The binary perfect phylogeny model with persistent characters

P. Bonizzoni **A. P. Carrieri** R. Dondi G. Trucco

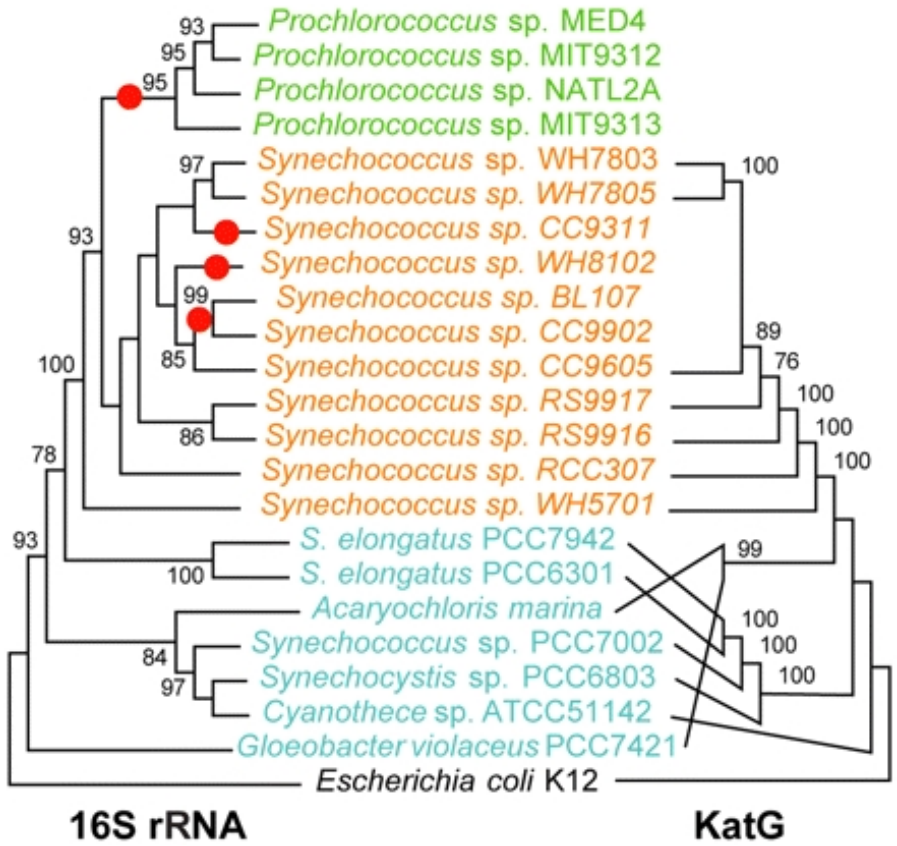
Dipartimento di Informatica, Sistemistica e Comunicazione
Universitá degli Studi di Milano–Bicocca - MILAN, ITALY

September 19th, 2012 - Varese, ITALY

The biological problem

The phylogenetic reconstruction

- **Phylogenetic tree** or **Phylogeny**: explains the evolutionary history of actual species or of genomic attributes (ex. tumor, protein domains phyogenies)



The character-based methods

Parsimony methods assume each species is specified by character states[1].

Maximum parsimony tree

- leaves labelled with character states associated with the input species
- internal nodes labelled with the inferred character states
- character state changes along its branches are minimized

[1] Felsenstein J. 2004. Inferring phylogenies. Sunderland (MA): Sinauer Associates

What is a character?

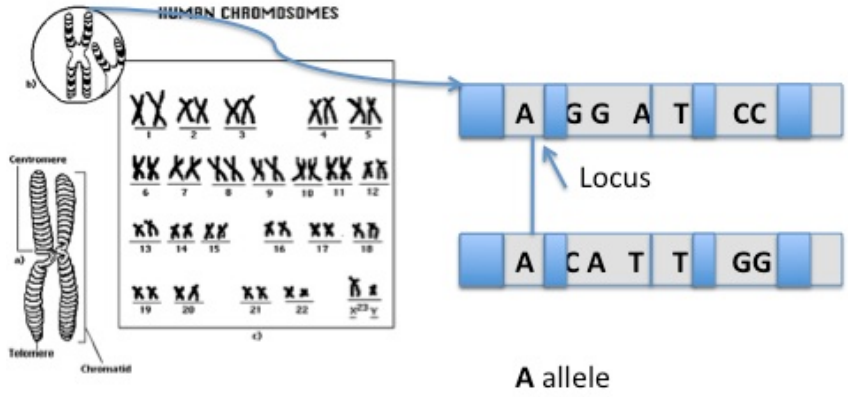
phenotype attribute

(wings, legs)

What is a character?

