

Buscas exatas:

Paisagem:

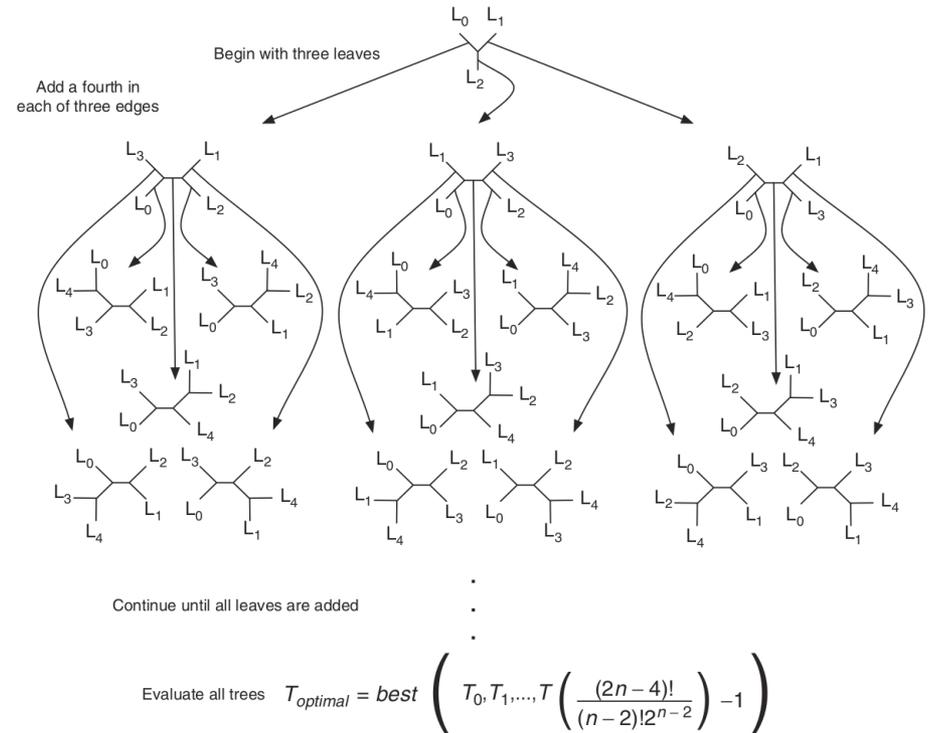
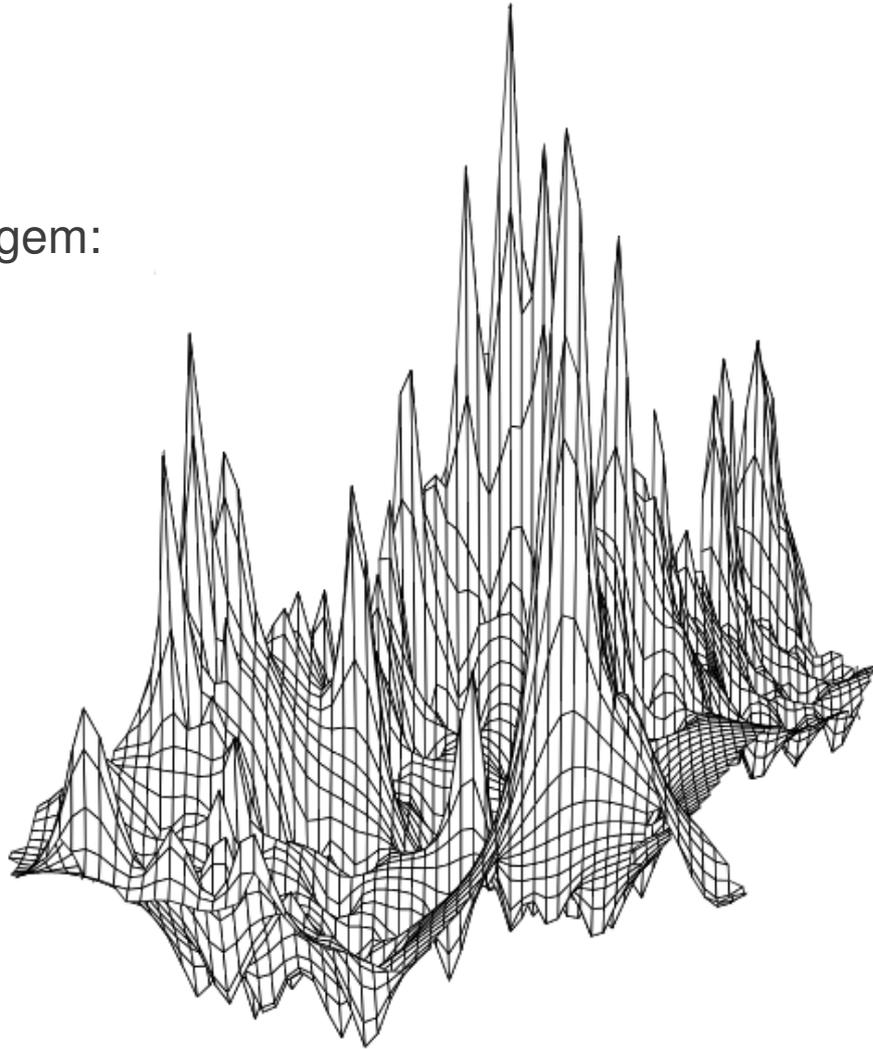
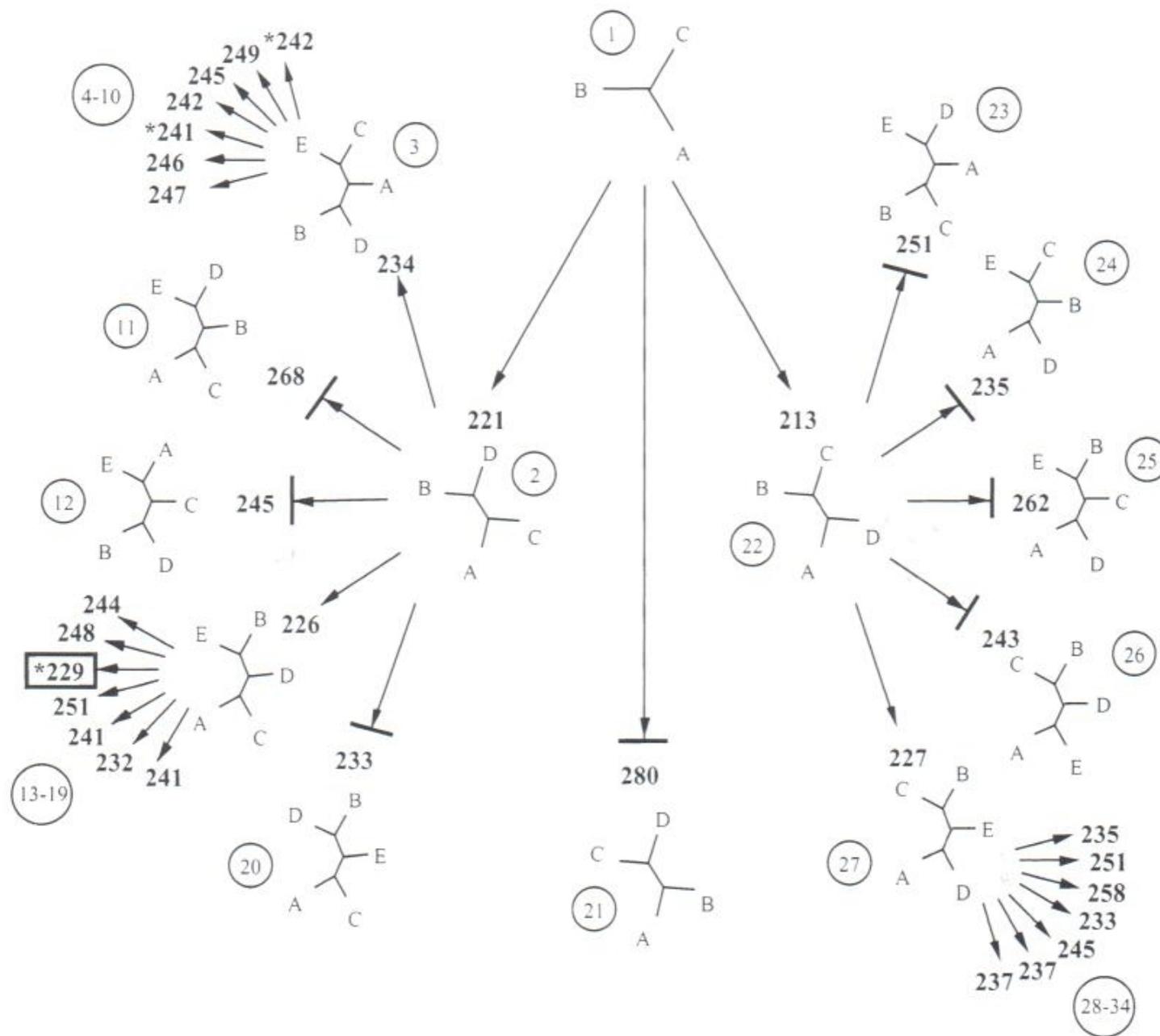
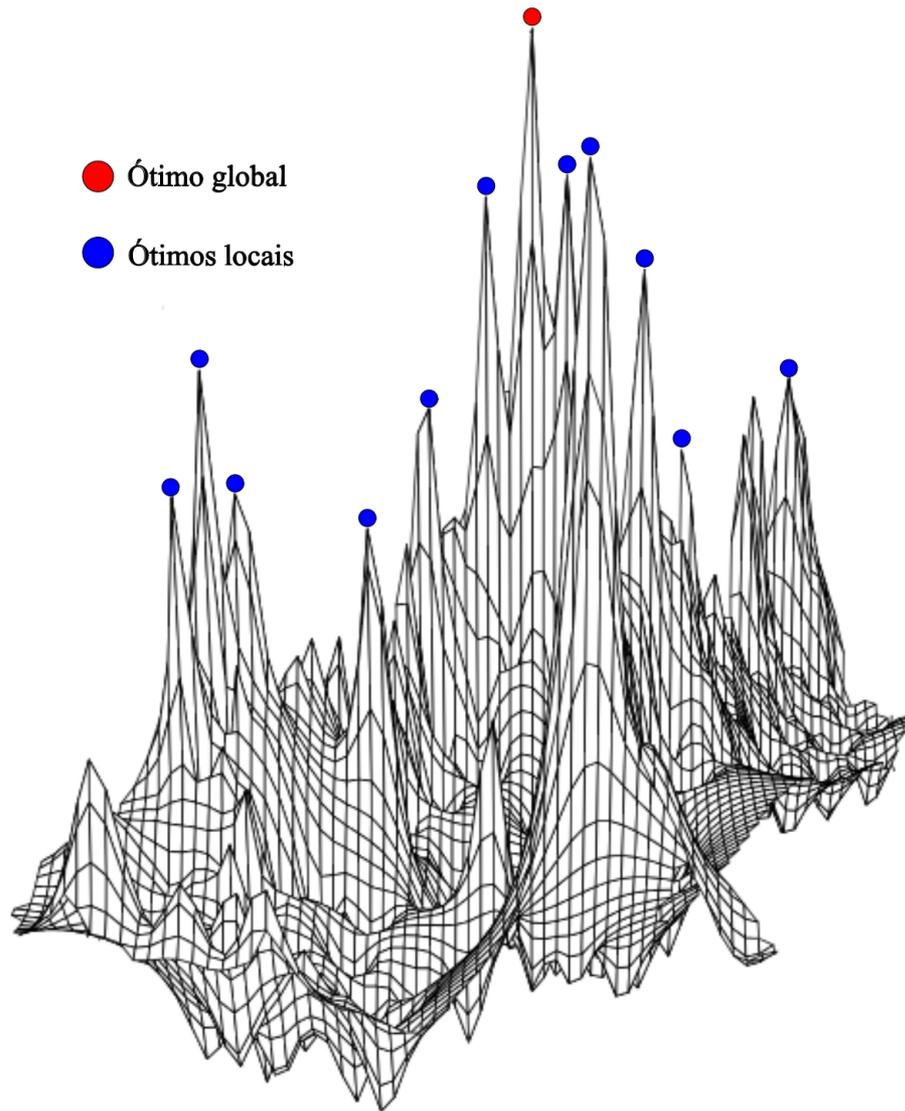


Figure 14.1: Explicit enumeration and evaluation of all trees.

