Notazioni per split

$$\{A, B\}, \qquad A \mid B$$

$$\{\{t, u\}, \{v, w\}\}, \qquad \{t, u\} \mid \{v, w\}, \qquad t, u \mid v, w, \qquad tu \mid vw$$

# *Split decomposition: A new and useful approach to phylogenetic analysis of distance data*

> Phylogenetic analysis of molecular sequence data often is carried out by first calculating pairwise similarity coefficients, converting these into evolutionary distances, and finally applying some distance-matrix method in order to estimate an unrooted phylogenetic tree. Goodness-of-fit would be judged by comparing the evolutionary distances with the additive distances read off the estimated tree. So, data are fit to a best (or at least, near-optimal) tree, whether or not they bear any resemblance with additive tree data. In practice, one tries to avoid methodological artifacts by applying different tree approximation methods (some operating on sequence data, others using derived distances) and then putting up with a strict consensus tree. Still, one may fall into the trap of systematic error when the methods are subject to the same bias and all disguise true phylogenetic relationships.

pairwise similarity coefficients ->
evolutionary distance ->
distance-matrix method ->
phylogenetic tree

judge result by comparing evolutionary distance with additive distance of the estimated tree

data are “fitted” to a tree (regardless of how tree-like they are)

to avoid bias of the method, use different tree approximation methods (e.g. some on sequence data, some on derived distances), then strict consensus tree

still subject to systematic error if all methods share the same bias

> We therefore propose to accompany any phylogenetic analysis by a nonapproximative method as well that allows for conflicting alternative groupings (to some extent) and hence is able to detect some of those distinctive minor features in distance data which are dominated by others and not supported by estimated trees. This goal can be achieved by *split decomposition*, developed by Bandelt and Dress (1992), which may be regarded as a kind of factor analysis for distance matrices. It decomposes any dissimilarity matrix $d$ into a number of “binary factors,” described as “splits” weighted by “isolation indices,” plus a residual indecomposable term (here interpreted as noise). For phylogenetic analysis split decomposition serves two purposes: (a) to exhibit tentative phylogenetic relationships even when they are overridden by parallel events, and (b) to detect groupings brought about by pronounced convergence or systematic error.

split decomposition allows for conflicting alternative groupings

purposes for phylogenetic analysis:

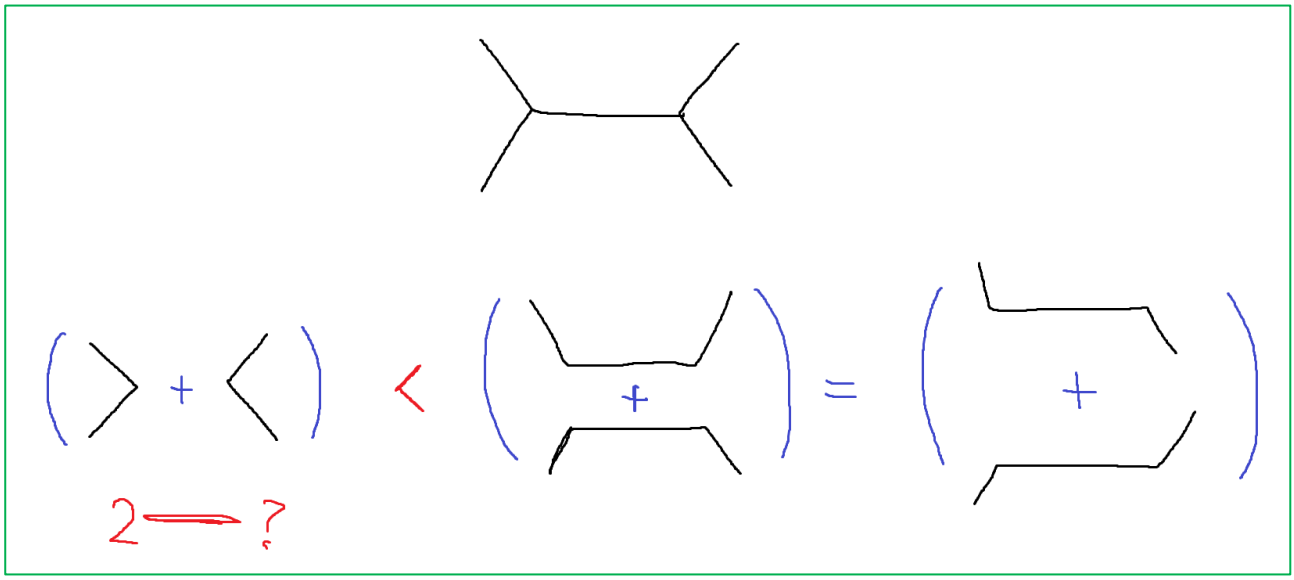
- exhibit tentative phylogenetic relationships, even when overridden by parallel events
- detect groupings caused by convergence or systematic error

The interpretation of residue as noise/error is questionable: if $d$ is non-negative, then also the residue is non-negative. We may expect that an error term should have both positive and negative elements.
Actually, it is reasonable to expect even more negative elements, since evolutionary distances underestimate the number of mutations occurred (even when using a correction model, we cannot be sure to have an estimate from above of the true distance).

As to point (a), assume a phyletic line separates two pairs of taxa 1, 2 and 3, 4; then with respect to phyletic distance $p$ the sum $p_{12} + p_{34}$ (of distances between 1 and 2, 3 and 4) is smaller than $p_{13} + p_{24} = p_{14} + p_{23}$ ("additivity" alias "4-point condition"). Evolutionary distances $d$, presenting only an estimate of true phyletic relationships, normally do not even respect the ordering $d_{12} + d_{34} < d_{13} + d_{24}$ and $d_{12} + d_{34} < d_{14} + d_{23}$, but one could hope that at least $d_{12} + d_{34}$ is not the largest of the three sums.

Given this as a working hypothesis, we would then expect such a pattern to be observed whenever the two taxa 1 and 2 are chosen from a group $\mathcal{J}$ which is separated from its complementary group $\mathcal{K}$ by a phyletic line, while taxa 3 and 4 are chosen from the complement $\mathcal{K}$. Consequently, any complementary pair $\mathcal{J}$, $\mathcal{K}$ satisfying this (comparatively weak) condition will be called a $d$-split.

To any such $d$-split one can, moreover, associate a positive weight, the *isolation index* (see Eq. (2) below), which in the case of additive distances would yield the length of the corresponding branch in the representing tree. However, there may be more $d$-splits than those supported by true phylogenetic relationships. These, typically exhibiting a low isolation index, often reflect traits of penetrating parallelism.



phyletic distance $p$ satisfies the four-point condition:

$$p_{12} + p_{34} < p_{13} + p_{24} = p_{14} + p_{23}$$

evolutionary distance $d$,
being only an estimate of true phylogenetic relationships,
normally does not even respect

$$d_{12} + d_{23} < \begin{matrix} d_{13} + d_{24} \\ d_{14} + d_{23} \end{matrix}$$

but we can hope that $d_{12} + d_{23}$ is not the largest among the three

$$d_{12} + d_{23} < \max\{d_{13} + d_{24}, d_{14} + d_{23}\}$$

we would expect this condition to hold for every two couples
chosen from two groups separated by a phyletic line;
a split that satisfies this condition is a $d$-split
(notice that this equivalent to saying that the isolation index is positive)

for additive distances, the isolation index represents the length of the edge* in the associated tree

there may be more $d$-splits than those supported by true phylogenetic relationships; typically they exhibit a low isolation index, and often reflect traits of parallelism

* also called branch or link

To illustrate point (b), imagine that an observed distance matrix $d$ is the sum $d = p + e$ of a matrix $p$ of linearly scaled phyletic distances plus an error term $e$ such that $e$ itself happens to be realized by some tree different from the one representing $p$. Then the $d$-splits would consist exactly of all splits which are either $p$-splits or $e$-splits or both (with isolation indices of $p$ and $e$ adding up to the indices for $d$). Which of the splits belong to $p$ and which to $e$, though, cannot be decided unambiguously. If the error term $e$ has considerably smaller entries than $p$, then the $d$-splits with larger isolation indices would belong to $p$ rather than $e$.

Let $d = p + e$, where $d$ is the observed distance, $p$ the phyletic distance linearly scaled and $e$ an error term such that the trees representing $p$ and $e$ are different. Then

$$\{d - \text{splits}\} = \{p - \text{splits}\} \cup \{e - \text{splits}\}$$

$$\alpha^d = \alpha^p + \alpha^e$$

The theory of split decomposition predicts at most $\binom{n}{2}$ $d$-splits for any $n$ by $n$ distance matrix (Bandelt and Dress, 1992; Theorem 3, p. 60, and Corollary 4, p. 62). This bound is considerably larger than $2n - 3$, the maximum number of splits in a tree connecting $n$ taxa, yet it is small enough to have all $d$-splits computed efficiently. Reanalysis of numerous distance matrices derived from sets of $n$ aligned ribosomal RNA sequences (with $n$ between 10 and 25, say) confirms that biologically relevant data typically bring about $2n$ splits, a large portion of which fit together on a single tree, and leave a small residue. In contrast, randomly generated distance matrices tend to have a rather large residue and to produce mostly *trivial* splits, separating one taxon from all the remaining ones, and only very few others, generally separating no more than two or at most three taxa from the rest.

max # $d$-splits: $\binom{n}{2} = \frac{1}{2}n^2 - \frac{1}{2}n$

max # splits in a tree: $2n - 3$

biologically relevant data bring about $2n$ splits, most of them fit on a single tree, and leave a small residue ($10 \leq n \leq 25$)

randomly generated distance matrices tend to have large residue and trivial or almost-trivial splits (separate only 2,3 taxa from the rest)

*Split Decomposition*

Assume we are given a matrix $d = (d_{ij})$ of dissimilarities between pairs of taxa $1, \ldots, n$. For any four taxa $i, j, k, l$ we compare the three distance sums $d_{ij} + d_{kl}$, $d_{ik} + d_{jl}$, $d_{il} + d_{jk}$. If $i, j, k, l$ were located on a tree such that there is a link separating $i, j$ from $k, l$, then the sum $d_{ij} + d_{kl}$ (with respect to the additive path length metric $d$) would be the smallest among those three sums. This pattern would thus be shown by any two pairs $i, j$ and $k, l$ separated by a fixed link of the tree, so that this link and its length can be reconstructed from the associated distance matrix. Since real data are far from such an ideal tree situation, we relax the criterion for accepting a partition of the taxa into two parts $\mathfrak{J}, \mathfrak{K}$ as a split supported by the distance matrix $d$: we require that for any choice of $i, j$ in $\mathfrak{J}$ and $k, l$ in $\mathfrak{K}$ the sum of the internal distances is at least not the largest among the three distance sums of the quartet $i, j, k, l$, that is,

$$d_{ij} + d_{kl} < \max\{d_{ik} + d_{jl}, d_{il} + d_{jk}\}; \qquad (1)$$

we then say that $\mathfrak{J}, \mathfrak{K}$ is a split with respect to $(d_{ij})$ or a $d$-split, for short. Every $d$-split receives a positive weight, viz., the quantity

$$\alpha_{\mathfrak{J},\mathfrak{K}} = \frac{1}{2} \cdot \min_{\substack{i,j\in\mathfrak{J},\\ k,l\in\mathfrak{K}}} (\max\{d_{ij} + d_{kl}, d_{ik} + d_{jl}, d_{il} + d_{jk}\} - d_{ij} - d_{kl}), \qquad (2)$$

which is called the *isolation index* of $\mathfrak{J}, \mathfrak{K}$. All other partitions of the taxa into two parts $\mathfrak{J}, \mathfrak{K}$ (that do not qualify as $d$-splits) thus have index 0. Notice that the isolation index of a split $\mathfrak{J}, \mathfrak{K}$ of an ideal tree is exactly the length of the link whose removal results in the two components $\mathfrak{J}$ and $\mathfrak{K}$.

Let $d = (d_{ij})$ be a $n \times n$ dissimilarity matrix.

In the case of a tree, for every four taxa $i, j, k, l$,
if there is an edge separating $i, j$ and $k, l$, then

$$d_{ij} + d_{kl} < d_{ik} + d_{jl} = d_{il} + d_{jk}$$

This edge can be reconstructed from the associated distance matrix.

A split $\mathcal{J}, \mathcal{K}$ is a **$d$-split** if for every $i, j \in \mathcal{J}$ and $k, l \in \mathcal{K}$ we have

$$d_{ij} + d_{kl} < \max\{d_{ik} + d_{jl}, d_{il} + d_{jk}\}$$

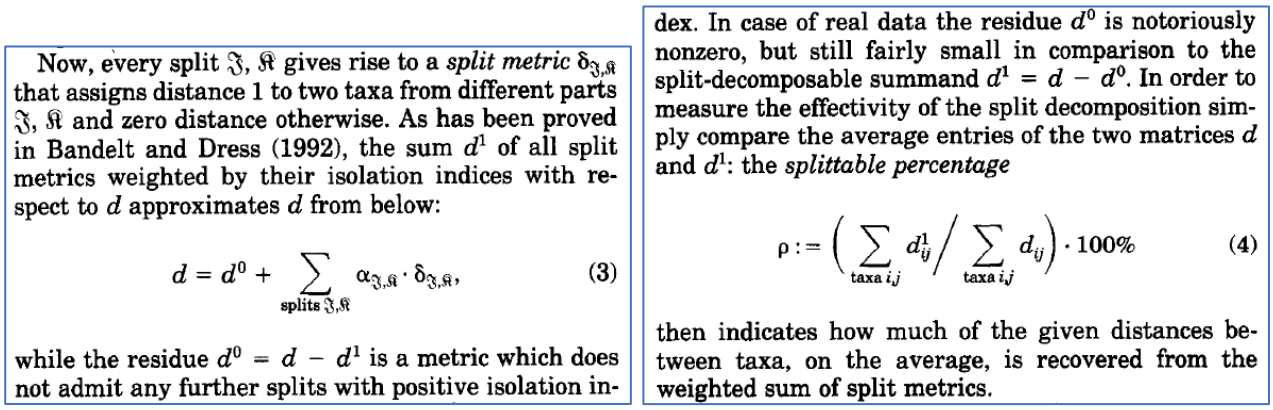
This is a relaxed condition representing splits supported by the distance matrix.

