Efficient algorithms for inferring evolutionary trees Gusfield, 1991

Marc Manceau

September 8, 2021

Motivation	Perfect phylogeny building	
•0		

Motivation

Motivation

Compatibility of characters

Perfect phylogeny building

Compatibility of trees

Conclusion

Motivation	Perfect phylogeny building	
•0		

Motivation

Motivation

Compatibility of characters

Perfect phylogeny building

Compatibility of trees

Conclusion

Motivation ⊙●	Compatibility of characters 00000	Compatibility of trees	

Motivation

Very strong assumptions:

0/1 matrix we observe presence/absence of *m* characters among *n* taxa.

known ancestral state for each character, call 0 the ancestral state.

irreversibility no transition is allowed from state 1 back to 0.

We assume that characters satisfying these biological assumptions are known, but point out that finding such characters is a difficult biological task in building a phylogenetic tree.

Allow for very elegant algorithms developed in this paper:

- check that the characters are actually compatible, in O(nm).
- Find the perfect phylogeny in O(nm).

- simply to improve our knowledge of phylogenetics basics,
- for lineage tracing in developmental biology,
- in other fields as well when (a partition of the) data can be assumed to follow these assumptions.
- this could be used as a way to quickly reduce the tree space that one wants to explore.

Motivation	Perfect phylogeny building	
00		
Motivation		

0/1 matrix we observe presence/absence of *m* characters among *n* taxa.

known ancestral state for each character, call 0 the ancestral state.

irreversibility no transition is allowed from state 1 back to 0.

We assume that characters satisfying these biological assumptions are known, but point out that finding such characters is a difficult biological task in building a phylogenetic tree.

Allow for very elegant algorithms developed in this paper:

- check that the characters are actually compatible, in O(nm).
- Find the perfect phylogeny in O(nm).

- simply to improve our knowledge of phylogenetics basics,
- for lineage tracing in developmental biology,
- in other fields as well when (a partition of the) data can be assumed to follow these assumptions.
- this could be used as a way to quickly reduce the tree space that one wants to explore.

Motivation O●	Compatibility of characters 00000		Compatibility of trees	
Motivation				
Very strong ass 0/1 matrix we known ancestra	umptions: observe presence/absence of a I state for each character, cal	<i>n</i> characters among <i>n</i> taxa. I 0 the ancestral state.		

We assume that characters satisfying these biological assumptions are known, but point out that finding such characters is a difficult biological task in building a phylogenetic tree.

Allow for very elegant algorithms developed in this paper:

- check that the characters are actually compatible, in O(nm).
- Find the perfect phylogeny in O(nm).

- simply to improve our knowledge of phylogenetics basics,
- for lineage tracing in developmental biology,
- in other fields as well when (a partition of the) data can be assumed to follow these assumptions.
- this could be used as a way to quickly reduce the tree space that one wants to explore.

Motivation O●	Compatibility of characters		Compatibility of trees			
Motivation						
Very strong	g assumptions:					
0/1 matrix	0/1 matrix we observe presence/absence of m characters among n taxa.					
known ance	estral state for each character	call 0 the ancestral state.				

irreversibility no transition is allowed from state 1 back to 0.

We assume that characters satisfying these biological assumptions are known, but point out that finding such characters is a difficult biological task in building a phylogenetic tree.

Allow for very elegant algorithms developed in this paper:

- check that the characters are actually compatible, in O(nm).
- Find the perfect phylogeny in O(nm).

- simply to improve our knowledge of phylogenetics basics,
- for lineage tracing in developmental biology,
- in other fields as well when (a partition of the) data can be assumed to follow these assumptions.
- this could be used as a way to quickly reduce the tree space that one wants to explore.

Motivation O●	Compatibility of characters	Perfect phylogeny building	Compatibility of trees	Conclusion O
Motivation				
Very strong as	sumptions:			

0/1 matrix we observe presence/absence of *m* characters among *n* taxa.

known ancestral state for each character, call 0 the ancestral state.

irreversibility no transition is allowed from state 1 back to 0.

We assume that characters satisfying these biological assumptions are known, but point out that finding such characters is a difficult biological task in building a phylogenetic tree.

Allow for very elegant algorithms developed in this paper:

- check that the characters are actually compatible, in O(nm).
- Find the perfect phylogeny in O(nm).

- simply to improve our knowledge of phylogenetics basics,
- for lineage tracing in developmental biology,
- in other fields as well when (a partition of the) data can be assumed to follow these assumptions.
- this could be used as a way to quickly reduce the tree space that one wants to explore.

Motivation O●	Compatibility of characters	Perfect phylogeny building	Compatibility of trees	Conclusion O
Motivation				
Verv strong as	sumptions:			

0/1 matrix we observe presence/absence of *m* characters among *n* taxa.

known ancestral state for each character, call 0 the ancestral state.

irreversibility no transition is allowed from state 1 back to 0.

We assume that characters satisfying these biological assumptions are known, but point out that finding such characters is a difficult biological task in building a phylogenetic tree.

Allow for very elegant algorithms developed in this paper:

- check that the characters are actually compatible, in O(nm).
- Find the perfect phylogeny in O(nm).

- simply to improve our knowledge of phylogenetics basics,
- for lineage tracing in developmental biology,
- in other fields as well when (a partition of the) data can be assumed to follow these assumptions.
- this could be used as a way to quickly reduce the tree space that one wants to explore.

Motivation O●	Compatibility of characters	Compatibility of trees	
Motivation			
Very strong as	sumptions:		

0/1 matrix we observe presence/absence of *m* characters among *n* taxa.

known ancestral state for each character, call 0 the ancestral state.

irreversibility no transition is allowed from state 1 back to 0.

We assume that characters satisfying these biological assumptions are known, but point out that finding such characters is a difficult biological task in building a phylogenetic tree.

Allow for very elegant algorithms developed in this paper:

- check that the characters are actually compatible, in O(nm).
- find the perfect phylogeny in O(nm).

- simply to improve our knowledge of phylogenetics basics,
- for lineage tracing in developmental biology,
- in other fields as well when (a partition of the) data can be assumed to follow these assumptions.
- this could be used as a way to quickly reduce the tree space that one wants to explore.

Motivation O●	Compatibility of characters	Perfect phylogeny building	Compatibility of trees	Conclusion O
Motivation				
No.				

0/1 matrix we observe presence/absence of *m* characters among *n* taxa.

known ancestral state for each character, call 0 the ancestral state.

irreversibility no transition is allowed from state 1 back to 0.

We assume that characters satisfying these biological assumptions are known, but point out that finding such characters is a difficult biological task in building a phylogenetic tree.

Allow for very elegant algorithms developed in this paper:

- check that the characters are actually compatible, in O(nm).
- Find the perfect phylogeny in O(nm).

- simply to improve our knowledge of phylogenetics basics,
- for lineage tracing in developmental biology,
- in other fields as well when (a partition of the) data can be assumed to follow these assumptions.
- this could be used as a way to quickly reduce the tree space that one wants to explore.

Motivation O●	Compatibility of characters	Perfect phylogeny building	Compatibility of trees	Conclusion O
Motivation				
No.				

0/1 matrix we observe presence/absence of *m* characters among *n* taxa.

known ancestral state for each character, call 0 the ancestral state.

irreversibility no transition is allowed from state 1 back to 0.

We assume that characters satisfying these biological assumptions are known, but point out that finding such characters is a difficult biological task in building a phylogenetic tree.

Allow for very elegant algorithms developed in this paper:

- check that the characters are actually compatible, in O(nm).
- Find the perfect phylogeny in O(nm).

