RECONSTRUCTING PHYLOGENIES FROM ORDINAL DISTANCE INFORMATION

Manuel Lafond, Université de Sherbrooke, Canada

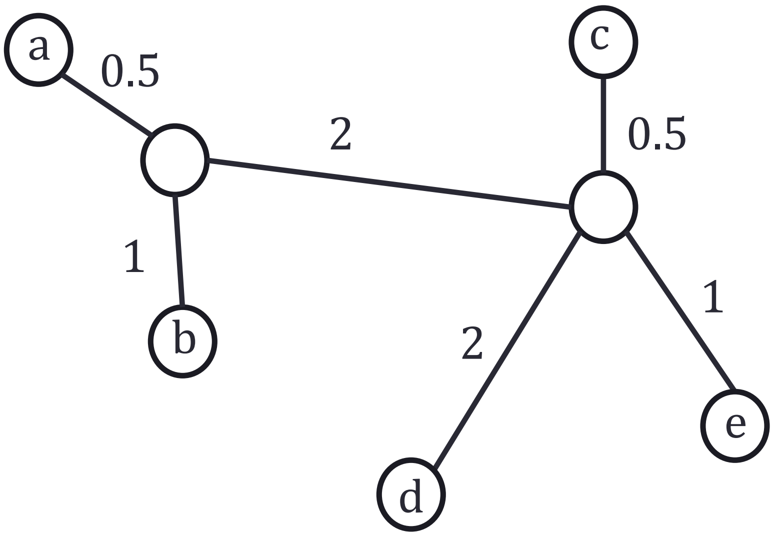
UNIVERSITÉ DE SHERBROOKE

Additive distances

	a	b	c	d	e
a	0	1.5	3	4.5	3.5
b	1.5	0	3.5	5	4
c	3	3.5	0	2.5	1.5
d	4.5	5	2.5	0	3
e	3.5	4	1.5	3	0

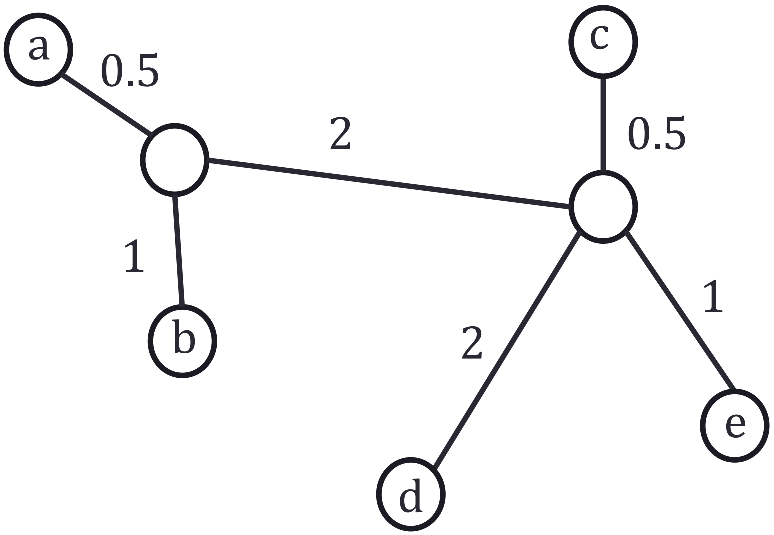
Additive distances

	a	b	c	d	e
a	0	1.5	3	4.5	3.5
b	1.5	0	3.5	5	4
c	3	3.5	0	2.5	1.5
d	4.5	5	2.5	0	3
e	3.5	4	1.5	3	0



NOT Additive

	a	b	c	d	e
a	0	1.8	2.7	3.9	3.5
b	1.8	0	4.1	5.5	4.2
c	2.7	4.1	0	2.5	1.7
d	3.9	5.5	2.5	0	3
e	3.5	4.2	1.7	3	0



Want: a tree that faithfully represents these distances.

What does "faithfully" mean?

NOT Additive

	a	b	c	d	e
a	0	1.8	2.7	3.9	3.5
b	1.8	0	4.1	5.5	4.2
c	2.7	4.1	0	2.5	1.7
d	3.9	5.5	2.5	0	3
e	3.5	4.2	1.7	3	0

The exact distance values are **unreliable**.
However, their **relative ordering** should be informative.

