AlgoB seminar, LIGM
26/05/2015 - Champs-sur-Marne

Finding a tree in a phylogenetic network

Philippe Gambette

Main results: joint work with Andreas Gunawan, Anthony Labarre, Stéphane Vialette and Louxin Zhang.

Outline

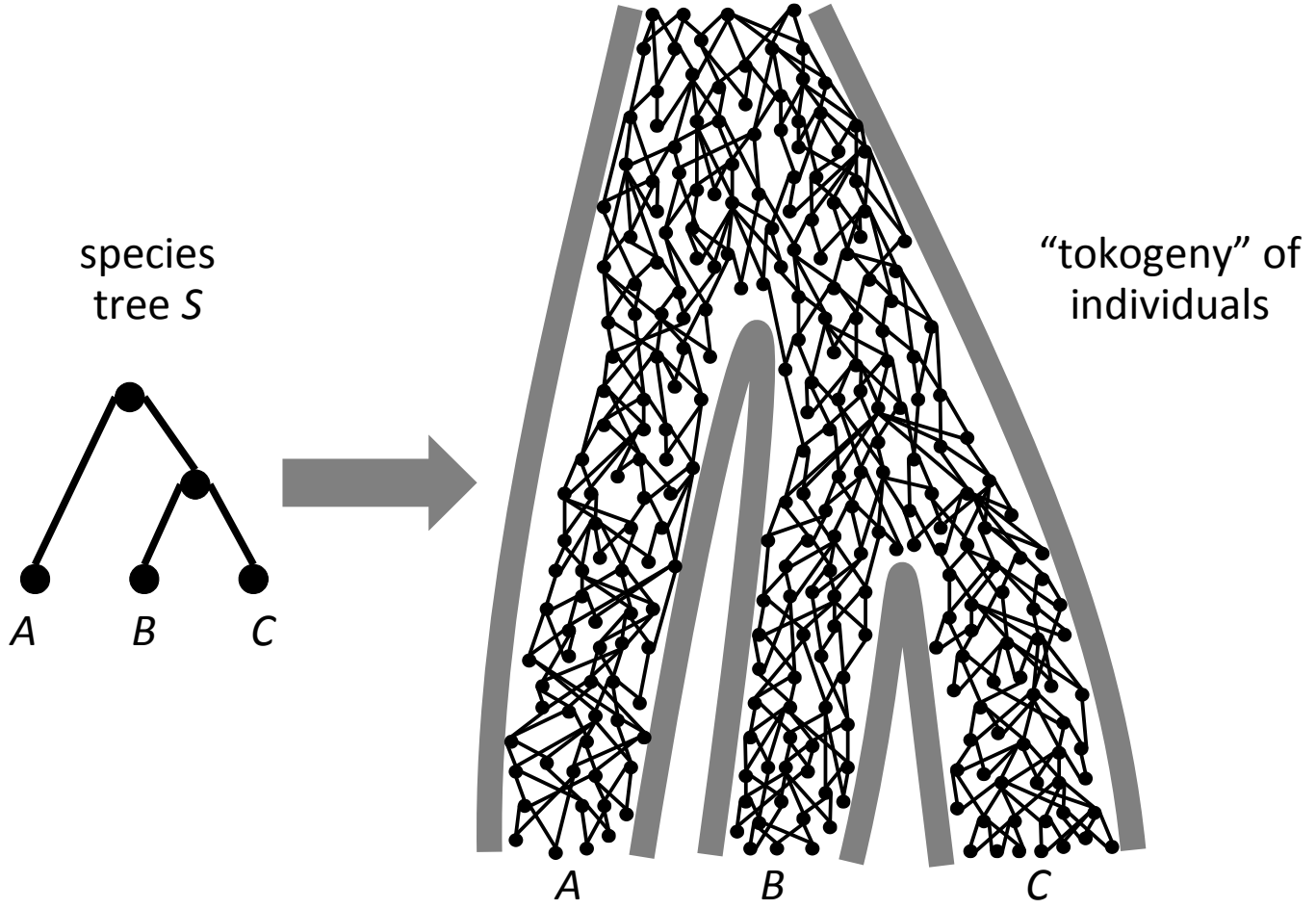
- Phylogenetic networks
- Who is who in phylogenetic networks?
- Classes of phylogenetic networks
- The Tree Contaiment Problem
- A quadratic-time algorithm on quasi-stable networks

Outline

- Phylogenetic networks
- Who is who in phylogenetic networks?
- Classes of phylogenetic networks
- The Tree Contaiment Problem
- A quadratic-time algorithm on quasi-stable networks

Phylogenetic trees

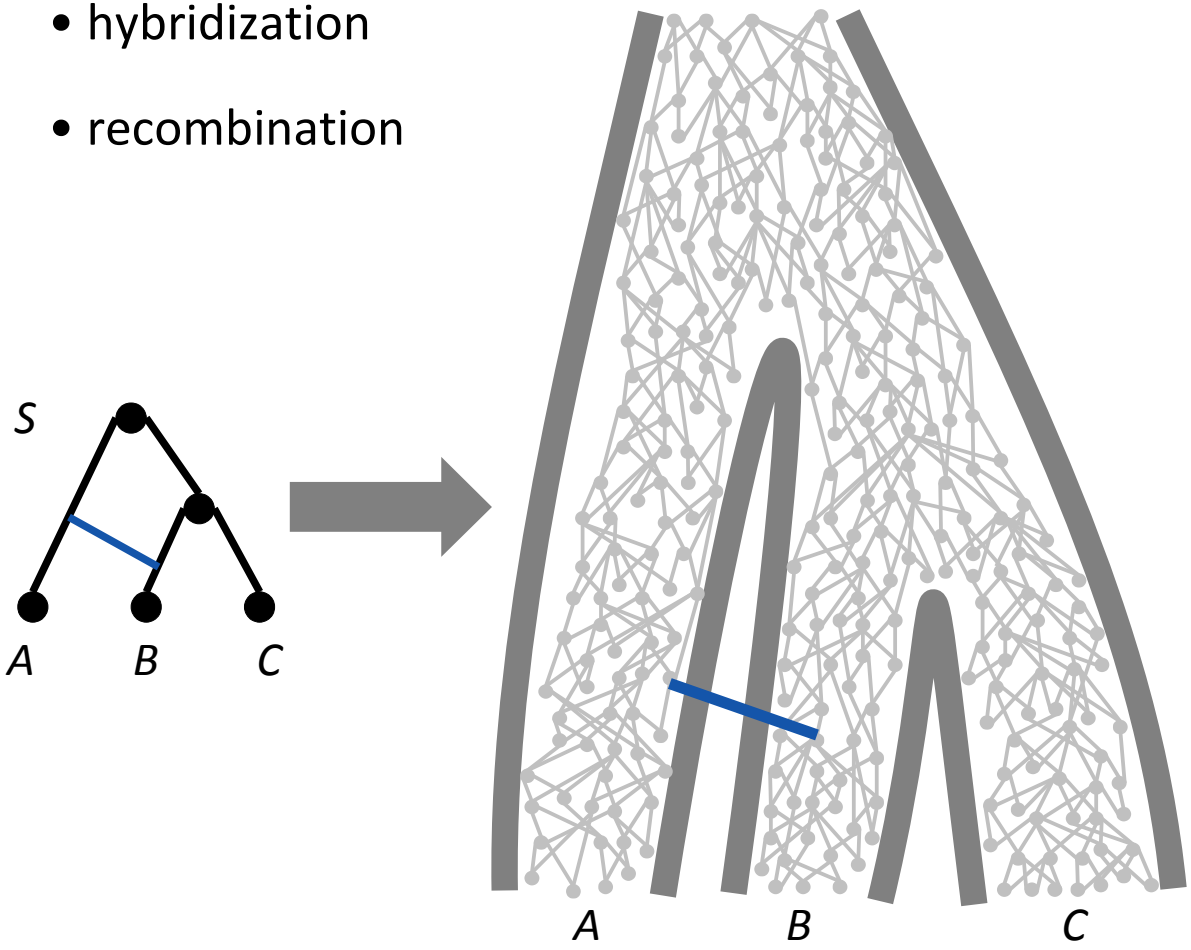
Phylogenetic tree of a set of species



Genetic material transfers

Transfers of genetic material between coexisting species:

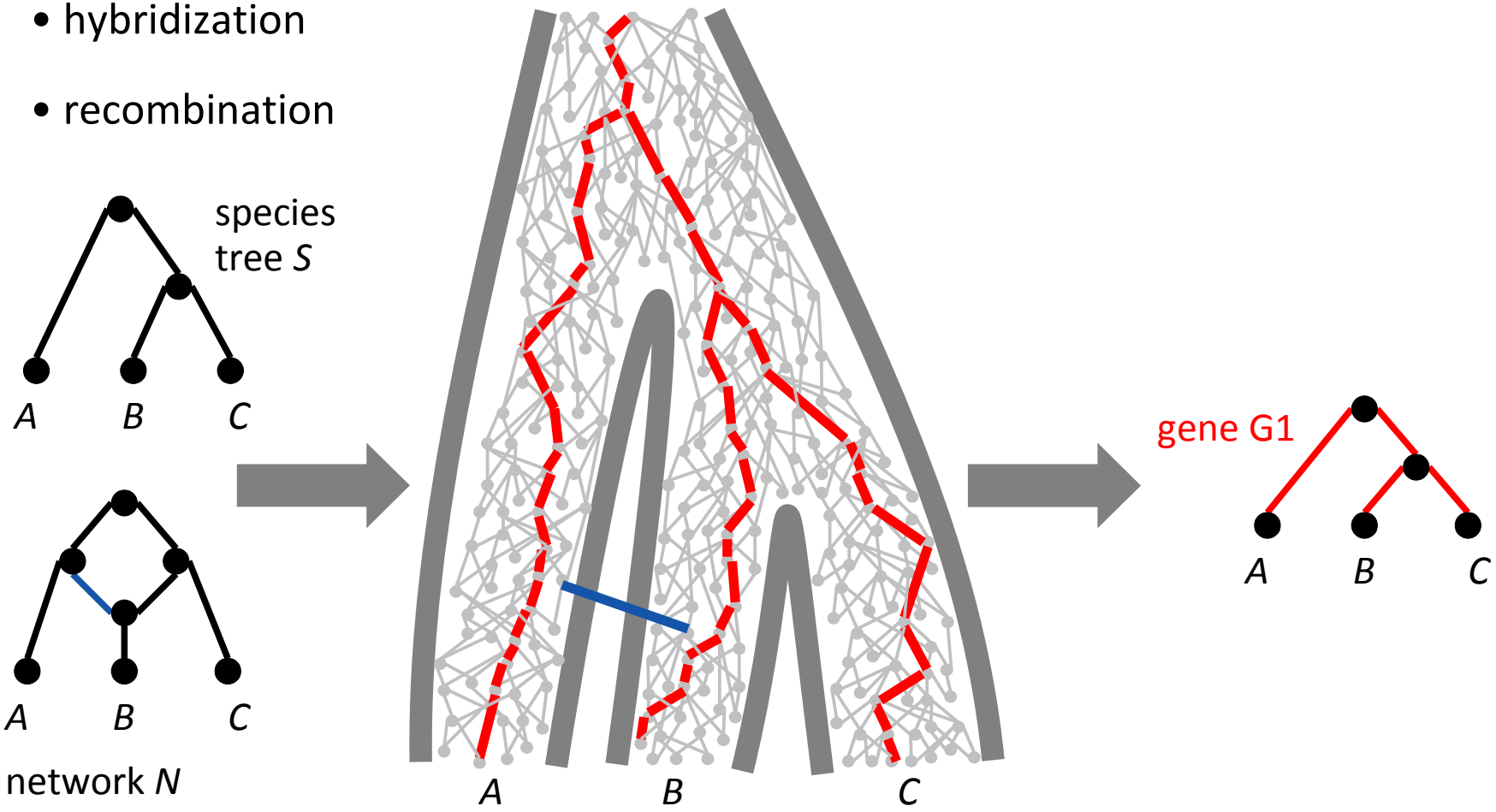
- lateral gene transfer
- hybridization
- recombination



Genetic material transfers

Transfers of genetic material between coexisting species:

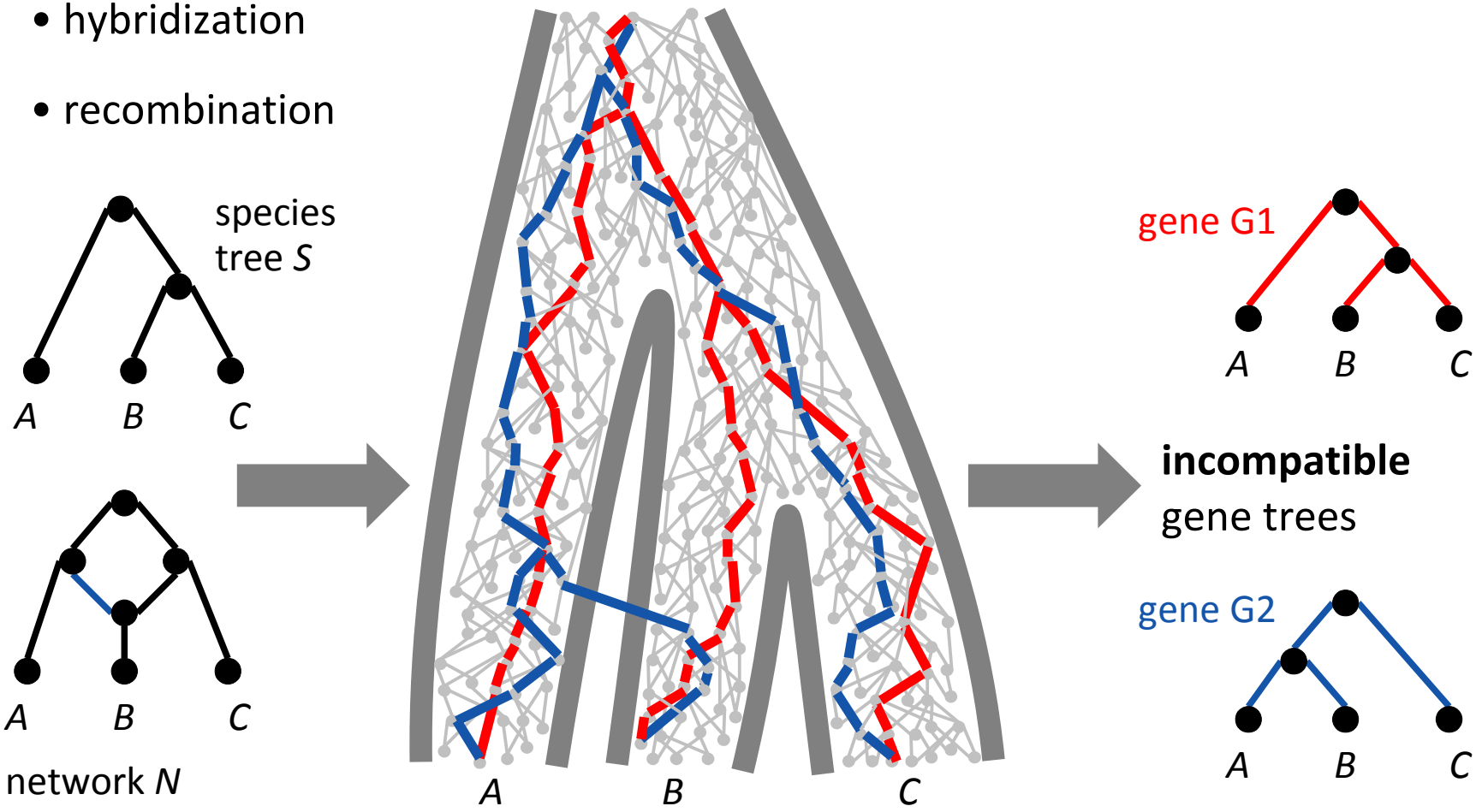
- lateral gene transfer
- hybridization
- recombination



Genetic material transfers

Transfers of genetic material between coexisting species:

- lateral gene transfer
- hybridization
- recombination

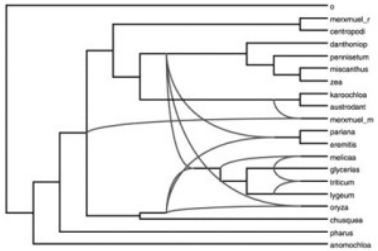


Phylogenetic networks

Phylogenetic network: network representing evolution data

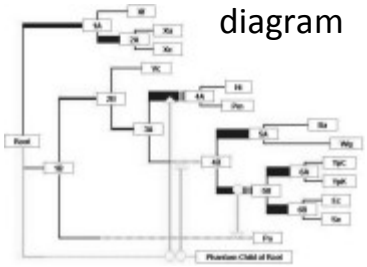
- **explicit** phylogenetic networks

model evolution



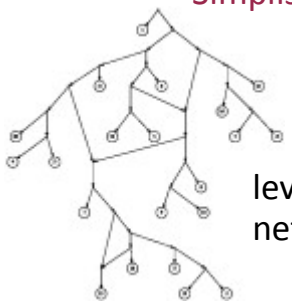
galled network

Dendroscope



synthesis diagram

HorizStory



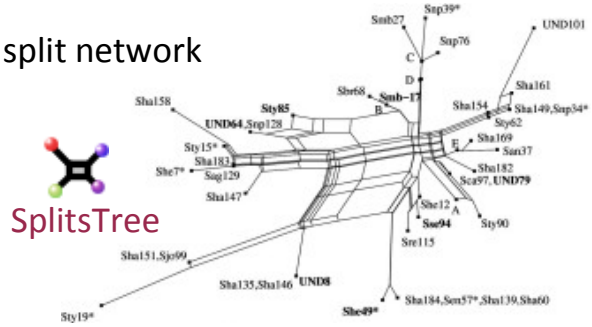
Simplistic

level-2 network

- **abstract** phylogenetic networks

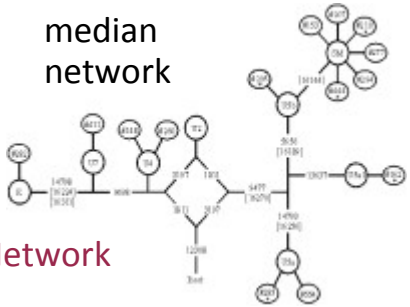
classify, visualize data

split network



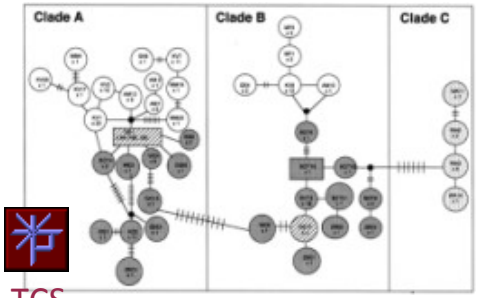
SplitsTree

median network



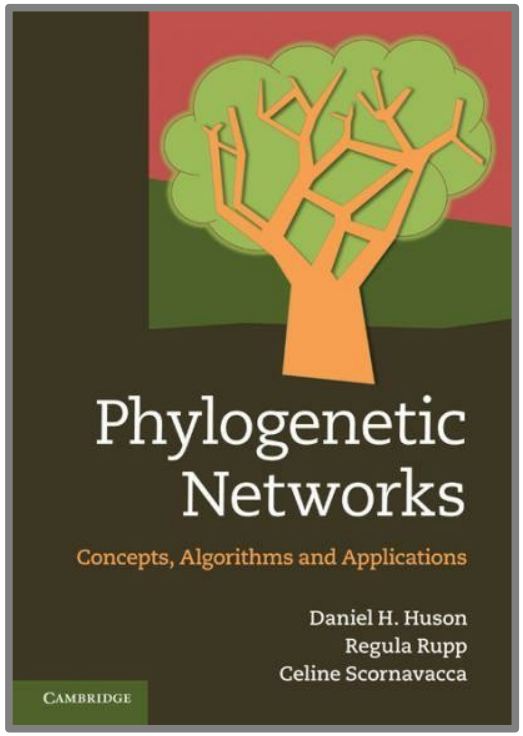
Network

minimum spanning network



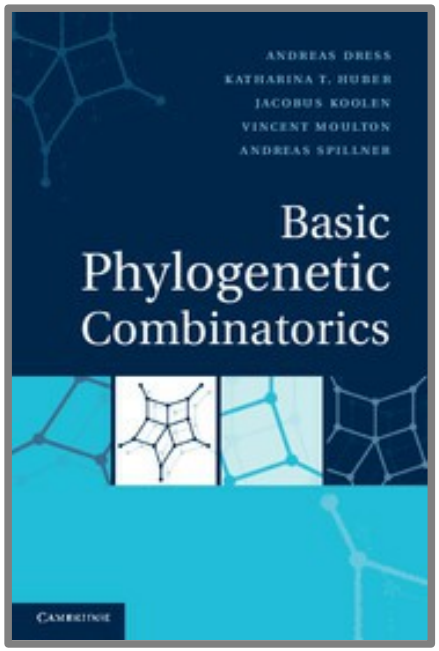
TCS

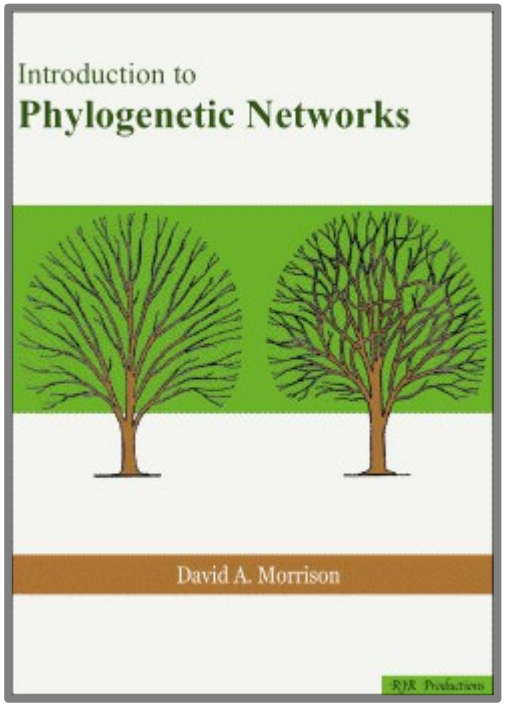
Books about phylogenetic networks



Huson, Rupp, Scornavacca, 2011

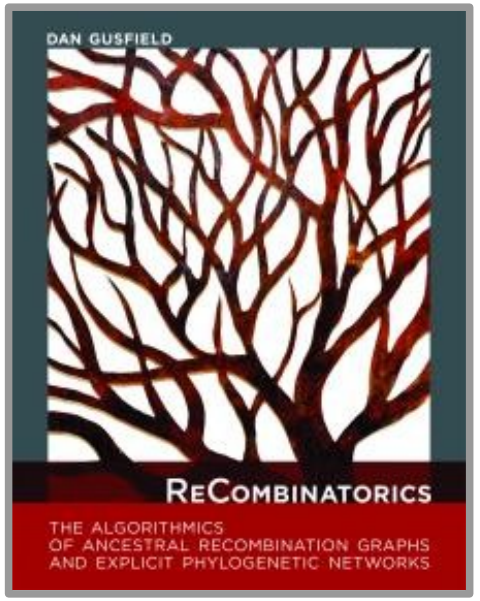
Dress, Huber, Koolen, Moulton, Spillner, 2012





Morrison, 2011

Gusfield, 2014



Workshops about phylogenetic networks

The Future of Phylogenetic Networks, 15-19 October **2012**, Lorentz Center, Leiden, The Netherlands