- simply to improve our knowledge of phylogenetics basics,
- for lineage tracing in developmental biology,
- in other fields as well when (a partition of the) data can be assumed to follow these assumptions.
- this could be used as a way to quickly reduce the tree space that one wants to explore.

Motivation O●	Compatibility of characters	Perfect phylogeny building	Compatibility of trees	Conclusion O
Motivation				
Many atward an				

0/1 matrix we observe presence/absence of *m* characters among *n* taxa.

known ancestral state for each character, call 0 the ancestral state.

irreversibility no transition is allowed from state 1 back to 0.

We assume that characters satisfying these biological assumptions are known, but point out that finding such characters is a difficult biological task in building a phylogenetic tree.

Allow for very elegant algorithms developed in this paper:

- check that the characters are actually compatible, in O(nm).
- Find the perfect phylogeny in O(nm).

- simply to improve our knowledge of phylogenetics basics,
- for lineage tracing in developmental biology,
- in other fields as well when (a partition of the) data can be assumed to follow these assumptions.
- this could be used as a way to quickly reduce the tree space that one wants to explore.

Motivation O●	Compatibility of characters	Perfect phylogeny building	Compatibility of trees	Conclusion O
Motivation				
Many atward an				

0/1 matrix we observe presence/absence of *m* characters among *n* taxa.

known ancestral state for each character, call 0 the ancestral state.

irreversibility no transition is allowed from state 1 back to 0.

We assume that characters satisfying these biological assumptions are known, but point out that finding such characters is a difficult biological task in building a phylogenetic tree.

Allow for very elegant algorithms developed in this paper:

- check that the characters are actually compatible, in O(nm).
- Find the perfect phylogeny in O(nm).

- simply to improve our knowledge of phylogenetics basics,
- for lineage tracing in developmental biology,
- in other fields as well when (a partition of the) data can be assumed to follow these assumptions.
- this could be used as a way to quickly reduce the tree space that one wants to explore.

Motivation O •	Compatibility of characters	Compatibility of trees	
Motivation			
Von strong og	umptions		

0/1 matrix we observe presence/absence of *m* characters among *n* taxa.

known ancestral state for each character, call 0 the ancestral state.

irreversibility no transition is allowed from state 1 back to 0.

We assume that characters satisfying these biological assumptions are known, but point out that finding such characters is a difficult biological task in building a phylogenetic tree.

Allow for very elegant algorithms developed in this paper:

- check that the characters are actually compatible, in O(nm).
- Find the perfect phylogeny in O(nm).

- simply to improve our knowledge of phylogenetics basics,
- for lineage tracing in developmental biology,
- in other fields as well when (a partition of the) data can be assumed to follow these assumptions.
- this could be used as a way to quickly reduce the tree space that one wants to explore.

Motivation O •	Compatibility of characters	Compatibility of trees	
Motivation			
Von strong og	umptions		

0/1 matrix we observe presence/absence of *m* characters among *n* taxa.

known ancestral state for each character, call 0 the ancestral state.

irreversibility no transition is allowed from state 1 back to 0.

We assume that characters satisfying these biological assumptions are known, but point out that finding such characters is a difficult biological task in building a phylogenetic tree.

Allow for very elegant algorithms developed in this paper:

- check that the characters are actually compatible, in O(nm).
- Find the perfect phylogeny in O(nm).

- simply to improve our knowledge of phylogenetics basics,
- for lineage tracing in developmental biology,
- in other fields as well when (a partition of the) data can be assumed to follow these assumptions.
- this could be used as a way to quickly reduce the tree space that one wants to explore.

Compatibility of characters	Perfect phylogeny building	
•0000		

Compatibility of characters The basics

Compatibility of characters: there exists a perfect phylogeny such that

- each character is associated to exactly one edge.
- \succ below this edge, all taxa have the derived state (1) otherwise they have the ancestral state (0).

Goal Given a 0/1 matrix of characters, find if there exists a perfect phylogeny.

Lemma It is the case iff $\forall i, j, O_i \cap O_j \in \{\emptyset, O_i, O_j\}$.

	Compatibility of characters ●0000	Compatibility of trees	
Compatibility The basics	of characters		

- each character is associated to exactly one edge.
- below this edge, all taxa have the derived state (1) otherwise they have the ancestral state (0).

Goal Given a 0/1 matrix of characters, find if there exists a perfect phylogeny.

Lemma It is the case iff $\forall i, j, O_i \cap O_j \in \{\emptyset, O_i, O_j\}$.

	Compatibility of characters ●0000	Compatibility of trees	
Compatibility The basics	of characters		

- each character is associated to exactly one edge.
- below this edge, all taxa have the derived state (1) otherwise they have the ancestral state (0).

Goal Given a 0/1 matrix of characters, find if there exists a perfect phylogeny.

Lemma It is the case iff $\forall i, j, O_i \cap O_j \in \{\emptyset, O_i, O_j\}$



Motivation	Compatibility of characters	Perfect phylogeny building	Compatibility of trees	Conclusion
OO	●0000	00000		O
Compatibility The basics	of characters			

- each character is associated to exactly one edge.
- below this edge, all taxa have the derived state (1) otherwise they have the ancestral state (0).

Goal Given a 0/1 matrix of characters, find if there exists a perfect phylogeny.

Lemma It is the case iff $\forall i, j, O_i \cap O_j \in \{\emptyset, O_i, O_j\}$.





	Compatibility of characters ●0000	Compatibility of trees	
Compatibility	of characters		

- each character is associated to exactly one edge.
- below this edge, all taxa have the derived state (1) otherwise they have the ancestral state (0).

Goal Given a 0/1 matrix of characters, find if there exists a perfect phylogeny.

Lemma It is the case iff $\forall i, j, O_i \cap O_j \in \{\emptyset, O_i, O_j\}$.





Motivation 00	Compatibility of characters ○●○○○	Perfect phylogeny building	Compatibility of trees	Conclusion O
Compatibility Step 1 : radix sor	of characters ting of columns			

- when you have data composed of digits that can be sorted lexicographically.
- when you want to avoid pairwise comparisons of elements.
- here each column is considered to be a binary number.

- 1. count the number of each element of the base,
- 2. order elements

	1	2	3	4	5	6
А	1	1	0	0	0	1
В	0	0	1	0	0	0
\mathbf{C}	1	1	0	0	1	1
D	0	0	1	1	0	0
Е	0	1	0	0	0	1

Motivation OO	Compatibility of characters ○●○○○	Perfect phylogeny building	Compatibility of trees	Conclusion O
Compatibility Step 1 : radix so	of characters ting of columns			

- when you have data composed of digits that can be sorted lexicographically.
- when you want to avoid pairwise comparisons of elements.
- here each column is considered to be a binary number.

- 1. count the number of each element of the base,
- 2. order elements

	1	2	3	4	5	6
А	1	1	0	0	0	1
В	0	0	1	0	0	0
\mathbf{C}	1	1	0	0	1	1
D	0	0	1	1	0	0
Е	0	1	0	0	0	1

Motivation	Compatibility of characters	Perfect phylogeny building	Compatibility of trees	Conclusion
00	○●○○○	00000		O
Compatibility Step 1 : radix so	of characters rting of columns			

- when you have data composed of digits that can be sorted lexicographically.
- when you want to avoid pairwise comparisons of elements.
- here each column is considered to be a binary number.