The **isolation index** of $\mathcal{J}, \mathcal{K}$ is the quantity

$$\alpha_{\mathcal{J},\mathcal{K}} := \frac{1}{2} \cdot \min_{\substack{i,j\in\mathcal{J}\\ k,l\in\mathcal{K}}} \left( \max\{d_{ij} + d_{kl}, d_{ik} + d_{jl}, d_{il} + d_{jk}\} - d_{ij} - d_{kl} \right)$$

Notice that $d$-splits have positive isolation index,
while non $d$-splits have zero isolation index.

Also, in the case of a tree, the isolation index of a split $\mathcal{J}, \mathcal{K}$ is the length of the edge separating $\mathcal{J}$ and $\mathcal{K}$ (that is removing this edge results in two connected components, namely $\mathcal{J}$ and $\mathcal{K}$).



Now, every split $\mathfrak{I}, \mathfrak{R}$ gives rise to a *split metric* $\delta_{\mathfrak{I},\mathfrak{R}}$ that assigns distance 1 to two taxa from different parts $\mathfrak{I}$, $\mathfrak{R}$ and zero distance otherwise. As has been proved in Bandelt and Dress (1992), the sum $d^1$ of all split metrics weighted by their isolation indices with respect to $d$ approximates $d$ from below:

$$d = d^0 + \sum_{\text{splits } \mathfrak{I},\mathfrak{R}} \alpha_{\mathfrak{I},\mathfrak{R}} \cdot \delta_{\mathfrak{I},\mathfrak{R}}, \qquad (3)$$

while the residue $d^0 = d - d^1$ is a metric which does not admit any further splits with positive isolation index. In case of real data the residue $d^0$ is notoriously nonzero, but still fairly small in comparison to the split-decomposable summand $d^1 = d - d^0$. In order to measure the effectivity of the split decomposition simply compare the average entries of the two matrices $d$ and $d^1$: the *splittable percentage*

$$\rho := \left( \sum_{\text{taxa } i,j} d^1_{ij} \Big/ \sum_{\text{taxa } i,j} d_{ij} \right) \cdot 100\% \qquad (4)$$

then indicates how much of the given distances between taxa, on the average, is recovered from the weighted sum of split metrics.

The **split metric** on $\mathcal{J}, \mathcal{K}$ is the function $\delta_{\mathcal{J},\mathcal{K}}$ defined as

$$\delta_{\mathcal{J},\mathcal{K}}(i,j) := \begin{cases} 0\,, & \text{if } i,j \in \mathcal{J} \text{ or } i,j \in \mathcal{K} \\ 1\,, & \text{otherwise} \end{cases}$$

We can write the **split decomposition**

$$d = d^0 + \sum_{\text{splits } \mathcal{J},\mathcal{K}} \alpha_{\mathcal{J},\mathcal{K}} \cdot \delta_{\mathcal{J},\mathcal{K}}$$

where the residue $d^0 = d - d^1$ is a **split-prime** dissimilarity matrix (that is it does not admit any $d^0$-split).

For real data, the residue $d^0$ is non-zero but small compared to $d^1$.

The **splittable percentage** is

$$\rho := \left( \sum_{\text{taxa } i,j} d^1_{ij} \Big/ \sum_{\text{taxa } i,j} d_{ij} \right) \cdot 100\%$$

This quantity represents how much $d$ is recovered from $d^1$, on average.

*Finding the d-Splits*

It is not difficult to compute the $d$-splits efficiently, since the number of all $d$-splits is bounded by $\binom{n}{2}$ where $n$ is the number of taxa. One proceeds recursively as follows: enumerate the taxa as $1, 2, \ldots, n$, and suppose the $d$-splits restricted to the subset $\{1, \ldots, i-1\}$ are already determined; then for each $d$-split $\mathfrak{I}$, $\mathfrak{K}$ of this subset check whether $\mathfrak{I} \cup \{i\}$, $\mathfrak{K}$ or $\mathfrak{I}$, $\mathfrak{K} \cup \{i\}$ qualifies as a $d$-split of the enlarged subset $\{1, \ldots, i-1, i\}$; further check whether $\{1, \ldots, i-1\}$, $\{i\}$ is a $d$-split of the enlarged set. This procedure stops after $i = n$ has been processed, providing us with the complete list of $d$-splits of the full set.

The total number of steps is bounded by a polynomial in $n$ of degree 6 (with a small leading coefficient). For example, the total number of inequalities (1) that have to be checked in case $n = 8$ is less than 1000 in the worst case and considerably smaller in general, so

Since the number of $d$-splits is at most $\binom{n}{2}$ ,
we can compute them in the following way:

suppose the $d$-splits on $\{1, \ldots, i-1\}$ have been already determined;
then for each $d$-split $\mathcal{J}, \mathcal{K}$ of this subset

- check if $\mathcal{J} \cup \{i\}, \mathcal{K}$ or $\mathcal{J}, \mathcal{K} \cup \{i\}$ are $d$-splits of $\{1, \ldots, i\}$
- check if $\{1, \ldots, i-1\}, \{i\}$ is a $d$-split of $\{1, \ldots, i\}$

This procedure leads to an algorithm of complexity $O(n^6)$.

Dovrebbe essere meno di 1000 $\beta$ indici per $n = 7$.

*Graphical Representation*

The splits of a tree are in one-to-one correspondence with the links and thus are easily read from the diagram. More generally, any split-decomposable metric $d^1$ can be represented by a mesh-like graph, the links of which are weighted by the corresponding isolation indices. The graphs in question can be chosen among the subgraphs of $\binom{n}{2}$-dimensional cubes (where $n$ is the number of taxa), but fortunately they are normally not too weird and can often be drawn in the plane without intersection of links. In contrast to the tree situation, a single split now corresponds to a family of several "parallel" links, which constitutes a cutset, that is, removing these links disconnects ("splits") the graph.

Successive application of the following rule determines a cutset: for each "cell," i.e., a cycle without short-cuts, opposite edges belong to the same cutset (and hence receive the same weight). The distance between two taxa $i$ and $j$ is then obtained as the sum of all weights along a path connecting $i$ and $j$ which has the smallest number of links. Note that there may be more than one representing graph meeting the requirements and having a minimal number of nodes; see Fig. 1. This graph also appears as a subgraph in Figs. 2 and 4 below.

In order to generate such graphs one proceeds iteratively by incorporating one split after the other: suppose a minimal graph representing a subcollection of splits has been constructed, then this graph is expanded so that the next split gets realized as well, thereby obeying the above rules on cycles, cf. Bandelt (1992). Observe that the order in which the splits are processed may affect the final outcome. For example, the split AFG in the upper graph of Fig. 1 cannot be the last one that gets processed since otherwise the predecessor graph would not have been minimal.

the splits of a tree are in correspondece with its edges

a split-decomposable metric can be represented by a mesh-like graph, where its edges are weighted by the corresponding isolation indices

the graph can be chosen among the subgraphs of $\binom{n}{2}$-dimensional cubes; often they can be drawn in the plane without intersections of edges

now a single split corresponds to a family of "parallel" edges, which constitutes a cutset, that is removing these edges disconnects ("splits") the graph

for each "cell" (a cycle without short-cuts), opposite edges belong to the same cutset, hence they have the same weight

the distance between two taxa is the sum of the weights of a path with the minimal number of edges

there may be more than one graph meeting the requirements and with minimal number of nodes

non è chiaro l'algoritmo per costruire un grafo minimo

the order in which splits are processed may affect the final outcome

*Greedy Tree Selection*

If the collection of splits for a matrix $d$ is sufficiently large, then it probably includes the splits of trees inferred from the data by other methods. Therefore, in order to estimate a tree, one could select a maximal subset of splits fitting into a tree, so that a certain optimality criterion is met. Indeed, for data sets of medium size the splits obtained from an estimated tree often coincide with the $d$-splits whose indices exceed a certain threshold value.

Recall that a set of splits is realizable on a tree if and only if the splits are pairwise *compatible*, i.e., any two splits $\mathfrak{I}_1$, $\mathfrak{R}_1$ and $\mathfrak{I}_2$, $\mathfrak{R}_2$ of that set have parts, $\mathfrak{I}_1$ and $\mathfrak{I}_2$ say, with empty intersection. For example, every *trivial split*, opposing one taxon to all others, is compatible with all splits.

An optimality criterion would require maximizing an appropriately defined function, e.g., the sum, of the isolation indices (of the chosen splits); optimal solutions could then, of course, be found by branch and bound methods. This bears some resemblance to the compatibility method of Meacham and Estabrook (1985) and the closest tree selection in Hendy and Penny's (1991) spectral analysis.

Even the greedy selection strategy seems to work surprisingly well (when compared to standard methods of tree inference): successively select a new $d$-split that has the highest isolation index and is still compatible with the splits collected so far. The Sarich (1969) data

if the set of $d$-splits is large enough, then it probably includes the splits of trees inferred by other methods

in order to estimate a tree, we can select a maximal subset of $d$-splits fitting into a tree according to a certain optimality criterion

for data sets of medium size, the splits obtained from estimated trees often coincide with the $d$-splits with high isolation index

a set of splits is realizable on a tree if and only if they are pairwise **compatible**, that is for any two splits $\mathcal{J}_1, \mathcal{K}_1$ and $\mathcal{J}_2, \mathcal{K}_2$ at least one of the following intersections is empty

$$\mathcal{J}_1 \cap \mathcal{J}_2 \,, \qquad \mathcal{J}_1 \cap \mathcal{K}_2 \,, \qquad \mathcal{K}_1 \cap \mathcal{J}_2 \,, \qquad \mathcal{K}_1 \cap \mathcal{K}_2$$

every trivial split is compatible with all splits

an optimality criterion would require maximizing an appropriately defined function (for example sum of isolation indices);
optimal solutions could be found by branch and bound methods

even greedy selection works well (compared with standard methods of tree inference): select a new $d$-split with highest isolation index that is still compatible with the previous ones

*Detecting Sequence Convergence*

Incompatible $d$-splits are obtained when split decomposition is applied to distances derived from a set of aligned sequences some of which have undergone massive parallel substitutions. The subsequent cases are instructive: first, parallel amino acid replacements in cow and langur lysozymes, and second, thermophilic convergence in eubacterial ribosomal RNA:

1. Stewart and Wilson (1987) compared the amino acid sequences of lysozymes from cows, langurs, baboons, humans, rats, and horses; cf. Table 4 of Li and

and rat (0.5). In particular, the convergence in the cow and langur lineages is manifest in the $d$-splits, but (with respect to isolation indices) it is less pronounced than the parallelism involving the horse and rat lineages. Since the two nontrivial $d$-splits with largest indices are incompatible, one concludes that inferring *phylogenetic trees* from these data would not yield reliable results, while concerning parallel evolution they offer rather interesting and valuable information.

When one focusses on a particular subgroup of taxa, e.g., the archaebacteria in the study of Leffers *et al.* (1987), then the corresponding distance submatrix should be investigated separately. Since a single quartet of taxa, two of which are in either part of a potential split, can cause the rejection of this split (see the definition of isolation index), the total number of splits (with positive index) tends to be relatively small for larger data sets. So, some of the "local" information on parallelism and systematic error reflected in the distance matrix for a subgroup of taxa is lost in split analysis (i.e., transferred to the residue) when other, distantly related taxa are taken into account.

transformation from sequence dissimilarities to evolutionary distances increases the residual parts, while the number of skew splits decreases. There is thus a trade-off between indecomposable "noise" and the incompatibility between splits for these data, which needs further analysis.

Remarkably, it is not always true that the number of splits for estimated evolutionary distances is smaller than the one for uncorrected sequence dissimilarities.

(dissimilarity), respectively. It may be noted that the Jukes and Cantor transformation increases also the variation in these data. This correlates with the

Split decomposition can enhance phylogenetic analysis of distance data by detecting opposite groupings ("splits") of organisms that are defined by distinctive distance features, caused by common ancestry, convergence, or systematic or random errors. A major part of random noise contained within the data is transferred to the split-prime residue, which is removed from the data in the course of analysis. This residue typically covers 10 to 30% of the total distance in the case of ribosomal RNA data (with about 10 to 25 taxa), whereas for randomly chosen metrics this amount easily exceeds 50%. Some portion of random and systematic error survives in the split-decomposable part and is manifest in the incompatibilities of splits. A split is likely to fall into this category when its isolation index is relatively small and it is incompatible with splits having much larger indices. Therefore, selecting a clique of compatible splits in a greedy fashion according to isolation indices often recovers most of the phylogenetic trees that are estimated by other methods, but normally leaves unresolved the most uncertain furcations.