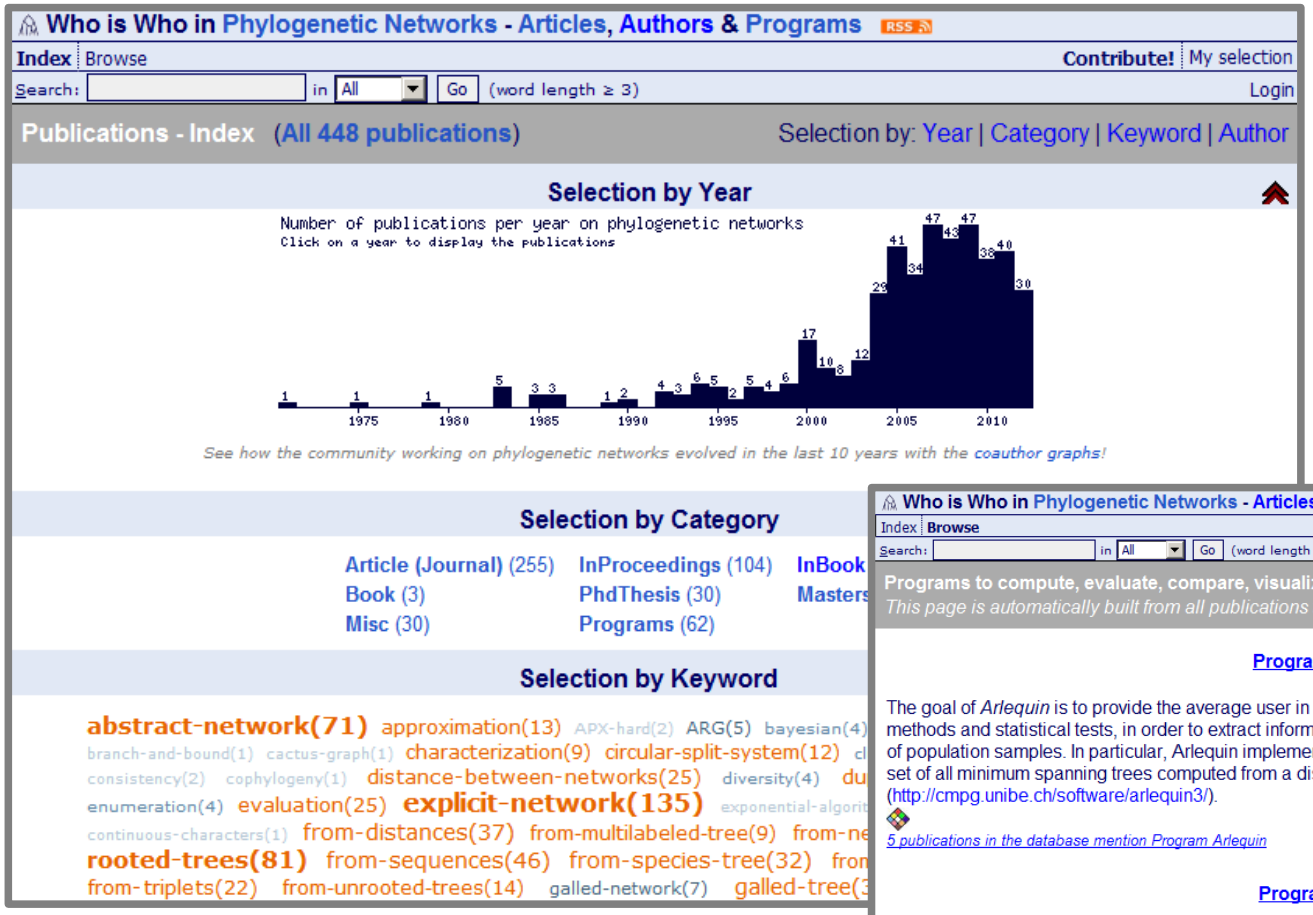
Utilizing Genealogical Phylogenetic Networks in Evolutionary Biology: Touching the Data, 7-11 July **2014**, Lorentz Center, Leiden, The Netherlands

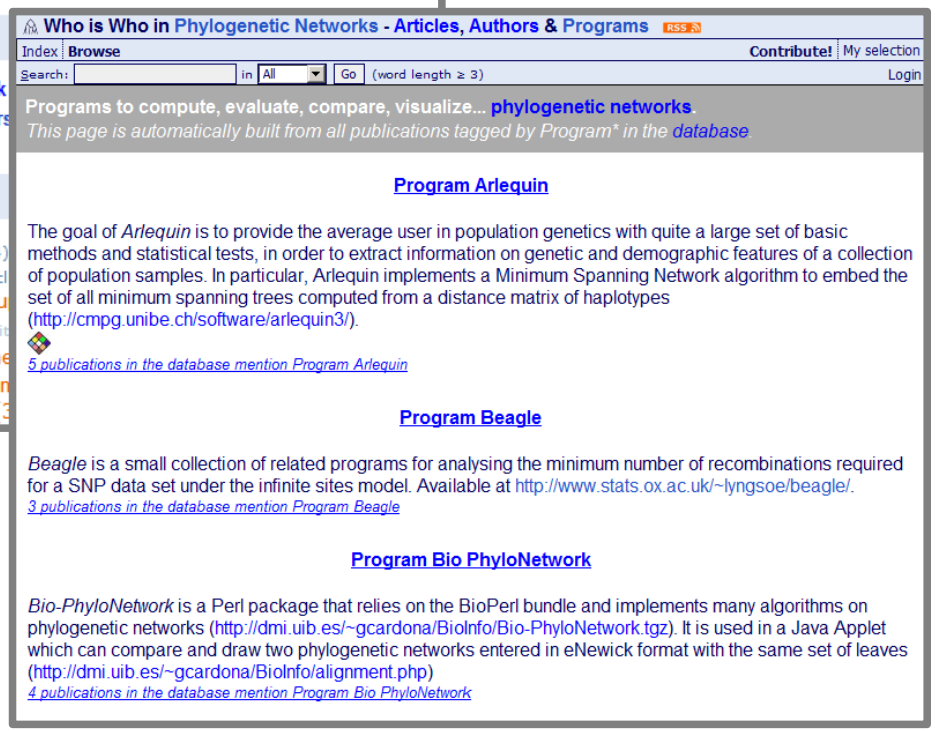
The Phylogenetic Network Workshop, 27 - 31 Jul **2015**, Institute for Mathematical Science (National University of Singapore)

Outline

- Phylogenetic networks
- Who is who in phylogenetic networks?
- Classes of phylogenetic networks
- The Tree Contaiment Problem
- A quadratic-time algorithm on quasi-stable networks

Who is who in Phylogenetic Networks?





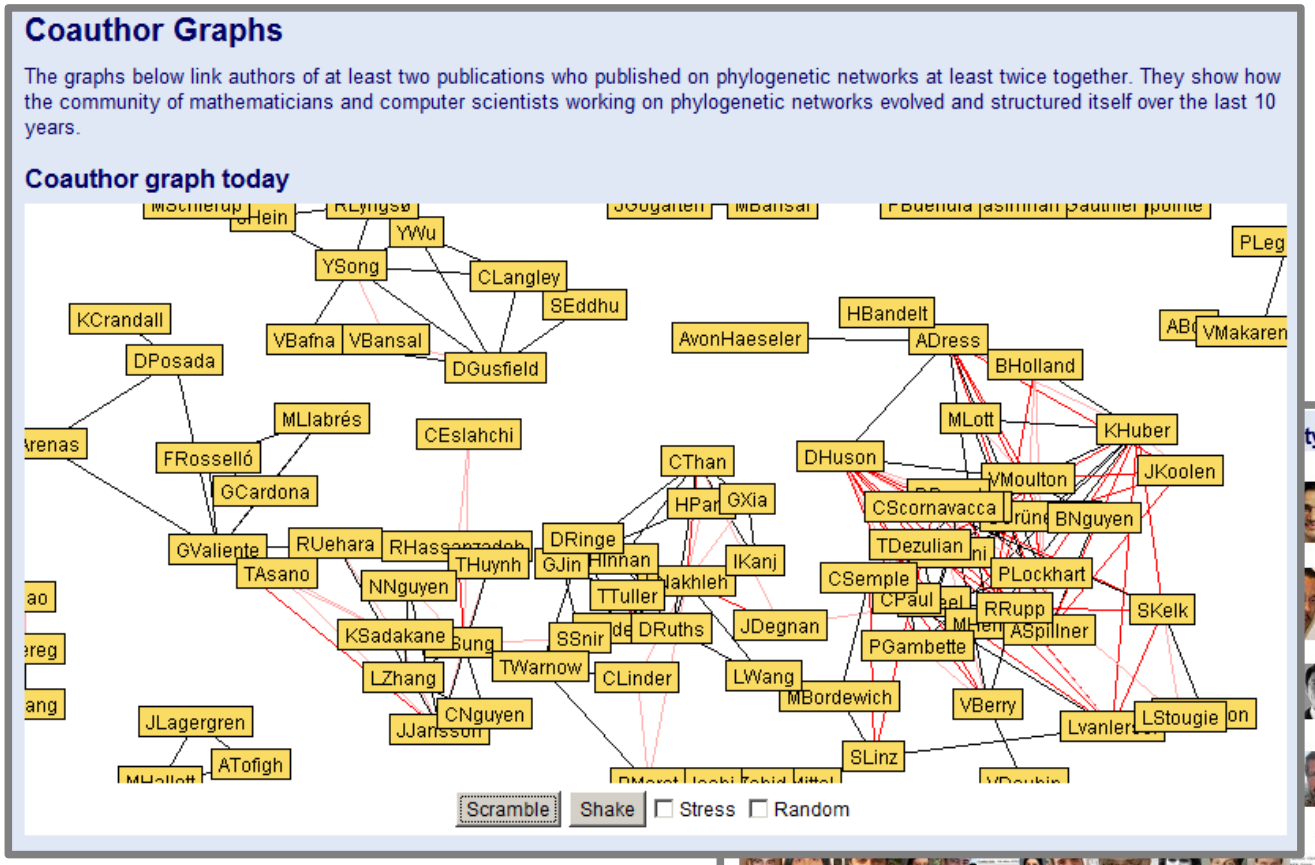
Who is Who in Phylogenetic Networks, **Articles**, Authors & **Programs**

Based on BibAdmin by Sergiu Chelcea
+ tag clouds, date histograms, journal lists, keyword definitions, co-author graphs

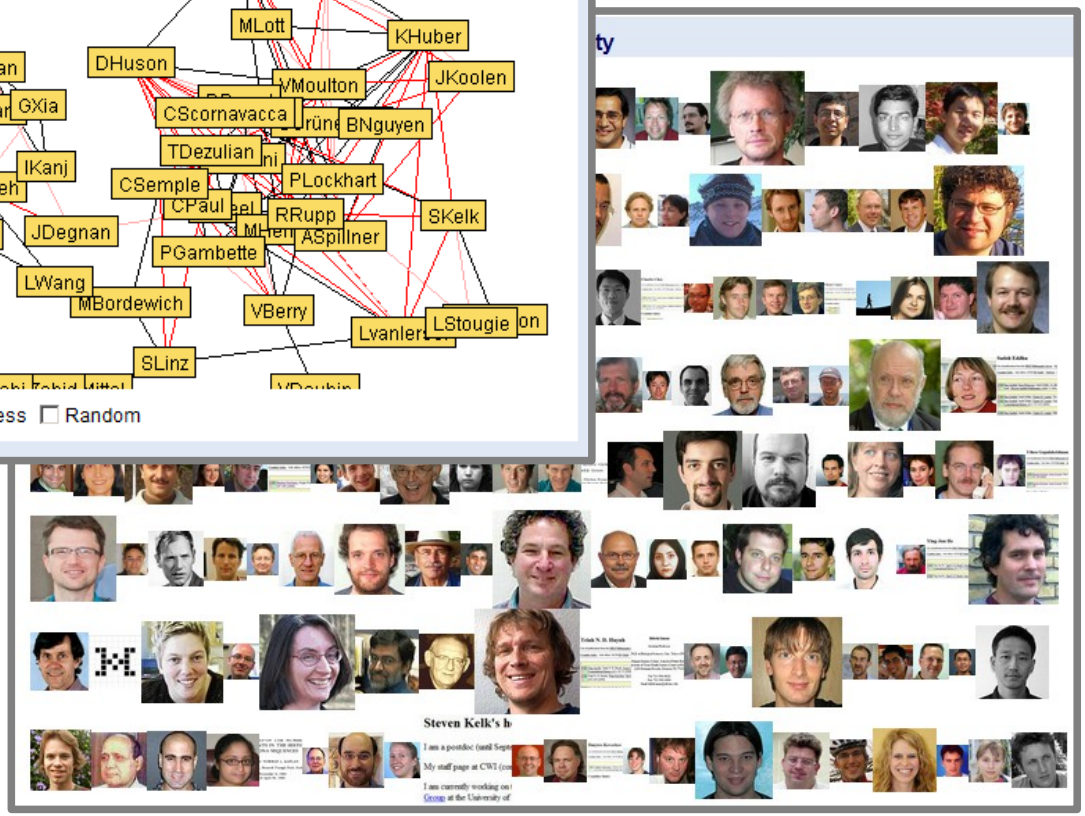
Analysis of the co-author and keyword graphs: internship of **Tushar Agarwal**

http://phylnet.univ-mlv.fr

Who is who in Phylogenetic Networks?