- count the number of each element of the base,
- order elements

	1	2	3	4	5	6
А	1	1	0	0	0	1
В	0	0	1	0	0	0
\mathbf{C}	1	1	0	0	1	1
D	0	0	1	1	0	0
Е	0	1	0	0	0	1

Motivation OO	Compatibility of characters ○●○○○	Perfect phylogeny building	Compatibility of trees	Conclusion O
Compatibility Step 1 : radix so	of characters rting of columns			

- when you have data composed of digits that can be sorted lexicographically.
- when you want to avoid pairwise comparisons of elements.
- here each column is considered to be a binary number.

How ? For each digit,

- 1. count the number of each element of the base,
- 2. order elements

	1	2	3	4	5	6
Α	1	1	0	0	0	1
В	0	0	1	0	0	0
\mathbf{C}	1	1	0	0	1	1
D	0	0	1	1	0	0
Е	0	1	0	0	0	1

binary number

	Compatibility of characters ○●○○○	Compatibility of trees	
Compatibility Step 1 : radix so	of characters rting of columns		

- when you have data composed of digits that can be sorted lexicographically.
- when you want to avoid pairwise comparisons of elements.
- here each column is considered to be a binary number.

- 1. count the number of each element of the base,
- 2. order elements



Motivation OO	Compatibility of characters ○●○○○	Perfect phylogeny building	Compatibility of trees	Conclusion O
Compatibility Step 1 : radix so	of characters rting of columns			

- when you have data composed of digits that can be sorted lexicographically.
- when you want to avoid pairwise comparisons of elements.
- here each column is considered to be a binary number.

- 1. count the number of each element of the base,
- 2. order elements



Motivation 00	Compatibility of characters ○●○○○	Perfect phylogeny building	Compatibility of trees	Conclusion O
Compatibility Step 1 : radix so	of characters rting of columns			

- when you have data composed of digits that can be sorted lexicographically.
- when you want to avoid pairwise comparisons of elements.
- here each column is considered to be a binary number.

- 1. count the number of each element of the base,
- order elements

	1	2	3	4	5	6
А	1	1	0	0	0	1
В	0	0	1	0	0	0
С	1	1	0	0	1	1
D	0	0	1	1	0	0
Е	0	1	0	0	0	1

Motivation 00	Compatibility of characters ○●○○○	Perfect phylogeny building	Compatibility of trees	Conclusion O
Compatibility Step 1 : radix so	of characters rting of columns			

- when you have data composed of digits that can be sorted lexicographically.
- when you want to avoid pairwise comparisons of elements.
- here each column is considered to be a binary number.

- 1. count the number of each element of the base,
- 2. order elements

	1	2	3	4	5	6	
А	1	1	0	0	0	1	
В	0	0	1	0	0	0	
С	1	1	0	0	1	1	
D	0	0	1	1	0	0	
Е	0	1	0	0	0	1	-

Motivation 00	Compatibility of characters ○●○○○	Perfect phylogeny building	Compatibility of trees	Conclusion O
Compatibility Step 1 : radix so	of characters rting of columns			

- when you have data composed of digits that can be sorted lexicographically.
- when you want to avoid pairwise comparisons of elements.
- here each column is considered to be a binary number.

- 1. count the number of each element of the base,
- 2. order elements

	2	6	1	3	4	5
А	1	1	1	0	0	0
В	0	0	0	1	0	0
С	1	1	1	0	0	1
D	0	0	0	1	1	0
Е	1	1	0	0	0	0

Motivation 00	Compatibility of characters ○●○○○	Perfect phylogeny building	Compatibility of trees	Conclusion O
Compatibilit Step 1 : radix	y of characters sorting of columns			

- when you have data composed of digits that can be sorted lexicographically.
- when you want to avoid pairwise comparisons of elements.
- here each column is considered to be a binary number.

- 1. count the number of each element of the base,
- 2. order elements



Motivation 00	Compatibility of characters ○●○○○	Perfect phylogeny building	Compatibility of trees	Conclusion O
Compatibility Step 1 : radix so	of characters rting of columns			

- when you have data composed of digits that can be sorted lexicographically.
- when you want to avoid pairwise comparisons of elements.
- here each column is considered to be a binary number.

- 1. count the number of each element of the base,
- 2. order elements

	3	4	2	6	1	5
А	0	0	1	1	1	0
В	1	0	0	0	0	0
С	0	0	1	1	1	1
D	1	1	0	0	0	0
Е	0	0	1	1	0	0

Motivation 00	Compatibility of characters ○●○○○	Perfect phylogeny building	Compatibility of trees	Conclusion O
Compatibilit Step 1 : radix	y of characters sorting of columns			

- when you have data composed of digits that can be sorted lexicographically.
- when you want to avoid pairwise comparisons of elements.
- here each column is considered to be a binary number.

- 1. count the number of each element of the base,
- 2. order elements



Motivation 00	Compatibility of characters ○●○○○	Perfect phylogeny building	Compatibility of trees	Conclusion O
Compatibility Step 1 : radix so	of characters rting of columns			

- when you have data composed of digits that can be sorted lexicographically.
- when you want to avoid pairwise comparisons of elements.
- here each column is considered to be a binary number.

- 1. count the number of each element of the base,
- 2. order elements

	2	6	1	5	3	4
А	1	1	1	0	0	0
В	0	0	0	0	1	0
С	1	1	1	1	0	0
D	0	0	0	0	1	1
Е	1	1	0	0	0	0

Motivation 00	Compatibility of characters ○●○○○	Perfect phylogeny building	Compatibility of trees	Conclusion O
Compatibilit Step 1 : radix	y of characters sorting of columns			

- when you have data composed of digits that can be sorted lexicographically.
- when you want to avoid pairwise comparisons of elements.
- here each column is considered to be a binary number.

- 1. count the number of each element of the base,
- 2. order elements



Motivation 00	Compatibility of characters ○●○○○	Perfect phylogeny building	Compatibility of trees	Conclusion O
Compatibility Step 1 : radix so	of characters rting of columns			

- when you have data composed of digits that can be sorted lexicographically.
- when you want to avoid pairwise comparisons of elements.
- here each column is considered to be a binary number.

- 1. count the number of each element of the base,
- 2. order elements

	3	2	6	1	5	4
А	0	1	1	1	0	0
В	1	0	0	0	0	0
С	0	1	1	1	1	0
D	1	0	0	0	0	1
Е	0	1	1	0	0	0
Motivation 00	Compatibility of characters ○●○○○	Perfect phylogeny building	Compatibility of trees	Conclusion O		
--------------------------------	---------------------------------------	----------------------------	------------------------	-----------------		
Compatibilit Step 1 : radix	y of characters sorting of columns					

When ?

- when you have data composed of digits that can be sorted lexicographically.
- when you want to avoid pairwise comparisons of elements.
- here each column is considered to be a binary number.

How ? For each digit,

- 1. count the number of each element of the base,
- 2. order elements



Motivation 00	Compatibility of characters ○●○○○	Perfect phylogeny building	Compatibility of trees	Conclusion O
Compatibility Step 1 : radix so	of characters rting of columns			

When ?

- when you have data composed of digits that can be sorted lexicographically.
- when you want to avoid pairwise comparisons of elements.
- here each column is considered to be a binary number.

How ? For each digit,

- 1. count the number of each element of the base,
- 2. order elements

	2	6	1	3	5	4
А	1	1	1	0	0	0
В	0	0	0	1	0	0
С	1	1	1	0	1	0
D	0	0	0	1	0	1
Е	1	1	0	0	0	0

Motivation 00	Compatibility of characters ○○●○○	Perfect phylogeny building	Compatibility of trees	Conclusion O
Compatibility Step 2 : delete d	of characters uplicate columns			

items are sorted, so removing duplicate columns is very quick,

these are characters that are exactly in the same state in our taxa.

