Appendix C
Molecular Phylogenetics

C.1 Introduction

The main goal in molecular phylogenetics is to reconstruct a phylogenetic tree from a set of protein or genomic sequences. The tree will represent the history of a set of organisms that share a common ancestor. For this purpose, sequences need to be homologous; that is, they should have evolved from a common ancestral sequence. Alignment of homologous sequences is the first step to any phylogenetic inference. For instance, let us try to align two small sequences that we suspect are homologous: $S_1 = ACTGCGAA$ and $S_2 = CCGTCTAA$. An alignment is a map between a set of strings of the same length that contain the same nucleotides and have the same order as in the original data, and could include gaps, represented by $-$. For instance, a potential alignment can be of the form:

$$S'_1 = ACTG - CGAA$$
$$S'_2 = -CCGTCTAA,$$

consisting of two strings of length 9, that differ in position 1 ($A \to -$), position 3 ($T \to C$), position 5 ($- \to T$), and position 7 ($G \to T$). Differences between two nucleotides represent mutations between the two bases, and a gap indicates a deletion or insertion of a particular nucleotide.

There are many potential alignments between a set of sequences, and we need criteria to determine which are optimal. A common approach is to consider a score for each alignment. For instance, one could count the number of differences. In the case of the alignment $\{S'_1, S'_2\}$, this would be 4. A simple score is just the number of similarities minus the number of differences, in this case $5-4 = 1$. But of course, not every change is necessarily weighted equally, indeed, there are some changes that are more likely to occur than others. The genomic material in all known organisms is polymers of nucleotides, each composed of a sugar, a nitrogenous base, and a phosphate group. The five-carbon sugar (ribose or deoxyribose) defines the type

of molecule, RNA or DNA. Bases can be divided into two types based on the chemical structure, purines (adenine and guanine) and pyrimidines (cytosine, uracil and thymine). A substitution, changing a base into another, could happen as a result of different chemical processes. For instance, one of the most common chemical processes is a spontaneous deamination (removing an amine group) of a cytosine, changing it into uracil. If not correctly repaired, that will lead to a $C \to T$ mutation in DNA. It turns out that changes within purines and within pyrimidines (transitions) are much more common than changes between purines and pyrimidines (transversions). The probabilities of these changes can be inferred experimentally, by considering data across different homologous sequences and estimating likelihoods. Similarly, one can evaluate the effect of small insertions and deletions by working with protein sequences.

A popular type of score assigns to every alignment a linear score that adds the weights for every substitution and adds a gap penalty for indels. There are several classic dynamic programming algorithms, like Needleman-Wunsch [379] and Smith-Waterman [468], for optimal alignment between a pair of sequences. When dealing with multiple sequences one could consider adding the pairwise scores, but this problem has been shown to be NP-complete [521]. In practice, heuristics are used.

Once the sequences are aligned one can start inferring trees (see Figure C.1). There are several kinds of trees to consider. In some cases, trees can be rooted, e.g., if they have a node that represents the common ancestor to all the analyzed taxa, which gives information about the temporal order of nodes in the tree. Alternatively, unrooted trees display the evolutionary relationships among taxa, without any ancestral root. The root is usually, but not always, determined by using an outgroup taxon that falls outside the group of taxa of interest. In general, when this information is not available or not used, one can construct an unrooted tree. For m sequences, an unrooted tree has m external nodes (or leaves) each of which is labeled with a different sequence. Internal nodes can be labeled by inferred sequences that represent the genomic information of common ancestors of two of the contiguous nodes. With m labeled leaves it is easy to see that the number of tree topologies is $(2m-5)!! = (2m-5)(2m-7)(2m-9)\cdots 1$. That is, for $m = 3$ branches, there is only one labeled unrooted tree, for $m = 4$ there are 3, for $m = 5$ there are 15, etc.