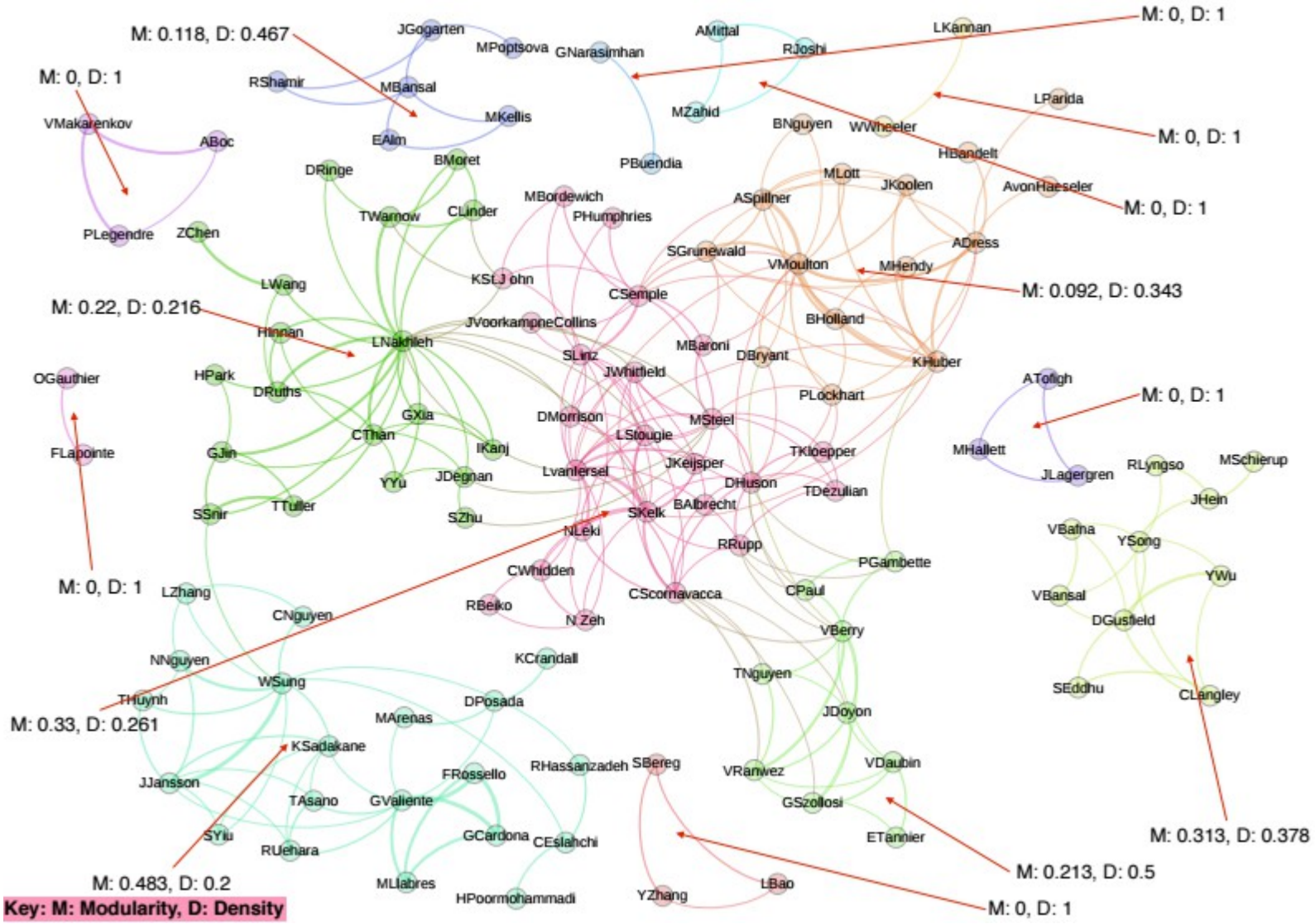
Who is Who in Phylogenetic Networks, Articles, **Authors** & Programs



Analysis of the co-author and keyword graphs: internship of **Tushar Agarwal**

http://phylnet.univ-mlv.fr

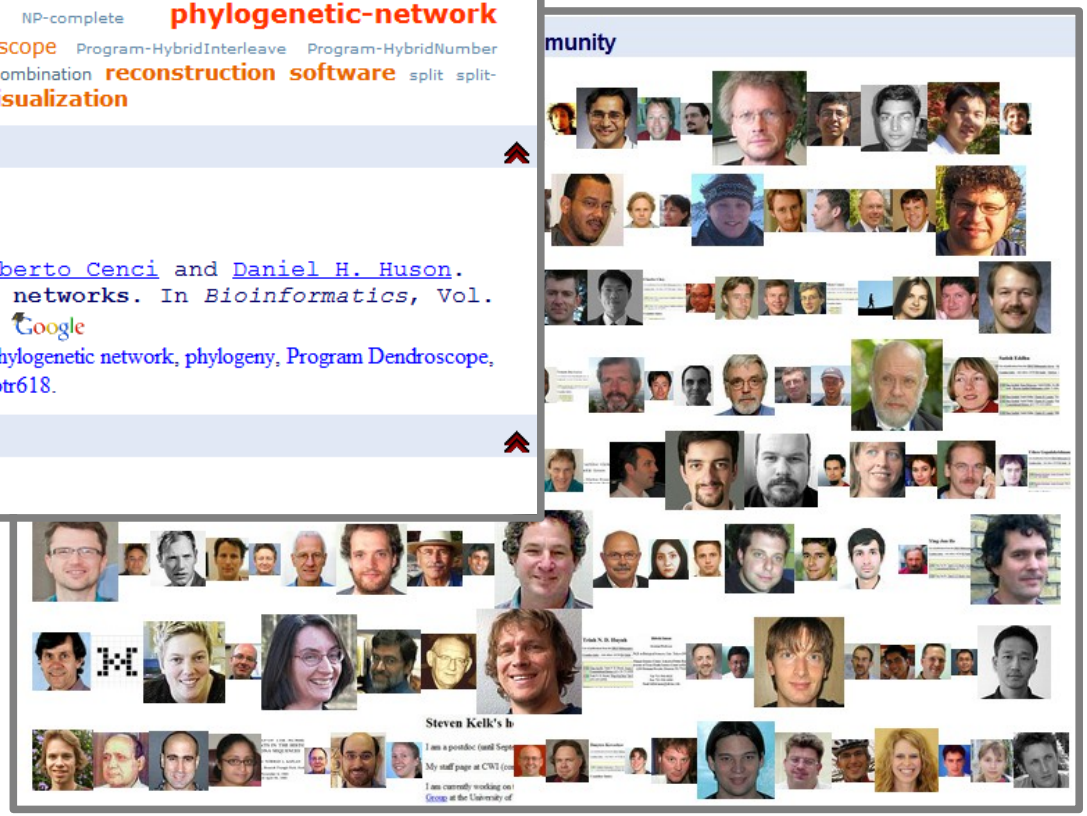
Who is who in Phylogenetic Networks?



Gephi visualization of the coauthor network of main authors by Tushar

Who is who in Phylogenetic Networks?

Who is Who in Phylogenetic Networks, **Articles**, **Authors** & Programs

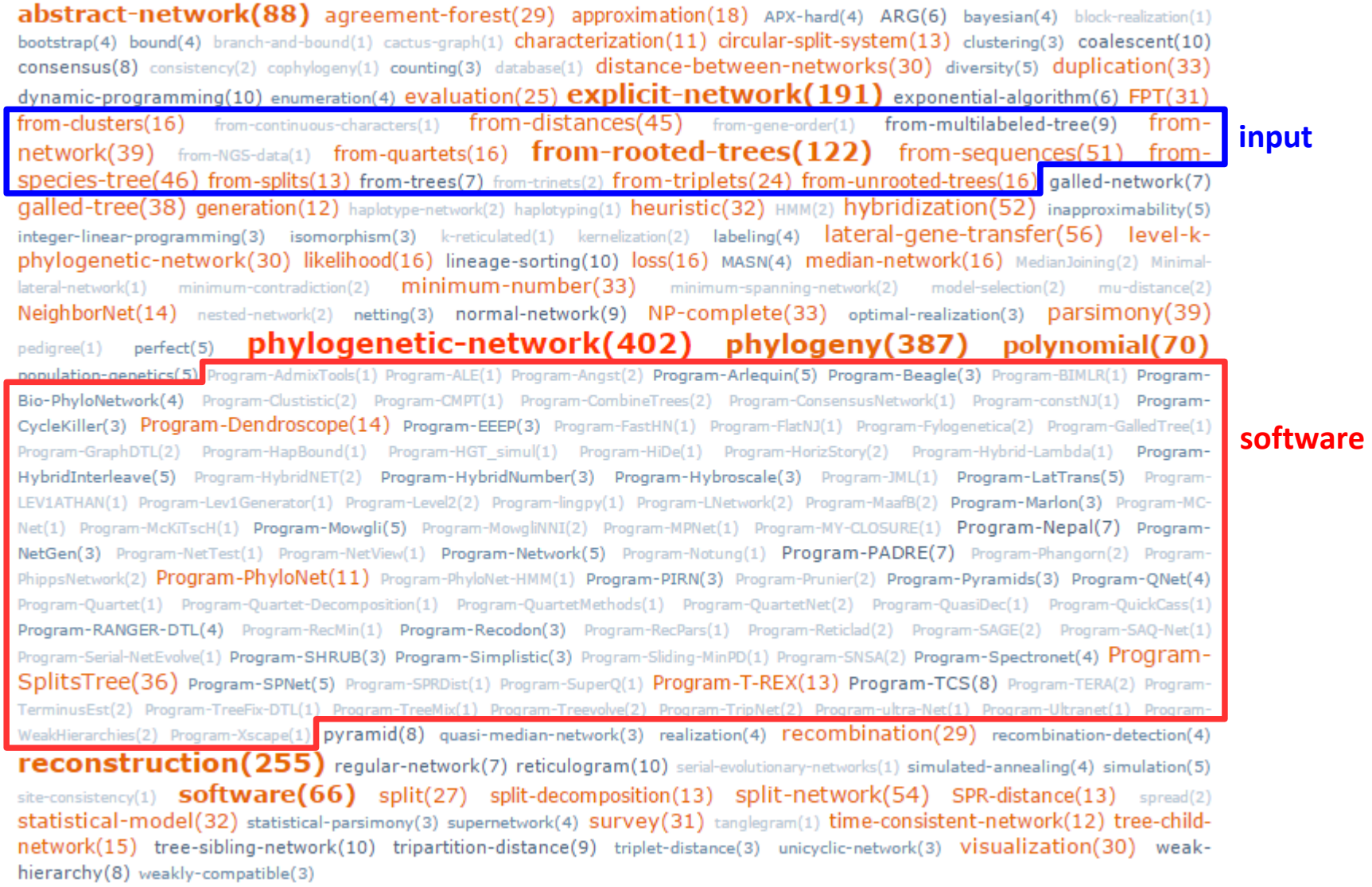


Basé sur BibAdmin
par Sergiu Chelcea
+ nuages de mots, histogramme des dates, liste des journaux, graphes de co-auteurs, définition des mots-clés.

http://phylnet.univ-mlv.fr

Who is who in Phylogenetic Networks?

Who is who in Phylogenetic Networks?

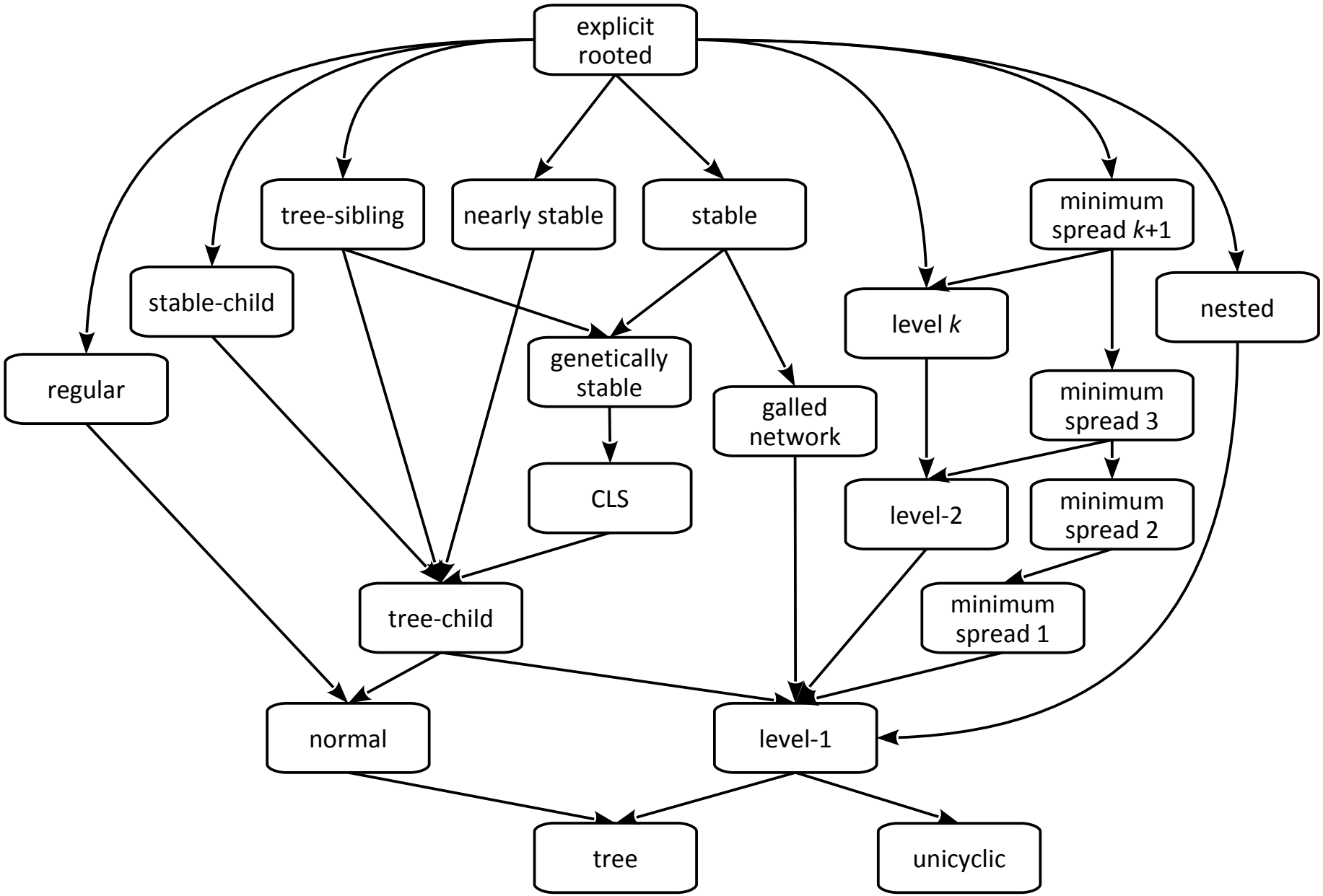


Who is who in Phylogenetic Networks?

