Chapter 5
Evolution Between Species: Phylogeny

5.1 Trees of Life

In 1834 Charles Darwin (1809–1882) wrote in his notebook "I think"—but instead of finishing the sentence, he drew a phylogeny. When he published his thinking in 1859, *The Origin of Species* contained a single figure: a phylogeny. We already saw in Sect. 4.8 that phylogenies in the form of guide trees are useful for computing multiple sequence alignments. But beyond clever computing, phylogenetic trees embody biologists' thinking. In this section, we describe how to write, visualize, and traverse trees such as that shown in Fig. 5.1.

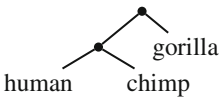
New Concepts

Name	Comment
inorder traversal	similar to preorder traversal
Newick tree format	standard tree notation
postorder traversal	similar to inorder traversal
preorder traversal	method of visiting each node of a tree once
recursive structure	trees are recursive structures

New Programs

Name	Source	Help
`genTree`	book website	`genTree -h`
`new2view`	book website	`new2view -h`
`traverseTree`	book website	`traverseTree -h`

Problem 370 Begin by making the directory `TreesOfLife` and change into it. Here is a simple phylogeny:



B. Haubold and A. Börsch-Haubold, *Bioinformatics for Evolutionary Biologists*, https://doi.org/10.1007/978-3-319-67395-0_5

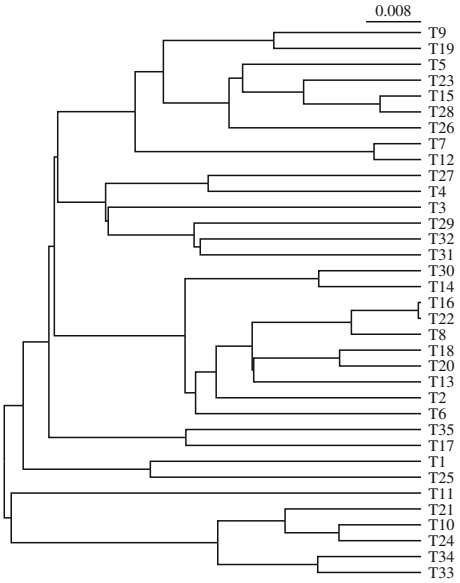
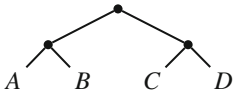


Fig. 5.1 A random tree of 35 taxa

Its textual representation is

```
((human:1,chimp:1):1,gorilla:1);
```

This notation is called the Newick format after "Newick's Lobster House" in New Hampshire, where the American Biologist Joe Felsenstein and a few colleagues adopted it on June 26, 1986. Translate the following tree to Newick and save it in `first.tree`.



Problem 371 The advantage of a standard tree notation is that programmers can write visualization software based on it. Use `new2view` to visualize `first.tree`.

Problem 372 The Newick format does not require branch lengths. Write `first.tree` without branch lengths, and save it as `second.tree`. How does `new2view` render branches without lengths?

Problem 373 In a Newick tree, any node can be either labeled or unlabeled. Remove the leaf labels from `second.tree` and label its root as "root". Save the result in `third.tree`.

Problem 374 Return to the tree with labeled leaves in `second.tree`. What happens when the labels within the pairs A, B, and C, D are switched? Does this change the biological interpretation of the tree? Save this tree as `fourth.tree`.

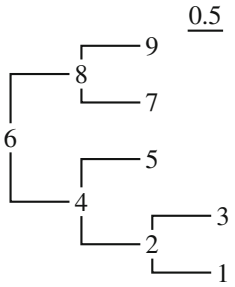
Problem 375 Use `new2view` to look at unrooted versions of all four trees generated so far. What is the difference to the rooted layout?

Problem 376 In biology, we often talk about "unrooted" trees, even though this is an oxymoron for computer scientists and mathematicians. To get a clearer understanding of what biologists mean when they refer to unrooted trees, save the code of the four trees in `fourTreesU.tree`. Then unroot them manually by editing each one such that their "root" has three children instead of the usual two. Use `new2view` to look at the unrooted trees in default layout and in rooted layout.