A graphical representation of the split-decomposable part of a distance matrix furthers the understanding of tentative phylogenetic relationships plus inherent parallelism. If the number of taxa is very small, then it is possible sometimes even to integrate the split-prime residue into the diagram as well, thus visualizing the full decomposition of the distance matrix.

For very large data sets that include fairly distant groups of taxa, one can additionally perform a secondary analysis of "partial $d$-splits." To this end the whole set of taxa is partitioned into smaller subcollections identified by compatible splits with large isolation indices.

When several distance matrices for one and the same set of organisms are available (for instance, through different weighting schemes of characters or particular methods of correction), it can be instructive to compare the corresponding splittable percentages and the structure of the resulting split systems, in order to evaluate the phylogenetic content of the respective distance matrices.

In closing, we may speculate on further potential applications of our method. In view of its ability to process incompatible splits, split decomposition perhaps can also serve as a tool for investigating reticulate evolution. It is, however, not obvious how to clearly discriminate between random and systematic

# *Phylogenetic Networks: Concepts, Algorithms and Applications*

> By definition, phylogenetic trees are well suited to represent evolutionary histories in which the main events are speciations (at the internal nodes of the tree) and descent with modification (along the edges of the tree). But such trees are less suited to model mechanisms of *reticulate evolution* [219], such as horizontal gene transfer, hybridization, recombination or reassortment. Moreover, mechanisms such as incomplete lineage sorting, or complicated patterns of gene duplication and loss, can lead to incompatibilities that cannot be represented on a tree. Although the analysis of individual genes or short stretches of genomic sequence often gives strong support to a phylogenetic tree, different genes or sequence segments usually support different trees.

horizontal/lateral gene transfer:

- direct transfer of genes from one organism to another
- occur very frequently in prokaryotes
- main mechanisms are transformation, conjugation, transduction

hybridization:

- each cell has genetic material from two organisms of different varieties/species/genera

recombination (and gene conversion):

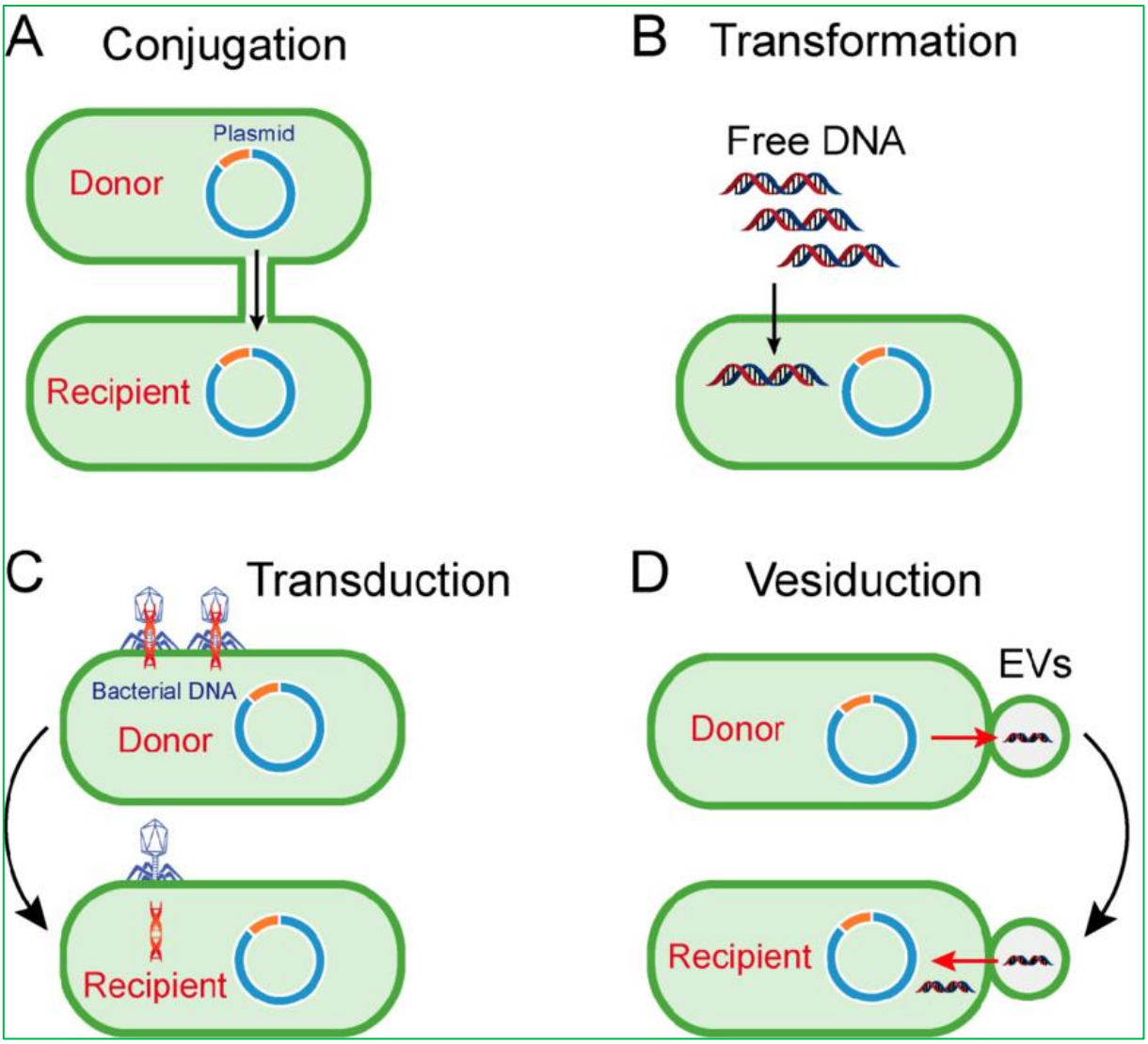
- new combinations of genetic material through pairing and shuffling of very similar DNA sequences
- studied by population genetics, which deals with the statistical analysis of the inheritance and prevalence of genes in populations

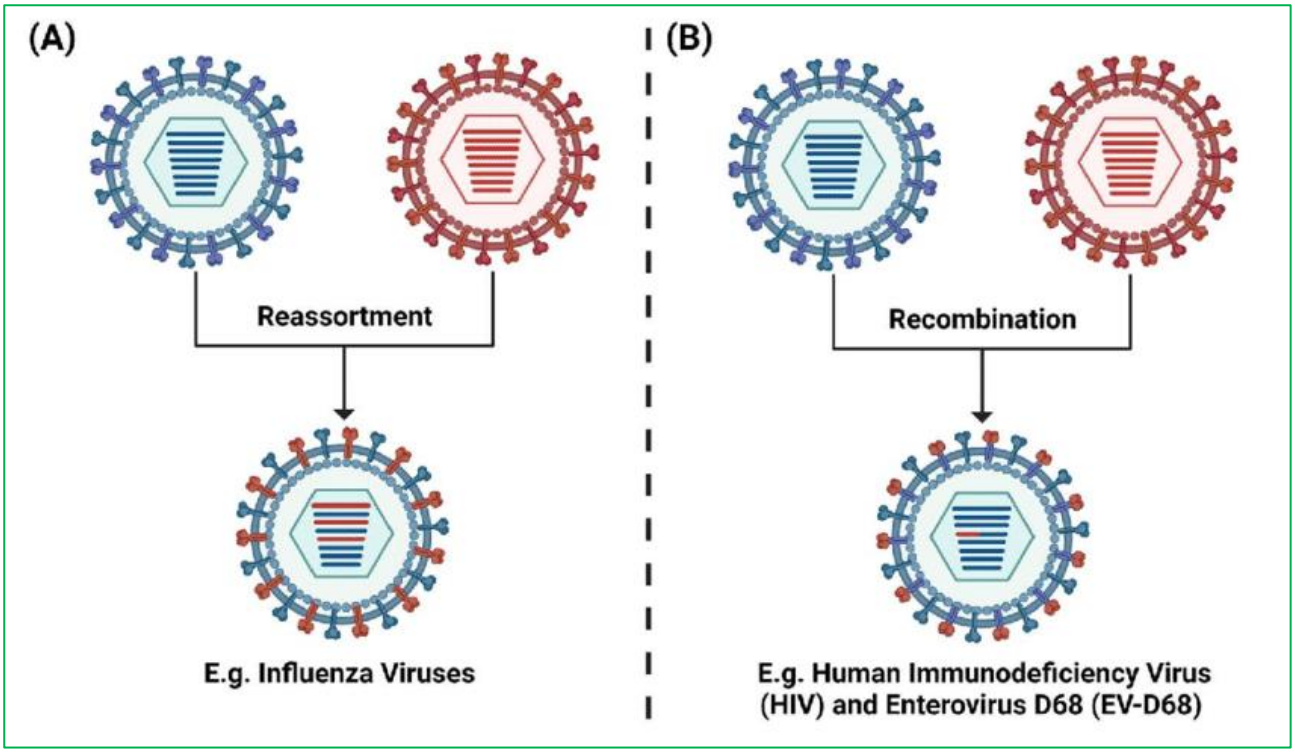
reassortment:

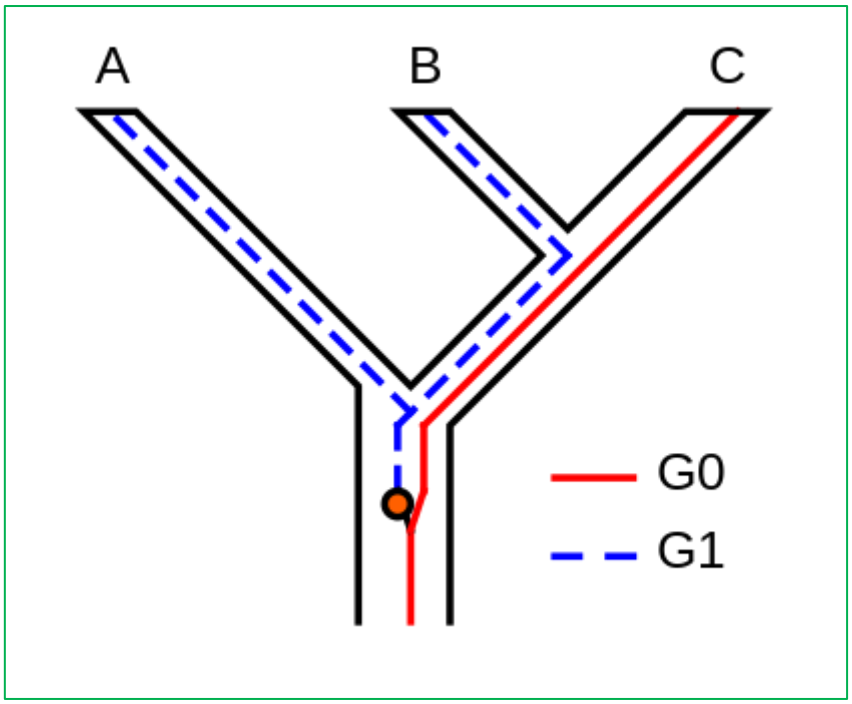
- involves swapping of genetic material between individual organisms
- occurs for example in viruses

incomplete lineage sorting:

- the tree of a single gene differs from the species tree


A
Conjugation
Plasmid
Donor
Recipient
B
Transformation
Free DNA
C
Transduction
Bacterial DNA
Donor
Recipient
D
Vesiduction
EVs
Donor
Recipient


(A)
Reassortment
E.g. Influenza Viruses
(B)
Recombination
E.g. Human Immunodeficiency Virus (HIV) and Enterovirus D68 (EV-D68)




The persistence of polymorphisms across different speciation events can cause **incomplete lineage sorting**.
Suppose two subsequent speciation events occur where an ancestor species gives rise firstly to species A, and secondly to species B and C.

When studying a single gene, it can have multiple versions (alleles) causing different characters to appear (polymorphisms).
In the example shown in Figure 1, the gene G has two versions (alleles), G0 and G1.
The ancestor of A, B and C originally had only one version of gene G, G0.

At some point, a mutation occurred and the ancestral population became polymorphic, with some individuals having G0 and others G1.
When species A split off, it retained only G1, while the ancestor of B and C remained polymorphic. When B and C diverged, B retained only G1 and C only G0; neither were now polymorphic in G.

The tree for gene G shows A and B as sisters, whereas the species tree shows B and C as sisters.
If the phylogeny of these species is based on gene G, it will not represent the actual relationships between the species.
In other words, the most related species will not necessarily inherit the most related genes.

While there is a great need for practical and reliable computational methods for inferring rooted phylogenetic networks to *explicitly* describe evolutionary scenarios involving reticulate events, generally speaking, such methods do not yet exist, or have not yet matured enough to become standard tools.

In contrast, there exist a number of established computational methods for inferring *unrooted* phylogenetic networks, which are used to *abstractly* describe reticulate evolution by providing a visualization of incompatible evolutionary pathways. Among the most widely used are methods for computing split networks [9], median networks [11] quasi-median networks [10], and other types of haplotype networks [52]. Such methods are not only used in phylogenetic analysis, but also in phylogeography and population genetics, as well.

---

**taxon**: taxonomic unit that represents some species/group/individual organism whose evolutionary history is of interest

**clade** (or **monophyletic group**): a group of organisms that contains all and only the descendants of a common ancestor and the ancestor itself

**paraphyletic group**: a group of organisms that does not contain all species descended from the last common ancestor of all its members

**polyphyletic group**: a group of organisms that consists of species that are descended from multiple last common ancestors, which each have other descendants that are not contained in the group

Let $\mathcal{X}$ be a set of taxa.

A **cluster** is any non-empty proper subset of $\mathcal{X}$.