phenotype attribute

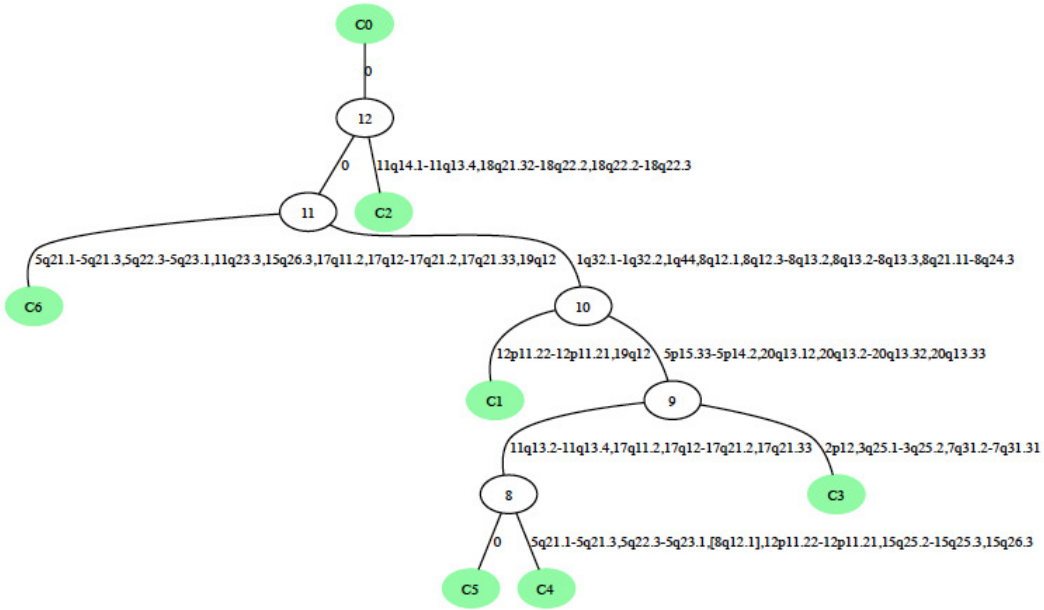
(wings, legs)



molecular information or genomic character

The character-based methods

Tumoral phylogeny



- characters → tumoral markers on genomic region
- inference of tumoral phylogeny[2]

[2] R. Schwartz et al. Inference of tumor phylogenies from genomic assays on heterogeneous samples *BCB '11 Proceedings of the 2nd ACM Conference on Bioinformatics, Computational Biology and Biomedicine*, 2011

Character Evolution

Binary characters have two states: 0 (absence) , 1 (presence)

Character mutations: $0 \to 1$ (acquisition), $1 \to 0$ (loss)

In the evolutionary tree

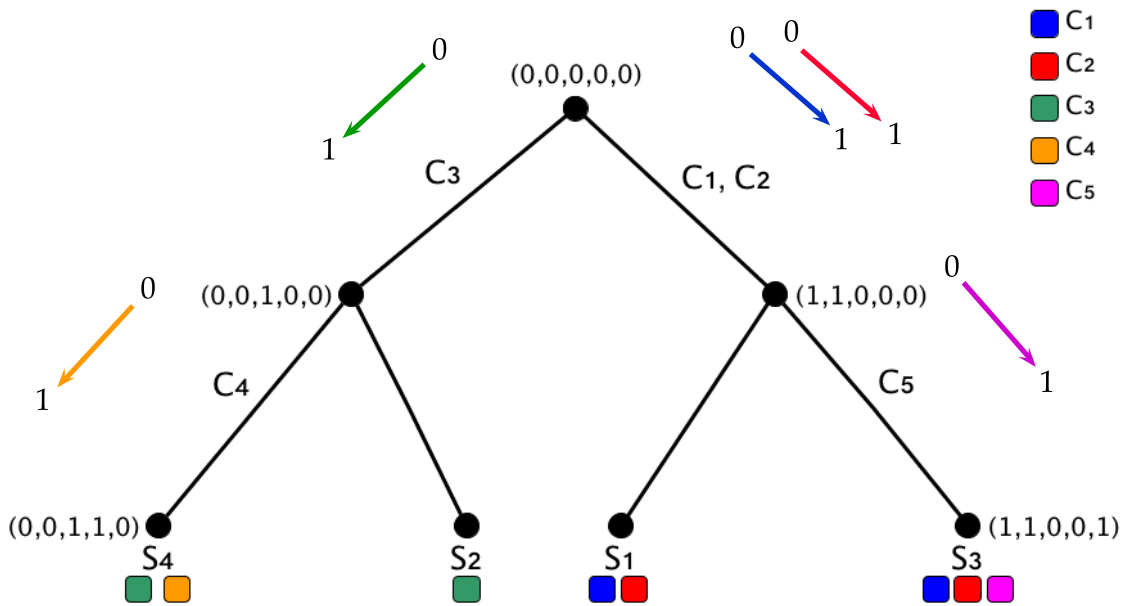
- $0 \to 1$ many times (recurrent mutations) for each character (*Camin-Sokal parsimony model*)
- $1 \to 0$ many times (back mutations) for each character (*Dollo parsimony model*)
- $0 \to 1$ only once for each character (*Perfect Phylogeny model*)

The Perfect Phylogeny model

Perfect Phylogeny (pp) for a binary matrix M of n species and m characters

- each node x is labelled by a m vector v_x giving in position j the state of character c_j