Enumeração implícita



Buscas heurísticas:



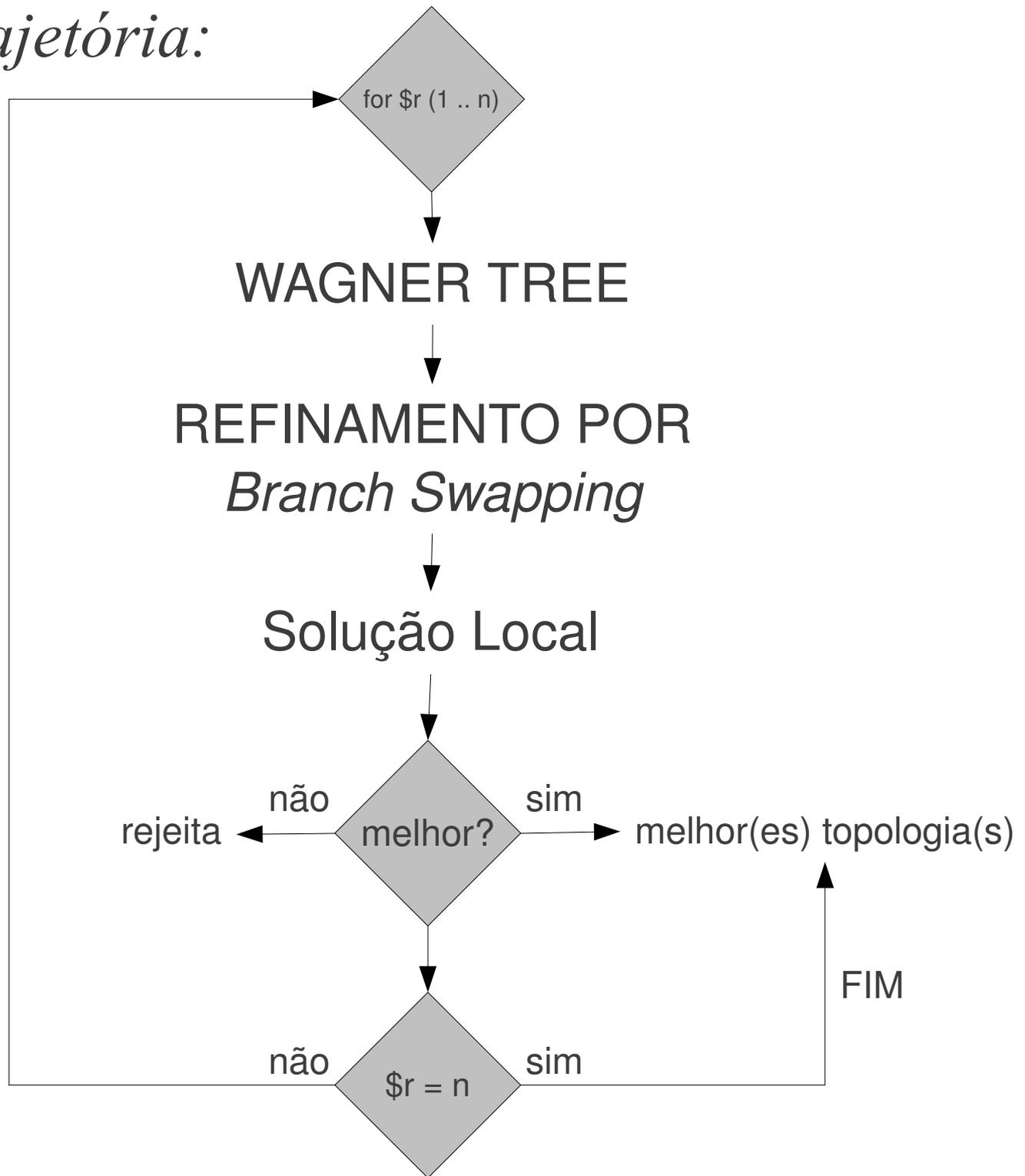
Desafios:

- Explorar adequadamente a paisagem.
- Evitar ótimos locais
- Eficiência computacional

Implementações:

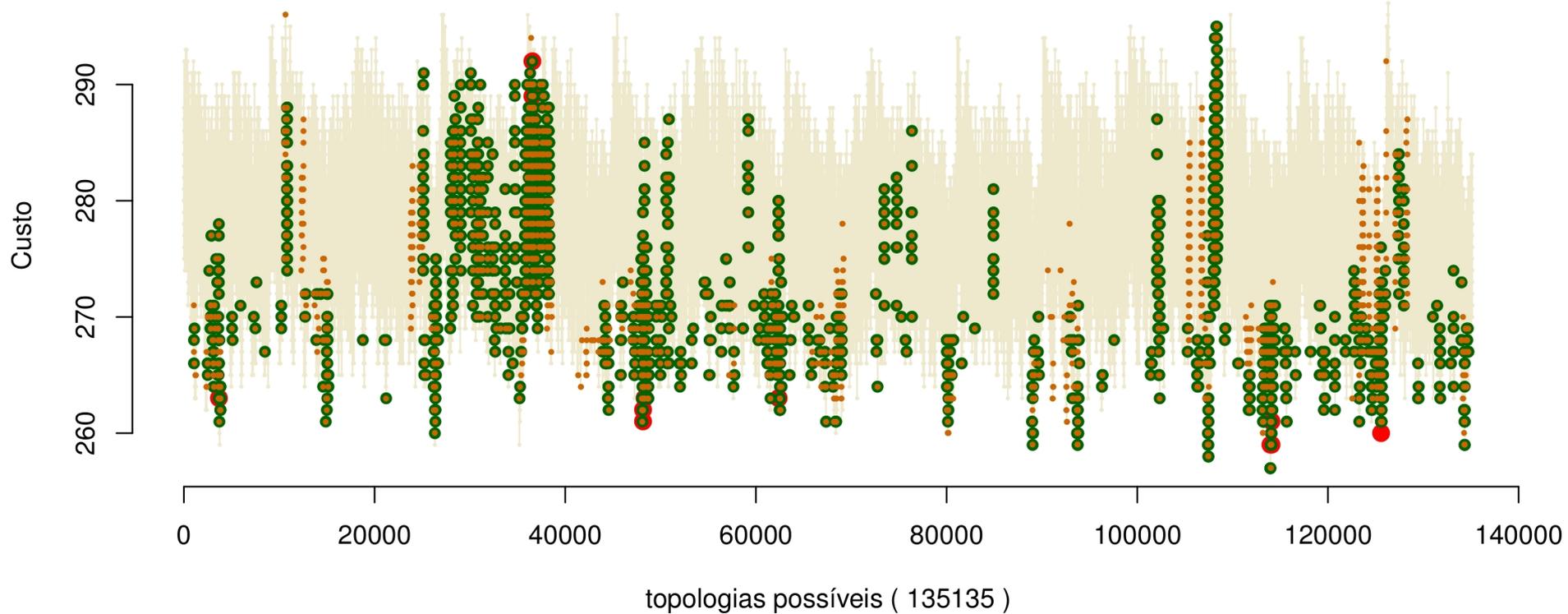
- Buscas por trajetórias
- Aleatorização
- Perturbação
- Trajetoárias subótimas
- Alg. genéticos

Buscas por trajetória:

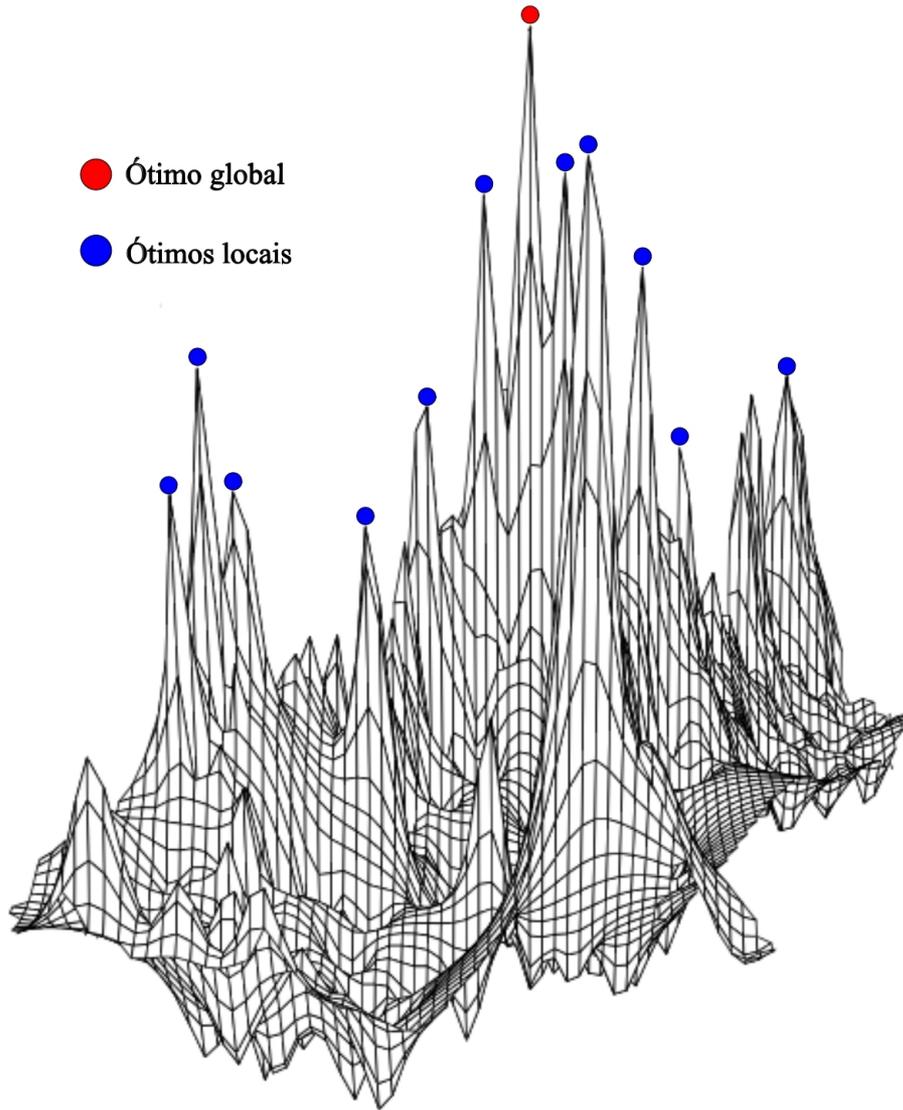


Buscas por trajetória: 10 RAS + TBR

Tree space



Buscas heurísticas:



Desafios:

- Explorar adequadamente a paisagem.
- Evitar ótimos locais
- Eficiência computacional

Implementações:

- Buscas por trajetórias
- Aleatorização
- Perturbação
- Trajetoárias subótimas
- Alg. genéticos

Buscas heurísticas: por perturbação

ITERAÇÃO:

1. Define uma solução inicial (e.g., via RAS+TBR).
2. Pesagem aleatória de uma fração dos caracteres por algum fator (geralmente 5-10% por 2).
3. Refinamento via *swapping*.
4. Retorna a pesagem inicial.
5. Refinamento via *swapping*.
6. Repete o ciclo por 1 a k vezes.
7. Compara com a solução inicial.

“ratchet”:

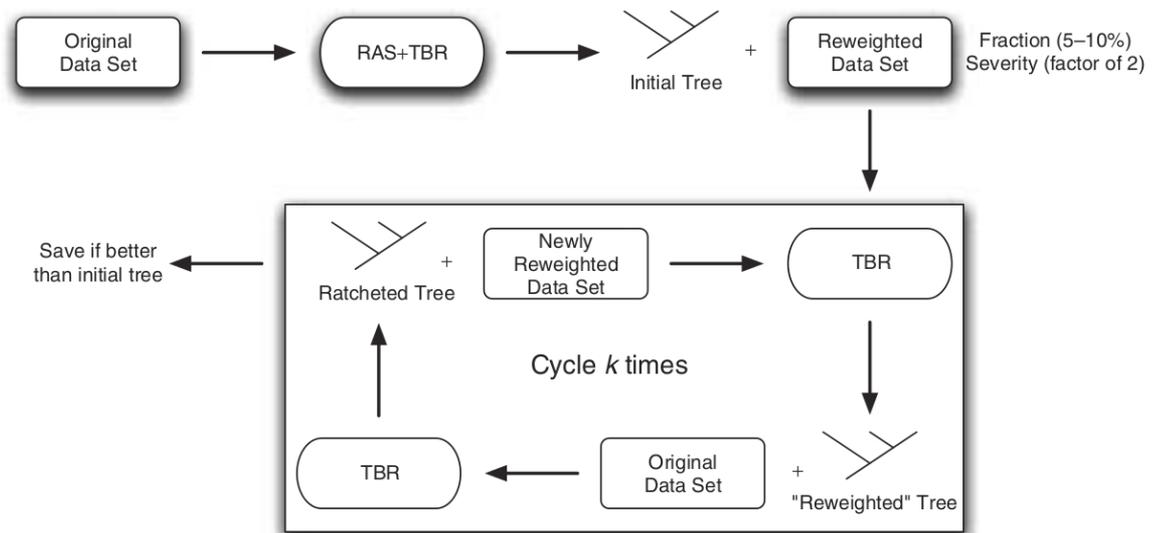


Figure 14.10: Ratchet refinement procedure of Nixon (1999).

