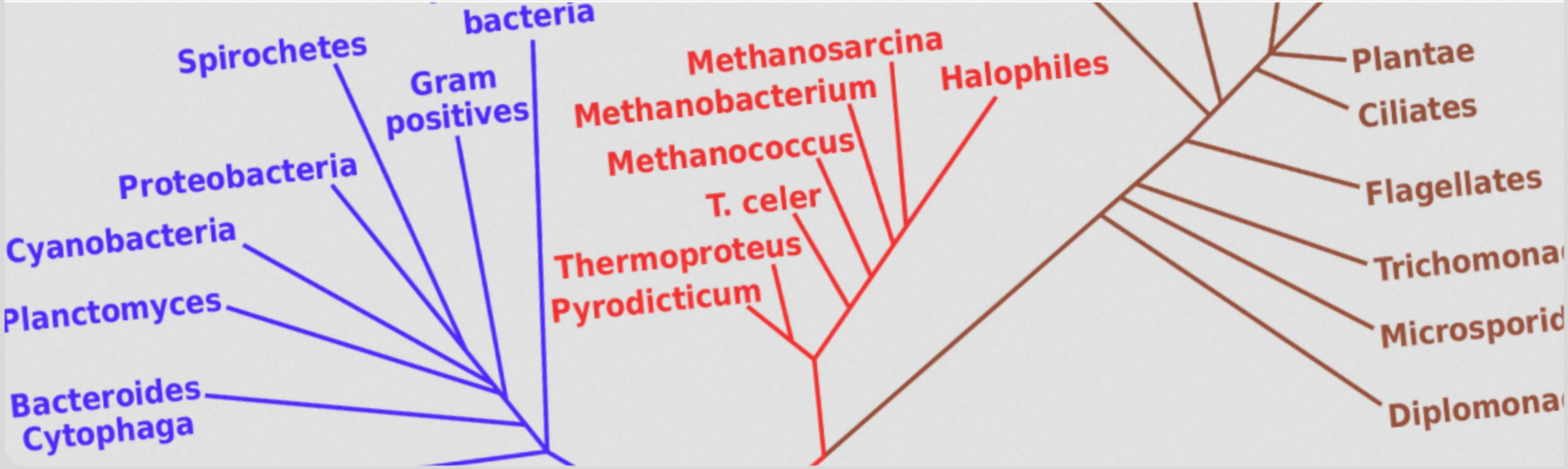
Karlsruher Institut für Technologie

Discrete Operations on Sets of Phylogenetic Trees

Introduction to Bioinformatics for Computer Scientists
Lecture 10

Pierre Barbera, HITS



Outline

- Bootstrapping in Phylogenetics
- Uses of the Bootstrap Tree Set
  - Support For Best-Known Tree
  - Consensus of Trees
- Distances between Trees
- Extra: Rogue Taxa

Tree Sets - Where do They Come From?

- distinct equally parsimonious trees (same parsimony score)
- distinct greedy maximum likelihood tree searches
- Bayesian tree search (millions of trees)

Tree Sets - Where do They Come From?

- distinct equally parsimonious trees (same parsimony score)
- distinct greedy maximum likelihood tree searches
- Bayesian tree search (millions of trees)
- Trees inferred on different datasets, for instance
  - 100 genes ⇒ best tree per gene
  - or one best tree found on concatenated alignment

Tree Sets - Where do They Come From?

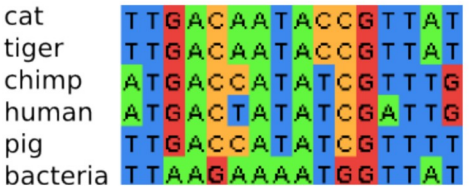
- distinct equally parsimonious trees (same parsimony score)
- distinct greedy maximum likelihood tree searches
- Bayesian tree search (millions of trees)
- Trees inferred on different datasets, for instance
  - 100 genes ⇒ best tree per gene
  - or one best tree found on concatenated alignment
- Also important: bootstrap trees

Outline

- **Bootstrapping in Phylogenetics**
- Uses of the Bootstrap Tree Set
  - Support For Best-Known Tree
  - Consensus of Trees
- Distances between Trees
- Extra: Rogue Taxa

Motivation: Searching for the Best Tree

Data=



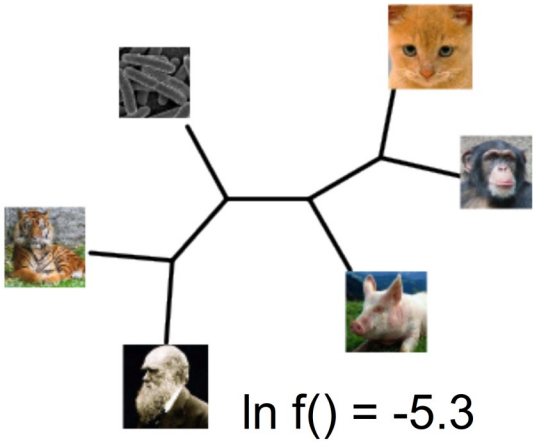
f = likelihood = P(Data | Model,Tree)

Motivation: Searching for the Best Tree

Data=

cat	TTGACAATACCGTTAT
tiger	TTGACAATACCGTTAT
chimp	ATGACCATATCGTTTG
human	ATGACTATATCGATTG
pig	TTGACCATATCGTTTT
bacteria	TTAAGAAAATGGTTAT

f = likelihood = P(Data | Model,Tree)



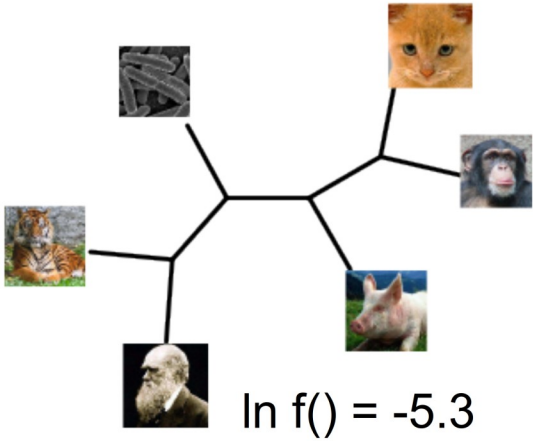
ln f() = -5.3

Motivation: Searching for the Best Tree

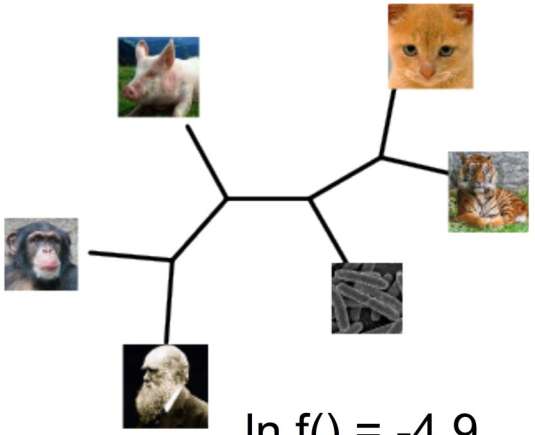
Data=

cat	TTGACAATACCGTTAT
tiger	TTGACAATACCGTTAT
chimp	ATGACCATATCGTTTG
human	ATGACTATATCGATTG
pig	TTGACCATATCGTTTT
bacteria	TTAAGAAAATGGTTAT

f = likelihood = P(Data | Model,Tree)



ln f() = -5.3



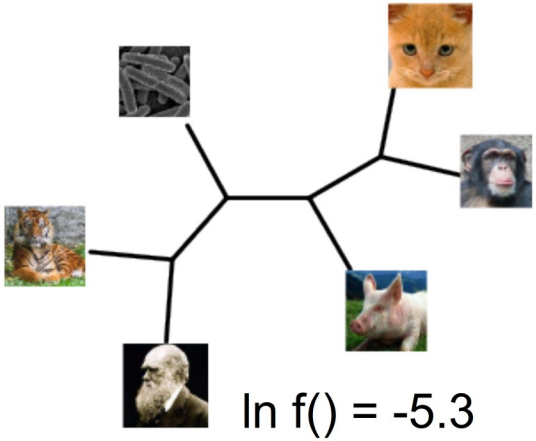
ln f() = -4.9

Motivation: Searching for the Best Tree

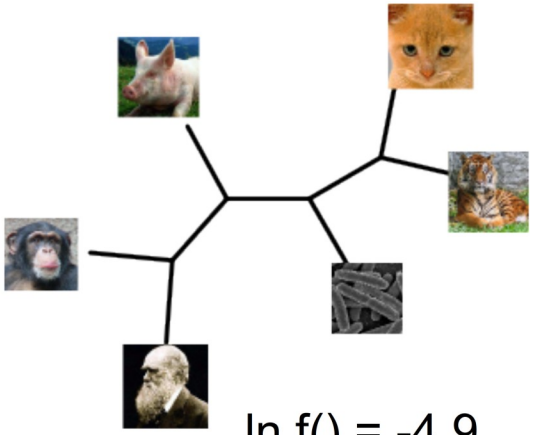
Data=

cat	TTGACAATACCGTTAT
tiger	TTGACAATACCGTTAT
chimp	ATGACCATATCGTTTG
human	ATGACTATATCGATTG
pig	TTGACCATATCGTTTT
bacteria	TTAAGAAAATGGTTAT

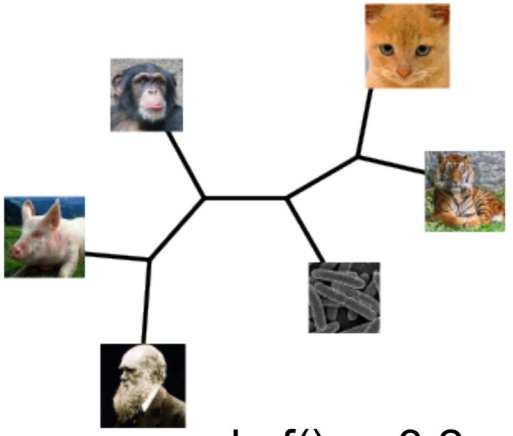
f = likelihood = P(Data | Model,Tree)



ln f() = -5.3



ln f() = -4.9



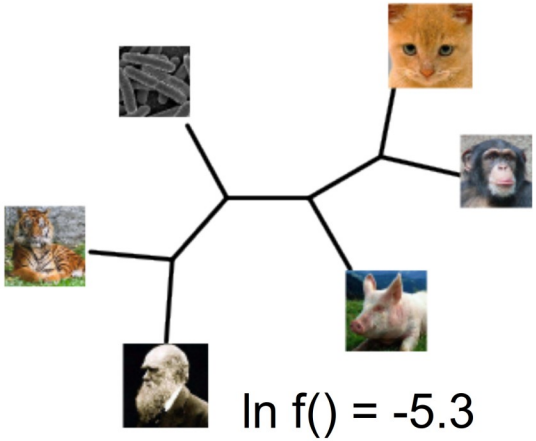
ln f() = -3.2

Motivation: Searching for the Best Tree

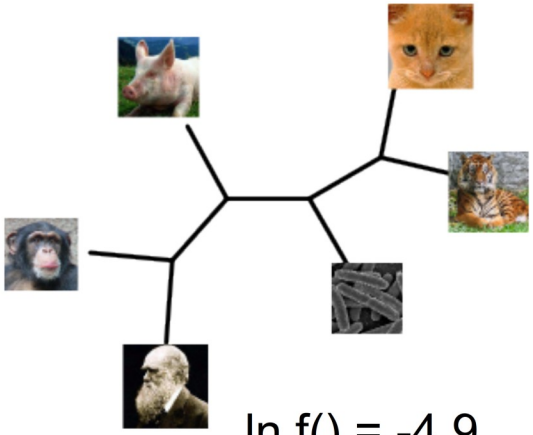
Data=

cat	TTGACAATACCGTTAT
tiger	TTGACAATACCGTTAT
chimp	ATGACCATATCGTTTG
human	ATGACTATATCGATTG
pig	TTGACCATATCGTTTT
bacteria	TTAAGAAAATGGTTAT

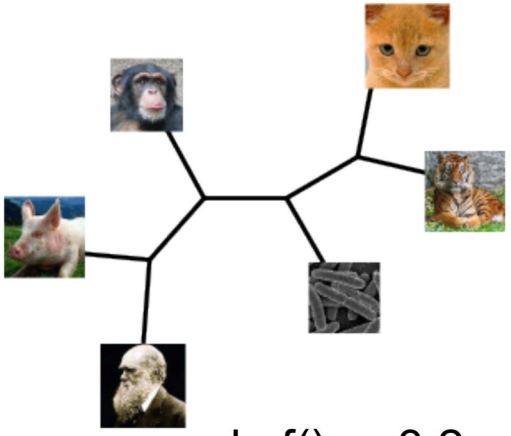
f = likelihood = P(Data | Model,Tree)



ln f() = -5.3



ln f() = -4.9



ln f() = -3.2

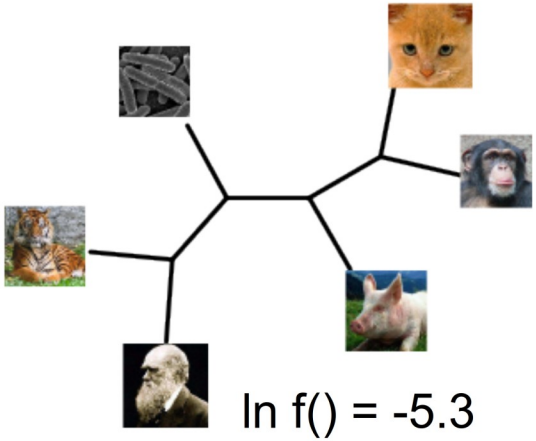
To what degree does *Data* support the tree?

Motivation: Searching for the Best Tree

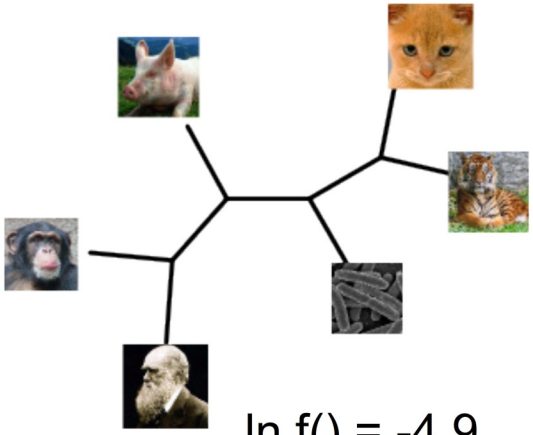
Data=

cat	TTGACAATACCGTTAT
tiger	TTGACAATACCGTTAT
chimp	ATGACCATATCGTTTG
human	ATGACTATATCGATTG
pig	TTGACCATATCGTTTT
bacteria	TTAAGAAAATGGTTAT

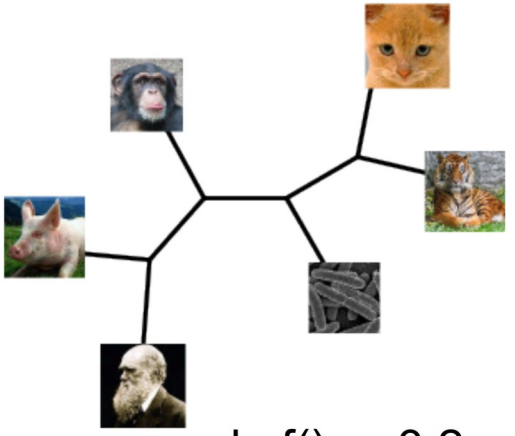
f = likelihood = P(Data | Model,Tree)



ln f() = -5.3



ln f() = -4.9



ln f() = -3.2

To what degree does *Data* support the tree?

What about *Data2*?

Data2=

cat	TTGACAATACCGTTAT
tiger	TTGACAATACCGTTAT
chimp	ATGACCATATCGTTTG
human	ATGACTATATCGATTG
pig	----CCA----------
bacteria	TTAAGAAAATGGTTAT

Confidence Measures in Bioinformatics

- In Bioinformatics, for most results, we need to provide confidence

Confidence Measures in Bioinformatics

- In Bioinformatics, for most results, we need to provide confidence
- For instance: motif search

ATGATAGTAGCGTACATCGTATCGTATGATCGATG

p-values: Do we find motif **AT** significantly more often than expected by chance?

Confidence Measures in Phylogenetics

- We want to know:

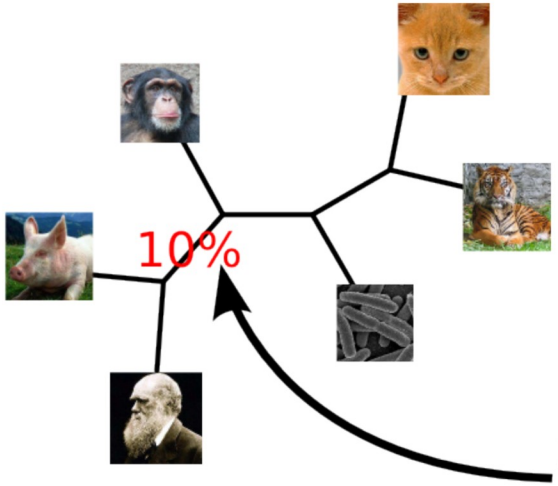
  To what degree does the data support evolutionary relationships?

