Stabilité des arbres phylogénétiques

Mahendra Mariadassou

Unité MIG
INRA Jouy-en-Josas

Sémin'R
MNHN

Outline

Pléthore de packages pour la phylogénie

CRAN Task View: Phylogenetics, Especially Comparative Methods

- `http://cran.r-project.org/web/views/Phylogenetics.html`
- De nombreux packages classés par utilisation;
- Beaucoup de méthodes comparatives sur des arbres (corrélation sur des arbres);

Focus du jour

- ape: package général pour la manipulation et la visualisation d'arbres;
- distory: spécialisé dans le calcul de distances entre arbres.

Représentation d'un arbre (objet de classe "phylo")

Un peu de vocabulaire

- branch: branche de l'arbre;
- node: noeud interne de l'arbre;
- tip: noeud terminal (feuille) de l'arbre;
- n: nombre de feuilles;
- m: nombre de noeuds

Structure minimale d'un objet `tree` de classe "phylo"

- `tree$edge` : matrice $(m+n) \times 2$ des branches de l'arbre (noeud de départ $\rightarrow$ noeud d'arrivé);
- `tree$tip.label`: vecteur de taille n des noms des feuilles;
- `tree$Nnode`: entier m, nombre de noeuds (internes) de l'arbre.

Représentation d'un arbre (objet de classe "phylo") 2

Éléments optionnels de `tree`

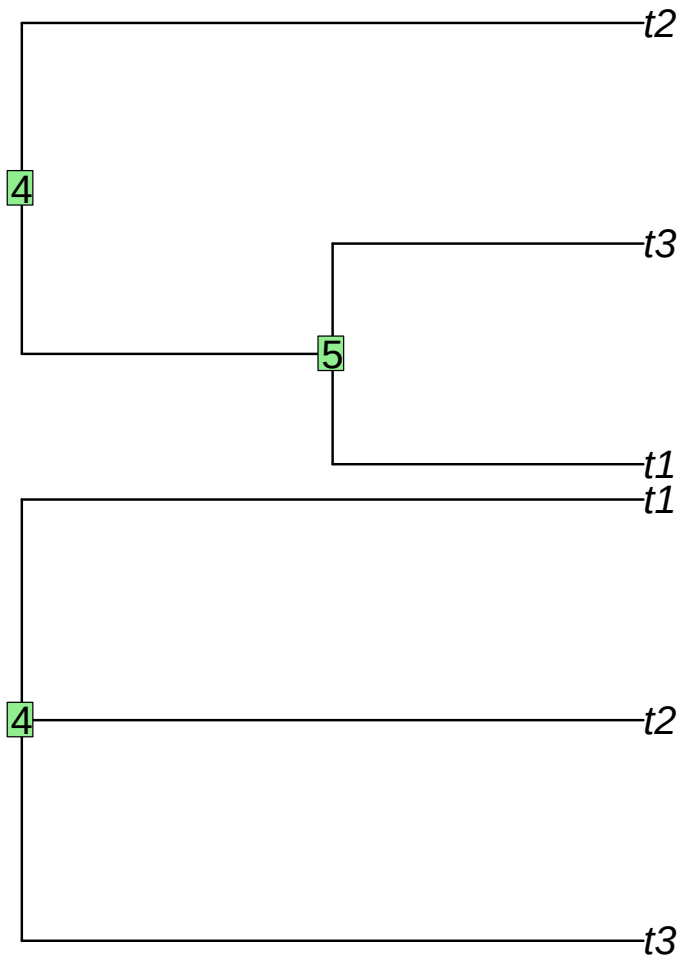
- `tree$edge.length`: vecteur de taille $(n + m - 1)$ des longeurs de branches;
- `tree$node.label`: vecteur de taille m des valeurs des noeuds internes (valeurs bootstrap, proba à posteriori);
- `tree$root.edge`: longeur de la branche à la racine.

Remarques

- Dans `tree$edge`: les feuilles sont codées de 1 à n, les noeuds de $n + 1$ à $n + m$, la racine est numérotée $n + 1$;
- L'arbre est stocké comme un arbre raciné même si la racine n'est pas résolue.
- Les valeurs bootstrap sont associés aux *noeuds*, pas aux *branches*.

Exemples d'arbres

```
>library(ape)
>tree <- rtree(3, br = NULL)
> str(tree)
List of 3
$ edge :  int [1:4, 1:2] 4 5 5 4 5 1 2 3
$ tip.label:  chr [1:3] "t1" "t3" "t2"
$ Nnode :  int 2
- attr(*, "class")= chr "phylo"
>tree <- rtree(3, br = NULL, rooted =
FALSE)
> str(tree)
List of 3
$ edge :  int [1:3, 1:2] 4 4 4 1 2 3
$ tip.label:  chr [1:3] "t3" "t2" "t1"
$ Nnode :  int 1
- attr(*, "class")= chr "phylo"
```



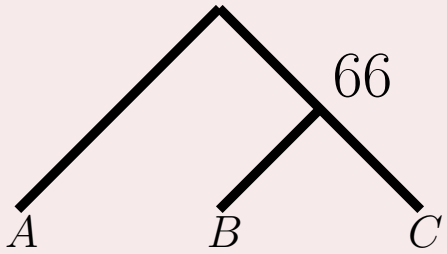
Valeurs bootstrap: la théorie

Alignement original:

Alignement

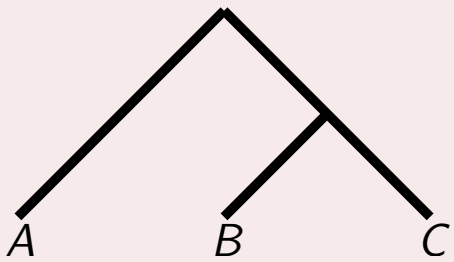
A	A	C	T	T
B	G	G	A	T
C	G	G	C	C

Phylogénie

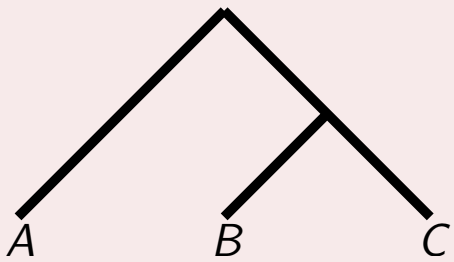


Alignements bootstrap:

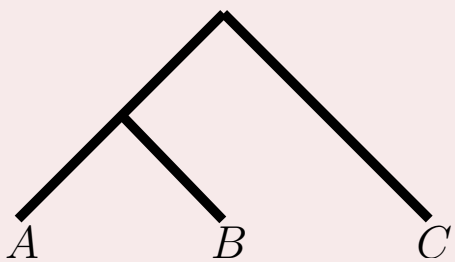
A	A	C	T	C
B	G	G	A	G
C	G	G	C	G



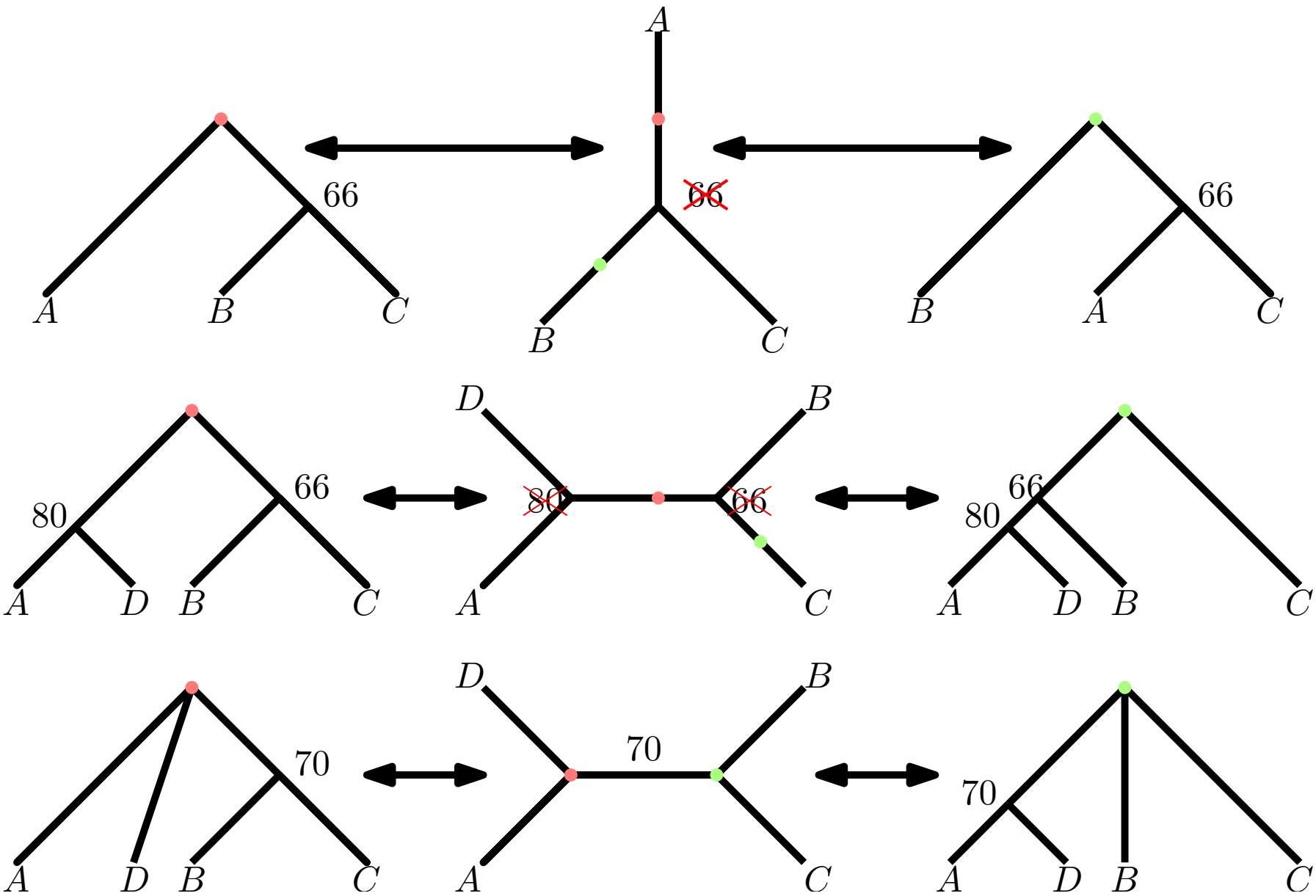
A	C	A	T	A
B	G	G	A	G
C	G	G	C	G



A	T	T	T	T
B	A	T	A	T
C	C	C	C	C



Valeurs bootstrap: une petite difficulté



Valeurs bootstrap: calculs avec R

Décomposition en clades

- `boot.phylo(phy, x, FUN, B = 100, block = 1, trees = FALSE, quiet = FALSE)` : fait une analyse bootstrap complète;
- `prop.part(..., check.labels = TRUE)` : fait un tableau de contingences des clades trouvés dans `...` (généralement une liste d'arbres);
- `prop.clades(phy, ...)` : compte, pour chaque clade de phy le nombre d'occurence du clade dans `...`

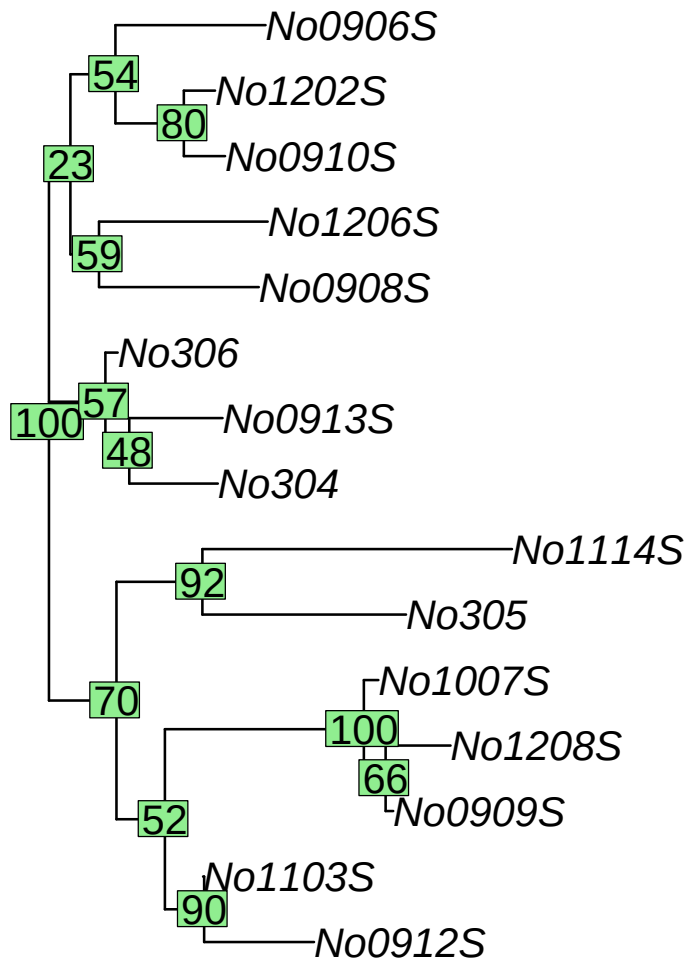
Remarques

- Toutes les fonctions citées précédemment travaillent sur des clades, attention à raciner les arbres de façon cohérente.
- `...` peut être une liste d'arbres produites par votre logiciel favori (phyml, raxml, mrbayes, etc).

boot.phylo

On travaille sur le jeu de données 'woodmouse', alignement du Cytochrome b de 15 woodmouse (965 nucléotides).

```
>data(woodmouse)
>phylo.infer <- function(x) nj(dist.dna(x))
>wood.tree <- phylo.infer(woodmouse)
>boot.values <- boot.phylo(wood.tree,
woodmouse, phylo.infer, quiet = TRUE)
>plot(wood.tree)
>nodelabels(boot.values, bg = "lightgreen")
```