Buscas heurísticas: buscas setoriais

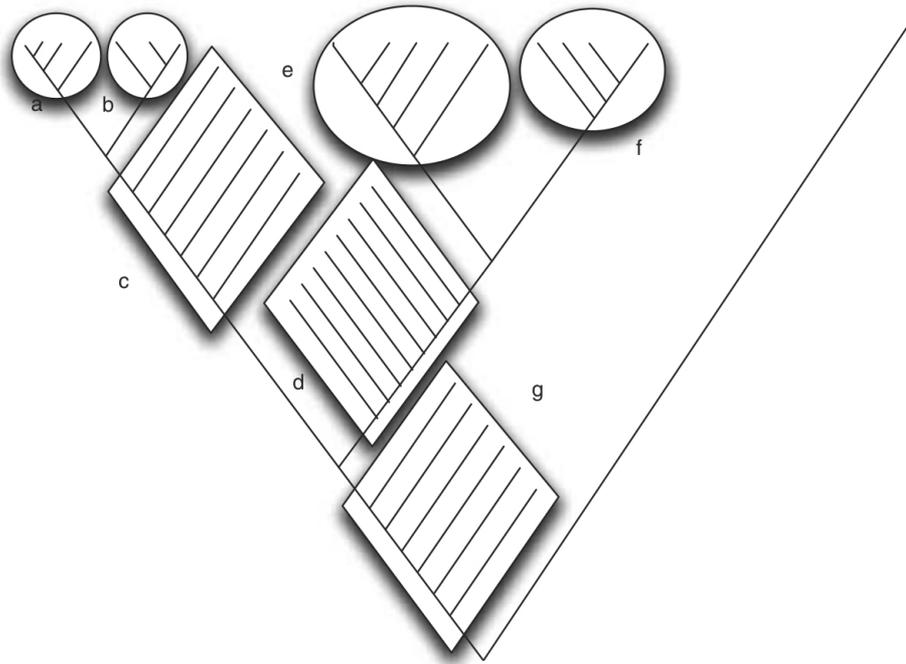


Figure 14.11: A tree showing several potential sectors (a-g).

“sectorial searches”:

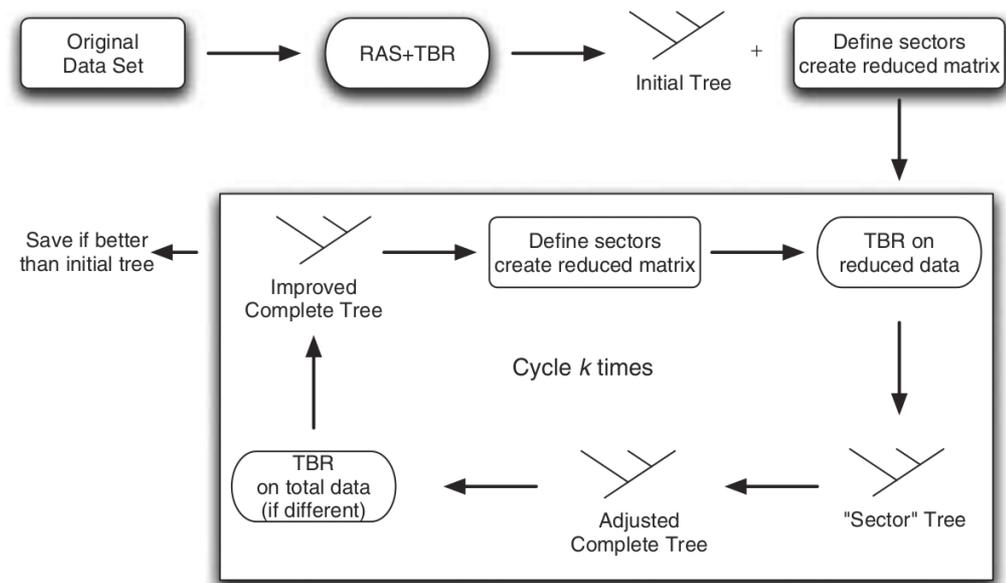


Figure 14.12: Sectorial Search refinement procedure of Goloboff (1999a).

Buscas heurísticas: simulated annealing

TBR convencional: Para $T =$ topologia inicial e $T' =$ rearranjo, T' é aceita se, e somente se, $T'_{custo} < T_{custo}$

“Tree-drifiting”:

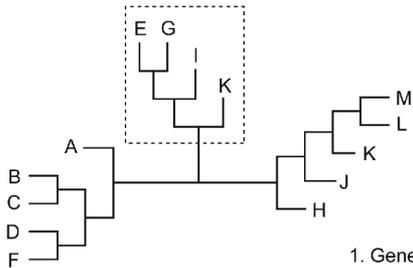
T' is accepted if $\left[\frac{F - C}{F} \leq \frac{RAND(0, 99)}{F - C + T'_{custo} - T_{custo}} \right]$ when $(T'_{custo} > T_{custo})$

$$F = \sum_i^{if(T_{custo}^i - T'_{custo}^i) > 0} T_{custo}^i - T'_{custo}^i$$

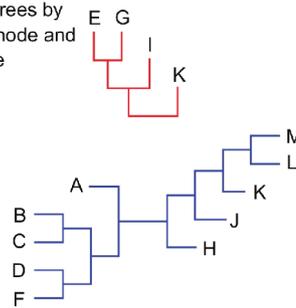
$$C = \sum_i^{if(T'_{custo}^i - T_{custo}^i) > 0} T'_{custo}^i - T_{custo}^i$$

$RAND(x, y) =$ a random integer $\in [x, y]$

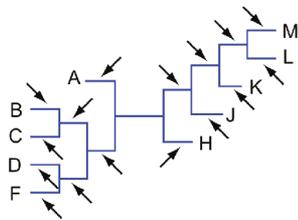
0. Starting tree



1. Generate two subtrees by breaking an internal node and re-rooting the subtree



2. Try to insert all possible rooted red subtrees at each node of the blue subtree



3. Evaluate the new tree according to the optimality criterion selected

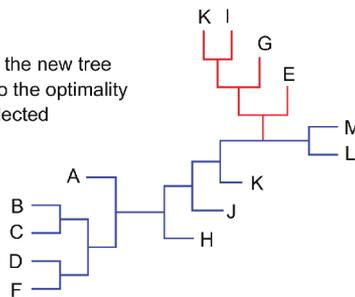


Figure 2. TBR branch swapping. An initial tree (0) gets broken into two subtrees (1). The red subtree is re-rooted on each possible internal branch and inserted in each possible branch of the blue subtree (arrows in step 2) and the resulting tree is evaluated (3).

Buscas heurísticas: algoritmos genéticos

COMPONENTES:

1. Inicialização: Uma população de soluções (topologias) é gerada.
2. Avaliação: Custos são atribuídos às soluções potenciais (“*aptidão*”).
3. Seleção: As soluções mais aptas (menores custos) são replicadas de forma que quanto maior sua *aptidão*, mas representativa elas são na população de soluções.
4. Recombinação: Componentes de duas ou mais soluções candidatas são combinadas para criar variação e potencialmente soluções melhores.
5. Mutação: Modificações são feitas em soluções individuais.
6. Reposição: As soluções criadas por seleção, recombinação e mutação substituem a geração existente na população de soluções.

“*Tree-fusing*”:

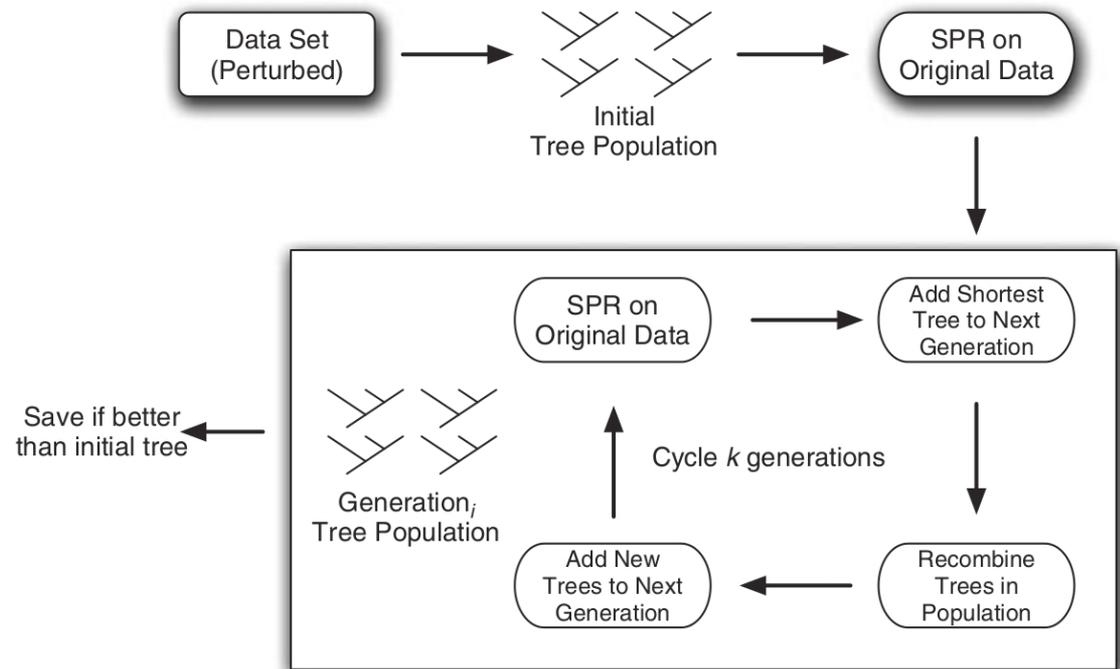
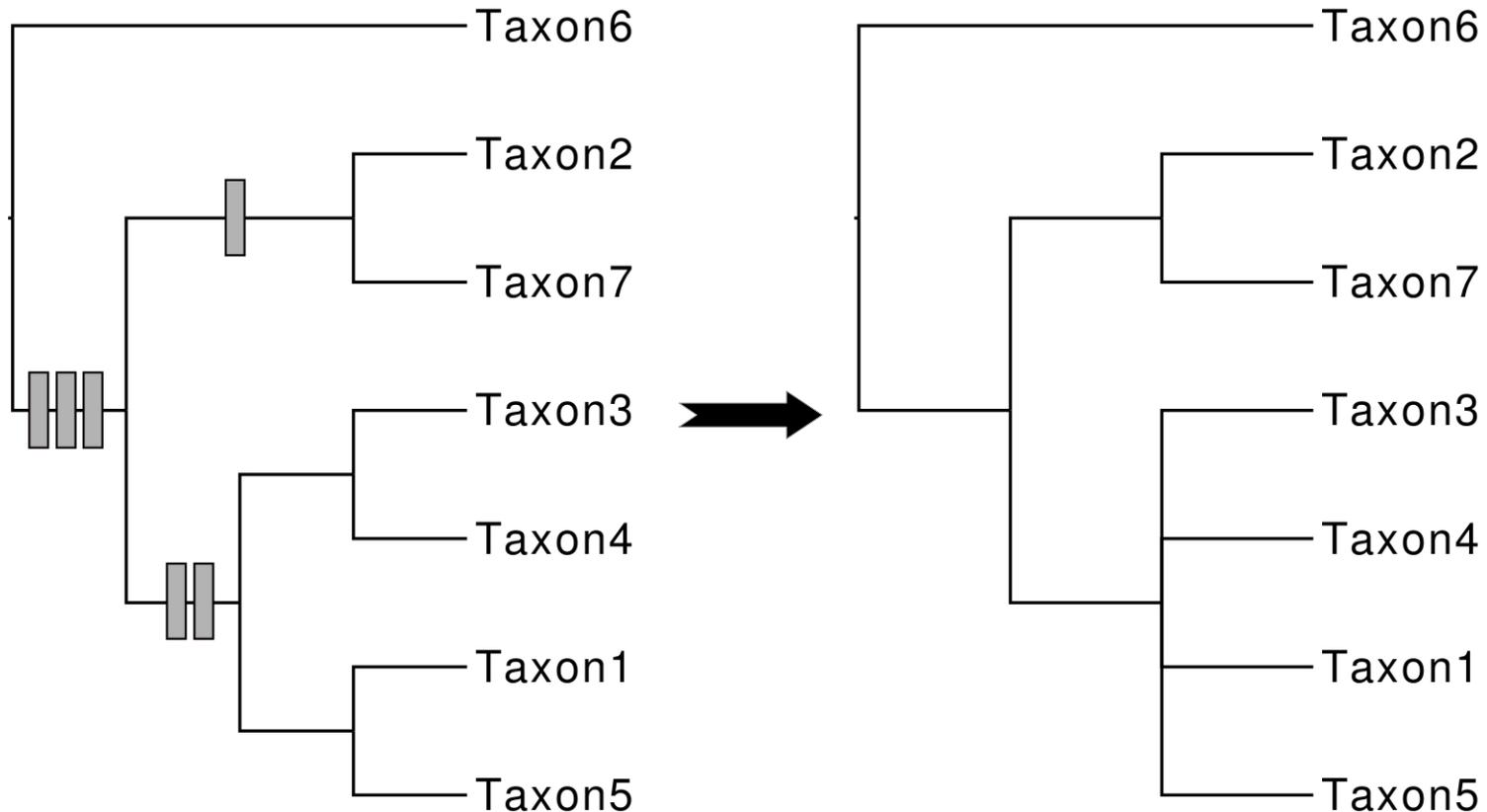


Figure 14.14: Genetic Algorithm tree search procedure of Moilanen (1999).

