Branch support

Molecular Phylogenetics Course 2019

Kevin Gori

kcg25@cam.ac.uk

Outline

- Bootstrapping in general
- Phylogenetic bootstrapping
  - Introduction
  - Walkthrough
  - Exercises
- Fast approximate techniques
  - aLRT
  - aBayes
  - RELL
- Fast bootstrapping – accelerated, not approximated
  - Rapid bootstrap
  - Ultrafast bootstrap

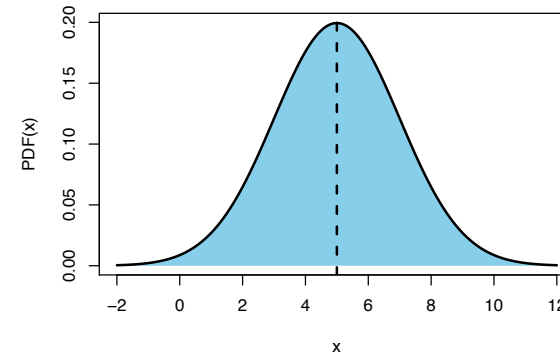# Bootstrapping

How to get something for nothing
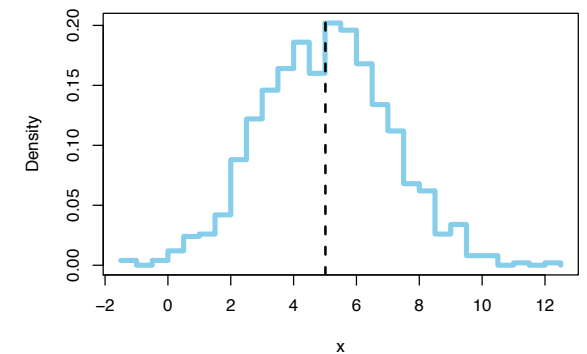
Introduction to the bootstrap

- Purpose
  - Indicates the degree of error in a statistical estimate

- Why?
  - Generating distribution is unknown
  - ... or difficult to work with
  - ... or additional data sets are impossible to obtain

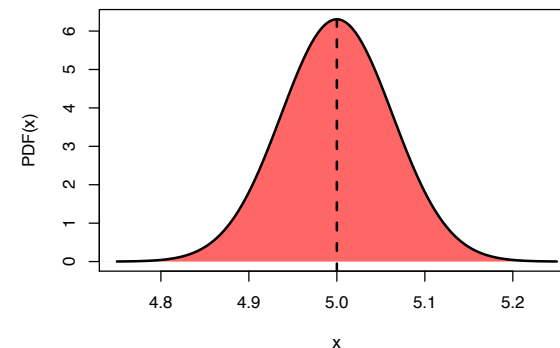Standard error of the mean of a Normal distribution
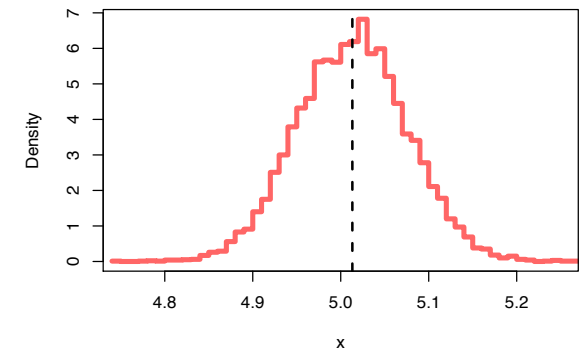
Generating distribution: Normal(μ=5, σ=2)

Sample (N=1000) from Normal(μ=5, σ=2)

Theoretical standard error: Normal(μ=5, σ=2/$\sqrt{1000}$)

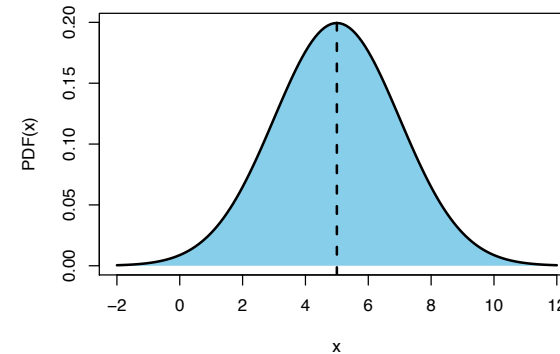Bootstrap estimate of standard error

Introduction to the bootstrap

- Purpose
  - Indicates the degree of error in a statistical estimate

- How?
  - Gather one data set
  - Randomly resample, with replacement, repeatedly
  - Calculate statistic for each bootstrap sample
  - Variability indicates error

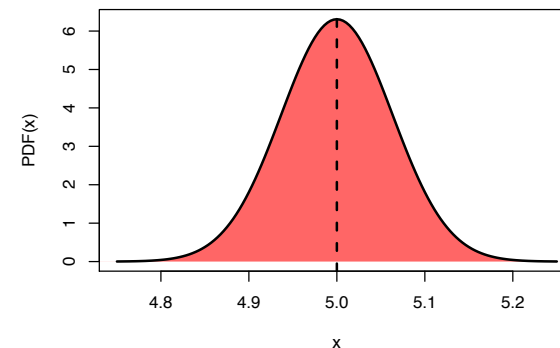Standard error of the mean of a Normal distribution
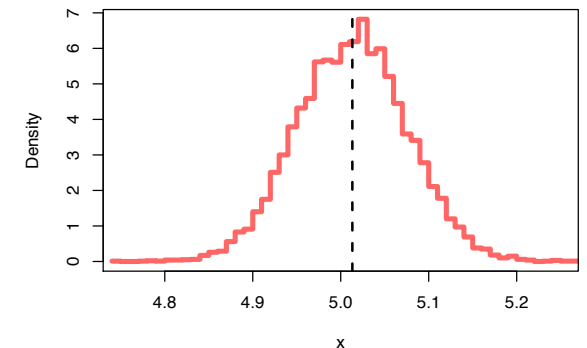
Generating distribution: Normal(μ=5, σ=2)

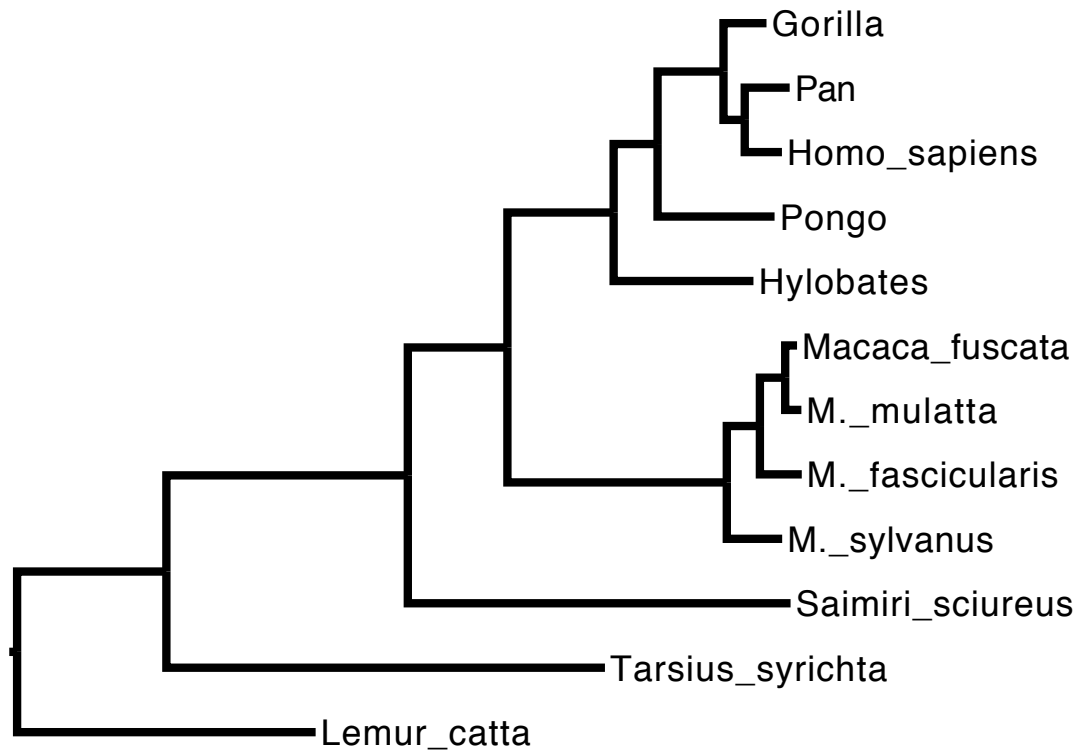Sample (N=1000) from Normal(μ=5, σ=2)
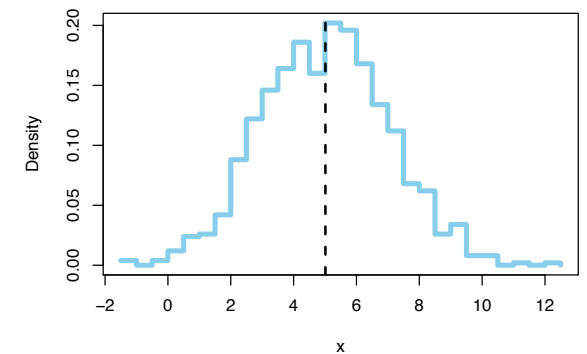
Theoretical standard error: Normal(μ=5, σ=2/$\sqrt{1000}$)