	2	6	1	3	5	4
А	1	1	1	0	0	0
В	0	0	0	1	0	0
С	1	1	1	0	1	0
D	0	0	0	1	0	1
Е	1	1	0	0	0	0

Motivation 00	Compatibility of characters ○○●○○	Perfect phylogeny building	Compatibility of trees	Conclusion O
Compatibility Step 2 : delete de	of characters uplicate columns			

- items are sorted, so removing duplicate columns is very quick,
- these are characters that are exactly in the same state in our taxa.

	2	6	1	3	5	4
А	1	1	1	0	0	0
В	0	0	0	1	0	0
С	1	1	1	0	1	0
D	0	0	0	1	0	1
Е	1	1	0	0	0	0

Motivation 00	Compatibility of characters 00●00	Perfect phylogeny building	Compatibility of trees	Conclusion O
Compatibility Step 2 : delete di	of characters uplicate columns			

- items are sorted, so removing duplicate columns is very quick,
- these are characters that are exactly in the same state in our taxa.

	2	1	3	5	4
А	1	1	0	0	0
В	0	0	1	0	0
С	1	1	0	1	0
D	0	0	1	0	1
Е	1	0	0	0	0

Motivation 00	Compatibility of characters ○○○●○	Perfect phylogeny building	Compatibility of trees	Conclusion O
Compatibility	of characters			

Step 3 : find the parents of each character

Column j to the right of column i can be:

- ▶ a child of *i*, i.e. $O_j \cap O_i = O_j$
- or belong to a different part of the tree, i.e. $O_j \cap O_i = \emptyset$.

$$L(i,j) := \max\{0\} \cup \{k < j, M'(i,k) = 1\}$$

$$I(j) := \max_{i} L(i,j)$$



	Compatibility of characters 000●0	Compatibility of trees	
Compatibility	of characters		

- ▶ a child of *i*, i.e. $O_i \cap O_i = O_i$
- or belong to a different part of the tree, i.e. $O_j \cap O_i = \emptyset$.

$$L(i,j) := \max\{0\} \cup \{k < j, M'(i,k) = 1\}$$

$$I(j) := \max_{i} L(i,j)$$



Motivation 00	Compatibility of characters ○○○●○	Perfect phylogeny building	Compatibility of trees	Conclusion O
Compatibility Step 3 : find the	of characters parents of each character			

- ▶ a child of *i*, i.e. $O_j \cap O_i = O_j$
- or belong to a different part of the tree, i.e. $O_i \cap O_i = \emptyset$.

$$L(i,j) := \max\{0\} \cup \{k < j, M'(i,k) = 1\}$$

$$I(j) := \max_{i} L(i,j)$$



	Compatibility of characters 000●0	Compatibility of trees	
Compatibility	of characters		

- ▶ a child of *i*, i.e. $O_i \cap O_i = O_i$
- or belong to a different part of the tree, i.e. $O_i \cap O_i = \emptyset$.

$$L(i,j) := \max\{0\} \cup \{k < j, M'(i,k) = 1\}$$

$$I(j) := \max_{i} L(i,j)$$



	Compatibility of characters 000●0	Compatibility of trees	
Compatibility	of characters		

- > a child of *i*, i.e. $O_i \cap O_i = O_i$
- or belong to a different part of the tree, i.e. $O_j \cap O_i = \emptyset$.

$$L(i,j) := \max\{0\} \cup \{k < j, M'(i,k) = 1\}$$

$$I(j) := \max_{i} L(i,j)$$



	Compatibility of characters 000●0	Compatibility of trees	
Compatibility	of characters		

- > a child of *i*, i.e. $O_i \cap O_i = O_i$
- or belong to a different part of the tree, i.e. $O_j \cap O_i = \emptyset$.

$$L(i,j) := \max\{0\} \cup \{k < j, M'(i,k) = 1\}$$

$$I(j) := \max_{i} L(i,j)$$



	Compatibility of characters 000●0	Compatibility of trees	
Compatibility	of characters		

- > a child of *i*, i.e. $O_i \cap O_i = O_i$
- or belong to a different part of the tree, i.e. $O_j \cap O_i = \emptyset$.

Compute the direct parent:

$$L(i,j) := \max\{0\} \cup \{k < j, M'(i,k) = 1\}$$

$$I(j) := \max_{i} L(i,j)$$



0

	Compatibility of characters 000●0	Compatibility of trees	
Compatibility	of characters		

- > a child of *i*, i.e. $O_i \cap O_i = O_i$
- or belong to a different part of the tree, i.e. $O_j \cap O_i = \emptyset$.

$$L(i,j) := \max\{0\} \cup \{k < j, M'(i,k) = 1\}$$

$$I(j) := \max_{i} L(i,j)$$



	Compatibility of characters 000●0	Compatibility of trees	
Compatibility	of characters		

- > a child of *i*, i.e. $O_i \cap O_i = O_i$
- or belong to a different part of the tree, i.e. $O_i \cap O_i = \emptyset$.

$$L(i,j) := \max\{0\} \cup \{k < j, M'(i,k) = 1\}$$

$$I(j) := \max_{i} L(i,j)$$



	Compatibility of characters ○○○○●	Compatibility of trees	
Compatibility	of characters		

Step 4 : check compatibility

check that the genealogical history is coherent,

$$\blacktriangleright \text{ i.e. } \forall i,j \ L(i,j) = l(j).$$



vector l = 0 1 0 2 3

This one example does show compatible characters.

	Compatibility of characters ○○○○●	Compatibility of trees	
Compatibility	of characters		

Step 4 : check compatibility

check that the genealogical history is coherent,

$$\blacktriangleright \text{ i.e. } \forall i,j \ L(i,j) = l(j).$$



vector l = 0 1 0 2 3

This one example does show compatible characters.

Motivation OO	Compatibility of characters	Perfect phylogeny building ●0000	Compatibility of trees	Conclusion O
Perfect phylog Step 1 : create t	geny building he backbone tree of characters			

- create a node,
- link it with an edge labeled j to the character in position I(j).

	Compatibility of characters	Perfect phylogeny building ●0000	Compatibility of trees	
Perfect phylo Step 1 : create	ogeny building the backbone tree of charad	cters		

4

- create a node,
- link it with an edge labeled j to the character in position I(j).



	Compatibility of characters	Perfect phylogeny building ●0000	Compatibility of trees	
Perfect phylo	geny building	c .		

- create a node,
- link it with an edge labeled j to the character in position I(j).



	Compatibility of characters	Perfect phylogeny building ●0000	Compatibility of trees	
Perfect phy Step 1 : create	logeny building the backbone tree of charad	cters		

- create a node,
- link it with an edge labeled j to the character in position I(j).



	Compatibility of characters	Perfect phylogeny building ●0000	Compatibility of trees	
Perfect phy Step 1 : create	logeny building the backbone tree of charad	cters		

- create a node,
- link it with an edge labeled j to the character in position I(j).



	Compatibility of characters	Perfect phylogeny building ●0000	Compatibility of trees	
Perfect phy Step 1 : create	logeny building the backbone tree of charad	cters		

- create a node,
- link it with an edge labeled j to the character in position I(j).



	Compatibility of characters	Perfect phylogeny building ●0000	Compatibility of trees	
Perfect phylo	geny building	ters		

- create a node,
- link it with an edge labeled j to the character in position I(j).