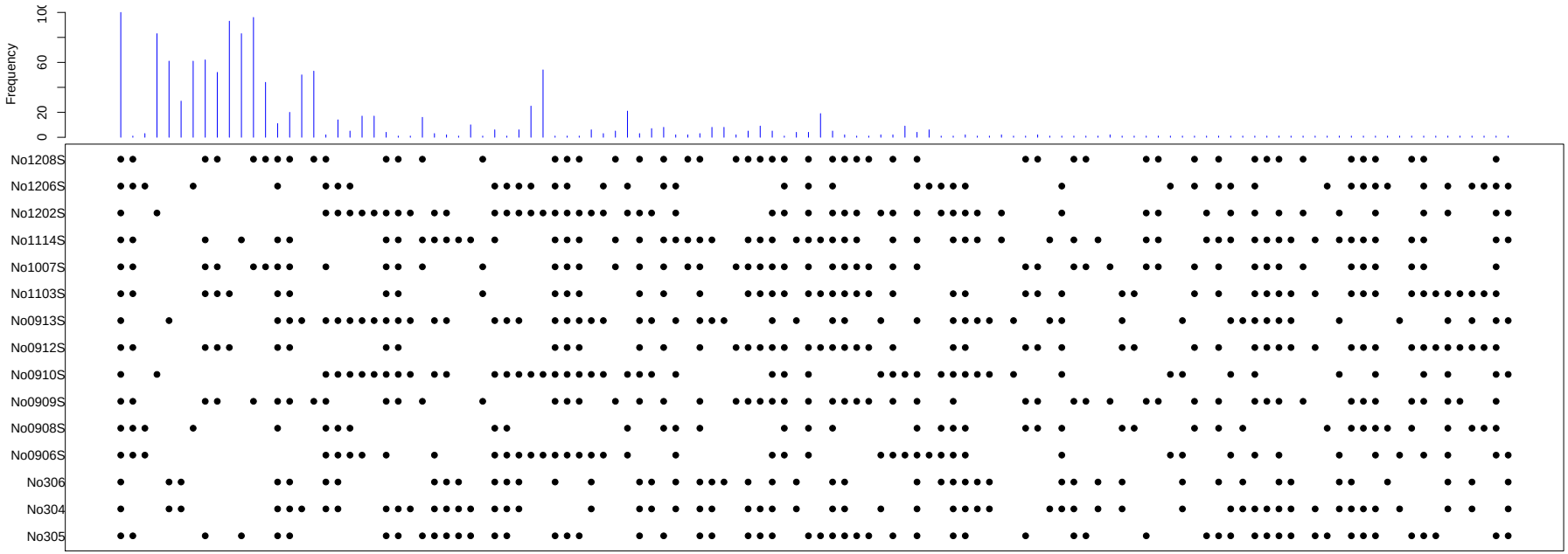


prop.part

```
> boot.analysis <- boot.phylo(wood.tree, woodmouse, phylo.infer, trees
= TRUE)
> attributes(boot.analysis)
$names
[1] "BP" "trees"
> boot.values <- boot.analysis$BP
> boot.trees <- boot.analysis$trees
> boot.part <- prop.part(boot.trees); plot(boot.part)
```

prop.part

```
> boot.analysis <- boot.phylo(wood.tree, woodmouse, phylo.infer, trees
= TRUE)
> attributes(boot.analysis)
$names
[1] "BP" "trees"
> boot.values <- boot.analysis$BP
> boot.trees <- boot.analysis$trees
> boot.part <- prop.part(boot.trees); plot(boot.part)
```

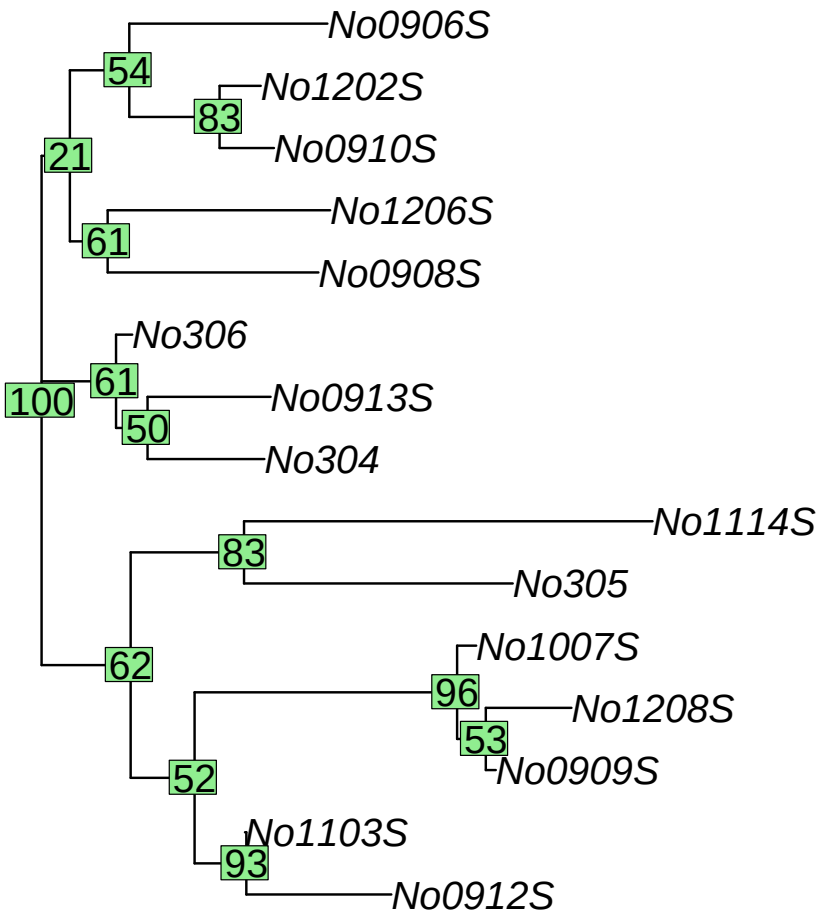


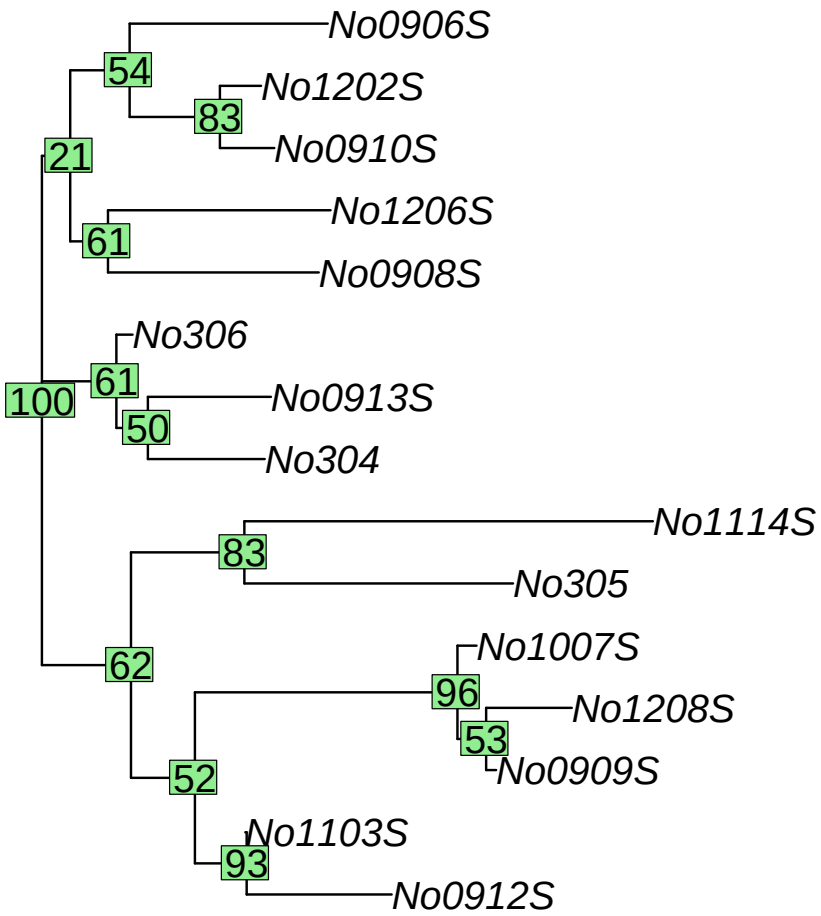
prop.clades

```
> clades.values <- prop.clades(wood.tree, boot.trees)
> plot(wood.tree); nodelabels(boot.values)
> plot(wood.tree); nodelabels(clades.values)
```

prop.clades

```
> clades.values <- prop.clades(wood.tree, boot.trees)
> plot(wood.tree); nodelabels(boot.values)
> plot(wood.tree); nodelabels(clades.values)
```



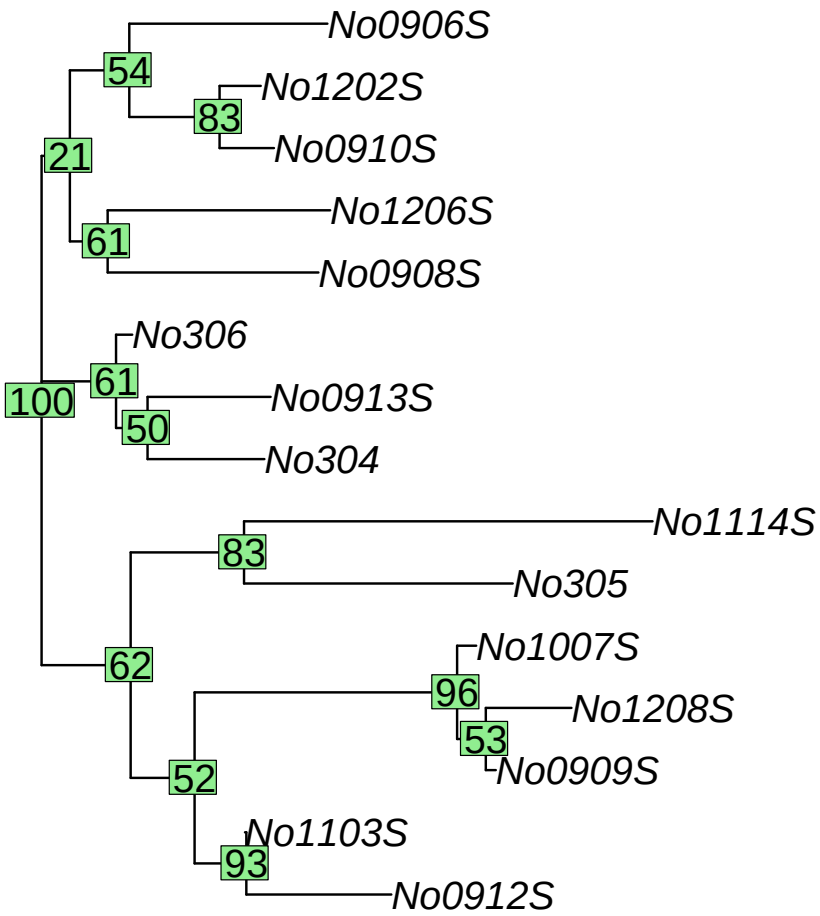


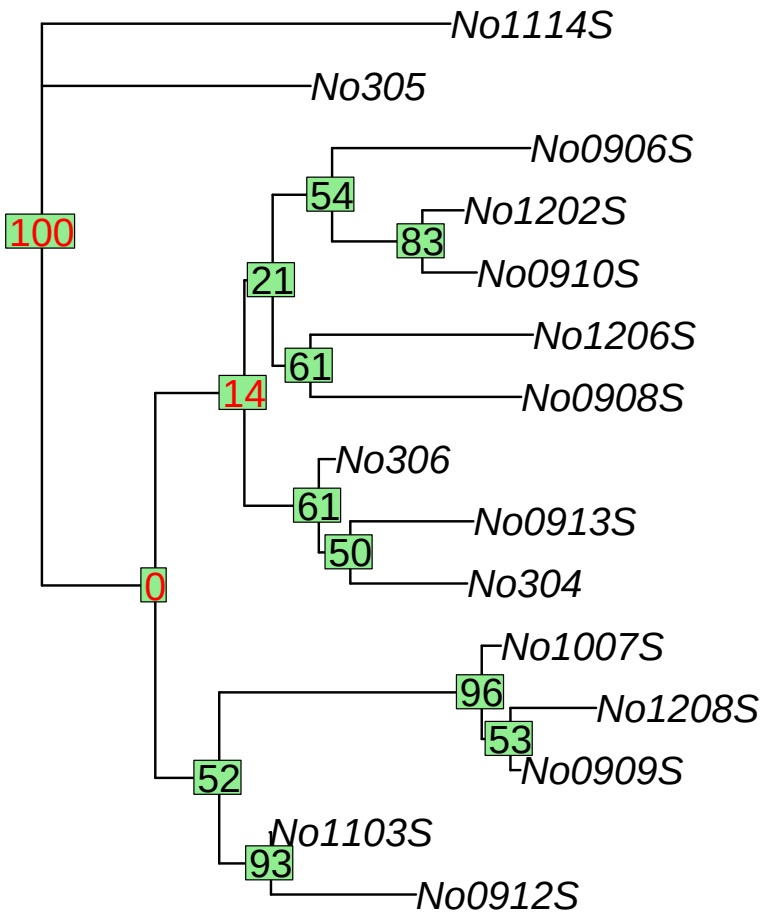
Problèmes de racine

```
> wood.tree.rerooted <- root(wood.tree, "No305")
> boot.values.rerooted <- prop.clades(wood.tree.rerooted, boot.trees)
> plot(wood.tree); nodelabels(boot.values)
> plot(wood.tree.rerooted); nodelabels(boot.values.rerooted)
```

Problèmes de racine

```
> wood.tree.rerooted <- root(wood.tree, "No305")
> boot.values.rerooted <- prop.clades(wood.tree.rerooted, boot.trees)
> plot(wood.tree); nodelabels(boot.values)
> plot(wood.tree.rerooted); nodelabels(boot.values.rerooted)
```





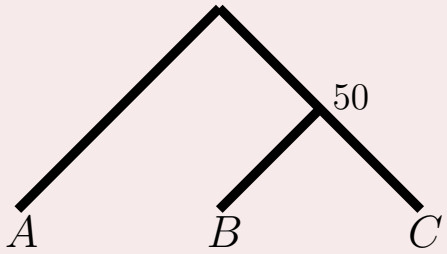
Valeurs jackknife: la théorie

Alignement original:

Alignement

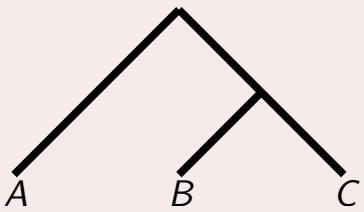
A	*A*	*C*	*T*	*T*
B	*G*	*G*	*A*	*T*
C	*G*	*G*	*C*	*C*

Phylogénie

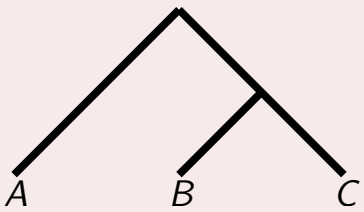


Alignements jackknife:

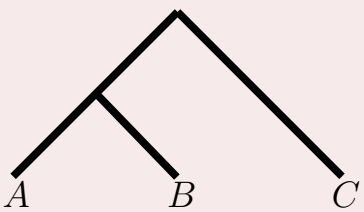
A	*C*	*T*	*T*
B	*G*	*A*	*T*
C	*G*	*C*	*C*



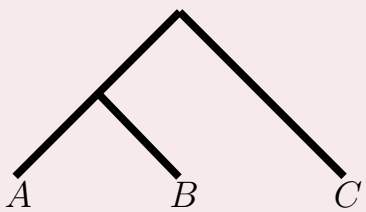
A	*A*	*T*	*T*
B	*G*	*A*	*T*
C	*G*	*C*	*C*