Bootstrap estimate of standard error

Phylogenetic Bootstrap

In phylogenetics the tree is our estimated statistic

Each branch is a division of taxa into groups

Bootstrapping helps us assess which groupings are statistically supported

Making Bootstrap Replicates

- A sequence alignment is a matrix:
  - rows = sequences
  - columns = site patterns

- Columns contain the information about evolutionary relationships

ctg-accatgta
ctggaca-tg--
ctg--ccctgt-

=

$$
\begin{bmatrix} c \\ c \\ c \end{bmatrix} + \begin{bmatrix} t \\ t \\ t \end{bmatrix} + \begin{bmatrix} g \\ g \\ g \end{bmatrix} + \begin{bmatrix} - \\ g \\ - \end{bmatrix} + \ldots
$$

Making Bootstrap Replicates

ctg-accatgta
ctggaca-tg--
ctg--ccctgt-

We resample our columns – with replacement – until we have a new alignment the same size as our original

Making Bootstrap Replicates
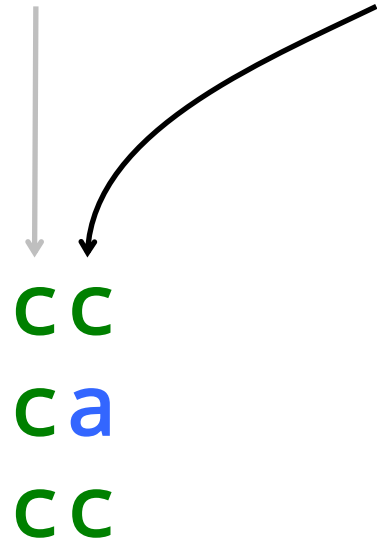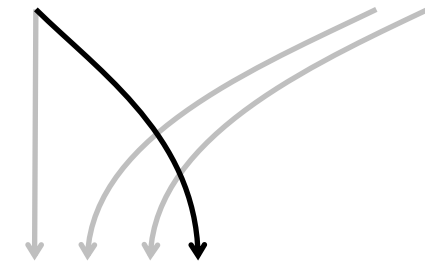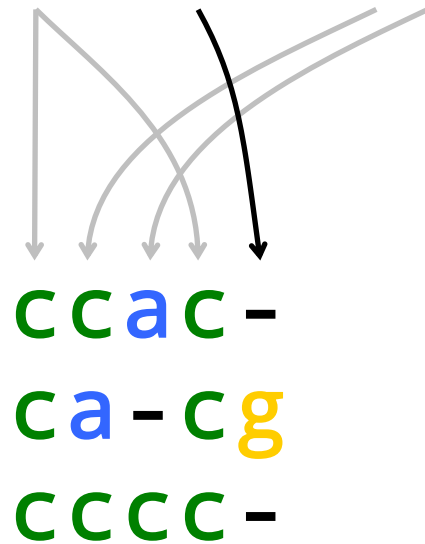
ctg-accatgta
ctggaca-tg--
ctg--ccctgt-

We resample our columns – with replacement – until we have a new alignment the same size as our original

c
c
c

Making Bootstrap Replicates

ctg-accatgta
ctggaca-tg--
ctg--ccctgt-

cc
ca
cc

We resample our columns – with replacement – until we have a new alignment the same size as our original

Making Bootstrap Replicates

ctg-accatgta
ctggaca-tg--
ctg--ccctgt-

We resample our columns – with replacement – until we have a new alignment the same size as our original
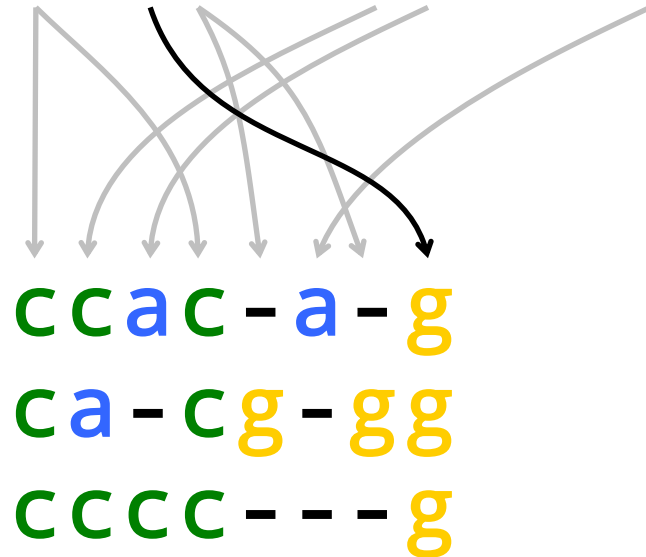
cca
ca-
ccc

Making Bootstrap Replicates

ctg-accatgta
ctggaca-tg--
ctg--ccctgt-

ccac
ca-c
cccc

We resample our columns – with replacement – until we have a new alignment the same size as our original

Making Bootstrap Replicates

We resample our columns – with replacement – until we have a new alignment the same size as our original

Making Bootstrap Replicates

ctg-accatgta
ctggaca-tg--
ctg--ccctgt-

ccac-a
ca-cg-
cccc--

We resample our columns – with replacement – until we have a new alignment the same size as our original
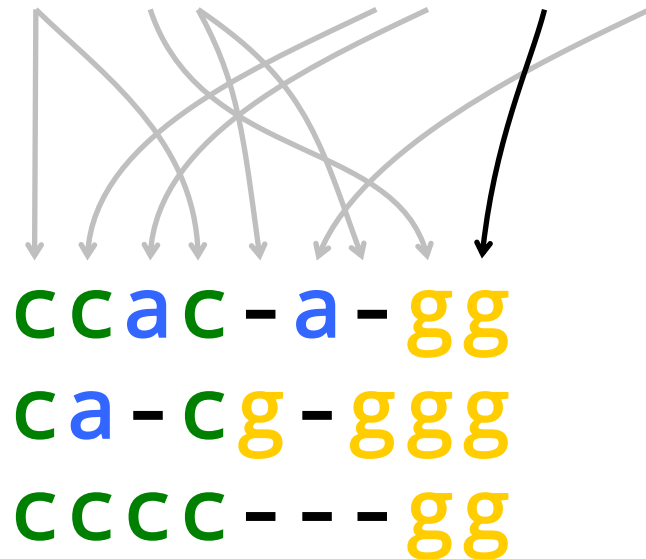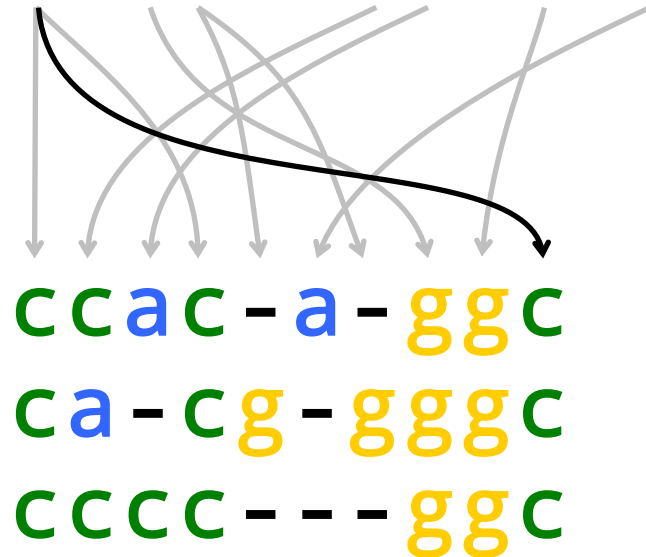
Making Bootstrap Replicates

ctg-accatgta
ctggaca-tg--
ctg--ccctgt-

ccac-a-
ca-cg-g
cccc---

We resample our columns – with replacement – until we have a new alignment the same size as our original