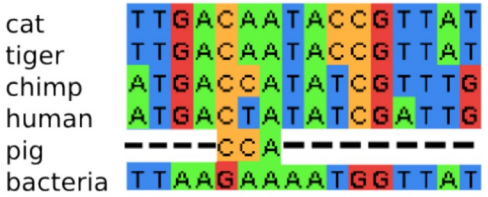
Confidence Measures in Phylogenetics

- We want to know:

  To what degree does the data support evolutionary relationships?

- ⇔ support values on inner branches





supported only by
10% of the data

The Non-Parametric Bootstrap

Bootstrapping in Statistics

- Essential statistical re-sampling technique
  (among jackknifing, Bayesian re-sampling methods, . . .)

- Instead of parametric statistical testing
  (assume distribution, calculate confidence intervals)

Bootstrapping in Statistics

- Essential statistical re-sampling technique
  (among jackknifing, Bayesian re-sampling methods, . . .)
- Instead of parametric statistical testing
  (assume distribution, calculate confidence intervals)
- **Non-parametric** bootstrap: resample a dataset, calculate statistics
  (e.g., mean) ⇒ obtain confidence

Bootstrapping in Statistics

- Essential statistical re-sampling technique
  (among jackknifing, Bayesian re-sampling methods, . . .)
- Instead of parametric statistical testing
  (assume distribution, calculate confidence intervals)
- **Non-parametric** bootstrap: resample a dataset, calculate statistics
  (e.g., mean) ⇒ obtain confidence
- Effron: *“Pulling oneself up by one’s bootstraps”*
  ⇒ computationally more expensive
- Further reading:
  https://projecteuclid.org/euclid.aos/1176344552

Biological Background

- Evolution:
  a stochastic process that yields a multinomial distribution of evolutionary events (→ alignment sites)

Biological Background

- Evolution:
  a stochastic process that yields a multinomial distribution of evolutionary events (→ alignment sites)

- We bootstrap new replicate alignments from original alignment
  ⇒ a bootstrap replicate strives to be an alternative version of the original alignment

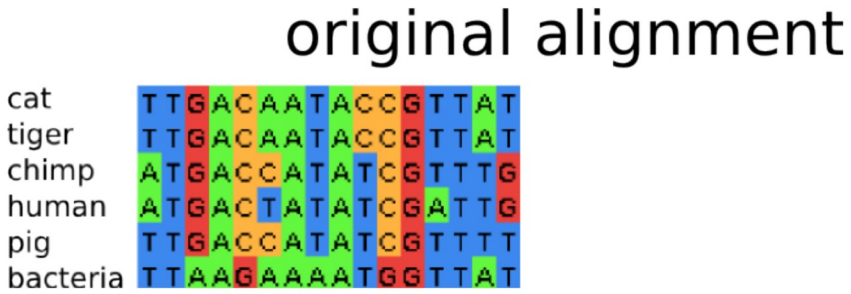
- But: does not introduce new sites that have not been observed

Creating Bootstrap Alignment Replicates

- From an alignment A with length n,
  re-sample n columns / sites with replacement

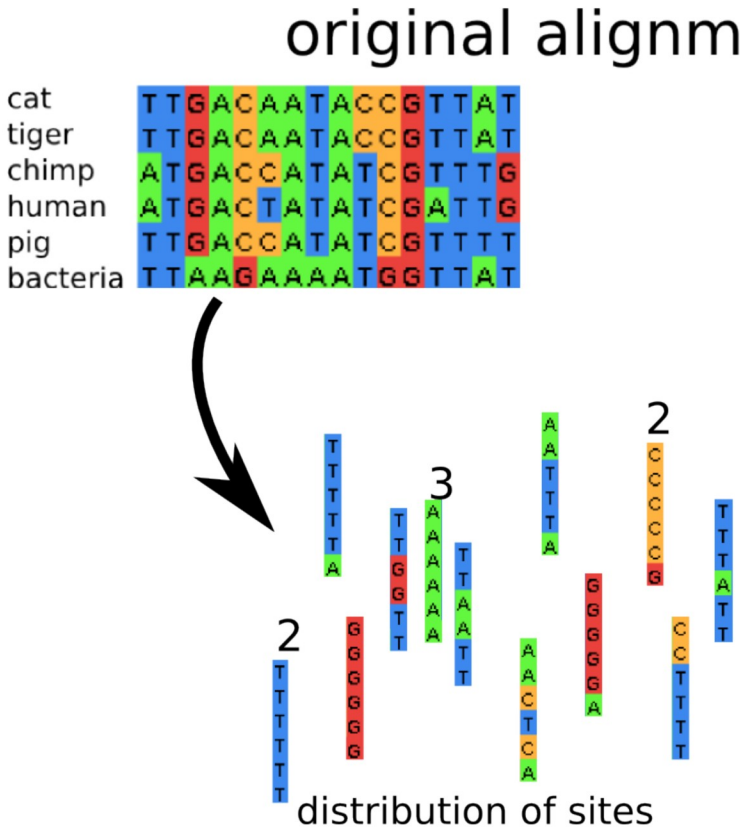
Creating Bootstrap Alignment Replicates

- From an alignment A with length n,
  re-sample n columns / sites with replacement



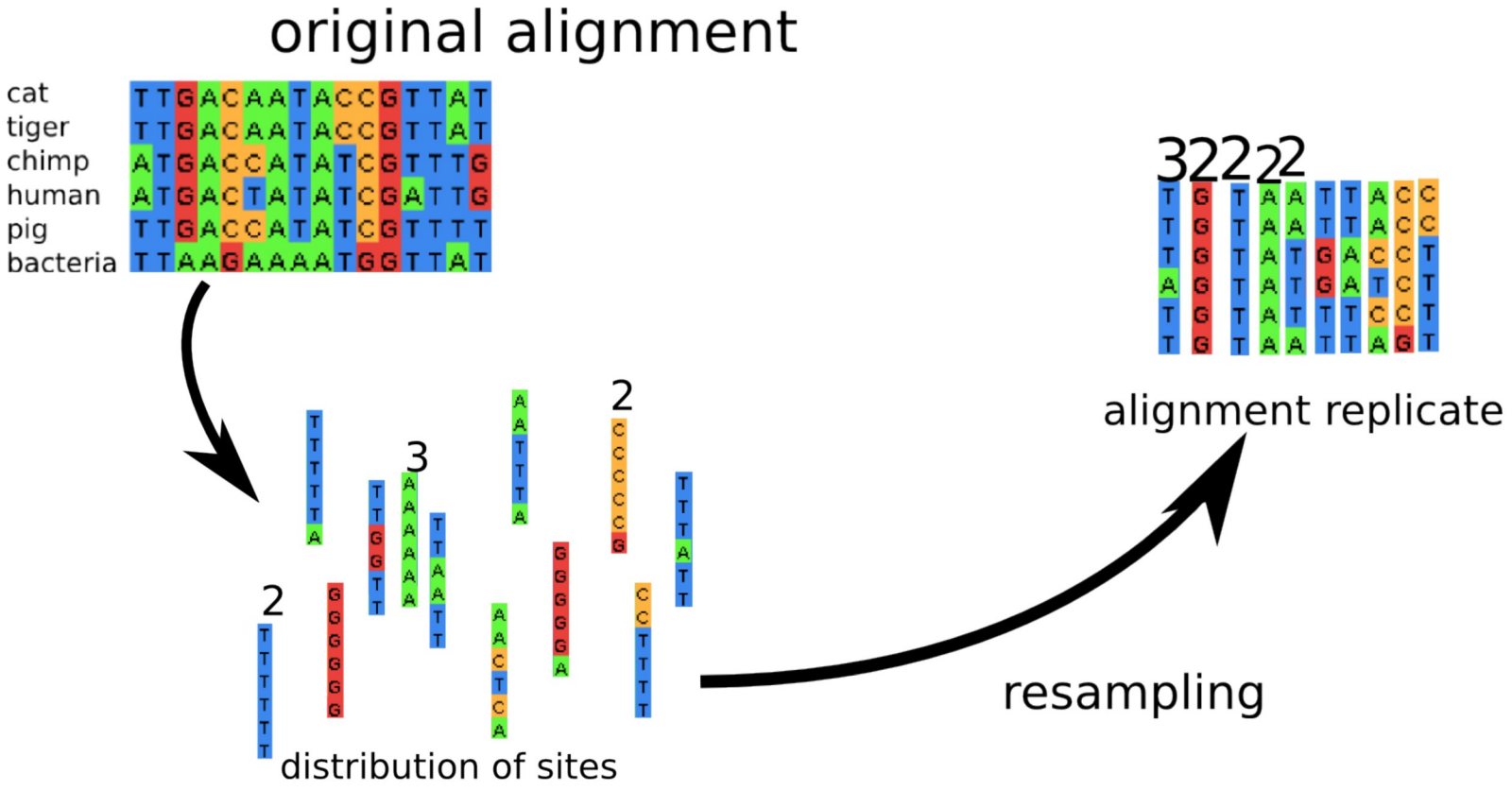
Creating Bootstrap Alignment Replicates

- From an alignment A with length n,
  re-sample n columns / sites with replacement

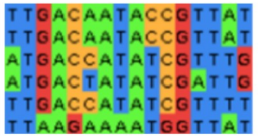


Creating Bootstrap Alignment Replicates

- From an alignment A with length n,
  re-sample n columns / sites with replacement

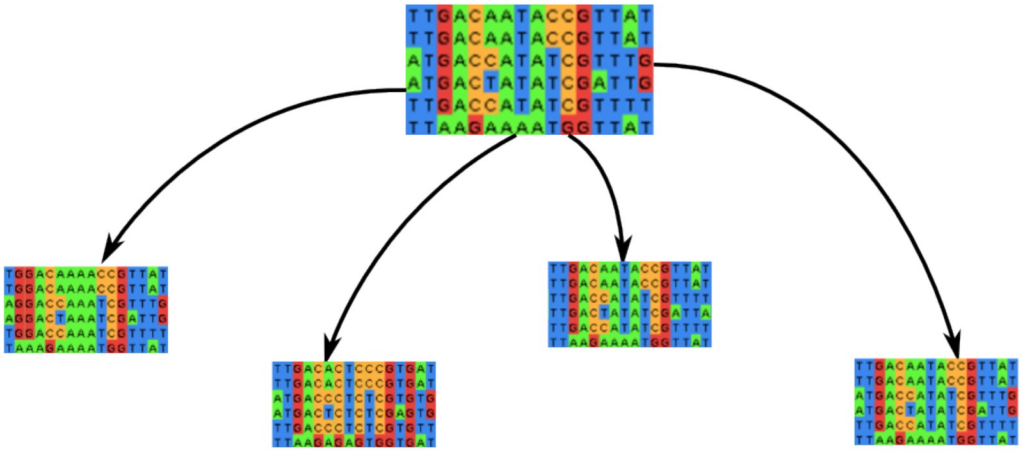


Inferring a Bootstrap Tree Set



original alignment

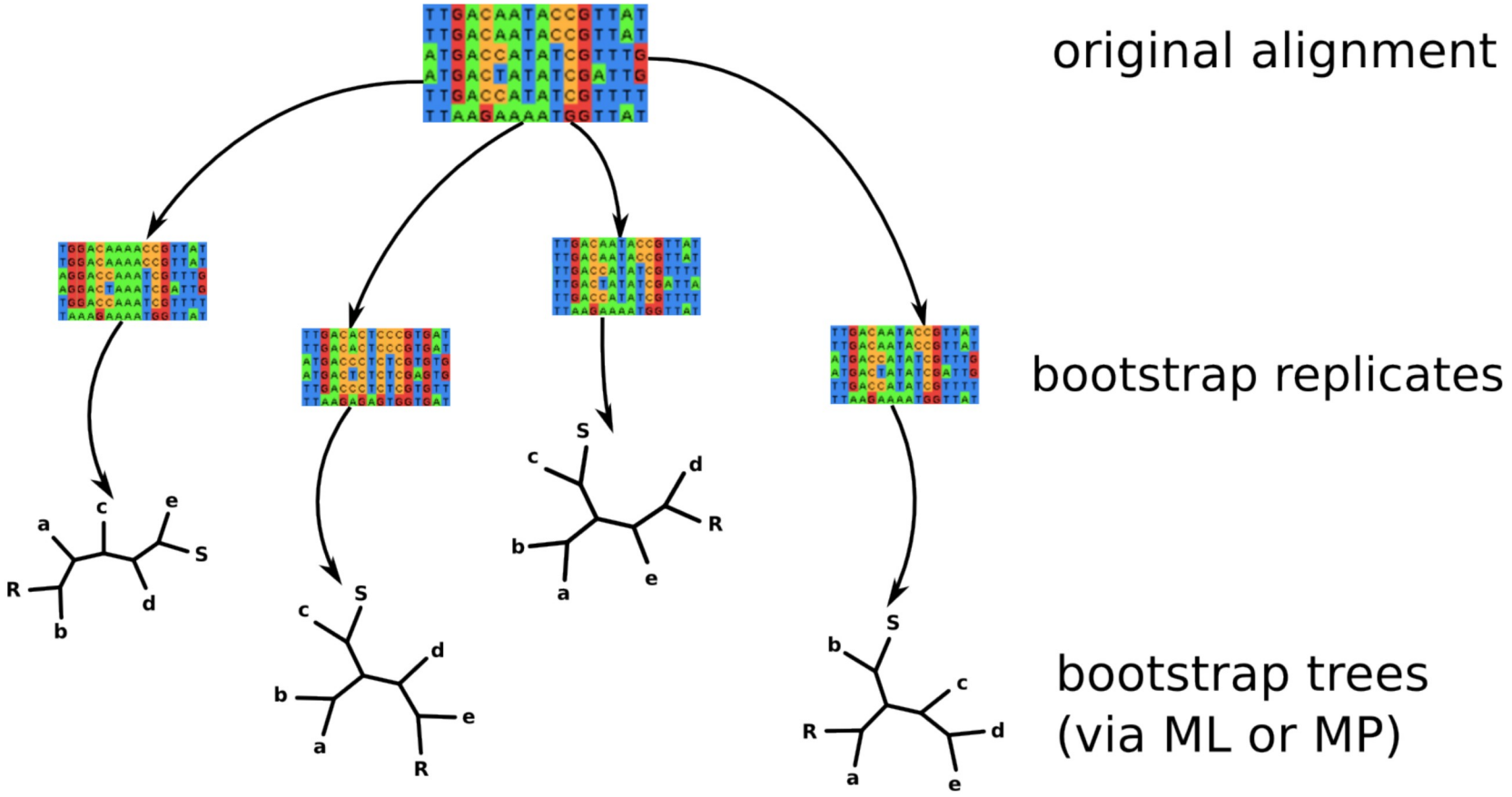
Inferring a Bootstrap Tree Set



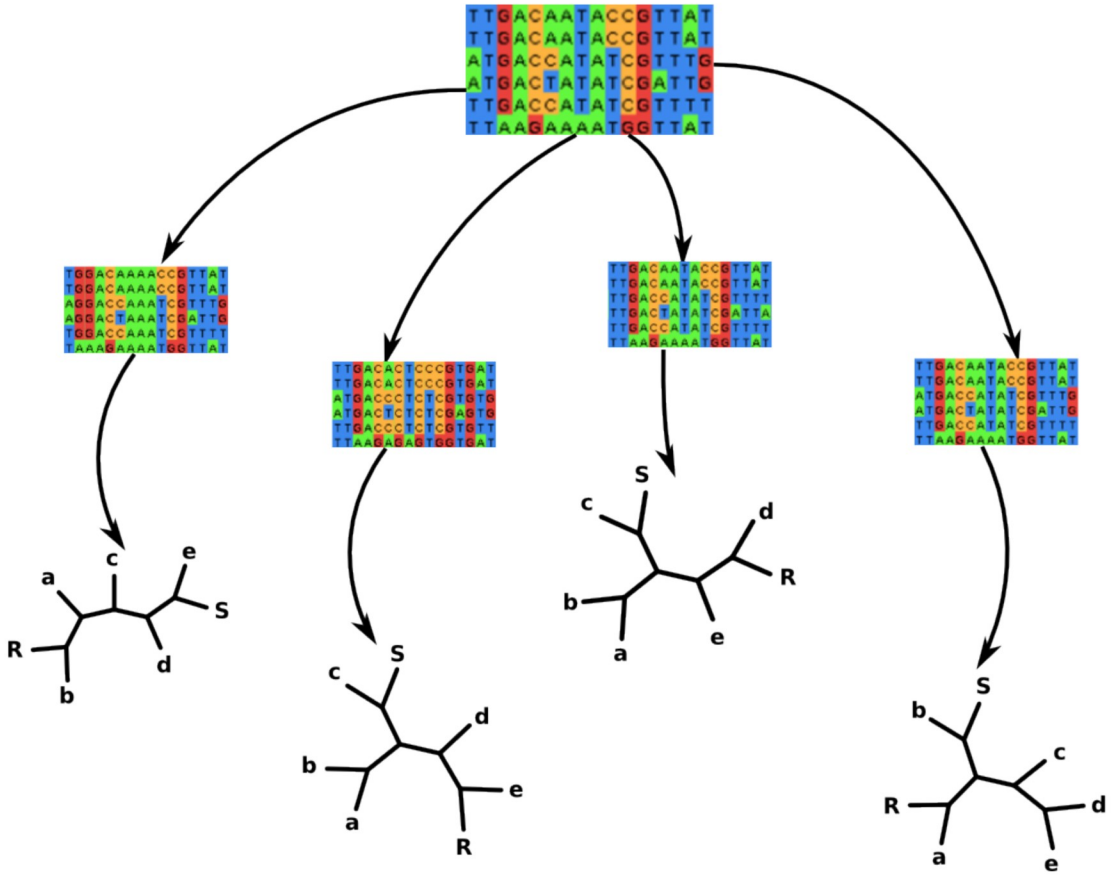
original alignment

bootstrap replicates

Inferring a Bootstrap Tree Set



Inferring a Bootstrap Tree Set



Now what?