Edges could be weighted by a positive number that is associated to the number of changes between nodes (for instance, it could be the number of changes, or a weighted version of this, assigning each change a different weight). The space of potential trees is enormous and some criterion is needed to find the optimal tree. There are many methods that have been proposed. Here we explain some of the most popular ones. Many methods work directly with the aligned sequences,

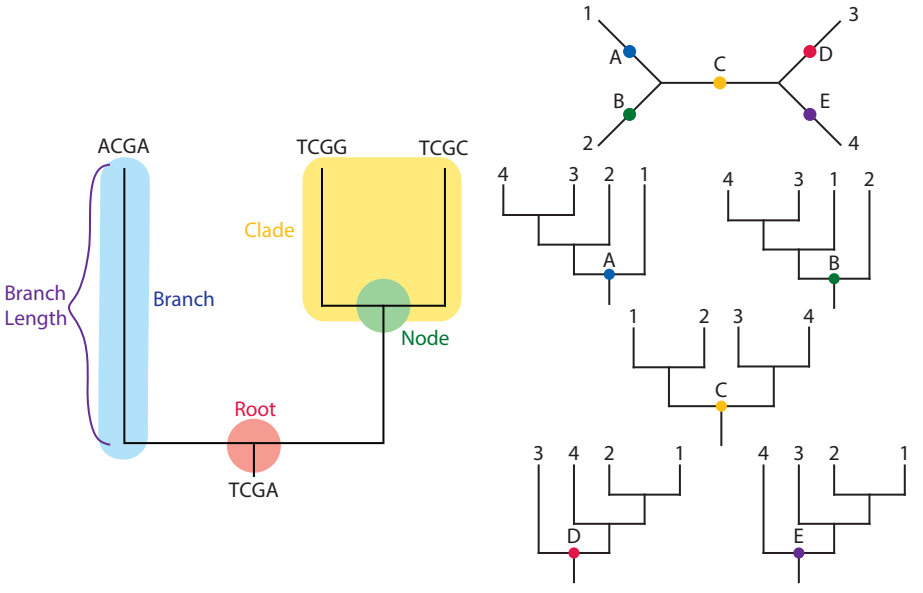

Figure C.1 Notation on a tree. There are $2m - 3$ possible ways of constructing a rooted tree from an unrooted tree of m leaves.

referred to as alignment-based methods. In all these methods the strategy is to minimize a criterion (a likelihood, for instance) by exploring a large number of trees. As previously described, the number of trees increases as $m!$, making it unfeasible to explore all possible tree topologies. Different heuristics are used to explore a reasonable set of topologies.

A second type of approach computes a distance metric from the data, and works directly using the distance data. These methods have the advantage that they scale polynomially in the number of sequences and genome length, and so one can easily work with thousands of sequences. On the other hand, the results are sometimes less biologically plausible than likelihood-based methods.

C.2 Sequence Based Methods

C.2.1 Parsimony

The parsimony principle is the preference for the simplest explanation of some facts. In the case of phylogenetic reconstruction, parsimony selects the tree with the minimum number of changes required to explain an alignment. Given a tree T, and a set of sequences S from a multiple alignment attached to the leaves (external vertices), we can assign hypothetical sequences H to internal nodes. We can compute for each edge a distance using the Hamming metric or a weighted version of

it. Adding the results for all edges, we obtain the parsimony score $P(T, H \mid S)$. The (large) maximum parsimony tree is the tree T and hypothetical internal sequences assignment H that minimizes $P(T, H \mid S)$. The task of computing the best H, given a particular tree topology T, from some external data S, is called the small parsimony problem and can be computed in polynomial time, using for example, some classical algorithms from phylogenetics such as the Fitch algorithm [176]. The large parsimony problem requires going through all possible topologies and for each one computing the optimal H. The output is the topology that minimizes the parsimony score. This problem has been shown to be NP-complete [180].

There are, however, some heuristic methods to explore possible solutions, without, of course, any guarantee that they will be the optimal solution. Branch and bound methods start with a subset of three sequences S_3 from the original data S. In this case, there is a unique tree and a maximum parsimony solution can easily be found. Now, we can select a sequence from S that is not in S_3 and attach a new leaf to any of the three leaves. There are three different possibilities to consider. Now we can select another sequence from S that was not previously considered, and repeat the procedure, but now there are five potential edges. In this way, one can construct iteratively all possible trees in a hierarchical fashion (a tree of trees). Now in each of these iterations we can compute the parsimony score, which will always increase when considering more branches. In the branch and bound method, one proceeds iteratively and selects the best tree in each iteration and only considers subsequent iterations along those particular branches in the tree of trees (see Figure C.2).