Making Bootstrap Replicates

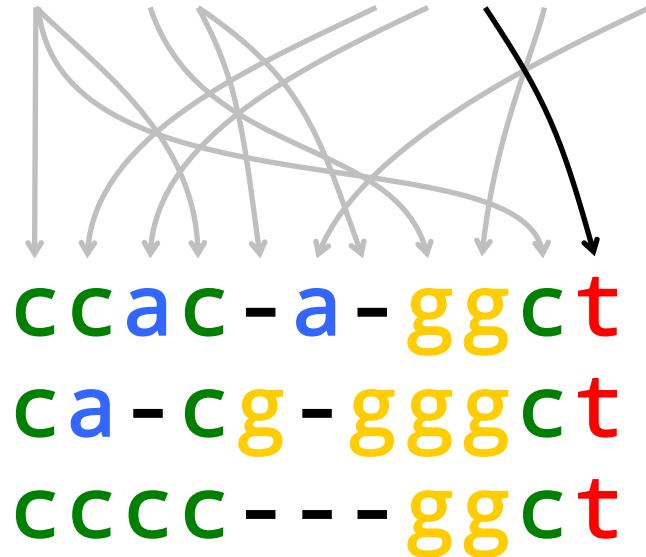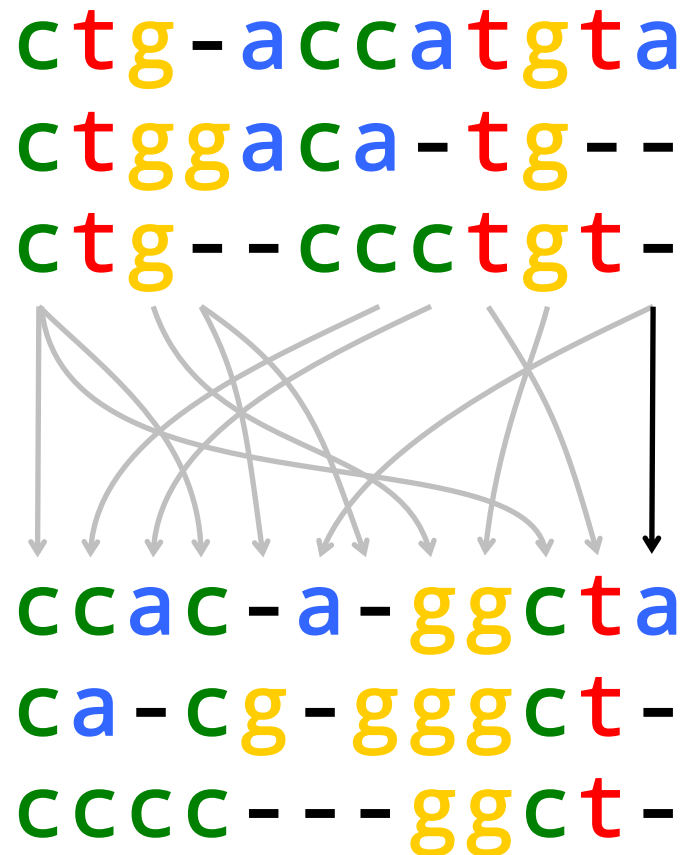
We resample our columns – with replacement – until we have a new alignment the same size as our original

Making Bootstrap Replicates

ctg-accatgta
ctggaca-tg--
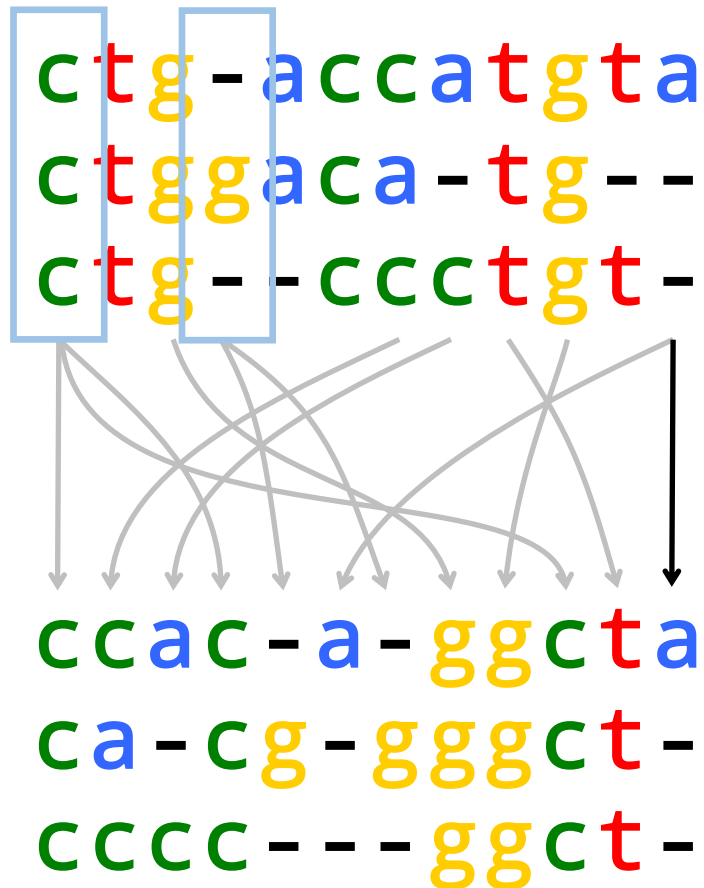ctg--ccctgt-

ccac-a-gg
ca-cg-ggg
cccc---gg

We resample our columns – with replacement – until we have a new alignment the same size as our original

Making Bootstrap Replicates

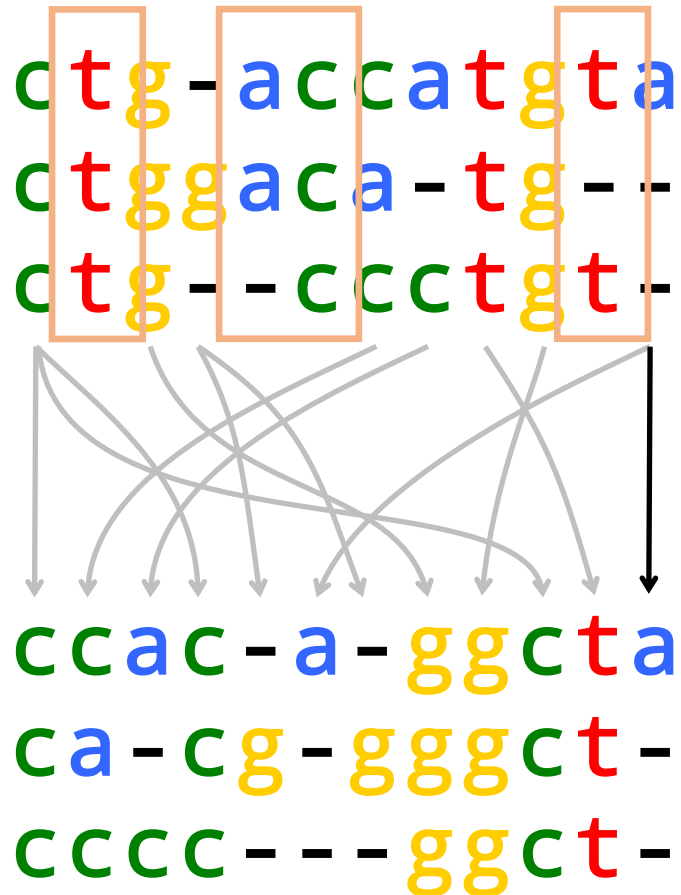We resample our columns – with replacement – until we have a new alignment the same size as our original

Making Bootstrap Replicates

We resample our columns – with replacement – until we have a new alignment the same size as our original

Making Bootstrap Replicates

ctg-accatgta
ctggaca-tg--
ctg--ccctgt-

ccac-a-ggcta
ca-cg-gggct-
cccc---ggct-

We resample our columns – with replacement – until we have a new alignment the same size as our original

Making Bootstrap Replicates

We resample our columns – with replacement – until we have a new alignment the same size as our original

Some columns are sampled more than once

Making Bootstrap Replicates

We resample our columns – with replacement – until we have a new alignment the same size as our original

Some columns are sampled more than once

Some are not sampled at all

Making Bootstrap Replicates

ctg-accatgta
ctggaca-tg--
ctg--ccctgt-

ccac-a-ggcta
ca-cg-gggct-
cccc---ggct-
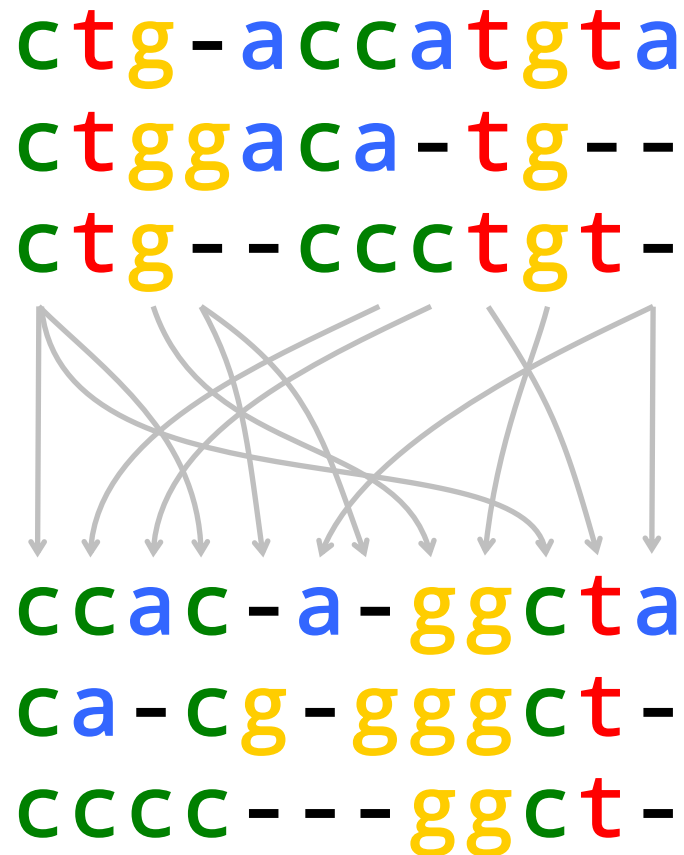
We do this some large number of times (100-1000 is typical)

Then we calculate a new tree – exactly the same way as for our original data – for each replicate

Finally, we use the bootstrap trees to tell us something about the reliability of the original tree