Problem 377 The bracket notation emphasizes that trees consist of subtrees, for example,

```
(((,),),(,));
```

It is a bit difficult to imagine what this tree looks like, but we can again use `new2view`, and this time label all nodes with their internal identifier by using the `-l` option



The subtree rooted on node 6 consists of subtrees rooted on nodes 8 and 4, and so on. This "recursive" structure leads to a method for traversing a tree, where the function `traverse` calls itself:

```
traverse(node)
   visit(node)
   traverse(leftChild(node))
   traverse(rightChild(node))
```

When we apply this procedure to node 6 in our tree, the nodes are visited in the order: 6, 4, 2, 1, 3, 5, 8, 7, and 9. Since the root is always visited first, then each one of the child nodes in turn, the procedure is called *preorder traversal*. In what order are the nodes of Fig. 5.2 visited during preorder traversal?

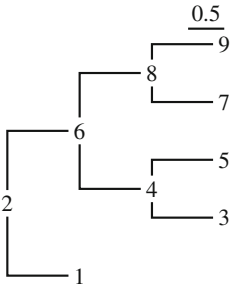


Fig. 5.2 A tree with all nodes labeled

Problem 378 Instead of preorder, a tree can also be traversed inorder by first visiting the left child, then the root in the middle, and finally the right child:

```
traverse(node)
   traverse(leftChild(node))
   visit(node)
   traverse(rightChild(node))
```

In what order does this algorithm visit nodes when applied to the root of Fig. 5.2?

Problem 379 Finally, the root can be visited last, which is called postorder traversal:

```
traverse(node)
   traverse(leftChild(node))
   traverse(rightChild(node))
   visit(node)
```

In what order does this algorithm visit nodes when applied to the root of Fig. 5.2?

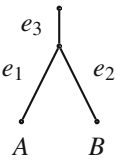
Problem 380 Instead of traversing the tree in Fig. 5.2 by hand, write its topology without any labels into the file `traverse.tree` and use `traverseTree` to traverse it in the three possible modes. Can you see why they are also called *depth first* traversals?

Problem 381 Sometimes it is useful to quickly generate a tree, for example to test software that draws trees. The program `genTree` generates random phylogenies. By default, the branches of these trees have lengths proportional to the number of mutations along the branches. Run `genTree` with default options and visualize its output. The branches in the tree do not all end at the same vertical line, which indicates the present. Can you think of a reason for this heterogeneity in the positions of the leaves?

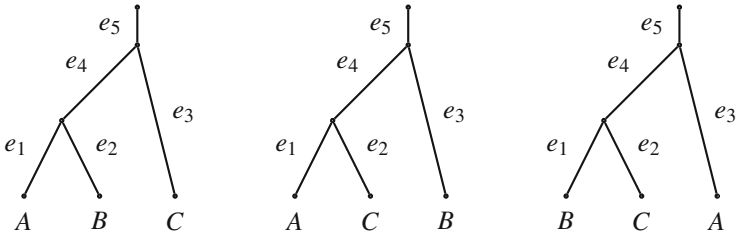
Problem 382 Generate two random phylogenies without mutations. How do they differ?

Problem 383 There are $n \times (n-1) \times ... \times 2 = n!$, that is, n-factorial, ways of ordering n objects. Write an AWK program for computing $n!$ as a function of $n = 1, 2, ..., 100$ and plot the result to get an idea of the number of possible phylogenies.

Problem 384 There are many trees that correspond to the same ordering of leaves. The number of phylogenies with n leaves is, therefore, probably larger than $n!$. Exactly how large, can be computed based on the following consideration [17, p. 20ff]: Two taxa are connected by a single rooted tree:



The third taxon, C can be added to any of the three edges e_1, e_2, and e_3, giving three trees:



The fourth taxon can be added to any one of the five edges $e_1, e_2, ..., e_5$, yielding $3 \times 5 = 15$ rooted trees. In general, the number of rooted, bifurcating trees for n taxa is

$$3 \times 5 \times ... \times (2n - 3).$$

Write the program `numTrees.awk` to compute the number of rooted phylogenies as a function of n. Plot the result for $n = 2, 3, ..., 100$ and compare it to $n!$.

Problem 385 Generate another random tree with no mutations, but this time display branch lengths that are proportional to the speciation time.