A **split** is a partition of $\mathcal{X}$ into two non-empty subsets $A, B \subseteq \mathcal{X}$ such that $A \cap B = \emptyset$ and $A \cup B = \mathcal{X}$.

A *phylogeny* describes the evolutionary history of a set of taxa and the ultimate goal of any phylogenetic analysis is to reconstruct some part of the *tree of life*, the evolutionary history of all life on Earth.

The main role of phylogenetic analysis is to compute a set of clusters on $\mathcal{X}$ such that each cluster is monophyletic, and not a paraphyletic or polyphyletic group, and then to represent the clusters as a rooted phylogenetic tree or network,

The concepts of clusters and splits are closely related: while clusters are employed to represent clades in rooted histories, splits are used to represent clades and their complements in unrooted settings.

---

## Sequence alignment

Two DNA or protein sequences that have a high similarity are usually presumed to be *homologous*, that is, to have evolved from a common ancestral sequence. Both high similarity and also homology of protein sequences often imply that the structure of the proteins is similar, which in turn usually implies that the proteins have a similar function.

Phylogenetic trees and networks are generally computed from aligned DNA or protein sequences and so the first step in an evolutionary study is often to build such an alignment.

The basic idea of alignment is to write two sequences one above the other so as to maximize the number of similar or identical bases or amino acids that occur underneath each other.

### Pairwise sequence alignment

**Definition 2.2.1** (Pairwise alignment) *Suppose we are given two sequences $a$ and $b$ over some alphabet $\Sigma$, for example $\Sigma = \{A, C, G, T\}$ for DNA sequences. A* pairwise (global) alignment *of $a$ and $b$ is obtained by inserting gaps ("–") into either or both sequences so that the resulting sequences $a^*$ and $b^*$ are of equal length and thus can be written underneath (or "opposite") each other in such a way that each symbol in the one string is opposite to exactly one symbol in the other string.*

Given a pairwise alignment, we can either score the distance between the two sequences, or their similarity. A popular distance measure is the *edit distance* that counts the number of *insertion*, *deletion* and *replacement* operations required to convert the one sequence into the other. In an alignment of sequences $a$ and $b$, the number of insertions is given by the number of symbols in $b$ that are aligned to a gap in $a$, the number of deletions is given by the number symbols in $a$ that are aligned to a gap in $b$, and the number of replacements is given by the number of symbols in $a$ that are aligned to a different symbol in $b$.

Although DNA alignments are sometimes scored using the edit distance, alignments of molecular sequences, especially protein sequences, are usually scored using a similarity score based on a *substitution matrix* that provides a score for each aligned pair of residues (or nucleotides):

**Definition 2.2.2** (Substitution matrix) *A* substitution matrix $S$ *over an alphabet* $\Sigma = \{y_1, y_2, \ldots, y_t\}$ *of size* $t$ *is a* $t \times t$ *matrix in which each entry* $s(y, z)$ *assigns a score to the substitution of the symbol* $y$ *by the symbol* $z$ *in an alignment.*

For DNA sequences, we have $\Sigma = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$ and $t = 4$, and the simplest possible substitution matrix treats all nucleotides identically and assigns a score of $+1$ to a pair of identical nucleotides and a score of $-1$ to the substitution of one nucleotide by a different one.

There are two types of nucleotides in DNA, namely *pyrimidines*, which include cytosine and thymine, and *purines*, which include adenine and guanine. For biochemical reasons, *transitions*, which are substitutions between two pyrimidines or between two purines, occur more frequently than *transversions*, which are substitutions between a purine and a pyrimidine. A more sophisticated substitution matrix for DNA will assign different scores to these different types of substitutions.

Altogether, there are these three main variants of the pairwise alignment problem, each of which can be combined either with the goal of minimizing distance or maximizing similarity – using some chosen substitution matrix – and also there is the choice of whether to use a linear or affine gap penalty.

These and most other variations of the pairwise alignment problem can all be solved efficiently using different variants of the same basic dynamic programming approach.

All these algorithms, including further modifications to accommodate affine gap penalties, run in time that is in proportion to the product of the lengths of the two sequences (that is, in $O(mn)$) and can be modified further to require only linear space.

## Multiple sequence alignment

In phylogenetics, a set of taxa $\mathcal{X} = \{x_1, \ldots, x_n\}$ is often represented by a collection of molecular sequences, $A = (a_1, \ldots, a_n)$, where the $i$-th sequence $a_i$ was obtained from taxon $x_i$ and corresponds to some specific gene or locus. For phylogenetic analysis, care is taken to ensure that the sequences are homologous, that is, have evolved from a common ancestor sequence.

The sequences in $A$ differ from each other due to evolutionary events such as insertions, deletions and mutations. To facilitate phylogenetic analysis, these sequences are usually aligned by inserting gaps into each sequence such that the resulting collection of sequences

$$M = \begin{Bmatrix} a^*_{11} & a^*_{12} & \ldots & a^*_{1m} \\ a^*_{21} & a^*_{22} & \ldots & a^*_{2m} \\ & & \ldots & \\ a^*_{n1} & a^*_{n2} & \ldots & a^*_{nm} \end{Bmatrix} \tag{2.11}$$

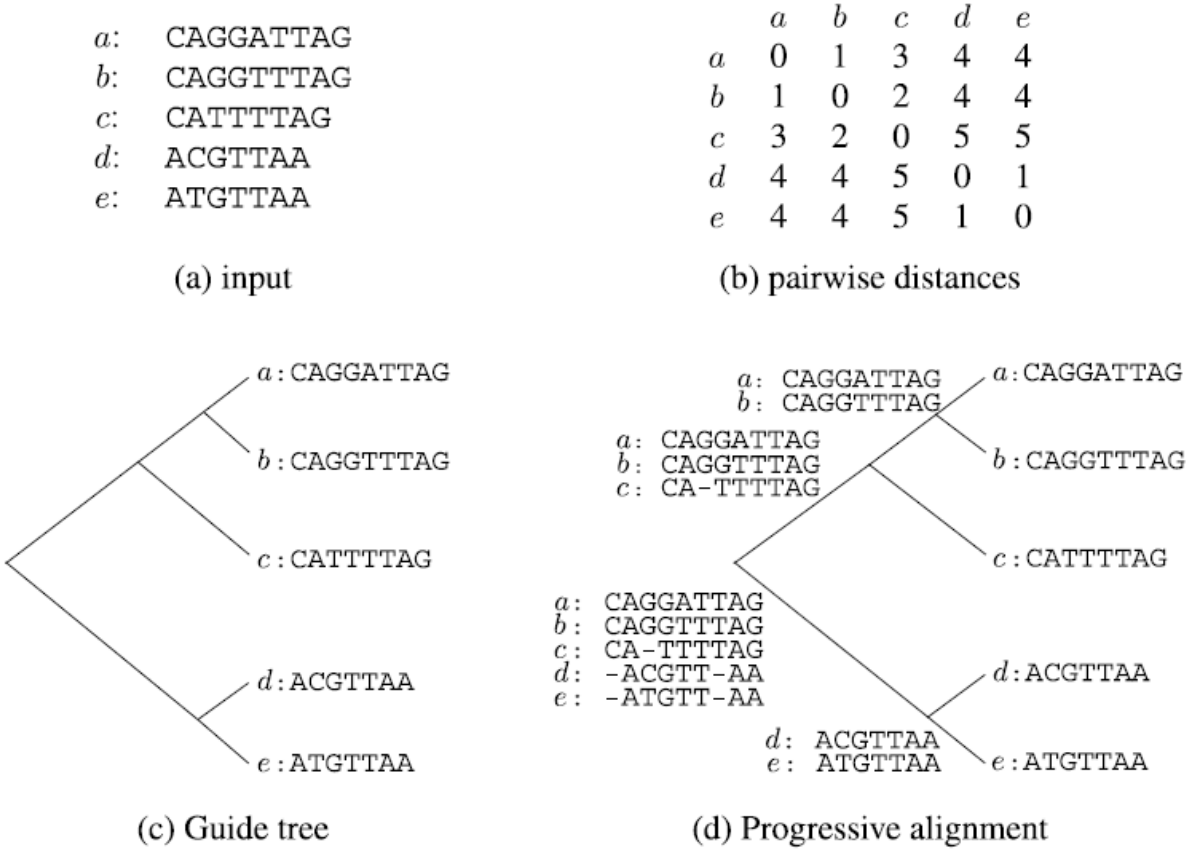
all have the same length $m$, forming a *multiple sequence alignment* of length $m$, see Figure 2.6. We are particularly interested in finding a multiple sequence alignment that achieves an optimal score according to an appropriate scoring scheme.

As discussed in the previous section, a pairwise sequence alignment of amino acid sequences is usually scored with the help of a substitution matrix such as a BLOSUM matrix, which assigns an empirically derived score to each pair of aligned amino acids; positive for favorable substitutions and negative for unfavorable ones. The score of the whole alignment is given by the sum of scores for each pair of aligned characters.

Multiple sequence alignments are usually scored using the *sum of pairs* approach: the score of a multiple sequence alignment is given by the sum of scores of all

pairwise alignments induced by pairs of sequences in the multiple sequence alignment. The problem of computing a multiple sequence alignment that optimizes the sum of pairs score is known to be computationally hard [240] and thus, in practice, heuristics such as *progressive alignment* are used.

The basic idea of *progressive alignment* is to build a multiple sequence alignment incrementally, or *progressively*, by first aligning pairs of similar sequences, and then aligning sequences to *profiles* (in this context, a *profile* is simply a multiple sequence alignment of a subset of the input sequences) and then finally aligning profiles with profiles to obtain a multiple sequence alignment of the total set of input sequences,




(a) input (b) pairwise distances

(c) Guide tree (d) Progressive alignment

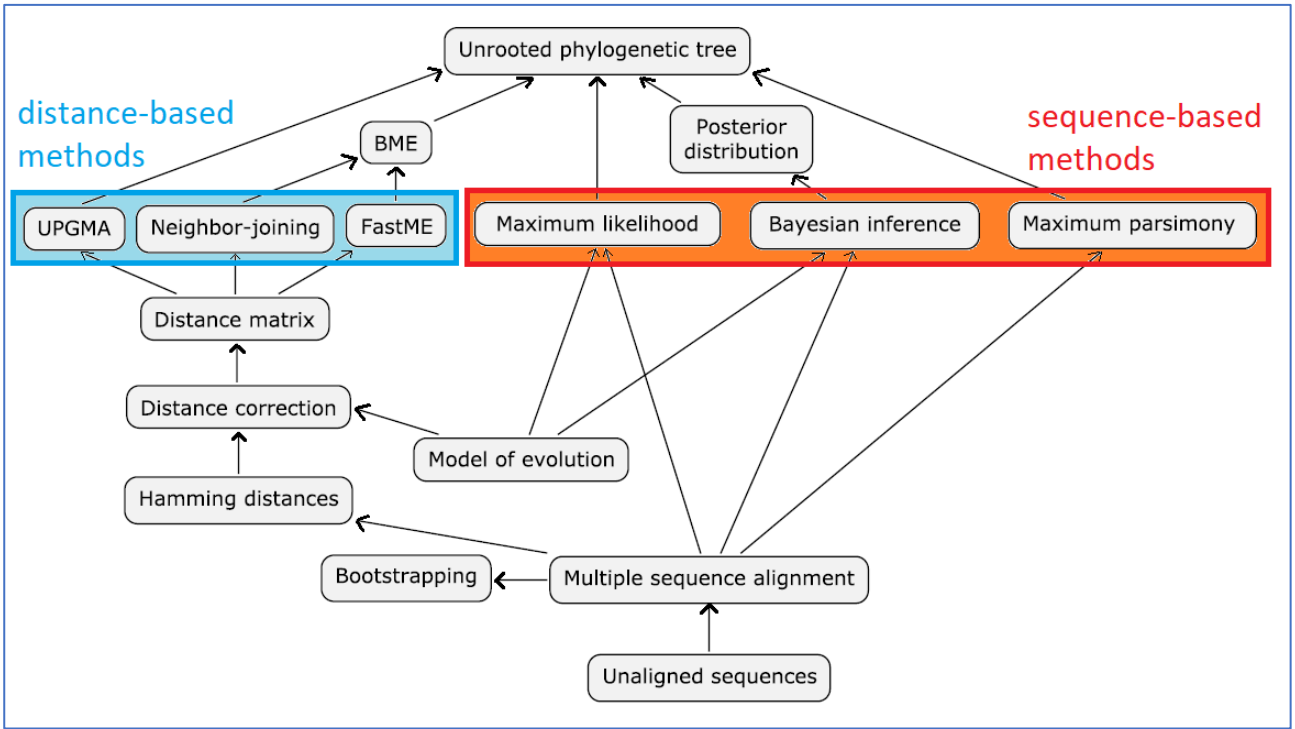
Outline of the progressive alignment approach. For a given set of input sequences (a), a distance matrix is computed (b), and from this a phylogenetic *guide tree* is derived (c). Then, sequences at the leaves of the tree are aligned to produce profiles at the internal nodes of the tree, which in turn are aligned to each other, continuing in this manner until the final alignment of all sequences is obtained at the root of the guide tree (d).

**Definition 2.3.2** (Characters in a multiple sequence alignment) *Let $M$ be a multiple sequence alignment on $\mathcal{X}$. Each column of $M$ is called a* character *and each symbol that occurs in such a column is called a* character state.