	Compatibility of characters	Perfect phylogeny building ○●○○○	Compatibility of trees	
Perfect phylog Step 2 : attach t	geny building axa			

- Find the most tippy derived character it belongs to, i.e. max $\{j, M'(i,j) = 1\}$
- if it is not a leaf, add a leaf labeled i,
- else simply change the label of the leaf for i.





	Compatibility of characters	Perfect phylogeny building ○●○○○	Compatibility of trees	
Perfect phylog Step 2 : attach ta	geny building axa			

- Find the most tippy derived character it belongs to, i.e. max $\{j, M'(i,j) = 1\}$
- if it is not a leaf, add a leaf labeled i,
- else simply change the label of the leaf for i.





	Compatibility of characters	Perfect phylogeny building ○●○○○	Compatibility of trees	
Perfect phylog Step 2 : attach t	geny building			

- find the most tippy derived character it belongs to, i.e. max $\{j, M'(i,j) = 1\}$
- if it is not a leaf, add a leaf labeled i,
- else simply change the label of the leaf for i.



	Compatibility of characters	Perfect phylogeny building ○●○○○	Compatibility of trees	
Perfect phylog	geny building			

- find the most tippy derived character it belongs to, i.e. max $\{j, M'(i, j) = 1\}$
- if it is not a leaf, add a leaf labeled i,
- else simply change the label of the leaf for i.



C

	Compatibility of characters	Perfect phylogeny building ○●○○○	Compatibility of trees	
Perfect phylog	geny building			

- find the most tippy derived character it belongs to, i.e. max $\{j, M'(i,j) = 1\}$
- if it is not a leaf, add a leaf labeled i,
- else simply change the label of the leaf for i.



C

	Compatibility of characters	Perfect phylogeny building ○●○○○	Compatibility of trees	
Perfect phylog	geny building			

- find the most tippy derived character it belongs to, i.e. max $\{j, M'(i,j) = 1\}$
- if it is not a leaf, add a leaf labeled i,
- else simply change the label of the leaf for i.



 \mathbf{C}

	Compatibility of characters	Perfect phylogeny building ○○●○○	Compatibility of trees	
Perfect phylogeny building				

- I have no idea if it can be useful for anything but it sounds a bit more like a "fun fact".
- Definition: all ones in a column are in consecutive rows.
- \triangleright M has a perfect phylogeny \Rightarrow rows can be permuted so as to satisfy the consecutive ones property.

	2	6	1	3	5	4
А	1	1	1	0	0	0
В	0	0	0	1	0	0
С	1	1	1	0	1	0
D	0	0	0	1	0	1
Е	1	1	0	0	0	0

Motivation	Compatibility of characters	Perfect phylogeny building	Compatibility of trees	Conclusion
OO	00000	00●00		O
Perfect phylog Consecutive ones	eny building property			

- I have no idea if it can be useful for anything but it sounds a bit more like a "fun fact".
- Definition: all ones in a column are in consecutive rows.
- M has a perfect phylogeny ⇒ rows can be permuted so as to satisfy the consecutive ones property.

	2	6	1	3	5	4
А	1	1	1	0	0	0
В	0	0	0	1	0	0
С	1	1	1	0	1	0
D	0	0	0	1	0	1
Е	1	1	0	0	0	0

	Compatibility of characters	Perfect phylogeny building 00●00	Compatibility of trees	
Perfect phylog Consecutive ones	geny building property			

- I have no idea if it can be useful for anything but it sounds a bit more like a "fun fact".
- Definition: all ones in a column are in consecutive rows.
- \triangleright M has a perfect phylogeny \Rightarrow rows can be permuted so as to satisfy the consecutive ones property.

	2	6	1	3	5	4
А	1	1	1	0	0	0
В	0	0	0	1	0	0
С	1	1	1	0	1	0
D	0	0	0	1	0	1
Е	1	1	0	0	0	0

	Compatibility of characters	Perfect phylogeny building 00●00	Compatibility of trees	
Perfect phylog Consecutive ones	geny building property			

- I have no idea if it can be useful for anything but it sounds a bit more like a "fun fact".
- Definition: all ones in a column are in consecutive rows.
- \triangleright M has a perfect phylogeny \Rightarrow rows can be permuted so as to satisfy the consecutive ones property.

	2	6	1	3	5	4
А	1	1	1	0	0	0
В	0	0	0	1	0	0
С	1	1	1	0	1	0
D	0	0	0	1	0	1
Е	1	1	0	0	0	0

	Compatibility of characters	Perfect phylogeny building 00●00	Compatibility of trees	
Perfect phylog Consecutive ones	geny building property			

- I have no idea if it can be useful for anything but it sounds a bit more like a "fun fact".
- Definition: all ones in a column are in consecutive rows.
- \triangleright M has a perfect phylogeny \Rightarrow rows can be permuted so as to satisfy the consecutive ones property.

	2	6	1	3	5	4
D	0	0	0	1	0	1
А	1	1	1	0	0	0
В	0	0	0	1	0	0
С	1	1	1	0	1	0
Е	1	1	0	0	0	0

	Compatibility of characters	Perfect phylogeny building 00●00	Compatibility of trees	
Perfect phylog Consecutive ones	geny building property			

- I have no idea if it can be useful for anything but it sounds a bit more like a "fun fact".
- Definition: all ones in a column are in consecutive rows.
- \triangleright M has a perfect phylogeny \Rightarrow rows can be permuted so as to satisfy the consecutive ones property.

	2	6	1	3	5	4
D	0	0	0	1	0	1
А	1	1	1	0	0	0
В	0	0	0	1	0	0
С	1	1	1	0	1	0
Е	1	1	0	0	0	0

	Compatibility of characters	Perfect phylogeny building 00●00	Compatibility of trees	
Perfect phylog Consecutive ones	geny building property			

- I have no idea if it can be useful for anything but it sounds a bit more like a "fun fact".
- Definition: all ones in a column are in consecutive rows.
- \triangleright M has a perfect phylogeny \Rightarrow rows can be permuted so as to satisfy the consecutive ones property.

	2	6	1	3	5	4
С	1	1	1	0	1	0
D	0	0	0	1	0	1
А	1	1	1	0	0	0
В	0	0	0	1	0	0
Е	1	1	0	0	0	0
	Compatibility of characters	Perfect phylogeny building 00●00	Compatibility of trees			
------------------------------------	-----------------------------	-------------------------------------	------------------------	--		
Perfect phylog Consecutive ones	geny building property					

- I have no idea if it can be useful for anything but it sounds a bit more like a "fun fact".
- Definition: all ones in a column are in consecutive rows.
- \triangleright M has a perfect phylogeny \Rightarrow rows can be permuted so as to satisfy the consecutive ones property.

	2	6	1	3	5	4
С	1	1	1	0	1	0
D	0	0	0	1	0	1
А	1	1	1	0	0	0
В	0	0	0	1	0	0
Е	1	1	0	0	0	0
				٨		

	Compatibility of characters	Perfect phylogeny building 00●00	Compatibility of trees	
Perfect phylog Consecutive ones	geny building property			

- I have no idea if it can be useful for anything but it sounds a bit more like a "fun fact".
- Definition: all ones in a column are in consecutive rows.
- \triangleright M has a perfect phylogeny \Rightarrow rows can be permuted so as to satisfy the consecutive ones property.