Facts about the Phylogenetic Bootstrap

- For n sites, probability for site to be sampled: $1/n$

Phylogenetic Bootstrap

- Probability for site to occur in alignment replicate:

$\frac{1}{n}$ Probability of a site being sampled once

$1 - \frac{1}{n}$ Probability of a site **not** being sampled once

$\left(1 - \frac{1}{n}\right)^n$ Probability of a site not being sampled in any of the replicate sites

$1 - \left(1 - \frac{1}{n}\right)^n$ Probability of a site **being** sampled in at least one replicate site

In total:

$$Pr(X \geq 1) = 1 - \left(1 - \frac{1}{n}\right)^n \simeq 1 - e^{-1} \approx 63.2$$

Facts about the Phylogenetic Bootstrap

- For *n* sites, probability for site to be sampled: 1/*n*
- Probability for site to occur in alignment replicate

$$Pr(X \geq 1) \;=\; 1 - \left(1 - \frac{1}{n}\right)^{n} \;\simeq\; 1 - e^{-1} \;\approx\; 63.2$$

Facts about the Phylogenetic Bootstrap

- For *n* sites, probability for site to be sampled: 1/*n*
- Probability for site to occur in alignment replicate

$$Pr(X \geq 1) \;=\; 1 - \left(1 - \frac{1}{n}\right)^{n} \;\simeq\; 1 - e^{-1} \;\approx\; 63.2$$

- → Searches on original alignment and alignment replicates are expected to be different to some degree

Facts about the Phylogenetic Bootstrap

- For *n* sites, probability for site to be sampled: 1/*n*
- Probability for site to occur in alignment replicate

$$Pr(X \geq 1) \;=\; 1 - \left(1 - \frac{1}{n}\right)^{n} \;\simeq\; 1 - e^{-1} \;\approx\; 63.2$$

- → Searches on original alignment and alignment replicates are expected to be different to some degree
- Interpretation of the bootstrap still open to discussion
- Hillis: Large simulation experiment to show conservativeness of bootstrap: for support values of ≥ 70%, the probability that the relationship is correct is ≥ 95%.

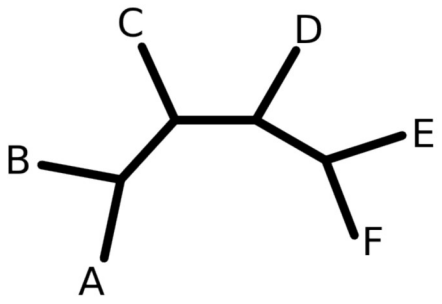
- Further reading:
  http://projecteuclid.org/DPubS?service=UI&version=1.0&verb=Display&handle=euclid.ss/1063994980
  http://sysbio.oxfordjournals.org/content/42/2/182.short

Outline

- Bootstrapping in Phylogenetics
- Uses of the Bootstrap Tree Set
  - **Support For Best-Known Tree**
  - Consensus of Trees
- Distances between Trees
- Extra: Rogue Taxa

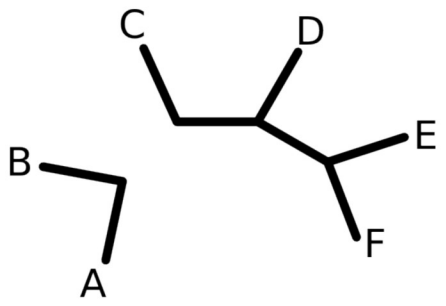
Inner Branches are Bipartitions

- Smallest unit of evolutionary relationship: an inner branch



Inner Branches are Bipartitions

- Smallest unit of evolutionary relationship: an inner branch
- **AB|CDEF**: taxa A,B are more closely related to each other than taxa C,D,E,F

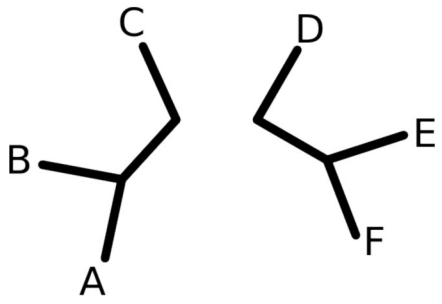


Bipartitions:

- AB|CDEF

Inner Branches are Bipartitions

- Smallest unit of evolutionary relationship: an inner branch
- **AB|CDEF**: taxa A,B are more closely related to each other than taxa C,D,E,F

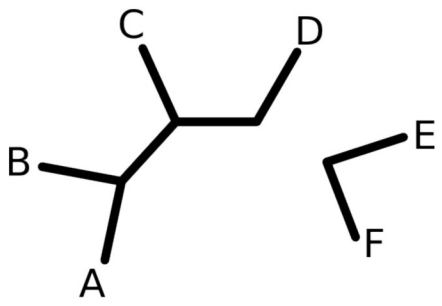


Bipartitions:

- AB|CDEF
- ABC|DEF

Inner Branches are Bipartitions

- Smallest unit of evolutionary relationship: an inner branch
- **AB|CDEF**: taxa A,B are more closely related to each other than taxa C,D,E,F



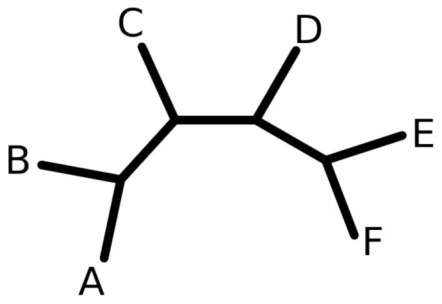
Bipartitions:

- AB|CDEF
- ABC|DEF
- ABCD|EF

Inner Branches are Bipartitions

- Smallest unit of evolutionary relationship: an inner branch
- **AB|CDEF**: taxa A,B are more closely related to each other than taxa C,D,E,F

- outer branches = trivial bipartitions
  - e.g., A|BCDE contained in every tree with taxa ABCDE
  - not informative



Bipartitions:

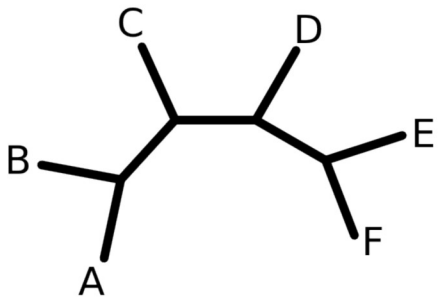
- AB|CDEF
- ABC|DEF
- ABCD|EF

Inner Branches are Bipartitions

- Smallest unit of evolutionary relationship: an inner branch
- **AB|CDEF**: taxa A,B are more closely related to each other than taxa C,D,E,F

- outer branches = trivial bipartitions
  - e.g., A|BCDE contained in every tree with taxa ABCDE
  - not informative

- ⇒ tree ≡ set of bipartitions



Bipartitions:

- AB|CDEF
- ABC|DEF
- ABCD|EF

Draw support on Best-Known Tree

- Algorithm
  - infer ML tree on full alignment

Draw support on Best-Known Tree

- Algorithm
  - infer ML tree on full alignment
  - create a bootstrap tree set

Draw support on Best-Known Tree

- Algorithm
  - infer ML tree on full alignment
  - create a bootstrap tree set
  - extract bipartitions from ML tree and bootstrap tree set

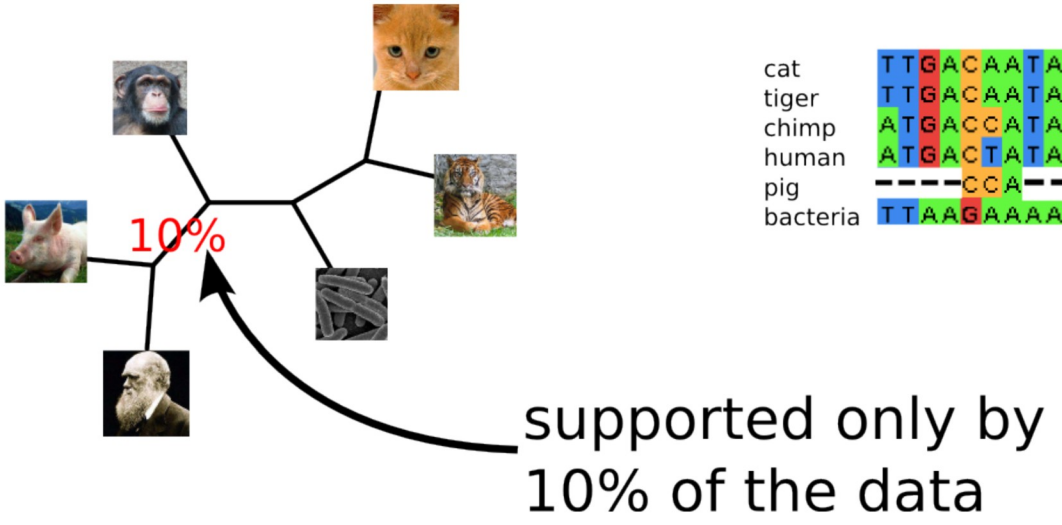
Draw support on Best-Known Tree

- Algorithm
  - infer ML tree on full alignment
  - create a bootstrap tree set
  - extract bipartitions from ML tree and bootstrap tree set
  - annotate ML tree with relative frequency of its bipartitions in the bootstrap tree set

Draw support on Best-Known Tree

- Algorithm
  - infer ML tree on full alignment
  - create a bootstrap tree set
  - extract bipartitions from ML tree and bootstrap tree set
  - annotate ML tree with relative frequency of its bipartitions in the bootstrap tree set

- Result: ML tree with support values

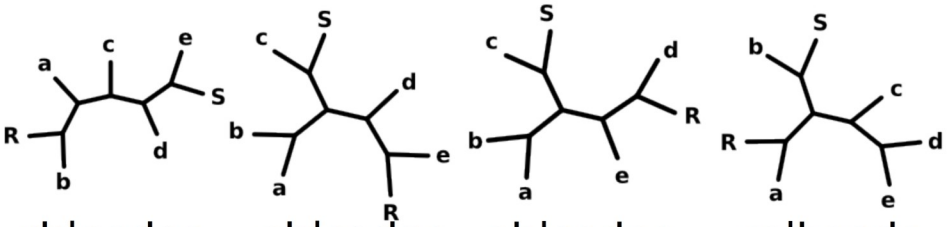


Outline

- Bootstrapping in Phylogenetics
- Uses of the Bootstrap Tree Set
  - Support For Best-Known Tree
  - **Consensus of Trees**
- Distances between Trees
- Extra: Rogue Taxa

Computing the Consensus of Trees

- Algorithm
  (for consensus threshold *t*)
  - Extract bipartitions from tree set



rb|acdes	ab|csder	ab|csder	ra|bscde
rba|cdes	abcs|der	abcs|der	rabs|cde
...	...	...	...

Computing the Consensus of Trees

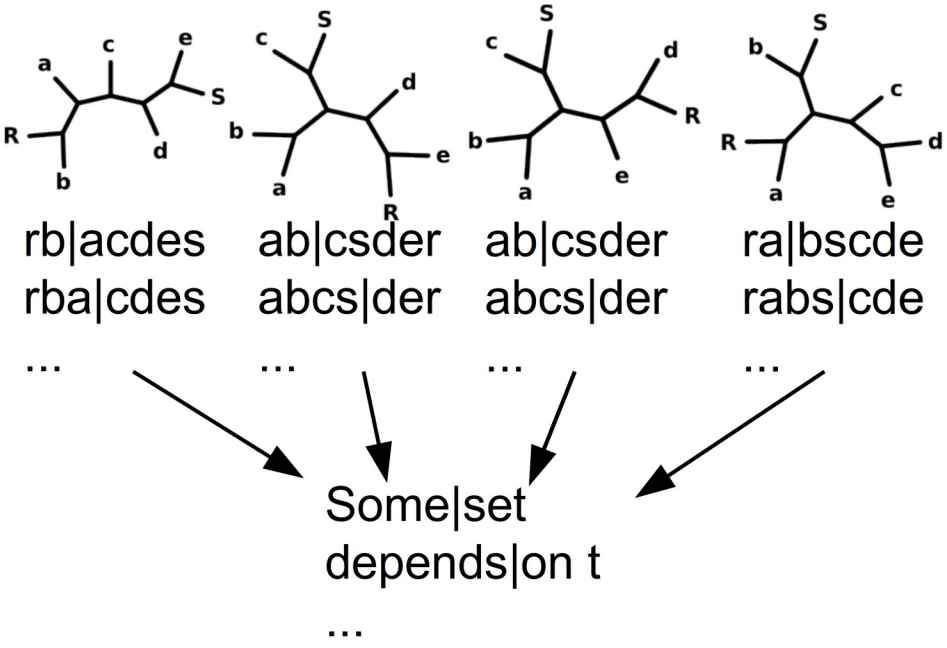
- Algorithm

  (for consensus threshold *t*)

  - Extract bipartitions from tree set

  - Determine consensus bipartitions

    (occur in more than *t*% of the trees)



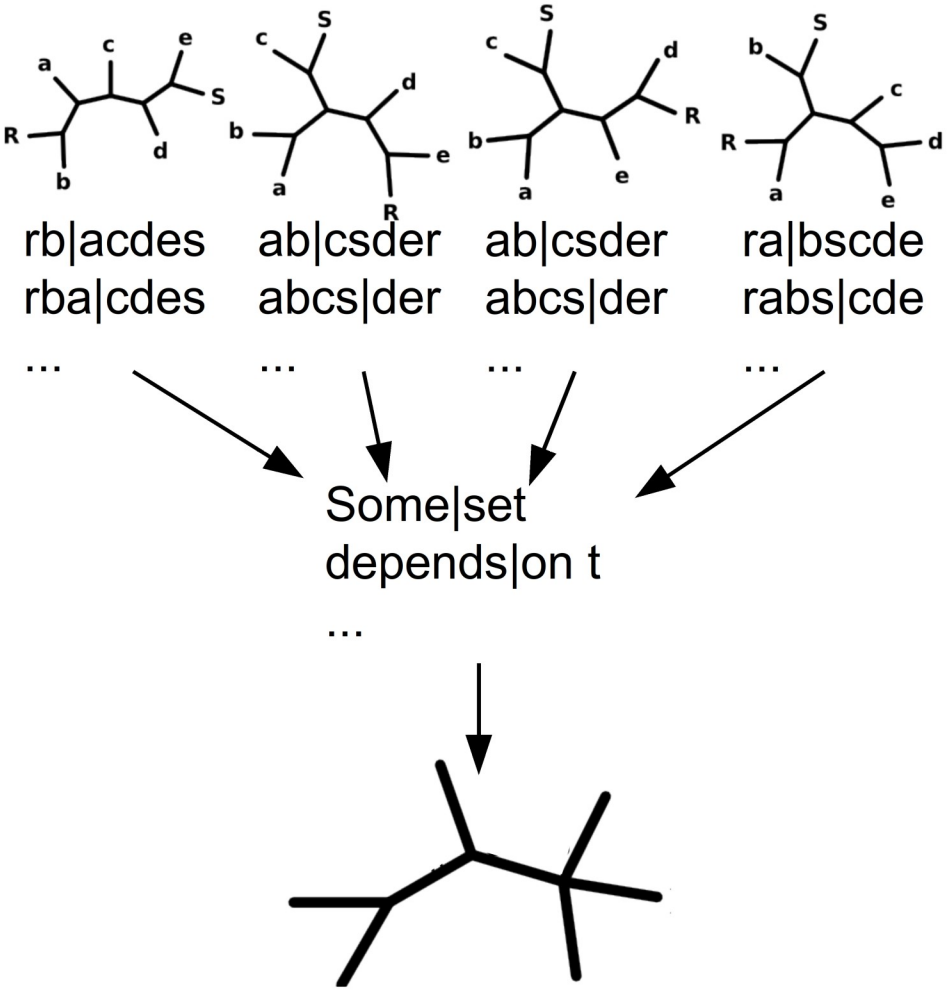
Computing the Consensus of Trees

- Algorithm

  (for consensus threshold *t*)