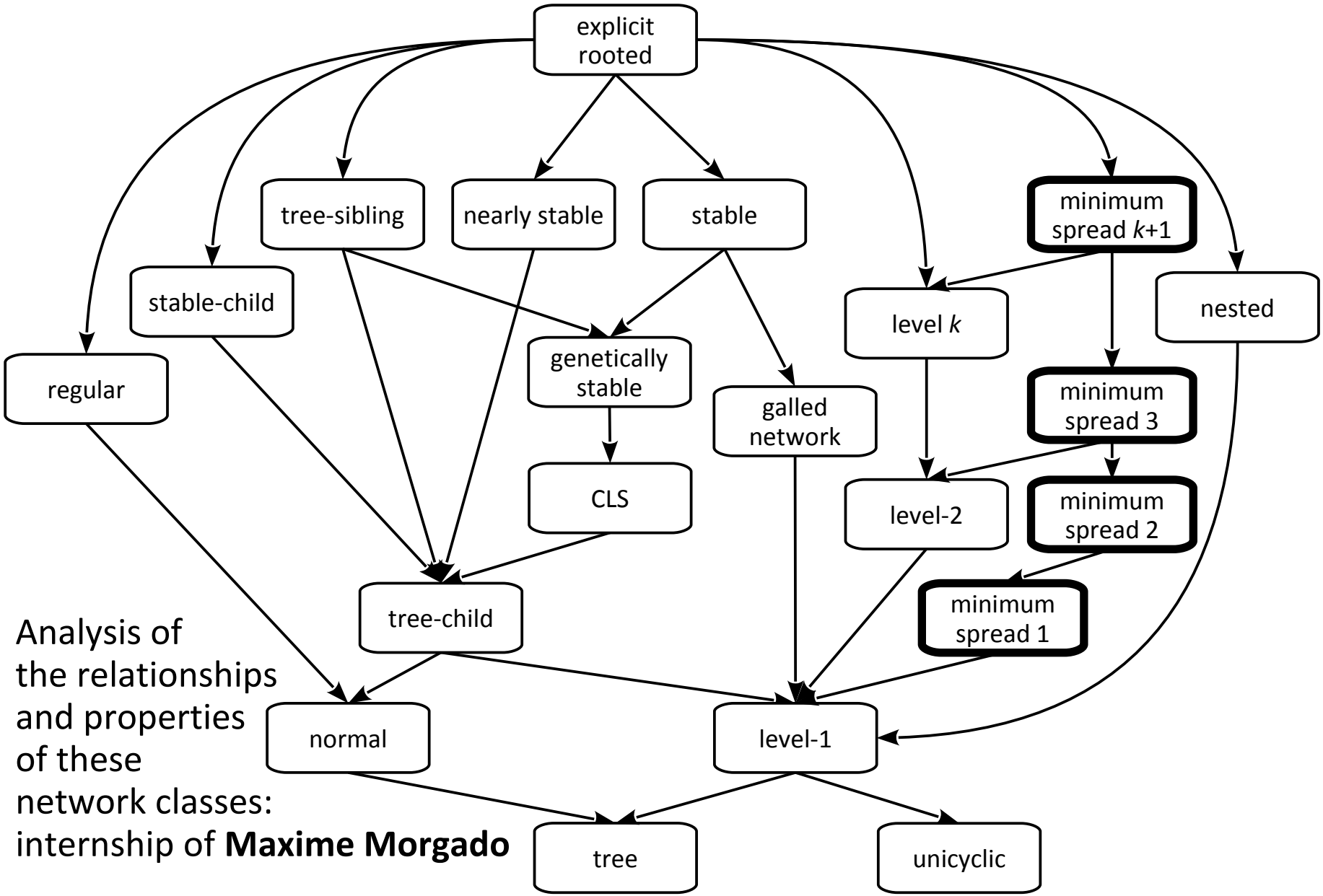
Outline

- Phylogenetic networks
- Who is who in phylogenetic networks?
- Classes of phylogenetic networks
- The Tree Contaiment Problem
- A quadratic-time algorithm on quasi-stable networks

Classes of Phylogenetic Networks



Classes of Phylogenetic Networks



Analysis of the relationships and properties of these network classes: internship of **Maxime Morgado**

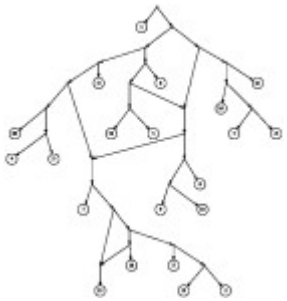
Outline

- Phylogenetic networks
- Who is who in phylogenetic networks?
- Classes of phylogenetic networks
- The Tree Containment Problem
- A quadratic-time algorithm on quasi-stable networks

Phylogenetic network reconstruction

```
espèce 1 : AATTGCAG TAGCCCAAAAT
espèce 2 : ACCTGCAG TAGACCAAT
espèce 3 : GCTTGCCG TAGACAAGAAT
espèce 4 : ATTTGCAG AAGACCAAAT
espèce 5 :          TAGACAAGAAT
espèce 6 : ACTTGCAG TAGCACAAAAT
espèce 7 : ACCTGGTG TAAAAT
```

G1 G2



{gene sequences}

distance methods

Bandelt & Dress 1992 - Legendre & Makarenkov 2000 - Bryant & Moulton 2002 - Chan, Jansson, Lam & Yiu 2006

parsimony methods

Hein 1990 - Kececioglu & Gusfield 1994 - Jin, Nakhleh, Snir, Tuller 2009 - Park, Jin & Nakhleh 2010 - Kannan & Wheeler, 2012

likelihood methods

Snir & Tuller 2009 - Jin, Nakhleh, Snir, Tuller 2009 - Velasco & Sober 2009 - Meng & Kubatko 2009

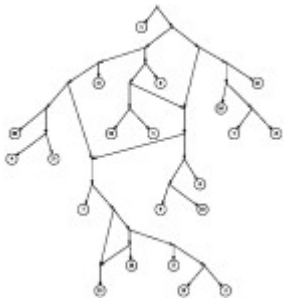
network *N*

Phylogenetic network reconstruction

Problem: methods are usually slow,
especially with rapidly increasing sequence length.

```
espèce 1 : AATTGCAG TAGCCCAAAAT
espèce 2 : ACCTGCAG TAGACCAAT
espèce 3 : GCTTGCCG TAGACAAGAAT
espèce 4 : ATTTGCAG AAGACCAAAT
espèce 5 :          TAGACAAGAAT
espèce 6 : ACTTGCAG TAGCACAAAAT
espèce 7 : ACCTGGTG TAAAAT
```

G1 G2



{gene sequences}

distance methods

Bandelt & Dress 1992 - Legendre & Makarenkov 2000 - Bryant & Moulton 2002 - Chan, Jansson, Lam & Yiu 2006

parsimony methods

Hein 1990 - Kececioglu & Gusfield 1994 - Jin, Nakhleh, Snir, Tuller 2009 - Park, Jin & Nakhleh 2010 - Kannan & Wheeler, 2012

likelihood methods

Snir & Tuller 2009 - Jin, Nakhleh, Snir, Tuller 2009 - Velasco & Sober 2009 - Meng & Kubatko 2009

network *N*

Phylogenetic network reconstruction

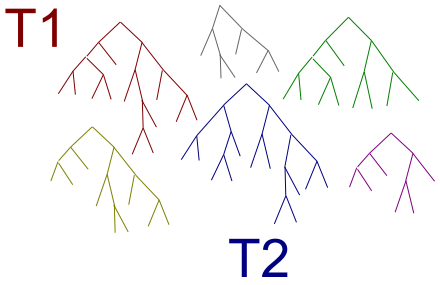
```
espèce 1 : AATTGCAG TAGCCCAAAAT
espèce 2 : ACCTGCAG TAGACCAAT
espèce 3 : GCTTGCCG TAGACAAGAAT
espèce 4 : ATTTGCAG AAGACCAAAT
espèce 5 :          TAGACAAGAAT
espèce 6 : ACTTGCAG TAGCACAAAAT
espèce 7 : ACCTGGTG TAAAAT
```

G1 G2

{gene sequences}

Reconstruction of a tree for each gene present in several species

Guindon & Gascuel, SB, 2003



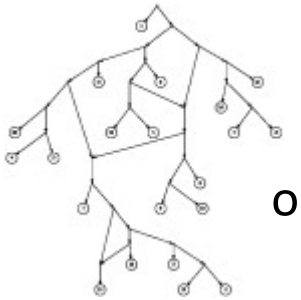
{trees}

HOGENOM Database

Dufayard, Duret, Penel, Gouy, Rechenmann & Perrière, BioInf, 2005

Tree reconciliation or consensus

explicit network



optimal super-network *N*:

- contains the input trees
- has the smallest number of reticulations

http://doua.prabi.fr/databases/hogenom/home.php?contents=hogenom4

Phylogenetic network reconstruction

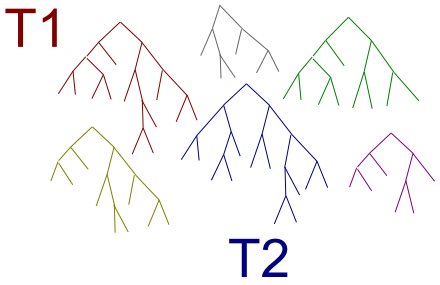
```
espèce 1 : AATTGCAG TAGCCCAAAAT
espèce 2 : ACCTGCAG TAGACCAAT
espèce 3 : GCTTGCCG TAGACAAGAAT
espèce 4 : ATTTGCAG AAGACCAAAT
espèce 5 :          TAGACAAGAAT
espèce 6 : ACTTGCAG TAGCACAAAAT
espèce 7 : ACCTGGTG TAAAAT
```

G1 G2

{gene sequences}

Reconstruction of a tree for each gene present in several species

Guindon & Gascuel, SB, 2003



{trees}

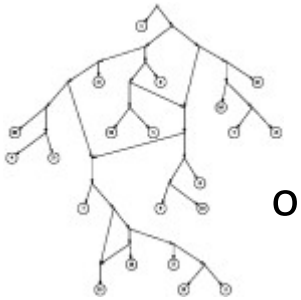
HOGENOM Database

Dufayard, Duret, Penel, Gouy, Rechenmann & Perrière, BioInf, 2005

1470 species, >290 000 trees

Tree reconciliation or consensus

explicit network



optimal super-network *N*:

- contains the input trees
- has the smallest number of reticulations

http://doua.prabi.fr/databases/hogenom/home.php?contents=hogenom4

Phylogenetic network reconstruction

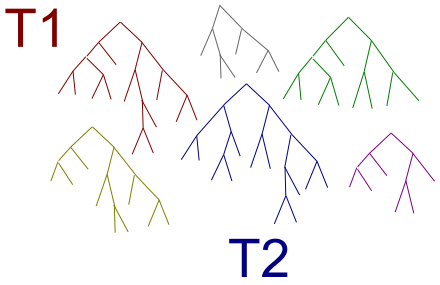
```
espèce 1 : AATTGCAG TAGCCCAAAAT
espèce 2 : ACCTGCAG TAGACCAAT
espèce 3 : GCTTGCCG TAGACAAGAAT
espèce 4 : ATTTGCAG AAGACCAAAT
espèce 5 :          TAGACAAGAAT
espèce 6 : ACTTGCAG TAGCACAAAAT
espèce 7 : ACCTGGTG TAAAAT
```

G1 G2

{gene sequences}

Reconstruction of a tree for each gene present in several species

Guindon & Gascuel, SB, 2003



{trees}

HOGENOM Database

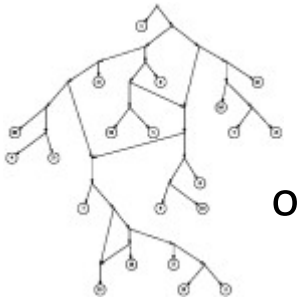
Dufayard, Duret, Penel, Gouy, Rechenmann & Perrière, BioInf, 2005

1470 species, >290 000 trees

Tree reconciliation or consensus

Tree Containment Problem

explicit network



optimal super-network *N*:

- contains the input trees
- has the smallest number of reticulations

http://doua.prabi.fr/databases/hogenom/home.php?contents=hogenom4

The Tree Containment Problem (T.C.P.)