A	*A*	*C*	*T*
B	*G*	*G*	*T*
C	*G*	*G*	*C*



A	*A*	*C*	*T*
B	*G*	*G*	*A*
C	*G*	*G*	*C*

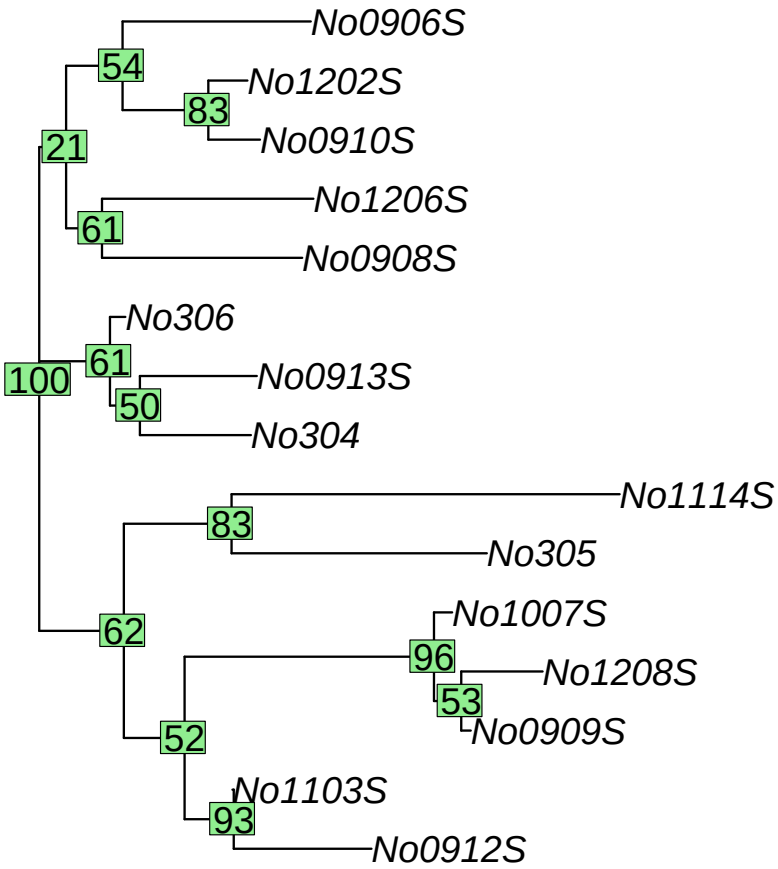


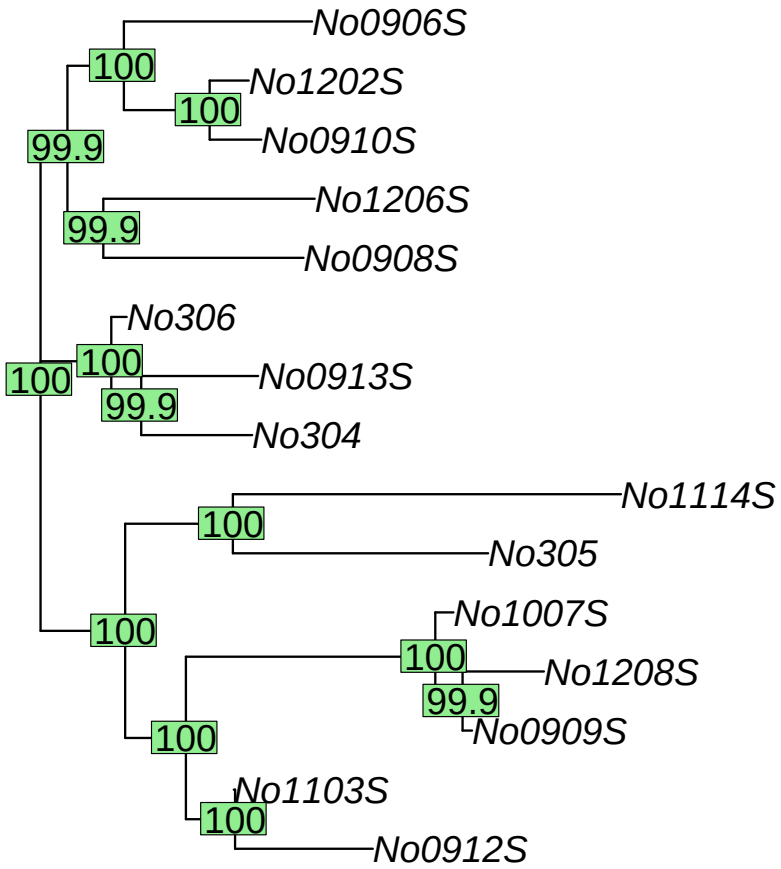
Valeurs jackknife

```
> wj.tree <- function(pos)  nj(dist.dna(woodmouse[ , -pos]))
> jack.trees <- vector("list", length = 965)
> outgroup <- c("No1114S", ..., "No0912S")
> for (pos in 1:965)  jack.trees[[pos]] <- root(wj.tree(pos), outgroup)
> jack.values <- prop.clades(root(wood.tree, outgroup), jack.trees)
```

Valeurs jackknife

```
> wj.tree <- function(pos)  nj(dist.dna(woodmouse[ , -pos]))
> jack.trees <- vector("list", length = 965)
> outgroup <- c("No1114S", ..., "No0912S")
> for (pos in 1:965)  jack.trees[[pos]] <- root(wj.tree(pos), outgroup)
> jack.values <- prop.clades(root(wood.tree, outgroup), jack.trees)
```





Arbres consensus

Motivation

- Résumer l'information de tous les arbres bootstrap/jackknife sans arbre de référence;
- Identifier l'information partagée par tous les arbres.
- Produire un arbre consensus qui soit compatible avec tous (une majorité) d'arbres.

Remarques

- L'arbre consensus est moins résolu que les arbres bootstrap.
- En général, le consensus n'a pas de longueurs de branches.

Les différents consensus

En fonction de la régle de consensus, l'arbre consensus contient:

- Strict Consensus: les clades présents dans tous les arbres;
- Majority Rule: les clades présents dans la moitié au moins des

Arbres consensus

Remarques

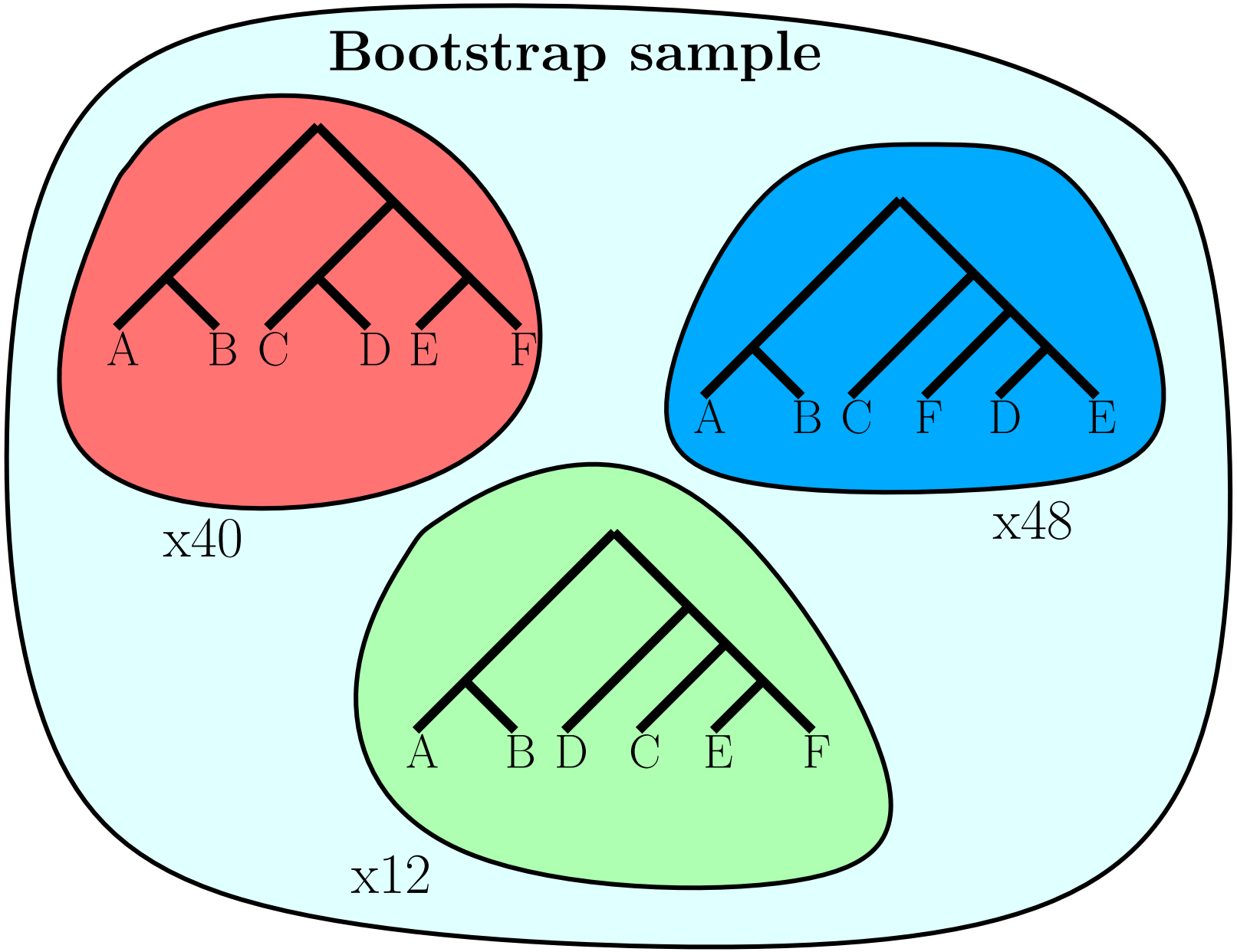
- L'arbre consensus est moins résolu que les arbres bootstrap.
- En général, le consensus n'a pas de longueurs de branches.

Les différents consensus

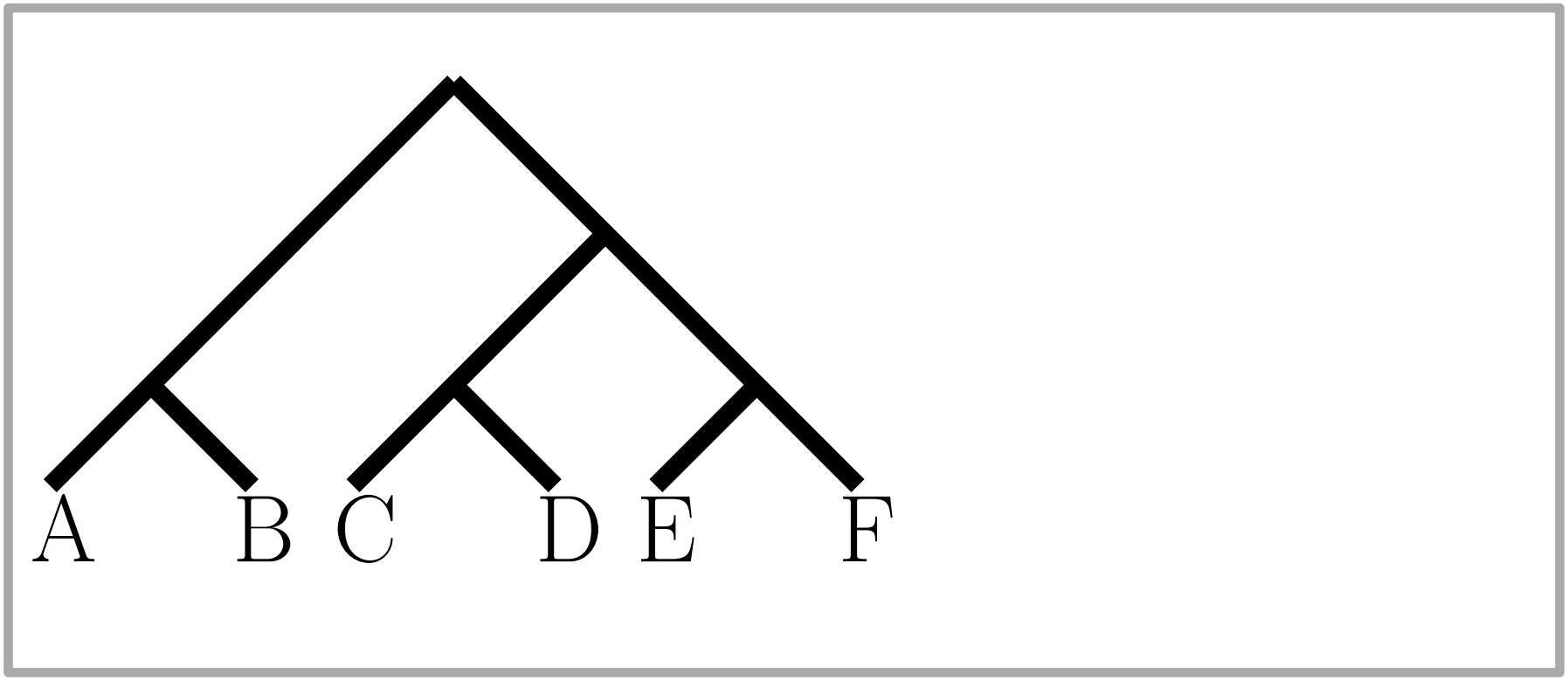
En fonction de la régle de consensus, l'arbre consensus contient:

- Strict Consensus: les clades présents dans tous les arbres;
- Majority Rule: les clades présents dans la moitié au moins des arbres ;
- Extended Majority Rule: clades présents dans la moitié au moins des arbres et plus jusqu'à résoudre l'arbre.