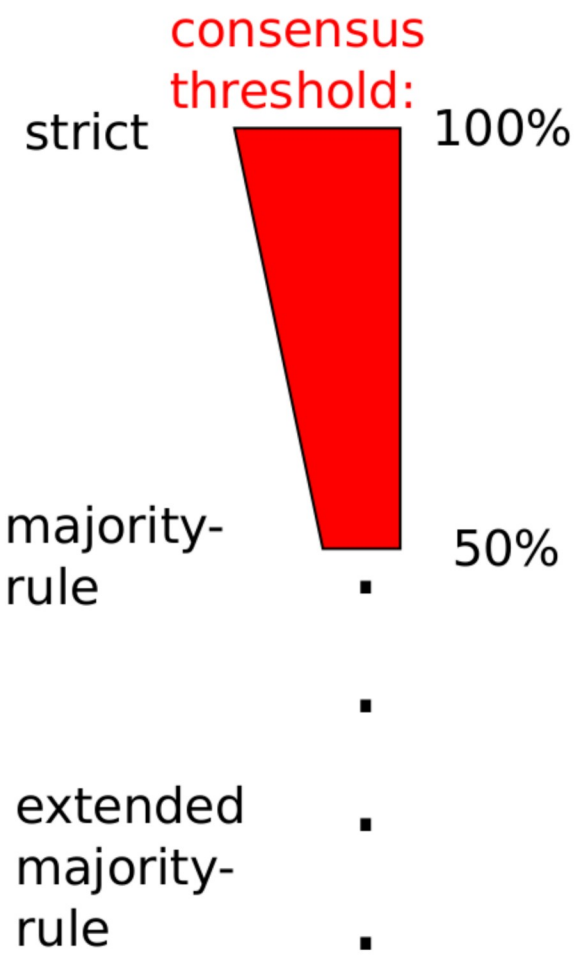
  - Extract bipartitions from tree set
  - Determine consensus bipartitions

    (occur in more than *t*% of the trees)
  - Transform consensus bipartitions into consensus tree



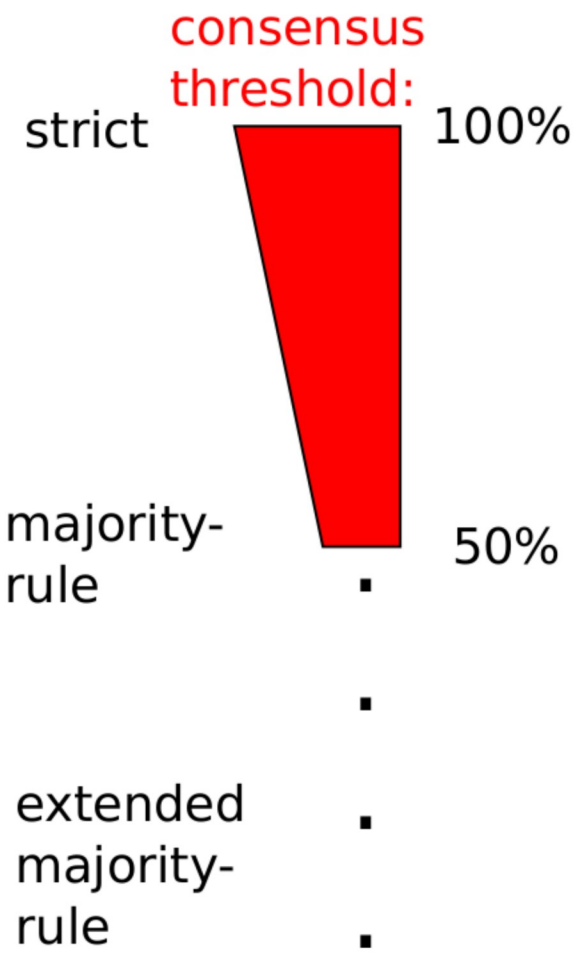
Most popular flavors of Consensi

- e.g., majority-rule consensus tree
  → only bips with freq > 50%
- strict consensus: overly conservative
  (often relaxed to 95%)



Most popular flavors of Consensi

- e.g., majority-rule consensus tree
  → only bips with freq > 50%
- strict consensus: overly conservative
  (often relaxed to 95%)
- threshold of < 50% not possible
  (see next slides)
- instead: extended majority rule consensus



Compatibility of Bipartitions

- the following bipartitions b_1 and b_2 can never be part of the same tree:
  - b_1 = ABC|DEF
  - b_2 = AD|BCEF
- We call b_1 and b_2 incompatible

Compatibility of Bipartitions

- the following bipartitions b_1 and b_2 can never be part of the same tree:
  - b_1 = ABC|DEF
  - b_2 = AD|BCEF
- We call b_1 and b_2 incompatible
- Definition

  Two bipartitions $b_1 = B|\overline{B}$ and $b_2 = C|\overline{C}$ are compatible, if

  $$(B \cap C = \emptyset) \vee (B \cap \overline{C} = \emptyset) \vee (\overline{B} \cap C = \emptyset)$$

Compatibility of Bipartitions

- the following bipartitions b_1 and b_2 can never be part of the same tree:
  - b_1 = ABC|DEF
  - b_2 = AD|BCEF
- We call b_1 and b_2 incompatible
- Definition

  Two bipartitions $b_1 = B|\overline{B}$ and $b_2 = C|\overline{C}$ are compatible, if

  $$(B \cap C = \emptyset) \vee (B \cap \overline{C} = \emptyset) \vee (\overline{B} \cap C = \emptyset)$$

- In our example

  $ABC \cap AD = A$ $\quad$ $ABC \cap BCEF = BC$ $\quad$ $AD \cap DEF = D$

The Extended Majority-Rule Consensus

- Each pair of bipartitions that occur in more than 50% of the trees is compatible

The Extended Majority-Rule Consensus

- Each pair of bipartitions that occur in more than 50% of the trees is compatible

  Proof: Given two bipartitions that both occur in more than 50% of the trees, there must be at least one tree that contains both of them. Thus, they must be compatible.

- Given the majority-rule consensus, we can refine this tree with non-consensus bipartitions

- Problem is NP-hard (however, a greedy approach is applicable)
  http://www.sciencedirect.com/science/article/pii/S0166218X96000716

Extended majority-rule consensus algorithm

- Given consensus bips B_c and non-consensus bips B_n

Extended majority-rule consensus algorithm

- Given consensus bips B_c and non-consensus bips B_n
- Remove most frequent bip $b \in B_n$ from B_n

Extended majority-rule consensus algorithm

- Given consensus bips B_c and non-consensus bips B_n
- Remove most frequent bip $b \in B_n$ from B_n
- If b is compatible to all bips $\in B_c$:

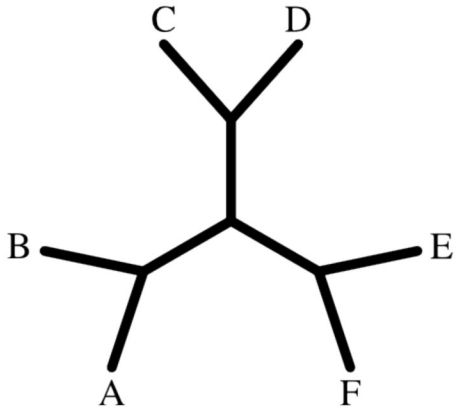
  $\rightarrow B_c = B_c \cup \{b\}$

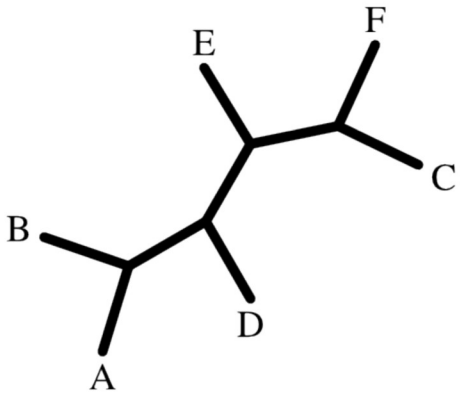
Extended majority-rule consensus algorithm

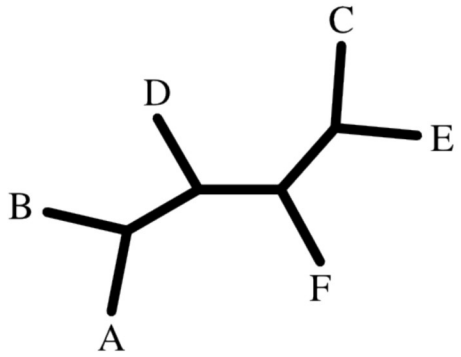
- Given consensus bips B_c and non-consensus bips B_n
- Remove most frequent bip $b \in B_n$ from B_n
- If b is compatible to all bips $\in B_c$:

  $\rightarrow B_c = B_c \cup \{b\}$
- Loop until $(|B_c| = (n - 3)) \quad \vee \quad (B_n = \emptyset)$

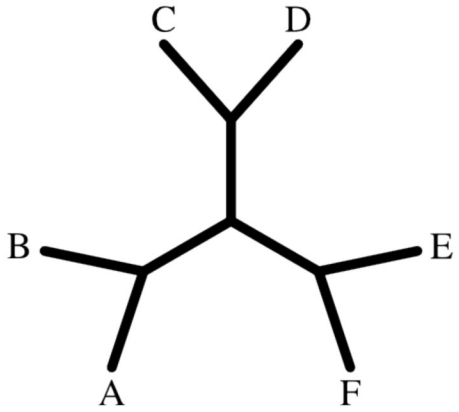
Exercise:
Compute strict and MR Consensus Tree and BS

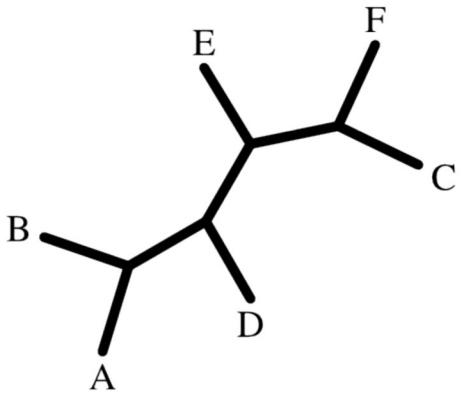


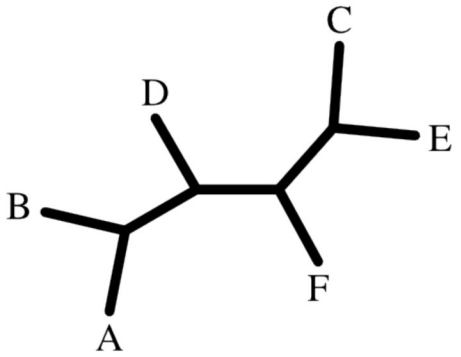




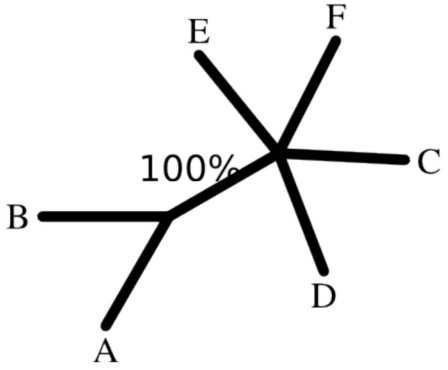
Exercise:
Compute strict and MR Consensus Tree and BS



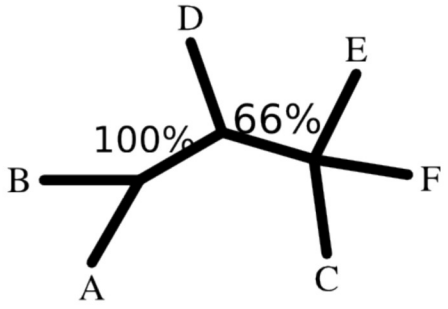




strict:



majority-rule:



Data structure: Bipartitions

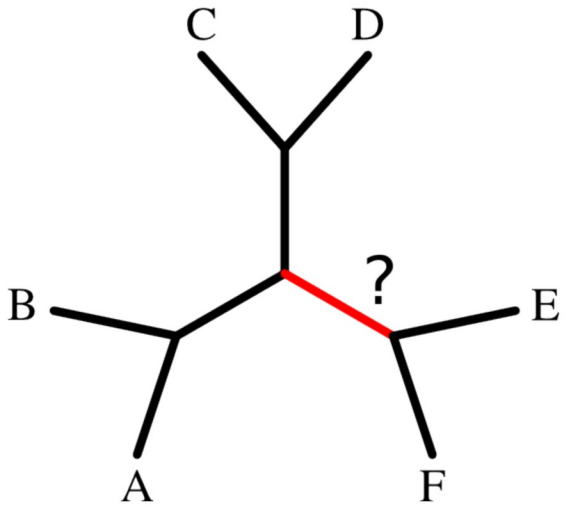
- Bit Vectors
  - Ideal for sets with pre-defined number of `n` elements
  - space requirements: `θ(n/8)`
  - e.g., `std::vector<bool>` or `uint32_t[]`
  - copying extremely efficient: `memcpy`

Data structure: Bipartitions

- Bit Vectors
  - Ideal for sets with pre-defined number of `n` elements
  - space requirements: $\Theta(n/8)$
  - e.g., `std::vector<bool>` or `uint32_t[]`
  - copying extremely efficient: `memcpy`
- Set Operations
  - `a ∪ b → c[i] = a[i] | b[i]`
  - `a ∩ b → c[i] = a[i] & b[i]`
  - Time in $\Theta(n/64)$ with `uint64_t[]`

Implementation: Efficient Extraction and Hashing

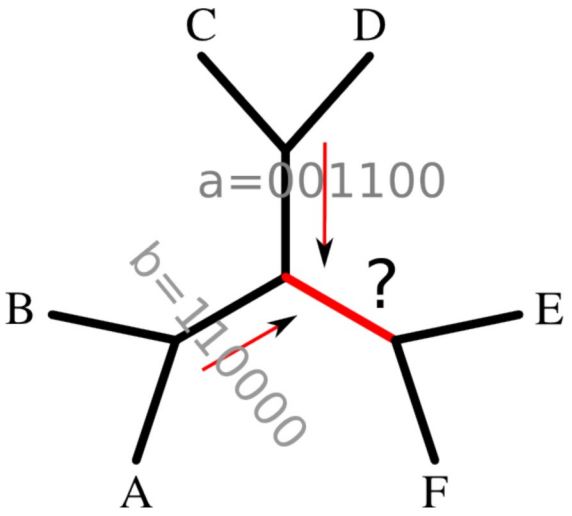
- Post-order traversal
- Store bipartitions in hashtable



Implementation: Efficient Extraction and Hashing

- Post-order traversal
- Store bipartitions in hashtable

- Quickly compute bipartition from adjacent bipartitions



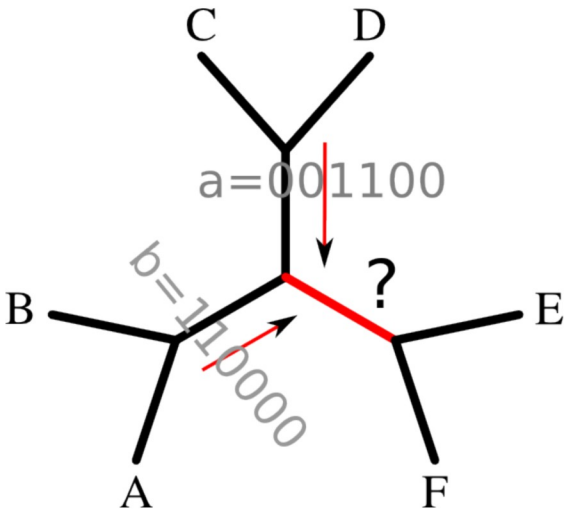

```
a = 001100 = CD|ABEF
b = 110000 = AB|CDEF
? = a | b = 111100 = ABCD|EF
```

Implementation: Efficient Extraction and Hashing

- Post-order traversal
- Store bipartitions in hashtable

- Quickly compute bipartition from adjacent bipartitions

- `r[]`: one random number per taxon
- hash: `xor` value of all corresponding random numbers in array



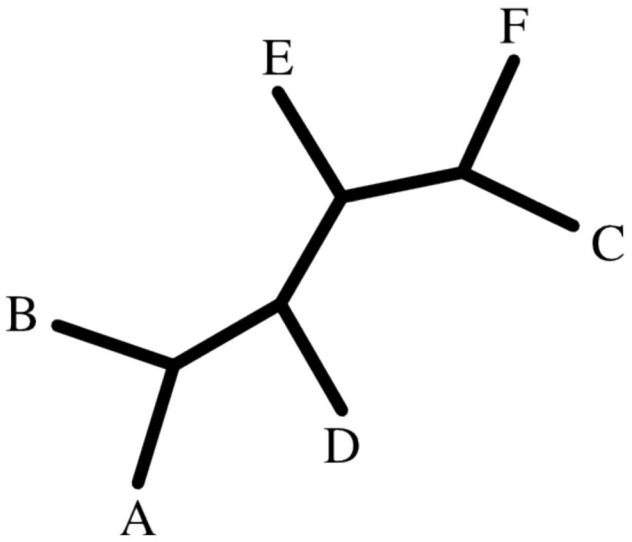

```
a = 001100 = CD|ABEF
b = 110000 = AB|CDEF
? = a | b = 111100 = ABCD|EF
```