Input: A binary phylogenetic network N and a tree T over the same set of taxa.

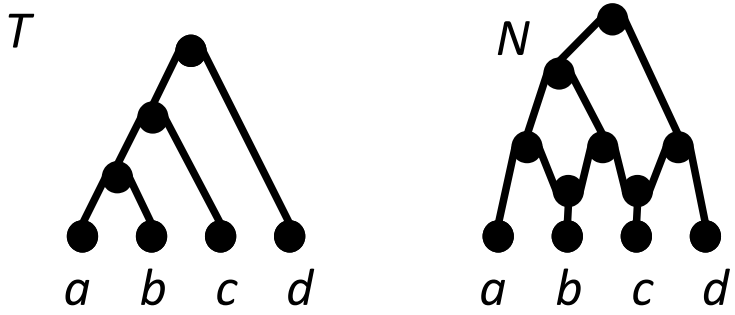
Question: Does N display T?

The Tree Containment Problem (T.C.P.)

Input: A binary phylogenetic network *N* and a tree *T* over the same set of taxa.

Question: Does *N* display *T*?

→ Can we remove one incoming arc, for each vertex with >1 parent in *N*, such that the obtained tree is equivalent to *T*?

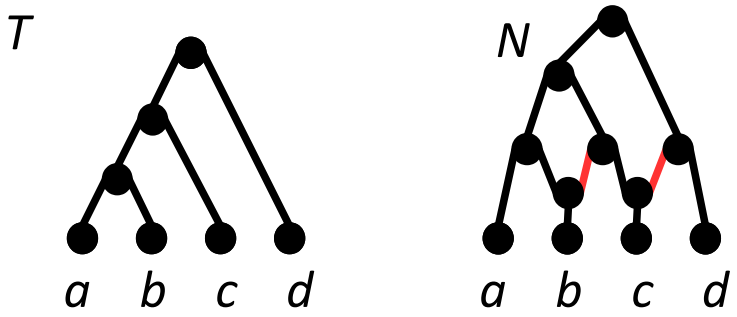


The Tree Containment Problem (T.C.P.)

Input: A binary phylogenetic network *N* and a tree *T* over the same set of taxa.

Question: Does *N* display *T*?

→ Can we **remove one incoming arc**, for each vertex with >1 parent in *N*, such that the obtained tree is equivalent to *T*?



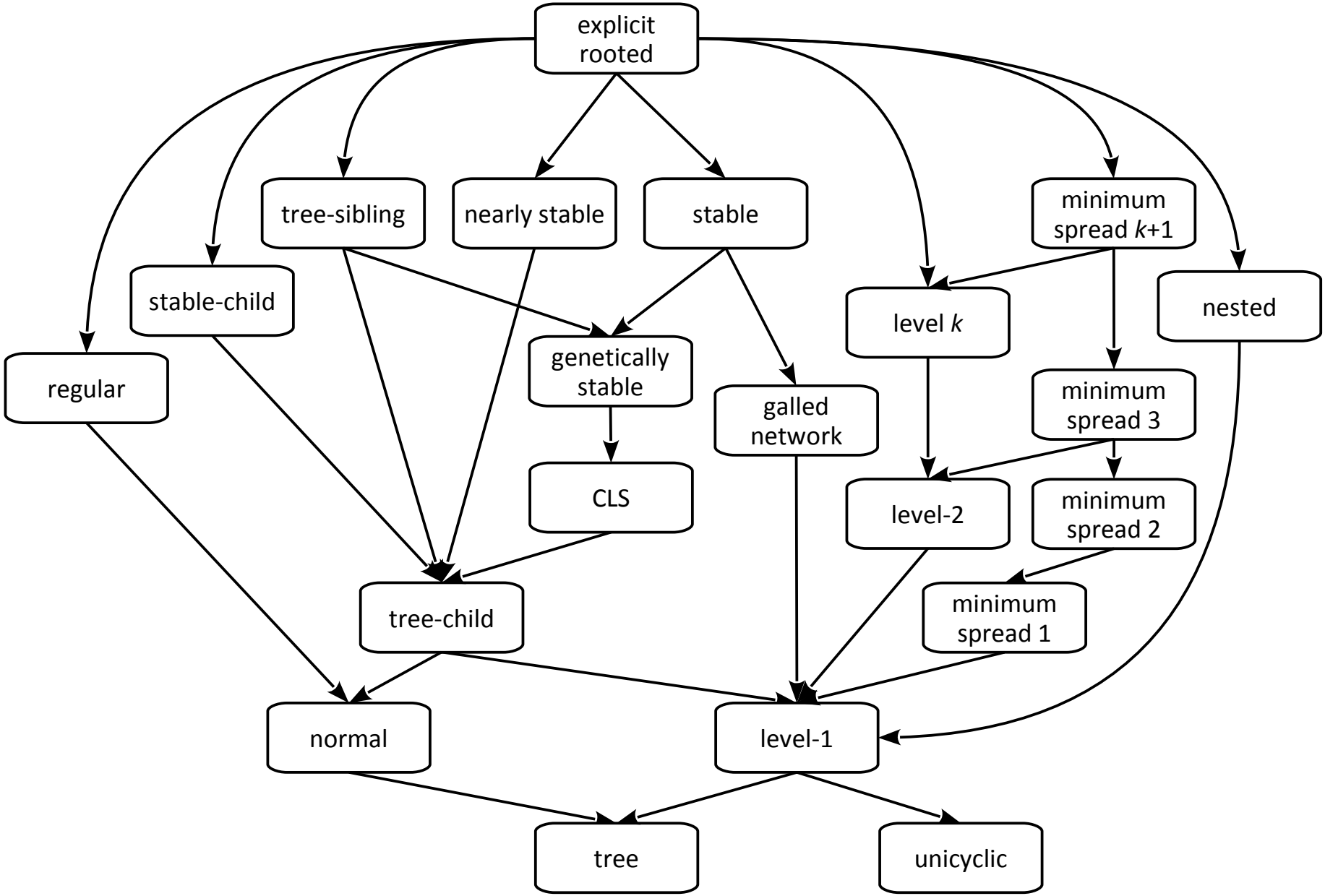
The Tree Containment Problem (T.C.P.)

Input: A binary phylogenetic network *N* and a tree *T* over the same set of taxa.

Question: Does *N* display *T*?

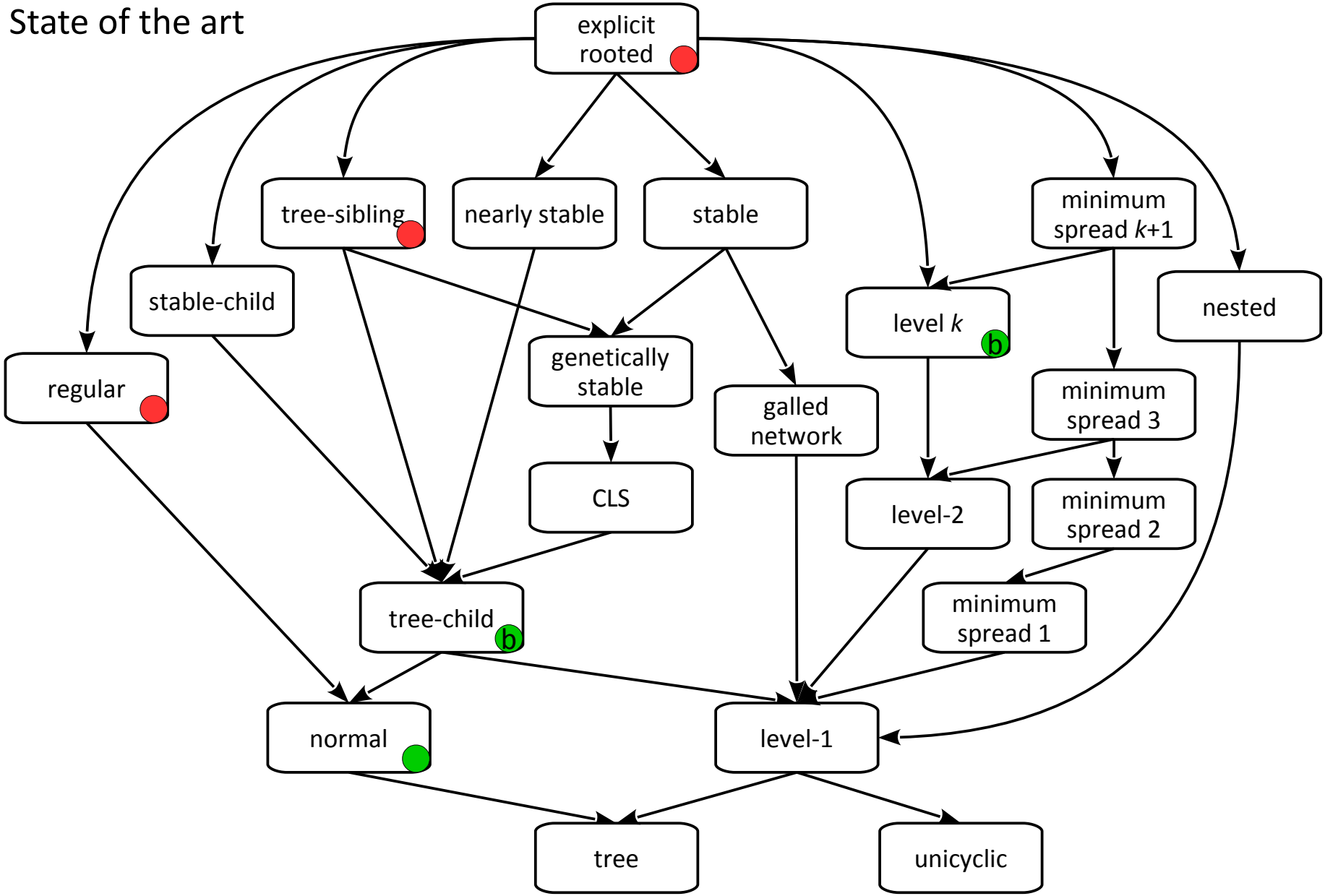
- **NP-complete** in general (Kanj, Nakhleh, Than & Xia, 2008)
- **NP-complete** for tree-sibling, time-consistent, regular networks (Iersel, Semple & Steel, 2010)
- **Polynomial-time solvable** for normal networks, for binary tree-child networks, and for level-k networks (Iersel, Semple & Steel, 2010)

Classes of Phylogenetic Networks and the T.C.P.