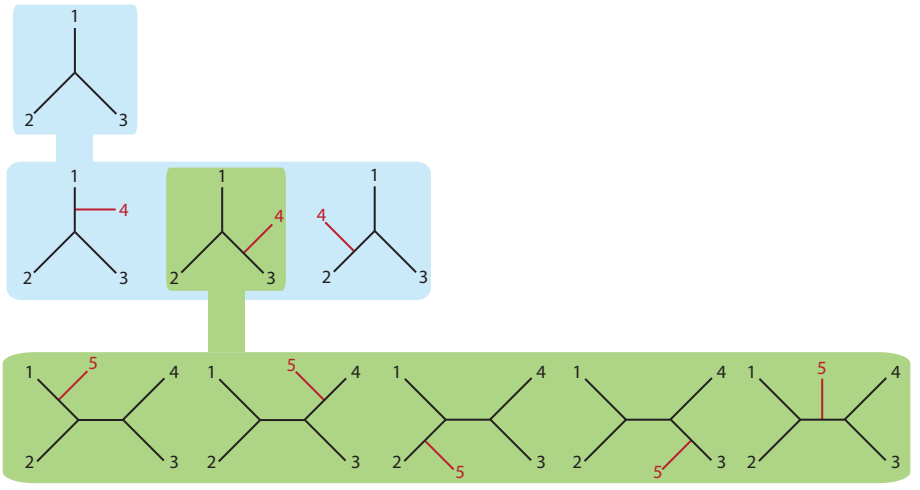

Figure C.2 Branch and bound.

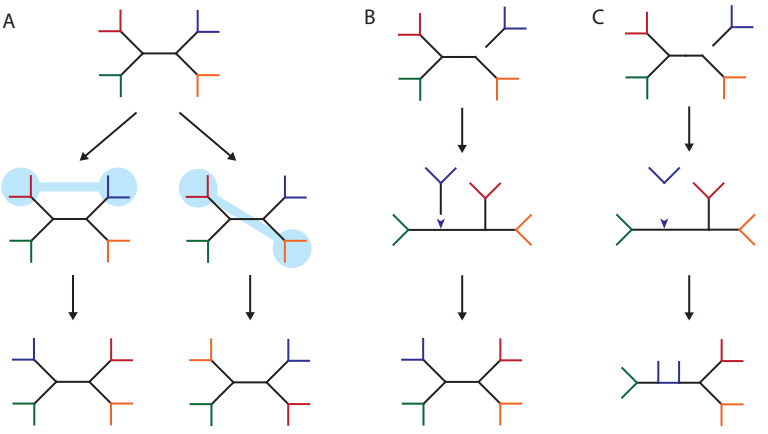

Figure C.3 As the number of taxa increases, the number of potential trees becomes extremely large. There are techniques to explore the space of potential trees by branch swapping strategies: (A) nearest neighbor interchange (NNI), (B) pruning and regrafting (SPR), and (C) tree bisection and reconnection (TBR).

A second type of strategy is based on the idea of swapping branches. Here we describe three major strategies, nearest neighbor interchange (NNI), pruning and regrafting (SPR), and tree bisection and reconnection (TBR). Any internal edge on a bifurcating tree has four neighbor subtrees, two attached to one vertex and two to the other. NNI is an operation that exchanges a tree of one of the neighbor vertices with another one (see Figure C.3). There are several implementations of these methods, but in the simplest version, a NNI procedure is accepted if it reduces the parsimony score. Pruning and regrafting is an idea along the same lines where a subtree is cut and regrafted in one of the edges, creating a new node (see Figure C.3). Tree bisection and reconnection (TBR) selects an edge and removes it completely from the larger tree, generating two smaller subtrees. Then one edge from each subtree is selected and two new nodes are introduced in each of the edges and finally joined by a new edge (see Figure C.3). NNI, SPR and TBR are operations in the space of trees, and different algorithms can be implemented to make sure that local minima are avoided. These heuristic approaches are commonly used in other phylogenetic techniques, like the likelihood methods we discuss next.