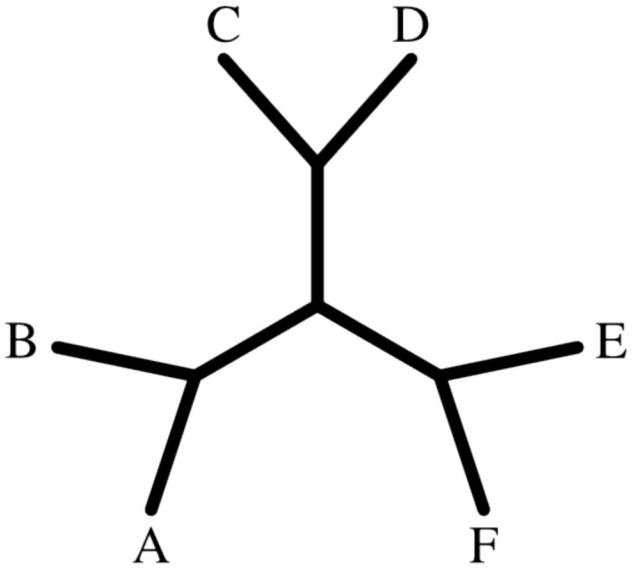
```
hash(a) = r[C] xor r[D]
hash(b) = r[A] xor r[B]
hash(?) = hash(a) xor hash(b)
```

Outline

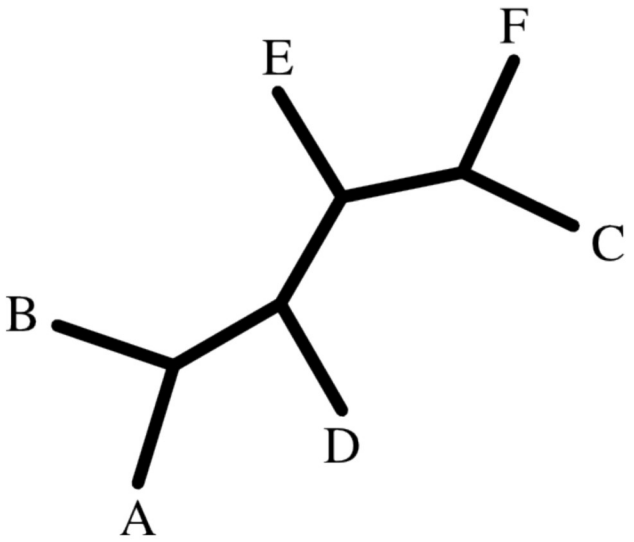
- Bootstrapping in Phylogenetics
- Uses of the Bootstrap Tree Set
  - Support For Best-Known Tree
  - Consensus of Trees
- **Distances between Trees**
- Extra: Rogue Taxa

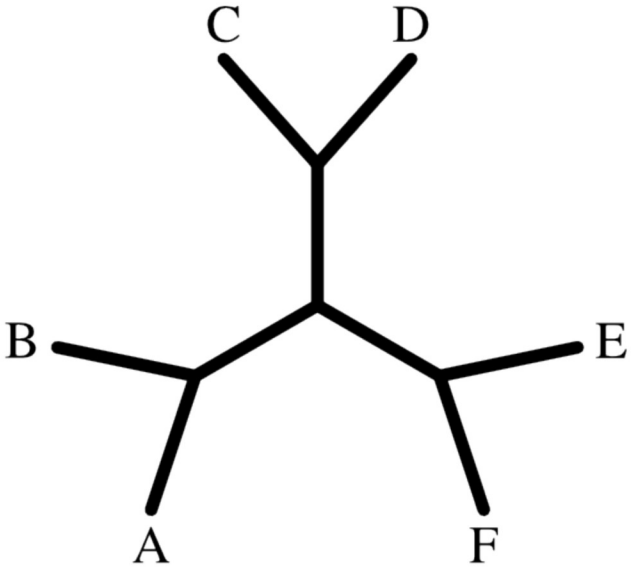
How *similar* are these Trees?





How *similar* are these Trees?





Bipartitions again!

AB|CDEF
ABD|CEF
ABDE|FC

AB|CDEF
ABCD|EF
CD|ABEF

The Robinson-Foulds Distance

- Definition

Given the unrooted trees T_1 and T_2 with bipartitions B_1 and B_2, the Robinson-Foulds (RF) distance is defined as:

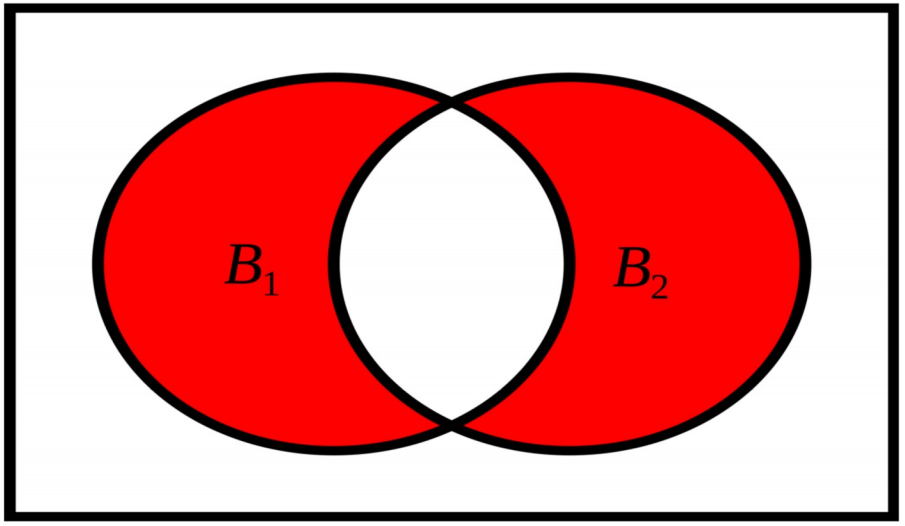
$$RF(T_1, T_2) = |B_1 \cup B_2| - |B_1 \cap B_2|$$

The Robinson-Foulds Distance

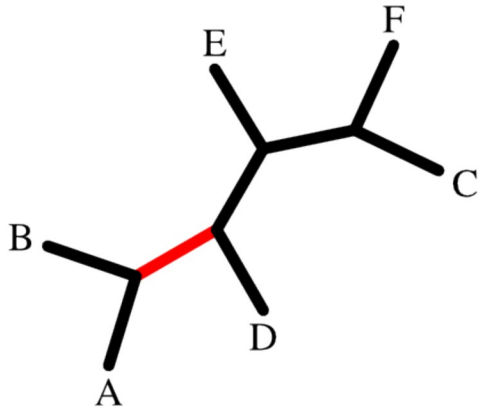
- Definition

Given the unrooted trees T_1 and T_2 with bipartitions B_1 and B_2, the Robinson-Foulds (RF) distance is defined as:

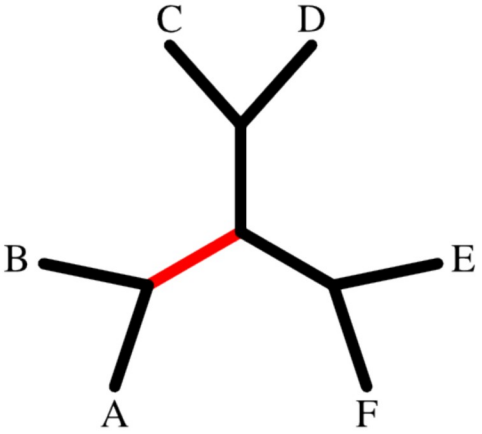
$$RF(T_1, T_2) = |B_1 \cup B_2| - |B_1 \cap B_2|$$



How distant are these Trees?

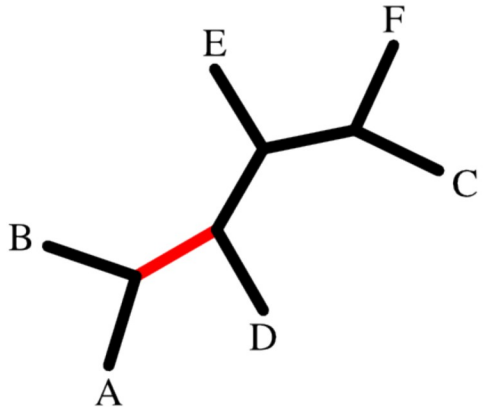


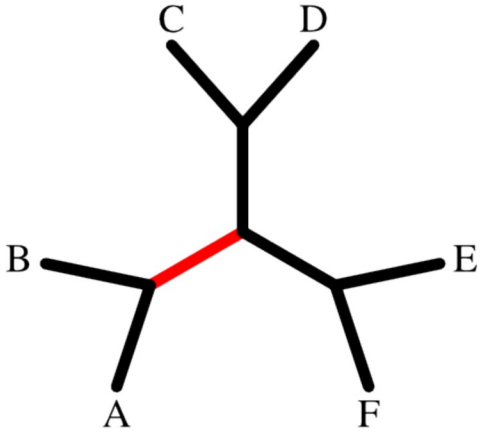
AB|CDEF
ABD|CEF
ABDE|FC



AB|CDEF
ABCD|EF
CD|ABEF

How distant are these Trees?





AB|CDEF
ABD|CEF
ABDE|FC

AB|CDEF
ABCD|EF
CD|ABEF

→ RF-distance = 4

Flavors of the RF-distance

- Let $\triangle$ denote the symmetric set difference:

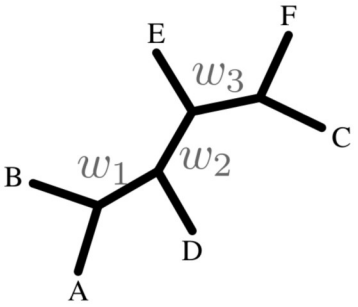
$$X \triangle Y = (X \setminus Y) \cup (Y \setminus X)$$

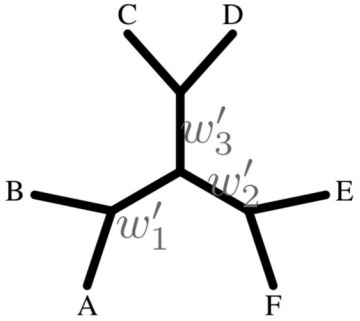
Flavors of the RF-distance

- Let $\triangle$ denote the symmetric set difference:

  $X \triangle Y = (X \setminus Y) \cup (Y \setminus X)$

- For unrooted trees T_1 and T_2 with bipartitions B_1 and B_2, let *w(i)* map the support value in the respective tree.



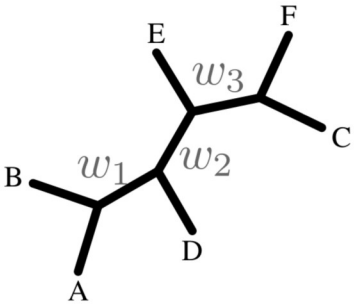


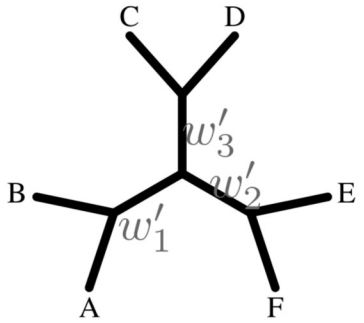
Flavors of the RF-distance

- Let Δ denote the symmetric set difference:

  $X \Delta Y = (X \setminus Y) \cup (Y \setminus X)$

- For unrooted trees T_1 and T_2 with bipartitions B_1 and B_2, let *w(i)* map the support value in the respective tree.





	unweighted	weighted
absolute	$\left(\lvert B_1 \cup B_2 \rvert - \lvert B_1 \cap B_2 \rvert\right)$	$\sum_{b \in B_1 \Delta B_2} w(b)$
relative	$\frac{\lvert B_1 \cup B_2 \rvert - \lvert B_1 \cap B_2 \rvert}{2(n-3)}$	$\frac{\sum_{b \in B_1 \Delta B_2} w(b)}{2(n-3)}$

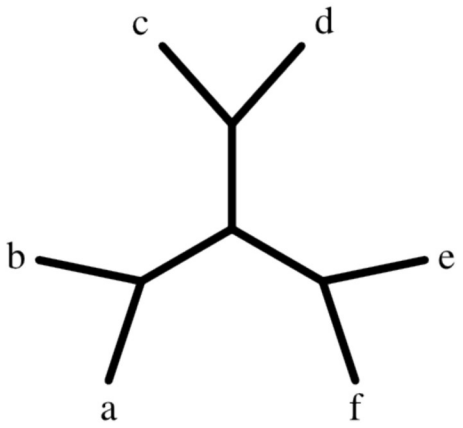
The RF-Distance is a Metric

- A distance $d : X \times X \rightarrow R$
  is called a metric, if $\forall\, x, y, z \in X$:

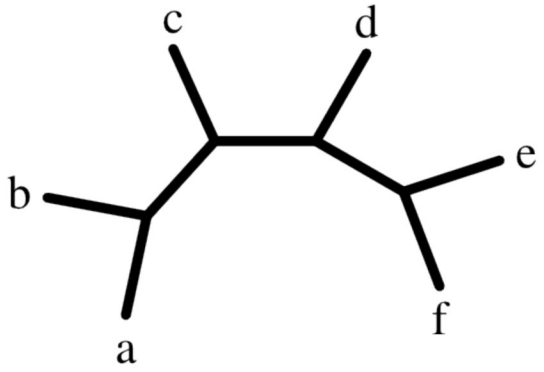
1. $d(x, y) \geq 0$ (separation axiom)
2. $d(x, y) = 0 \Leftrightarrow x = y$ (coincidence axiom)
3. $d(x, y) = d(y , x)$ (symmetry)
4. $d(x, z) \leq d(x, y) + d(y , z)$ (triangle inequality)

