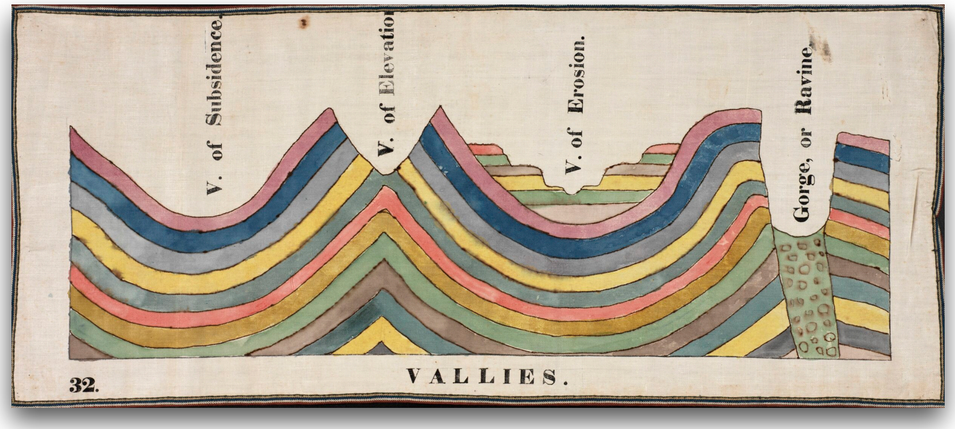

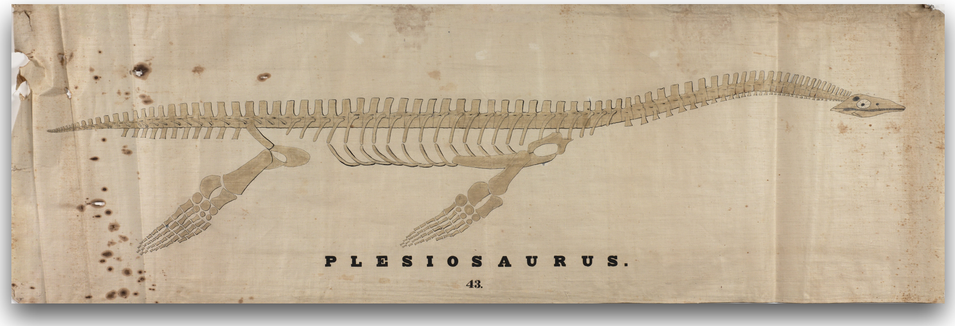

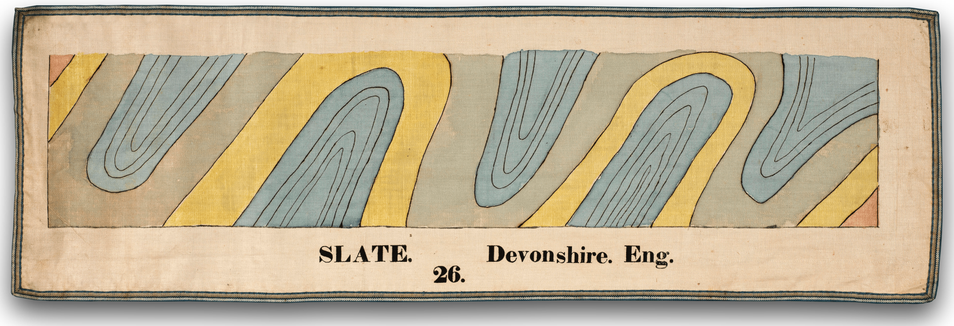

Hypothesis testing

Phylogenetic trees are just …

Phylogenetic trees are just … hypotheses!

Phylogenetic trees are just … hypotheses!

… and we can test them!

Tree topology tests

Statistical tests used to compare alternative phylogenetic trees and assess whether differences in their fit to the data are significant.

Why use them?

- to test specific evolutionary hypotheses e.g. monophyly of a group
- to assess the impact of constraints or model choices on tree inference
- to reduce search space when exhaustive searches are impossible

Tree topology tests and **constrained inference** serve different purposes, however they are closely related conceptually.

We enforce certain relationships during tree search, often based on well-established knowledge. For example, Vertebrata is monophyletic.

Narrows tree space and focuses on biologically plausible topologies.

- **hard constraints**: enforce strict relationships
  A and B must be sisters.

- **soft constraints**: more flexibile within broader groupings
  A, B, and C must be monophyletic, but their branching order is not fixed

Common tree topology tests:

- **bp-RELL test**
  fast but less robust than formal statistical tests.

- **Kishino–Hasegawa (KH) test**
  pairwise test - assumes trees were pre-specified, not data-driven

- **Shimodaira–Hasegawa (SH) test**
  Adjusts for multiple comparisons between topologies

- **Approximately Unbiased (AU) test**
  More accurate, especially for many competing trees

- **Expected Likelihood Weights (ELW) test**
  Computes relative support for each tree based on expected likelihoods.

They all rely on the **RELL approximation**!

The **Resampling Estimated Log-Likelihoods (RELL) approximation** (Kishino et al. 1990) is a cornerstone method used in tree topology tests.

In topology tests we want to estimate how **tree lnL vary across replicates**.

Full parametric bootstrapping would require **re-estimating parameters and tree likelihoods** for each replicate - computationally expensive.