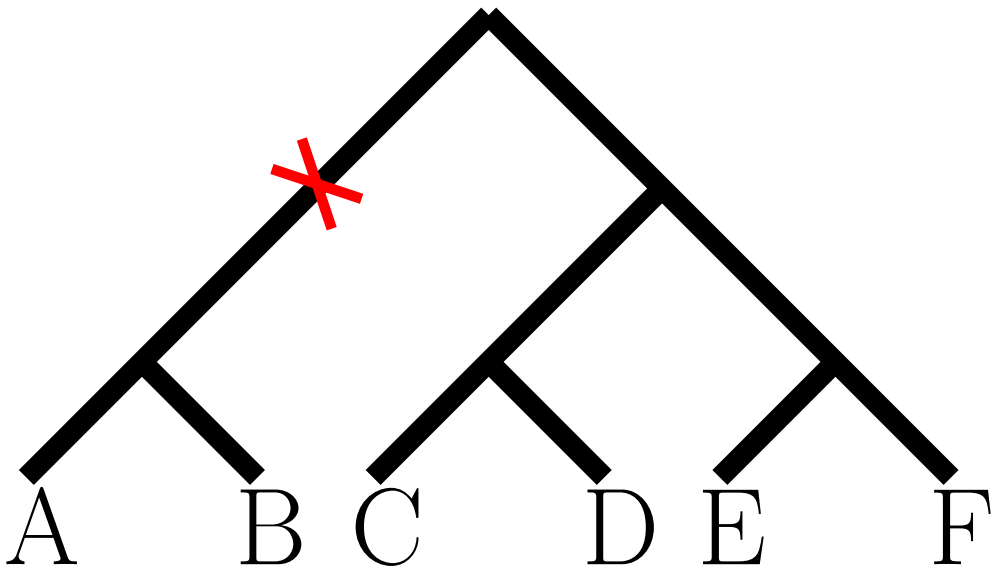
Arbres consensus: un example



Arbres consensus: un example

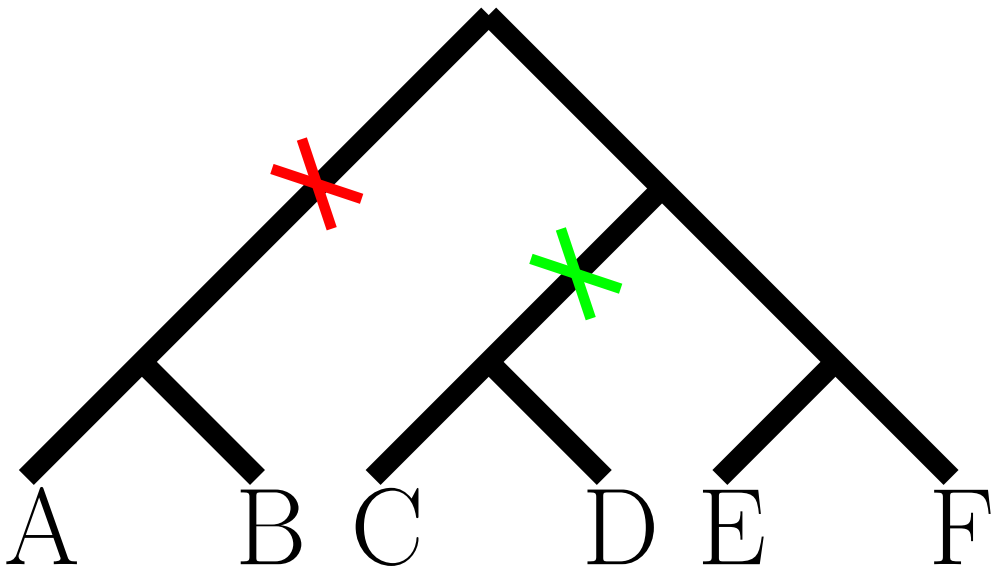


Arbres consensus: un example



AB—CDEF

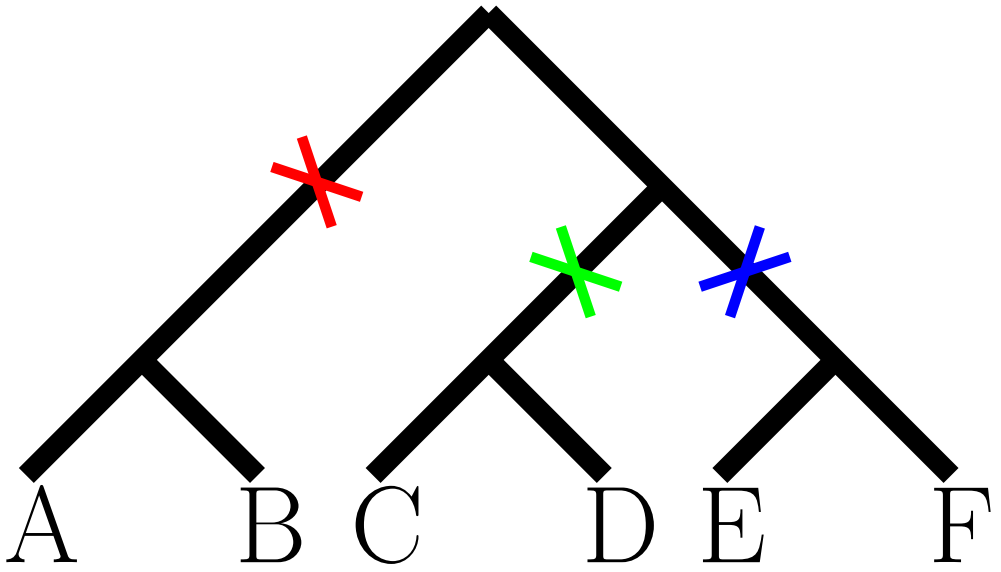
Arbres consensus: un example



AB—CDEF

ABEF—CD

Arbres consensus: un example

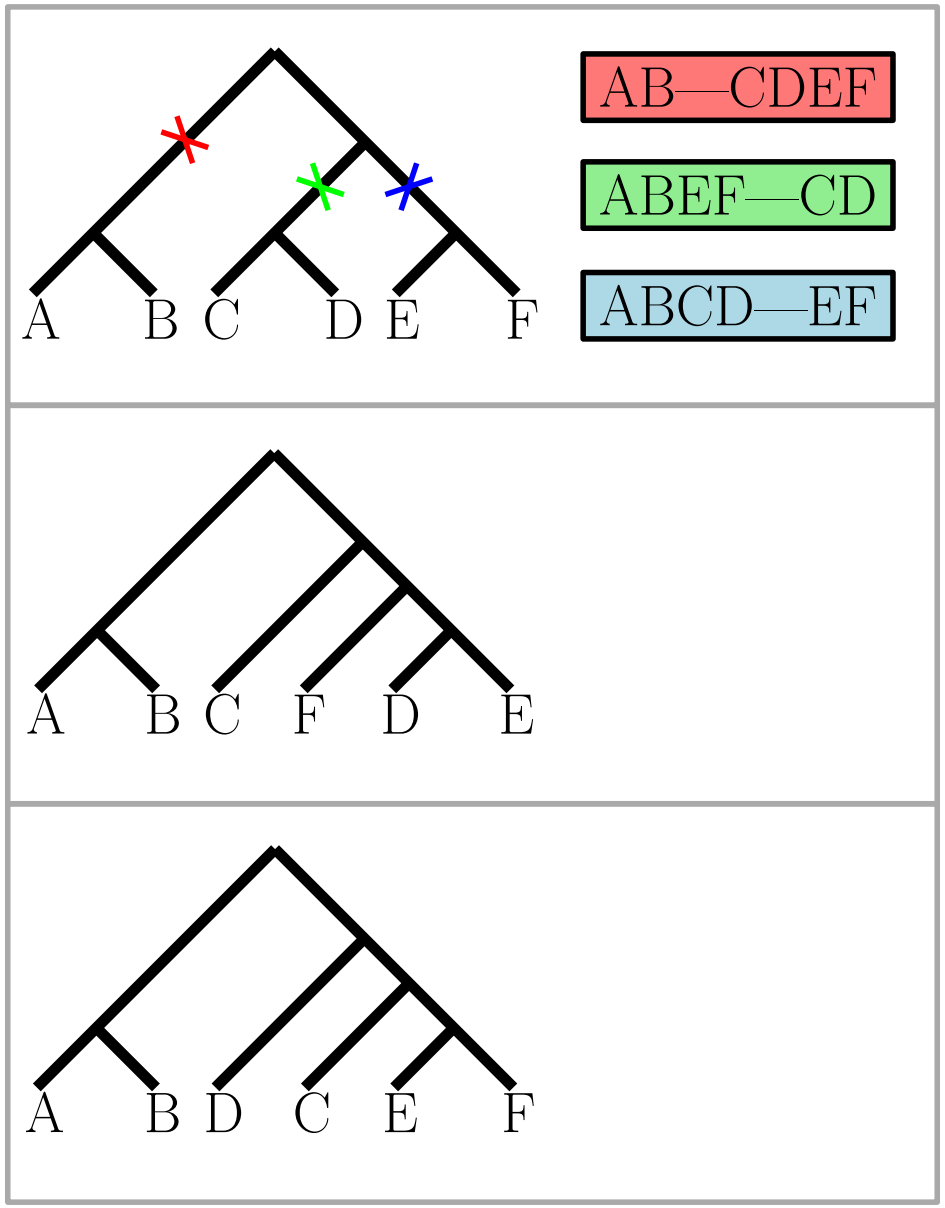


AB—CDEF

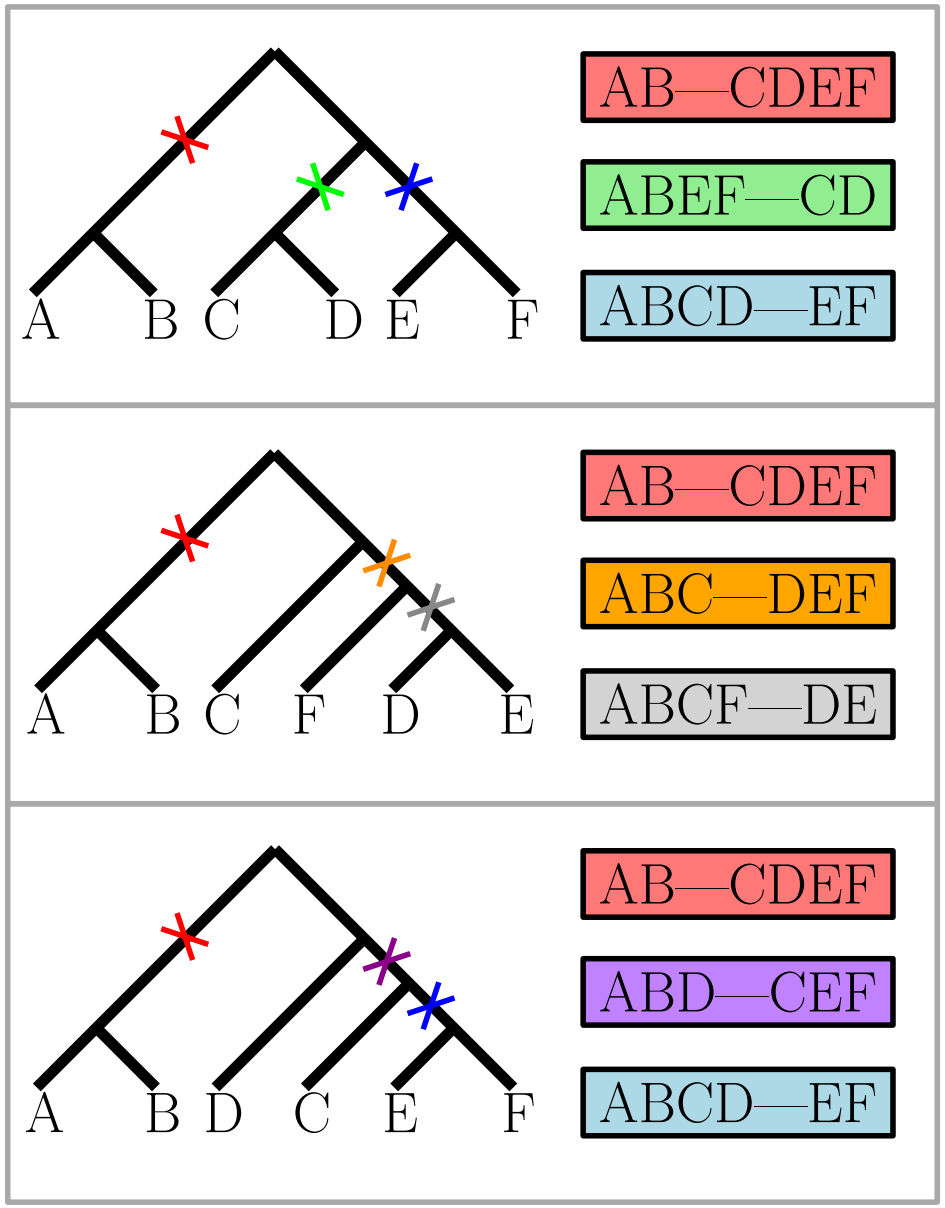
ABEF—CD

ABCD—EF

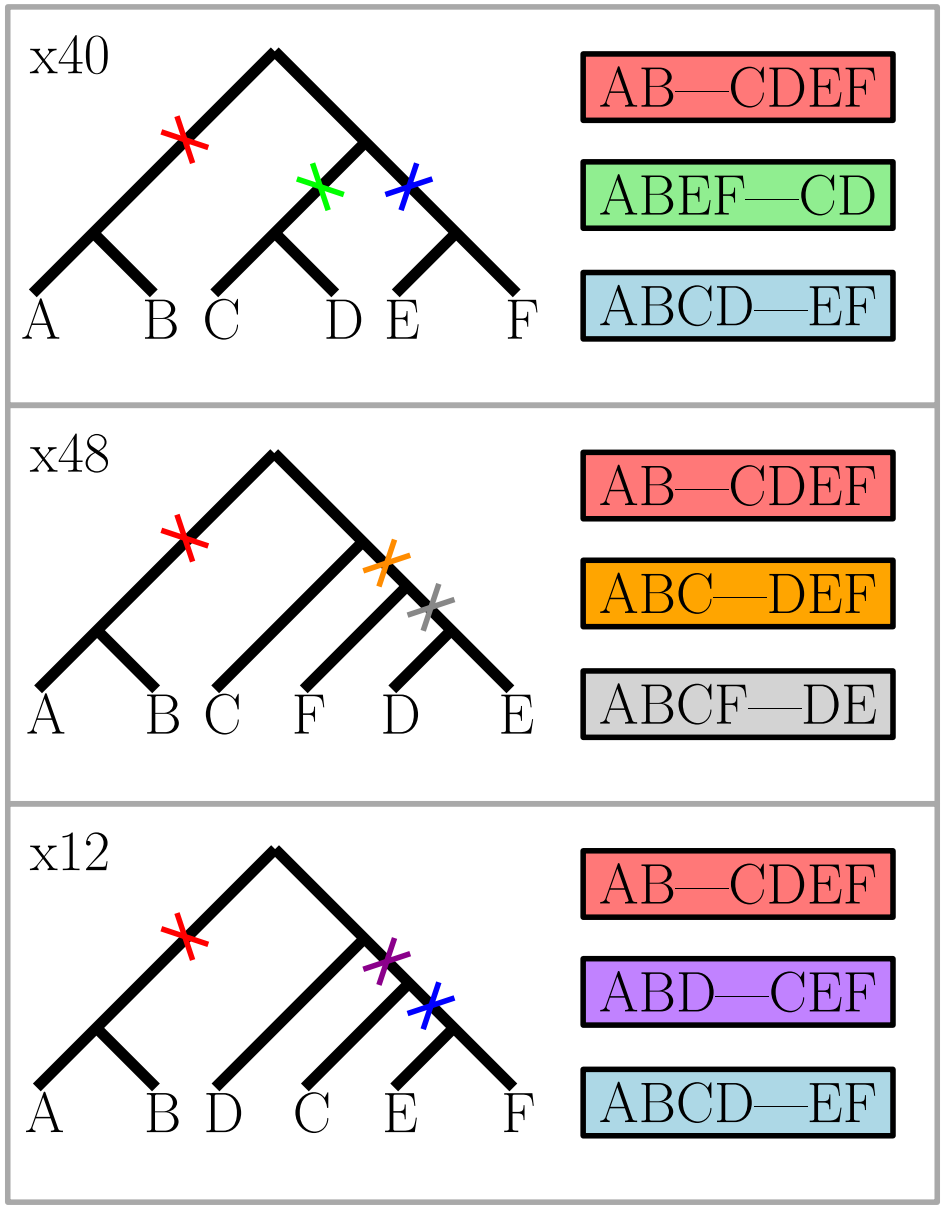
Arbres consensus: un example



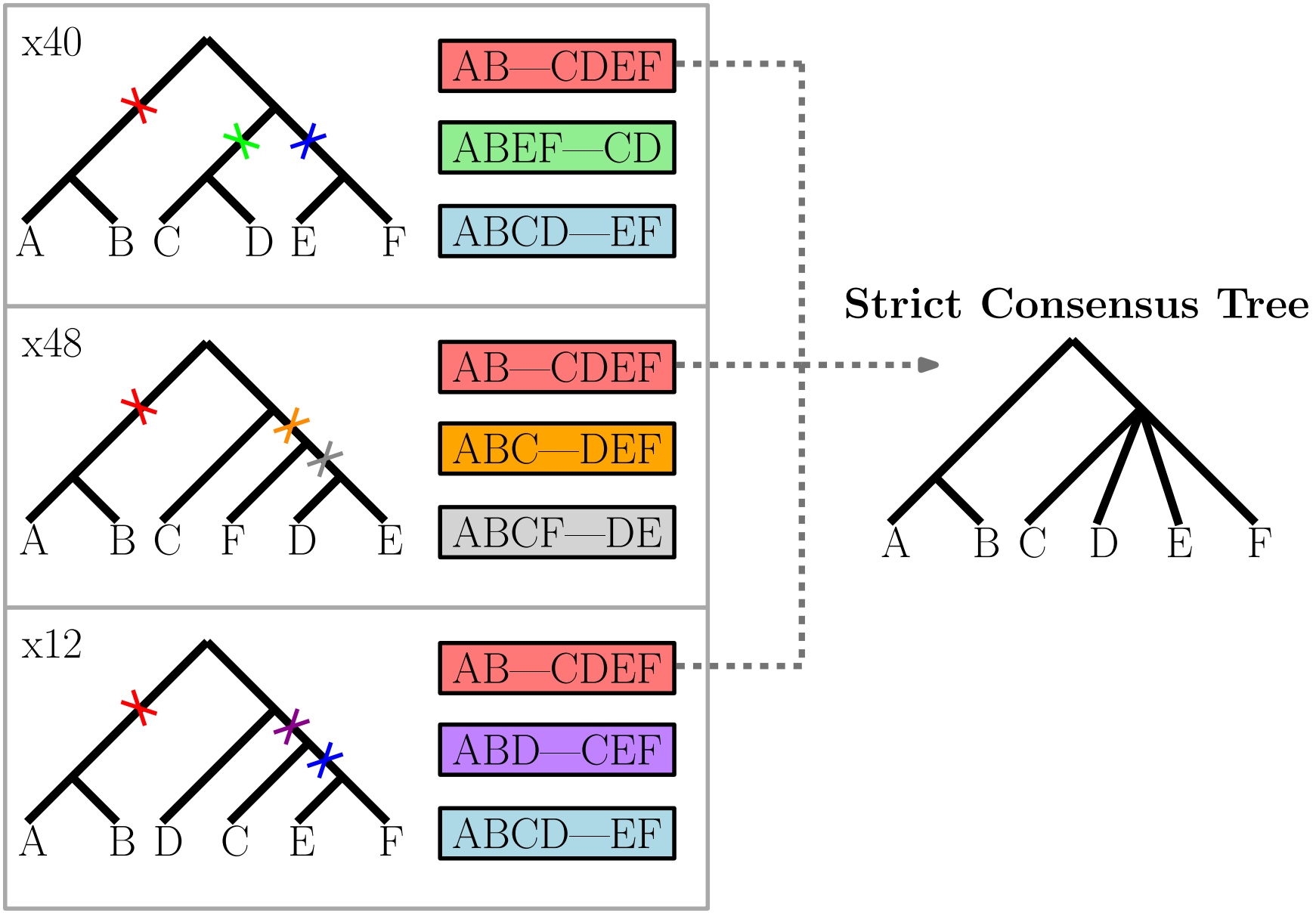