C.2.2 Likelihood Methods

Likelihood methods optimize a likelihood function $\Psi(S \mid T, L, M)$, that calculates the probability of obtaining the observed sequence data S given a tree T with branch lengths L and a model M that determines the probability of a particular

mutation to occur. The advantage of probabilistic models is that they incorporate realistic assumptions based on empirical data and they can be used to estimate parameters. For instance, it is easy to incorporate the probability of back mutations, different rates for transitions and transversions, and so forth, and estimate the likelihood for those parameters. For instance, one can compute the probability of a base to change after some particular time l (associated to a branch of length l in the tree) and at a constant mutation rate μ to be $p(l) = \frac{3}{4}(1 - e^{-\frac{4}{3}\mu l})$. The formula can be easily adapted to allow different mutation rates across different bases, and even different rates in different branches and genomic positions.

Most likelihood methods assume that the likelihood for a sequence is the product of likelihoods for all positions (independence among sites): $\Psi(S \mid T, L, M) = \prod \Psi(s_i \mid T, L, M)$, where s_i is the alignment data for genomic position i. For each edge on the tree and site i, one can associate characters and compute the probability for change ($p(l)$) or staying the same ($1 - p(l)$), where l is the length associated to the edge. For a given tree T with edge length L, the likelihood for the observed data S can be computed using the Felsenstein algorithm [173, 174].

However, the full solution to the likelihood problem requires that all different tree topologies and branch lengths are explored, and as such finding the maximum likelihood tree is NP-hard [115]. There are, however, good approximations that adjust tree topology and branch lengths simultaneously. For instance, in [216] a hill-climbing algorithm is proposed, that starts from a fast distance-based method and modifies this tree to improve its likelihood at each iteration.

C.2.3 Bayesian Methods

Bayesian methods are based on a similar idea to likelihood methods, but instead of estimating the probability of the observed data S given a tree, they estimate the posterior probability $P(T, L \mid S)$ of a weighted tree (T, L) given the observed data S [257]. The basic object here is a distribution on the space of all potential trees. The whole distribution cannot be estimated analytically, but it is possible to sample the distribution. Most of the implementations are based on variations of Markov chain Monte Carlo (MCMC) approaches. The main idea is simple: one can take a tree T_i, modify it to obtain a new tree T', and compute the ratio between the posterior probabilities:

$$R = \frac{P(T', L' \mid S)}{P(T_i, L_i \mid S)}.$$

Using Bayes theorem, this can be shown to be equivalent to:

$$R = \frac{P(S \mid T', L')P(T', L')}{P(S \mid T_i, L_i)P(T, L)},$$

where $P(T, L)$ is the prior probability of observing a weighted tree (T, L).

Based on the posterior ratio R, we can decide whether or not to accept the modification T'. If so, we take the accepted tree as new starting point and we continue the operation. This procedure generates a random walk on the space of trees sampling a distribution that approximates $P(T, L \mid S)$. One of the most popular implementations is the Metropolis-Hastings algorithm [234]: we accept the new tree with a probability $\min(1, R)$. If so, we define $(T_{i+1}, L_{i+1}) = (T', L')$, and now we iterate. Trees with higher posterior probability will tend to be sampled more frequently. In the end, the result is a set of trees with high posterior probabilities; this allows us to account for uncertainty. This set of trees can then be summarized in a consensus tree if necessary.

Different implementations have been carried out, involving different perturbations of the trees (for instance, the ones discussed above, NNI, SPR, TRD), different evolutionary models (constant and non-constant rates, different rates for different mutation types), variations on initial location in tree space, and many others.

In some simple cases, Bayesian techniques have been reported to be more accurate (the output tree topology displays evolutionary relationships closer to reality) than parsimony or distance based methods, especially when analyzing highly divergent taxa [256].

C.3 Distance Based Methods

Distance based methods reduce the complexity of the inference tree dramatically by considering only the distances between the sequences S; the problem is then to reconstruct a weighted tree. As not all the information regarding particular positions is used, there is no attempt to reconstruct sequences attached to internal nodes (ancestral states).