NOT Additive

	a	b	c	d	e
a	0	1.8	2.7	3.9	3.5
b	1.8	0	4.1	5.5	4.2
c	2.7	4.1	0	2.5	1.7
d	3.9	5.5	2.5	0	3
e	3.5	4.2	1.7	3	0

In the tree, a should have **b** as its closest taxon, **c** as its second closest, **e** third, **d** fourth

The exact distance values are **unreliable**.
However, their **relative ordering** should be informative.

NOT Additive

	a	b	c	d	e
a	0	1	2	4	3
b	1.8	0	4.1	5.5	4.2
c	2.7	4.1	0	2.5	1.7
d	3.9	5.5	2.5	0	3
e	3.5	4.2	1.7	3	0

In the tree, a should have b as its closest taxon, c as its second closest, e third, d fourth

The exact distance values are **unreliable**.
However, their **relative ordering** should be informative.

NOT Additive

	a	b	c	d	e
a	0	1	2	4	3
b	1.8	0	4.1	5.5	4.2
c	2.7	4.1	0	2.5	1.7
d	3.9	5.5	2.5	0	3
e	3.5	4.2	1.7	3	0

Replace elements of the row by their rank.
Do this for every row.

The exact distance values are **unreliable**.
However, their **relative ordering** should be informative.

Ranking matrices

	a	b	c	d	e
a	0	1	2	4	3
b	1	0	2	4	3
c	3	4	0	2	1
d	3	4	1	0	2
e	3	4	1	2	0

Replace elements of the row by their rank.
Do this for every row.

The exact distance values are **unreliable**.
However, their **relative ordering** should be informative.

Ranking matrices

	a	b	c	d	e
a	0	1	2	4	3
b	1	0	2	4	3
c	3	4	0	2	1
d	3	4	1	0	2
e	3	4	1	2	0

The Ranked Distance Phylogeny problem

Given: ranking matrix R.

Find: an edge-weighted tree T that *realizes* these rankings.

Ranking matrices

	a	b	c	d	e
a	0	1	2	4	3
b	1	0	2	4	3
c	3	4	0	2	1
d	3	4	1	0	2
e	3	4	1	2	0

The Ranked Distance Phylogeny problem

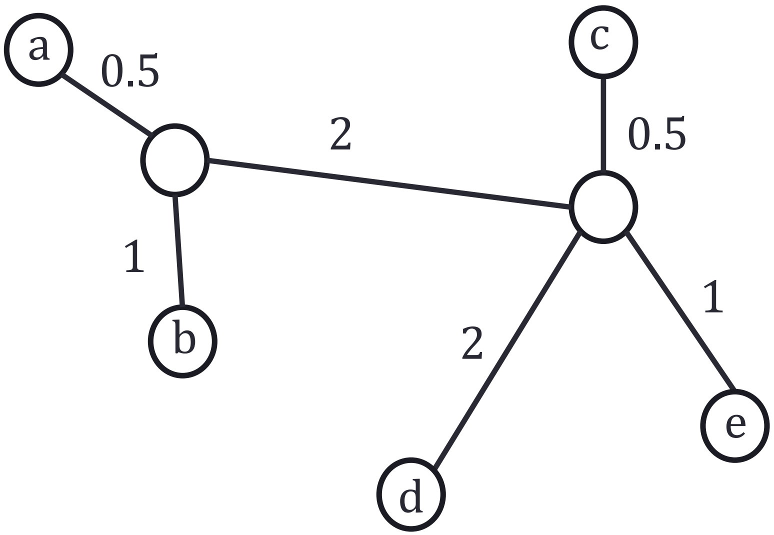
Given: ranking matrix R.

Find: an edge-weighted tree T that *realizes* these rankings.

For each x, y, z, $R[x,y] < R[x,z] \Rightarrow dist_T(x,y) < dist_T(x,z)$

Ranking matrices

	a	b	c	d	e
a	0	1	2	4	3
b	1	0	2	4	3
c	3	4	0	2	1
d	3	4	1	0	2
e	3	4	1	2	0



<u>The Ranked Distance Phylogeny problem</u>

Given: ranking matrix R.