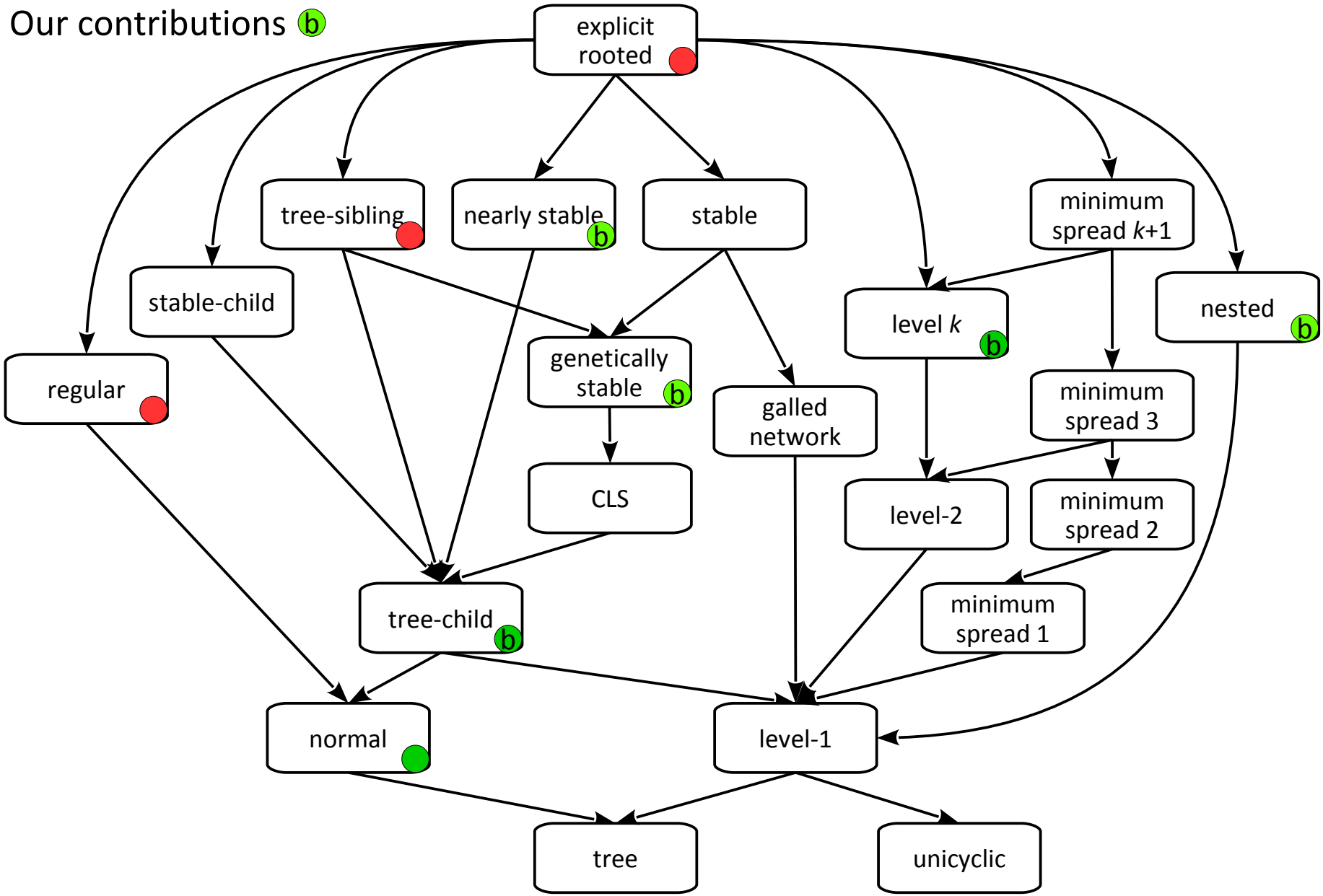
Classes of Phylogenetic Networks and the T.C.P.

State of the art



Classes of Phylogenetic Networks and the T.C.P.

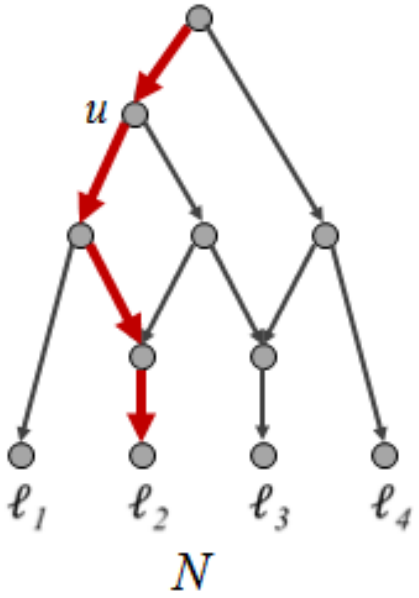
Our contributions b



Outline

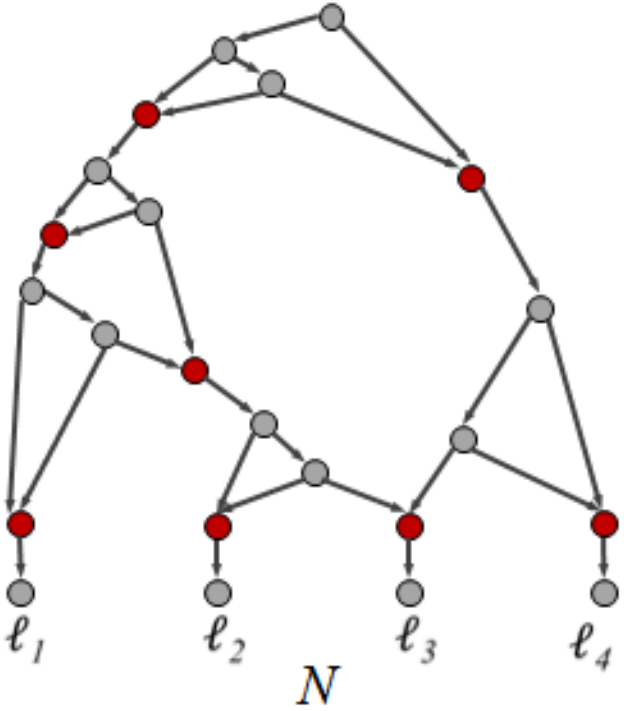
- Phylogenetic networks
- Who is who in phylogenetic networks?
- Classes of phylogenetic networks
- The Tree Contaiment Problem
- A quadratic-time algorithm on quasi-stable networks

Stable and nearly-stable networks



A vertex *u* is **stable** if there exists a leaf *l* such that all paths from the root to *l* go through *u*.

A phylogenetic network is **stable** if every reticulation vertex is stable.



A phylogenetic network is **nearly-stable** if for each vertex, either it is stable or its parents are.

Strategy to get a quadratic time algorithm for T.C.P.

Given N, a phylogenetic network with n leaves and the input tree T of the T.C.P.

Theorem 1: If N is stable then: $\#\{\text{reticulation vertices of } N\} \leq 4(n-1)$
$\#\{\text{vertices of } N\} \leq 9n$

Theorem 2: If N is nearly-stable then $\#\{\text{reticulation vertices of } N\} \leq 12(n-1)$

Theorem 3: Considering a longest path in N, and following a case analysis:
- either realize that T is not contained in N
- or build a network N' with less arcs than N such that T contained in N if and only if T contained in N'

Number of reticulations of a stable network

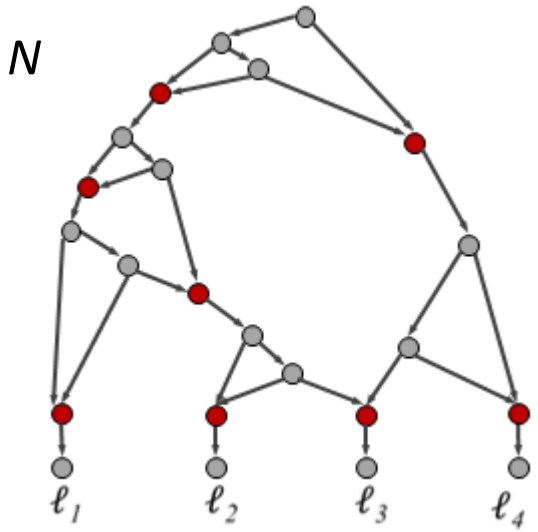
Decompose *N* into **2*n*-2 paths**:

- remove one reticulation arc per reticulation,
ensuring we get no « dummy leaf », to get a tree *T* with *n* leaves

- summarize *T* into a rooted binary tree *T'* with *n* leaves... and 2*n*-2 arcs

We can prove (technical) that:

each path contains at most 2 reticulation vertices



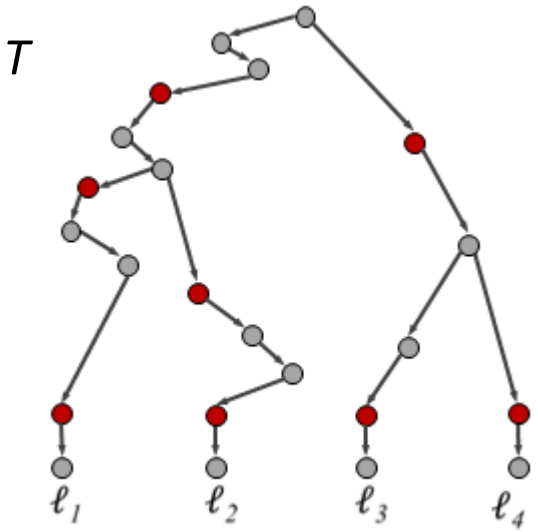
Number of reticulations of a stable network

Decompose *N* into **2*n*-2 paths**:

• remove one reticulation arc per reticulation,
ensuring we get no « dummy leaf », to get a tree *T* with *n* leaves

• summarize *T* into a rooted binary tree *T'* with *n* leaves... and 2*n*-2 arcs

We can prove (technical) that:
each path contains at most 2 reticulation vertices



Number of reticulations of a stable network

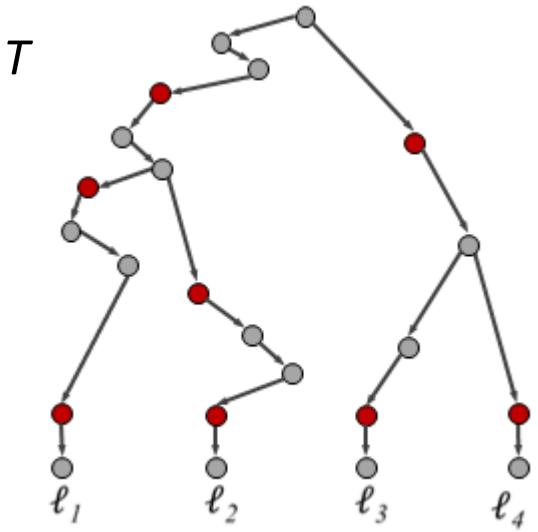
Decompose *N* into **2*n*-2 paths**:

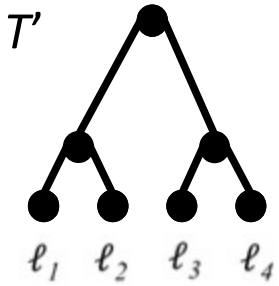
- remove one reticulation arc per reticulation,
ensuring we get no « dummy leaf », to get a tree *T* with *n* leaves

- summarize *T* into a rooted binary tree *T'* with *n* leaves... and 2*n*-2 arcs

We can prove (technical) that:

each path contains at most 2 reticulation vertices





Number of reticulations of a stable network

Decompose *N* into **2*n*-2 paths**:

• remove one reticulation arc per reticulation,
ensuring we get no « dummy leaf », to get a tree *T* with *n* leaves

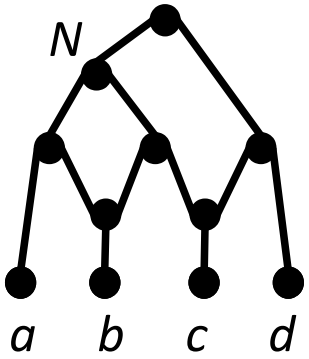
• summarize *T* into a rooted binary tree *T'* with *n* leaves... and 2*n*-2 arcs

We can prove (technical) that:
each path contains at most 2 reticulation vertices

→ *N* contains at most 4(*n*-1) reticulation vertices

Number of reticulations of a stable network

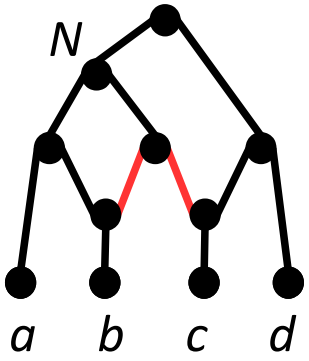
« Dummy leaves »?



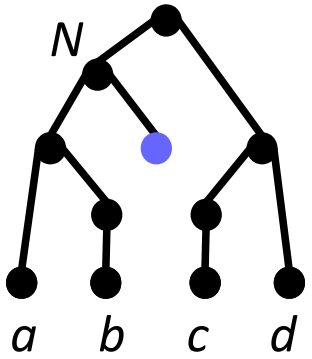
Deleting reticulation arcs can create « **dummy leaves** »

Number of reticulations of a stable network

« Dummy leaves »?



Deleting reticulation arcs can create « **dummy leaves** »



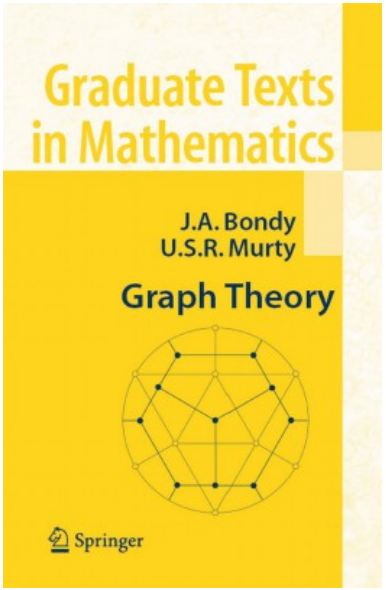
Number of reticulations of a stable network

Possible to avoid creating « dummy leaves »?

Number of reticulations of a stable network

Possible to avoid creating « dummy leaves »?