	2	6	1	3	5	4
D	0	0	0	1	0	1
В	0	0	0	1	0	0
С	1	1	1	0	1	0
А	1	1	1	0	0	0
Е	1	1	0	0	0	0

	Compatibility of characters	Perfect phylogeny building 00●00	Compatibility of trees	
Perfect phylog Consecutive ones	geny building property			

- I have no idea if it can be useful for anything but it sounds a bit more like a "fun fact".
- Definition: all ones in a column are in consecutive rows.
- \triangleright M has a perfect phylogeny \Rightarrow rows can be permuted so as to satisfy the consecutive ones property.

	2	6	1	3	5	4
D	0	0	0	1	0	1
В	0	0	0	1	0	0
С	1	1	1	0	1	0
А	1	1	1	0	0	0
Е	1	1	0	0	0	0

	Compatibility of characters	Perfect phylogeny building 00●00	Compatibility of trees	
Perfect phylog Consecutive ones	geny building property			

- I have no idea if it can be useful for anything but it sounds a bit more like a "fun fact".
- Definition: all ones in a column are in consecutive rows.
- \triangleright M has a perfect phylogeny \Rightarrow rows can be permuted so as to satisfy the consecutive ones property.

	2	6	1	3	5	4
С	1	1	1	0	1	0
А	1	1	1	0	0	0
D	0	0	0	1	0	1
В	0	0	0	1	0	0
Е	1	1	0	0	0	0

	Compatibility of characters	Perfect phylogeny building 00●00	Compatibility of trees	
Perfect phylog Consecutive ones	geny building property			

- I have no idea if it can be useful for anything but it sounds a bit more like a "fun fact".
- Definition: all ones in a column are in consecutive rows.
- \triangleright M has a perfect phylogeny \Rightarrow rows can be permuted so as to satisfy the consecutive ones property.

	2	6	1	3	5	4
С	1	1	1	0	1	0
А	1	1	1	0	0	0
D	0	0	0	1	0	1
В	0	0	0	1	0	0
Е	1	1	0	0	0	0

	Compatibility of characters	Perfect phylogeny building 00●00	Compatibility of trees	
Perfect phylog Consecutive ones	geny building property			

- I have no idea if it can be useful for anything but it sounds a bit more like a "fun fact".
- Definition: all ones in a column are in consecutive rows.
- \triangleright M has a perfect phylogeny \Rightarrow rows can be permuted so as to satisfy the consecutive ones property.

	2	6	1	3	5	4
С	1	1	1	0	1	0
А	1	1	1	0	0	0
Е	1	1	0	0	0	0
D	0	0	0	1	0	1
В	0	0	0	1	0	0
	٨					

	Compatibility of characters	Perfect phylogeny building 00●00	Compatibility of trees	
Perfect phylog Consecutive ones	geny building property			

- I have no idea if it can be useful for anything but it sounds a bit more like a "fun fact".
- Definition: all ones in a column are in consecutive rows.
- \triangleright M has a perfect phylogeny \Rightarrow rows can be permuted so as to satisfy the consecutive ones property.

	2	6	1	3	5	4
С	1	1	1	0	1	0
А	1	1	1	0	0	0
Е	1	1	0	0	0	0
D	0	0	0	1	0	1
В	0	0	0	1	0	0

Motivation 00	Compatibility of characters	Perfect phylogeny building 000●0	Compatibility of trees	Conclusion O
Perfect phylog This algorithm is	geny building optimal			

Shows that all cells of M must be queried in order to know whether there exists a perfect phylogeny.

- And so Ω(nm) operations are required.
- The algorithm is optimal.

Motivation 00	Compatibility of characters	Perfect phylogeny building 000●0	Compatibility of trees	Conclusion O
Perfect phylog This algorithm is	geny building optimal			

- > Shows that all cells of *M* must be queried in order to know whether there exists a perfect phylogeny.
- And so $\Omega(nm)$ operations are required.
- The algorithm is optimal.

Motivation 00	Compatibility of characters	Perfect phylogeny building 000●0	Compatibility of trees	Conclusion O
Perfect phylog This algorithm is	geny building optimal			

- > Shows that all cells of *M* must be queried in order to know whether there exists a perfect phylogeny.
- And so $\Omega(nm)$ operations are required.
- The algorithm is optimal.

Motivation 00	Compatibility of characters	Perfect phylogeny building 0000●	Compatibility of trees	Conclusion O
Perfect phylog	eny building			
Extension to undi	rected characters			

- Majority choice: call ancestral state the majority state.
- There exists a perfect phylogeny for at least one of the 2^m states ⇒ there exists one for the majority choice.
- Alternatively: root at a given leaf.

Motivation 00	Compatibility of characters	Perfect phylogeny building 0000●	Compatibility of trees	Conclusion O
Perfect phylog Extension to undi	geny building irected characters			

Majority choice: call ancestral state the majority state.

- There exists a perfect phylogeny for at least one of the 2^m states ⇒ there exists one for the majority choice.
- Alternatively: root at a given leaf.

Motivation 00	Compatibility of characters	Perfect phylogeny building 0000●	Compatibility of trees	Conclusion O
Perfect phylog Extension to und	geny building irected characters			

- Majority choice: call ancestral state the majority state.
- ▶ There exists a perfect phylogeny for at least one of the 2^m states ⇒ there exists one for the majority choice.
- Alternatively: root at a given leaf.

Motivation	Compatibility of characters	Perfect phylogeny building	Compatibility of trees	Conclusion
00	00000	0000●		O
Perfect phylog Extension to undi	eny building rected characters			

- Majority choice: call ancestral state the majority state.
- ► There exists a perfect phylogeny for at least one of the 2^m states ⇒ there exists one for the majority choice.
- Alternatively: root at a given leaf.

Motivation 00	Compatibility of characters	Perfect phylogeny building	Compatibility of trees ●000	Conclusion O
Compatibility Definition	of trees			

- > T_3 refines T_1 if T_1 can be obtained by applying edge contractions to T_3 .
- T₁ and T_2 are compatible if there exists T_3 refining both T_1 and T_2 .



- Build M_1 and M_2 with internal nodes in columns and $M(i,j) = \mathbb{1}_i$ is subtended by j
- Build M₃, taking the union of columns of M₁ and M₂.
- Use the previous algo to find if M_3 has a perfect phylogeny: $O(n^2)$.

Motivation 00	Compatibility of characters	Perfect phylogeny building	Compatibility of trees ●○○○	Conclusion O
Compatibility Definition	of trees			

- > T_3 refines T_1 if T_1 can be obtained by applying edge contractions to T_3 .
- \succ T_1 and T_2 are compatible if there exists T_3 refining both T_1 and T_2 .



- Build M_1 and M_2 with internal nodes in columns and $M(i,j) = \mathbb{1}_i$ is subtended by j .
- Build M₃, taking the union of columns of M₁ and M₂.
- Use the previous algo to find if M_3 has a perfect phylogeny: $O(n^2)$.

Motivation 00	Compatibility of characters	Perfect phylogeny building	Compatibility of trees ●○○○	Conclusion O
Compatibility Definition	of trees			

- > T_3 refines T_1 if T_1 can be obtained by applying edge contractions to T_3 .
- \succ T_1 and T_2 are compatible if there exists T_3 refining both T_1 and T_2 .



- Build M_1 and M_2 with internal nodes in columns and $M(i,j) = \mathbb{1}_i$ is subtended by j.
- Build M₃, taking the union of columns of M₁ and M₂.
- Use the previous algo to find if M_3 has a perfect phylogeny: $O(n^2)$.