	c_1	c_2	c_3	c_4
s_1	1	1	0	0
s_2	0	0	1	0
s_3	1	1	0	0
s_4	0	0	1	1

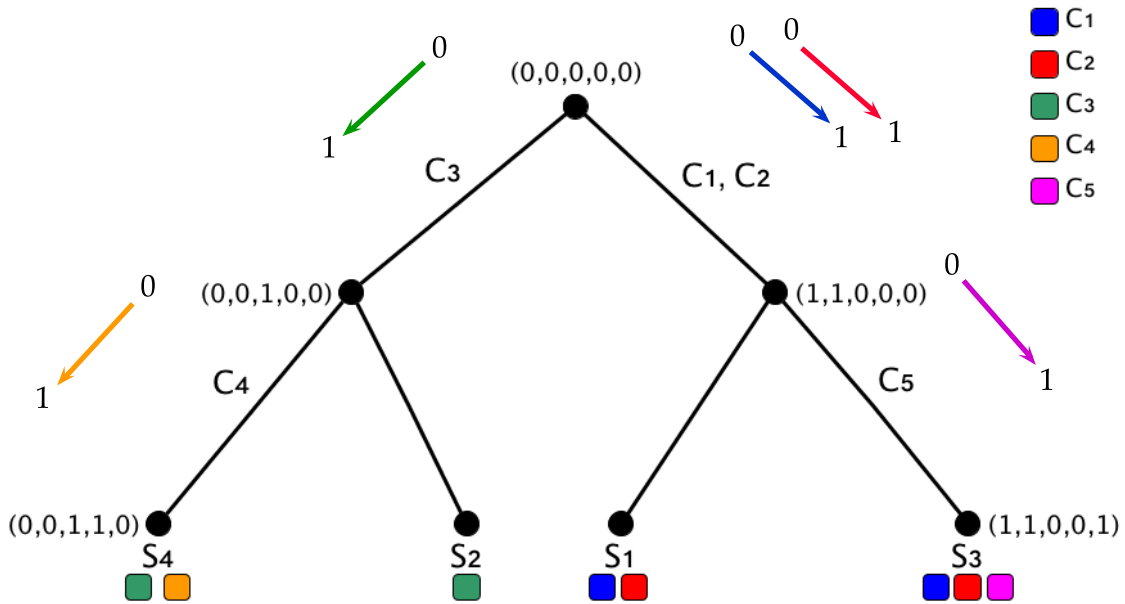


The Perfect Phylogeny model

Perfect Phylogeny (pp) for a binary matrix M of n species and m characters

- for each c_j there is at most one edge e, labelled c_j, where c_j changes state $0 \rightarrow 1$,

	c_1	c_2	c_3	c_4
s_1	1	1	0	0
s_2	0	0	1	0
s_3	1	1	0	0
s_4	0	0	1	1

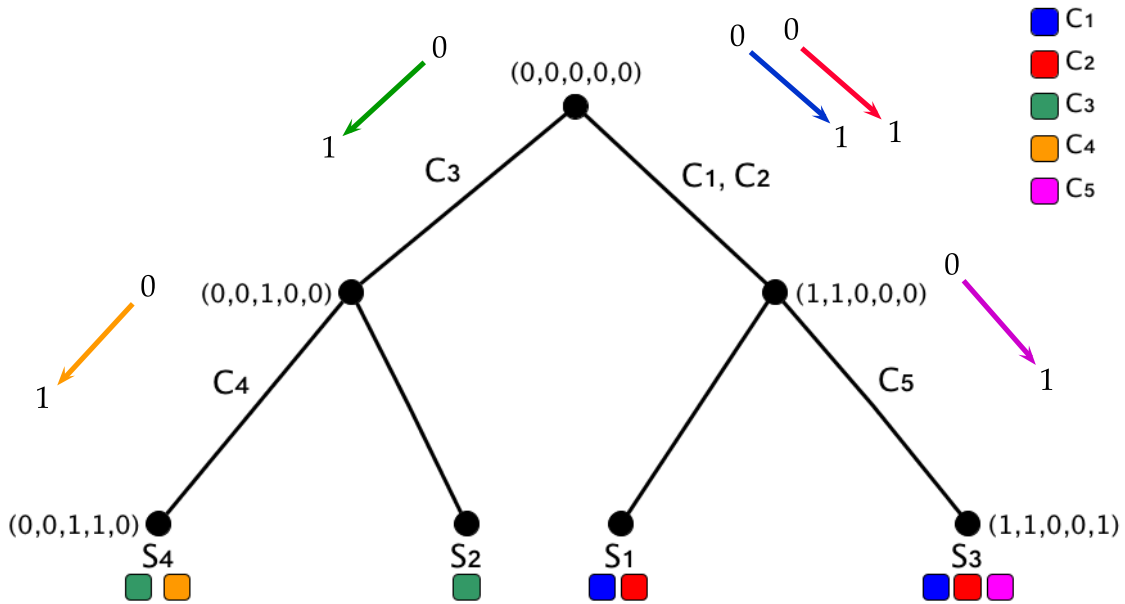


The Perfect Phylogeny model

Perfect Phylogeny (pp) for a binary matrix M of n species and m characters

- each row of matrix M labels exactly one leaf of T, the root is labelled by the zero m vector

	c_1	c_2	c_3	c_4
s_1	1	1	0	0
s_2	0	0	1	0
s_3	1	1	0	0
s_4	0	0	1	1



The computational problem

The Perfect Phylogeny Problem (PP)

Input: a binary $n \times m$ matrix M
Output: a pp tree for M, if it exists

The Perfect Phylogeny Problem (PP)

Input: a binary $n \times m$ matrix M
Output: a pp tree for M, if it exists

Camin-Sokal and Dollo parsimony models

- NP-complete (Day, 1986)
- recurrent mutations (Camin-Sokal)
- back mutations (Dollo)

Perfect Phylogeny model

- linear time algorithm[a]
- quite restrictive model

[a] D. Gusfield. Efficient algorithms for inferring evolutionary trees *Networks*, 1991

The Perfect Phylogeny Problem (PP)

Input: a binary $n \times m$ matrix M
Output: a pp tree for M, if it exists

Camin-Sokal and Dollo parsimony models

- NP-complete (Day, 1986)
- recurrent mutations (Camin-Sokal)
- back mutations (Dollo)

