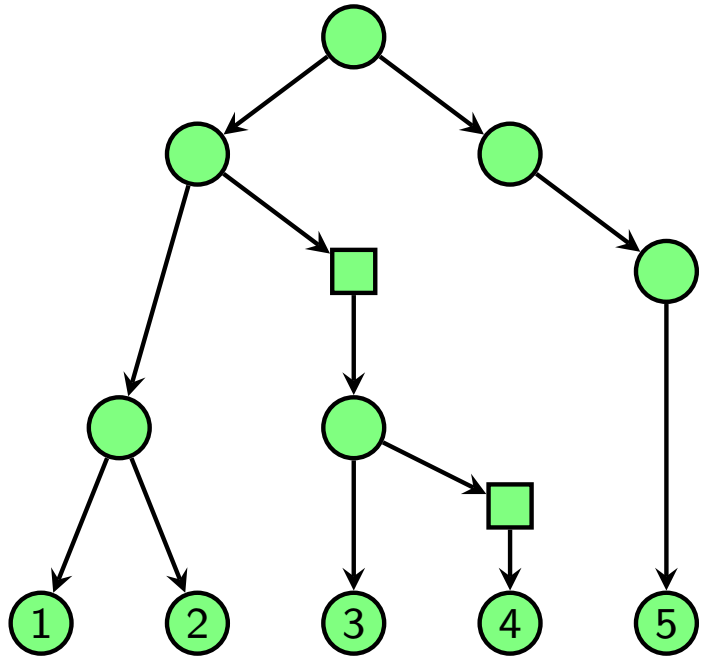
$$\mathcal{C}_{soft}(N) = \{\{1\},\{2\},\{3\},\{4\},\{5\},\{1,2\},\{4,5\},\{3,4,5\}$$
$$\{3,4\}$$

Hardwired and softwired clusters



$$\mathcal{C}_{soft}(N) = \{\{1\},\{2\},\{3\},\{4\},\{5\},\{1,2\},\{4,5\},\{3,4,5\}$$
$$\{3,4\},\{1,2,3\}$$

Hardwired and softwired clusters



$$\mathcal{C}_{soft}(N) = \big\{\{1\},\{2\},\{3\},\{4\},\{5\},\{1,2\},\{4,5\},\{3,4,5\}$$
$$\{3,4\},\{1,2,3\},\{1,2,3,4\}\big\}$$