From replicates to trees

1. ccac-a-ggcta
   ca-cg-gggct-
   cccc---ggct-
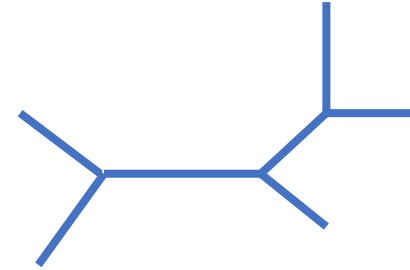
2. ctg-accatgta
   ctggaca-tg--
   ctg--ccctgt-

   ...
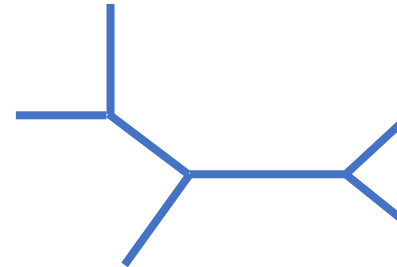
100. atggaac-ccta
     -tgg--agac--
     ctgg-cc-cct-
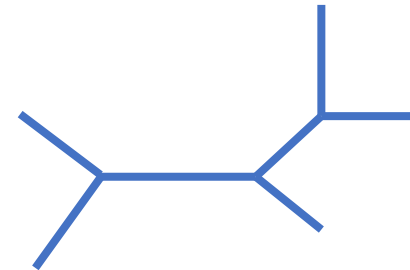
What do bootstrap trees tell us?
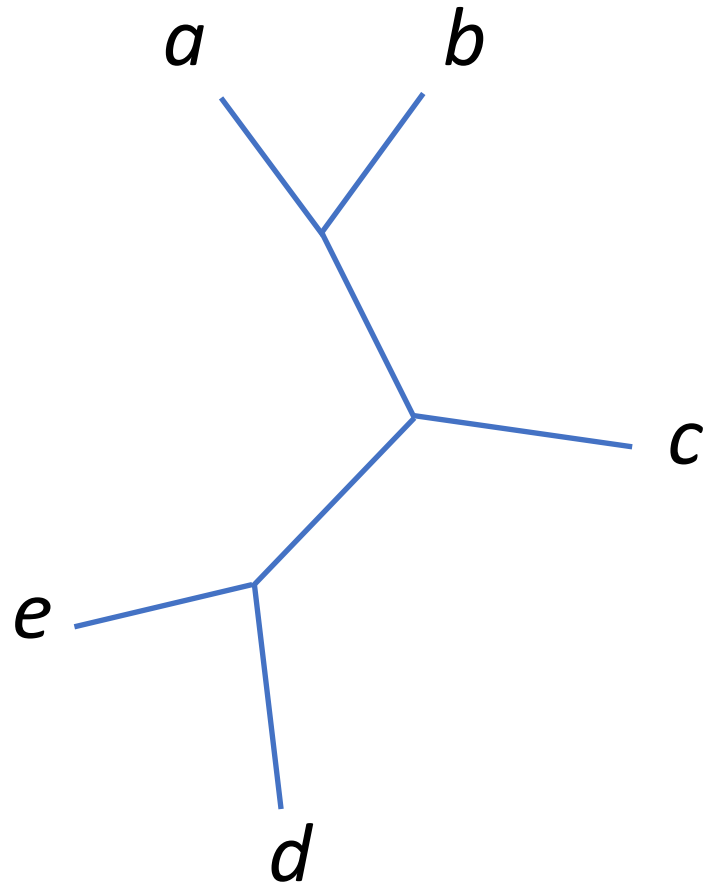
- Bootstrap trees show us a number of plausible relationships between species
- The original tree shows us just one set of relationships
- We can map the relationships – and their frequencies – in the bootstrap sample back on to the original tree
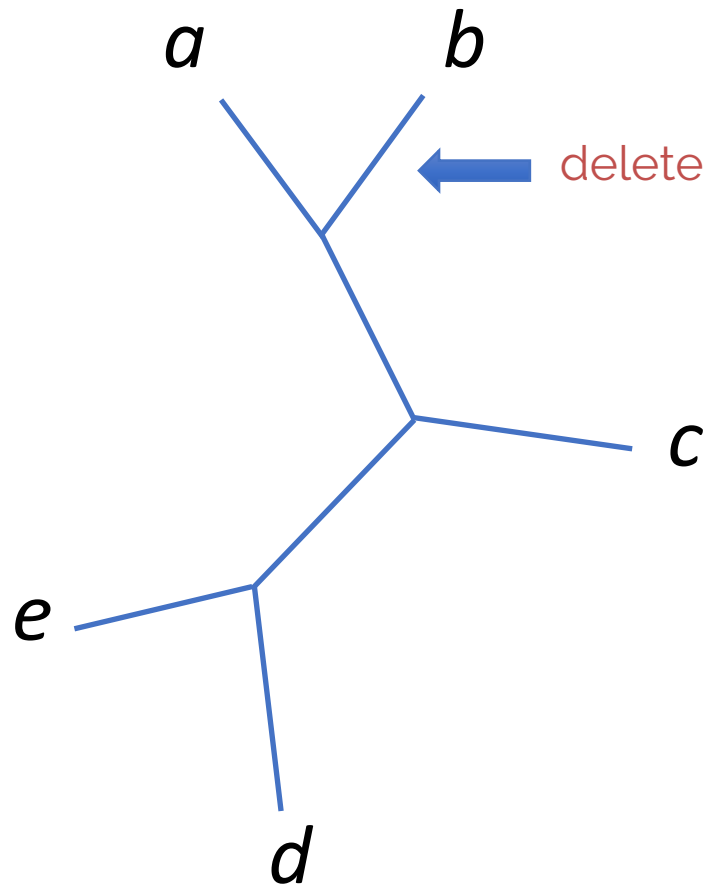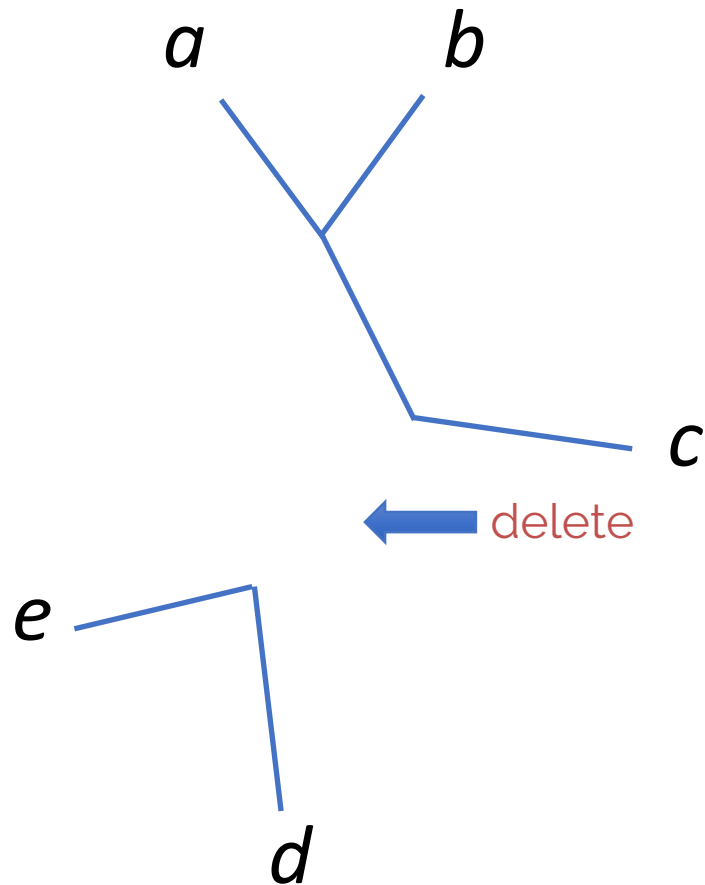- We do this using **splits**