Exercise:
Compute the relative and absolute RF-distance

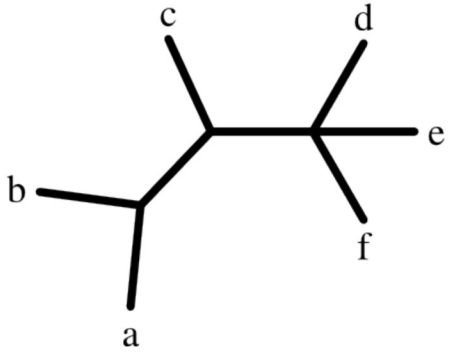
Tree 1



Tree 2



Tree 3

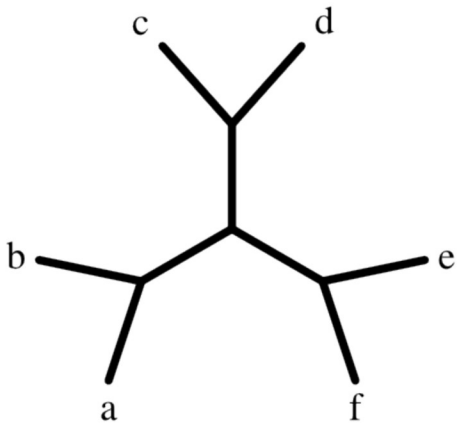


$$RF(T_1, T_2) = |B_1 \cup B_2| - |B_1 \cap B_2|$$

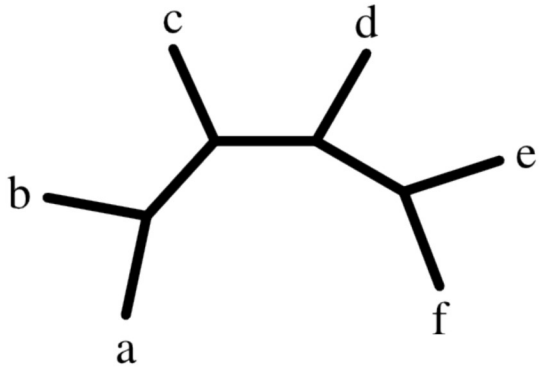
$$rRF(T_1, T_2) = \frac{RF}{2(n-3)}$$

Exercise:
Compute the relative and absolute RF-distance

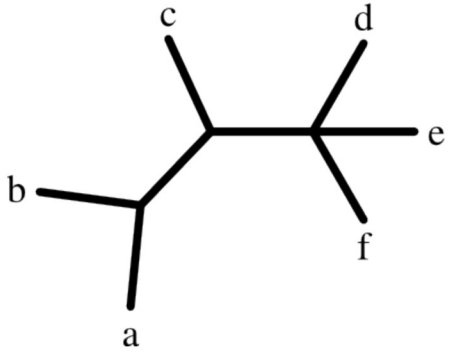
Tree 1



Tree 2



Tree 3



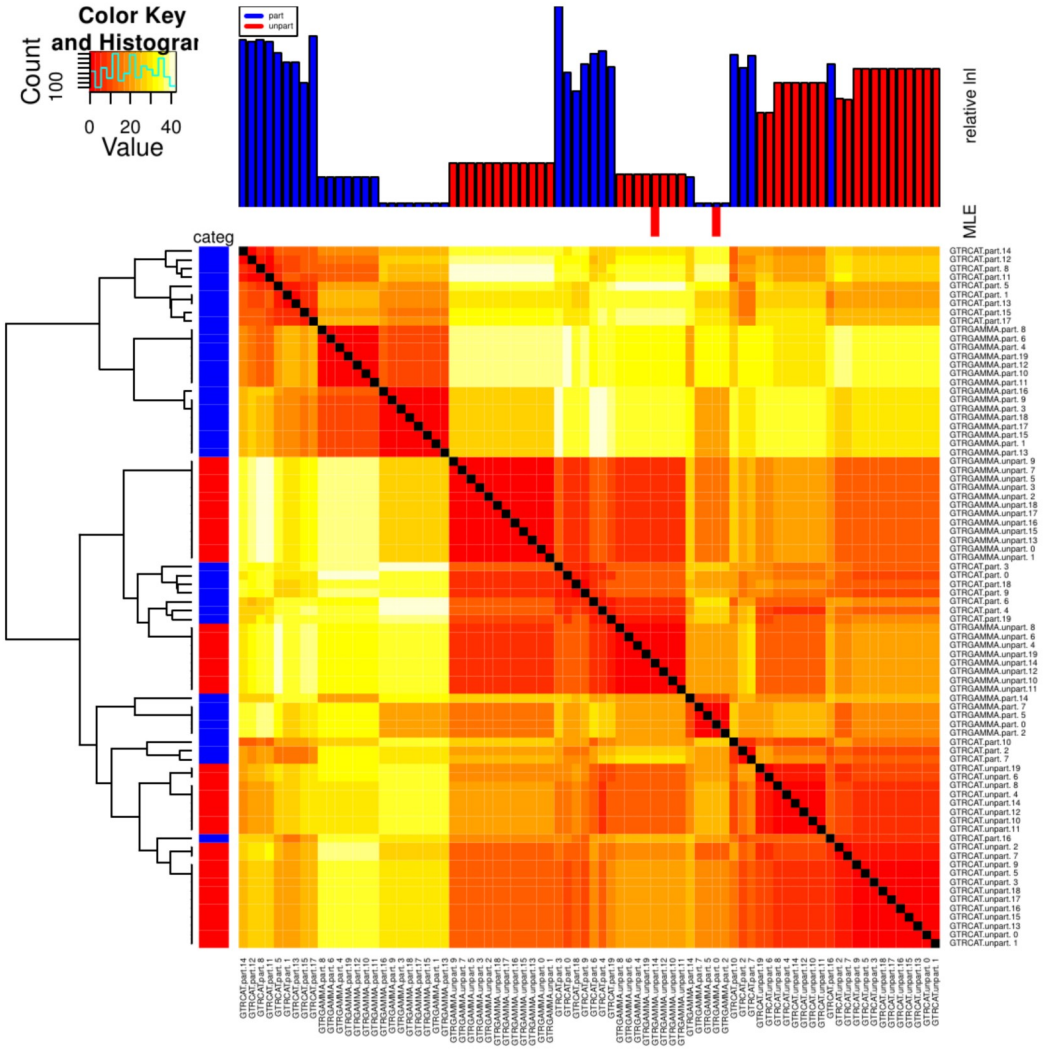
Solution:

RF(T1, T2) = 2 rRF(T1,T2) ~ 0.33

RF(T1, T3) = 3 rRF(T1,T3) ~ 0.5

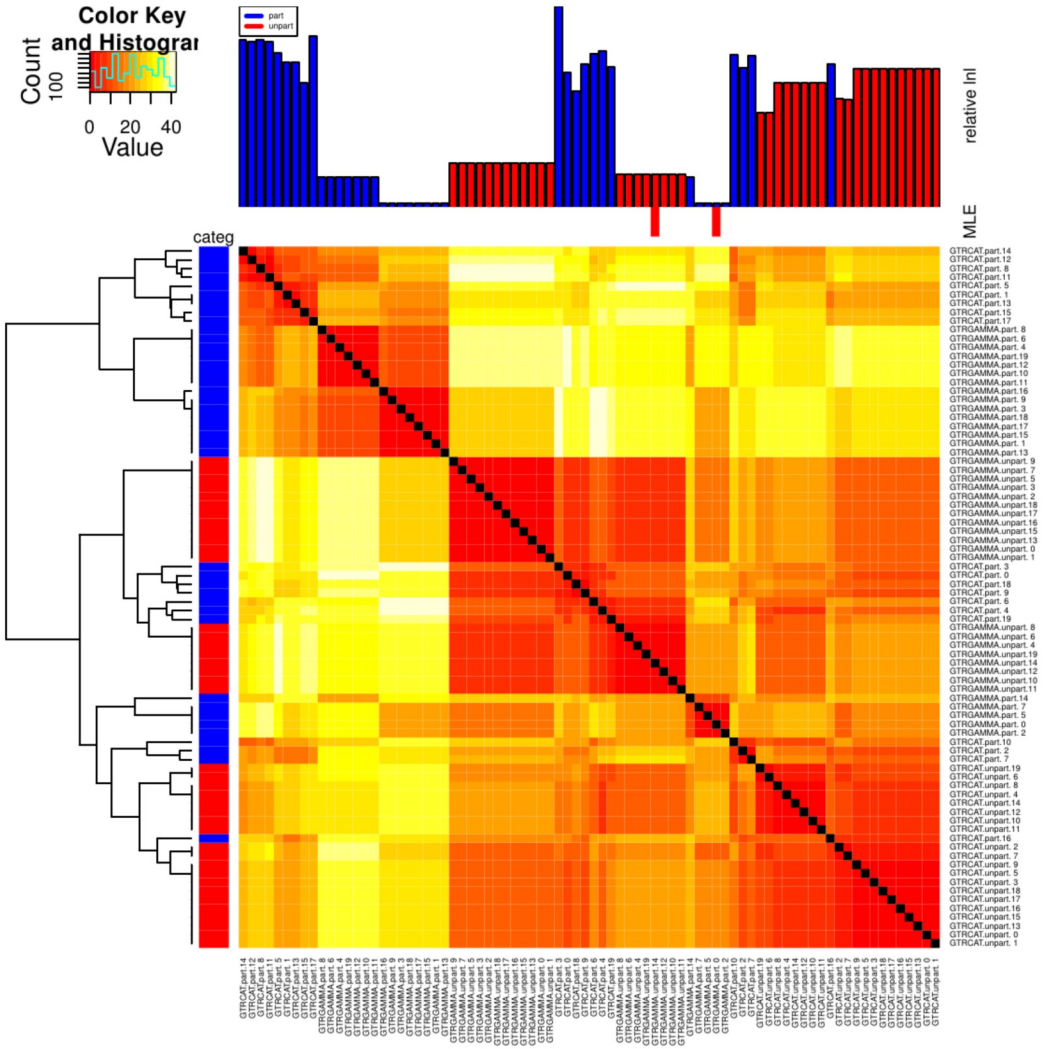
RF(T2, T3) = 1 rRF(T2,T3) ~ 0.17

RF-Distances in Practice



- 40 ML tree searches (same dataset, different models)
- cluster trees by RF-distance

RF-Distances in Practice



- 40 ML tree searches (same dataset, different models)
- cluster trees by RF-distance
- important: likelihood not comparable among different models/datasets
- ⇒ visualize tree/likelihood landscape

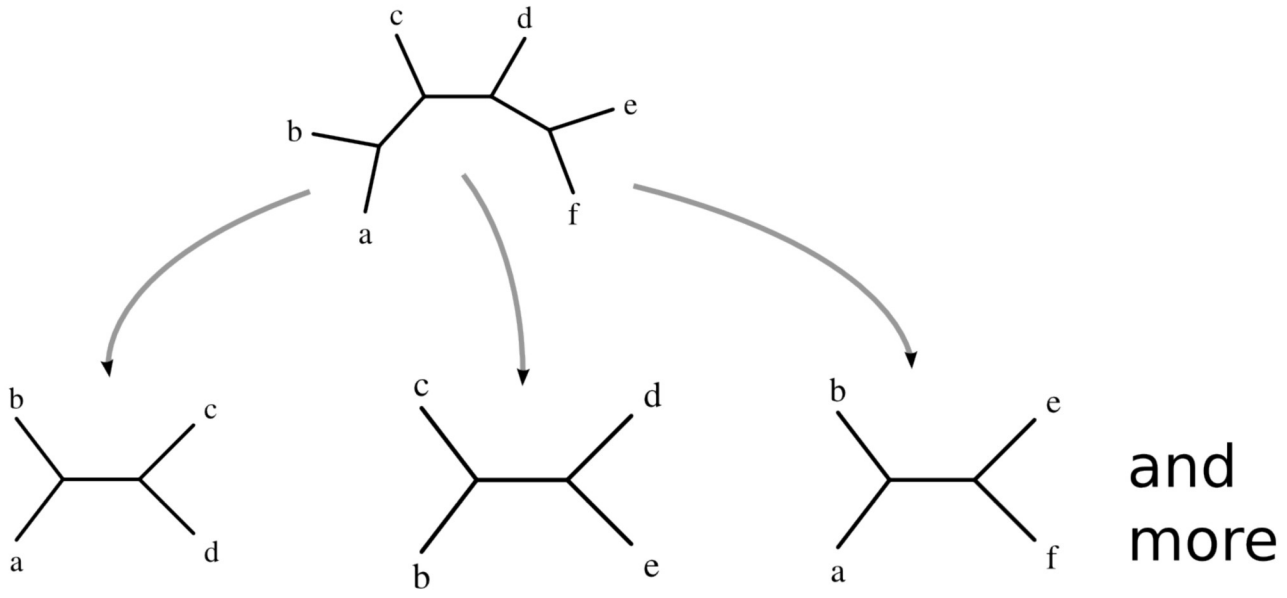
Further reading: http://www.sciencemag.org/content/346/6215/1320.abstract

Alternative: Triplet/Quartet-based Distance

- Previously: bipartition ⇒ unit of phylogenetic relationship
- Now instead: triplets for rooted, quartets for unrooted trees

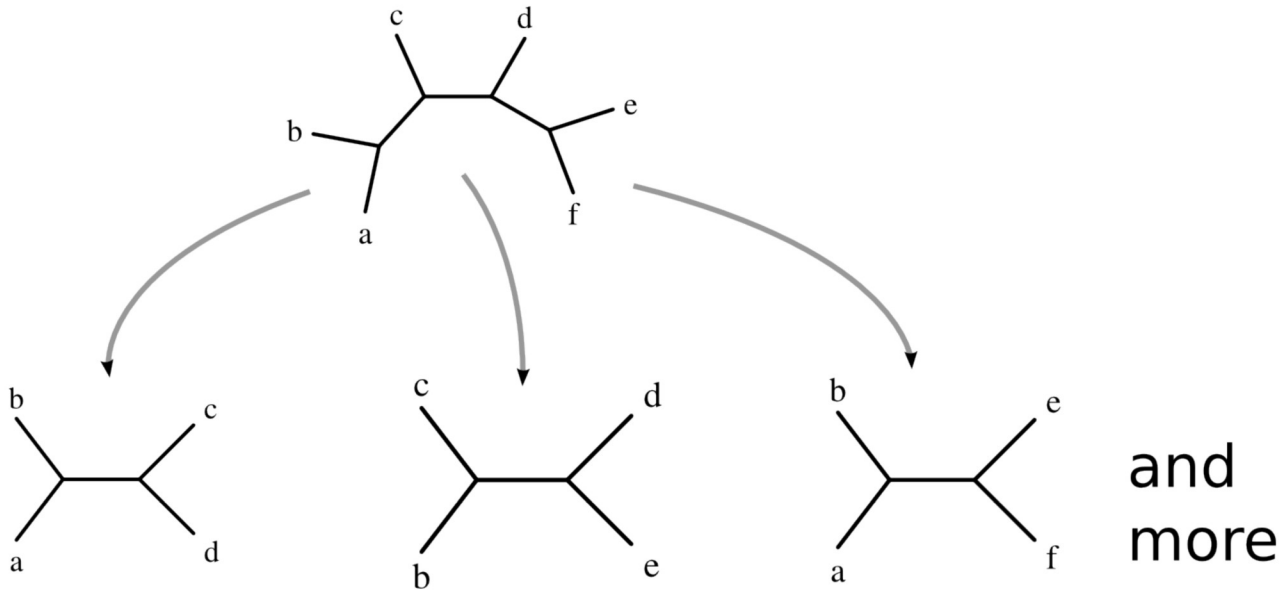
Alternative: Triplet/Quartet-based Distance

- Previously: bipartition ⇒ unit of phylogenetic relationship
- Now instead: triplets for rooted, quartets for unrooted trees



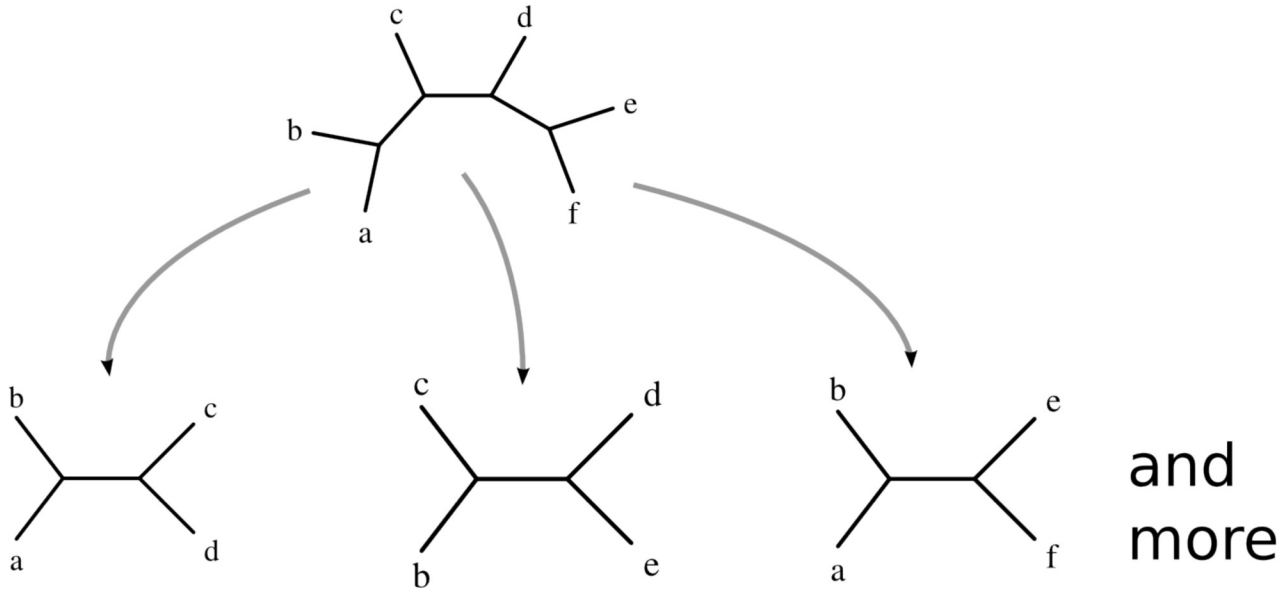
Alternative: Triplet/Quartet-based Distance

- Previously: bipartition ⇒ unit of phylogenetic relationship
- Now instead: triplets for rooted, quartets for unrooted trees
- Feature: for each combination of 4 taxa: only 3 possible trees



Alternative: Triplet/Quartet-based Distance

- Previously: bipartition ⇒ unit of phylogenetic relationship
- Now instead: triplets for rooted, quartets for unrooted trees
- Feature: for each combination of 4 taxa: only 3 possible trees
- Naïve algorithm: $O(n^4)$ to extract all quartets



Outline

- Bootstrapping in Phylogenetics
- Uses of the Bootstrap Tree Set
  - Support For Best-Known Tree
  - Consensus of Trees
- **Distances between Trees**
- Extra: Rogue Taxa

Bootstopping: How many replicates should we compute?