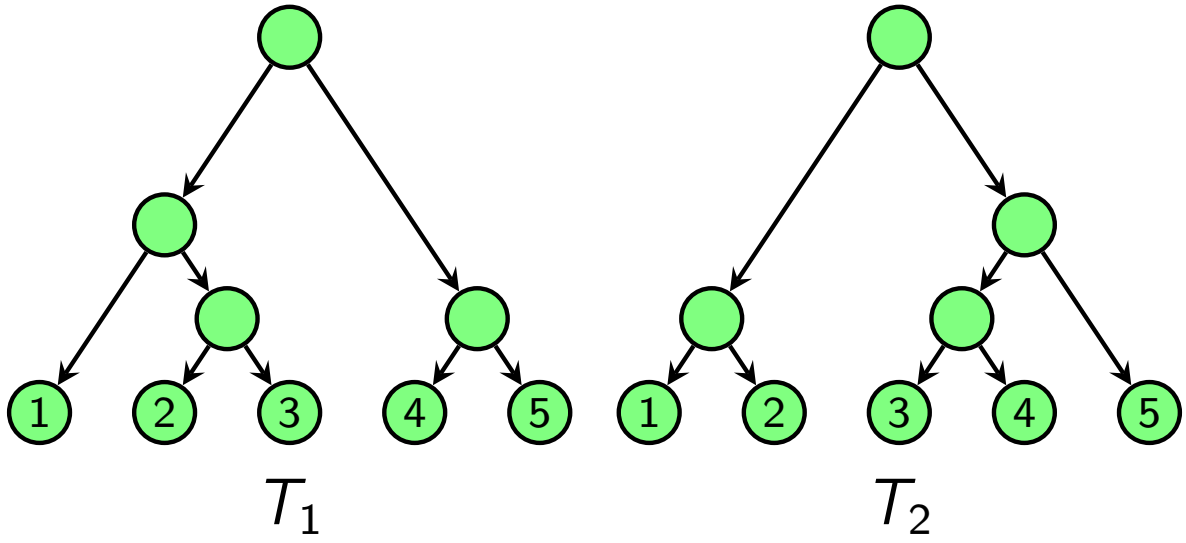
Hardwired and softwired clusters

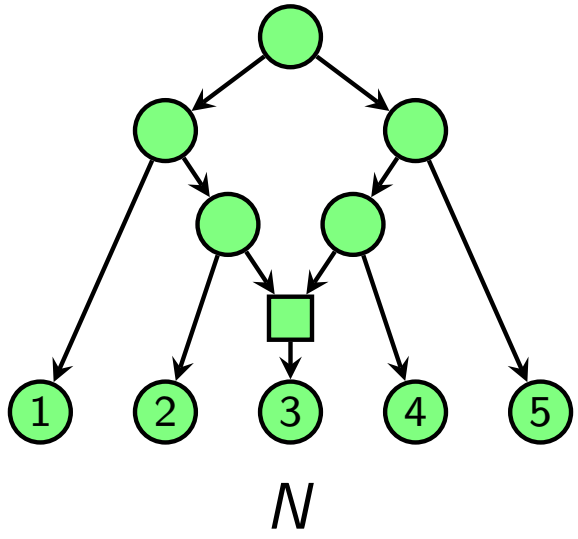
Proposition

$\mathcal{C}(N) \subseteq \mathcal{C}_{soft}(N)$

Hardwired and softwired clusters

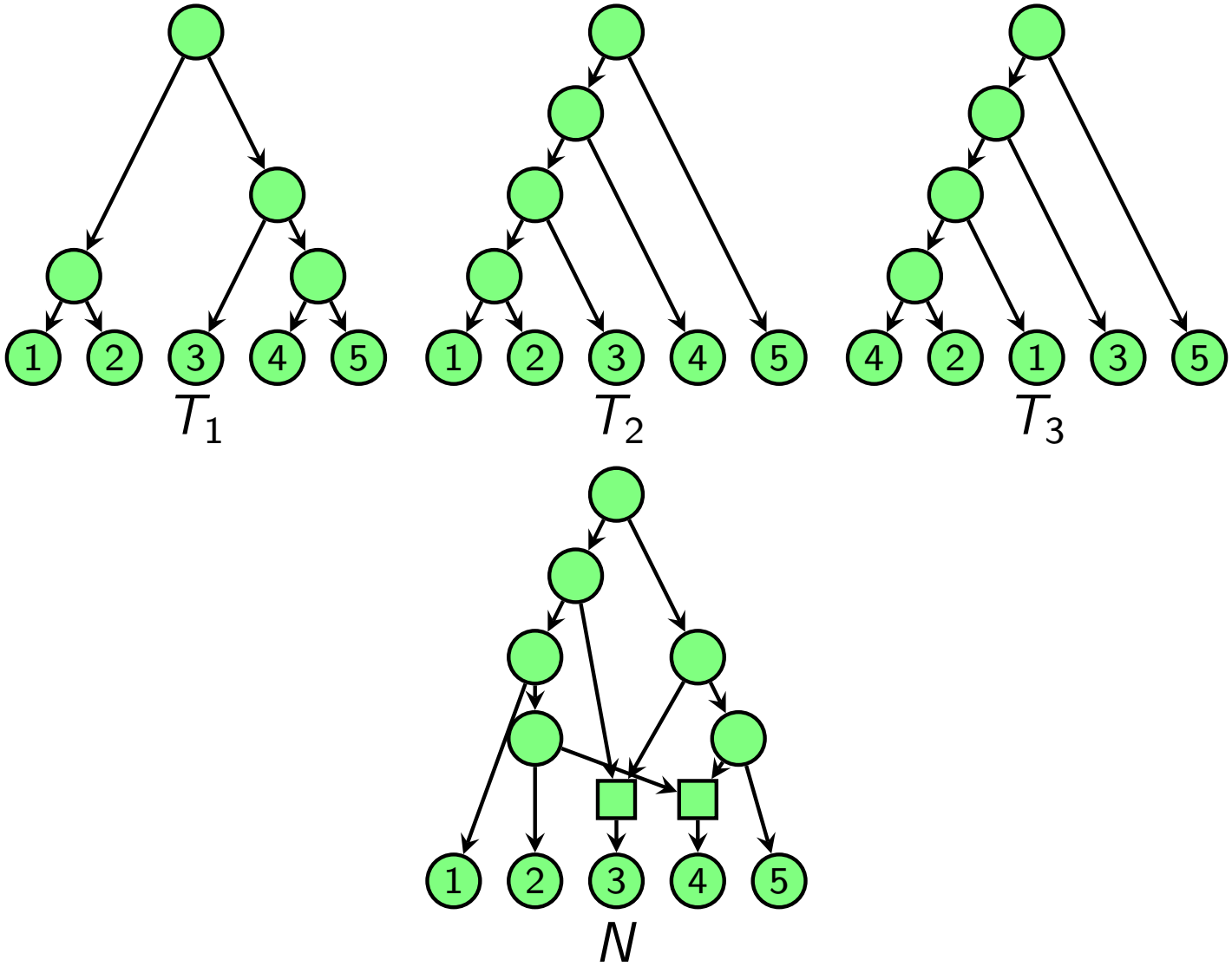
If N represents $\mathcal{T}$, then $\mathcal{C}(\mathcal{T}) \subseteq \mathcal{C}_{soft}(N)$, but not necessarily $\mathcal{C}(\mathcal{T}) \subseteq \mathcal{C}(N)$