## Phylogenetic trees

Phylogenetic analysis aims at uncovering the evolutionary relationships between different species or taxa, to obtain an understanding of the evolution of life on Earth. Phylogenetic trees are widely used to address this task and are usually computed from molecular sequences. They also have applications in many other areas. For example, they are used to determine the age and rate of diversification of taxa, to understand the evolutionary history of gene families, in sequence-analysis methods to allow *phylogenetic footprinting*, in epidemiology to trace the origin and transmission of infectious diseases, or to study the co-evolution of hosts and parasites.

The main focus of this book is on phylogenetic networks. However, as phylogenetic trees generalize to phylogenetic networks and also to make the book reasonably self-contained, in this chapter we give a brief introduction to some of the main methods used to infer phylogenetic trees.



The focus of this chapter is on how to compute *unrooted phylogenetic trees*. Usually, the process of phylogenetic inference is begun with a *multiple sequence alignment*. From this, one can pursue either a distance-based analysis, or a sequenced-based one.

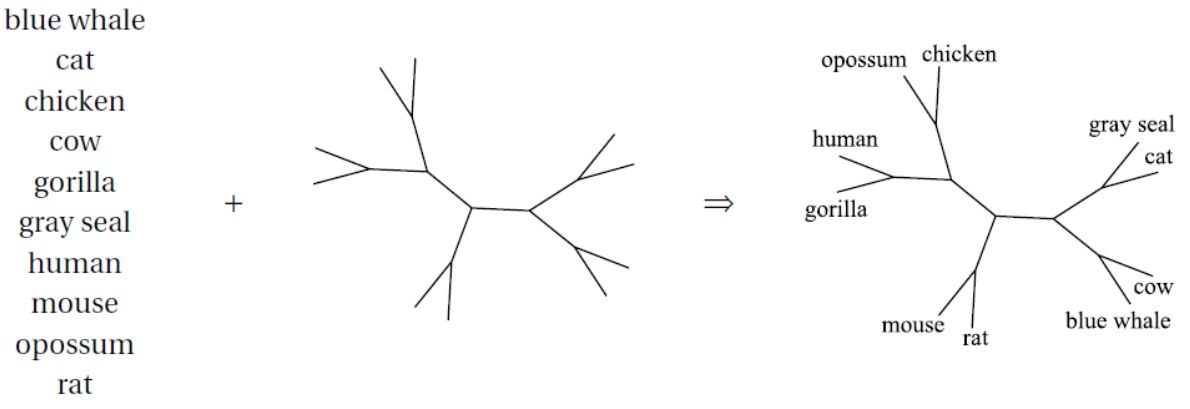
In a distance-based analysis of DNA sequences, first the *Hamming distances* between pairs of sequences are computed. These distances are then exposed to a *distance correction* that is based on some appropriate *model of evolution*. The resulting *distance matrix* is then provided to a method such as *UPGMA*, *neighbor-joining* or *FastME*, to produce an unrooted phylogenetic tree. The latter two are both heuristics for computing the *balanced minimum evolution* (BME) tree. Distance methods are often applied to distance matrices obtained in other ways, too.

For a sequence-based analysis, there are three main types of approaches to choose from. A *maximum parsimony* method searches for a phylogenetic tree that explains the given dataset using a minimum number of observable mutations. A *maximum likelihood* approach aims at determining a tree that maximizes the likelihood of generating the given dataset, under a given model of evolution. Bayesian methods attempt to compute the *posterior distribution* of trees, based on the given input data, a specified model of evolution and a presumed prior distribution of phylogenetic trees.

The posterior distribution provided by a Bayesian method provides a means of evaluating the support of computed groups of taxa. For other methods, *bootstrapping* can be used to perform such evaluations.

## Phylogenetic trees

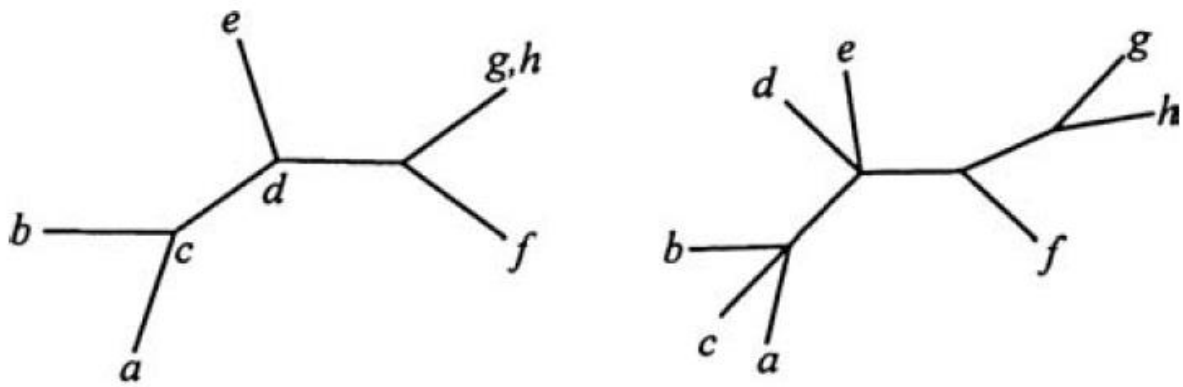
**Definition 3.2.1** (Phylogenetic tree) *Given a set of taxa $\mathcal{X}$, a* phylogenetic tree *$T$ on $\mathcal{X}$ consists of a tree $T = (V, E)$, in which all nodes have degree $\neq 2$, together with a taxon labeling $\lambda : \mathcal{X} \to V$ that assigns exactly one taxon to every leaf and none to any internal node.*




(a) Taxa (b) Tree (c) Phylogenetic tree

In mathematical phylogeny, the more general concept of an *$\mathcal{X}$-tree* is sometimes considered, which is obtained by weakening the requirements imposed on the tree, by allowing nodes of degree 2, and by weakening the requirements imposed on the taxon labeling $\lambda$, so as to demand only that every node of degree 0 or 2 obtains at least one label, thus allowing (multiple) labels on both leaves and internal nodes

any $\mathcal{X}$-tree can be converted into a corresponding phylogenetic tree simply by introducing a new leaf for every taxon that appears as the label of an internal node, or as the label of a leaf node that is labeled by more than one taxon,




(a) $\mathcal{X}$-tree (b) Phylogenetic tree

**Definition 3.2.2** (Rooted phylogenetic tree) *Given a set of taxa $\mathcal{X}$, a* rooted phylogenetic tree *consists of a rooted tree $T = (V, E, \rho)$ and a taxon labeling $\lambda : \mathcal{X} \to V$ that assigns exactly one taxon to every leaf and none to any internal node. All nodes, except $\rho$, must have degree $\neq 2$.*

From a theoretical and algorithmic point of view, unrooted phylogenetic trees are sometimes easier to work with than rooted ones. However, in biology, rooted phylogenetic trees are usually of more interest, as the placement of the root defines a direction of time along the phylogenetic tree, namely away from the root, and because rooted phylogenetic trees explicitly define clades or clusters of putatively related taxa.

An unrooted phylogenetic tree can be *rooted* simply by declaring one of its nodes to be the root, or by inserting a new root node into the interior of one of the edges of the tree. In practice, a popular way to determine where to *root* a phylogenetic tree is by using an *outgroup*. An outgroup is a taxon that is closely related to the main group of taxa under consideration, but lies outside of it. The root is placed on the edge leading to the outgroup.

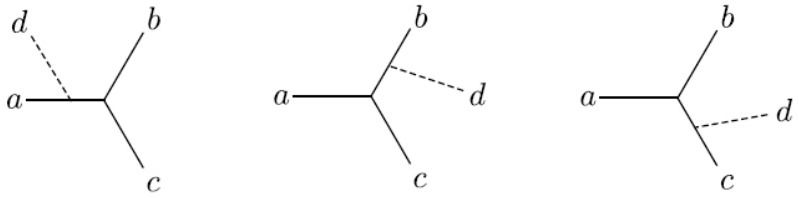
In many cases, the edges of a phylogenetic tree are each assigned a *weight*, or *length*, which often indicates the number of mutations that have happened along an edge or is correlated to evolutionary time in some other way:

**Definition 3.2.4** (Edge weights) *A phylogenetic tree $T$ is called an* edge-weighted tree *if we are given a map $\omega$ that assigns a non-negative* weight *or* length *$\omega(e)$ to every edge $e$ of the tree.*

## The number of phylogenetic trees

Let $V(n)$ and $E(n)$ be respectively the number of nodes and edges of a bifurcating (unrooted) phylogenetic tree on $n$ taxa.
Notice that these numbers are well defined, since they do not depend on the exact topology of the tree.

We can obtain a tree on $n + 1$ taxa by "grafting" (*innestare*) a new taxon onto a tree on $n$ taxa.




All possible ways of adding a fourth taxon $d$ to an unrooted phylogenetic tree on three taxa $\{a, b, c\}$.

Observe that by adding a taxon, the number of nodes increases by 2 (the node corresponding to the new taxa plus the node of the intersection) and also the number of edges increases by 2 (the new edge connecting the node of the new taxa plus the "grafted" edge that gets splitted in two edges).

Therefore, these numbers can be characterized by the following relations

$$\begin{cases} V(n+1) = V(n) + 2 \\ V(2) = 2 \end{cases}, \qquad \begin{cases} E(n+1) = E(n) + 2 \\ E(2) = 1 \end{cases}$$

hence, by induction

$$V(n) = 2n - 2\,, \qquad E(n) = 2n - 3$$

For the number of bifurcating trees $Tree(n)$ we have

$$\begin{cases} Tree(n+1) = Tree(n) \cdot E(n) \\ Tree(2) = 1 \end{cases}$$

and again by induction

$$\begin{aligned} Tree(n) &= 1 \cdot 3 \cdot 5 \cdot \ldots \cdot (2n-5) \\ &= (2n-5)!! \end{aligned}$$

Rooted trees can be obtained by unrooted trees by inserting a node (the root) in the middle of an edge. Thus

$$V_R(n) = V_U(n) + 1 = 2n - 1$$

$$E_R(n) = E_U(n) + 1 = 2n - 2$$

$$Tree_R(n) = Tree_U(n+1) = (2n-3)!!$$

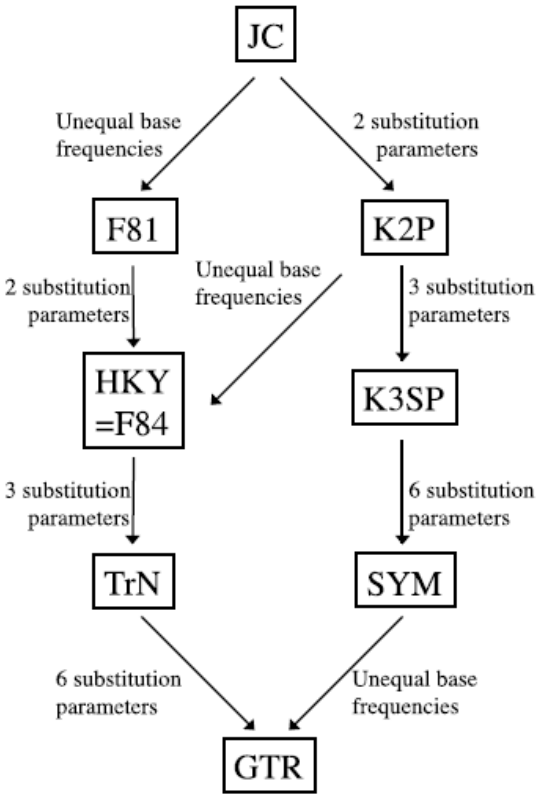
## Models of DNA evolution

The main idea is to start with a random DNA sequence placed at the root of a rooted, bifurcating phylogenetic tree. This sequence is then *evolved* along the tree: mutations are applied to the sequences along the edges of the tree, whereas speciation events are implicitly given by the tree as the bifurcations at internal nodes. There are no insertion or deletion events and thus all sequences produced under this model have the same length.

**Definition 3.4.1** (Jukes-Cantor model) *Let $T_0$ be a rooted phylogenetic tree, called the* model tree. *The* Jukes-Cantor *model of evolution makes the following assumptions:*

(i) *The possible states for each site are* A, C, G *and* T.
(ii) *The sequence length is an input parameter and the state of each site of the initial sequence at the root is drawn independently and uniformly at random from the set of possible states.*
(iii) *The sites evolve* identically and independently *(that is, all sites evolve following the same rules, but independently of each other) along the edges of the tree $T_0$ from the root $\rho$ at a fixed rate $u$.*
(iv) *With each edge $f \in E$ we associate a duration $\tau(f)$ and the expected number of mutations per site along $f$ is given by $u\tau(f)$. The probabilities of change to each of the three remaining states are equal.*

In summary, the Jukes-Cantor model of DNA evolution assumes that all four bases (A, C, G and T) occur with equal frequencies (= 0.25) and that changes from one base to another occur at the same rate between all bases. There are many ways to relax these conditions to obtain more general models. For example, if we let the bases occur with different and arbitrary frequencies (although they have to sum to 1), and allow two different rates of change, one for transitions (that is, changes between A and G or between C and T) and a second one for transversions (all other changes), then we obtain the so-called *Hasegawa-Kishino-Yano model.*




| | |
|---|---|
| JC | "Jukes-Cantor" [144] |
| F81 | "Felsenstein 81" [73] |
| K2P | "Kimura 2-Parameter" [148] |
| K3SP | "Kimura 3-Parameter" [149] |
| HKY | "Hasegawa-Kishino-Yano" [103] |
| F84 | "Felsenstein 84" [78] |
| TrN | "Tamura-Nei" [232] |
| SYM | "Symmetric" [247] |
| GTR | "General Time Reversible" [158, 205] |

## The phylogenetic tree reconstruction problem

A main goal of phylogenetic analysis is to reconstruct the phylogenetic tree along which a given set of species has evolved. In nearly all cases, there is no way to verify whether a given phylogenetic tree represents the *true tree* along which sequences or species actually evolved. Thus, the problem is usually addressed indirectly, as follows.