Problem 386 When reconstructing phylogenies, biologists often talk about a *molecular clock*. Under the molecular clock model, mutations occur with constant rate along all branches of a phylogeny. Trees generated by `genTree` have this property. But in contrast to the clocks of everyday life, the molecular clock is stochastic. To emphasize the difference, compare two random trees with default mutations, one with branch lengths proportional to mutations (molecular clock), the other with branch lengths proportional to time (usual clock). Use the same seed for the random number generator to make the trees comparable.

5.2 Rooted Phylogeny

How can we calculate phylogenies from data rather than just simulate them? To construct a rooted phylogeny, start from a distance matrix like that shown in the top panel of Fig. 5.3a. Cluster the most similar organisms, A and B, such that the distance between A and its parent is half that between A and B (Fig. 5.3a, bottom panel). Then recompute the distance matrix with the new cluster, (A, B). Distances between groups of taxa are the average of the distances between the members of the groups. For example, the distance between C and (A, B) is $(4 + 4)/2 = 4$. This averaging of distances gives the method its name: "Unweighted Pair-Group Method using an Arithmetic average," or UPGMA [17, 46]. The most similar pair of taxa is chosen from the new matrix and clustered again (Fig. 5.3b). The cycle of matrix adjustment and clustering is repeated one more time to yield the final rooted phylogeny (Fig. 5.3c).

New Concept

Name	Comment
Unweighted Pair-Group Method using an Arithmetic average	clustering method

New Programs

Name	Source	Help
`andi`	book website	`-h`
`clustDist`	book website	`clustDist -h`
`dnaDist`	book website	`dnaDist -h`
`gd`	book website	`gd -h`
`lscpu`	system	`man lscpu`

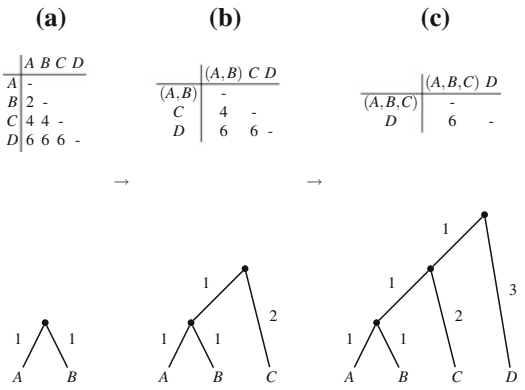
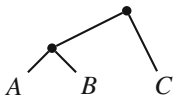


Fig. 5.3 Construction of a rooted phylogeny in three steps (**a**–**c**) using the "Unweighted Pair-Group Method using an Arithmetic average," UPGMA

Problem 387 Use UPGMA, paper, and pencil to draw the phylogeny implied by the distance matrix

	A	B	C	D
A	-			
B	6	-		
C	2	6	-	
D	6	4	6	-

Problem 388 There might be discrepancies between a UPGMA tree and the distances from which it was constructed. Perfect agreement is also known as "ultrametricity." Ultrametric distances fulfill the criterion that for three distances between taxa A, B, C we can find a labeling such that $d_{A,B} \leq d_{A,C} = d_{B,C}$. This "three point criterion" ensures that all leaves fall on a horizontal line, the present:



An ultrametric tree implies not only a molecular clock, but also that this stochastic clock behaves like a conventional clock. These strong assumptions underlie the simplicity of UPGMA. Are the distances in Problem 387 ultrametric?

Problem 389 Let's construct a phylogeny from some real sequence data. The file `hominidae.fasta` contains the extant genera of the *Hominidae*, also known as "great apes".

Problem 390 Look up the trivial names of these organisms in the taxonomy database on the NCBI website at

`http://www.ncbi.nlm.nih.gov/taxonomy`

Problem 391 We now recapitulate in detail how to get from sequence data to a tree. The sequences in `hominidae.fasta` are already aligned. Use `gd`, `getSeq`, and `cutSeq` to extract the first ten polymorphic positions in the alignment. From this, count by hand the pairwise differences between these taxa.

Problem 392 Use UPGMA, pencil, and paper to construct the phylogeny from the distances. Write the final tree in Newick format. Then enter the distances in the file `test.dist` using the format

```
4
name1 d_11 d_12 d_13 d_14
name2 d_21 d_22 d_23 d_24
name3 d_31 d_32 d_33 d_34
name4 d_41 d_42 d_43 d_44
```

Apply `clustDist` to check the intermediate matrices and final tree. Visualize the tree with `new2view`.