Trees as splits

- Every edge on a tree represents a split – deleting the edge splits the tree into two groups
- Deleting the edge leading to *b* gives us the split *b* | *acde*
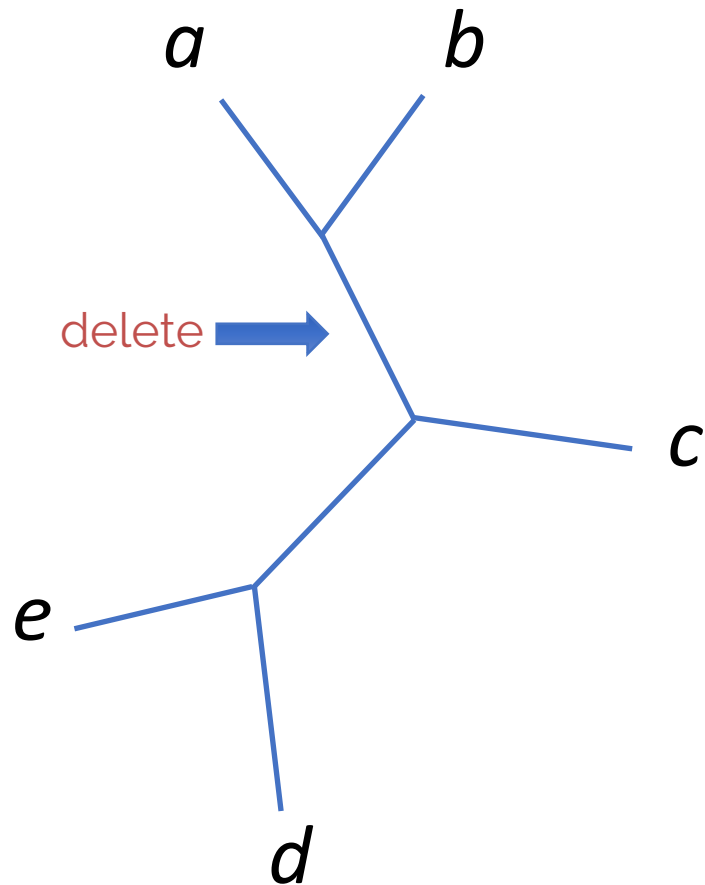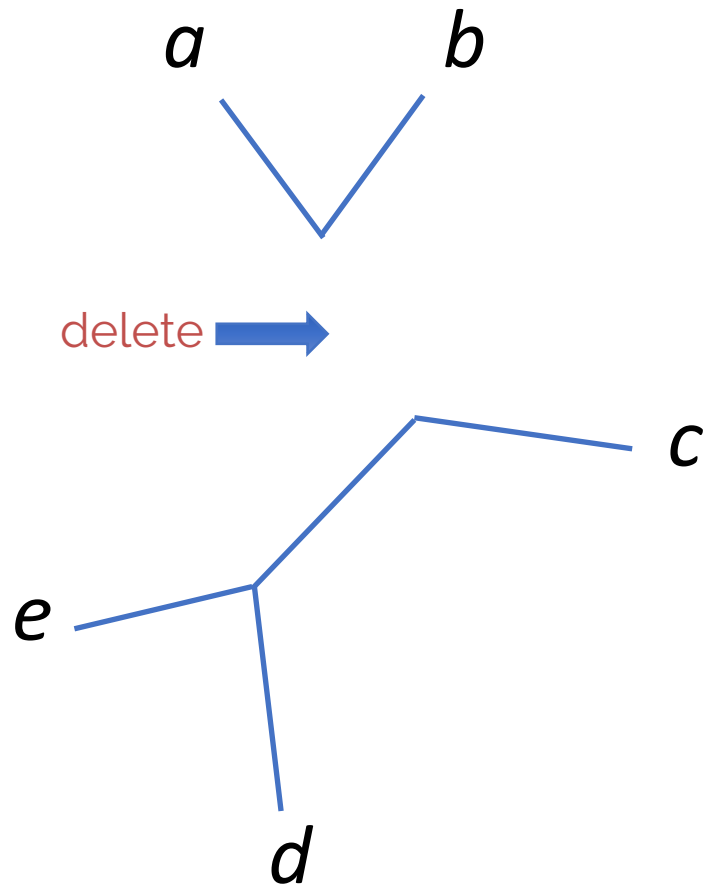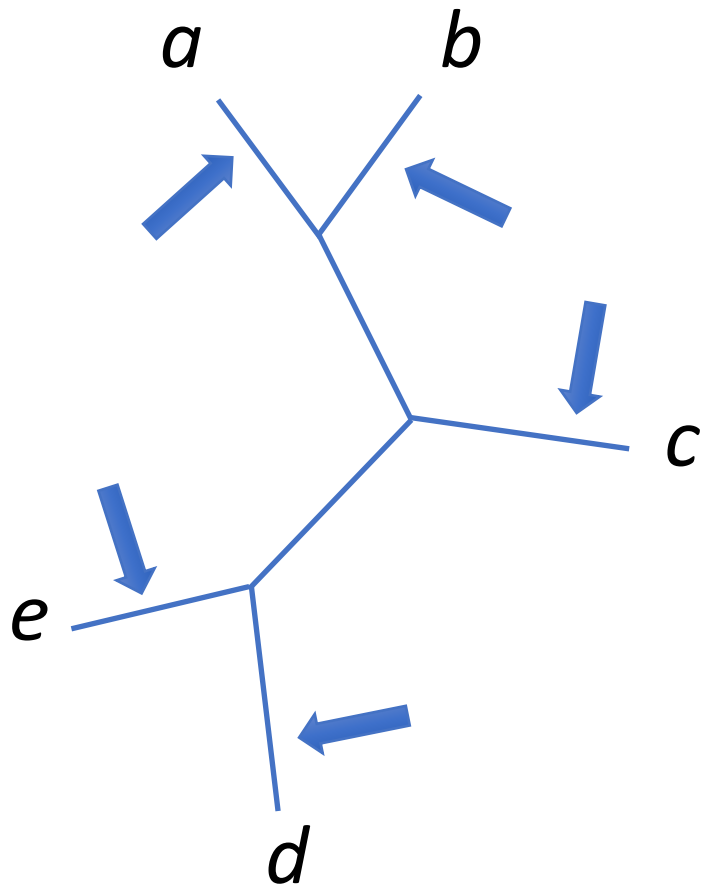
Trees as splits

- Every edge on a tree represents a split – deleting the edge splits the tree into two groups
- Deleting the edge leading to b gives us the split $b\,|\,acde$

Trees as splits

- Every edge on a tree represents a split – deleting the edge splits the tree into two groups

- Deleting the edge leading to *b* gives us the split *b | acde*

Trees as splits

- Every edge on a tree represents a split – deleting the edge splits the tree into two groups
- Deleting the edge here gives us the split *abc | de*

Trees as splits

- Every edge on a tree represents a split – deleting the edge splits the tree into two groups
- Deleting the edge here gives us the split *abc | de*

Trees as splits

- Every edge on a tree represents a split – deleting the edge splits the tree into two groups
- Deleting the edge here gives us the split *abc* | *de*
- And here, *ab* | *cde*

Trees as splits

- Every edge on a tree represents a split – deleting the edge splits the tree into two groups

- Deleting the edge here gives us the split *abc | de*

- And here, *ab / cde*

Trees as splits

- Splits leading to leaves are the same on any tree on the same taxa – they are called "trivial splits", and are ignored

Exercise 1

What are the splits on these trees? Ignore trivial splits
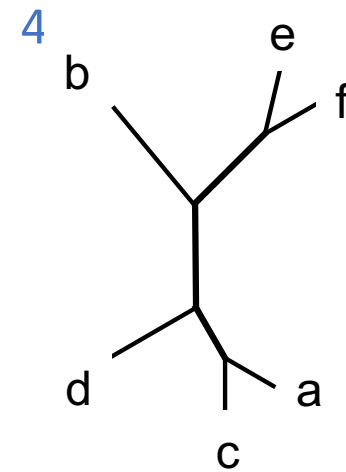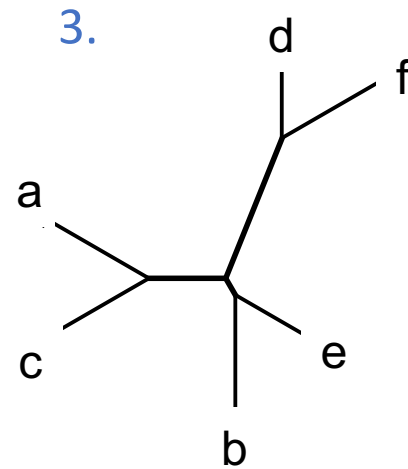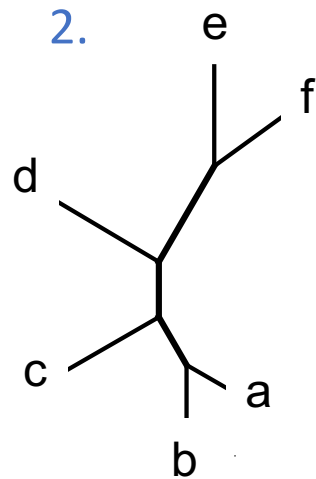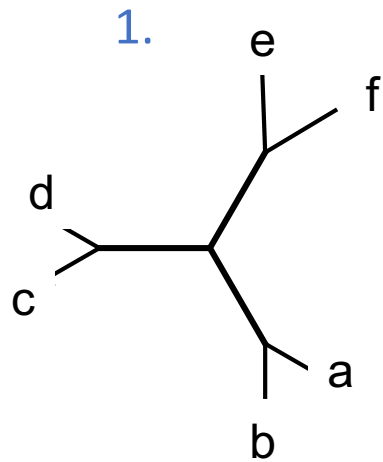
Add bootstrap values to the original tree

1. Find the splits present in the original tree

2. Count each split's frequency in the bootstrap sample, and divide by the size of the sample – this is the **bootstrap split frequency**