- Algorithm
  1. Randomly split tree set T in sub-sets t' and t''
  2. Compute consensus trees c(t') and c(t'')
  3. Compute weighted RF-distance wRF(c(t'), c(t''))
  4. Repeat steps (1-3), with cut-off values l and m:
     - If less than l permutations have a wRF < m
       ⇒ support values stable, stop!
     - Else: more replicates needed! Continue bootstrap.
- Further reading:
  http://online.liebertpub.com/doi/abs/10.1089/cmb.2009.0179

Outline

- Bootstrapping in Phylogenetics
- Uses of the Bootstrap Tree Set
  - Support For Best-Known Tree
  - Consensus of Trees
- Distances between Trees
- **Extra: Rogue Taxa**

Definition: Rogue taxa

- They assume various positions in a bootstrap tree set
- or change their position for
  - different model parameters
  - or datasets

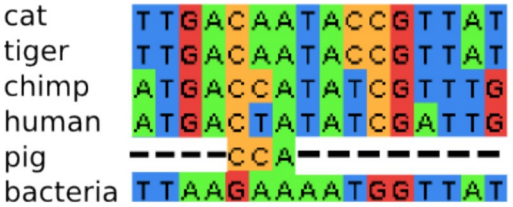
Definition: Rogue taxa

- They assume various positions in a bootstrap tree set
- or change their position for
  - different model parameters
  - or datasets

- This is not general phylogenetic instability:
  In most rogue scenarios, surrounding topology is (relatively) stable

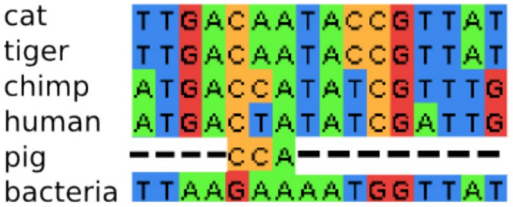
- But: Rogues may influence alignment / phylogenetic inference process