Several distance functions can be constructed from a set of sequences. The simplest one is the Hamming distance d_H, which is the fraction of bases that differ between two sequences. The Hamming distance considers all substitutions equally likely and it does not consider the probability that for long times there could be mutations in already mutated positions. A natural way to assign a distance that takes into account the possibility of back mutations is estimating μt by the fraction of bases changed after a time t if the mutation rate is μ, which we can compute by inverting $p(l) = \frac{3}{4}(1 - e^{-\frac{4}{3}\mu l})$. This defines the Jukes-Cantor distance:

$$d_{JC} = -\frac{3}{4}\log(1 - \frac{4}{3}d_H).$$

The Jukes-Cantor distance is just a transformation of the unit interval into the positive numbers. When $d_H \to 0$, $d_{JC} \sim d_H$, i.e. when they are near zero both distances are similar. But when two random sequences with four bases are aligned and there is an equal number of the four bases, only one quarter of the bases will be the same. Then $d_H \to 3/4$ and $d_{JC} \to \infty$. More complicated models incorporate different rates of transitions and transversions, and different frequencies of nucleotides. For instance the K80 model [298] considers that all bases are equally frequent but that there are different rates for transitions and transversions. If p is the fraction of transitions (like the Hamming distance but only counting transitions) and q the fraction of transversions, the K80 distance is defined as:

$$d_{K80} = -\frac{1}{2}\log(1-2p-q) - \frac{1}{4}\log(1-2q).$$

If the frequency of the four nucleotides is different from 25%, Jukes-Cantor can be modified to the Tajima-Nei distance:

$$d_{TN84} = -\beta \log(1 - \frac{d_H}{\beta})$$

where $\beta = \sum_i f_i^2$, and f_i is the frequency of the nucleotide i. Further generalizations include different rates for each mutation and different frequencies per nucleotide.

Now assume the whole data is reduced to a distance metric. How can we infer a weighted tree from this metric? One of the oldest methods is the least squares method, proposed in 1967 by Fitch and Margoliash [177]. The basic idea is to find the weighted tree that minimizes the sum of the squares of differences between the distances between two sequences and the sequence in the tree (sum of branches connecting the two leaves d_{ij}^T, also called patristic distance):

$$s = \sum_{i,j} (d_{ij} - d_{ij}^T)^2.$$

The method requires exploration of all topologies; unsurprisingly, the method is NP-complete [135].

Agglomerative or clustering methods are usually much more convenient and faster, generating a solution in polynomial time. The main idea of these methods is to start from the pair of closest sequences that are linked. Then eliminate the columns and rows from these sequences and introduce a new one where distances are computed using a particular rule. Now, the distance matrix contains one fewer column and row. By iterations, one quickly arrives at a single element. The most popular algorithm of this type is neighbor-joining (NJ) [442], which proceeds as follows.

1. First calculate the matrix $T_{ij} = (m-2)d_{ij} - \sum_k d_{ik} - \sum_k d_{jk}$, where m is the number of sequences.
2. Find the leaves with lowest T_{ij}, i and j.
3. Define a new leaf k, and join i and j with k.
4. Compute distances to the new node from the leaves being joined:

$$d_{ik} = \frac{1}{2}d_{ij} + \frac{1}{2(m-2)}\left(\sum_s (d_{is} - d_{js})\right).$$

5. Compute distances to the other leaves from the new node k,

$$d_{ks} = \frac{(d_{is} + d_{js} - d_{ij})}{2}$$

6. Replace the joined neighbors with a new node, using the recomputed distance. And restart the algorithm.

The NJ algorithm generates a tree in polynomial time, and generates the right tree if the distance matrix satisfies the four point condition. However, in more general cases, it could lead to strange results such as negative branch lengths.

C.4 Phylogenetic Networks

As we have seen in Chapter 5, phylogenetic trees fail to capture reticulate events including recombinations and reassortments in viruses, horizontal gene transfer in bacteria, and meiotic recombination and species hybridization in eukaryotes. Phylogenetic networks aim to represent these events as a generalization of a tree with external nodes representing the observed data and a graph, with cycles representing incompatibilities. Like phylogenetic trees, phylogenetic networks can be constructed from sequences or distances. We will briefly mention a few methods that we have discussed in this book.