Problem 393 Use the program `dnaDist` to compute the pairwise distances from `hominidae.fasta`. Are the distances ultrametric?

Problem 394 Use `clustDist` with UPGMA to compute the *Hominidae* phylogeny from `hominidae.dist`. Pipe the result through `new2view` to visualize the tree.

Problem 395 Next, we compute the phylogeny of primates from their mitochondrial genome sequences. How many sequences are contained in `primates.fasta` and what is the range of their lengths?

Problem 396 The program `andi` [23] can quickly compute distances between genomes. Use `andi` to compute the pairwise distances between the primate mitochondria. `Andi` is a "multi-threaded" program, which means it can use more than one CPU at a time to carry out its computation. To find out the number of threads available, enter

```
lscpu
```

and look for the line starting

```
CPU(s):
```

Say, there are eight CPUs, then run the mitochondria computation with 1, 2, 4, and 8 threads and use the program `time` to measure the run time; for two threads this would be

```
time andi -t 2 primates.fasta
```

Problem 397 Cluster the distances using UPGMA and visualize the primate phylogeny with `new2view`. Do you get the same branching order for human, chimp, gorilla, and orangutan as with the *Hominidae* data set used in Problem 392?

5.3 Unrooted Phylogeny

All groups of organisms have a common ancestor, which means that all real phylogenies are rooted. The UPGMA algorithm we used in Sect. 5.2 is popular because it generates rooted phylogenies. The underlying assumption that the mutation rate ticks like a real clock along all branches fits the intuition that the leaves of a phylogeny should all be located in the present time. However, the mutation rates might vary along branches as shown in Fig. 5.4a. If we apply UPGMA to the distance matrix in Fig. 5.4b, B and C are clustered first, giving us the wrong tree. In this section, we look at the neighbor-joining algorithm [42], which overcomes this limitation of UPGMA and recovers the correct tree at the cost of losing the root node (Fig. 5.4c). However, we can reintroduce the root by placing it in the middle between the two most divergent taxa [16], a procedure known as "midpoint rooting".

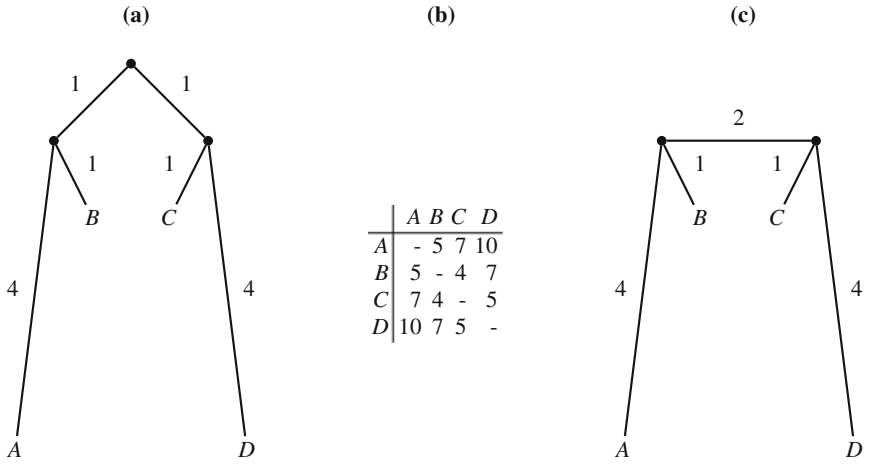


Fig. 5.4 Tree with varying mutation rates (**a**) and the corresponding distance matrix (**b**). By removing the root node from (**a**) we get (**c**)

New Concepts

Name	Comment
midpoint rooting	place root in the middle between most distant leaves
neighbor joining method	clustering method for distance data

New Program

Name	Comment	Help
`retree`	book website	menu-driven

Problem 398 Trees with varying mutation rates along their branches like in Fig. 5.4a result in distances that are called additive. Of the six distances among organisms A, B, C, and D, four cross the root of the tree, d_{AC}, d_{AD}, d_{BC}, and d_{BD}. These can be divided in two groups of two, the sums of which are identical, $d_{AC} + d_{BD} = d_{AD} + d_{BC}$. These sums are always larger than the sum of the remaining two distances, d_{AB} and d_{CD}. This yields the "four point criterion" for additive distances:

$$d_{AC} + d_{BD} = d_{AD} + d_{BC} \geq d_{AB} + d_{CD}.$$

Check manually that the distances in Fig. 5.4b are additive.