Diagnose:

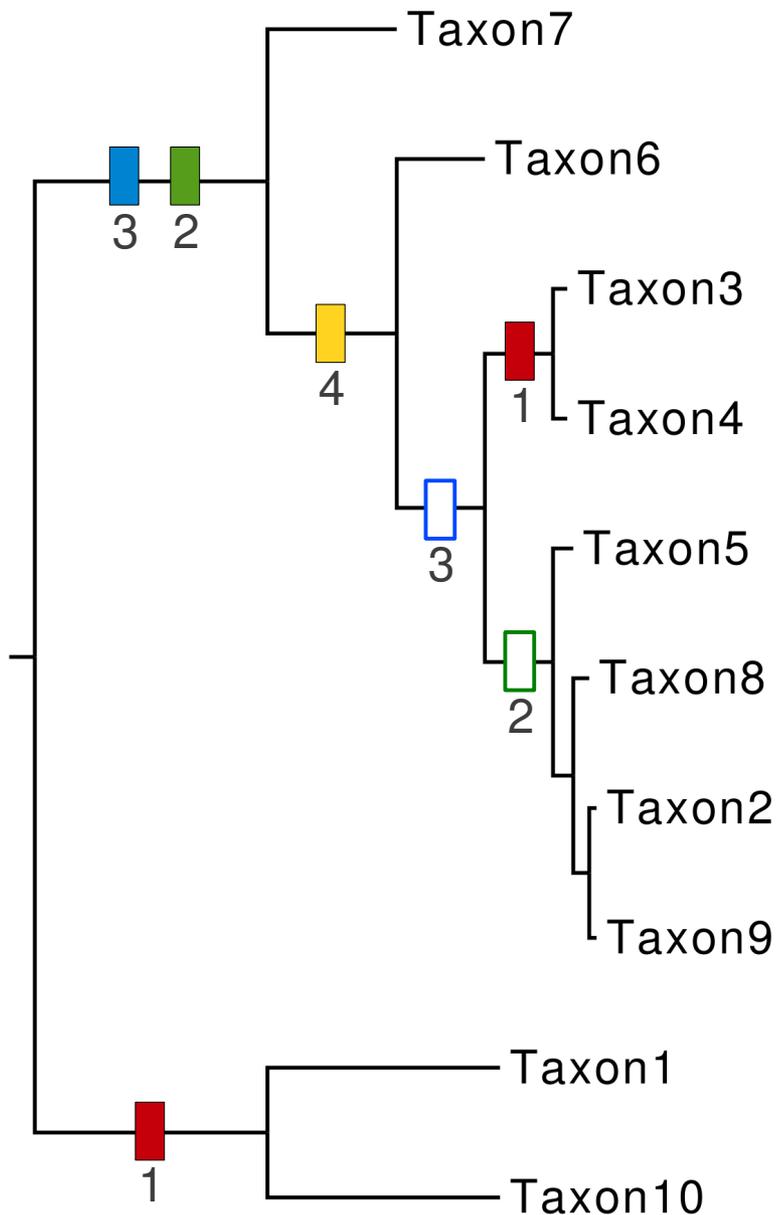
Suporte e resolução



1. Não existe suporte (evidência) para os grupos: (Taxon2,Taxon4) e (Taxon1,Taxon5).
2. Não há resolução para o nó compartilhado pelos terminais: Taxon2, Taxon4, Taxon1 e Taxon5.
3. A resolução de um nó depende da existência de pelo menos uma transformação.

Diagnose:

Suporte e resolução

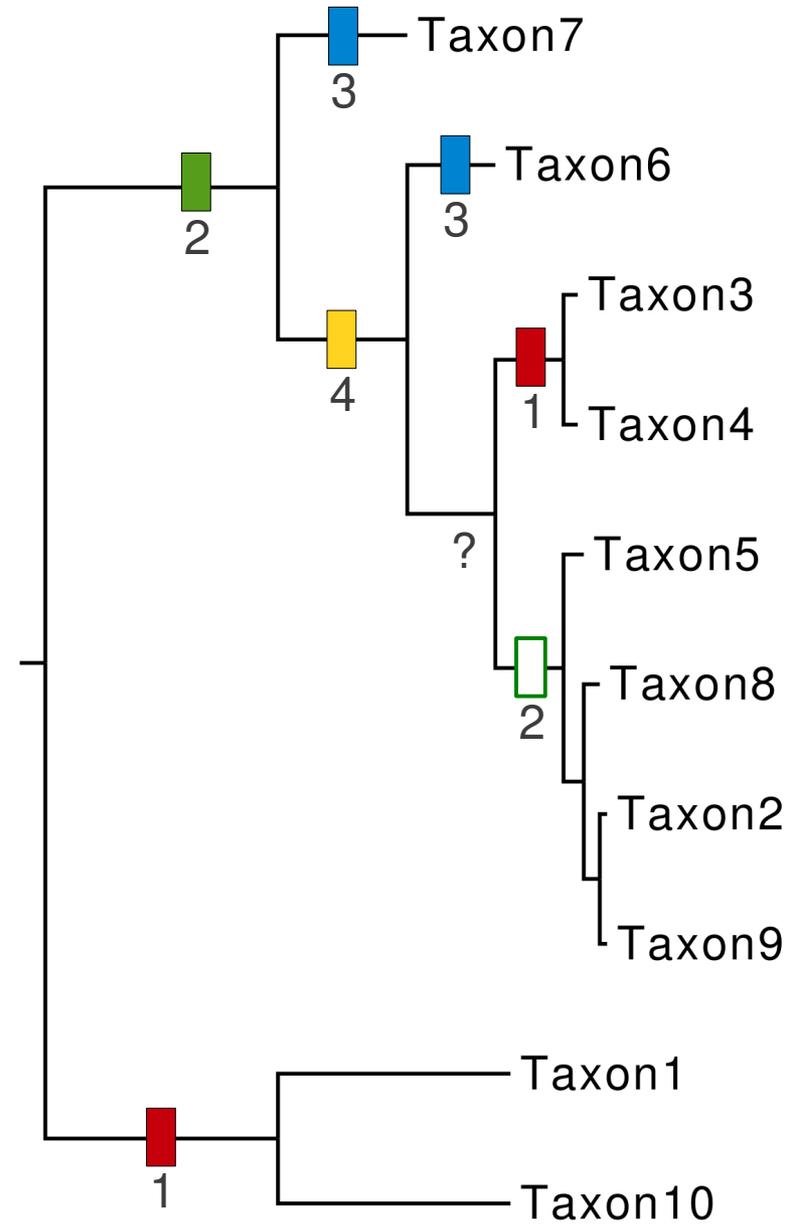
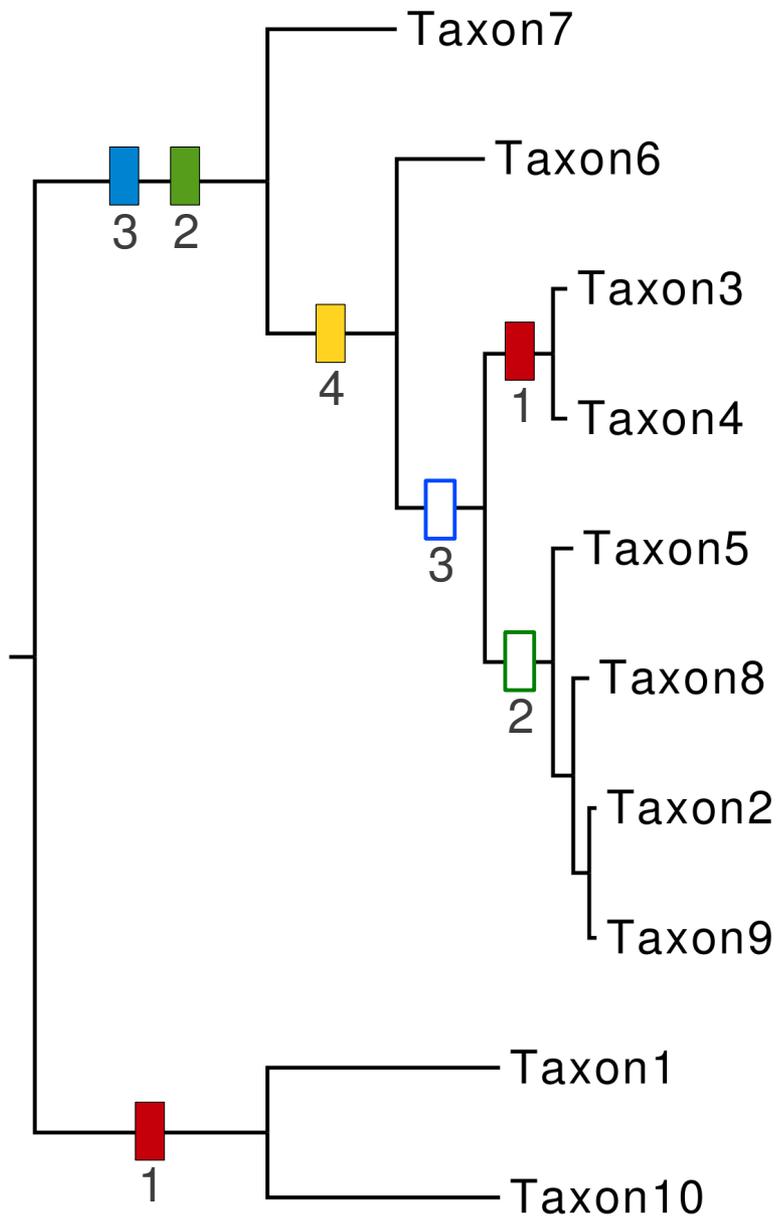


Considere os nós que estão sendo sustentados por transformações e responda:

Existe algum nó com suporte ambíguo neste cladograma?

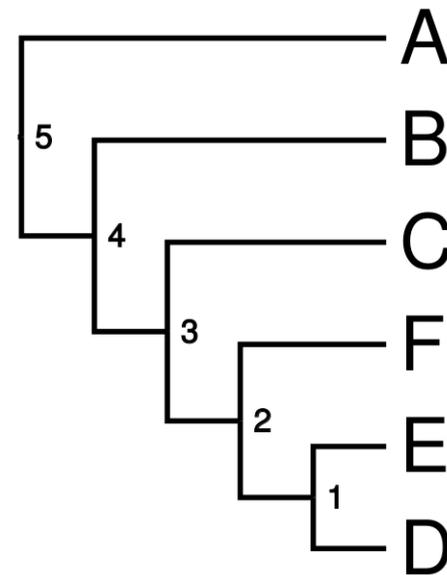
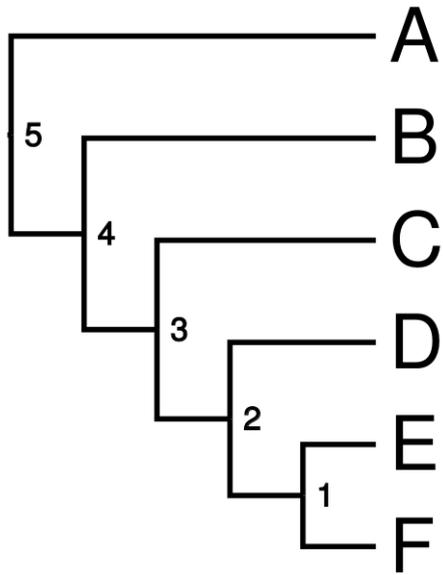
Diagnose:

Suporte e resolução



Consenso

Considere os seguintes cladogramas fundamentais:



Qual é o conteúdo informativo comum destas topologias?

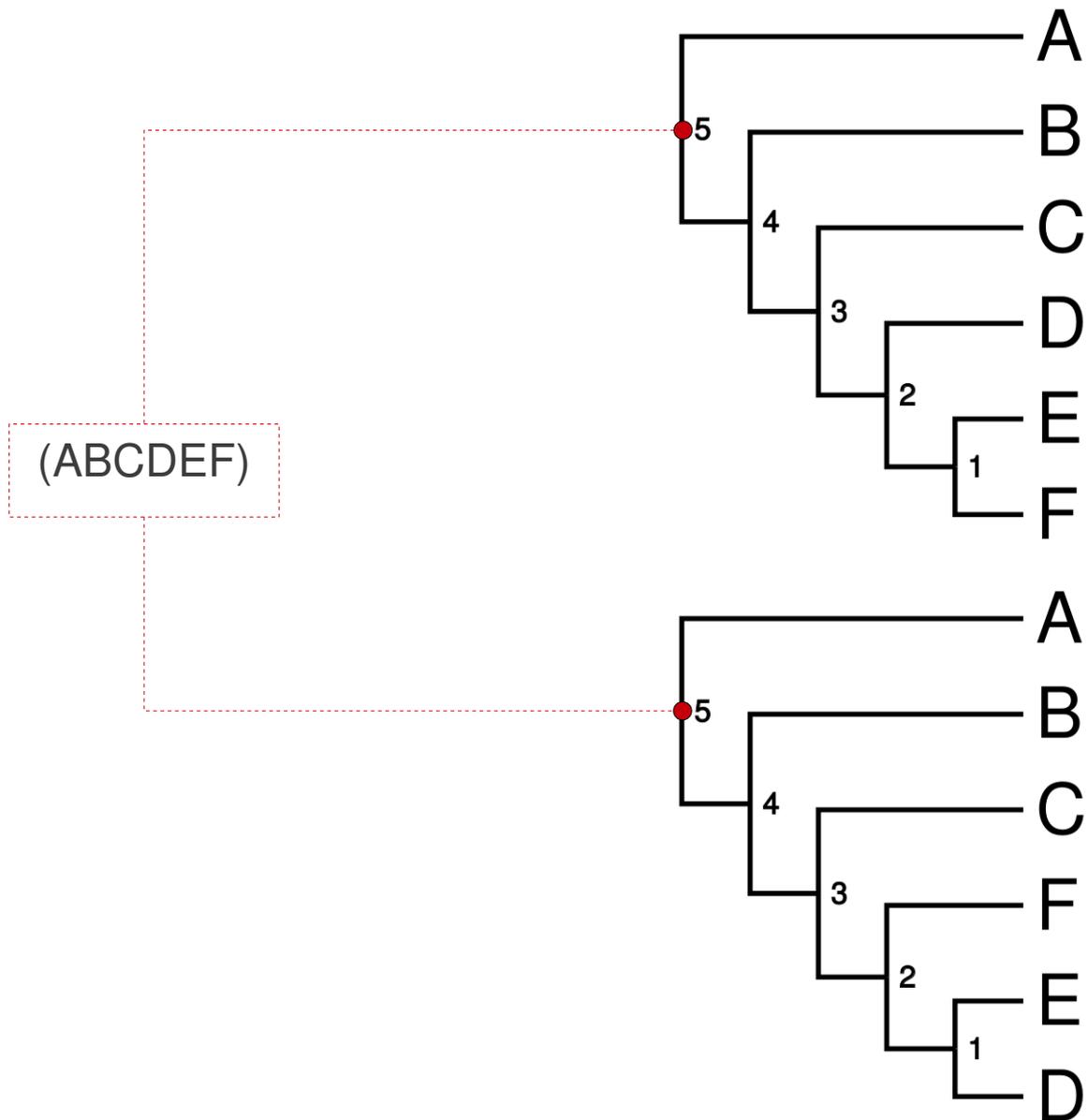
Quais componentes são comuns entre estas duas topologias?