Perfect Phylogeny model

- linear time algorithm[a]
- quite restrictive model

[a] D. Gusfield. Efficient algorithms for inferring evolutionary trees *Networks*, 1991

The Perfect Phylogeny Problem (PP)

Input: a binary $n \times m$ matrix M
Output: a pp tree for M, if it exists

Camin-Sokal and Dollo parsimony models

- NP-complete (Day, 1986)
- recurrent mutations (Camin-Sokal)
- back mutations (Dollo)

Perfect Phylogeny model

- linear time algorithm[a]
- quite restrictive model

[a] D. Gusfield. Efficient algorithms for inferring evolutionary trees *Networks*, 1991

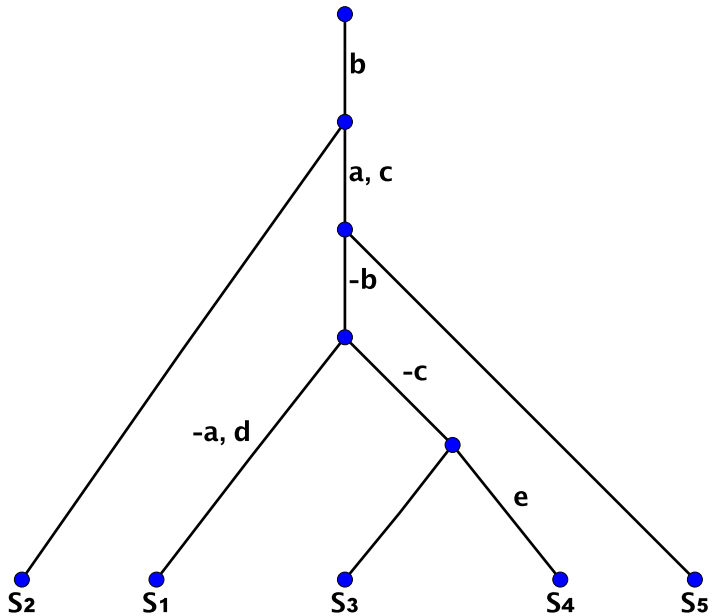
Our solution: a new model

The Persistent Perfect Phylogeny model (P-PP) $\rightarrow$ a Perfect Phylogeny with *persistent characters*

The Persistent Perfect Phylogeny (p-pp)

A perfect phylogeny but characters may be persistent[3]

- for each character c_j there may exists at most one edge where c_j mutates $0 \rightarrow 1$ and at most one edge where c_j mutates $1 \rightarrow 0$ (denoted as negated $\bar{c_j}$)



[3] T. Przytycka et al. Graph theoretical insights into dollo parsimony and evolution of multidomain proteins. *Journal of Computational Biology*, 2006

Our results

Persistent Perfect Phylogeny Problem (P-PP)

Input: a binary $n \times m$ matrix M
Output: a p-pp tree for M, if it exists

Our results

Persistent Perfect Phylogeny Problem (P-PP)

Input: a binary $n \times m$ matrix M
Output: a p-pp tree for M, if it exists

Question: is P-PP solvable by a polynomial time algorithm?

Our results

Persistent Perfect Phylogeny Problem (P-PP)

Input: a binary $n \times m$ matrix M
Output: a p-pp tree for M, if it exists

Question: is P-PP solvable by a polynomial time algorithm?

Our Results

1. A **polynomial time algorithm** for input matrices that have e-empty conflict graph
2. An **optimized exact algorithm** that runs in polynomial time in n (species) and exponential time in m (characters). It improves the execution time of the previous exact algorithm[a]

[a] P. Bonizzoni e Gabriella Trucco e Riccardo Dondi e Chiara Braghin. The binary perfect phylogeny with persistent characters. *TCS*, 2012

The conflict graph

The conflict graph G_c of M

- $G_c = \big(C, E \subseteq (C \times C)\big)$, where $(u, v) \in E$ if and only if u, v are in conflict in matrix M, that is (u, v) have the four-gametes $(0, 1)$, $(1, 1)$, $(1, 0)$, $(0, 0)$

About the algorithm

About our solution: the P-PP problem

- **P-PP problem reduced $\rightarrow$ Incomplete Persistent Perfect Phylogeny problem (IP-PP)**

Matrix M $(n \times m)$

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

$\rightarrow$

Extended matrix E $(n \times 2m)$

	a	$\bar{a}$	b	$\bar{b}$	c	$\bar{c}$	d	$\bar{d}$
1	1	0	?	?	?	?	?	?
2	1	0	1	0	?	?	?	?
3	?	?	1	0	?	?	1	0
4	?	?	?	?	1	0	1	0

About our solution: the P-PP problem

- **P-PP problem reduced $\rightarrow$ Incomplete Persistent Perfect Phylogeny problem (IP-PP)**

Matrix M $(n \times m)$

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

$\rightarrow$

Extended matrix E $(n \times 2m)$

	a	$\bar{a}$	b	$\bar{b}$	c	$\bar{c}$	d	$\bar{d}$
1	1	0	?	?	?	?	?	?
2	1	0	1	0	?	?	?	?
3	?	?	1	0	?	?	1	0
4	?	?	?	?	1	0	1	0