Arbres consensus: un example



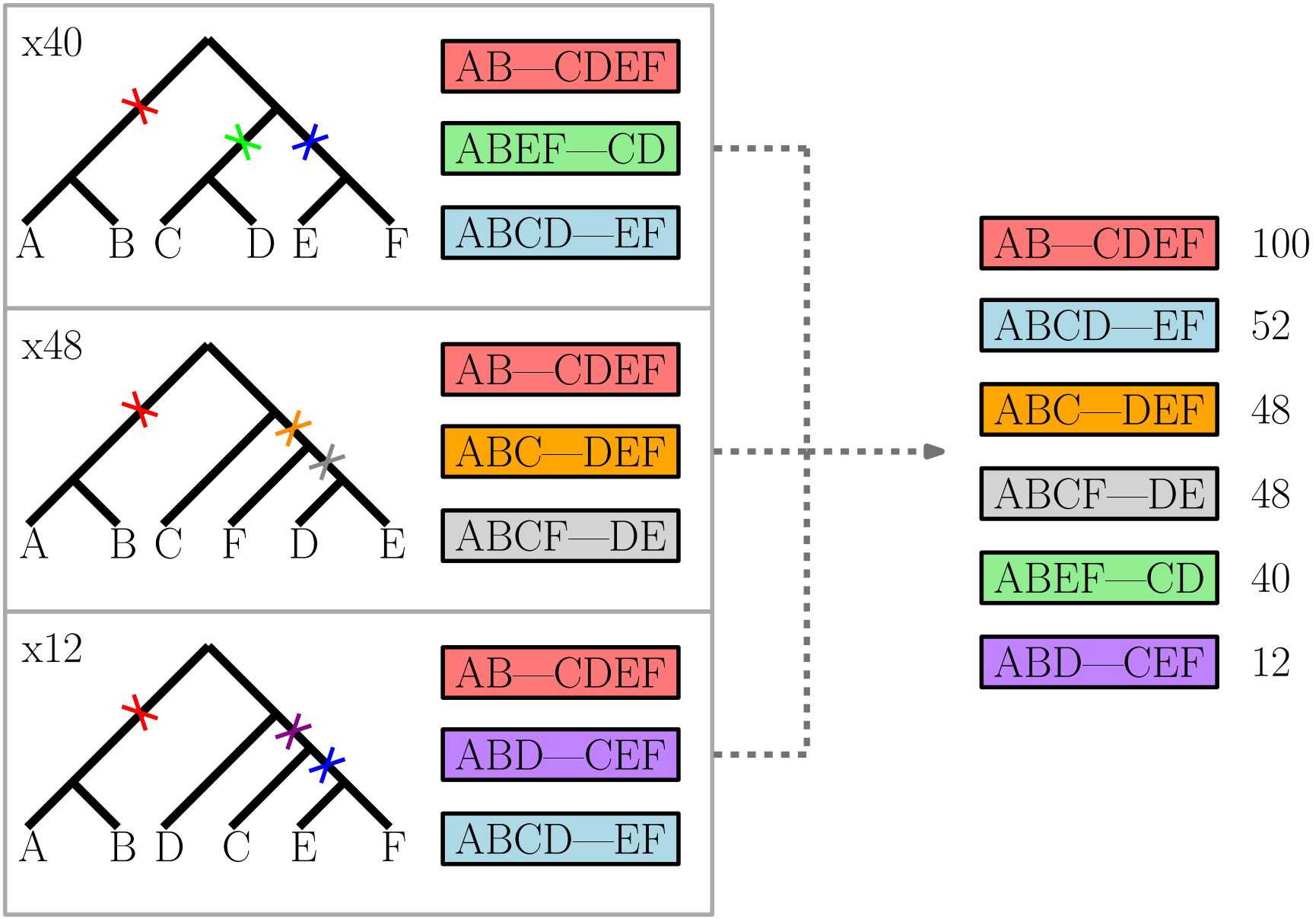
Arbres consensus: un example



Arbres consensus: un example

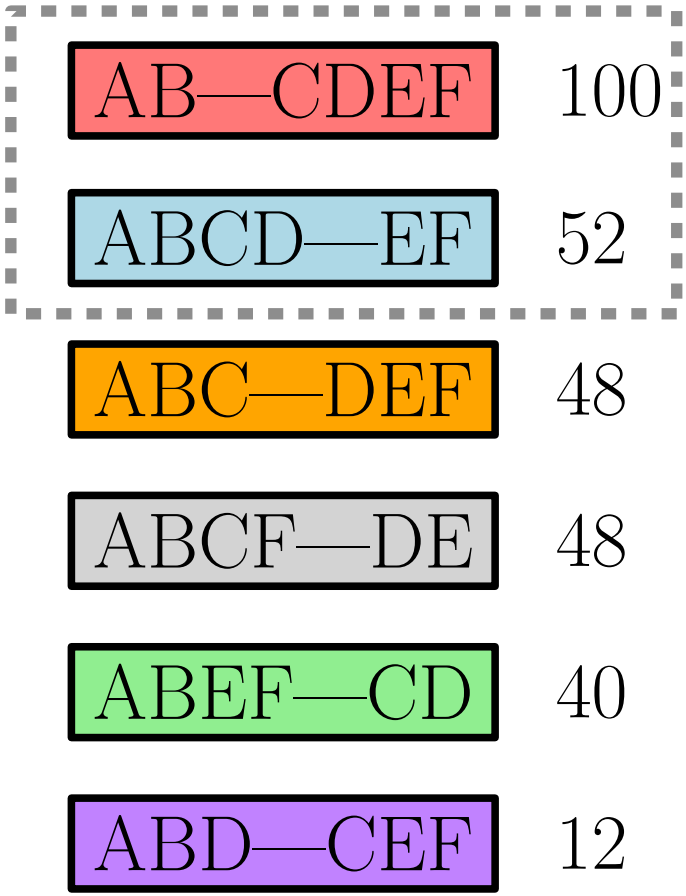


Arbres consensus: un example

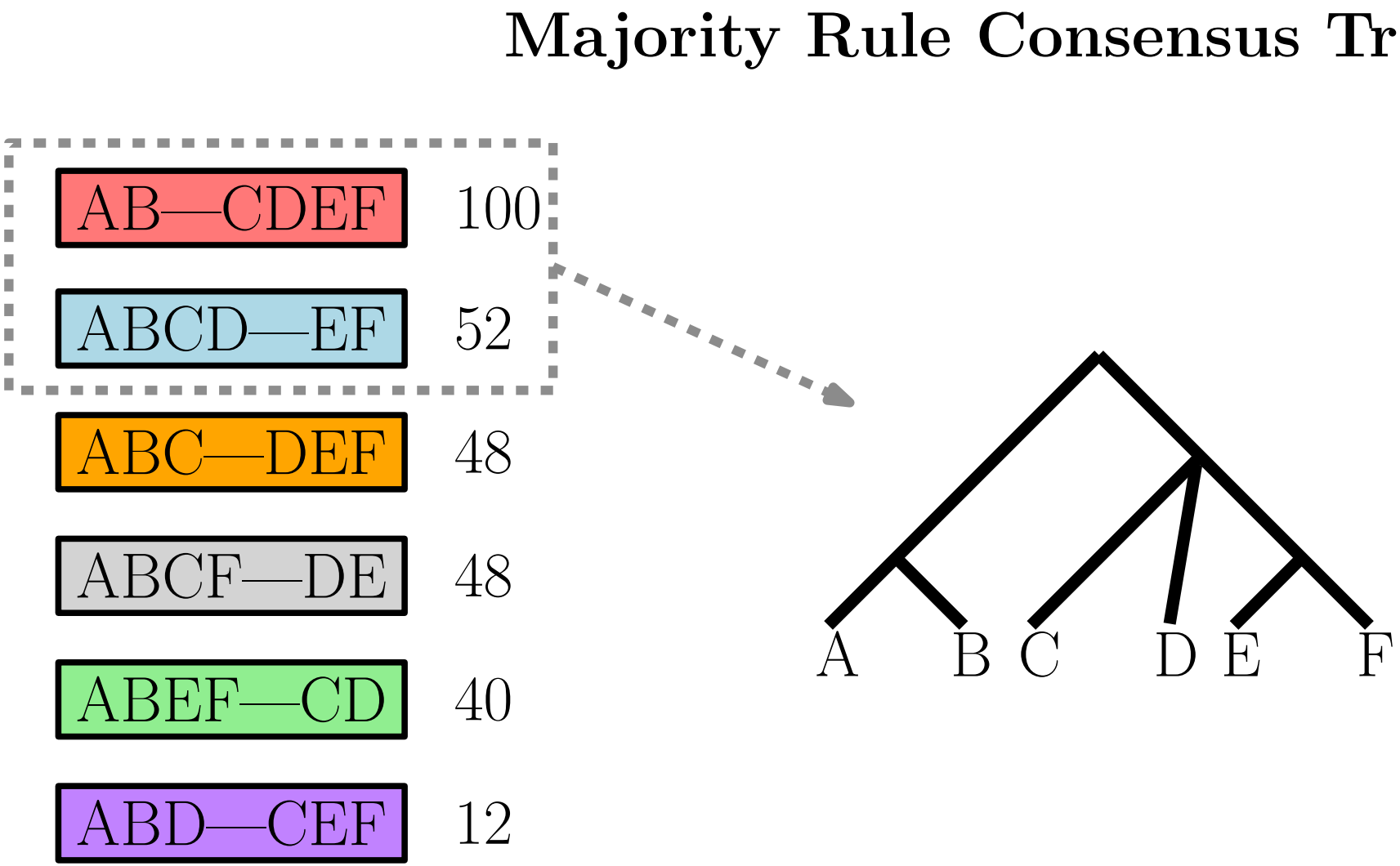


Arbres consensus: un example

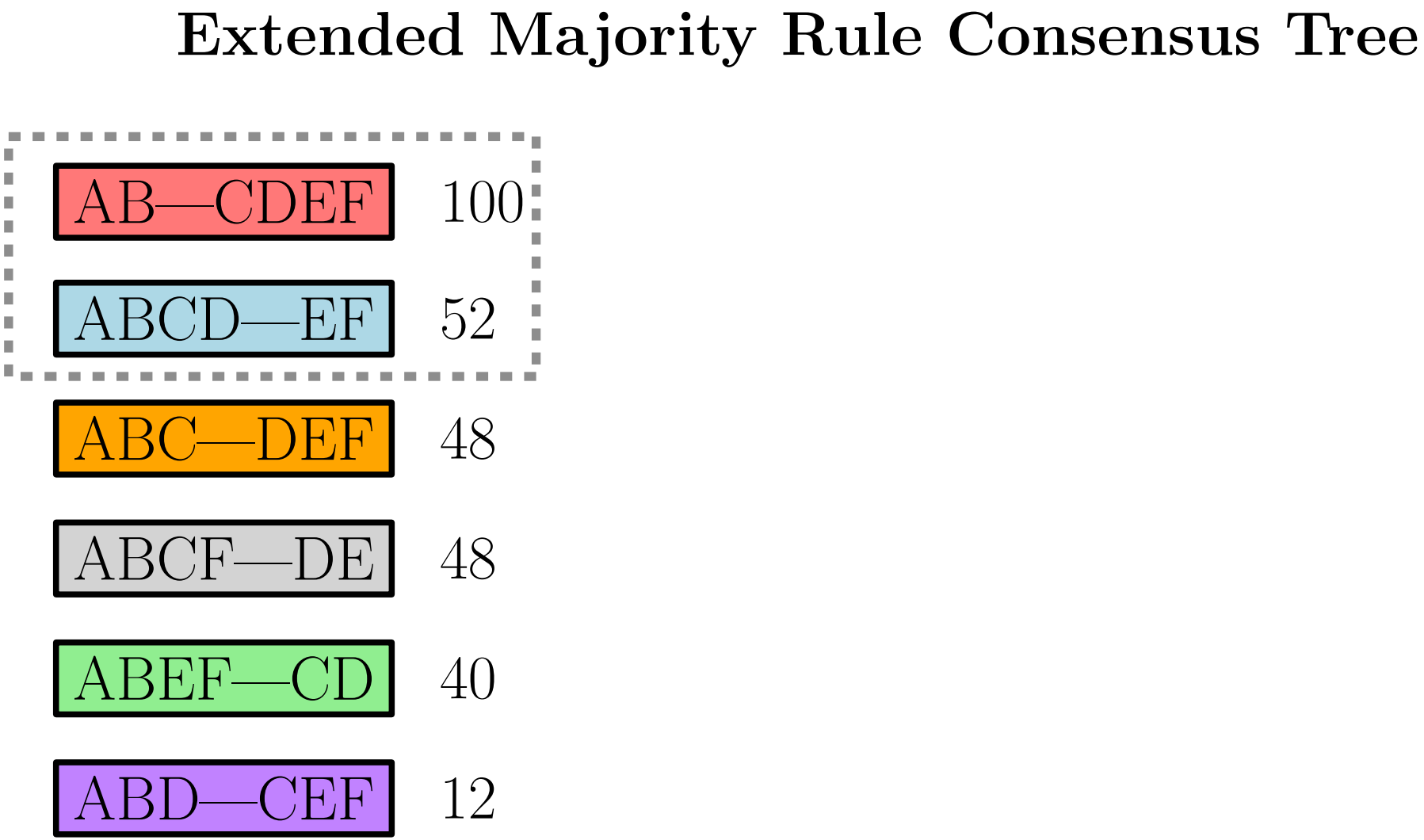
Majority Rule Consensus Tree



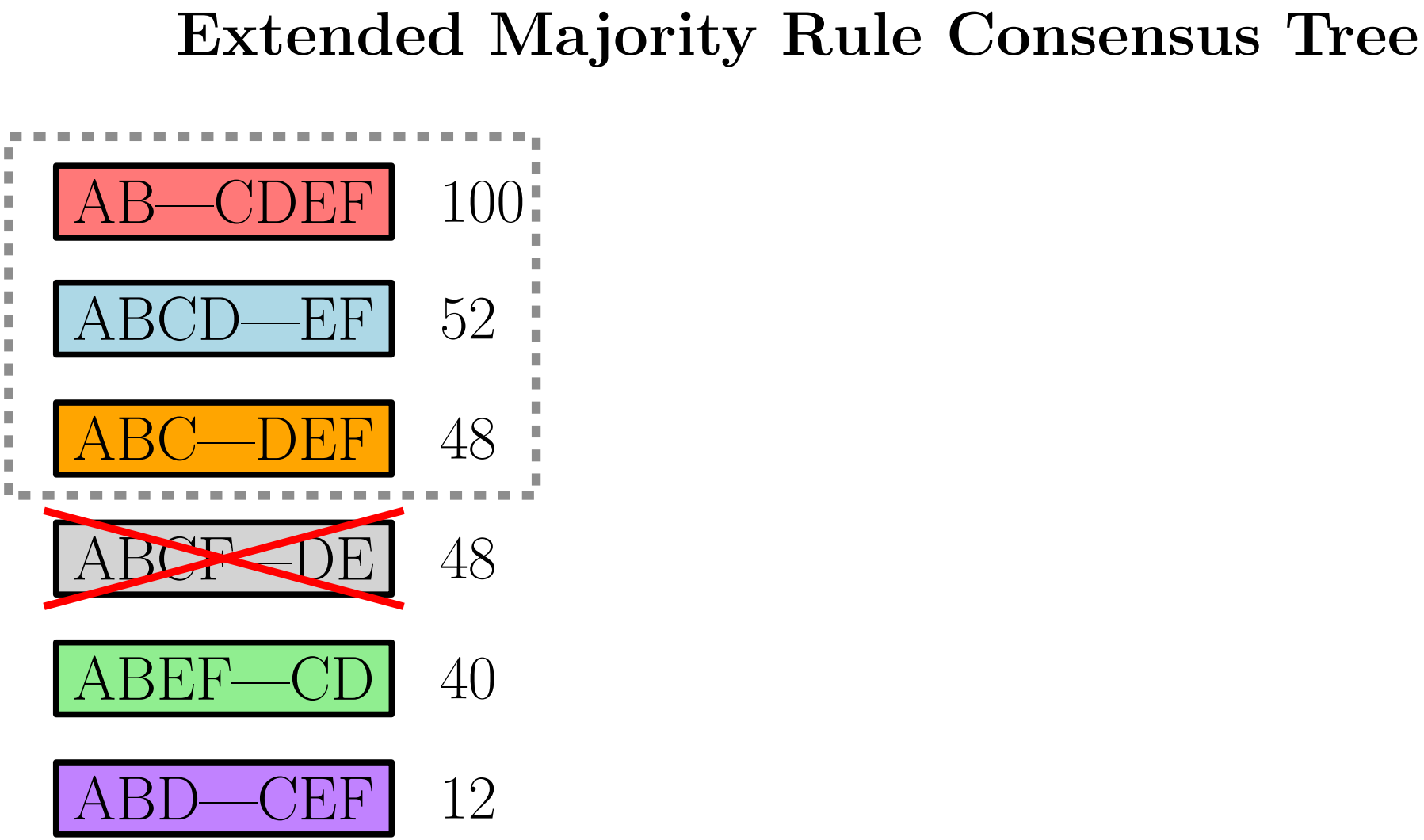
Arbres consensus: un example



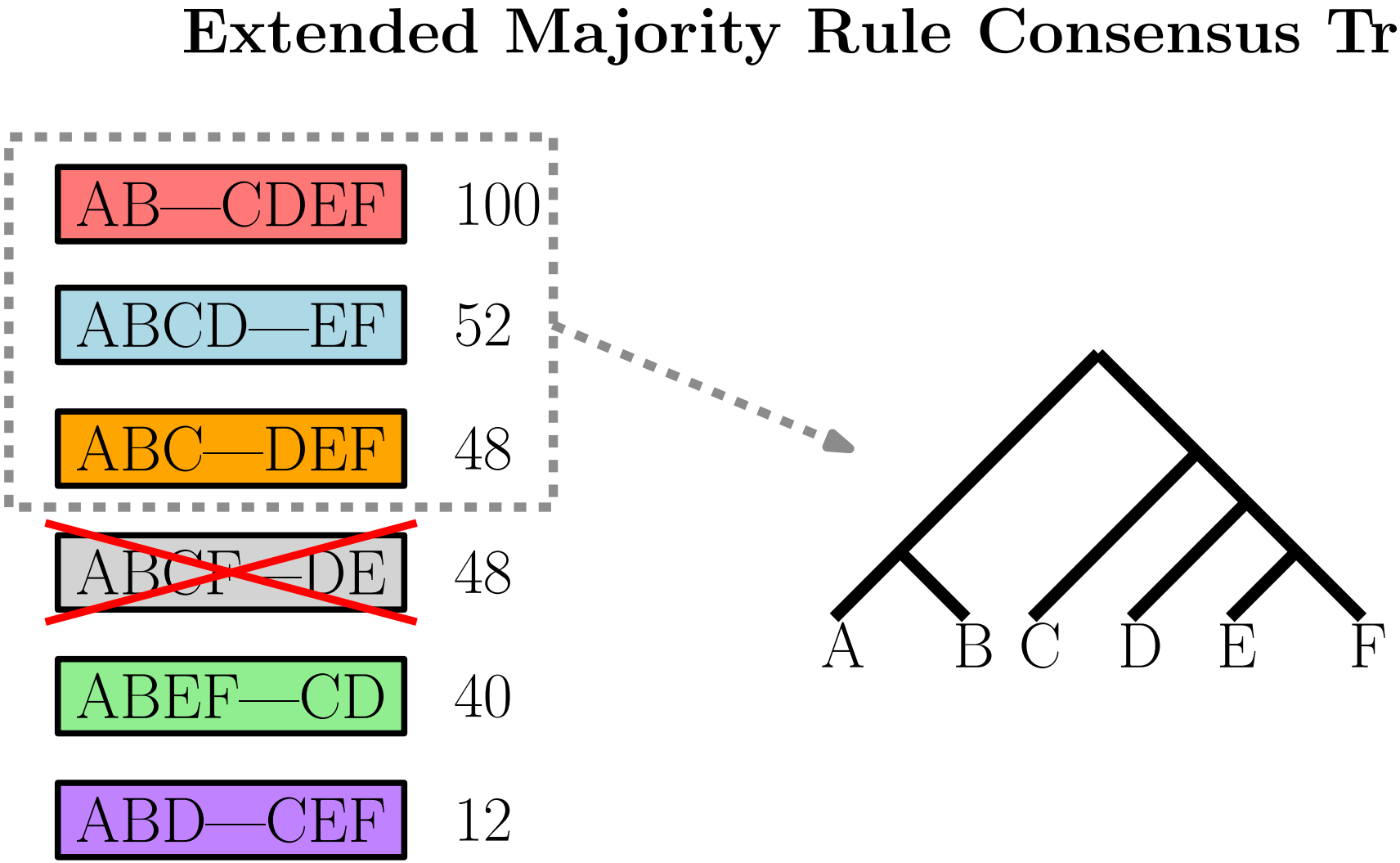
Arbres consensus: un example



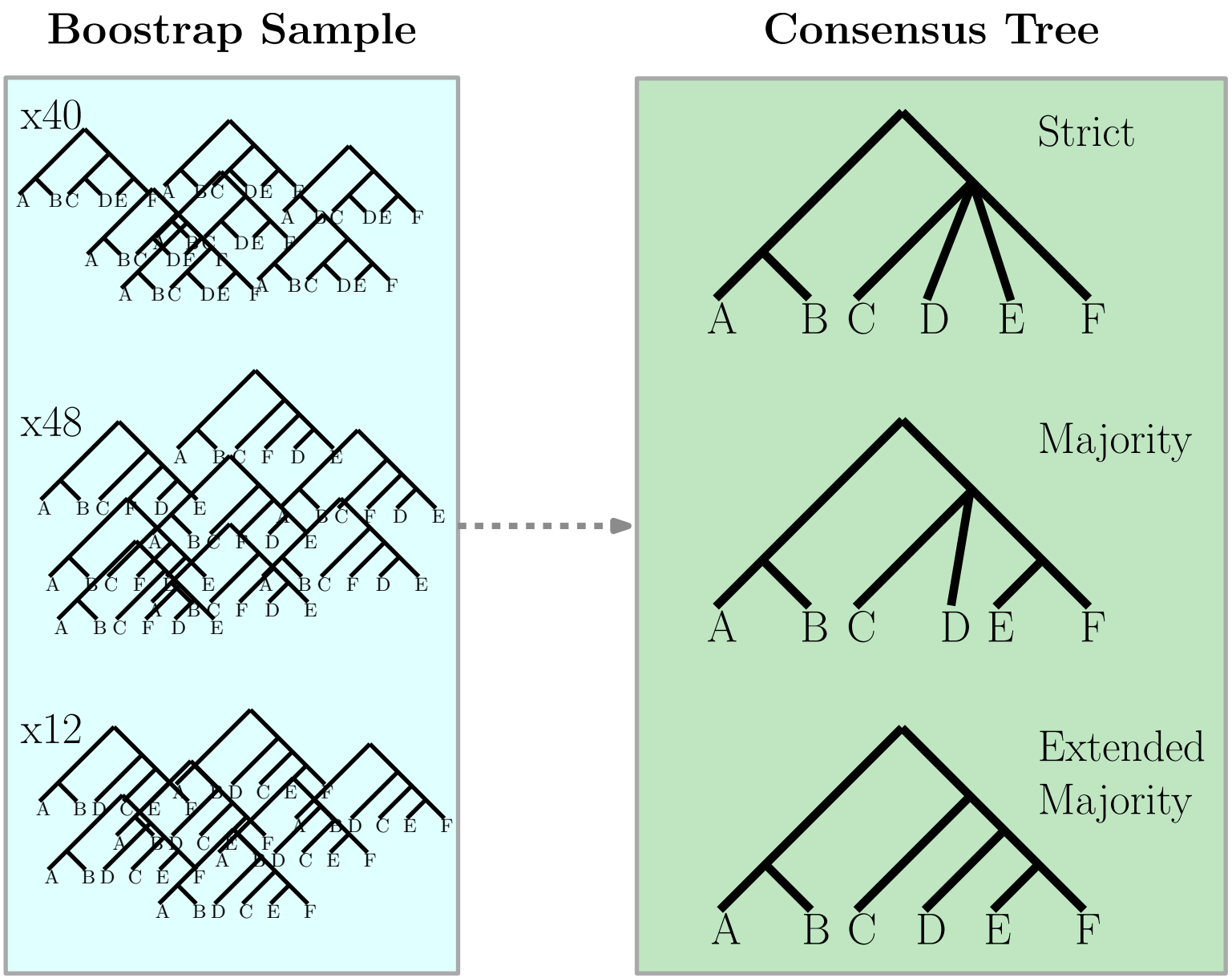