Biological Reasons For Rogues

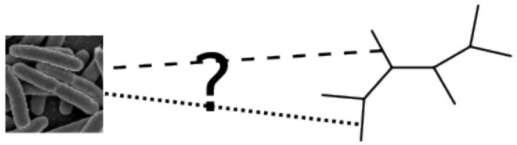


missing data /
gappy alignments

Biological Reasons For Rogues

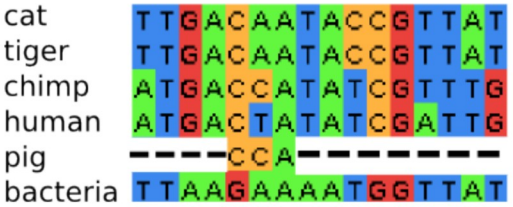


missing data /
gappy alignments

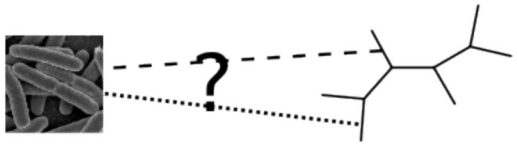


excessively long
branch lengths

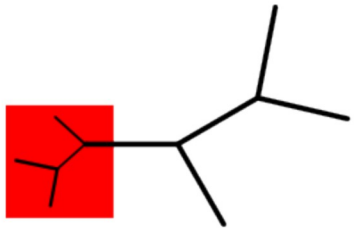
Biological Reasons For Rogues



missing data /
gappy alignments

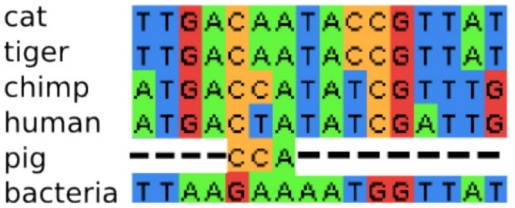


excessively long
branch lengths

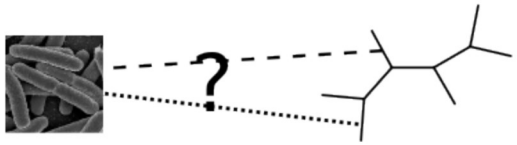


low mutation rates

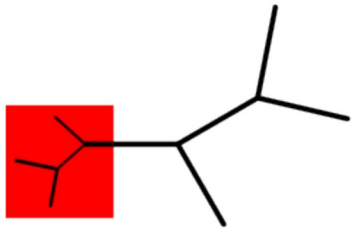
Biological Reasons For Rogues



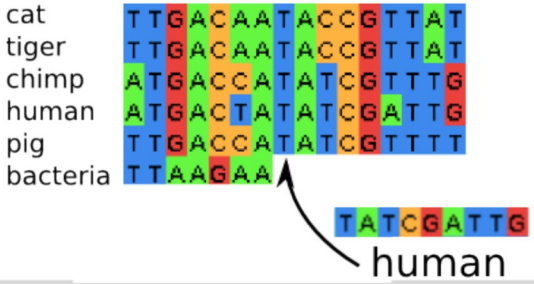
missing data /
gappy alignments



excessively long
branch lengths

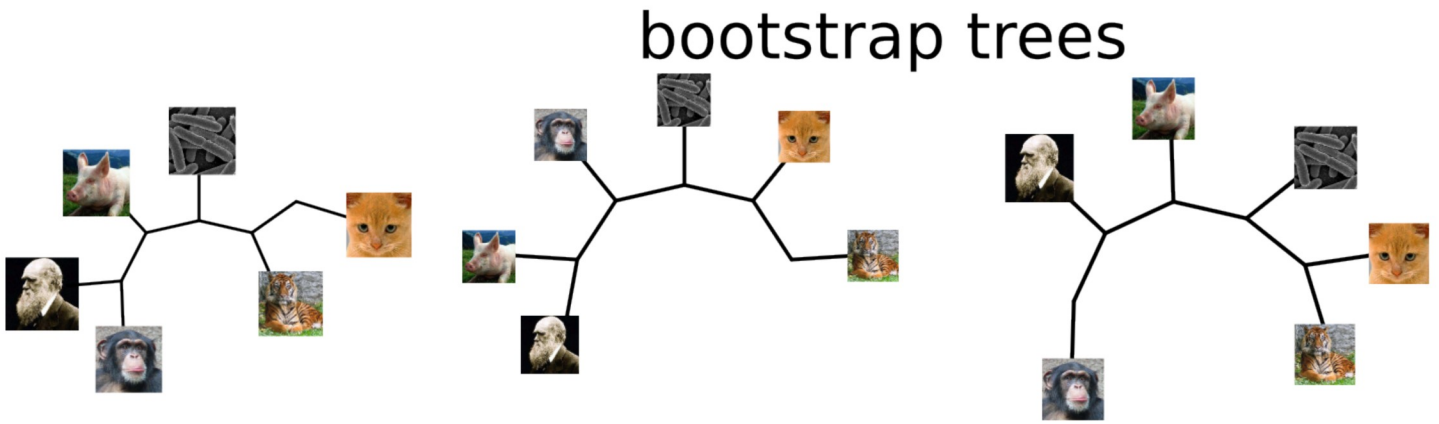


low mutation rates

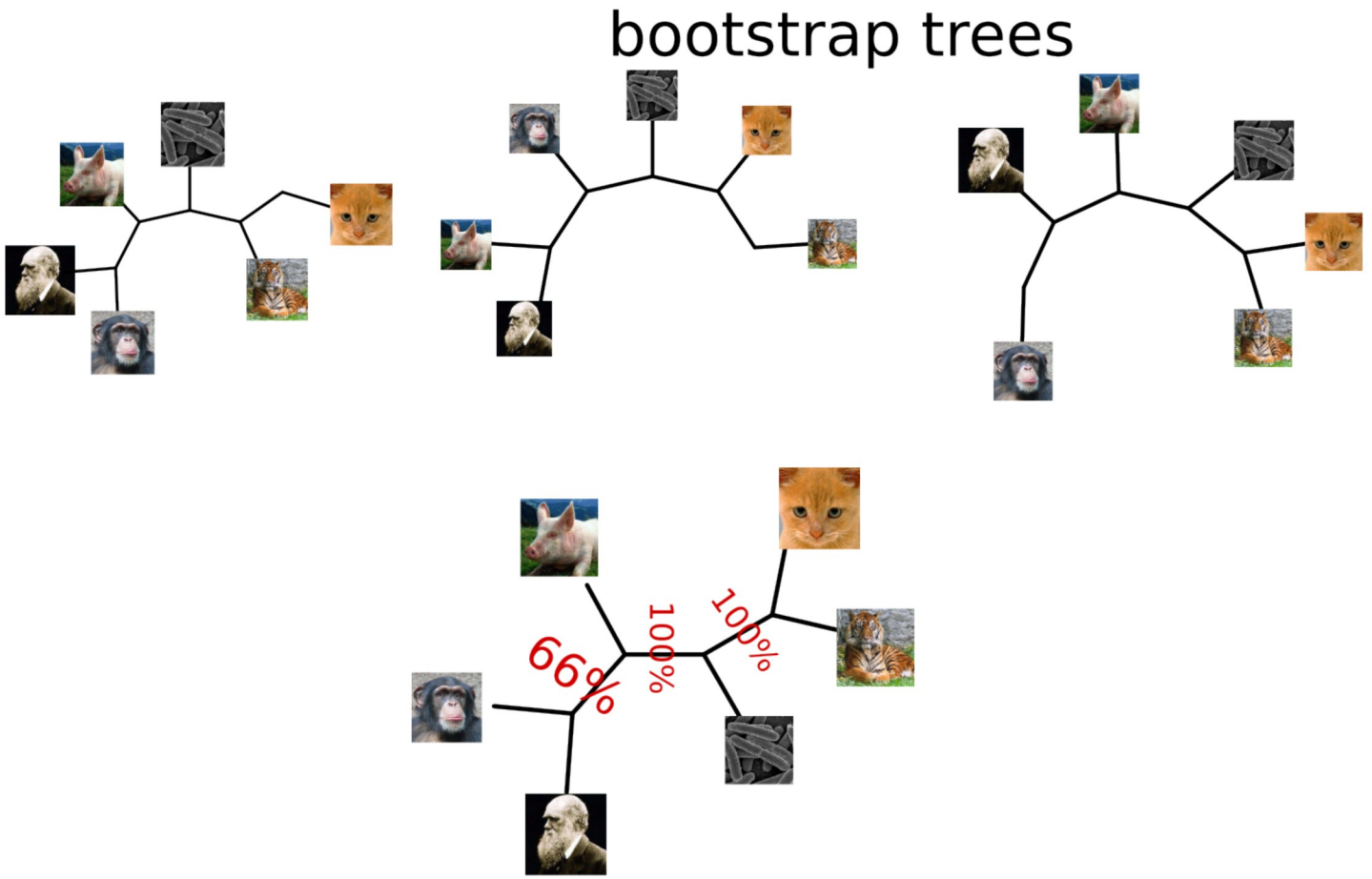


chimeric sequences /
ambiguous signal

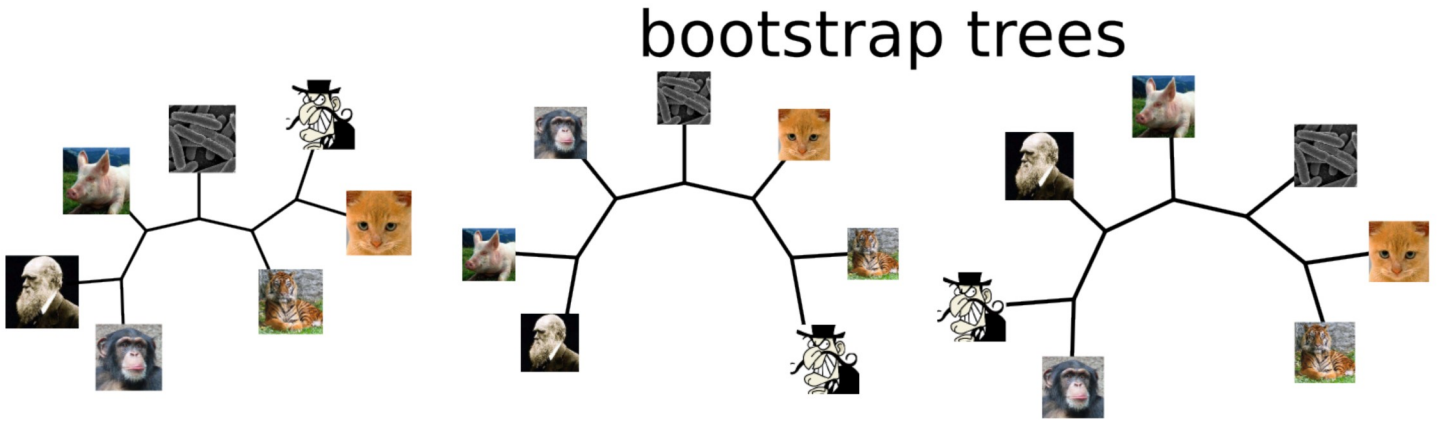
Effect On Consensus Trees



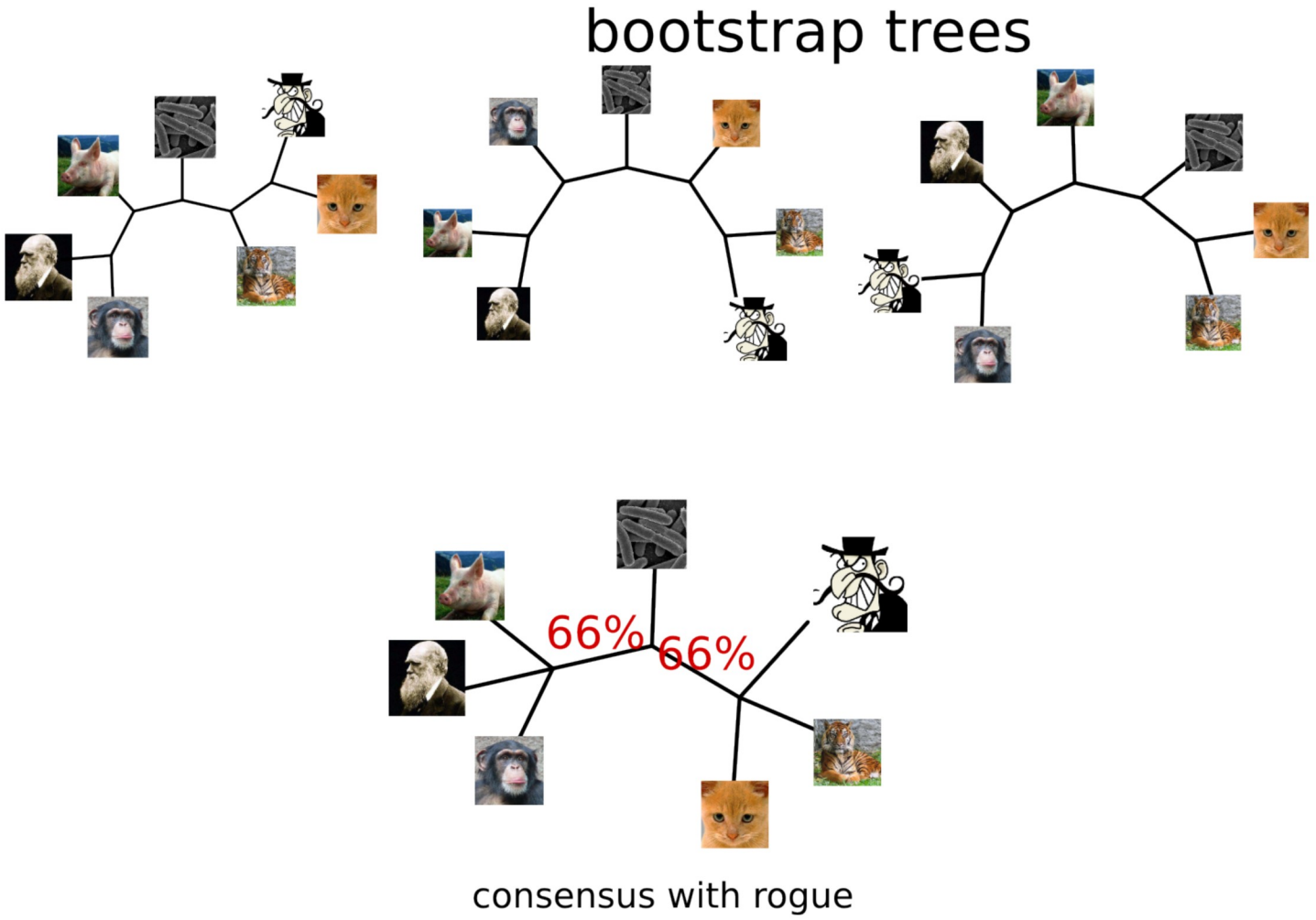
Effect On Consensus Trees



Effect On Consensus Trees

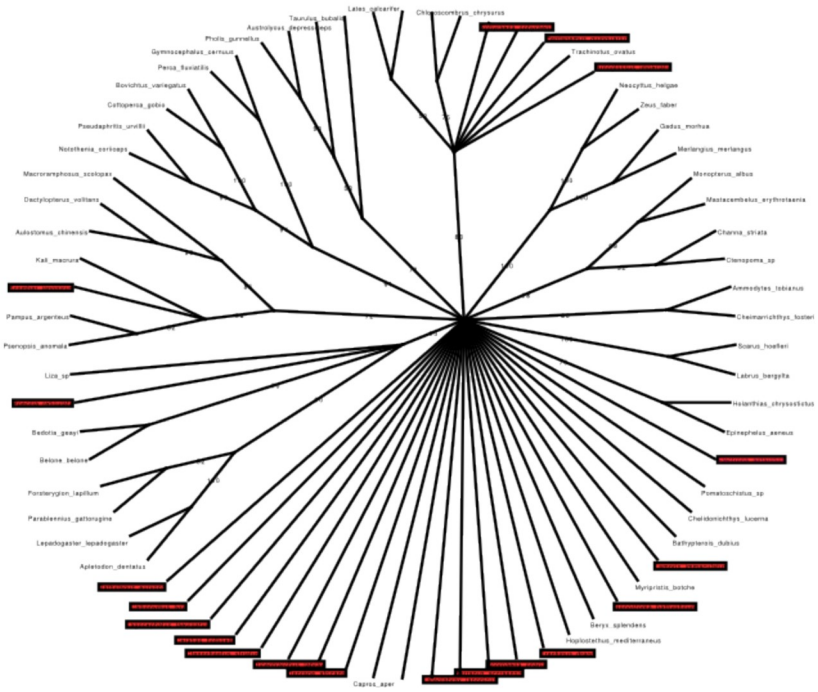


Effect On Consensus Trees



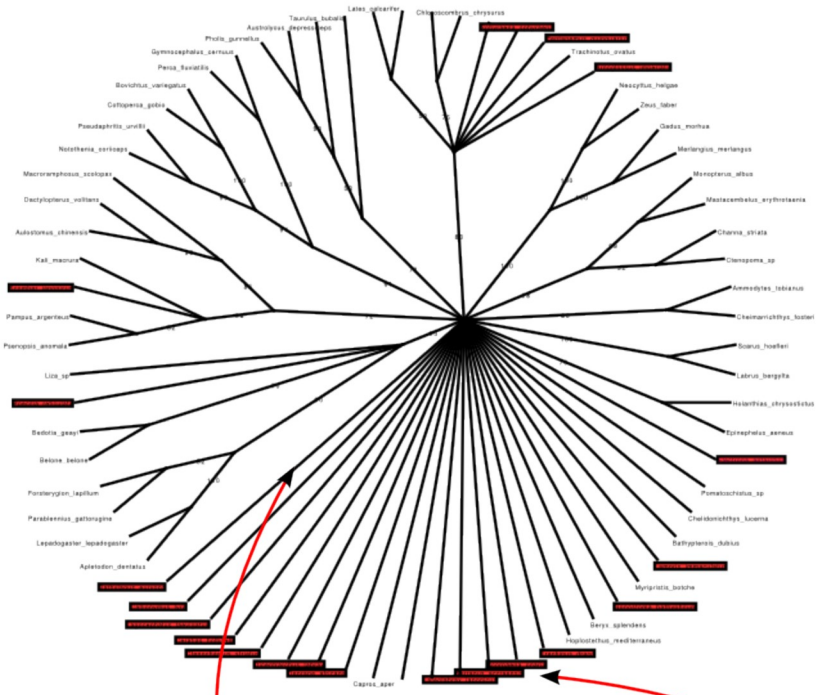
Real-World Example

72 taxa



Real-World Example

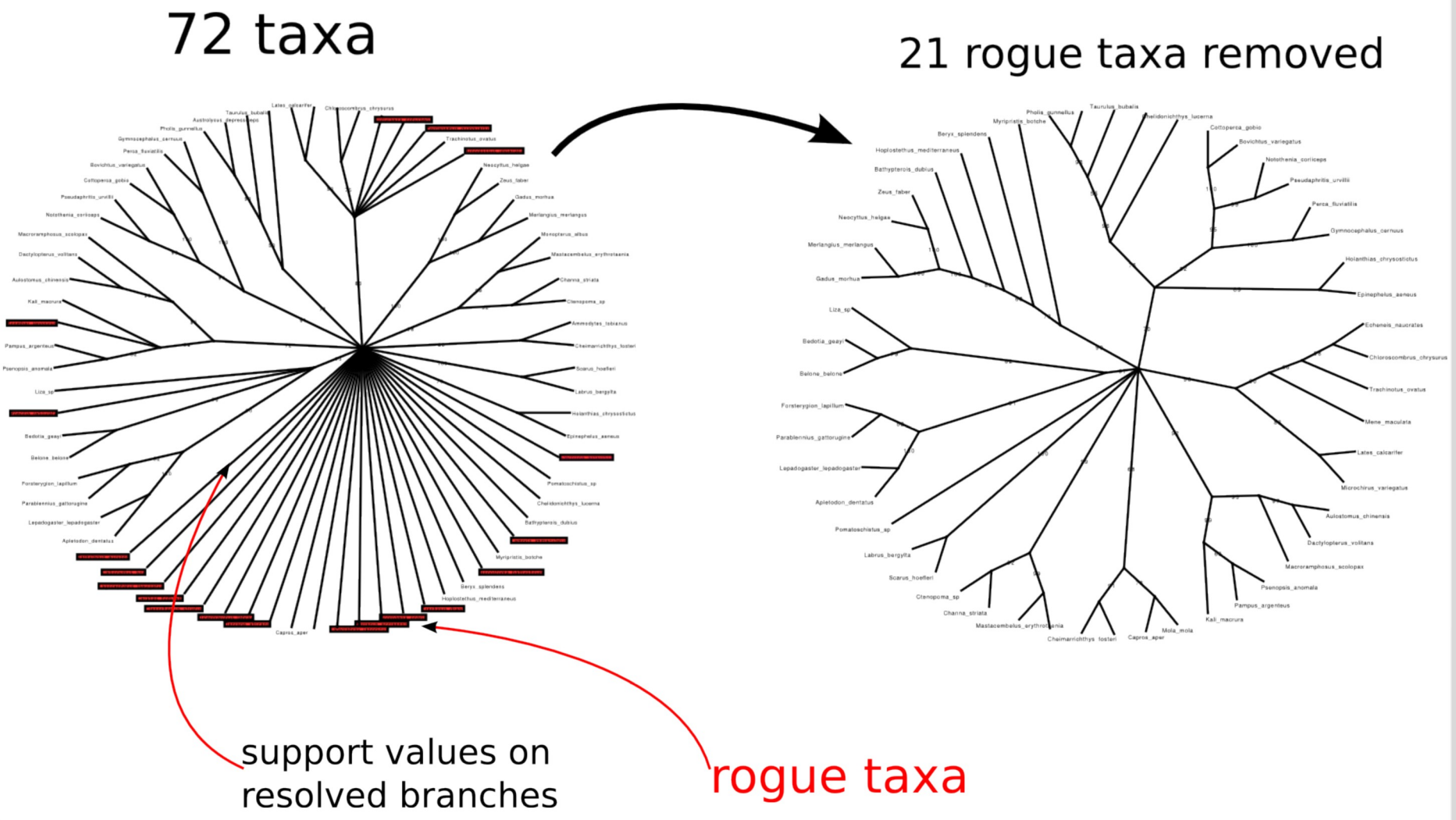
72 taxa



support values on resolved branches

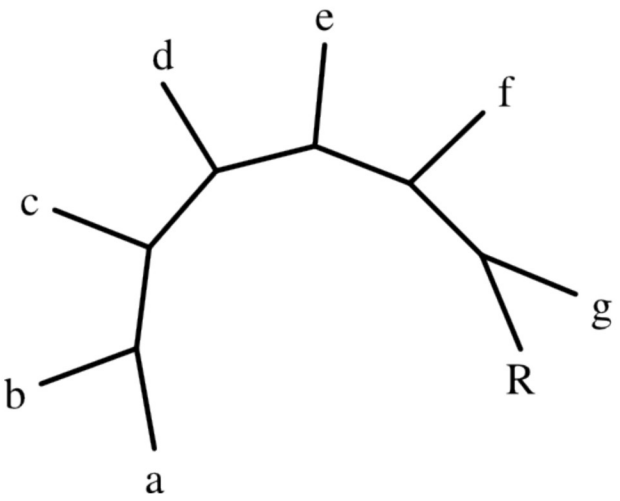
rogue taxa

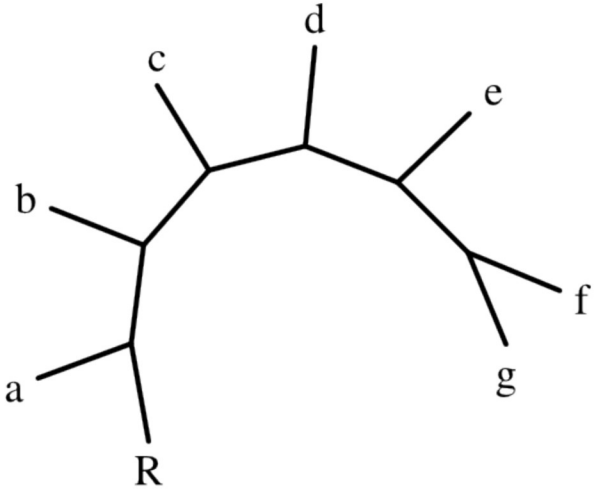
Real-World Example



Problematic for the RF-distance

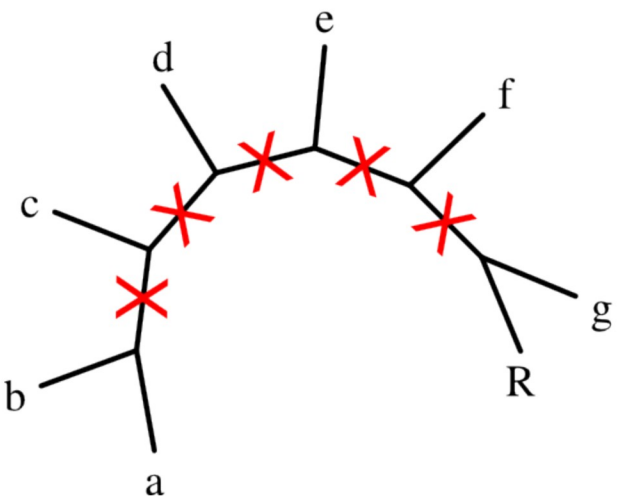
- Consider these comb/caterpillar-like trees:

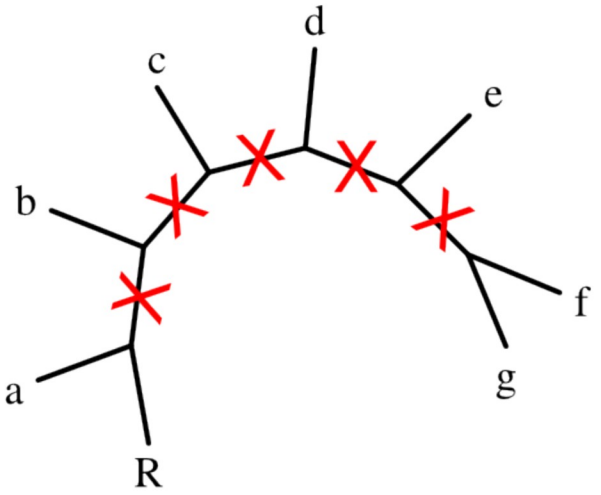




Problematic for the RF-distance

- Consider these comb/caterpillar-like trees:

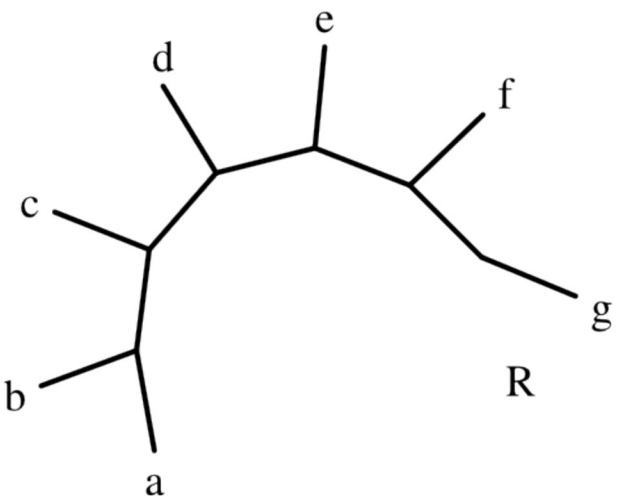


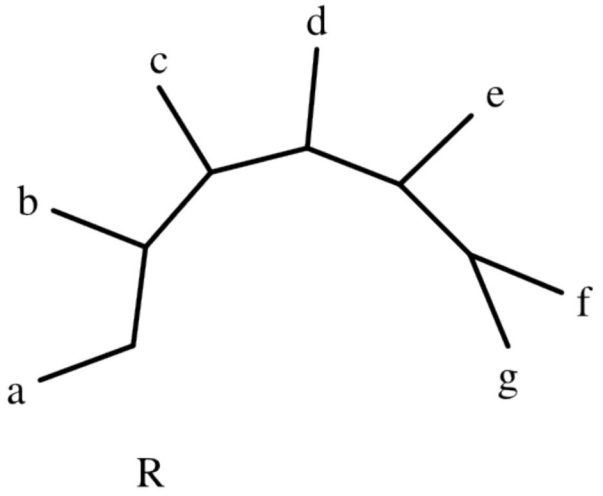


With R: abs. RF-distance = 5; rel. RF-distance = 100%

Problematic for the RF-distance

- Consider these comb/caterpillar-like trees:

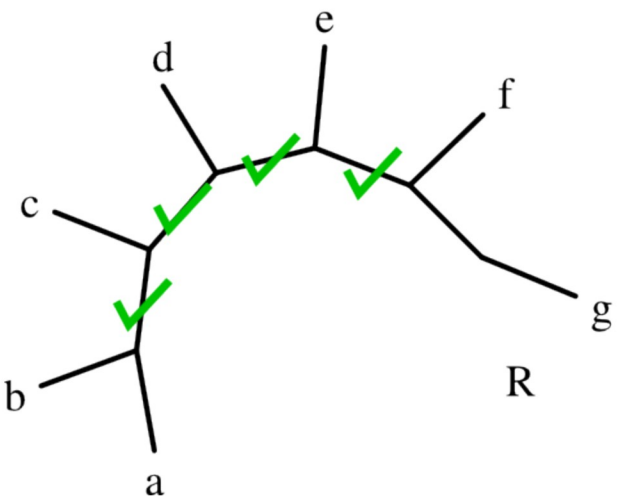


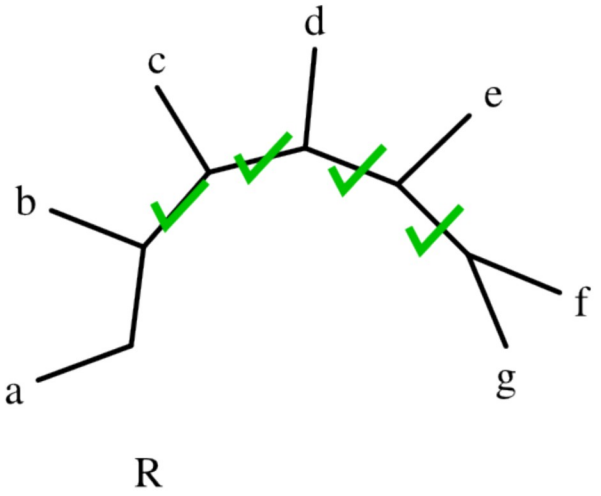


With R: abs. RF-distance = 5; rel. RF-distance = 100%

Problematic for the RF-distance

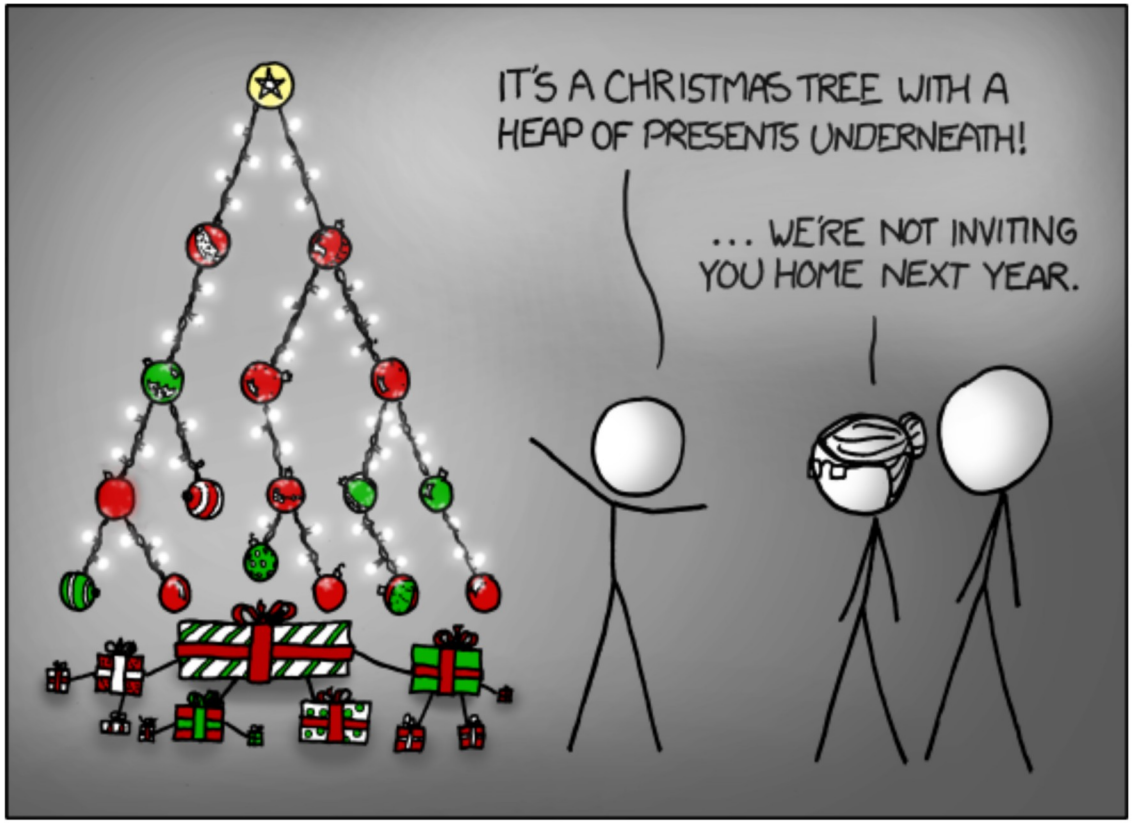
- Consider these comb/caterpillar-like trees:





With R:	abs. RF-distance = 5;	rel. RF-distance = 100%
Without R:	abs. RF-distance = 0;	rel. RF-distance = 0%

Time for questions



https://xkcd.com/835/