Consenso

Considere os seguintes cladogramas fundamentais:

Componentes
comuns:

(ABCDEF)



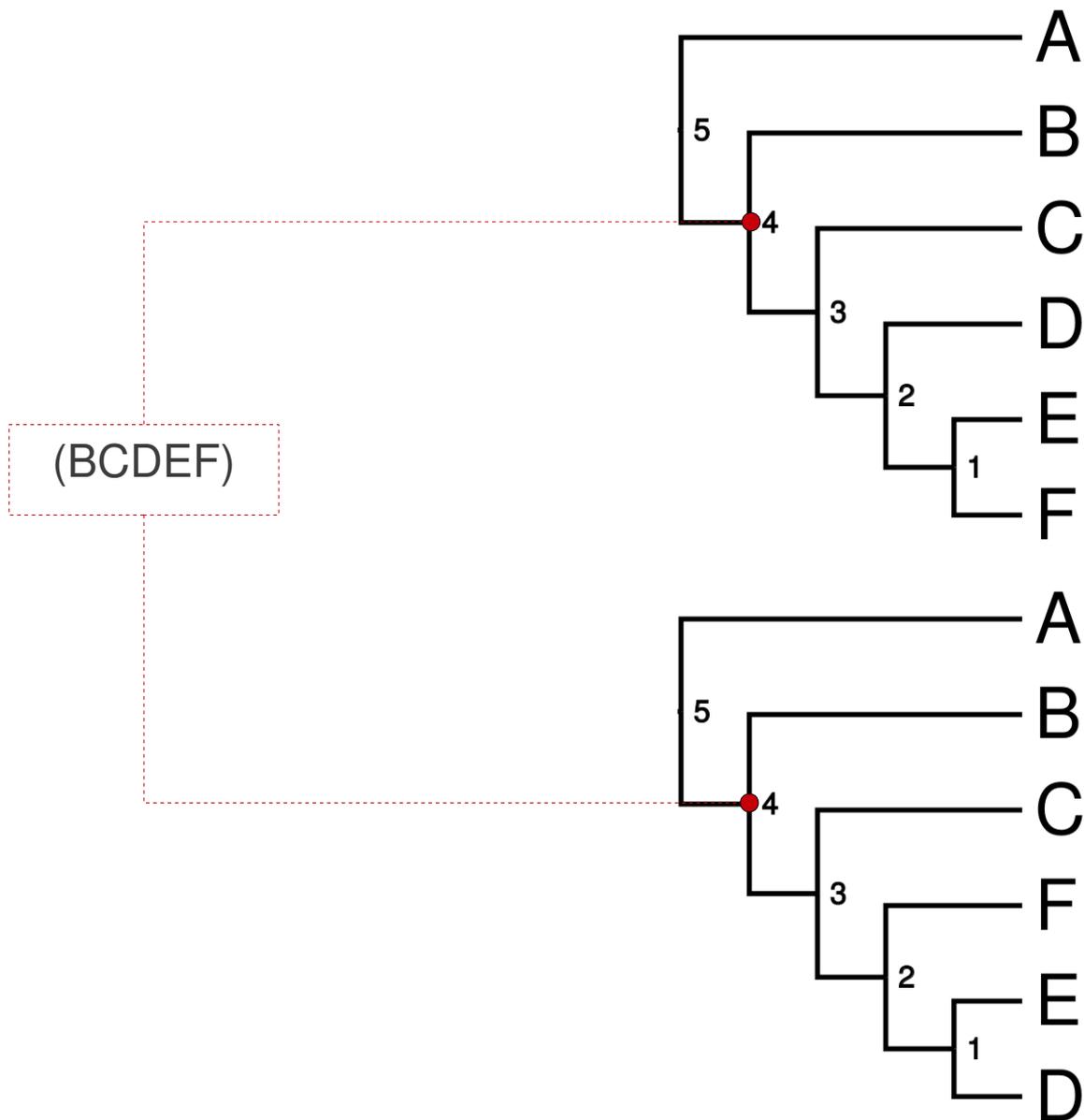
Consenso

Considere os seguintes cladogramas fundamentais:

Componentes
comuns:

(ABCDEF)

(BCDEF)

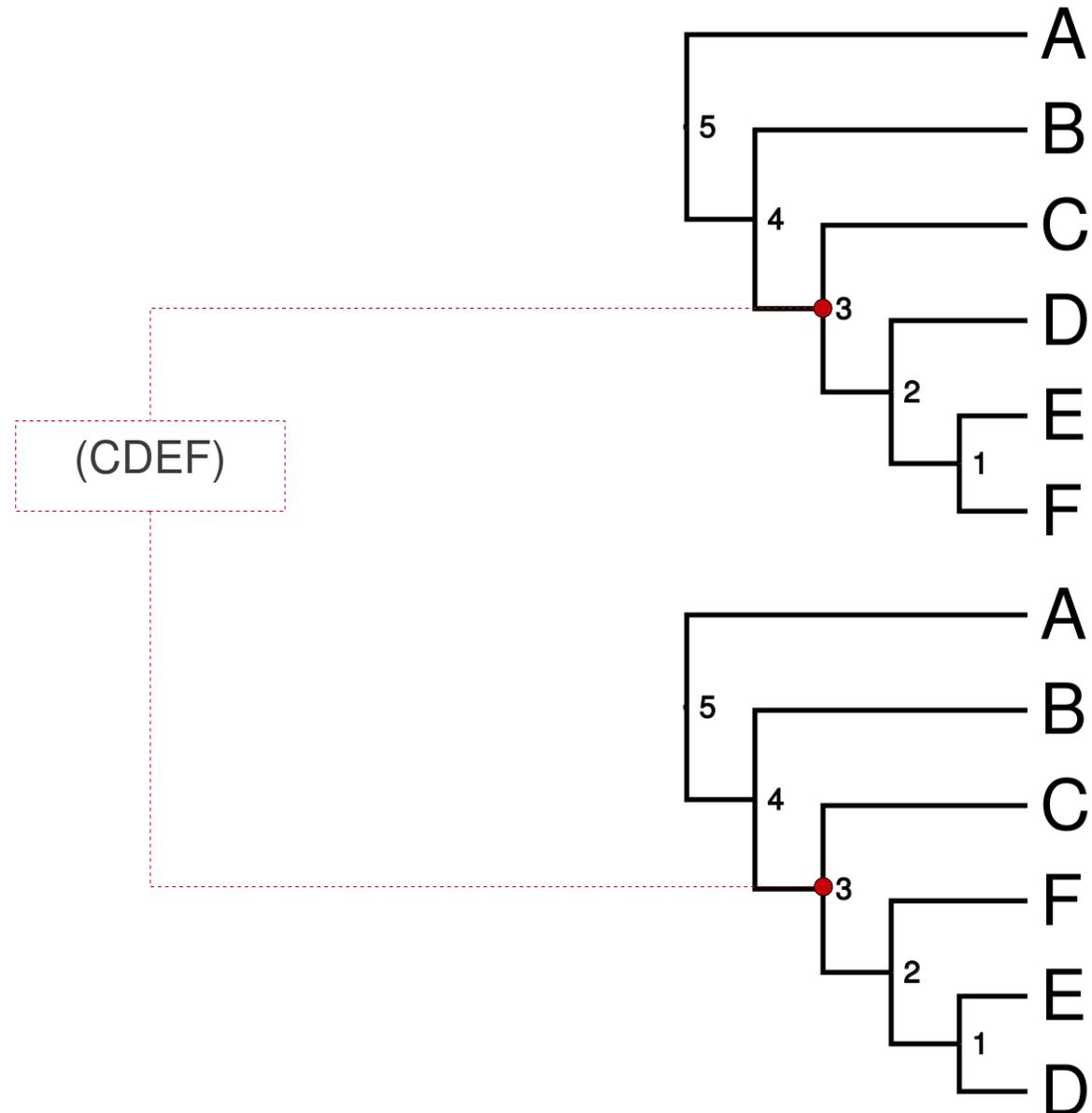


Consenso

Considere os seguintes cladogramas fundamentais:

Componentes
comuns:

(ABCDEF)
(BCDEF)
(CDEF)

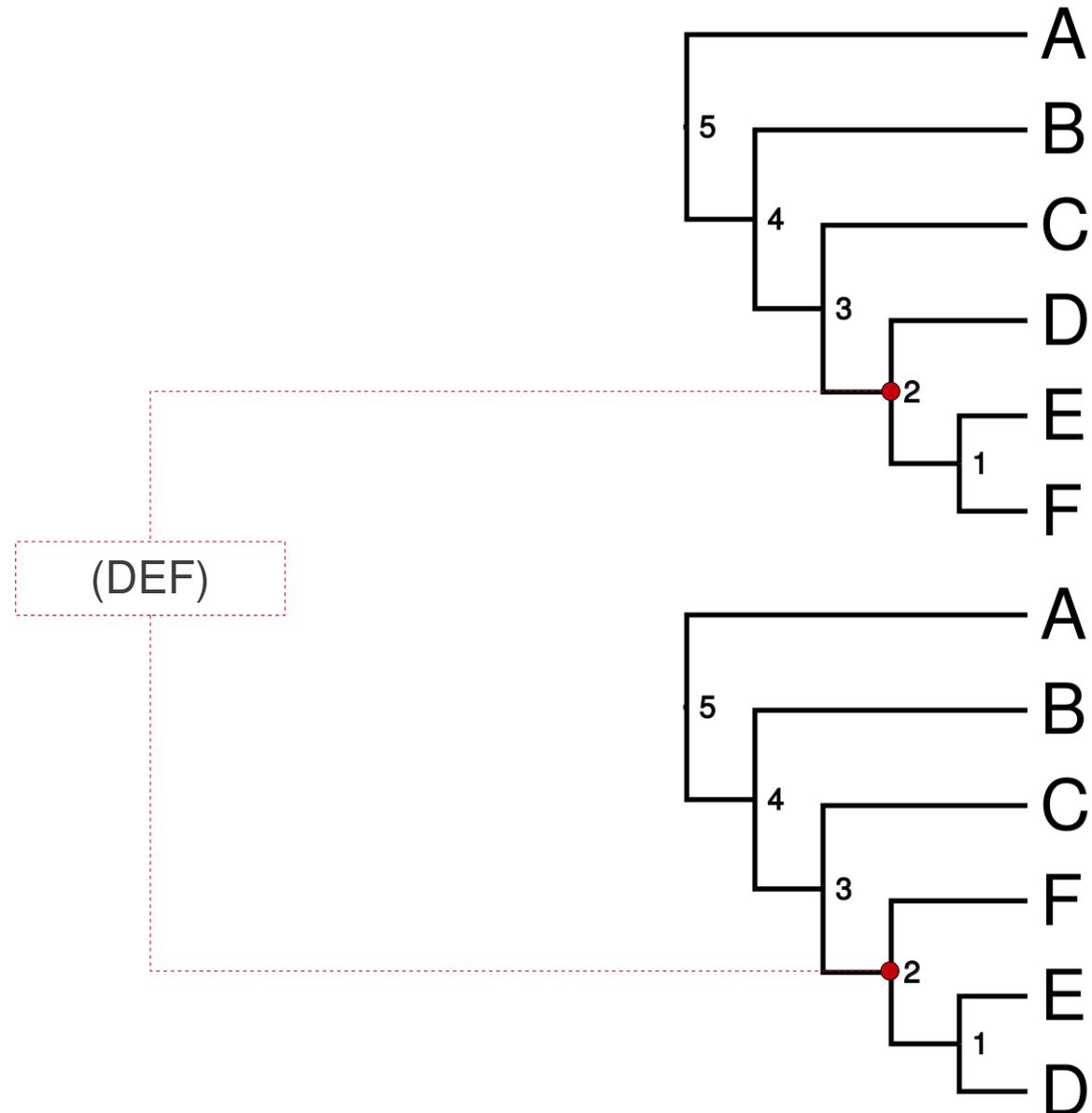


Consenso

Considere os seguintes cladogramas fundamentais:

Componentes
comuns:

- (ABCDEF)
- (BCDEF)
- (CDEF)
- (DEF)

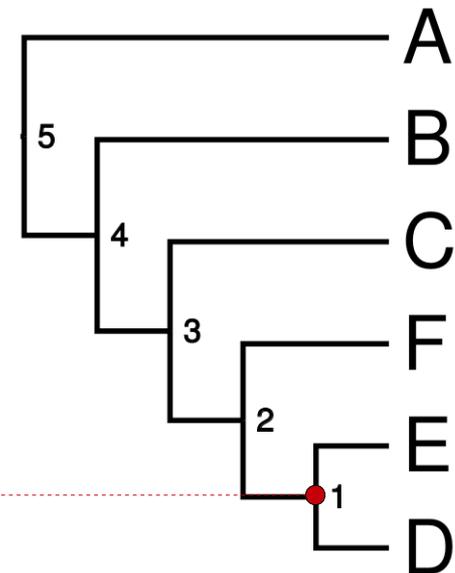
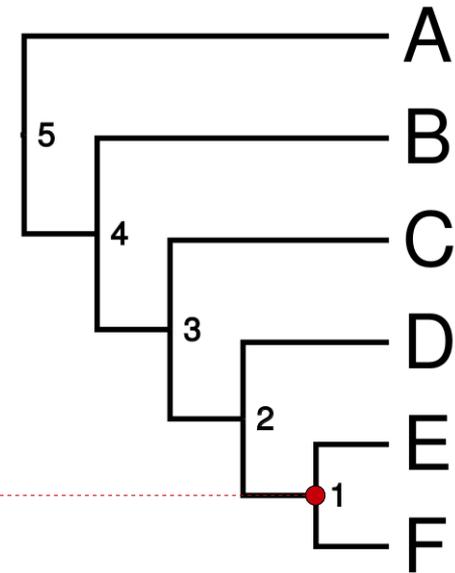
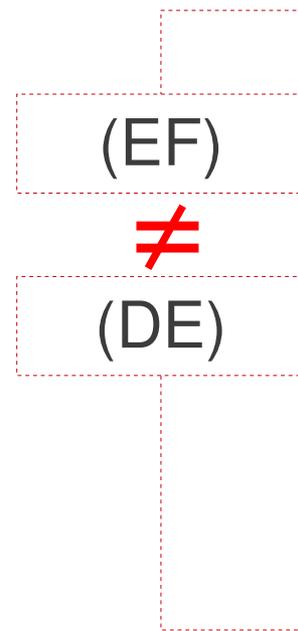


Consenso

Considere os seguintes cladogramas fundamentais:

Componentes
comuns:

- (ABCDEF)
- (BCDEF)
- (CDEF)
- (DEF)



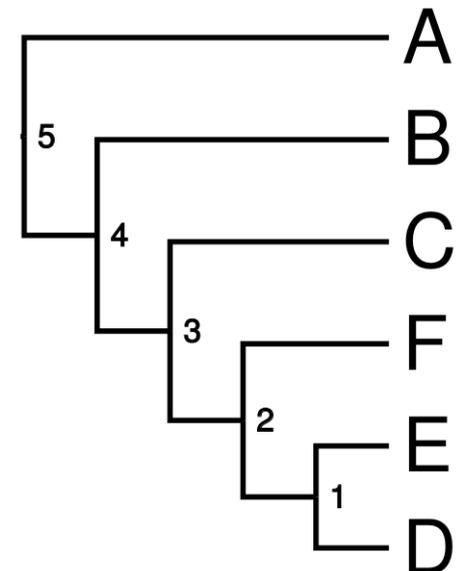
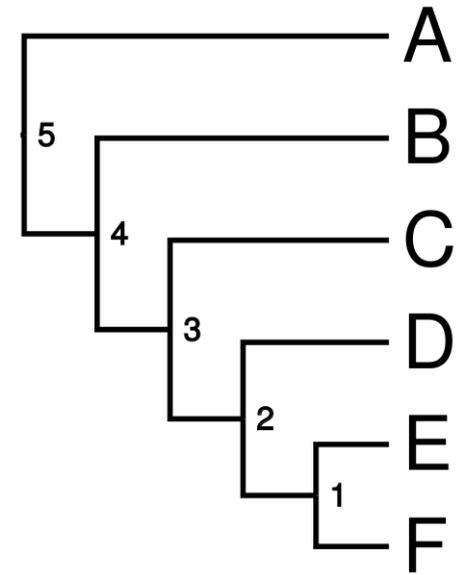
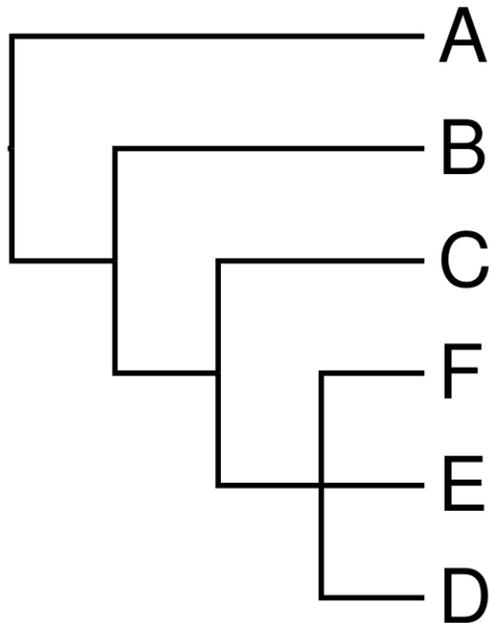
Consenso

Considere os seguintes cladogramas fundamentais:

Componentes
comuns:

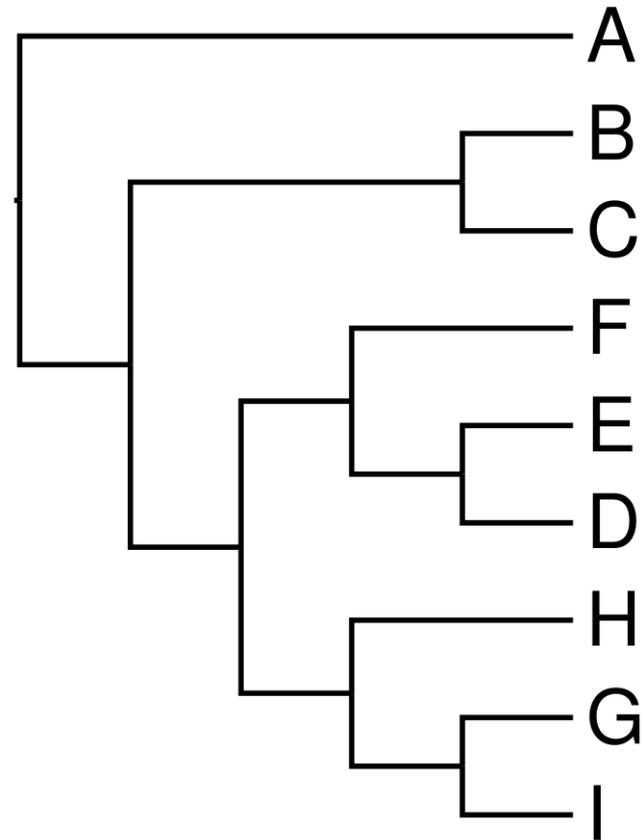
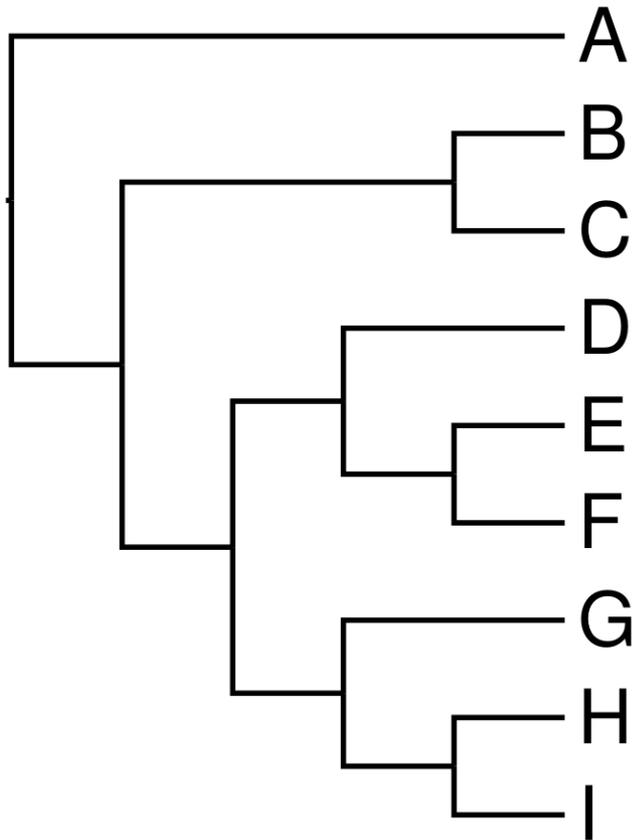
(ABCDEF)
(BCDEF)
(CDEF)
(DEF)

Consenso estrito:



Consenso

Exemplo 1

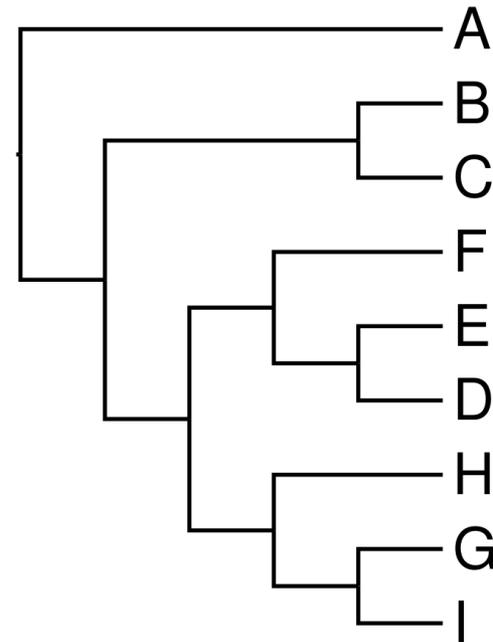
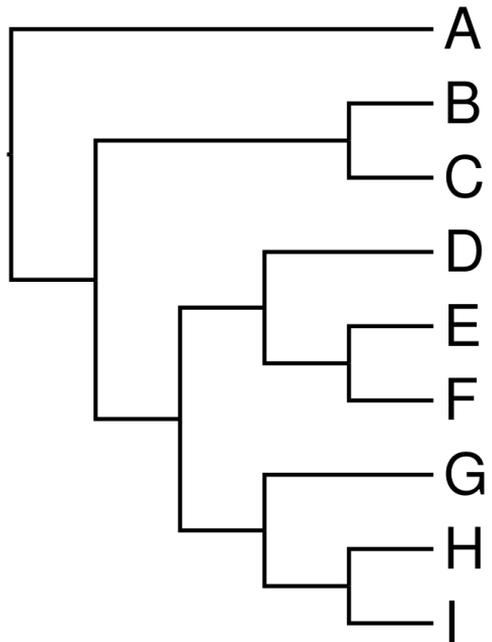


Quais componentes são comuns entre estas duas topologias?

Qual é a topologia de consenso?

Consenso

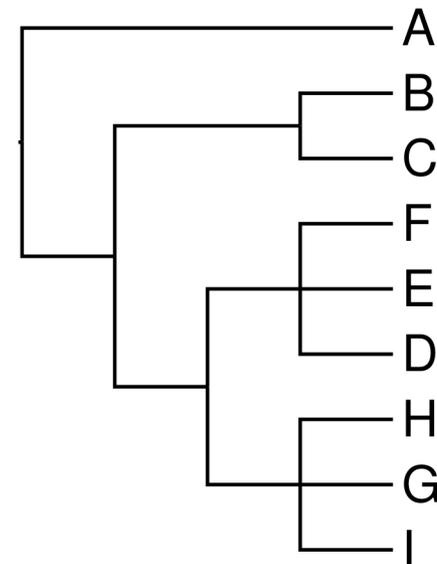
Exemplo



Quais componentes são comuns entre estas duas topologias?