Motivation 00	Compatibility of characters	Perfect phylogeny building	Compatibility of trees ●○○○	Conclusion O
Compatibility Definition	of trees			

- > T_3 refines T_1 if T_1 can be obtained by applying edge contractions to T_3 .
- \succ T_1 and T_2 are compatible if there exists T_3 refining both T_1 and T_2 .



- Build M_1 and M_2 with internal nodes in columns and $M(i,j) = \mathbb{1}_i$ is subtended by j.
- Build M₃, taking the union of columns of M₁ and M₂
- Use the previous algo to find if M_3 has a perfect phylogeny: $O(n^2)$.

Motivation 00	Compatibility of characters	Perfect phylogeny building	Compatibility of trees ●○○○	Conclusion O
Compatibility Definition	of trees			

- > T_3 refines T_1 if T_1 can be obtained by applying edge contractions to T_3 .
- \succ T_1 and T_2 are compatible if there exists T_3 refining both T_1 and T_2 .



- Build M_1 and M_2 with internal nodes in columns and $M(i, j) = \mathbb{1}_i$ is subtended by j.
- Build M₃, taking the union of columns of M₁ and M₂.
- Use the previous algo to find if M_3 has a perfect phylogeny: $O(n^2)$.

Motivation 00	Compatibility of characters	Perfect phylogeny building	Compatibility of trees ●○○○	Conclusion O
Compatibility Definition	of trees			

- > T_3 refines T_1 if T_1 can be obtained by applying edge contractions to T_3 .
- \succ T_1 and T_2 are compatible if there exists T_3 refining both T_1 and T_2 .



- Build M_1 and M_2 with internal nodes in columns and $M(i, j) = \mathbb{1}_i$ is subtended by j.
- Build M₃, taking the union of columns of M₁ and M₂.
- Use the previous algo to find if M_3 has a perfect phylogeny: $O(n^2)$.

Motivation 00	Compatibility of characters	Perfect phylogeny building	Compatibility of trees ●○○○	Conclusion O
Compatibility Definition	of trees			

- > T_3 refines T_1 if T_1 can be obtained by applying edge contractions to T_3 .
- \succ T_1 and T_2 are compatible if there exists T_3 refining both T_1 and T_2 .



- Build M_1 and M_2 with internal nodes in columns and $M(i,j) = \mathbb{1}_i$ is subtended by j.
- Build M₃, taking the union of columns of M₁ and M₂.
- Use the previous algo to find if M_3 has a perfect phylogeny: $O(n^2)$.

Motivation 00	Compatibility of characters	Perfect phylogeny building	Compatibility of trees ○●○○	Conclusion O
Commentibility	of two of			

Compatibility of trees Step 1 : Initialization with a depth-first traversal

- identify $V_1(i)$, the leaf that taxon *i* is attached to in T_1 ,
- and N₁(i), the number of taxa attached to the same leaf as i,
- do the same for T₂.





	Compatibility of characters	Compatibility of trees ○●○○	
Compatibility	of trees		

Step 1 : Initialization with a depth-first traversal

identify V₁(i), the leaf that taxon i is attached to in T₁,

- and N₁(i), the number of taxa attached to the same leaf as i,
- do the same for T₂.





	Compatibility of characters		Compatibility of trees ○●○○	
Compatibility	of trees	al		

- identify $V_1(i)$, the leaf that taxon *i* is attached to in T_1 ,
- > and $N_1(i)$, the number of taxa attached to the same leaf as i,

do the same for T_2 .





	Compatibility of characters	Compatibility of trees ○●○○	
Compatibility	of trees		

1'

1'

3'

5'

5'

7'

9'

1

а

с

е

g

Step 1 : Initialization with a depth-first traversal

- identify $V_1(i)$, the leaf that taxon *i* is attached to in T_1 ,
- and $N_1(i)$, the number of taxa attached to the same leaf as i,
- do the same for T_2 .





Motivation 00	Compatibility of characters	Perfect phylogeny building	Compatibility of trees ○○●○	Conclusion O
Compatibility	of trees			

Step 2 : refine both trees using the other

▶ *j* = 1, *k* = 2

 $N_1(c) > N_2(c)$, refine T_1 using a subtree picked from T_2 .

 \triangleright $N_1(f) > N_2(f)$, refine T_1 using another subtree picked from T_2 .

i = 2, k = 1

 $M_2(a) > N_1(a)$, refine T_2 using a subtree picked from T_1 .



Motivation 00	Compatibility of characters	Perfect phylogeny building	Compatibility of trees ○○●○	Conclusion O
Compatibility	of trees			

 \triangleright $N_1(c) > N_2(c)$, refine T_1 using a subtree picked from T_2 .

 \triangleright $N_1(f) > N_2(f)$, refine T_1 using another subtree picked from T_2

i = 2, k = 1

 $M_2(a) > N_1(a)$, refine T_2 using a subtree picked from T_1 .



Motivation 00	Compatibility of characters	Perfect phylogeny building	Compatibility of trees ○○●○	Conclusion O
Compatibility	of trees			

 \triangleright $N_1(c) > N_2(c)$, refine T_1 using a subtree picked from T_2 .

 \triangleright $N_1(f) > N_2(f)$, refine T_1 using another subtree picked from T_2

i = 2, k = 1

 $N_2(a) > N_1(a)$, refine T_2 using a subtree picked from T_1 .



Motivation 00	Compatibility of characters	Perfect phylogeny building	Compatibility of trees ○○●○	Conclusion O
Compatibility	of trees			

 \triangleright $N_1(c) > N_2(c)$, refine T_1 using a subtree picked from T_2 .

 \triangleright $N_1(f) > N_2(f)$, refine T_1 using another subtree picked from T_2

i = 2, k = 1

 $T_{1}(a) > N_{1}(a)$, refine T_{2} using a subtree picked from T_{1} .



Motivation 00	Compatibility of characters	Perfect phylogeny building	Compatibility of trees ○○●○	Conclusion O	
Compatibility of trees					

 \triangleright $N_1(c) > N_2(c)$, refine T_1 using a subtree picked from T_2 .

 \triangleright $N_1(f) > N_2(f)$, refine T_1 using another subtree picked from T_2

i = 2, k = 1

 $T = N_2(a) > N_1(a)$, refine T_2 using a subtree picked from T_1 .



	Compatibility of characters	Compatibility of trees ○○●○	
Compatibility	y of trees both trees using the other		

- N₁(c) > N₂(c), refine T_1 using a subtree picked from T_2 .
- \triangleright $N_1(f) > N_2(f)$, refine T_1 using another subtree picked from T_2 .

i = 2, k = 1

• $N_2(a) > N_1(a)$, refine T_2 using a subtree picked from T_1



	Compatibility of characters	Compatibility of trees ○○●○	
Compatibility	of trees oth trees using the other		

- N₁(c) > N₂(c), refine T_1 using a subtree picked from T_2 .
- \triangleright $N_1(f) > N_2(f)$, refine T_1 using another subtree picked from T_2 .

i = 2, k = 1

• $N_2(a) > N_1(a)$, refine T_2 using a subtree picked from T_1



	Compatibility of characters	Compatibility of trees ○○●○	
Compatibility	of trees oth trees using the other		

- \triangleright $N_1(c) > N_2(c)$, refine T_1 using a subtree picked from T_2 .
- \triangleright $N_1(f) > N_2(f)$, refine T_1 using another subtree picked from T_2 .

▶ j = 2, k = 1

 $M_2(a) > N_1(a)$, refine T_2 using a subtree picked from T_1 .