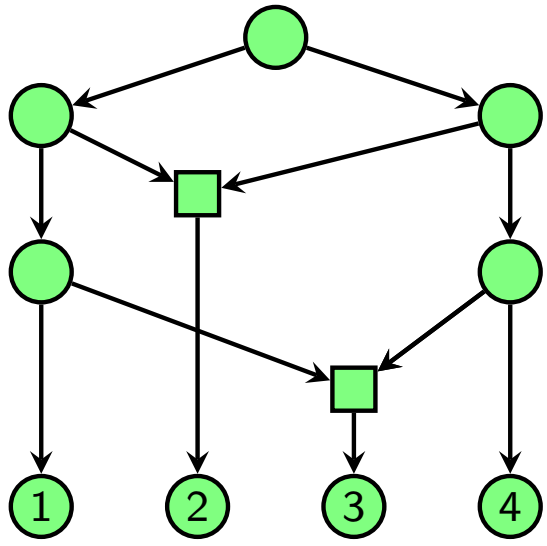
Hardwired and softwired clusters

N representing softwired $\mathcal{C}(\mathcal{T})$ need not represent $\mathcal{T}$



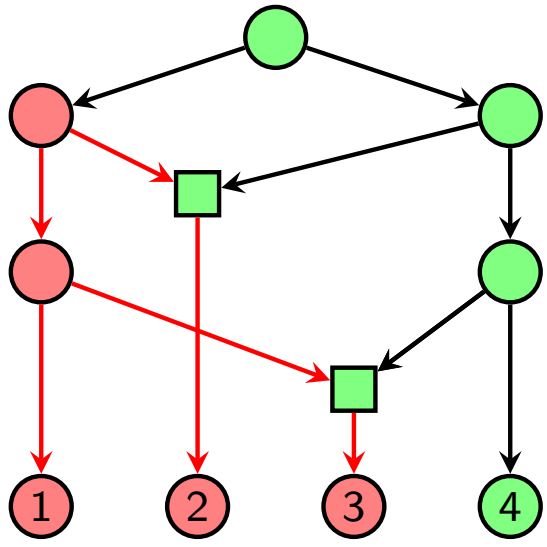
Triples in a network

A triple $ab|c$ is embedded in a reticulate network $N = (V, E)$ when there exist $u, v \in V$ and paths $u \rightsquigarrow c$, $u \rightsquigarrow v$, $v \rightsquigarrow a$ and $v \rightsquigarrow v$ that are node-disjoint (except at their end-points)