First, a *model of sequence evolution* is developed for which one hopes that it covers the most important aspects of the evolution of the taxa under consideration. Such models may be a simple modification of the Jukes-Cantor model, obtained, for example, by adding a difference in nucleotide composition, say, or can be quite substantial modifications, obtained, for example, by adding dependencies between different sites in the sequence.

Then, a tree reconstruction method is used that is known to perform well for sequences that have evolved under the given model.

In practice, there is a trade-off that needs to be taken into account: a more simple model of evolution is usually computationally less expensive and requires less data to get a robust result, whereas a more elaborate model of evolution is usually computationally more expensive and requires more data. However, a more detailed model should provide more reliable results. On the other hand, the variance of the solution increases with more parameters.

An important property of any such model of evolution is that it be *identifiable*, meaning that, in the idealized situation that the sequences under consideration have infinite length, it must, in principle, allow one to determine precisely which model tree a given set of sequences was generated on. When identifiability fails, this is because the same set of sequences can be generated with the same probability on two different model trees. Simple models, such as the Jukes-Cantor model, are usually identifiable. As models become more realistic or complex, care must be taken not to lose identifiability. In current research, mathematicians are interested in designing models that are as general as possible, while still being identifiable.

How well any phylogenetic tree can be reconstructed from sequences depends on the length of the sequences provided. For identifiable models, an important concept is *statistical consistency*: a tree reconstruction method is called *statistically consistent* with respect to a given (identifiable) model of evolution, if it is guaranteed to correctly reconstruct the phylogenetic tree from sequences that evolved along that phylogenetic tree under the model, given long enough sequences.

- *Sequence-based methods* usually search for a phylogenetic tree $T$ that optimally explains a given multiple sequence alignment $M$.
- *Distance-based methods* usually construct a phylogenetic tree $T$ from a given distance matrix $D$.

## Sequence-based methods

The first main group of tree reconstruction methods are called *sequence-based methods*. In this type of approach, the input consists of a multiple sequence alignment $M$ on $\mathcal{X}$ and a phylogenetic tree $T$ is determined for $M$, usually by performing a search in tree space to find an optimal phylogenetic tree or trees.

## Maximum parsimony

The *maximum parsimony method* is one of the most widely used sequence-based tree reconstruction methods. In science, the principle of *maximum parsimony* can be stated as a preference for the least complex explanation for an observation. In phylogenetic analysis, the *maximum parsimony problem* is to find a phylogenetic tree that explains a given set of aligned sequences using a minimum number of *evolutionary events.*

In the simplest form of this approach, the evolutionary events to be minimized are nucleotide mutations. In this context, the *difference* between two aligned sequences $x = (x_1, \ldots, x_m)$ and $y = (y_1, \ldots, y_m)$ is given by the number of positions at which they differ,

$$\text{diff}(x, y) = |\{i \mid x_i \neq y_i\}|, \tag{3.6}$$

which is also known as the (unnormalized) *Hamming distance.*

**Definition 3.7.1** (Parsimony score of a tree) *Suppose we are given a multiple sequence alignment $M$ of length $m$ and a corresponding phylogenetic tree $T$ on $\mathcal{X}$.*

*The* parsimony score *of $T$ and $M$ is defined as:*

$$PS(T, M) = \min_{\alpha} \sum_{\{x,y\}} \text{diff}(x, y), \tag{3.7}$$

*where the minimum is taken over all possible assignments $\alpha$ of* hypothetical ancestor *sequences of length $m$ to the internal nodes of $T$ and the sum is taken over all pairs of sequences $x$, $y$ that are assigned to opposite ends of some edge $f$ of $T$.*

The task of computing $PS(T, M)$ is known as the *small parsimony problem*. This problem can be solved efficiently by the *Fitch algorithm* [79] if the phylogenetic trees considered are bifurcating and mismatches between different character states are all weighted equally. For trees with multifurcations, a generalization of this algorithm, called the *Fitch-Hartigan* algorithm [101] can be used. In even more general settings, for example, when changes between different states are weighted differently, Sankoff's algorithm can be applied [210] in a modification of (3.7).

The Fitch algorithm runs in time proportional to the length of the alignment times the number of taxa. The bottom-up pass computes the optimal score $PS(T, M)$. The top-down pass provides at least one labeling $\alpha$ that attains that score, but cannot be used to generate all possible optimal labellings.

**Definition 3.7.3** (Parsimony score of an alignment) *Given a multiple sequence alignment* $M = (a_1, \ldots, a_n)$ *on* $\mathcal{X}$, *its* parsimony score *is defined as*

$$PS(M) = \min\{PS(T, M) \mid T \text{ is a phylogenetic tree on } \mathcal{X}\}. \tag{3.8}$$

The task of computing $PS(M)$ is known as the *large parsimony problem*. Potentially, we need to consider all $(2n-5)!!$ possible unrooted phylogenetic trees. Unfortunately, it is most probably not possible to find a solution much faster, as the maximum parsimony problem is known to be NP-hard.

## Maximum likelihood estimation

Let $M = (a_1, \ldots, a_n)$ be a multiple sequence alignment of length $m$ on $\mathcal{X}$. The basic idea of the *maximum likelihood estimation (MLE)* method is to determine a phylogenetic tree $T$, together with edge lengths $\omega$, that maximizes the *likelihood*

$$L(T) = \mathrm{P}(M \mid T, \omega, \mathcal{M}) \tag{3.10}$$

of generating the given multiple sequence alignment $M$ on the phylogenetic tree $T$ with edge lengths $\omega$ under a given model of sequence evolution $\mathcal{M}$.

Such a model $\mathcal{M}$ specifies how to choose the initial sequence at the root of the tree $T$ and how sequences evolve along edges of the tree.

As with maximum parsimony, the main draw-back of this approach is that finding an optimal phylogenetic tree is NP-hard [50]. In fact, the situation may be in some sense worse: for a given phylogenetic tree $T$, the complexity of the problem of defining edge lengths $\omega$ on $T$ so as to maximize the likelihood $P(M \mid T, \omega, \mathcal{M})$ is currently unknown.

One can efficiently compute the likelihood for a given bifurcating, rooted phylogenetic tree $T$ with edge lengths $\omega$ using *Felsenstein's algorithm* [73].

## Bootstrap analysis

Both maximum parsimony and maximum likelihood methods aim at producing an optimal phylogenetic tree (or a group of optimal trees). This can be viewed as a single point estimate of the true phylogeny (as opposed to a distribution of trees of similarly high quality) and the question arises how to evaluate the robustness of the

different parts of the computed tree. A standard statistical technique for addressing this type of question is called *bootstrapping* [74].

In *nonparametric bootstrapping*, the general approach is to sample replicate datasets from the original input data, to apply the computational method under consideration to each of the replicate datasets and then to count how many times features of the original result are found in the results obtained for the replicates to estimate their support.

Bootstrapping provides a measure for assessing how well different portions of a phylogenetic tree are supported. A split that has a low bootstrap support is sensitive to the exact combination of characters in the original input dataset and is deemed statistically unreliable. In practice, a bootstrap support of at least 70%, say, is required for a split to be considered trustworthy.

## Bayesian methods

The posterior probability of $T$ can be obtained from the prior probability of $T$ with the help of the likelihood $P(M \mid T) = P(M \mid T, \omega, \mathcal{M})$ using Bayes' Theorem:

**Theorem 3.11.1** (Bayes' Theorem)

$$P(T \mid M) = \frac{P(M \mid T) \times P(T)}{P(M)}. \tag{3.18}$$

The posterior probability $P(T \mid M)$ can be interpreted as the probability that the tree $T$ is correct, given the data.

In Bayesian tree inference the goal is not to determine a single phylogentic tree of maximum posterior probability, but rather to compute a sample of phylogenetic trees according to the posterior probability distribution of trees. Such a sample of trees is then processed further, for example, to generate a single consensus tree (as defined in Section 3.17) and then to label the splits of such a tree by their posterior probabilities.

It is usually impossible to solve analytically the complete expression of the posterior probability.

To avoid this problem, Bayesian inference employs the *Markov chain Monte Carlo (MCMC)* approach, which is based on the idea of sampling from the posterior probability distribution using a suitably constructed chain of results. This is a general approach that is used in a wide range of applications.

In the context of phylogenetic tree reconstruction, the MCMC approach constructs a chain of phylogenetic trees [88, 104, 175]. At each step, a modification of the current phylogenetic tree is proposed and then a probabilistic decision is made whether to *accept* the newly proposed phylogenetic tree or whether to keep the current one.

One of the main attractions of Bayesian inference is that it produces a distribution of phylogenetic trees rather than a point estimate, in many cases using less time than a maximum-likelihood analysis followed by a bootstrap analysis. Areas of ongoing research are how to ensure convergence and also how to choose appropriate prior distributions.

## Distance-based methods

The second main group of tree reconstruction methods are called *distance-based methods*. In this type of approach, first a distance matrix $D$ is computed and then a phylogenetic tree $T$ is constructed from the distance matrix $D$.

**Definition 3.12.1** (Distance function) *A distance function on a set $\mathcal{X}$ is a function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{\geq 0}$ that has the following three properties:*

(i) Identity and separation: *For any two elements $x, y \in \mathcal{X}$ we have $d(x, y) = 0$ if and only if $x = y$.*
(ii) Symmetry: *For any two elements $x, y \in \mathcal{X}$ we have $d(x, y) = d(y, x)$.*
(iii) Triangle inequality: *For any three elements $x, y, z \in \mathcal{X}$ we have $d(x, z) \leq d(x, y) + d(y, z)$.*

In practice, empirically derived distances do not always satisfy the identity property, for example, when different species have exactly the same sequence for a given gene, and sometimes they do not even fulfill the triangle inequality. To avoid these issues, throughout this book we use the term *distance matrix* to mean any function for which at least the symmetry property holds, and usually also the triangle inequality. The term *metric* is only used for distance functions that have all three required properties.

Distance-based methods are quite popular because they can be used for many different types of data. It is often easy to define a distance measure for a novel type of data, even when it is difficult or impossible to develop an appropriate model of evolution. An added advantage over sequence-based methods is speed. The neighbor-joining algorithm (described below) can be applied to distance matrices on thousands of taxa in reasonable time.

When sequence data is available, trees built using distance-based methods are often considered only a first, fast approximation, and more elaborate sequence-based methods are then used to obtain a more trusted phylogeny.

Let $M$ be a multiple alignment of DNA sequences on $\mathcal{X}$. The simplest approach is to use the *Hamming distance* $\mathrm{H}(a, b)$ (also called *observed p-distance*), defined as the proportion of positions at which two aligned sequences $a$ and $b$ differ. To be more precise, this quantity is often referred to as the *normalized Hamming distance*. The *unnormalized Hamming distance* is defined as the raw number of positions at which two aligned sequences differ.

Note that the Hamming distance between two sequences *underestimates* their true evolutionary distance (average number of mutations that took place per site over the elapsed time), as back mutations and multiple mutations at the same position are not counted. To rectify this, a correction formula based on some model of evolution is often used.

For example, in the case of the Jukes-Cantor model, by inverting the probability-of-change formula, we obtain the so-called *Jukes-Cantor transformation*:

$$JC(a, b) = -\frac{3}{4} \ln \left( 1 - \frac{4}{3} H(a, b) \right). \tag{3.24}$$

So, if we are given a multiple alignment of sequences that we believe evolved according to the Jukes-Cantor model of evolution, to compute a distance matrix that approximates the true evolutionary distances, we first determine the (normalized) Hamming distance between any two sequences $a$ and $b$ and then apply this transformation to get a corrected value.

Let $T$ be a phylogenetic tree on $\mathcal{X}$ with edge lengths $\omega$.

The **tree distance** $D_T(a, b)$ between any two taxa $a, b \in \mathcal{X}$ is the sum of lengths $\omega(e)$ of all edges $e$ on the unique path from the leaf labeled $a$ to the leaf labeled $b$.

The **tree length** of $T$ is the sum of lengths over all edges in $T$.

Let $D$ be a distance matrix on $\mathcal{X}$.

We say that $D$ is an **additive distance** (or **tree-like**)
if there exists a phylogenetic tree $T$ such that $D = D_T$ .

We say that $D$ satisfies the **four-point condition** if for every $w, x, y, z \in \mathcal{X}$

$$wx + yz \leq \max\{wy + xz, wz + xy\}$$

**Fact** $D$ is additive if and only if it satisfies the four-point condition.

We say that $D$ satisfies the **three-point condition** if for every $x, y, z \in \mathcal{X}$

$$xy = yz = xz \qquad \text{or} \qquad ij < ik = jk, \quad \{i, j, k\} = \{x, y, z\}$$

or, equivalently, the **strong triangle inequality** if for every $x, y, z \in \mathcal{X}$

$$xz \leq \max\{xy, yz\}$$

**Dim.** ($\Rightarrow$) Let $x, y, z \in \mathcal{X}$. Then if $xy = yz = xz$ , we are done.

Suppose $xz < xy = yz$ . Then it is clear that

$$xz \leq \max\{xy, yz\}$$
$$xy \leq yz = \max\{yz, xz\}$$
$$yz \leq xy = \max\{xy, xz\}$$

($\Leftarrow$) Let $x, y, z \in \mathcal{X}$. By hypothesis we know that

$$xz \leq \max\{xy, yz\}$$
$$xy \leq \max\{xz, yz\}$$
$$yz \leq \max\{xy, xz\}$$

It is clear that if one of the previous inequalities holds as equal, then there must be another one that holds as equal as well. Therefore, the third can also holds as equal or as strictly less.