3. Map this value onto the edge in the original tree that encodes the split

Exercise 2

- These trees appear in a bootstrap sample. The number of times they appear, out of 100, is given. The left-hand tree (n=70) is the original tree – what are its bootstrap values? (Ignore trivial splits)

n=70
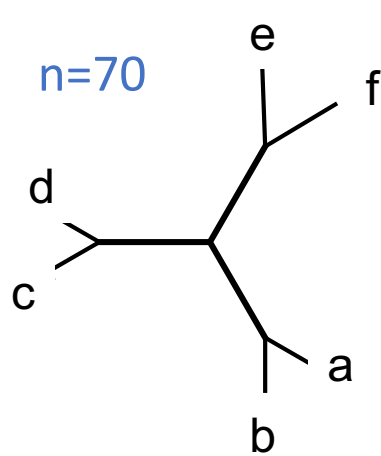
ab|cdef
cd|abef
ef|abcd

n=15
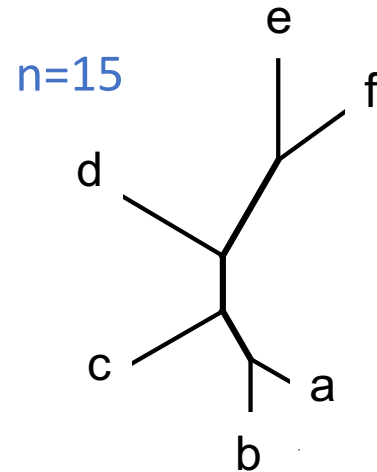
ab|cdef
abc|def
ef|abcd

n=5
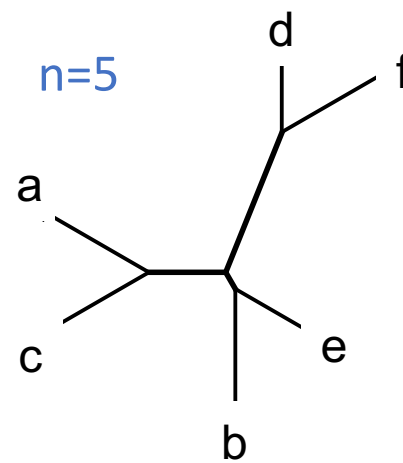
ac|bdef
be|acdf
df|abce

n=10
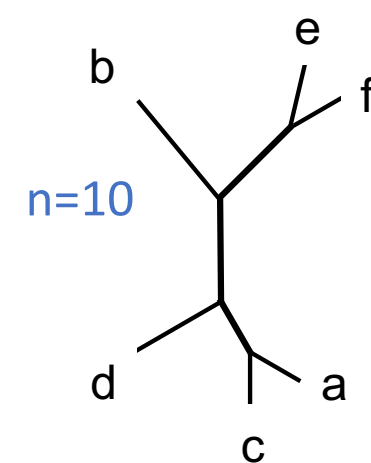
ac|bdef
acd|bef
ef|abcd

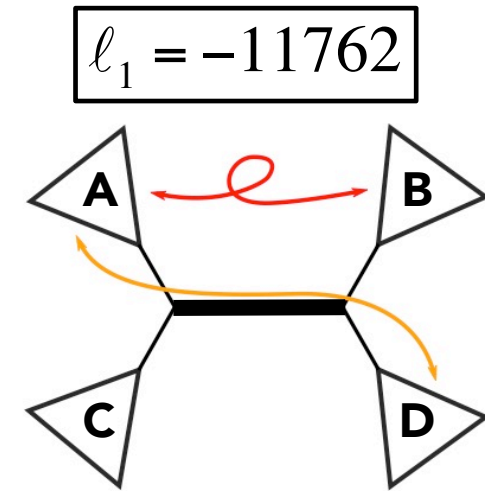# Fast Approximate Techniques

aLRT, aBayes, RELL

aLRT – approx. Likelihood Ratio Test
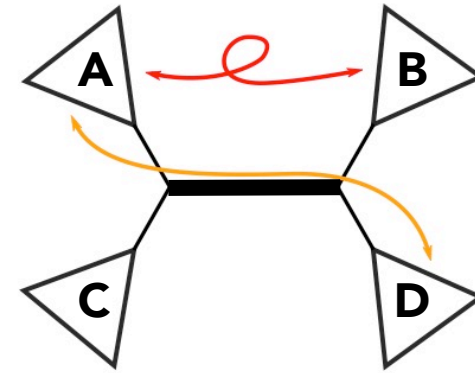
Starting with fully optimised, maximum likelihood tree,

For each internal branch:

- Make two nearest neighbours by switching subtrees (NNI)

aLRT – approx. Likelihood Ratio Test

Starting with fully optimised, maximum likelihood tree,

For each internal branch:

- Make two nearest neighbours by switching subtrees (NNI)

NNI = Nearest neighbour interchange

aLRT – approx. Likelihood Ratio Test

Starting with fully optimised, maximum likelihood tree,

For each internal branch:

- Make two nearest neighbours by switching subtrees (NNI)

- Optimise new branches and calculate new likelihoods

NNI = Nearest neighbour interchange

$$\ell_1 = -11762$$

$$\ell_2 = -11764$$

$$\ell_3 = -11765$$

aLRT – approx. Likelihood Ratio Test

Starting with fully optimised, maximum likelihood tree,

For each internal branch:

- Make two nearest neighbours by switching subtrees (NNI)

- Optimise new branches and calculate new likelihoods

- Compute approximate likelihood ratio test statistic

$$\ell_1 = -11762$$

NNI = Nearest neighbour interchange
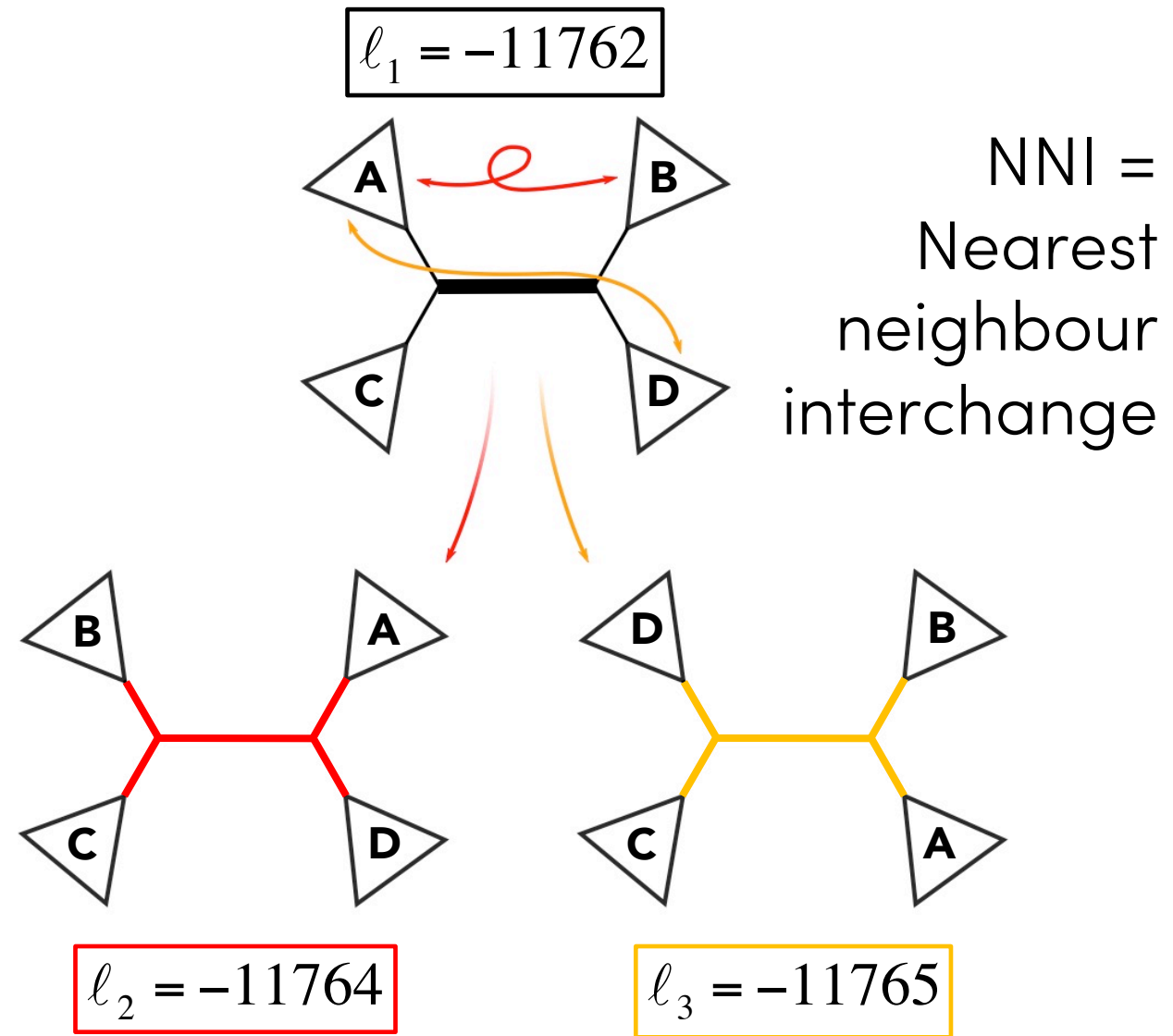
aLRT statistic = $2(\ell_1 - MAX(\ell_2, \ell_3))$
$$= 4$$

$$\ell_2 = -11764 \qquad \ell_3 = -11765$$

aLRT – approx. Likelihood Ratio Test

Starting with fully optimised, maximum likelihood tree,

For each internal branch:

- Make two nearest neighbours by switching subtrees (NNI)

- Optimise new branches and calculate new likelihoods

- Compute approximate likelihood ratio test statistic

- p-value is branch support

$$\ell_1 = -11762$$
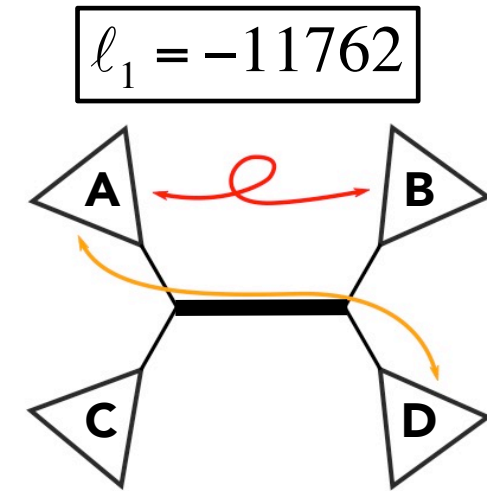
NNI = Nearest neighbour interchange