Triples in a network

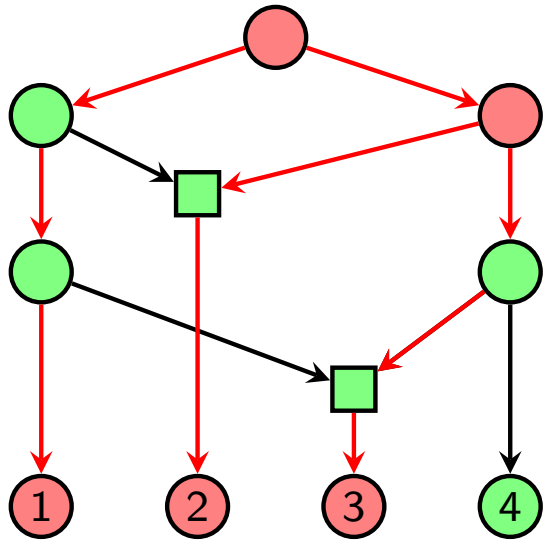
A triple $ab|c$ is embedded in a reticulate network $N = (V, E)$ when there exist $u, v \in V$ and paths $u \rightsquigarrow c$, $u \rightsquigarrow v$, $v \rightsquigarrow a$ and $v \rightsquigarrow v$ that are node-disjoint (except at their end-points)



$13|2$

Triples in a network

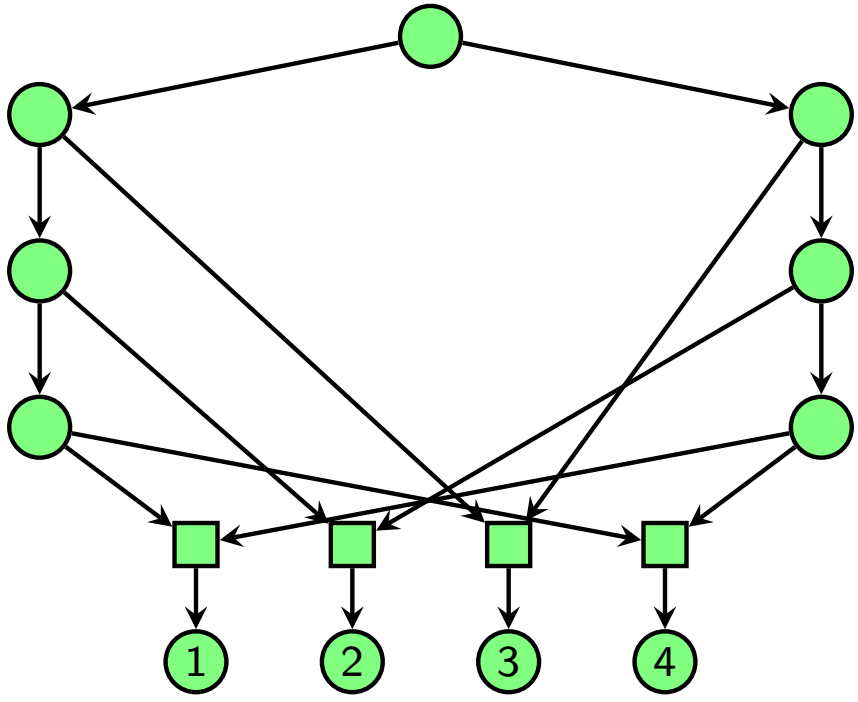
A triple $ab|c$ is embedded in a reticulate network $N = (V, E)$ when there exist $u, v \in V$ and paths $u \rightsquigarrow c$, $u \rightsquigarrow v$, $v \rightsquigarrow a$ and $v \rightsquigarrow v$ that are node-disjoint (except at their end-points)



$23|1$

Triples in a network

There exist reticulate networks containing all triples over S



Restrictions

Most problems related to general reticulate networks are hard:

- The isomorphism problem is believed to lie in NP−P
- Deciding whether a reticulate network represents in the softwired sense a given cluster (Cluster containment problem) is NP-complete
- Deciding the minimum number of reticulations in a reticulate network representing in the softwired sense a given family of clusters is NP-complete
- Deciding the minimum number of reticulations in a reticulate network representing in the softwired sense a given family of triples is NP-complete
- ...