Find: an edge-weighted tree T that *realizes* these rankings.

For each $x, y, z,\ \ R[x,y] < R[x,z] \Rightarrow dist_T(x,y) < dist_T(x,z)$

Related work

- Given distance D, compute a new distance D' where $D'(x, y) =$ # of disagreements in ranking others (inversions)
- For which D is D' tree-like?
  - [Bonnot, Guénoche, Perrier, *Ordinal and Symb. Analysis*, 1996]
  - [Guénoche, *J of Classification*, 1997]
  - [Guénoche, *Discrete Mathematics*, 1998]
  - [Moulton & Spillner, *Order*, 2022]

Related work

- Ranked Distance Phylogeny Problem
  - [Kannan & Warnow, *WADS*, 1993] (triangle total orders)
  - [Kearney, *JCB*, 1997] (mandatory splits from ranks)
  - [Kearney, *RECOMB*, 1998] (extract quartets from ranks)
  - [Kearney, Hayward, Meijer, *Algorithmica*, 1999] (total order on D)
  - [Shah & Farach-Colton, *J of Classification*, 2006] (total order is NP-hard)

Related work

- Ranked Distance Phylogeny Problem
  - [Kannan & Warnow, *WADS*, 1993] (triangle total orders)
  - [Kearney, *JCB*, 1997] (**mandatory splits** from ranks)
  - [Kearney, *RECOMB*, 1998] (extract quartets from ranks)
  - [Kearney, Hayward, Meijer, *Algorithmica*, 1999] (total order on D)
  - [Shah & Farach-Colton, *J of Classification*, 2006] (total order is NP-hard)

	a	b	c	d	e
a	0	1	2	4	3
b	1	0	2	4	3
c	3	4	0	2	1
d	3	4	1	0	2
e	3	4	1	2	0

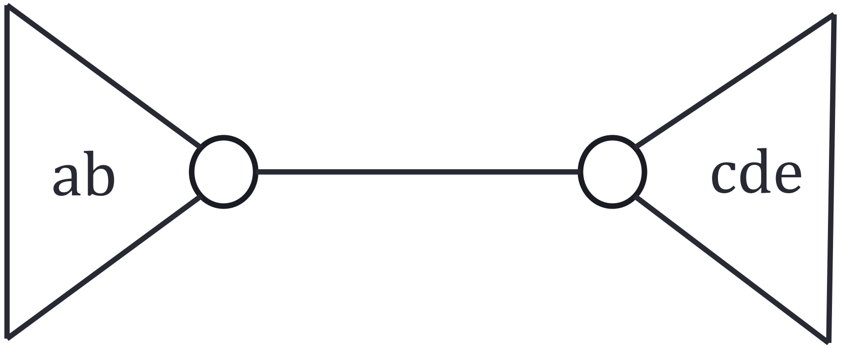
	a	b	c	d	e
a		1	2		
b		0	2		
c		4	0		
d		4	1		
e		4	1		

	a	b	c	d	e
a		1	2		
b		0	2		
c		4	0		
d		4	1		
e		4	1		

According to columns b and c, there are two types of taxa:

- those who prefer b $\{a, b\}$
- those who prefer c $\{c, d, e\}$

	a	b	c	d	e
a		1	2		
b		0	2		
c		4	0		
d		4	1		
e		4	1		

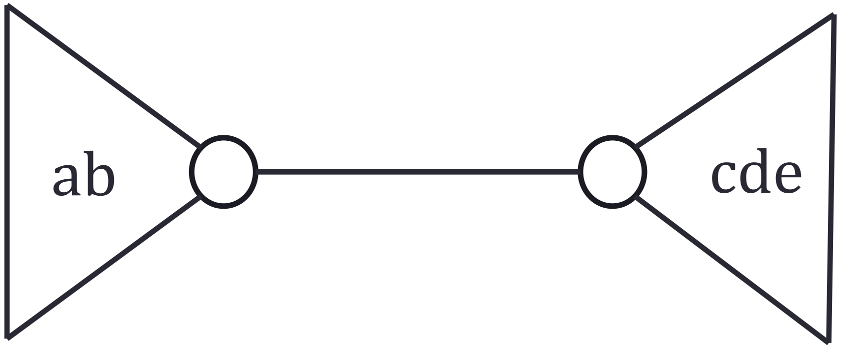


According to columns b and c, there are two types of taxa:

- those who prefer b $\{a, b\}$
- those who prefer c $\{c, d, e\}$

Can be proved: if a tree realizes R, it contains the split $ab|cde$

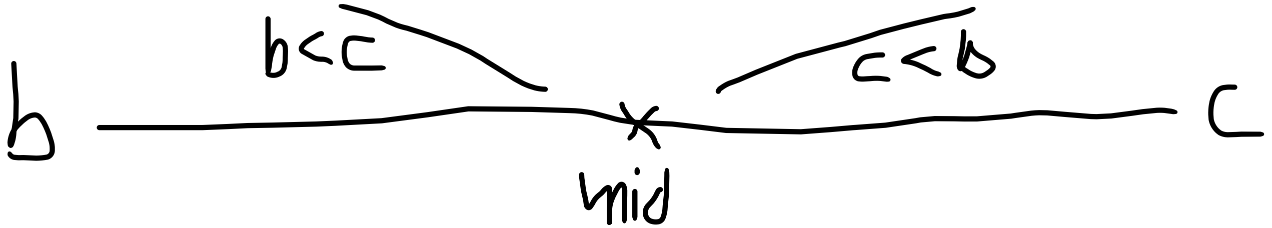
	a	b	c	d	e
a		1	2		
b		0	2		
c		4	0		
d		4	1		
e		4	1		



According to columns b and c, there are two types of taxa:

- those who prefer b $\{a, b\}$
- those who prefer c $\{c, d, e\}$

Can be proved: if a tree realizes R, it contains the split $ab|cde$



	a	b	c	d	e
a		1	2		
b		0	2		
c		4	0		
d		4	1		
e		4	1		

Proposition: for any u, v, let $S_{uv} = \{s : R(s,u) < R(s,v)\}$. If a tree T realizes R, then it contains the split $S_{uv}|X - S_{uv}$ (where X is the set of taxa).

	a	b	c	d	e
a	0	1	2	4	3
b	1	0	2	4	3
c	3	4	0	2	1
d	3	4	1	0	2
e	3	4	1	2	0

Proposition: for any u, v, let $S_{uv} = \{s : R(s,u) < R(s,v)\}$. If a tree T realizes R, then it contains the split $S_{uv} | X - S_{uv}$ (where X is the set of taxa).

	a	b	c	d	e
a	0	1			
b	1	0			
c	3	4			
d	3	4			
e	3	4			

acde|b

Proposition: for any u, v, let $S_{uv} = \{s : R(s,u) < R(s,v)\}$. If a tree T realizes R, then it contains the split $S_{uv}|X - S_{uv}$ (where X is the set of taxa).

Algorithm

- Compute all the mandatory splits $S_{uv}|X - S_{uv}$
- Find the tree T for this split system (if it exists)
- Find the edge weights to realize R using a LP.

Conjecture [Kearney, 1995]

If R is realizable, then this algorithm finds a tree that realizes R.

Algorithm

- Compute all the mandatory splits $S_{uv}|X - S_{uv}$
- Find the tree T for this split system (if it exists)
- Find the edge weights to realize R using a LP.

Conjecture [Kearney, 1995]

If R is realizable, then this algorithm finds a tree that realizes R.

Even if false, still useful to build a "backbone tree".

Algorithm

- Compute all the mandatory splits $S_{uv}|X - S_{uv}$
- Find the tree T for this split system (if it exists)
- Find the edge weights to realize R using a LP.
  - **OPEN : algorithm for edge weights without LP**

Conjecture [Kearney, 1995]

If R is realizable, then this algorithm finds a tree that realizes R.

Even if false, still useful to build a “backbone tree”.

The Ranked Distance Phylogeny problem

Given: ranking matrix R.

Find: an edge-weighted tree T that satisfies these rankings.