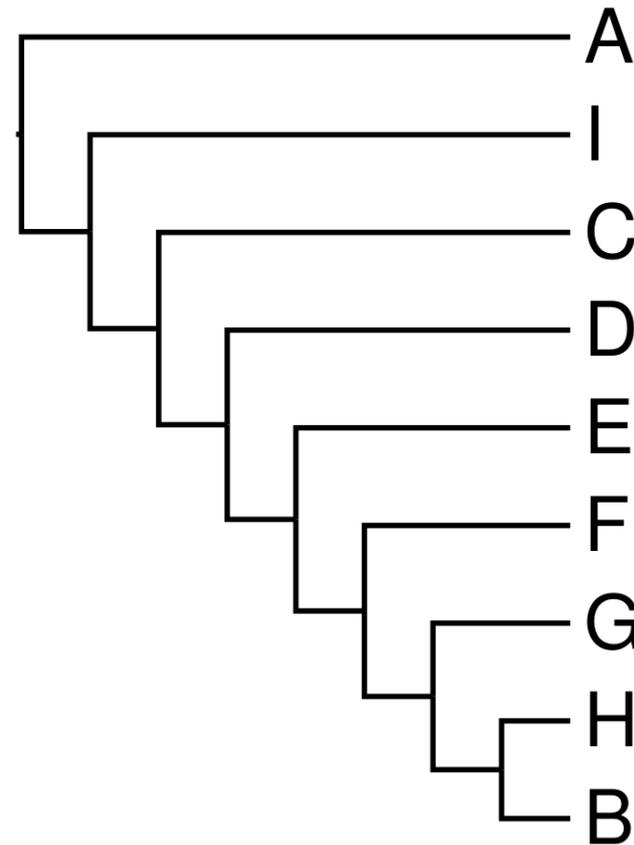
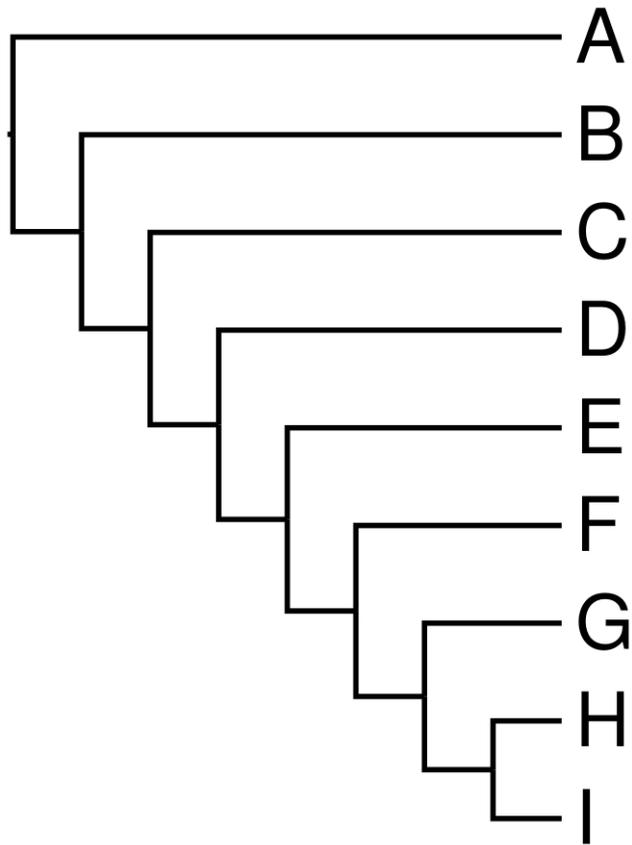
- 1: (ABCDEFGHI)
- 2: (BCDEFGHI)
- 3: (DEFGHI)
- 4: (DEF)
- 5: (GHI)
- 6: (BC)

Qual é a topologia de consenso?



Consenso

Exemplo 2



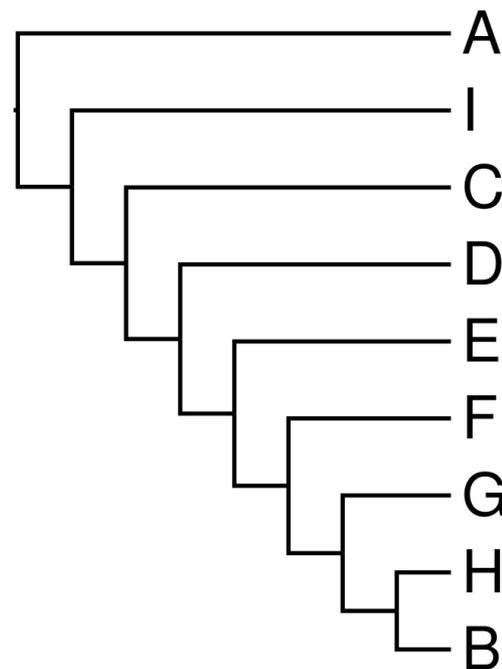
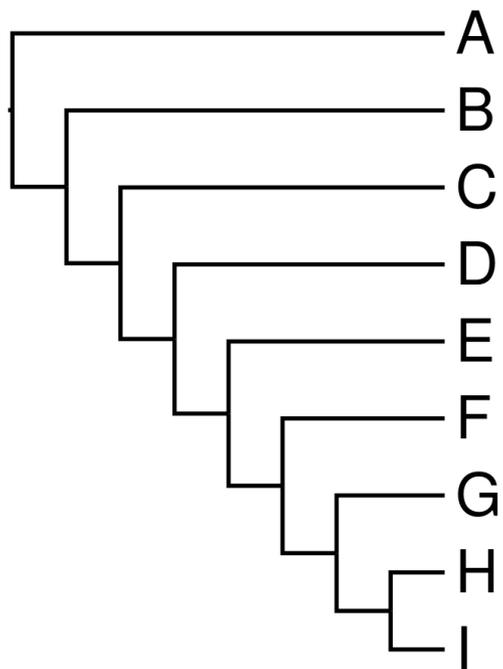
Quais componentes são comuns entre estas duas topologias?

Qual é a topologia de consenso?

Existe algum terminal que está forçando a topologia de consenso?

Consenso

Exemplo 2

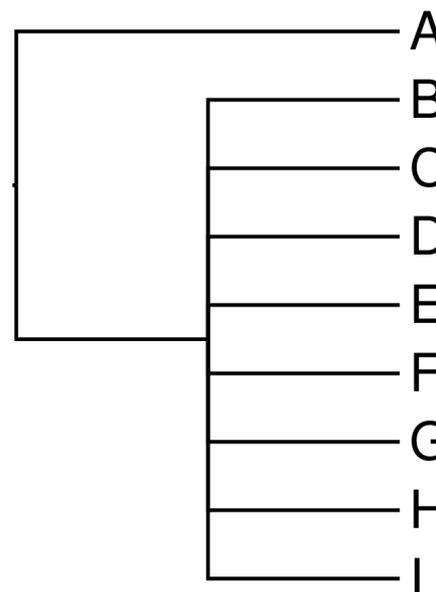


Quais componentes são comuns entre estas duas topologias?

1: (ABCDEFGHI)

2: (BCDEFGHI)

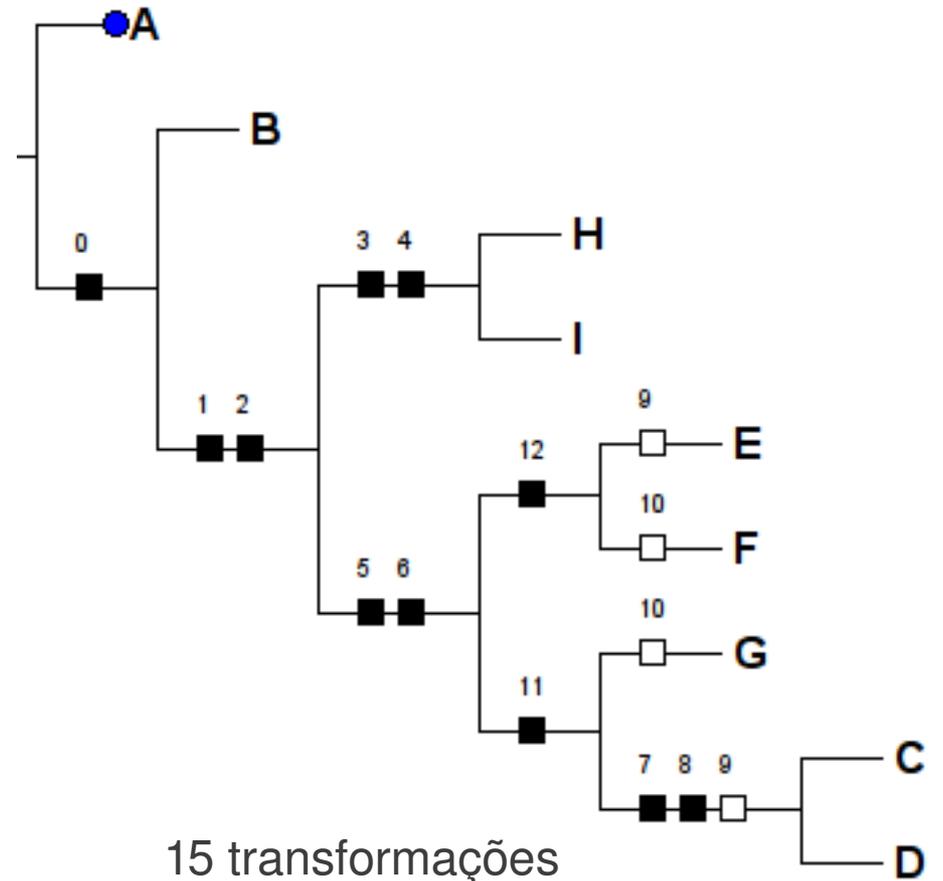
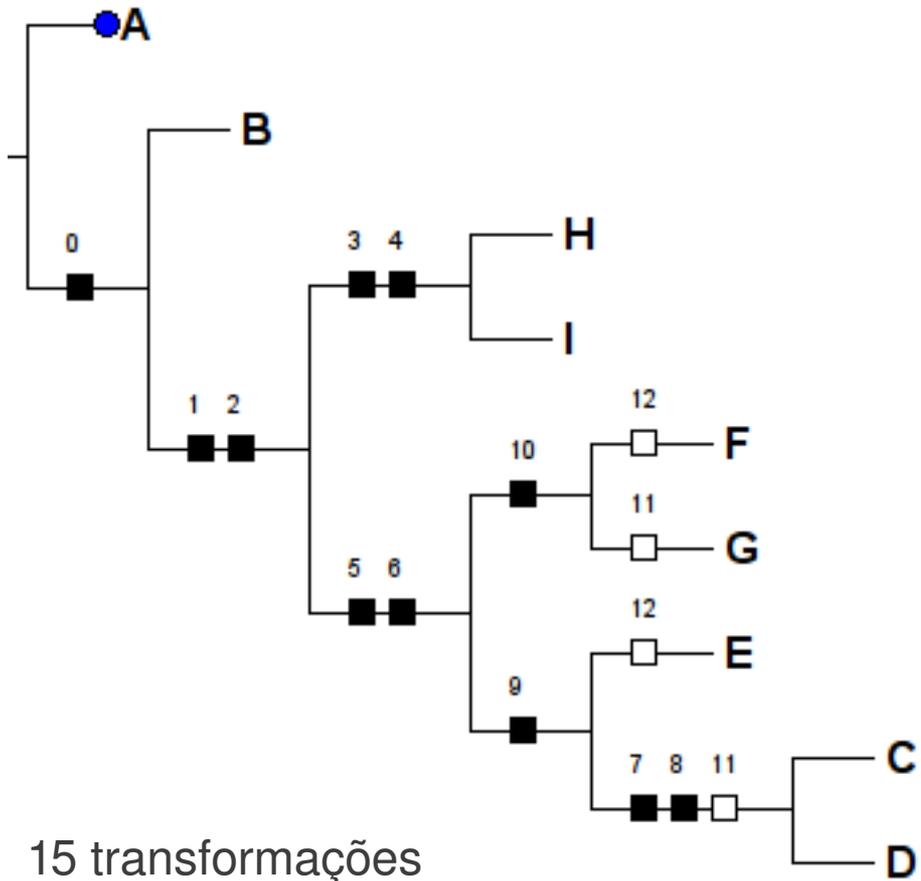
Qual é a topologia de consenso?