Restrictions

A solution is to restrict the class of reticulate networks

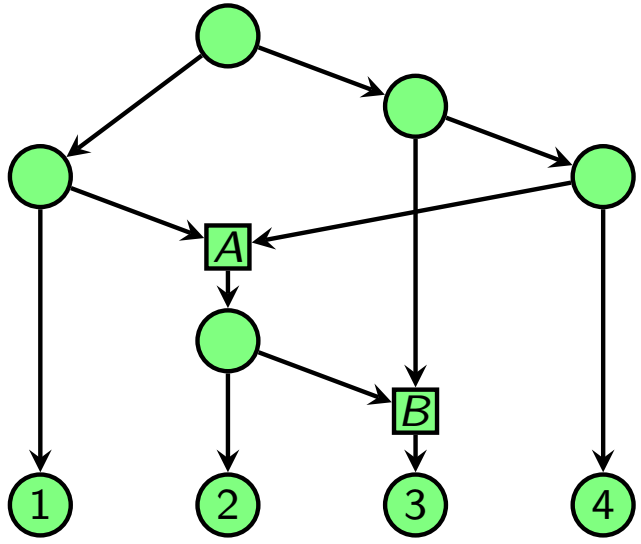
Several restricted classes have been introduced so far, some with biological meaning, some artificial but useful

11 such "simple" restrictions are discussed in:

- `http://phylonetworks.blogspot.com.es/2013/03/different-topological-restrictions-of.html`
- `http://phylonetworks.blogspot.com.es/2013/03/topological-restrictions-some-comments.html`

Reticulation cycles

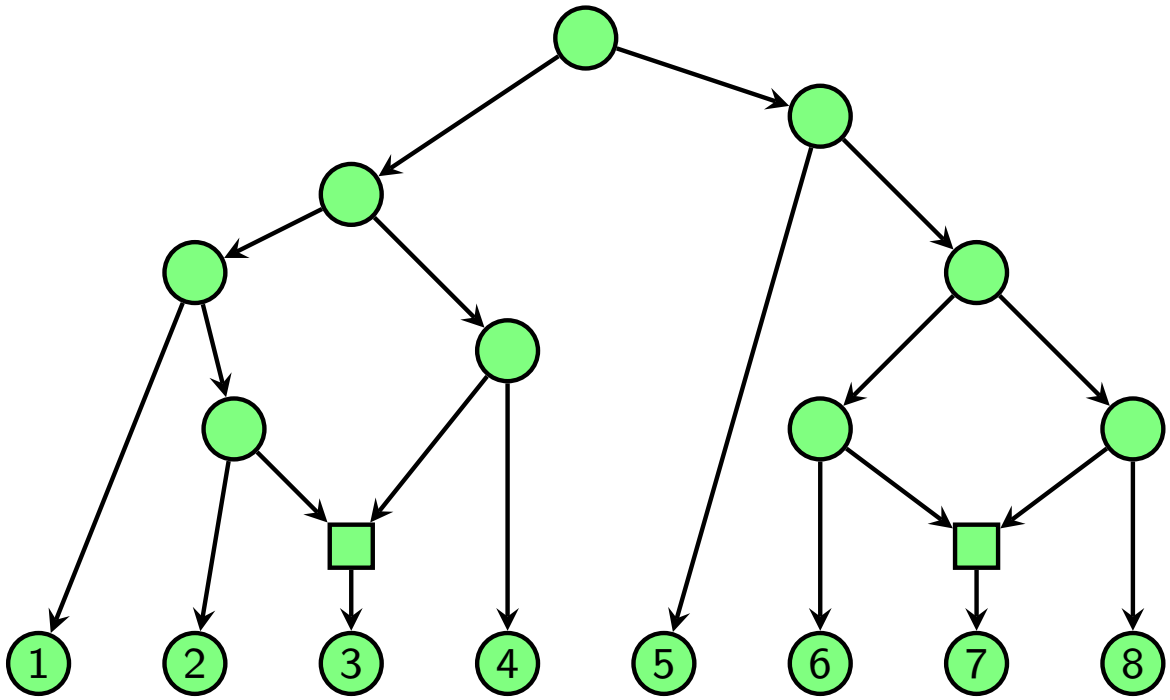
A reticulation cycle for a reticulate node H is any pair of paths ending in H with the same origin and no other node in common