Problem 399 List three taxa from Fig. 5.4a whose distances contradict the three point criterion of ultrametricity. This posits that given a tree, any triplet of leaves can be labeled such that

$$d_{AB} \leq d_{AC} = d_{BC}.$$

Problem 400 To trace the neighbor-joining algorithm, note down the top triangle of the distance matrix in Fig. 5.4b:

	A	B	C	D	r_i
A	-	5	7	10	
B		-	4	7	
C			-	5	
D				-	

The last column, r_i, is reserved for the row sums. Compute by hand the values of r_i, remembering that the distance matrix is symmetrical.

Problem 401 In the next step of neighbor-joining we compute for each pair of taxa, i, j, the difference between its distance, d_{ij}, and the normalized sum of the corresponding row sums, r_i, r_j:

$$S_{ij} = d_{ij} - \frac{r_i + r_j}{n-2},$$

where n is the number of taxa. Compute the S_{ij}-values by hand and write them in the lower triangle of the distance matrix.

Problem 402 Instead of clustering the pair of taxa with the smallest distance as in UPGMA, neighbor-joining clusters the taxa with the smallest S_{ij}. If the new cluster is called c, the distances between c and some other cluster, k, is

$$d_{kc} = (d_{ik} + d_{jk} - d_{ij})/2.$$

The other distances are unchanged. Cluster one of the two pairs with the smallest S_{ij} and adjust the distance matrix accordingly.

Problem 403 We still need to know the lengths of the branches connecting the new cluster c and leaves i and j:

$$d_{ic} = \frac{(n-2)d_{ij} + r_i - r_j}{2(n-2)},$$

and

$$d_{jc} = \frac{(n-2)d_{ij} + r_j - r_i}{2(n-2)}.$$

Compute (by hand) the branch lengths for the new cluster.

Problem 404 To summarize, the neighbor-joining algorithm consists of four steps; given distances d_{ij},

- compute

$$r_i = \sum_j d_{ij},$$

- compute

$$S_{ij} = d_{ij} - (r_i + r_j)/(n-2),$$

- cluster pair of taxa with smallest S_{ij} in node c with

$$d_{kc} = (d_{ik} + d_{jk} - d_{ij})/2,$$

- calculate branch lengths

$$d_{ic} = \frac{(n-2)d_{ij} + r_i - r_j}{2(n-2)}$$
$$d_{jc} = \frac{(n-2)d_{ij} + r_j - r_i}{2(n-2)}$$

This procedure is repeated until there are only three clusters left, i, j, k, which is the stage we have reached in our example. These are joined to the pseudo-root r, by branches with the following lengths

$$d_{ri} = (d_{ij} + d_{ik} - d_{jk})/2$$
$$d_{rj} = (d_{ji} + d_{jk} - d_{ik})/2$$
$$d_{rk} = (d_{ki} + d_{kj} - d_{ij})/2$$

What are the lengths of the last three branches added to our example tree?

Problem 405 Create the directory `NeighborJoining` for this session and change into it. Enter our example distances (Fig. 5.4) in file `test.dist` and use `clustDist` to automatically trace the steps just completed by hand. Save the tree as `testU.tree` and draw it using `new2view`.

Problem 406 Root the example tree by placing the root in the middle of the longest path from one leaf to another. Write down the midpoint-rooted phylogeny in Newick format and save it to `testR.tree`. Then draw it with `new2view`.

Problem 407 Use the PHYLIP program `retree` to midpoint-root our unrooted example tree from Problem 405, which was saved as `testU.tree`. Save the final tree in rooted form. Does `retree` return the same result as the manual midpoint rooting?

Problem 408 Use `andi` together with `clustDist` to compute the neighbor joining tree of the mitochondrial genomes in `primates.fasta`. Midpoint-root this version of the primate phylogeny and compare it to the UPGMA tree (Problem 396). Can you spot any differences?