A common approach to capture reticulate events is to use split networks. Split networks represent incompatible splits, but the interpretation in terms of biological processes that could generate these splits is obscure. For example, it not easy to tell if an incompatible tree was generated by recombination, back mutations, or a horizontal gene transfer event. Nor can one determine how many events generated the incompatibility, whether just one reticulate event is enough to generate an incompatibility, how the number of incompatibilities scale with recombination rates, etc. The lack of interpretability of split representations constitutes a serious obstacle to a wider adoption of these representations for the biological community.

Clearer biological interpretations arise from a reticulate network, where each loop is supposed to represent a reticulate event (recombination, gene transfer,

reassortment, etc.). Ancestral recombinant graphs (ARGs), for instance, introduce nodes that correspond to potential recombination.

C.4.1 Split Networks

Let X be a set. Then a split $S_1 \mid S_2$ is any partition of X into two non-empty sets:

- $S_1 \neq \emptyset$ and $S_2 \neq \emptyset$,
- $S_1 \cup S_2 = S$,
- $S_1 \cap S_2 = \emptyset$.

A weighted set of splits (S, L) is a collection of splits $\{S_i\}$ together with a set of weights $\{l_i \geq 0\}$. In a tree, each edge provides a split: if we cut the edge, the data splits into two non-overlapping subsets. More interestingly, trees provide a set of splits $\{S_i\}$ that satisfy an extra condition – they are compatible. Two splits $S_i = X_i \mid Y_i$ and $S_j = X_j \mid Y_j$ are compatible if and only if one of the intersections $X_i \cap X_j$, $X_i \cap Y_j$, $Y_i \cap X_j$, or $Y_i \cap Y_j$ is empty. A set of splits S is compatible if all possible pairs are compatible.

Trees generate compatible sets of splits, and compatible sets of splits can be represented by trees. A weighted tree (T, L) is in this way equivalent to a weighted set of compatible splits (S, L). But splits that are not compatible generalize the notion of a tree. A representation of incompatible splits is through a split network. A split network represents a set (S, L) where each element of S labels a single node, and each edge is labeled by splits, in such a way that removing the edges corresponding to a split partitions the graph into two, where labeled nodes are split correspondingly. In a split network one can use one or more edges to represent a split, in such a way that the deletion of such edges generates the two elements of the split (see Figure C.4).

C.4.2 Sequence Based Methods

We will briefly mention two sequence based network techniques that we described in Section 5.10: median networks and ancestral recombinant graphs (ARGs).

Median networks. Median networks take as input a set of aligned sequences S, that we will assume have letters 0 and 1. First a simplification of S is performed by taking a condensed representation that discards positions that are the same in all sequences, and taking only one representative position for every set of positions that displays the same pattern. Each of the representatives is assigned a weight β_i corresponding to the number of positions that are in the same class. Let us call the set of representatives with weights the condensed representation, S'. The median operation takes any three binary sequences and computes another sequence that for each character takes the median (the most common character in that position). For

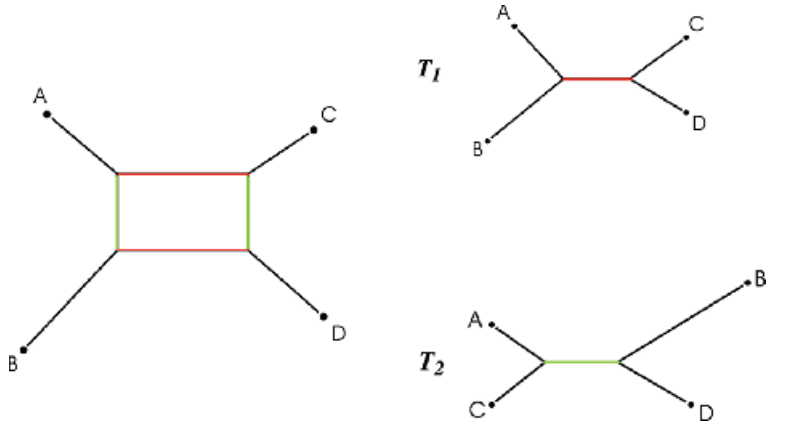

Figure C.4 A split network captures incompatible splits. Each taxon or sequence is associated to a vertex. Each edge on the network is labeled by a a split, in such a way that cutting the edges cuts the network into two, separating the labeled vertices corresponding to the split.