1 reticulation cycle for A, 2 for B

Galled trees

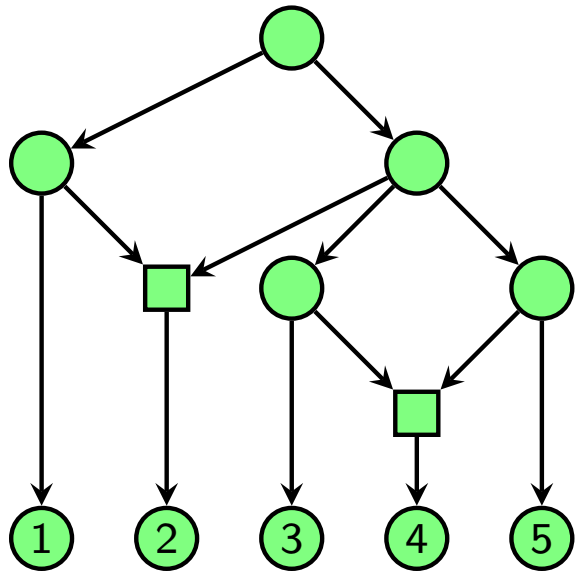
A reticulate network is a galled tree when every pair of reticulation cycles have disjoint sets of edges



A galled tree

Galled trees

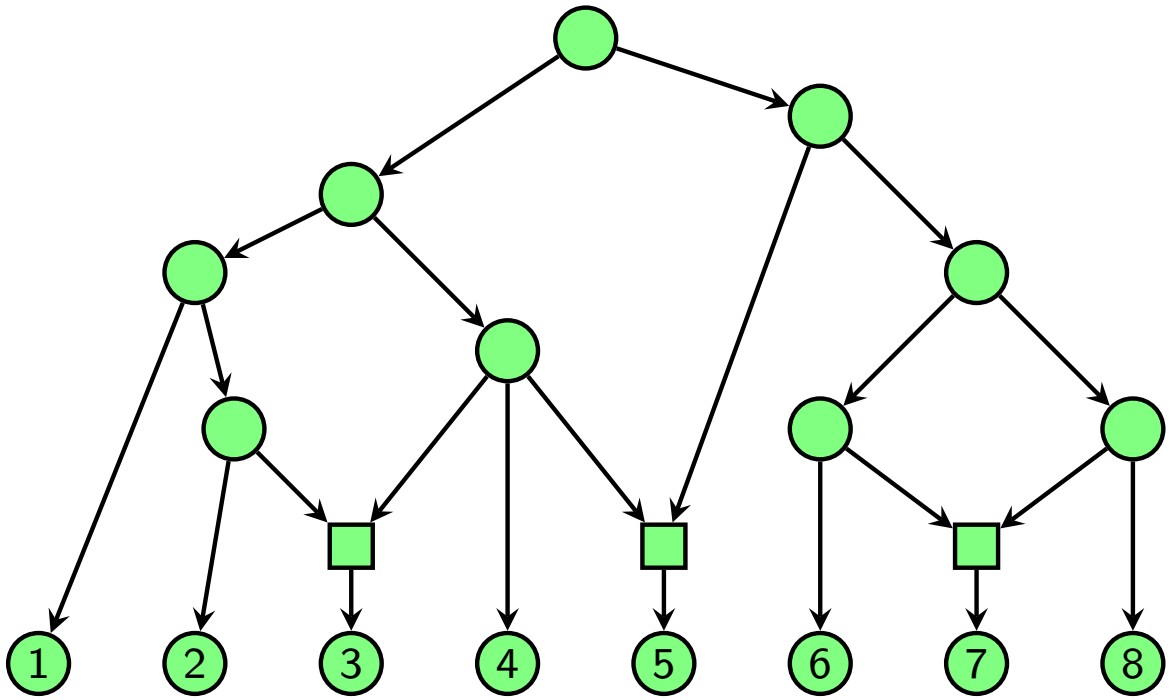
A reticulate network is a galled tree when every pair of reticulation cycles have disjoint sets of edges