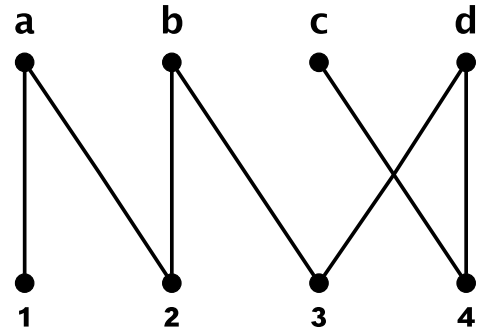
- **IP-PP problem reduced $\rightarrow$ Coloured Graph Reduction problem**

Extended matrix E $(n \times 2m)$

	a	$\bar{a}$	b	$\bar{b}$	c	$\bar{c}$	d	$\bar{d}$
1	1	0	?	?	?	?	?	?
2	1	0	1	0	?	?	?	?
3	?	?	1	0	?	?	1	0
4	?	?	?	?	1	0	1	0

$\rightarrow$

Red-black graph $G_{R,B}$

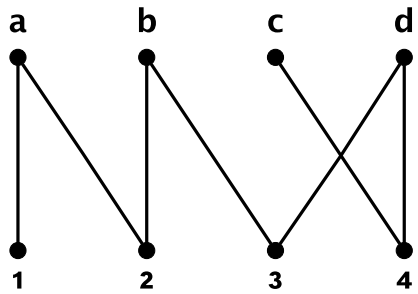


Graph solution of IP-PP problem

Goal $\rightarrow$ find an ordering $r = \langle c_{i_1}, \cdots, c_{i_m} \rangle$ of characters such that their realization reduces the red-black graph to the empty one!

Realization of a character c

1. Add red edges, remove black edges of c
2. (c, s) red-edge $\rightarrow$ complete $E(s, c)$ and $E(s, \bar{c})$ as $(1, 1)$



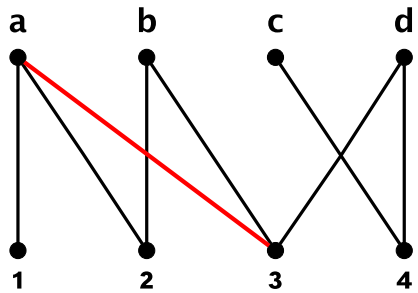
	a	$\bar{a}$	b	$\bar{b}$	c	$\bar{c}$	d	$\bar{d}$
1	1	0	?	?	?	?	?	?
2	1	0	1	0	?	?	?	?
3	?	?	1	0	?	?	1	0
4	?	?	?	?	1	0	1	0

Graph solution of IP-PP problem

Goal $\rightarrow$ find an ordering $r = \langle c_{i_1}, \cdots, c_{i_m} \rangle$ of characters such that their realization reduces the red-black graph to the empty one!

Realization of a character c

1. Add red edges, remove black edges of c
2. (c, s) red-edge $\rightarrow$ complete $E(s, c)$ and $E(s, \bar{c})$ as $(1, 1)$



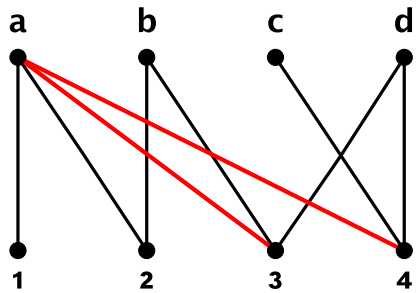
	a	$\bar{a}$	b	$\bar{b}$	c	$\bar{c}$	d	$\bar{d}$
1	1	0	?	?	?	?	?	?
2	1	0	1	0	?	?	?	?
3	1	1	1	0	?	?	1	0
4	?	?	?	?	1	0	1	0

Graph solution of IP-PP problem

Goal $\rightarrow$ find an ordering $r = \langle c_{i_1}, \cdots, c_{i_m} \rangle$ of characters such that their realization reduces the red-black graph to the empty one!

Realization of a character c

1. Add red edges, remove black edges of c
2. (c, s) red-edge $\rightarrow$ complete $E(s, c)$ and $E(s, \bar{c})$ as $(1, 1)$



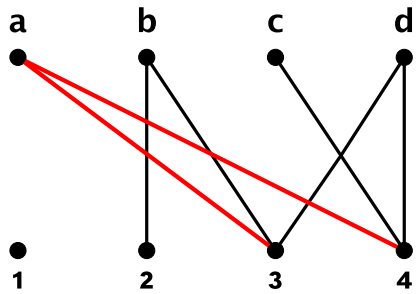
	a	$\bar{a}$	b	$\bar{b}$	c	$\bar{c}$	d	$\bar{d}$
1	1	0	?	?	?	?	?	?
2	1	0	1	0	?	?	?	?
3	1	1	1	0	?	?	1	0
4	1	1	?	?	1	0	1	0

Graph solution of IP-PP problem

Goal $\rightarrow$ find an ordering $r = \langle c_{i_1}, \cdots, c_{i_m} \rangle$ of characters such that their realization reduces the red-black graph to the empty one!