instance, the median of 0000, 1100, and 0111 is 0100. Adding the median sequence to the set and iterating the procedure until no new sequences are generated gives the median closure. A network, the median network or Buneman graph, can be constructed by taking as many nodes as binary sequences in the condensed representation S' and edges connecting them if they differ by only one character. The median network has nice properties as it is made of cubes of different dimensions and it contains all trees with minimal parsimony scores. But typically the number of nodes generated by the median operation is extremely large, and the biological interpretation is extremely obscure. The reduced median (RM) network algorithm and median-joining algorithm [31, 33] selects a subnetwork in the median network, reducing significantly the complexity of the network. Generalizations to more than two states sequences are called quasi-median networks.

Ancestral recombinant graphs (ARGs). Ancestral recombinant graphs constitute the most interpretable of all phylogenetic networks. An ARG provides a potential reconstruction of the history that gave rise to the data S through a series of mutations and recombinations. For a full explanation of ARGs and extensions using topological data analysis we refer the reader to Section 5.10.

C.4.3 Distance Based Methods

In distance based methods, we compute a distance between a set of sequences S, and we work exclusively with the distance matrix. These methods, as with phylogenetic trees, are fast and easily implementable, with the caveat that the interpretation of cycles in the network is obscure.

Split decomposition. In split decomposition, one takes advantage of the unique decomposition of a finite metric space into a set of splits using the Bandelt and Dress theorem [30]. The main idea of the Bandelt-Dress decomposition is that a finite metric space can be decomposed into a sum of independent metrics associated to weighted splits plus a remnant. The weight of each split $S_1|S_2$, called the isolation index, can be computed as follows:

$$\alpha_S = \frac{1}{2} \min_{\substack{i_1,j_1 \in S_1 \\ i_2,j_2 \in S_2}} (\max(d_{i_1,j_1} + d_{i_2,j_2}, d_{i_1,i_2} + d_{j_1,j_2}, d_{i_1,j_2} + d_{i_2,j_1}) - d_{i_1,j_1} - d_{i_2,j_2}).$$

For every split S one can define a split metric d_S to be 0 if two elements are in the same split and 1 if not. The Bandelt-Dress result decomposes the original metric:

$$d = \sum_S \alpha_S d_S + r.$$

The simplest example of a residue metric corresponds to 5 points with distances derived from a complete bipartite graph $K_{2,3}$. For s sequences, the approach provides at most $\binom{s}{2}$ non-zero weight splits. Remember that a tree is a set of compatible $2s-3$ splits: if the set (S, L) is compatible then there is a single tree with edges corresponding to compatible splits and edge weights corresponding to split weights. In particular, finite metric spaces that satisfy the four point condition correspond to totally decomposable metrics and compatible splits, corresponding to the underlying tree. Focusing on the non-remnant part, this construction allows finite metric spaces to be mapped to weighted splits that can be represented by a split network. But in the case of non-compatible splits the split decomposition generalizes this result.

Neighbor-net (NN). Developed in 2004 by David Bryant and Vincent Moulton [75], this is an agglomerative distance based method that generalizes the neighbor-joining algorithm we discussed before. Given a finite metric space, NN constructs a collection of weighted splits and then represents the results using a split graph. Like NJ, this is an agglomerative method that starts by selecting pairs of nodes, but instead of replacing them immediately by a new node, it waits until it is paired a second time. Then the three linked nodes become two nodes and the distance matrix is reduced (see Figure C.5). The procedure continues until the number of nodes is reduced to two or three. Being an agglomerative distance based method, the speed and throughput is very high, similar to NJ. Like other split networks, the main problem is the interpretability of the results. A nice mathematical description of the NN algorithm in terms of some discrete metric spaces, called circular decomposable metrics, can be found in [329].