Mixture of Chi-squared distributions (corrected for multiple testing)

p = 0.966

$$\sim \frac{1}{2}\chi_0^2 + \frac{1}{2}\chi_1^2$$

aBayes – approximate Bayes

Probability of tree given data+model

$$P\left(T_1|D\right) = \frac{P\left(D|T_1\right)P\left(T_1\right)}{\sum_{i=1}^{3} P\left(D|T_i\right)P\left(T_i\right)}$$

Equal priors:

$$P\left(T_1\right) = P\left(T_2\right) = P\left(T_3\right)$$

Bayesian alternative to aLRT

- Produce & optimise three NNI trees, as with aLRT
- Posterior probability ∝ Likelihood × prior probability
- Normalising term is simple, only three configurations

aBayes – approximate Bayes

Probability of tree given data+model

$$P\left(T_1|D\right) = \frac{P\left(D|T_1\right)\,\cancel{P\left(T_1\right)}}{\sum_{i=1}^{3} P\left(D|T_i\right)\,\cancel{P\left(T_i\right)}}$$

Equal priors:

$$P\left(T_1\right) = P\left(T_2\right) = P\left(T_3\right)$$

Bayesian alternative to aLRT

- Produce & optimise three NNI trees, as with aLRT
- Posterior probability ∝ Likelihood × prior probability
- Normalising term is simple, only three configurations
- Equal priors cancel out

aBayes – approximate Bayes

Probability of tree given data+model

$$P\left(T_1|D\right) = \frac{P\left(D|T_1\right)\,\cancel{P\left(T_1\right)}}{\sum_{i=1}^{3} P\left(D|T_i\right)\,\cancel{P\left(T_i\right)}}$$

Simplifies:

$$\text{aBayes} \;=\; \frac{L_1}{L_1 + L_2 + L_3}$$

Bayesian alternative to aLRT

- Produce & optimise three NNI trees, as with aLRT

- Posterior probability ∝ Likelihood × prior probability

- Normalising term is simple, only three configurations

- Simplifies to calculation involving three likelihoods
  (note: not log-likelihoods)

Exercise 3

1. We are calculating the aBayes value for a branch. If the **log-likelihood** of the tree is **–1**, and the **log-likelihoods** of the two NNI rearrangements are **–3** and **–4**, what is the aBayes value?

2. As for question 1, but the log-likelihood of the tree is **–2**, and the log-likelihoods of the two NNI rearrangements are **–4** and **–5**, what is the aBayes value?

$$\text{aBayes} = \frac{L_1}{L_1 + L_2 + L_3}$$

log-Lik	-1	-2	-3	-4	-5
Lik	0.368	0.135	0.0500	0.0183	0.00674

Fast local bootstrap and SH-aLRT

These two methods use **R**esampled **E**stimated **L**og-Likelihoods (RELL)

How to RELL:

- Randomly resample with replacement the sitewise likelihoods

- Branch support is frequency among RELL samples that
  - T1 > max(T2, T3) (local bootstrap)
  - T1 passes modified Shimodaira-Hasegawa test (SH-aLRT)

1: Find the neighbour trees as in aLRT
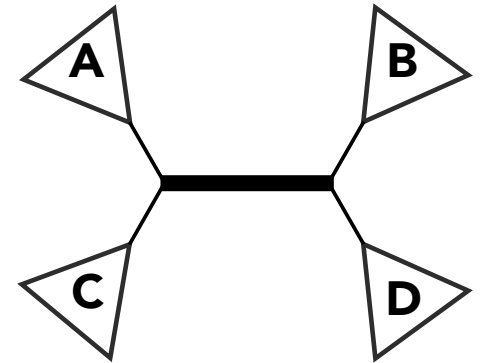
2: Reoptimise branches
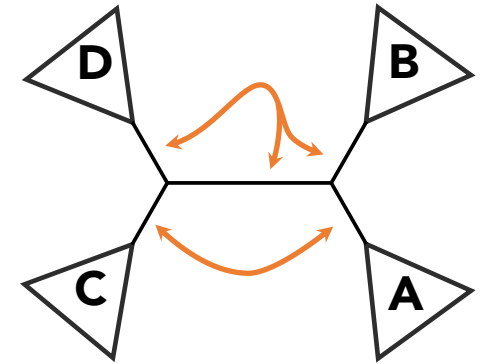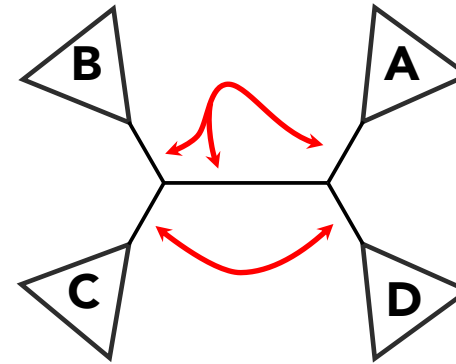
3: Calculate likelihoods per-site

Fast local bootstrap and SH-aLRT

Calculate sitewise log-likelihoods for each of the 3 trees

c t g - a c c a t g t a
c t g g a c a - t g - -   ...
c t g - - c c c t g t -

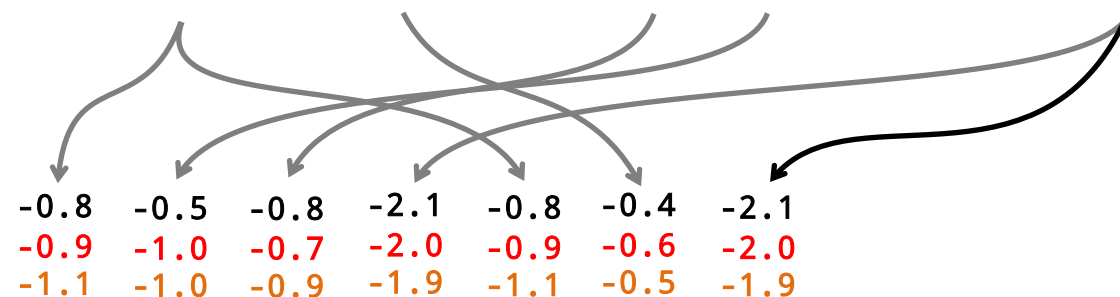| -1.2 | -0.8 | -1.0 | -0.4 | -2.1 | -0.8 | -0.5 | -1.1 | -1.3 | -2.1 | -0.6 | -0.9 |
| -1.4 | -0.9 | -1.0 | -0.6 | -2.4 | -0.7 | -1.0 | -1.3 | -1.4 | -2.0 | -1.0 | -0.9 |
| -1.4 | -1.1 | -1.0 | -0.5 | -2.5 | -0.9 | -1.0 | -1.3 | -1.5 | -1.9 | -0.9 | -0.9 |

Fast local bootstrap and SH-aLRT

Resample Estimated Log-Likelihoods (RELL)

Fast local bootstrap and SH-aLRT

Resample Estimated Log-Likelihoods (RELL)

Fast local bootstrap and SH-aLRT

Resample Estimated Log-Likelihoods (RELL)

Fast local bootstrap and SH-aLRT

Resample Estimated Log-Likelihoods (RELL)

Fast local bootstrap and SH-aLRT

Resample Estimated Log-Likelihoods (RELL)

Fast local bootstrap and SH-aLRT

Resample Estimated Log-Likelihoods (RELL)

Fast local bootstrap and SH-aLRT

Resample Estimated Log-Likelihoods (RELL)

Fast local bootstrap and SH-aLRT

Resample Estimated Log-Likelihoods (RELL)

Fast local bootstrap and SH-aLRT

Resample Estimated Log-Likelihoods (RELL)

Fast local bootstrap and SH-aLRT

Resample Estimated Log-Likelihoods (RELL)

Fast local bootstrap and SH-aLRT

Resample Estimated Log-Likelihoods (RELL)

Fast local bootstrap and SH-aLRT

Resample Estimated Log-Likelihoods (RELL)

Fast local bootstrap and SH-aLRT

Resample Estimated Log-Likelihoods (RELL)

Fast local bootstrap and SH-aLRT

Total log-likelihoods are the row sums for the three trees

SH-aLRT

For each RELL:

o the **centred** likelihood is:-

$$\ell_i^C = \ell_i^* - \ell_i$$

o For each RELL replicate, we have 3 centred likelihoods – one for each NNI rearrangement

o Calculate: $\quad \Delta = \ell_1^C - MAX(\ell_2^C, \ell_3^C)$

o SH-aLRT$_n$, the branch support given by RELL replicate n, is:

$\quad$ 1, $\quad\quad$ if $\ell_1 - MAX(\ell_2, \ell_3) > \Delta$

$\quad$ 0, $\quad\quad$ otherwise

Faster Bootstrapping

Rapid bootstrap, Ultrafast bootstrap

RAxML – Rapid Bootstraps

Optimisation shortcuts:

1. Parameters estimated once on an initial parsimony tree

2. Shallow tree search

3. CAT model of rate variation (faster than Gamma)

4. Uses final tree of each bootstrap run to initialise the next (resets to parsimony every 10 steps)

Correlation of standard and rapid bootstraps
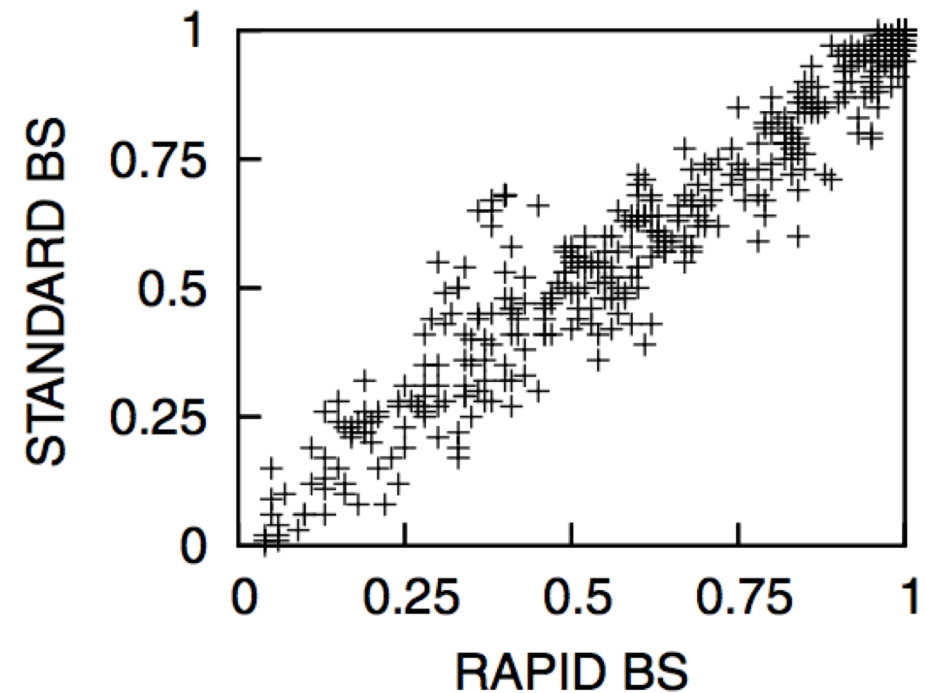
IQtree – Ultrafast Bootstraps
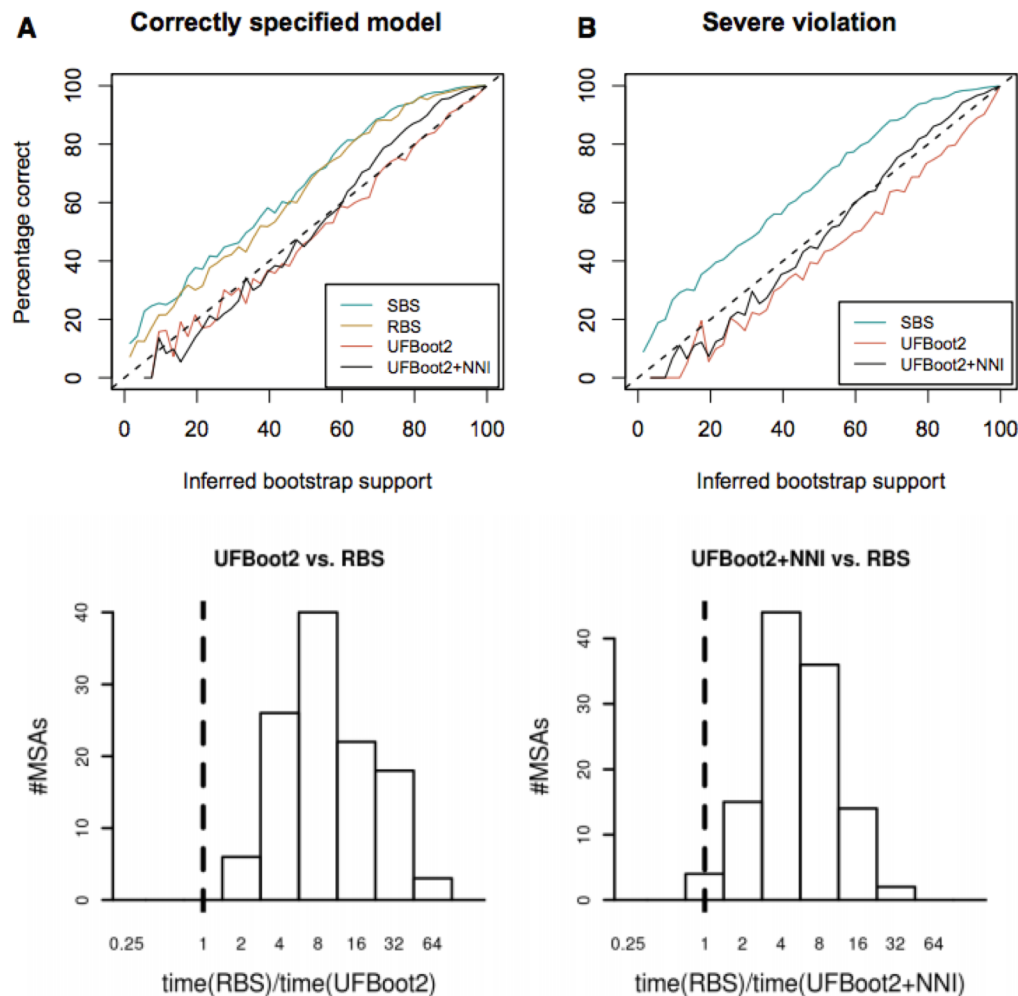
Bootstrapping via tree search

- Create N bootstrapped alignments (BA)

- Conduct tree search on original alignment

- Assess each candidate tree for each BA using RELL

- Assign best RELL tree to BA (never search BA directly)

- +NNI: Final round of optimisation on BA at end of tree search

Practical

1. Use IQtree to run GTR+G4 on `carnivores_16S.fas`s
   - Use the ultrafast bootstrap to compute 1000 bootstrap replicates
   - Repeat the analysis, using aLRT, SH–aLRT, aBayes and local bootstrap probabilities.
   - Do the different measures agree?
   - Can you identify poorly supported branches?
   - Compute 100 full bootstrap replicates (~15 minutes)

2. Calculate Bayesian trees using Beast, and compare posterior probabilities to bootstrap values
   - compare run times – which is faster, MCMC or bootstrapping?.

IQtree quick start guide

Run analysis using model, e.g. HKY

```
iqtree -s alignment.fa -m HKY -pre hky
```

Add gamma model of rate variation

```
iqtree -s alignment.fa -m HKY+G4 -pre hky_gamma
```

Use 100 standard bootstraps

```
iqtree -s alignment.fa -m HKY+G4 -b 100 -pre hky_gamma_boot
```

Use 1000 ultrafast bootstraps

```
iqtree -s alignment.fa -m HKY+G4 -bb 1000 -pre hky_gamma_ufboot
```

Use 100 SH-aLRT RELL replicates

```
iqtree -s alignment.fa -m HKY+G4 -alrt 100 -pre hky_gamma_shalrt
```

Use alrt

```
iqtree -s alignment.fa -m HKY+G4 -alrt 0 -pre hky_gamma_alrt
```

Use aBayes

```
iqtree -s alignment.fa -m HKY+G4 -abayes -pre hky_gamma_abayes
```

Use 1000 fast local bootstrap probabilities

```
iqtree -s alignment.fa -m HKY+G4 -lbp 1000 -pre hky_gamma_lbp
```

References

Invention of the bootstrap:

Efron, B (1979) Bootstrap Methods: Another Look at the Jackknife. Ann. Stat

First use of bootstrap in phylogenetics:

Felsenstein, J. (1985). Confidence Limits on Phylogenies: An Approach Using the Bootstrap. Evolution; International Journal of Organic Evolution, 39(4), 783–791.

aLRT:

Anisimova, M., & Gascuel, O. (2006). Approximate likelihood-ratio test for branches: A fast, accurate, and powerful alternative. Systematic Biology, 55(4), 539–552.

RELL:

Kishino, Miyata, and Hasegawa. (1990). "Maximum Likelihood Inference of Protein Phylogeny and the Origin of Chloroplasts." *Journal of Molecular Evolution* 31 (2): 151–60.

Hasegawa, M., and H. Kishino. (1994). "Accuracies of the Simple Methods for Estimating the Bootstrap Probability of a Maximum-Likelihood Tree." *Molecular Biology and Evolution* 11 (1): 142–142.

aBayes and SH-aLRT:

Anisimova, M., *et al.* (2011). Survey of branch support methods demonstrates accuracy, power, and robustness of fast likelihood-based approximation schemes. Systematic Biology, 60(5), 685–699.

RAxML rapid bootstrap:

Stamatakis, A, *et al.* (2008). A Rapid Bootstrap Algorithm for the RAxML Web Servers. *Systematic Biology* 57 (5)

Ultrafast bootstrap:

Minh, B. Q., *et al.* (2013). "Ultrafast Approximation for Phylogenetic Bootstrap." *Molecular Biology and Evolution* 30 (5): 1188–95.

Hoang, *et al.* (2018) UFBoot2: Improving the Ultrafast Bootstrap Approximation, *Molecular Biology and Evolution* 35 (2): 518–522