Arbres consensus: un example



Arbres consensus: un example

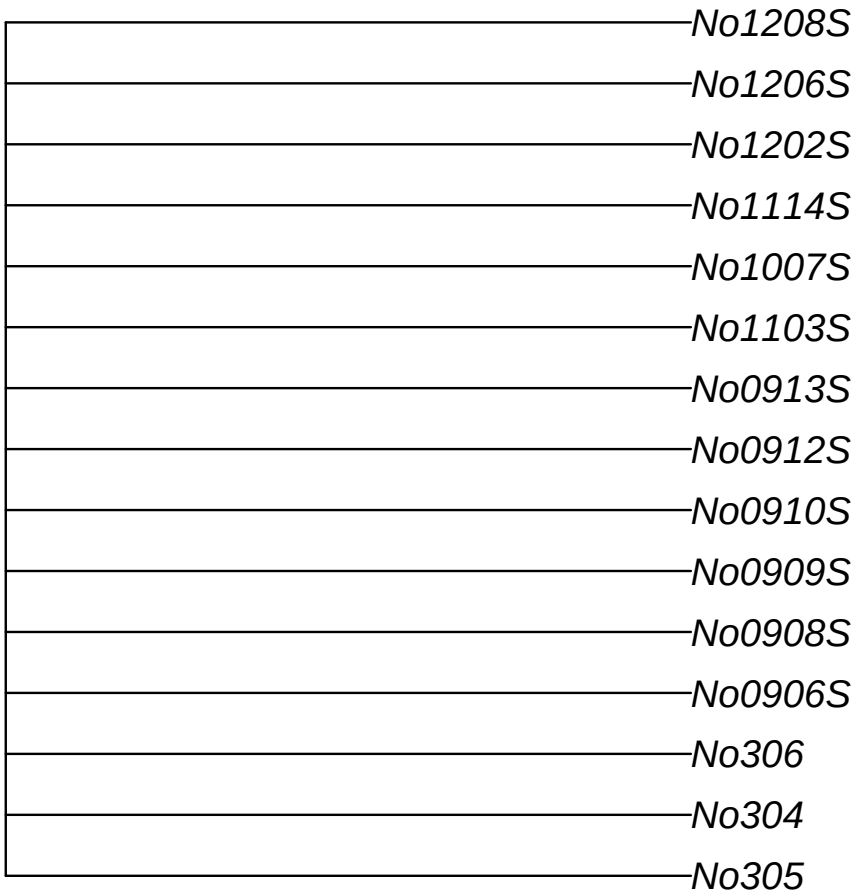


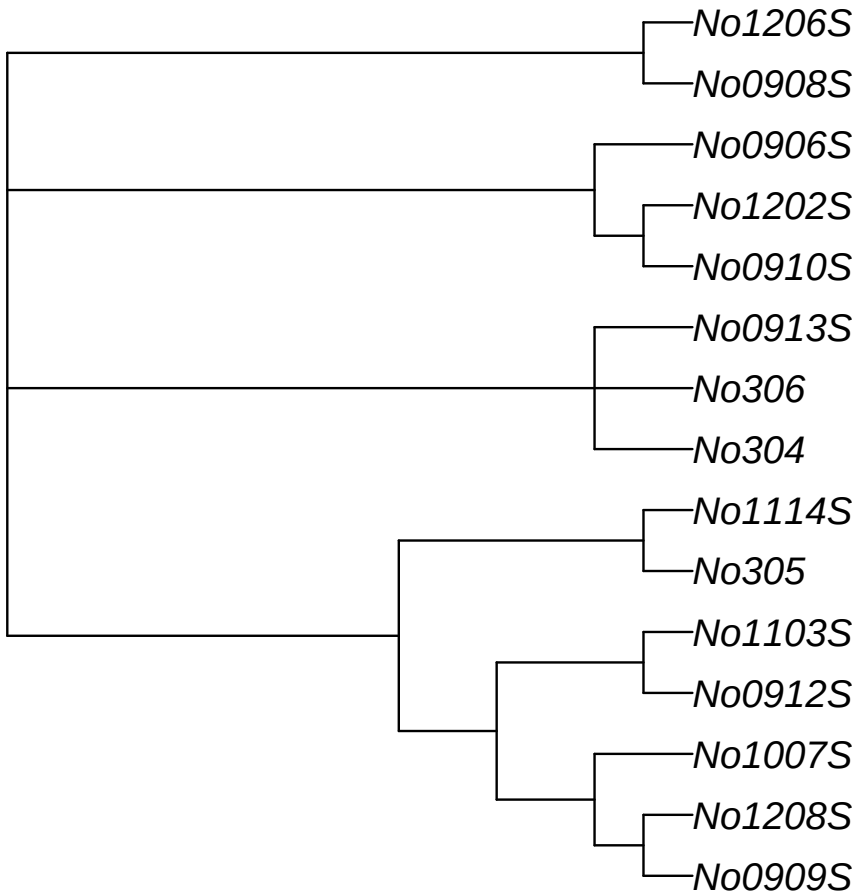
Arbres consensus: un example



Avec consensus

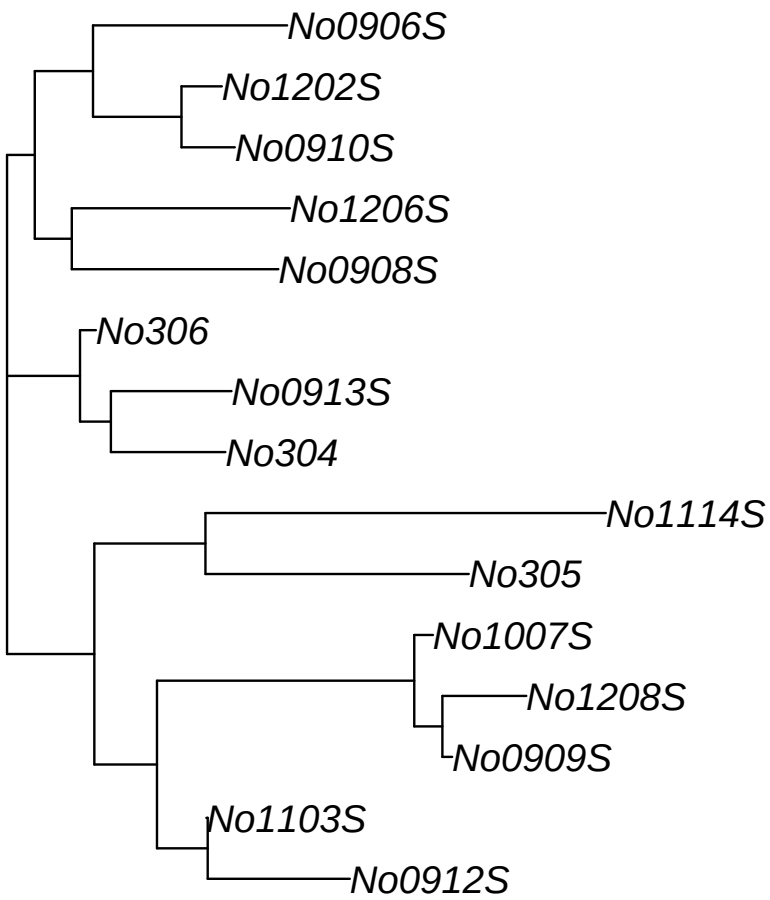
```
> strict.cons.tree <- consensus(boot.trees) ## strict consensus
> maj.cons.tree <- consensus(boot.trees, p = 0.5) ## majority rule
```

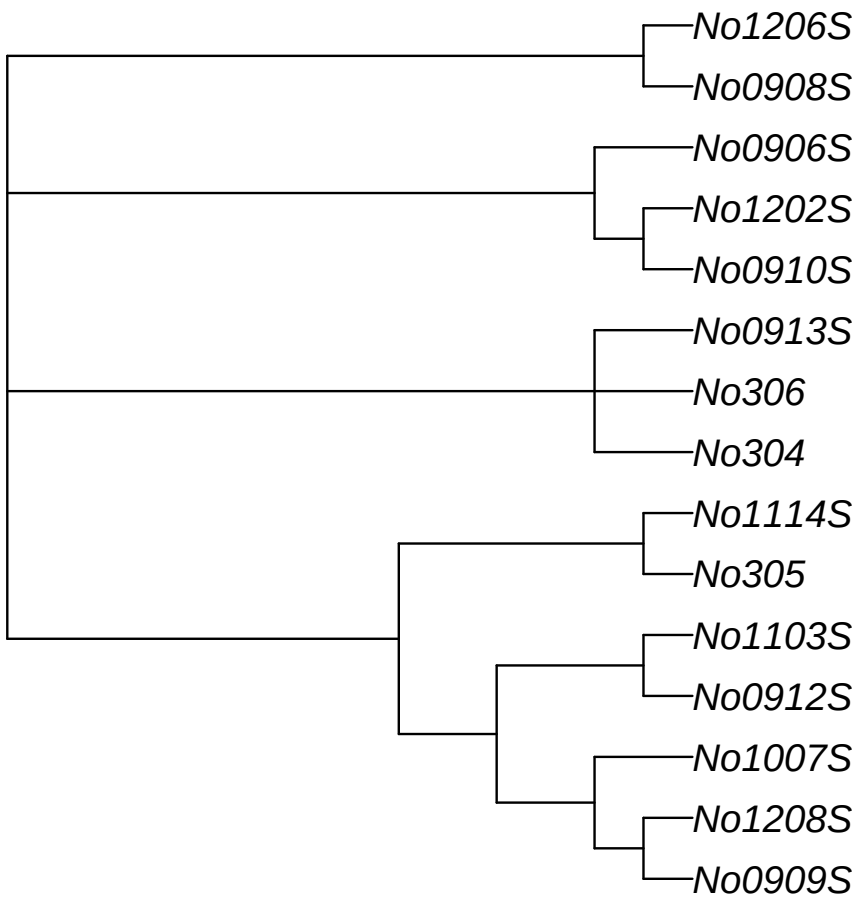




Avec consensus

```
> strict.cons.tree <- consensus(boot.trees) ## strict consensus