Realization of a character c

1. Add red edges, remove black edges of c
2. (c, s) red-edge $\rightarrow$ complete $E(s, c)$ and $E(s, \bar{c})$ as $(1, 1)$



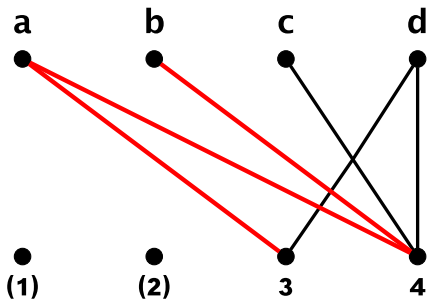
	a	$\bar{a}$	b	$\bar{b}$	c	$\bar{c}$	d	$\bar{d}$
1	1	0	?	?	?	?	?	?
2	1	0	1	0	?	?	?	?
3	1	1	1	0	?	?	1	0
4	1	1	?	?	1	0	1	0

Graph solution of IP-PP problem

Goal $\rightarrow$ find an ordering $r = \langle c_{i_1}, \cdots, c_{i_m} \rangle$ of characters such that their realization reduces the red-black graph to the empty one!

Realization of a character c

1. remove red-edges of $c \leftrightarrow c$ is connected by reg-edges to all species
2. the columns c and $\bar{c}$ are complete



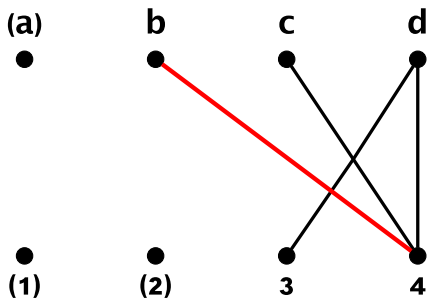
	a	$\bar{a}$	b	$\bar{b}$	c	$\bar{c}$	d	$\bar{d}$
1	1	0	0	0	0	0	0	0
2	1	0	1	0	?	?	?	?
3	1	1	1	0	?	?	1	0
4	1	1	?	?	1	0	1	0

Graph solution of IP-PP problem

Goal $\rightarrow$ find an ordering $r = \langle c_{i_1}, \cdots, c_{i_m} \rangle$ of characters such that their realization reduces the red-black graph to the empty one!

Realization of a character c

1. remove red-edges of $c \leftrightarrow c$ is connected by reg-edges to all species
2. the columns c and $\bar{c}$ are complete



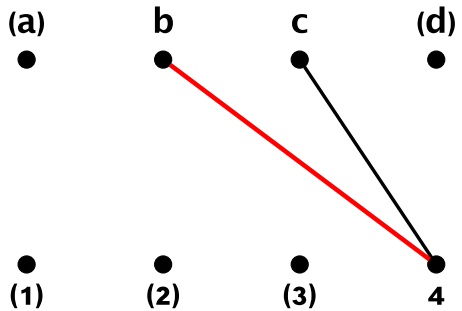
	a	$\bar{a}$	b	$\bar{b}$	c	$\bar{c}$	d	$\bar{d}$
1	1	0	0	0	0	0	0	0
2	1	0	1	0	0	0	0	0
3	1	1	1	0	?	?	1	0
4	1	1	1	1	1	0	1	0

Graph solution of IP-PP problem

Goal $\rightarrow$ find an ordering $r = \langle c_{i_1}, \cdots, c_{i_m} \rangle$ of characters such that their realization reduces the red-black graph to the empty one!

Realization of a character c

1. remove red-edges of $c \leftrightarrow c$ is connected by reg-edges to all species
2. the columns c and $\bar{c}$ are complete



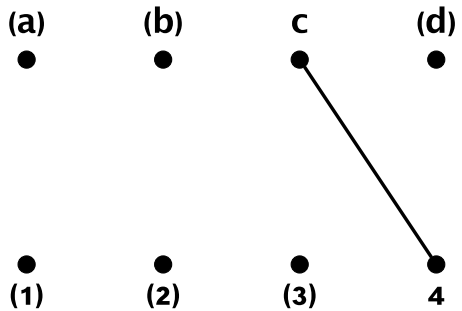
	a	$\bar{a}$	b	$\bar{b}$	c	$\bar{c}$	d	$\bar{d}$
1	1	0	0	0	0	0	0	0
2	1	0	1	0	0	0	0	0
3	1	1	1	0	?	?	1	0
4	1	1	1	1	1	0	1	0

Graph solution of IP-PP problem

Goal $\rightarrow$ find an ordering $r = \langle c_{i_1}, \cdots, c_{i_m} \rangle$ of characters such that their realization reduces the red-black graph to the empty one!

Realization of a character c

1. remove red-edges of $c \leftrightarrow c$ is connected by reg-edges to all species
2. the columns c and $\bar{c}$ are complete



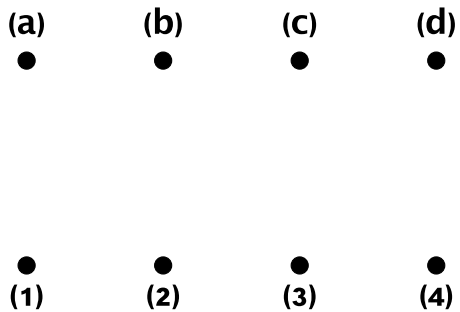
	a	$\bar{a}$	b	$\bar{b}$	c	$\bar{c}$	d	$\bar{d}$
1	1	0	0	0	0	0	0	0
2	1	0	1	0	0	0	0	0
3	1	1	1	0	?	?	1	0
4	1	1	1	1	1	0	1	0

Graph solution of IP-PP problem

Goal $\rightarrow$ find an ordering $r = \langle c_{i_1}, \cdots, c_{i_m} \rangle$ of characters such that their realization reduces the red-black graph to the empty one!