Consenso

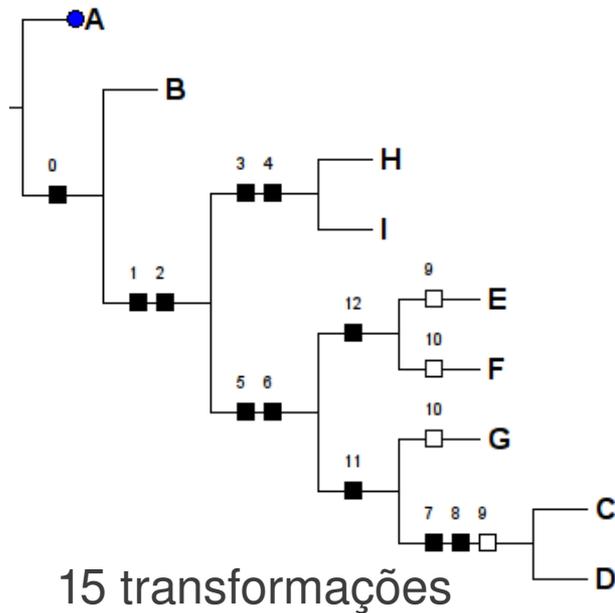
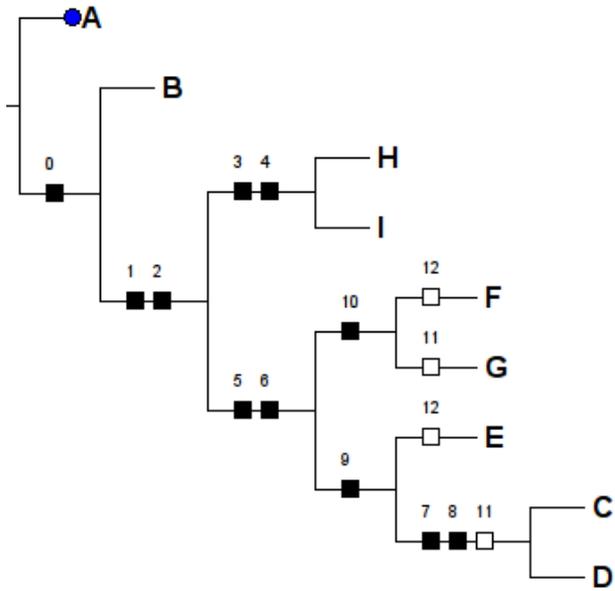
Exemplo 3

```
xread
13 9
A 0000000010000
B 1000000010000
C 1110011101010
D 1110011101010
E 1110011011001
F 1110011010101
G 1110011010110
H 1111100010000
I 1111100010000;
```

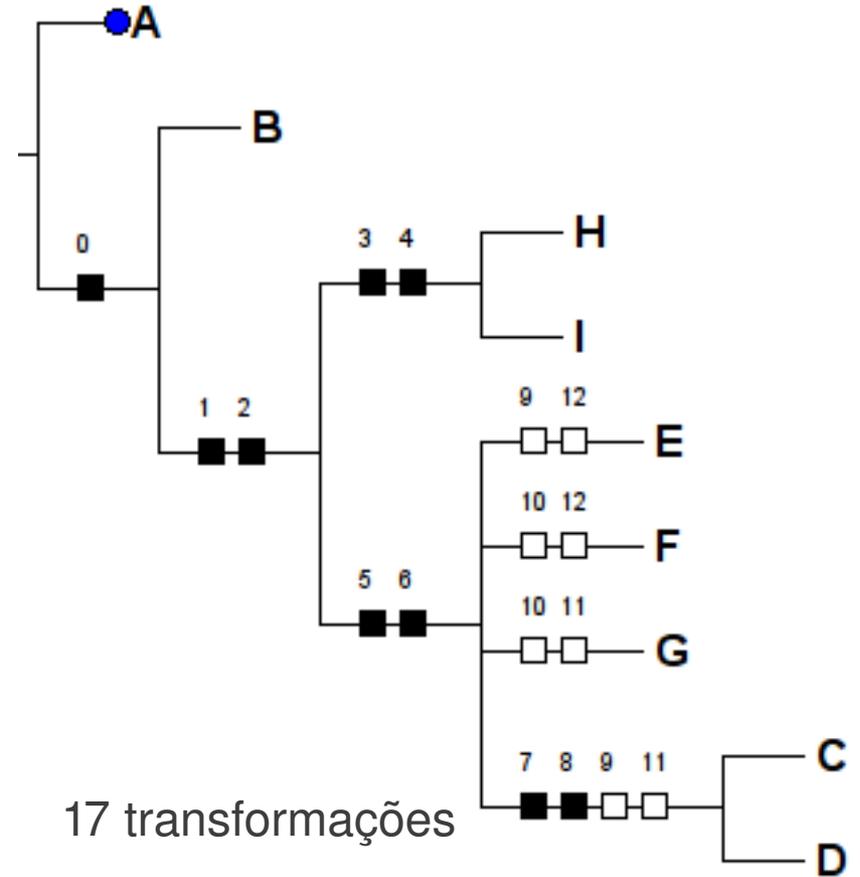


Consenso

Exemplo 3



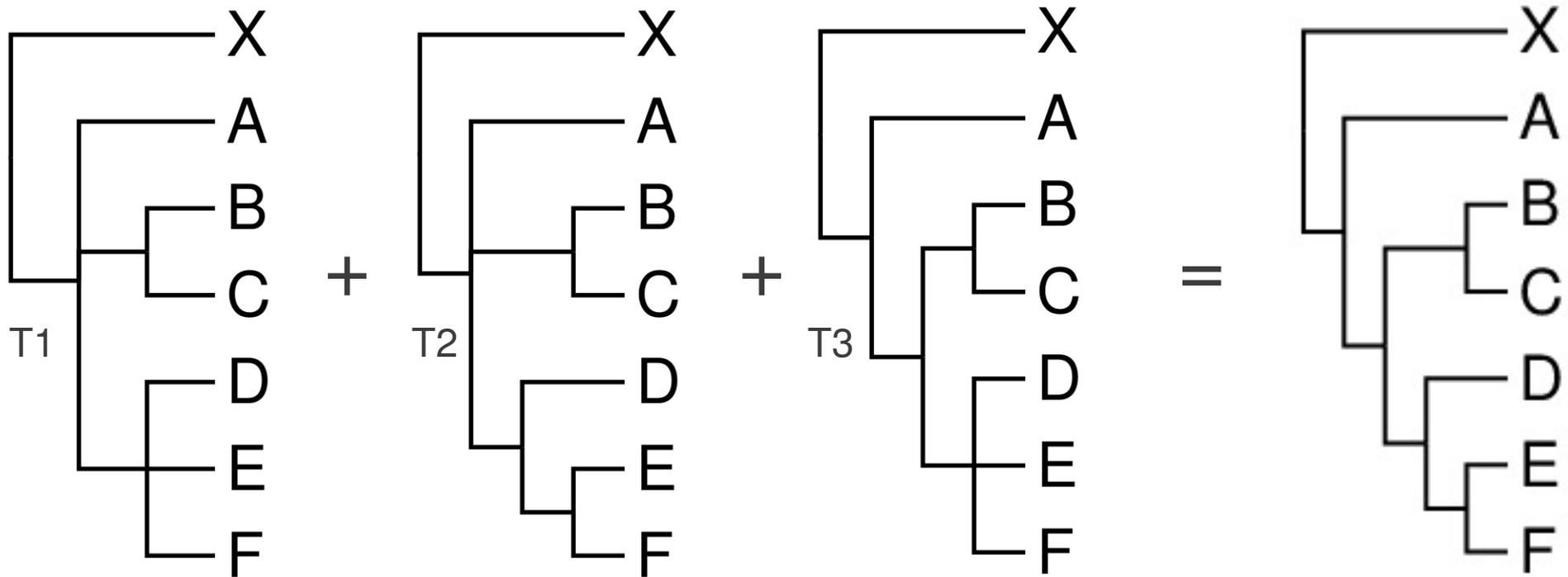
15 transformações



17 transformações

Consenso

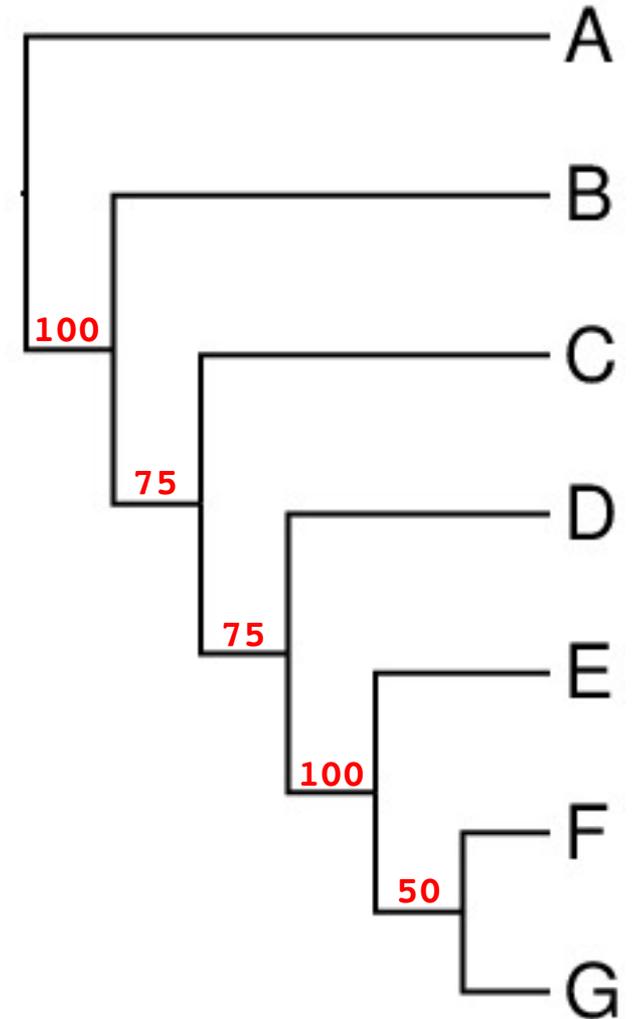
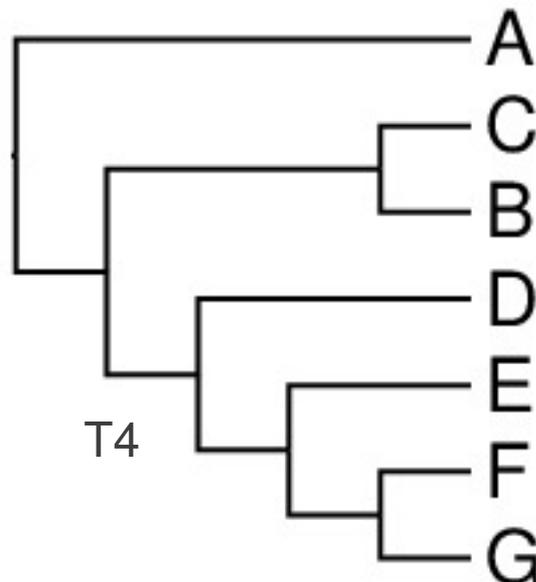
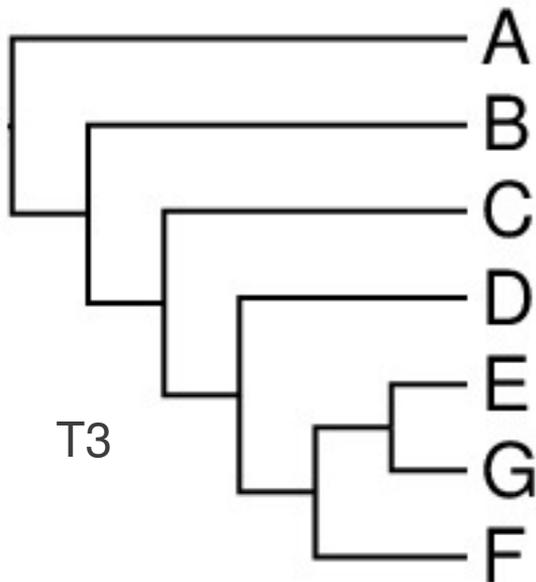
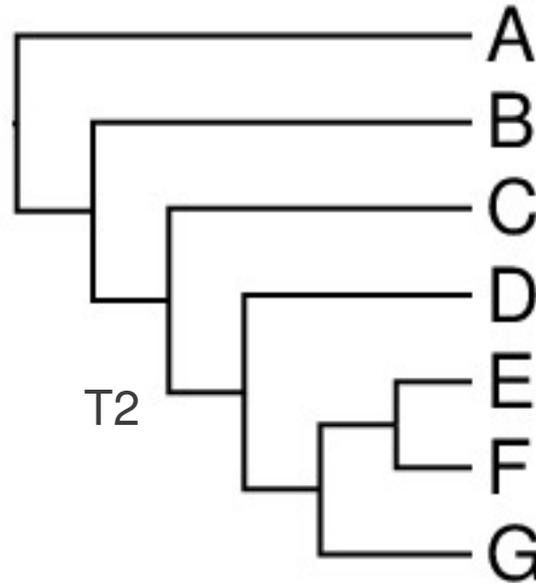
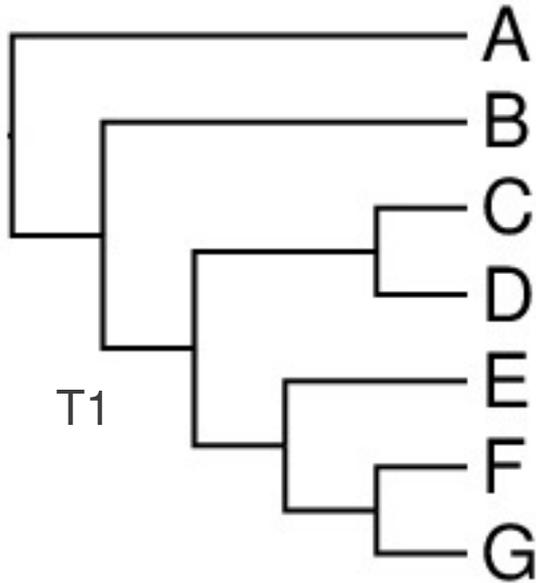
Semi-estrito (“compatible components”)



1. Note que o componente resolvido da T2 envolvendo (DEF) não contradiz o mesmo componente nas topologias T1 e T3.
2. Note que o componente resolvido da T3 envolvendo (BCDEF) não contradiz o mesmo componente nas topologias T1 e T2.
3. Portanto, a topologia de consenso pode ser mais resolvida do que as topologias fundamentais.

Consenso

Maioria (“majority rule”)



1. Neste consenso são considerados os componentes com maior frequência.
2. Note que a topologia de consenso não existe no conjunto de árvores fundamentais.