For each x, y, z, $R[x, y] < R[x, z] \Rightarrow dist_T(x, y) < dist_T(x, z)$

This definition allows ties in the rankings.

Equality = don't care

	a	b	c	d	e
a	0	1.8	1.7	3.9	3.5
b	1.8	0	5.3	5.5	5.4
c	1.7	5.3	0	2.5	1.7
d	3.9	5.5	2.5	0	3
e	3.5	5.4	1.7	3	0

The Ranked Distance Phylogeny problem

Given: ranking matrix R.

Find: an edge-weighted tree T that satisfies these rankings.

For each x, y, z, $R[x, y] < R[x, z] \Rightarrow dist_T(x, y) < dist_T(x, z)$

This definition allows ties in the rankings.

Equality = don't care

	a	b	c	d	e
a	0	1.8	1.7	3.9	3.5
b	1.8	0	5.3	5.5	5.4
c	1.7	5.3	0	2.5	1.7
d	3.9	5.5	2.5	0	3
e	3.5	5.4	1.7	3	0

	a	b	c	d	e
a	0	1	1	2	2
b	1	0	2	2	2
c	1	3	0	2	1
d	3	4	1	0	2
e	2	3	1	2	0

Conjecture

If R allows ties (don't cares), then it is NP-hard to decide whether R is realizable.

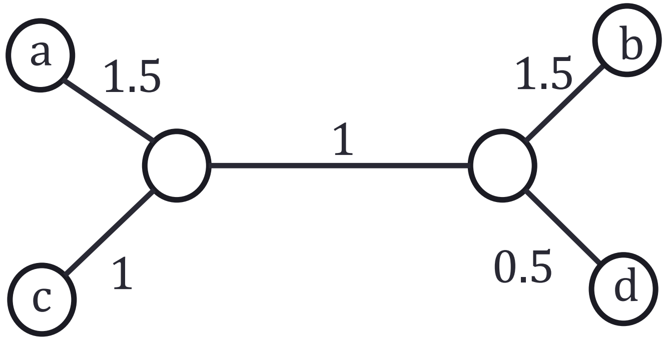
	a	b	c	d	e
a	0	1	1	2	2
b	1	0	2	2	2
c	1	3	0	2	1
d	3	4	1	0	2
e	2	3	1	2	0

Variant: R is binary and symmetric.

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

Variant: R is binary and symmetric.

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



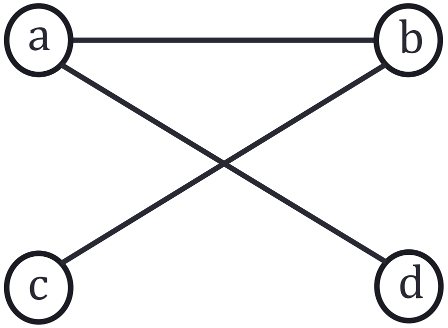
Variant: R is binary and symmetric.

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

Variant: R is binary and symmetric.

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

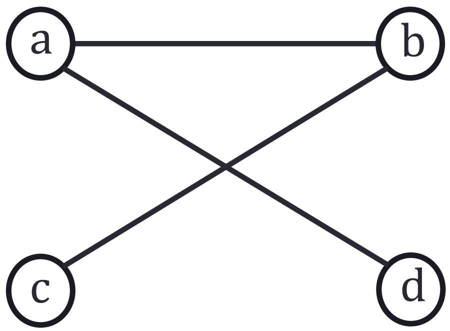
$G(R)$



Variant: R is binary and symmetric.