Realization of a character c

1. remove red-edges of $c \leftrightarrow c$ is connected by reg-edges to all species
2. the columns c and $\bar{c}$ are complete



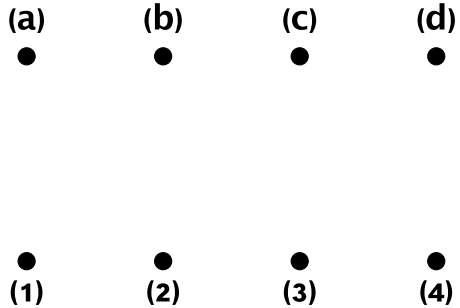
	a	$\bar{a}$	b	$\bar{b}$	c	$\bar{c}$	d	$\bar{d}$
1	1	0	0	0	0	0	0	0
2	1	0	1	0	0	0	0	0
3	1	1	1	0	0	0	1	0
4	1	1	1	1	1	0	1	0

Graph solution of IP-PP problem

Goal $\rightarrow$ find an ordering $r = \langle c_{i_1}, \cdots, c_{i_m} \rangle$ of characters such that their realization reduces the red-black graph to the empty one!

Realization of a character c

1. remove red-edges of $c \leftrightarrow c$ is connected by reg-edges to all species
2. the columns c and $\bar{c}$ are complete



	a	$\bar{a}$	b	$\bar{b}$	c	$\bar{c}$	d	$\bar{d}$
1	1	0	0	0	0	0	0	0
2	1	0	1	0	0	0	0	0
3	1	1	1	0	0	0	1	0
4	1	1	1	1	1	0	1	0

Observation

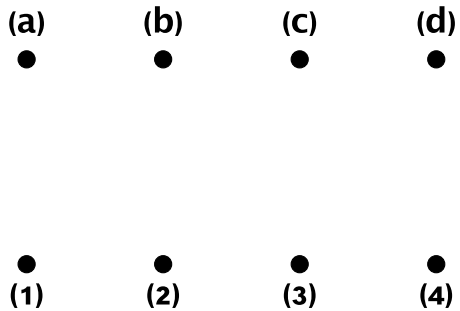
- $r = \langle a, b, d, c \rangle$ is a successful reduction of $G_{R,B}$
- The operations on $G_{R,B} \rightarrow$ construction a p-pp T for M in standard form

Graph solution of IP-PP problem

Goal $\rightarrow$ find an ordering $r = \langle c_{i_1}, \cdots, c_{i_m} \rangle$ of characters such that their realization reduces the red-black graph to the empty one!

Realization of a character c

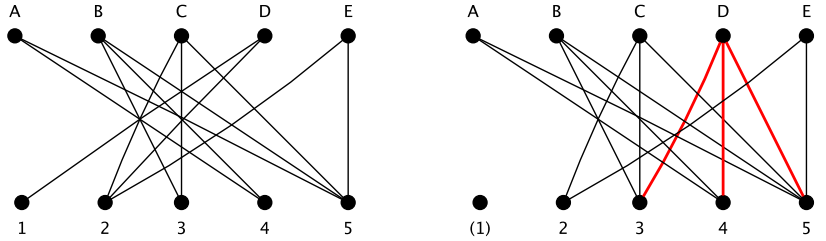
1. remove red-edges of $c \leftrightarrow c$ is connected by reg-edges to all species
2. the columns c and $\bar{c}$ are complete



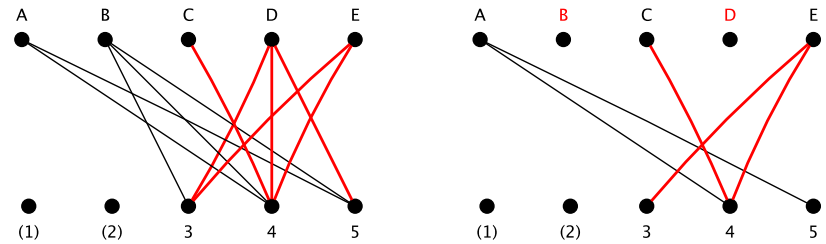
	a	$\bar{a}$	b	$\bar{b}$	c	$\bar{c}$	d	$\bar{d}$
1	1	0	0	0	0	0	0	0
2	1	0	1	0	0	0	0	0
3	1	1	1	0	0	0	1	0
4	1	1	1	1	1	0	1	0

Theorem

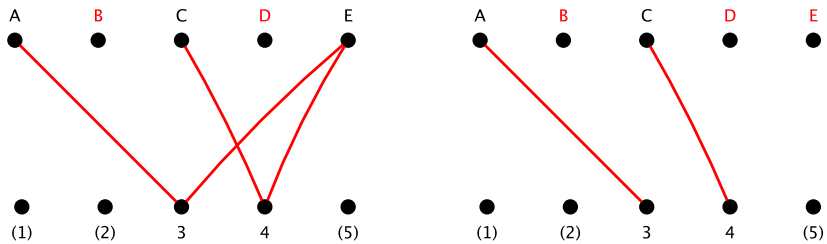
IP-PP has a solution on an extended matrix E if and only if the red-black graph G_{RB} for E has a successful reduction