A galled tree

Galled trees

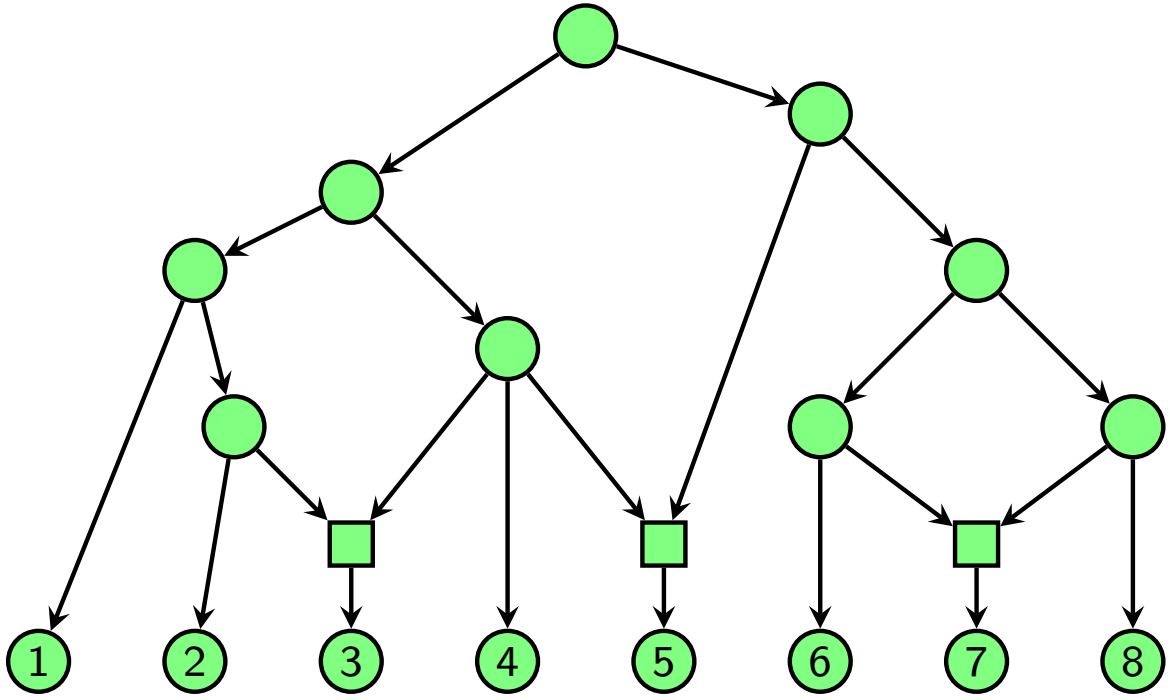
A reticulate network is a galled tree when every pair of reticulation cycles have disjoint sets of edges



Not a galled tree

Tree-child networks

A reticulate network is a tree-child network when reticulations have exactly one (tree) child and every internal node has some tree child



A tree-child network

Properties of tree-child networks

- Every galled tree with out-degree 1 reticulations is tree-child
- Every reticulate node is a strict ancestor of all its descendants
- The cluster containment problem can be solved in polynomial time
- The isomorphism problem can be solved in polynomial time

Reconstruction of restricted networks

- Polynomial-time algorithm that computes a galled tree that represents (in the softwired or in the hardwired sense) a given family of clusters, if one exists
- Polynomial-time algorithm that computes a galled tree that represents a given family of dense triples, if one exists
- The non-dense case is open
- Polynomial-time algorithm that computes a tree-child network that represents (in the softwired or in the hardwired sense) a given family of clusters, if one exists
- The reconstruction of tree-child networks from triples is an open problem (we are working on it)

Reconstruction of phylogenetic networks

A very active field or research

A further important problem: Interpreting the reticulations

Who's who in phylogenetic networks:
`http://www2.lirmm.fr/~gambette/PhylogeneticNetworks/`

Basic bibliography