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

$G(R)$

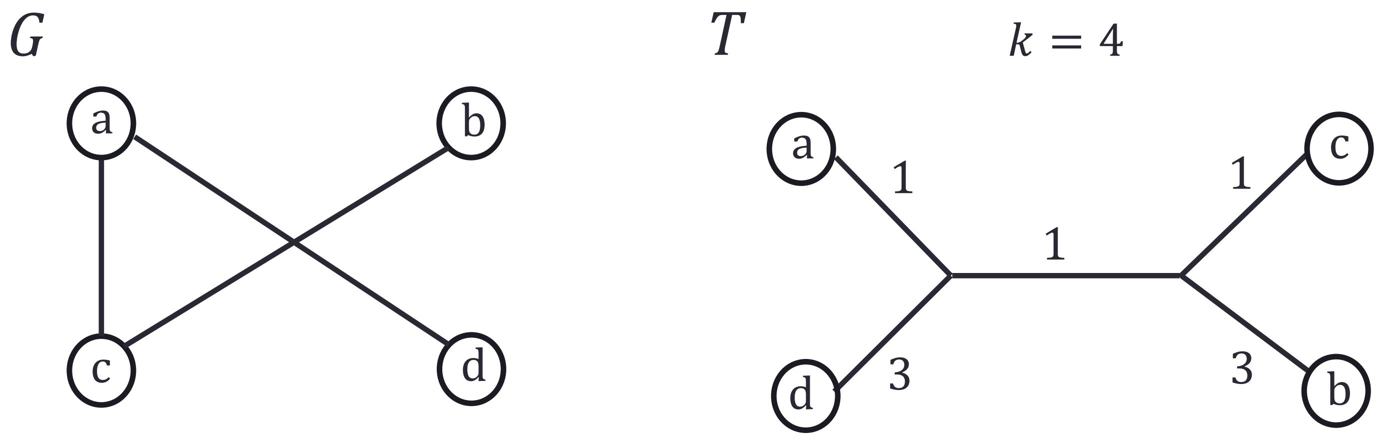


Theorem
If R is binary and symmetric, then R is realizable if and only if **the complement of $G(R)$ is a k-leaf power for some k.**

Definition (Nishimura et al., 2002)

A graph G is a ***k*-leaf power** if there exist a tree T such that:

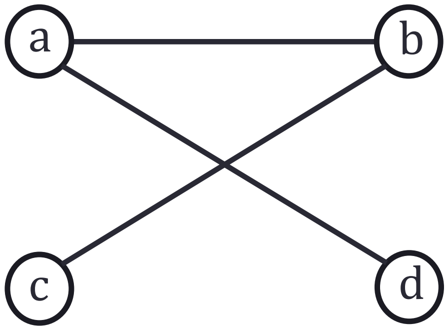
- $\boldsymbol{L(T) = V(G)}$, where $L(T)$ is the set of leaves of T
- $\boldsymbol{uv \in E(G) \Leftrightarrow dist_T(u, v) \leq k}$



Variant: R is binary and symmetric.

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

$G(R)$



Definition (Nishimura, 2002)

A graph G is a k-**leaf power** if there exist a tree T such that:

- $L(T) = V(G)$, where $L(T)$ is the set of leaves of T
- $uv \in E(G) \Leftrightarrow dist_T(u,v) \leq k$

Theorem

If R is binary and symmetric, then R is realizable if and only if the complement of $G(R)$ is a k-leaf power for some k.

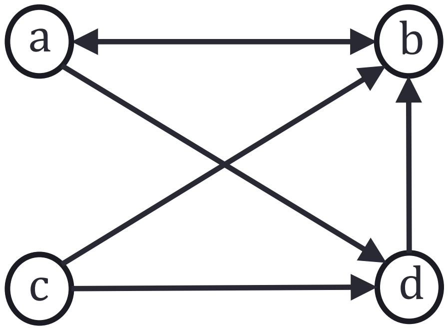
R is binary and symmetric.

- Equivalent to recognizing leaf powers.
- Complexity open since 2002.
- For fixed k, can decide if a graph G is a k-leaf power in time $O(n^{f(k)})$ [L, SODA2022]
- In general, complexity open.

<u>Variant</u>: R is binary but not symmetric.

	a	**b**	**c**	**d**
a	0	1	0	1
b	1	0	0	0
c	0	1	0	1
d	0	1	0	0

$G(R)$



Nothing known...

Some open problems

- **Problem 1** : when each row is a total order, are the mandatory splits sufficient?
- **Problem 1.1**: infer edge weights on given tree without LP.
- **Problem 2** : when ties are allowed, is realizability NP-hard?
- **Problem 3** : complexity of recognizing binary symmetric R, aka leaf powers.
- **Problem 4** : characterize binary R that may be non-symmetric.
- …