> maj.cons.tree <- consensus(boot.trees, p = 0.5) ## majority rule
```





Méthodes de distance

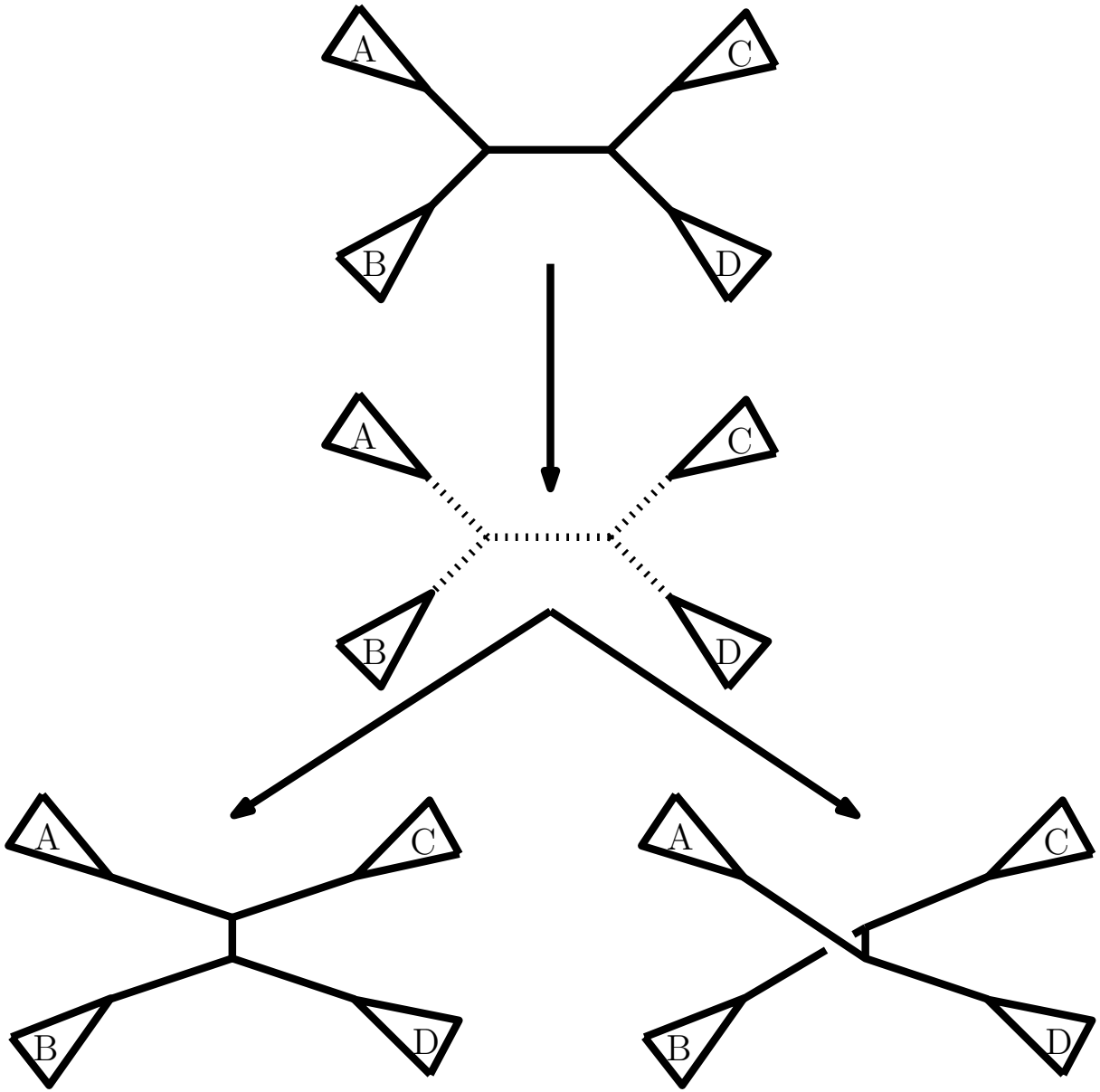
Distances

- Conserver l'information de tous les arbres et essayer de calculer une "moyenne" des arbres en s'appuyant sur une notion de distance.
- Au moins 5 notions de distance entre arbres (NNI, SPR, TBR, RF, BHV)
- Intérêt particulier pour la distance BHV (implémentée dans `distory`), qui prend en compte les longueurs de branches et pour la distance RF qui compte le nombre de bipartitions différentes.