In biology, the *molecular clock hypothesis* states that the mutation rate is constant over all sites of the sequences and over all edges of the model tree.

Under this assumption, the leaves of the model tree all have the same distance from the root. In this case the tree is called an *ultrametric tree* and any distance matrix obtained from such a tree is called an *ultrametric*.

**Fact** $D$ is an ultrametric if and only if it satisfies the three-point condition (or equivalently the strong triangle inequality).

## UPGMA and Neighbor-Joining

UPGMA stands for *unweighted pair group method using arithmetic averages* [220]. Given a distance matrix $D$ on $\mathcal{X}$, UPGMA produces a rooted phylogenetic tree $T$ with edge lengths.

The method operates by clustering the given taxa, at each stage merging two clusters and at the same time creating a new node in the tree. The tree is assembled bottom-up, first clustering pairs of leaves, then pairs of clustered leaves, etc. Each node is given a height and the length of an edge is obtained as the difference of heights of its two end nodes.

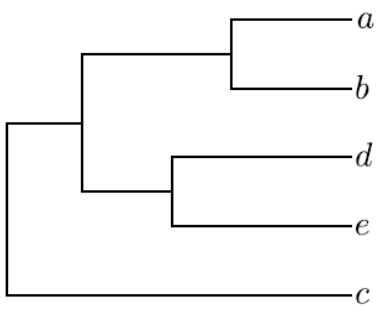
We define the distance $d(i, j) = d(C_i, C_j)$ between two disjoint clusters $C_i \subset \mathcal{X}$ and $C_j \subset \mathcal{X}$ as the average distance between pairs of taxa from each cluster:

$$d(i, j) = \frac{1}{|C_i||C_j|} \sum_{x \in C_i, y \in C_j} d(x, y). \tag{3.26}$$

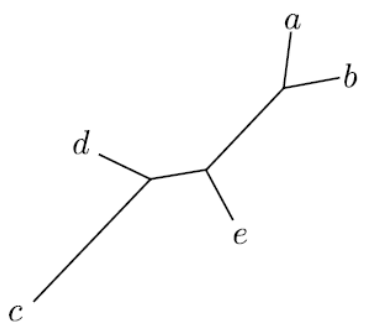
The UPGMA algorithm repeatedly merges two closest clusters until only one cluster remains, computing a phylogenetic tree $T$ in $O(n^3)$ steps, with $n = |\mathcal{X}|$.

| | $a$ | $b$ | $c$ | $d$ | $e$ |
|---|---|---|---|---|---|
| $a$ | 0 | 2 | 7 | 5 | 4 |
| $b$ | 2 | 0 | 7 | 5 | 4 |
| $c$ | 7 | 7 | 0 | 4 | 5 |
| $d$ | 5 | 5 | 4 | 0 | 3 |
| $e$ | 4 | 4 | 5 | 3 | 0 |

(a) Distances




(b) UPGMA tree




(c) Neighbor-joining tree

Given a distance matrix $D$ on $\mathcal{X}$, the *neighbor-joining method* produces an unrooted phylogenetic tree $T$, with edge lengths $\omega$ [208, 229]. It is more widely applicable than UPGMA, as it does not assume a molecular clock.

The neighbor-joining algorithm is a modification of the UPGMA algorithm. Both algorithms are agglomerative methods that repeatedly decide which two clusters to join so that their nodes are "neighbors" in the resulting phylogenetic tree. In UPGMA, this decision is based on the current distances and two clusters of smallest distance are chosen. This works correctly, when the distances come from an ultrametric tree, because then closest nodes are indeed also neighbors. In a more general setting, two clusters or nodes may be separated by only a short distance without being true neighbors, see Figure 3.20, and in this case, UPGMA produces an incorrect tree.

To avoid this problem, the neighbor-joining algorithm subtracts the average distance (almost) of each cluster to all other clusters to compensate for the effect of large distances.

There are two further differences between neighbor-joining and UPGMA, namely how the distance matrix is updated after merging two clusters and how the lengths of edges are set.

**Facts** Let $D$ be a distance matric on $\mathcal{X}$.

Then UPGMA computes a (rooted) tree $T$ such that $D_T = D$ if and only if $D$ is ultrametric.

And NJ computes an (unrooted) tree $T$ such that $D_T = D$ if and only if $D$ is additive.

Distance matrices considered in practice rarely fulfill the three-point or four-point condition. In such cases neighbor-joining, and, to a lesser extent, also UPGMA, is nevertheless applied and the hope is that the deviations from the required conditions will have only a small influence on which phylogenetic tree is computed by the algorithm.

## Balanced minimum evolution

Let $D$ be a distance matrix on $\mathcal{X}$ and assume that we have some unrooted, bifurcating phylogenetic tree $T$ on $\mathcal{X}$. Within the BME setting, every edge $e$ is assigned a *balanced edge length* $\omega(e)$ that is computed from the *balanced average distances* between the taxa that span the edge.

This averaged distance is called *balanced* because two subsets such as $B_1$ and $B_2$ enter the calculation with equal weight, regardless of how many taxa they contain, in contrast, say, to the definition of distances between clusters used in the UPGMA

**Definition 3.15.3** (Balanced minimum evolution tree) *For a given distance matrix $D$ on $\mathcal{X}$, determine a phylogenetic tree $T$ on $\mathcal{X}$, with balanced edge lengths $\omega$ from $D$, for which the tree length $l(T)$ is minimum over all such trees. We call such a tree $T$ a* balanced minimum evolution tree *(or BME tree) for $D$.*

Computing the BME tree is a statistically consistent tree reconstruction method [58]. Unfortunately, the problem of finding an optimal balanced minimum evolution tree is believed to be NP-hard. So we need to turn to heuristics.

Interestingly, the neighbor-joining algorithm is in fact a greedy heuristic for computing the BME tree [84].

We now turn to the FastME heuristic. The algorithm has two phases. In the first phase, an initial phylogenetic tree is built in a stepwise fashion. In the second phase, the tree is iteratively improved using NNI operations until no further improvement can be attained:

The balanced minimum evolution approach can be viewed as an improvement over the classical *minimum evolution* approaches. Let $D$ be a distance matrix on $\mathcal{X}$. In the *ordinary least squares* approach, for a given unrooted phylogenetic tree $T$, edge lengths $\omega$ are computed that provide a least squares fit to the distance matrix $D$. The method then returns such a tree of shortest length. A weakness of this method is that it explicitly assumes that all distances have the same variance, which is usually not true, as the variances of larger distances tend to be larger. To address this problem, in the *weighted least squares approach*, estimates for variances are taken into account [76, 81].

In the BME approach, the variances of all pairwise distances are not assumed to be constant, but rather they are taken to depend on the (topological) distance between taxa. The tree length formula Equation (3.39) implies that large (topological) distances have low weight. BME can be interpreted as a weighted least squares approach and the FastME algorithm runs substantially faster than all previous weighted least square approaches.

## The least squares method

from [Felsenstein2004]

The fundamental idea of distance matrix methods is that we have an observed table (matrix) of distances ($D_{ij}$), and that any particular tree that has branch lengths leads to a predicted set of distances (which we will denote the $d_{ij}$). It does so by making the prediction of the distance between two species by adding up the branch lengths between the two species. Figure 11.1 shows a tree and the distance matrix that it predicts. We also have a measure of the discrepancy between the observed and the expected distances. The measure that is used in the least squares methods is

$$Q = \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}(D_{ij} - d_{ij})^2 \qquad (11.1)$$

where the $w_{ij}$ are weights that differ between different least squares methods. Cavalli-Sforza and Edwards (1967) defined the unweighted least squares method in which $w_{ij} = 1$. Fitch and Margoliash (1967) used $w_{ij} = 1/D_{ij}^2$, and Beyer et al. (1974) suggested $w_{ij} = 1/D_{ij}$. We are searching for the tree topology and the branch lengths that minimize $Q$. For any given tree topology it is possible to solve for the branch lengths that minimize $Q$ by standard least squares methods.

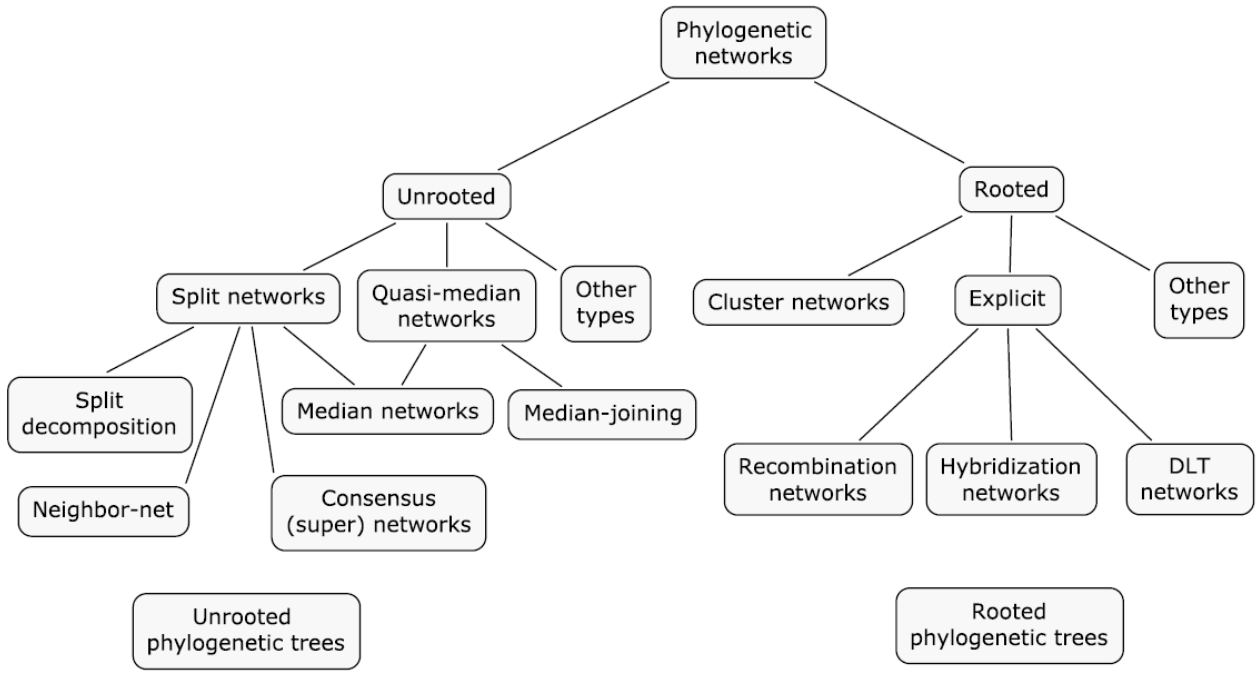
In the minimum evolution method the tree is fit to the data, and the branch lengths are determined, using the unweighted least squares method. The least squares trees are determined for different topologies, and the choice is made among them by choosing the one of shortest total length. Thus this method makes partial use of the least squares criterion. In effect it uses two criteria at the same time, one for choosing branch lengths, another for choosing the tree topology.

## Phylogenetic networks

Phylogenetic networks provide an alternative to phylogenetic trees and may be more suitable for datasets whose evolution involve significant amounts of reticulate events caused by hybridization, horizontal gene transfer, recombination, gene conversion or gene duplication and loss [56, 61, 89, 201, 219, 231]. Moreover, even for a set of taxa that have evolved according to a tree-based model of evolution, phylogenetic networks can be usefully employed to explicitly represent conflicts in a dataset that may, for example, be due to mechanisms such as incomplete lineage sorting or to inadequacies of an assumed evolutionary model [125].

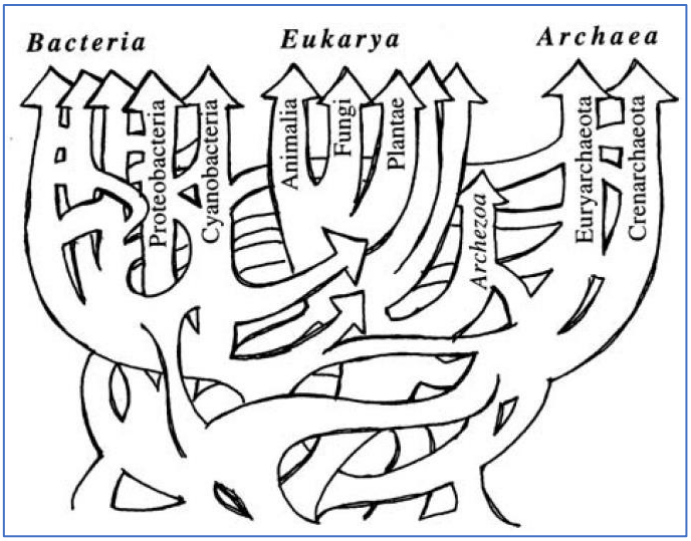
While rooted phylogenetic networks can, in theory, be used to explicitly describe evolution in the presence of reticulate events, their calculation is difficult and computational methods for doing so have not yet matured into practical and widely used tools [24, 98, 106, 127, 225, 237]. In contrast, there are a number of established tools for computing unrooted phylogenetic networks, which can be used to visualize incompatible evolutionary scenarios in phylogeny and phylogeography [9, 10, 11, 32, 52, 122, 125].