It is a **bootstrap-based method** used to **approximate the sampling distribution** of differences in tree likelihoods across sites - without the need to optimize again parameters for each replicate.

RELL simplifies this by:
- keeping likelihoods fixed per site and tree
- resampling sites only, not full trees or parameters
- making bootstrap replicates of site-wise likelihoods

Compute site-wise log-likelihoods for each tree topology using a matrix with:
- rows = **sites**
- columns = **trees**
- values = **lnL**

	site 1	site 2	site 3	site 4	site 5	site 6	site 7	site 8	site 9
tree 1	-104124.0580	-1043324.0580	-104344.0350	-204124.0580	-1045324.0580	-104344.0350	-204124.0580	-1945324.0580	-1047724.0580
tree 2	-272924.0580	-104124.0580	-1043324.0580	-272924.0580	-104344.0350	-204124.0580	-1045324.0580	-104344.0350	-204124.0580
tree 3	-1043324.0580	-104344.0350	-272924.0580	-104124.0580	-1043324.0580	-204124.0580	-204124.0580	-1043324.0580	-104344.0350
tree 4	-104124.0580	-1043324.0580	-104344.0350	-204124.0580	-104124.0580	-1043324.0580	-204124.0580	-104344.0350	-204124.0580

bp-RELL test

In brief:

1. calculate Site-wise Log-Likelihoods for each input tree
2. use the RELL method to bootstrap alignment sites and generate many pseudo-replicates

   then for each bootstrap pseudo-replicate:

   3. sum the resampled site log-likelihoods to get total likelihoods
   4. identify the best-scoring tree in that replicate
5. the bootstrap proportion (bp) for a tree is the fraction of replicates where it had the highest lnL

Interpreting results:

• bp values range from 0 to 1. Higher values represent more frequent best performance

• bp-RELL is fast and easy to compute, but it is not a formal statistical test

Kishino-Hasegawa (KH) test

Start with *two trees* that you want to compare. Assumes *trees are hypothesis-driven*.

1. calculate Site-wise Log-Likelihoods of the 2 trees - how much each site supports each tree
2. for each site, calculate the difference in lnL between Tree $_A$ and Tree $_B$

   $\Delta_{lnL} = lnL_{Tree\ A} - lnL_{Tree\ B}$

   this creates a list of *per site* differences
3. paired t-test to assess if the average log-likelihood difference is significantly different from zero
4. returns a p value indicating whether the worse-scoring tree is statistically rejected

Interpreting results:

- $p < 0.05$: the worse-scoring tree is significantly worse - reject it!
- $p \geq 0.05$: no significant difference between the two trees - cannot reject either tree 🙃

Shimodaira-Hasegawa (SH) test

Designed to compare multiple phylogenetic trees using the same alignment data. It asks: *“Are any of these trees significantly worse than the best one?”*. In brief:

1. calculate Site-wise Log-Likelihoods of multiple input trees

2. identify the “*best tree*” by summing site log-likelihoods of each tree

3. bootstrap resampling of sties using the RELL method to create many replicates of dataset

4. for each replicate:
   - sum the resampled site-wise log-likelihoods for each tree
   - compute log-likelihood differences between each tree and the “*best tree*”

5. this produces a bootstrap distribution of likelihood differences for each tree

6. p values are calculated as described previously and adjusted for multiple test comparisons

Both KH and SH give back a p value, the latter for each tree relative to the “*best tree*”. However:

- KH test compares two or few trees, has no correction for multiple testing
- SH test compares many trees, conservative as it adjusts for testing multiple topologies

Approximately Unbiased (AU) test

In brief:

1. calculate Site-wise Log-Likelihoods of multiple input trees

2. identify the "*best tree*" by summing site log-likelihoods of each tree

3. multiscale bootstrap resampling of sites using the RELL method, but at different block sizes (e.g. 0.5X 1X and 2X the dataset size) to better approximate the true distribution

4. for each replicate:
   - sum the resampled site-wise log-likelihoods for each tree
   - compute log-likelihood differences between each tree and the "*best tree*"

5. calculate an accurate distribution of likelihood differences, accounting for selection bias

6. p values are then calculated from this improved distribution

AU provide p values for each tree relative to the best, and is recommended over KH and SH tests:

- KH test is too liberal if used with data-derived trees, leading to inflated Type I errors
- SH test becomes too conservative and rejects few trees when testing many trees

Expected Likelihood Weights (ELW) test

ELW test provides a way to estimate the relative support of each candidate tree, based on *how well it explains the data*. Rather than returning p values, it assigns weights that sum to 1 across all trees.

1. calculate Site-wise Log-Likelihoods of multiple input trees

2. identify the "*best tree*" by summing site log-likelihoods of each tree

3. bootstrap resampling of sties using the RELL method to create many replicates of dataset

4. for each replicate:
   - sum the resampled site-wise log-likelihoods for each tree
   - compute log-likelihood differences between each tree and the "*best tree*"
   - compute relative likelihood of each tree, based on how its total likelihood compares to others.

5. calculate ELW for each tree as the proportion of replicates in which it performs well, weighted by its relative likelihood

Interpreting results:

ELW values range from 0 to 1 and sum to 1 across all trees A tree with a high ELW (e.g., >0.95) has strong support Lower values indicate less support, but not necessarily rejection

FINISH