	Compatibility of characters	Compatibility of trees ○○●○	
Compatibility	of trees		

j = 1, k = 2
N₁(c) > N₂(c), refine T₁ using a subtree picked from T₂.
N₁(f) > N₂(f), refine T₁ using another subtree picked from T₂.
j = 2, k = 1
N₂(a) > N₁(a), refine T₂ using a subtree picked from T₁.



	Compatibility of characters	Compatibility of trees ○○●○	
Compatibility	y of trees		

j = 1, k = 2
N₁(c) > N₂(c), refine T₁ using a subtree picked from T₂.
N₁(f) > N₂(f), refine T₁ using another subtree picked from T₂.
j = 2, k = 1
N₂(a) > N₁(a), refine T₂ using a subtree picked from T₁.



	Compatibility of characters	Compatibility of trees ○○●○	
Compatibility	y of trees		

j = 1, k = 2
N₁(c) > N₂(c), refine T₁ using a subtree picked from T₂.
N₁(f) > N₂(f), refine T₁ using another subtree picked from T₂.
j = 2, k = 1
N₂(a) > N₁(a), refine T₂ using a subtree picked from T₁.


	Compatibility of characters	Compatibility of trees 000●	
Compatibility Step 3 : compare	of trees refinements		

build a list of leaves of T_1 whose siblings are leaves.

pick cherries, compare to T₂, delete if they are identical, stop otherwise.





	Compatibility of characters	Compatibility of trees 000●	
Compatibility Step 3 : compare	of trees e refinements		

- build a list of leaves of T_1 whose siblings are leaves.
- \triangleright pick cherries, compare to T_2 , delete if they are identical, stop otherwise.





	Compatibility of characters	Compatibility of trees 000●	
Compatibility Step 3 : compare	of trees e refinements		

- build a list of leaves of T_1 whose siblings are leaves.
- \triangleright pick cherries, compare to T_2 , delete if they are identical, stop otherwise.





	Compatibility of characters	Compatibility of trees 000●	
Compatibility Step 3 : compare	of trees e refinements		

- build a list of leaves of T_1 whose siblings are leaves.
- **b** pick cherries, compare to T_2 , delete if they are identical, stop otherwise.





	Compatibility of characters	Compatibility of trees 000●	
Compatibility Step 3 : compare	of trees refinements		

- build a list of leaves of T_1 whose siblings are leaves.
- \triangleright pick cherries, compare to T_2 , delete if they are identical, stop otherwise.





	Compatibility of characters	Compatibility of trees 000●	
Compatibility Step 3 : compare	of trees refinements		

- build a list of leaves of T_1 whose siblings are leaves.
- **b** pick cherries, compare to T_2 , delete if they are identical, stop otherwise.





	Compatibility of characters	Compatibility of trees 000●	
Compatibility Step 3 : compare	of trees refinements		

- build a list of leaves of T_1 whose siblings are leaves.
- \triangleright pick cherries, compare to T_2 , delete if they are identical, stop otherwise.





Motivation 00	Compatibility of characters	Perfect phylogeny building	Compatibility of trees 000●	Conclusion O
Compatibility Step 3 : compare	of trees refinements			

- build a list of leaves of T₁ whose siblings are leaves.
- **b** pick cherries, compare to T_2 , delete if they are identical, stop otherwise.

 $\mathbf{6}$

6

Motivation 00	Compatibility of characters	Perfect phylogeny building	Compatibility of trees 000●	Conclusion O
Compatibility Step 3 : compare	of trees refinements			

- build a list of leaves of T₁ whose siblings are leaves.
- **b** pick cherries, compare to T_2 , delete if they are identical, stop otherwise.

 $\mathbf{6}$

 $\mathbf{6}$

	Perfect phylogeny building	Conclusion
		•

Take-home message

- 1. Presence/absence data for a m characters among n taxa.
- 2. Optimized algo in O(nm) to find if there exists a perfect phylogeny,
- 3. ... and to reconstruct it if it exists.

Opportunities for application:

- 1. On anything that can be modeled using the very strong assumption of an infinite sites model.
- 2. On barcode data in developmental biology.
- 3. As a first step to constrain a bit the exploration of the tree topology in phylogenetics/phylodynamics.

	Perfect phylogeny building	Conclusion
		•

Take-home message

- 1. Presence/absence data for a m characters among n taxa.
- 2. Optimized algo in O(nm) to find if there exists a perfect phylogeny,
- 3. ... and to reconstruct it if it exists.

Opportunities for application:

- 1. On anything that can be modeled using the very strong assumption of an infinite sites model.
- 2. On barcode data in developmental biology.
- 3. As a first step to constrain a bit the exploration of the tree topology in phylogenetics/phylodynamics.

	Perfect phylogeny building	Conclusion
		•

Take-home message

- 1. Presence/absence data for a m characters among n taxa.
- 2. Optimized algo in O(nm) to find if there exists a perfect phylogeny,
- 3. ... and to reconstruct it if it exists.

Opportunities for application:

- 1. On anything that can be modeled using the very strong assumption of an infinite sites model.
- 2. On barcode data in developmental biology.
- 3. As a first step to constrain a bit the exploration of the tree topology in phylogenetics/phylodynamics.

	Perfect phylogeny building	Conclusion
		•

Take-home message

- 1. Presence/absence data for a m characters among n taxa.
- 2. Optimized algo in O(nm) to find if there exists a perfect phylogeny,
- 3. ... and to reconstruct it if it exists.

Opportunities for application:

- 1. On anything that can be modeled using the very strong assumption of an infinite sites model.
- 2. On barcode data in developmental biology.
- 3. As a first step to constrain a bit the exploration of the tree topology in phylogenetics/phylodynamics.

	Perfect phylogeny building	Conclusion
		•

Take-home message

- 1. Presence/absence data for a m characters among n taxa.
- 2. Optimized algo in O(nm) to find if there exists a perfect phylogeny,
- 3. ... and to reconstruct it if it exists.

Opportunities for application:

- 1. On anything that can be modeled using the very strong assumption of an infinite sites model.
- 2. On barcode data in developmental biology.
- 3. As a first step to constrain a bit the exploration of the tree topology in phylogenetics/phylodynamics.

	Perfect phylogeny building	Conclusion
		•

Take-home message

- 1. Presence/absence data for a m characters among n taxa.
- 2. Optimized algo in O(nm) to find if there exists a perfect phylogeny,
- 3. ... and to reconstruct it if it exists.

Opportunities for application:

- 1. On anything that can be modeled using the very strong assumption of an infinite sites model.
- 2. On barcode data in developmental biology.
- 3. As a first step to constrain a bit the exploration of the tree topology in phylogenetics/phylodynamics.

	Perfect phylogeny building	Conclusion
		•

Take-home message

- 1. Presence/absence data for a m characters among n taxa.
- 2. Optimized algo in O(nm) to find if there exists a perfect phylogeny,
- 3. ... and to reconstruct it if it exists.

Opportunities for application:

- 1. On anything that can be modeled using the very strong assumption of an infinite sites model.
- 2. On barcode data in developmental biology.
- 3. As a first step to constrain a bit the exploration of the tree topology in phylogenetics/phylodynamics.

	Perfect phylogeny building	Conclusion
		•

Take-home message

- 1. Presence/absence data for a m characters among n taxa.
- 2. Optimized algo in O(nm) to find if there exists a perfect phylogeny,
- 3. ... and to reconstruct it if it exists.

Opportunities for application:

- 1. On anything that can be modeled using the very strong assumption of an infinite sites model.
- 2. On barcode data in developmental biology.
- 3. As a first step to constrain a bit the exploration of the tree topology in phylogenetics/phylodynamics.