A B C D E
1 2 3 4 5

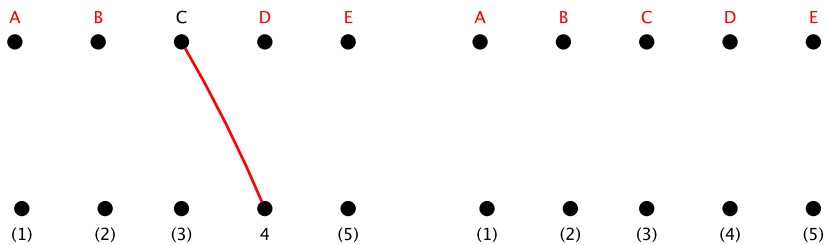
A B C D E
(1) 2 3 4 5


A B C D E
(1) (2) 3 4 5

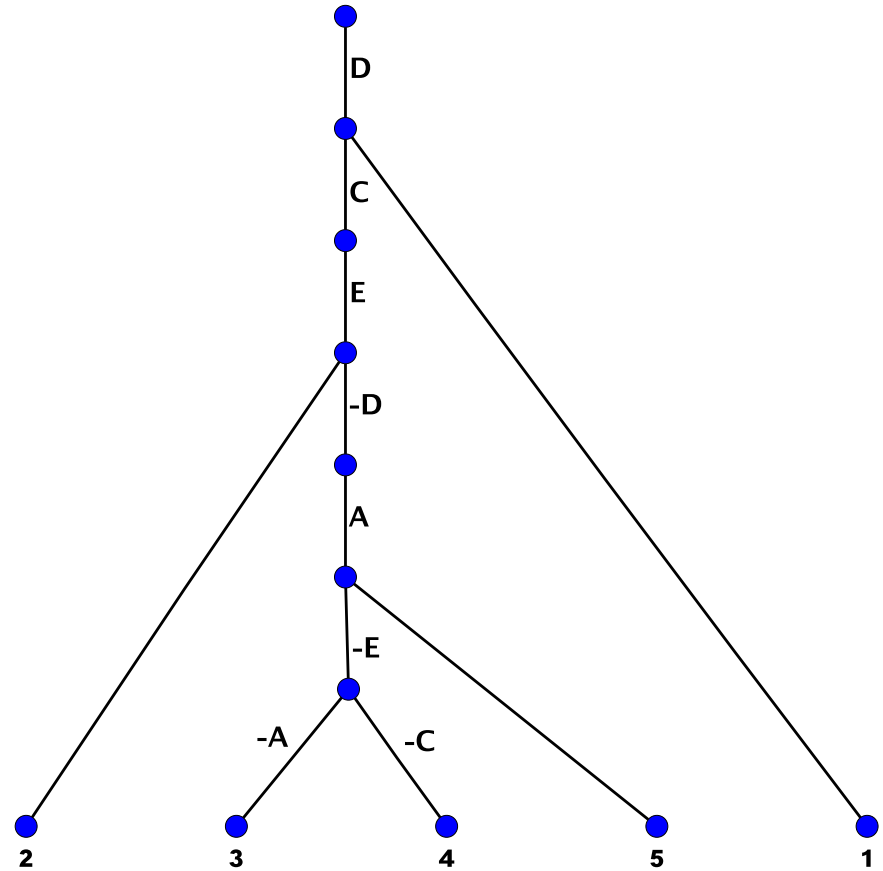
A B C D E
(1) (2) 3 4 5


A B C D E
(1) (2) 3 4 (5)

A B C D E
(1) (2) 3 4 (5)


A B C D E
(1) (2) (3) 4 (5)

A B C D E
(1) (2) (3) (4) (5)


D
C
E
-D
A
-E
-A
-C
2
3
4
5
1

Our results: more details

E-empty conflict graph: polynomial solution

Theorem

Let M be a binary matrix that has an e-empty conflict graph. Then matrix M admits a persistent perfect phylogeny and there exists a polynomial time algorithm to build the p-pp tree for M.

Partial order graph of C in M induced by $<$

- $c < c'$ if and only if $M[s, c] <= M[s, c']$, for each row s

E-empty conflict graph: polynomial solution

Theorem

Let M be a binary matrix that has an e-empty conflict graph. Then matrix M admits a persistent perfect phylogeny and there exists a polynomial time algorithm to build the p-pp tree for M.

Partial order graph of C in M induced by $<$

- $c < c'$ if and only if $M[s, c] <= M[s, c']$, for each row s

Polynomial time algorithm

- build the poset $(C, <)$
- iterate: add to r all the maximal elements in $(C, <)$, remove them from $(C, <)$[a]

[a] P. Bonizzoni (Algorithmica 2007): A linear time algorithm for the PPH problem via partial orders

The P-PP problem: the optimized algorithm

Input: a binary matrix M

Output: a successful reduction r for G_{RB} by a Branch and Bound like strategy, if it exists

Main steps

- construct a partial depth-first visit T of the decision tree $\mathcal{T}$
- compute a partial completion E' obtained by the realization of the characters along the path π from the root to a node x of $\mathcal{T}$
- if G_c of E' is e-empty $\rightarrow$ apply the polynomial time algorithm

Time complexity: polynomial in n (species) exponential in m (characters)

Our experiments

Comparing the execution times

The optimized algorithm

- tested over simulated data produced by Hudson tool
- table reports the computation time to solve sets of 50 matrices for each dimension

nxm	**solved matrices**		**total time in s**		**average time in s**	
	exact	*optimized*	*exact*	*optimized*	*exact*	*optimized*
50x15	47	50	89.12	32.32	1.90	0.65
100x15	48	50	436.02	194.63	9.08	3.89
200x15	48	50	1583.50	43.21	32.99	0.86
500x15	44	50	888.59	889.43	20.20	17.79

Open problems:

- Apply the P-PP model to real biological data
- Is the P-PP problem in P?

Thank you!