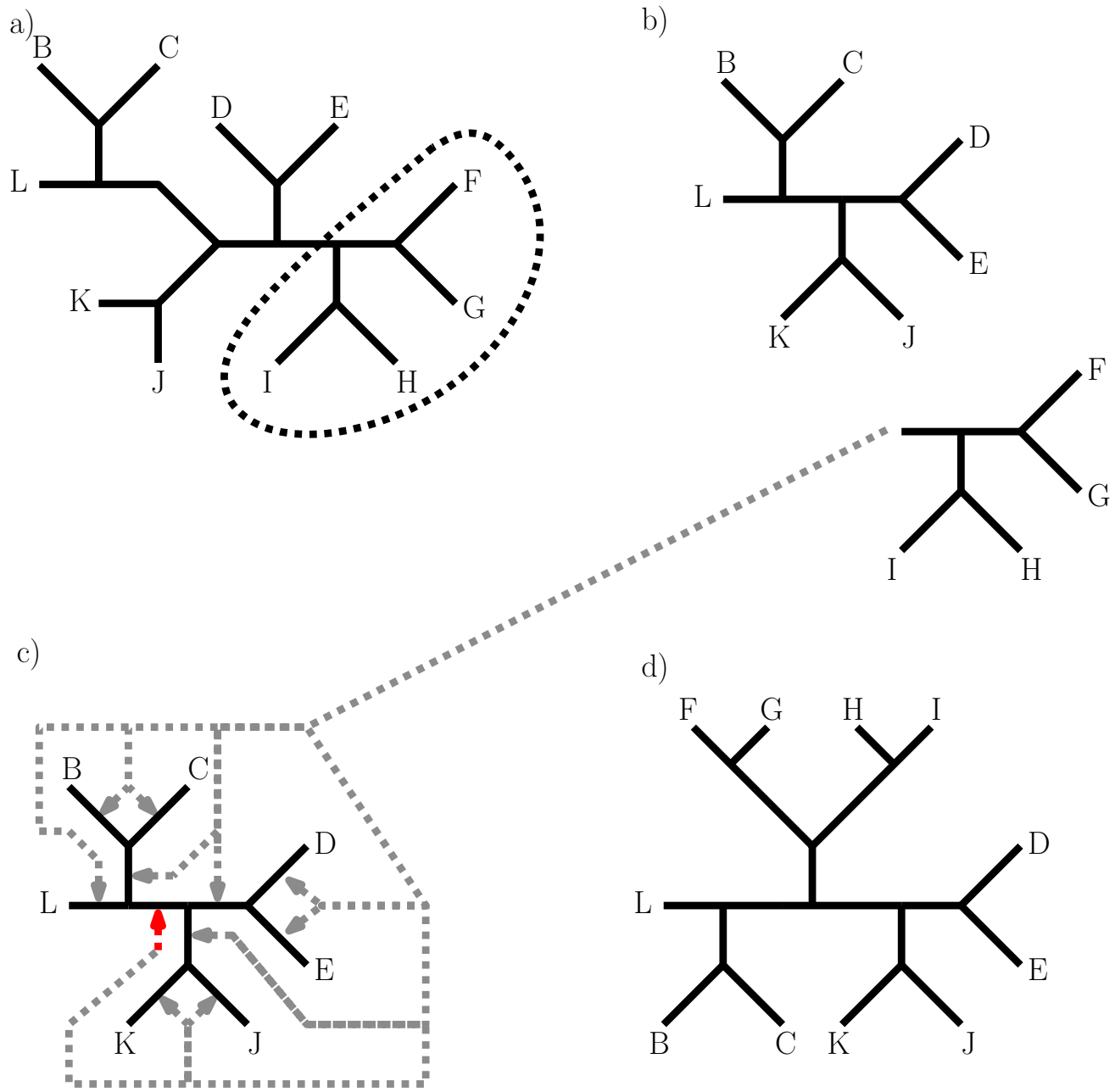
Motivations

- Caractériser le caractère exceptionnel d'une position/d'un taxon.
- À partir de matrice de distances (entre arbres), utiliser des méthodes d'ACP, de MDS pour regrouper les arbres.

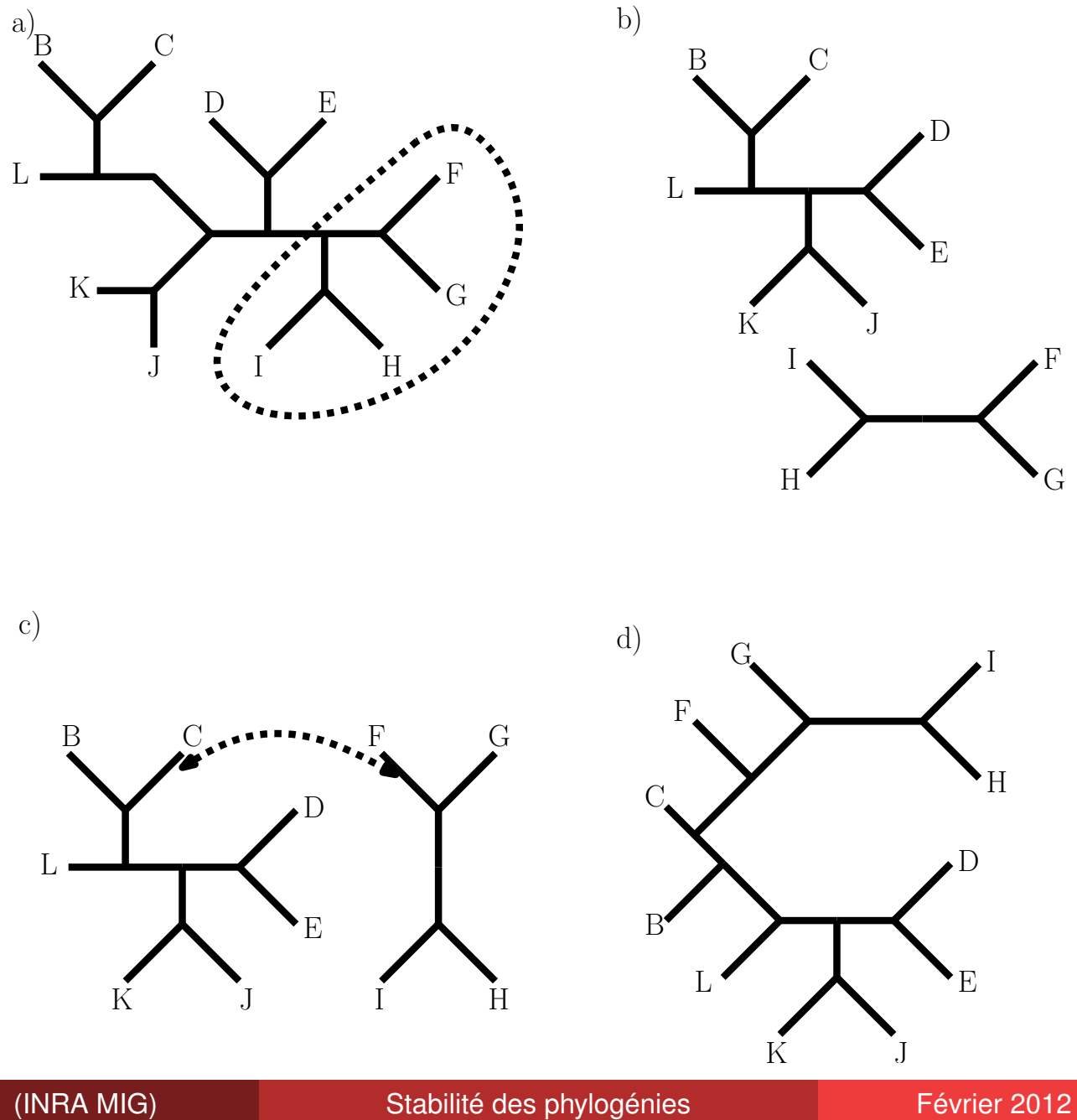
Les différents types de mouvements: NNI



Les différents types de mouvements: SPR



Les différents types de mouvements: TBR



Avec `distory` et ape

`dist.topo`

Pas de contraintes.

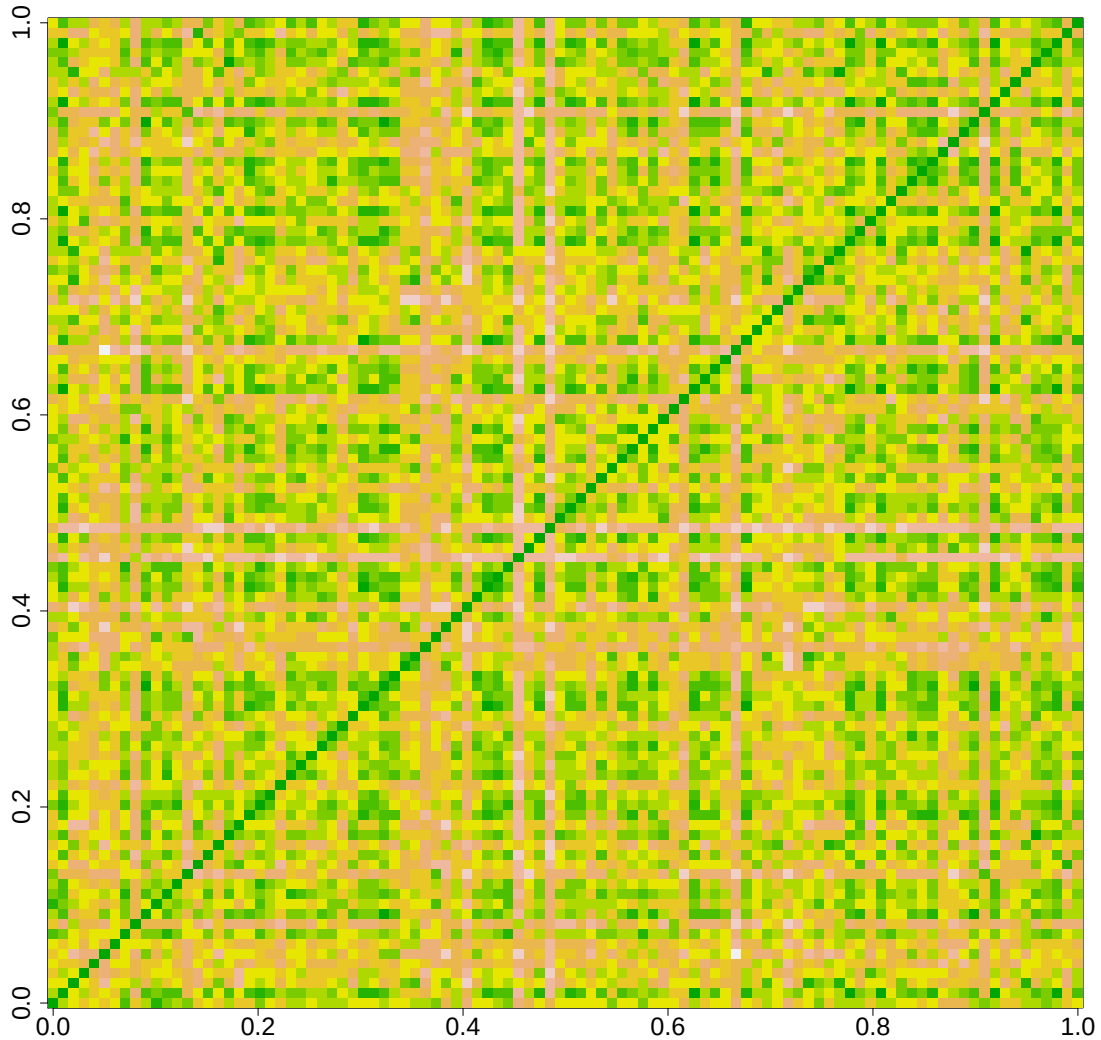
```
> dist.RF <- matrix(0, ncol = 100, nrow = 100)
> for (i in 1:99) {
+ for (j in (i+1):100) {
+ dist.RF[i,j] <- dist.RF[j,i] <- dist.topo(boot.trees[[i]],
boot.trees[[j]]) }}
> image(as.matrix(dist.RF), col = terrain.colors(12))
```

`dist.multiPhylo`

Fournir des arbres racinés résolus sans `$node.label`.

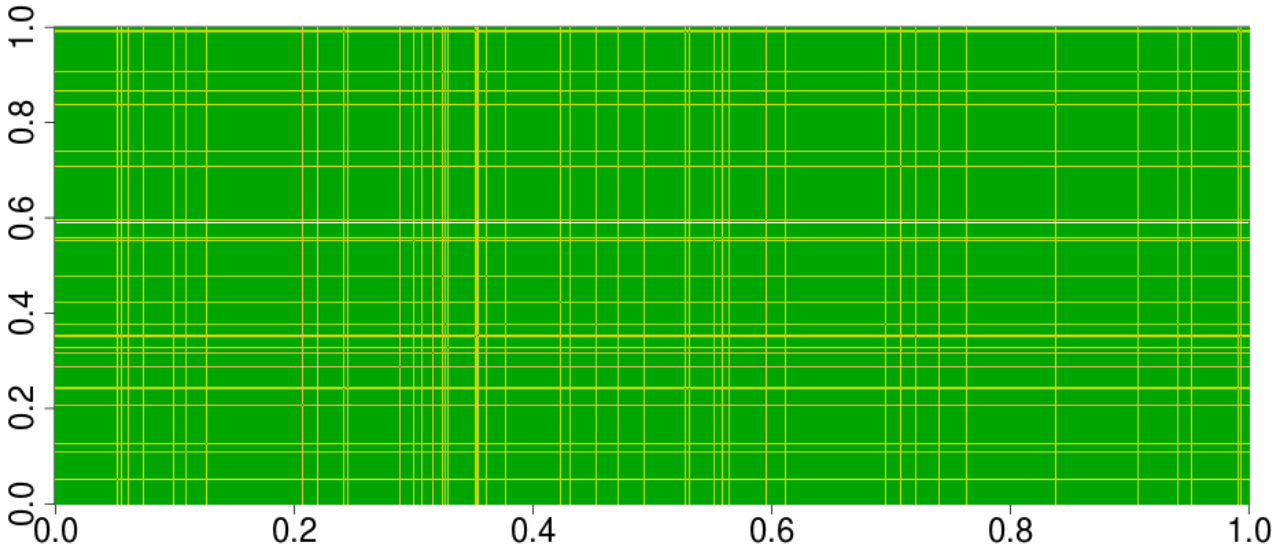
```
> rooted.jack.trees <- lapply(jack.trees, function(x) root(x, outgroup,
+ resolve = TRUE))
> rooted.jack.trees[[966]] <- root(wood.tree, outgroup, resolve = TRUE)
> dist.BHV <- dist.multiPhylo(rooted.jack.trees)
> image(as.matrix(dist.BHV), col = terrain.colors(12))
```

Distance RF pour arbres bootstrap



Pas de structure évidente dans les arbres bootstrap.

Distance BHV pour arbres jackknife



Beaucoup de positions concordantes, quelques positions différentes.

MDS pour distances BHV

MDS pour représenter en 2D les arbres jackknifes et identifier ainsi les positions *exceptionnelles*.

```
> mds.wood <- cmdscale(dist.BHV, k = 4, add=TRUE) ## MDS à 4 dimension
> plot(mds.wood$points[,1], mds.wood$points[,2],
+ col = c(rep("black", 965), "red"),
+ xlab = "Axis 1", ylab = "Axis 2",
+ main = "2D projection of jackknife trees")
> text(mds.wood$points[,1], mds.wood$points[,2],
+ labels = c(seq(965), 0), col = c(rep("black", 965), "red"))
```

MDS pour distances BHV