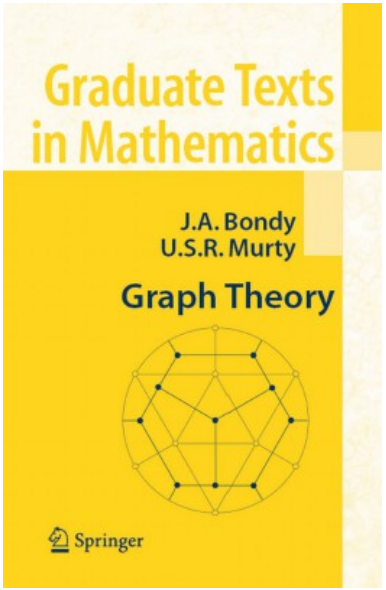
16.2.13 Let $G := G[X, Y]$ be a bipartite graph such that $d(x) \geq 1$ for all $x \in X$ and $d(x) \geq d(y)$ for all $xy \in E$, where $x \in X$ and $y \in Y$. Show that G has a matching covering every vertex of X. (N. Alon)



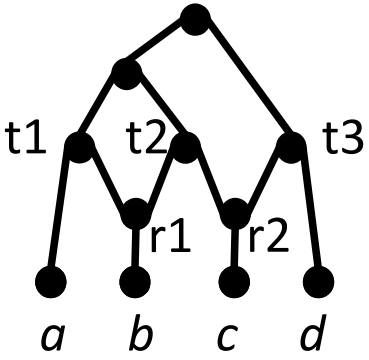
Number of reticulations of a stable network

Possible to avoid creating « dummy leaves »?

16.2.13 Let $G := G[X, Y]$ be a bipartite graph such that $d(x) \geq 1$ for all $x \in X$ and $d(x) \geq d(y)$ for all $xy \in E$, where $x \in X$ and $y \in Y$. Show that G has a matching covering every vertex of X. (N. Alon)

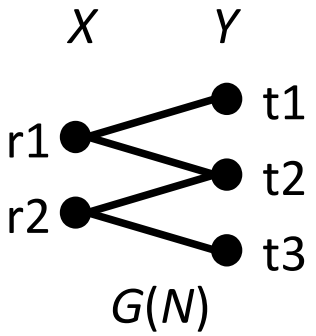


Given *N*



Build *G*(*N*), bipartite graph such that:

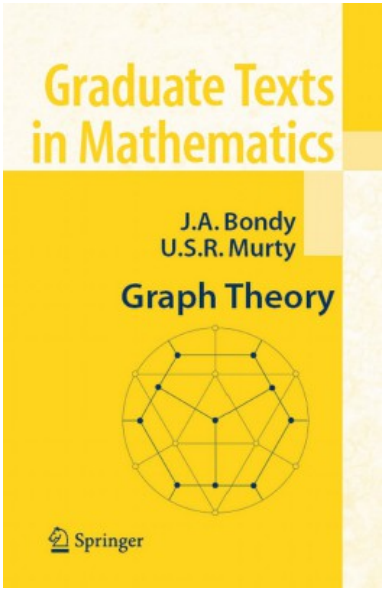
- *X* = reticulation vertices of *N*
  → all vertices in *X* have degree 2
- *Y* = tree vertices of *N* with at least one reticulation child
  → all vertices in *Y* have degree 1 or 2
- edge between *x* and *y* iff *x* is a child of *y*



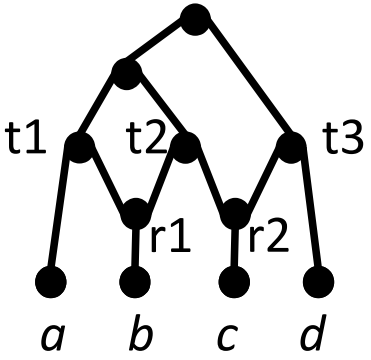
Number of reticulations of a stable network

Possible to avoid creating « dummy leaves »?

16.2.13 Let $G := G[X, Y]$ be a bipartite graph such that $d(x) \geq 1$ for all $x \in X$ and $d(x) \geq d(y)$ for all $xy \in E$, where $x \in X$ and $y \in Y$. Show that G has a matching covering every vertex of X. (N. Alon)

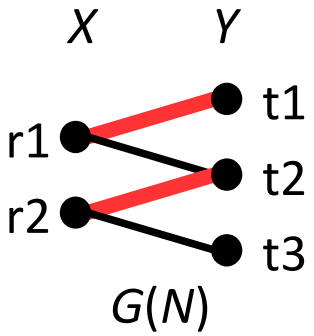


Given *N*



Build *G*(*N*), bipartite graph such that:

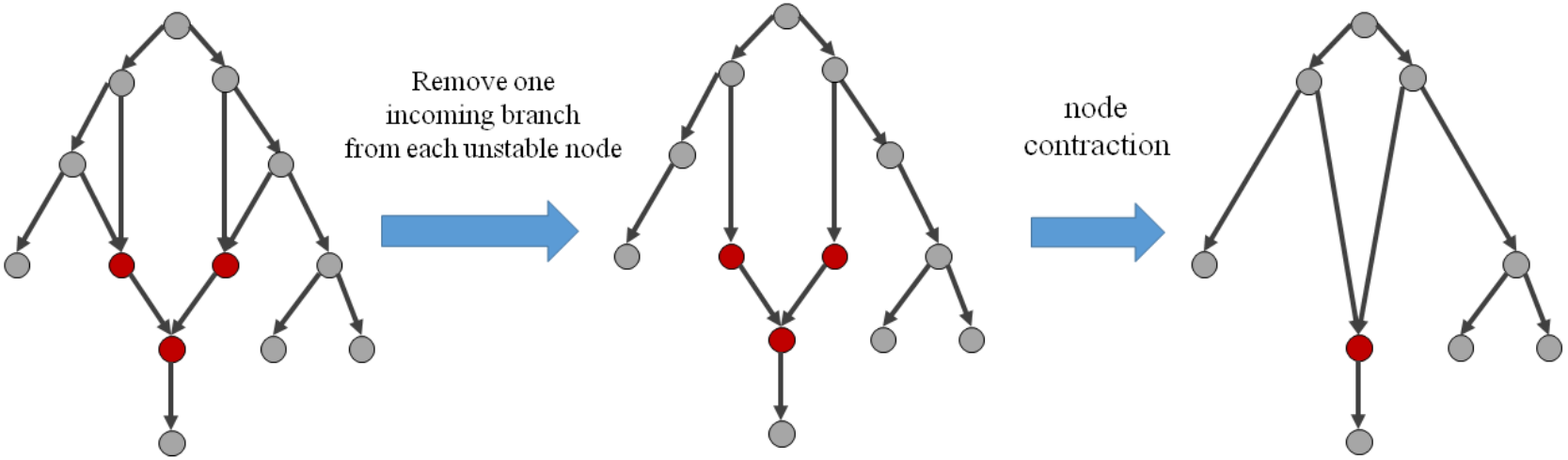
- *X* = reticulation vertices of *N*
  → all vertices in *X* have degree 2
- *Y* = tree vertices of *N* with at least one reticulation child
  → all vertices in *Y* have degree 1 or 2
- edge between *x* and *y* iff *x* is a child of *y*



→ **matching covering every vertex of *X* → edges to remove from *N***

Number of reticulations of a nearly-stable network

Reduce nearly-stable networks to stable networks



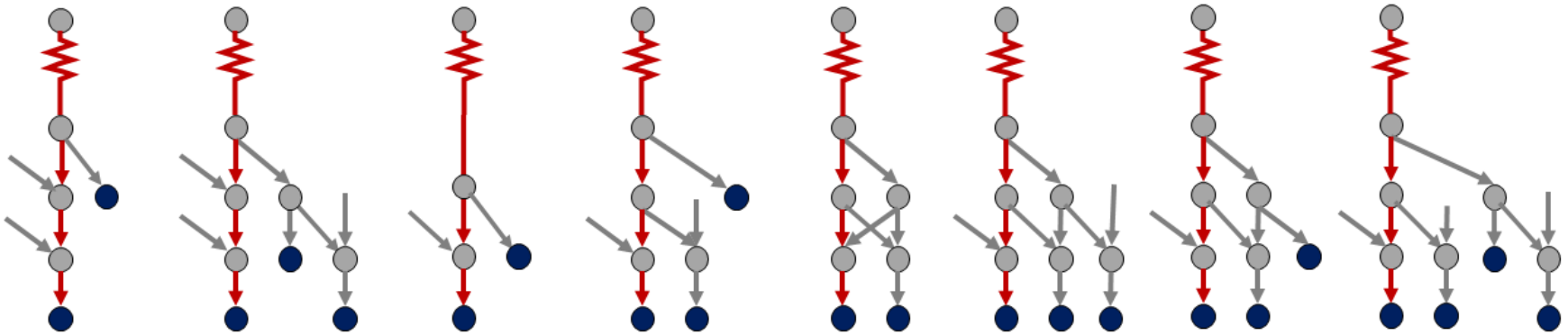
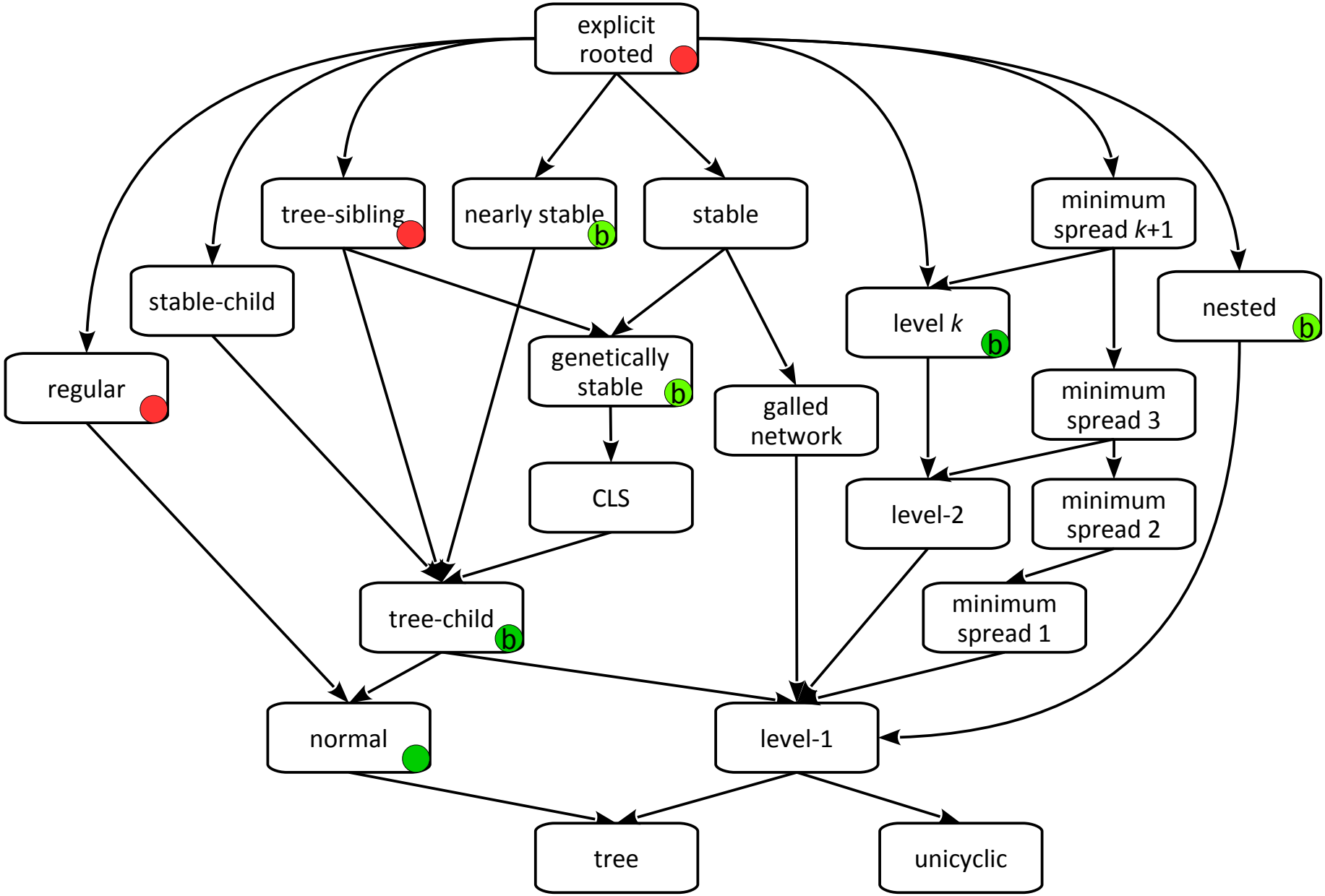
$\#UnstableRet \leq 2$ $\#StableRet \leq 8(n\text{-}1)$

$\#StableRet \leq 4(n\text{-}1)$

Deleting reticulation arcs to simplify the question

Simplify *N* by removing an edge near the end of a **longest path *P***.

Case analysis (8 cases):



Perspectives...