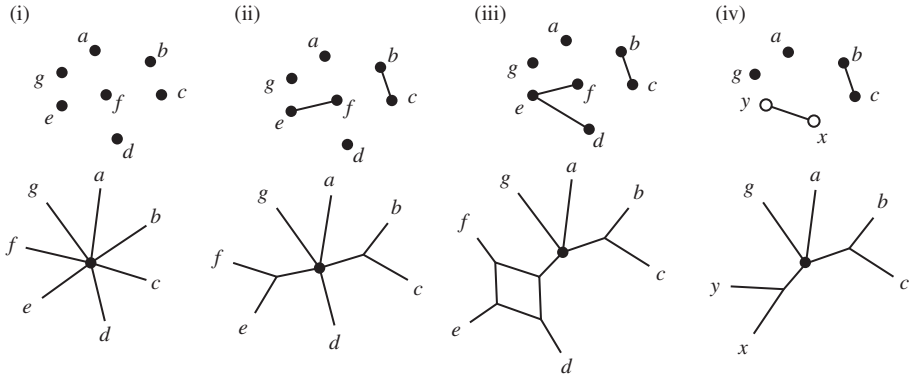

Figure C.5 This figure illustrates how the neighbor-net algorithm works. It is an agglomerative method that takes distance matrices as input. In (i) each node represents a single sequence. One first looks for closest neighbors; in (ii) the closest to e is f and the closest to b is c. Other neighbors are identified in (iii); e has as neighbors f and d. Two incompatible splits $ef|abcdg$ and $de|acdfg$ are represented and d, e, and f are substituted by new nodes x and y. Source: [75]. Bryant, David, and Vincent Moulton. "Neighbor-net: an agglomerative method for the construction of phylogenetic networks." Molecular Biology and Evolution, 2004, 21.2: 255–265, by permission of Oxford University Press.

C.5 Suggestions for Further Reading

There are excellent books and reviews on phylogenetics for the reader who wants to dive into this topic.

- *Inferring Phylogenies*, by J. Felsenstein [174] is a complete and clear exposition on different approaches to phylogenetic trees which is highly recommended.
- *Molecular Evolution and Phylogenetics*, by Masatoshi Nei and Sudhir Kumar [380] is a nice didactical overview on phylogenetic methods.
- *The Phylogenetic Handbook: A Practical Approach to Phylogenetic Analyses and Hypothesis Testing*, by Philippe Lemey, Marco Salemi and A. M. Vandamme [443].
- *ReCombinatorics: The Algorithmics of Ancestral Recombination Graphs and Explicit Phylogenetic Networks*, by Dan Gusfield [220] is highly recommended to learn more about biologically interpretable phylogenetic networks.
- *Phylogenetic Networks*, by Daniel Huson, Regula Rupp and Celine Scornavacca [262] is a nice clear survey on methods for inference of phylogenetic networks.
- *Basic Phylogenetic Combinatorics*, by Andreas Dress and colleagues [151] is a nice mathematical introduction to the relationship between finite metric spaces, split systems, and systems of quartets.

C.6 Data and Software

There is a large number of software programs for phylogenetic inference. Here is a very incomplete list of some of the most commonly used.

- PHYLIP: phylogenetic tree inference package by Felsenstein incorporating now classical methods, such as maximum parsimony, distance based algorithms, and maximum likelihood. It can be found at `http://evolution.genetics.washington.edu/phylip/general.html`.
- PhyML [217] and RaxML [482]: two of the most commonly used likelihood algorithms.
- Bayesian Evolutionary Analysis Sampling Trees (BEAST) [153]: a very commonly used Bayesian method for tree inference and parameter estimation.
- MrBayes: Bayesian posterior probability estimation for phylogenetic trees [436].
- SplitsTree [260]: a very wide platform that provides a wide range of phylogenetic tree and network inference methods, including median networks, parsimony splits, spectral analysis, split decomposition, and neighbor-net.
- Dendroscope [264]: provides a platform for visualizing trees and networks.