In practice, most currently available algorithms for computing phylogenetic networks are based on combinatorics



**Definition 4.2.1** (Phylogenetic network) *A* phylogenetic network *is any graph used to represent evolutionary relationships (either abstractly or explicitly) between a set of taxa that labels some of its nodes (usually the leaves).*

The envisioned role of rooted phylogenetic networks in biology is to describe the evolution of life in a way that explicitly includes reticulate events. Ultimately, the main goal is to work out the details of a rooted phylogenetic *network of life*, such as the one proposed by Ford Doolittle [61], see Figure 4.2.




Phylogenetic networks can be used in two different ways: The first type of usage is as a tool for visualizing incompatible datasets in a fruitful manner, in which case we speak of an *abstract* phylogenetic network. The second type of usage is as a representation of a putative evolutionary history involving reticulate events, in which case the network is called *explicit*.

By definition, most (if not all) types of *unrooted* phylogenetic networks are abstract networks, as evolution in inherently rooted (and thus any unrooted phylogenetic *tree* is also abstract in this sense). However, *rooted* phylogenetic networks can be either abstract or explicit, depending on how they are constructed and interpreted.

The necessity of distinguishing between abstract and explicit networks was pointed out in [180]. They are called *implicit* and *explicit* in [123]. In [181], abstract and explicit networks are named *data-display* networks and *evolutionary* networks, respectively.

Phylogenetic networks can be computed from a wide range of data, including multiple sequence alignments, distance matrices, sets of trees, clusters, splits, rooted triplets or unrooted quartets.

Unrooted phylogenetic networks

- split networks
  - from distances
    - split decomposition
    - **neighbor-net**
  - from trees
    - **consensus (super) split networks**
  - from sequences
    - median networks
  - from quartets
    - quartet-net
- quasi-median networks
  - median networks
  - **median joining**
- others
  - haplotype networks
  - reticulograms

Rooted phylogenetic networks

- from clusters
  - hardwired
    - cluster networks
  - softwired
    - galled trees
    - level-k networks
    - galled networks
- explicit networks
  - hybridization networks
  - recombination networks
  - DLT networks
- others
  - reassortment networks
  - networks from multi-labeled trees
  - networks from rooted triples

Exactly how does a rooted phylogenetic network $N$ on $\mathcal{X}$ *represent* a cluster? There are two different answers to this question. We say that a network $N$ represents a given cluster $C$ on $\mathcal{X}$ in the *hardwired* sense, if there exists a tree edge $e$ in $N$ such that the set of labels of leaves below $e$ equals $C$. Alternatively, we say that $N$ represents $C$ in the *softwired* sense, if there exists a rooted phylogenetic tree $T$ that is contained in $N$ and represents $C$ (in the hardwired sense).

## Networks used in practice

For unrooted phylogenetic networks, most of the methods mentioned are routinely used in phylogenetic analysis or phylogeography, in particular, neighbor-net, consensus split (super) networks and median-joining, given distances, trees or sequences, respectively.

This is not the case for rooted phylogenetic networks. While a number of algorithms have been described for computing rooted phylogenetic networks, there are some problems to overcome. First, for many of the algorithms there exist only proof-of-concept implementations that are not designed to be used as tools in real studies. Second, the computational problems are often hard and the algorithms have impractical running times. Third, the calculation of rooted phylogenetic networks must be more closely linked to detailed biological models of reticulate evolution so as to produce more plausible results.

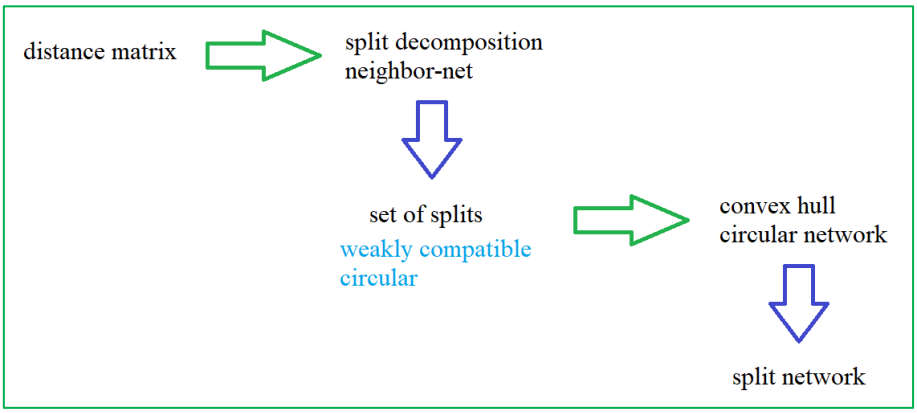
At present, none of the existing methods for computing a rooted phylogenetic method is widely or routinely used as a tool to help understand the evolutionary history of a given set of taxa in terms of mutations, speciations and specific types of reticulate events. While rooted phylogenetic networks are conceptually very appealing, the development of suitable methods for their computation remains a formidable challenge.

## Split networks

The foundation for split networks was laid in [9]. Let $\mathcal{X}$ be a set of taxa and assume that we are given a set of splits $\mathcal{S}$ on $\mathcal{X}$, usually together with a *weighting* that assigns a non-negative weight to each split, which may represent character changes distance, or may also have a more abstract interpretation. If the set of splits $\mathcal{S}$ is compatible, then it can be represented by an unrooted phylogenetic tree and each edge in the tree corresponds to exactly one of the splits. More generally, $\mathcal{S}$ can always be represented by a *split network*, which is an unrooted phylogenetic network with the property that every split $S$ in $\mathcal{S}$ is represented by an array of parallel edges in $N$ (see Section 5.5).

A split network $N$ can be obtained from a number of different types of data. To be more precise, the algorithms mentioned below do not compute a split network directly, but rather they all compute a set of weighted splits $\mathcal{S}$. A split network $N$ is then computed using the *convex hull algorithm* or *circular network algorithm*,

## Split networks from distances




The split decomposition method takes as input a distance matrix $D$ on $\mathcal{X}$ and produces as output a set of weighted splits $\mathcal{S}$ on $\mathcal{X}$ that is *weakly compatible*, a property that ensures that the corresponding split network will not be too complicated (see Section 5.8). In practice, the split decomposition method is a very conservative method, in the sense that a split will only be present in the output if there is global support for it in the given dataset. For large or diverse datasets, the method tends to exhibit very low resolution and thus its use is limited to small datasets of less than 100 taxa, say.

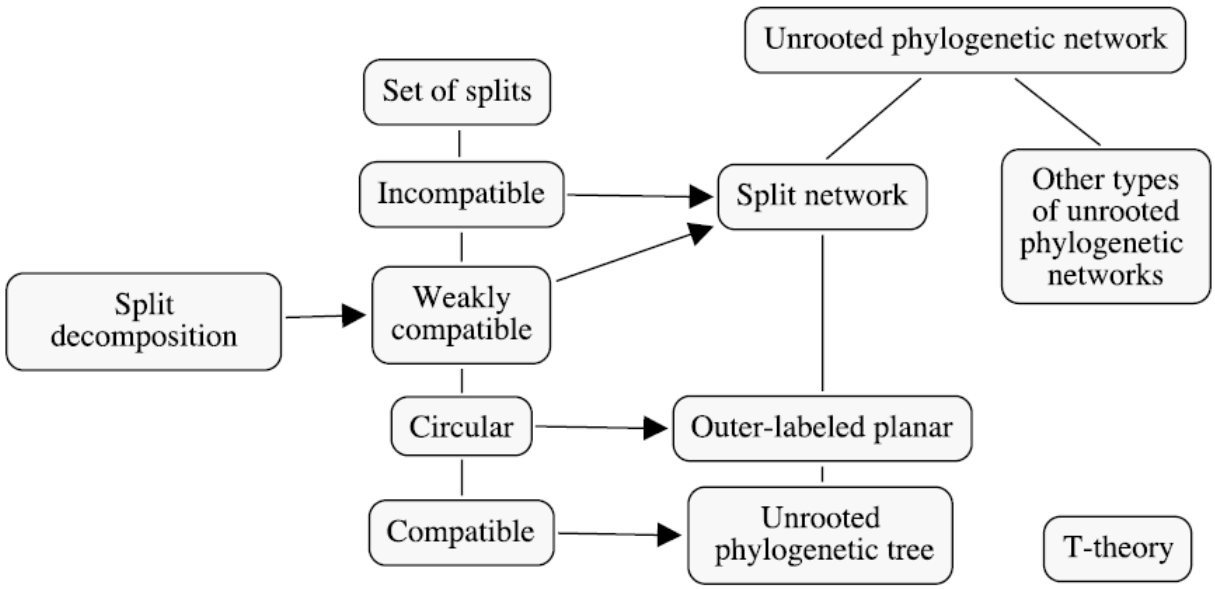
The neighbor-net method takes as input a distance matrix $D$ on $\mathcal{X}$ and produces as output a set of weighted splits $\mathcal{S}$ on $\mathcal{X}$ that is *circular*, which implies that it can be represented by an *outer-planar* split network (see Section 5.7), if used together with the circular network algorithm (see Section 7.21). The neighbor-net method is more popular than the split decomposition method because it is less conservative and so does not lose resolution on larger datasets. Moreover, the fact that the output of the method can always be represented by a planar split network and is thus easy to visualize adds to its attraction, see Figure 4.3.

Both network methods have the nice property that they produce a tree when given tree-like data.

## Splits and unrooted phylogenetic networks

The concept of a *split* plays an important role in the mathematics of phylogeny. It is motivated by the simple, but crucial, observation that every edge $e$ in an unrooted phylogenetic tree $T$ defines a bipartition of the underlying taxon set $\mathcal{X}$ into two non-empty and disjoint subsets, $A$ and $B$, known as a *split*. The splits of an unrooted phylogenetic tree uniquely define the topology of the tree and splits are used, for example, to compare different trees or to compute consensus trees. Any set of splits that is *compatible* corresponds to a phylogenetic tree and so one possible way to generalize from trees to networks is to consider sets of splits that are *incompatible*.

Splits provide the basis of unrooted phylogenetic trees and a large class of unrooted phylogenetic networks, namely *split networks*, just as clusters provide the basic building blocks for rooted phylogenetic trees and networks (see Chapter 6). The foundation for the theory of split networks was laid down in a seminal paper by Bandelt and Dress [9].



## Splits

> Splits and clusters are closely related concepts. While clusters *group* taxa to emphasize their common features, splits *divide* taxa to emphasize their distinctive features.

A **split** is a partition of $\mathcal{X}$ into two non-empty subsets $A, B \subseteq \mathcal{X}$ such that $A \cap B = \emptyset$ and $A \cup B = \mathcal{X}$.

A **weighted split** is a split $S$ that has been assigned a weight $\omega(S) \geq 0$.

We use the notations: $A \mid B$ or $\frac{A}{B}$.
Notice that there is no ordering in a split, so $A \mid B = B \mid A$.
We call $A$ and $B$ the **parts** of the split.

The **size** of a split $S = A \mid B$ is the minimal cardinality of its parts

$$\text{size}(S) \coloneqq \min\{|A|, |B|\}$$

A **trivial split** is a split of size one.

For any taxon $x \in \mathcal{X}$, we denote with $S(x)$ the split part that contains $x$ and with $\bar{S}(x)$ the other part.

Let $\mathcal{S}$ be a set of splits and $\mathcal{X}' \subset \mathcal{X}$ a subset of taxa.
The set of **splits induced** by $\mathcal{X}'$ or **restriction** of $\mathcal{S}$ to $\mathcal{X}'$ is the set

$$\mathcal{S}|_{\mathcal{X}'} \coloneqq \left\{ \frac{A \cap \mathcal{X}'}{B \cap \mathcal{X}'} \;:\; \frac{A}{B} \in \mathcal{S},\ \begin{matrix} A \cap \mathcal{X}' \neq \emptyset, \\ B \cap \mathcal{X}' \neq \emptyset \end{matrix} \right\}$$

that is the splits restricted such that they remain splits.

Any edge $e$ of a phylogenetic tree $T$ defines a split of the underlying taxon set $\mathcal{X}$ : deleting $e$ produces two subtrees, and their taxon labels form the parts of the split.
In fact, since every leaf in a phylogenetic tree is label by some taxon, the two parts are non-empty (the two subtrees must contains some leaves); and, since each taxon occurs precisely once, it follows that the two parts are disjoint and cover all the taxa.
We denote with $\sigma_T(e)$ the split **represented** by the edge $e$.

If the edges of the tree have lengths/weights,
then these can be assigned to the corresponding split.

If $T$ does not contain any unlabeled nodes of degree two, then any two different edges $e$ and $f$ always represent two different splits, that is, $\sigma_T(e) \neq \sigma_T(f)$ must hold. The only situation in which a phylogenetic tree can contain an unlabeled node of degree 2 is when it has a root with outdegree two. In this case, the two edges $e$ and $f$ that originate at the root $\rho$ give rise to the same split $\sigma_T(e) = \sigma_T(f)$, but to two complementary clusters.

A common construction to avoid this special case at the root $\rho$ is to attach an additional leaf to $\rho$ that is labeled by a special taxon $o$, which we call a (formal) *outgroup*. Then the root of a phylogenetic tree is specified as the node to which the leaf edge of $o$ attaches. All phylogenetic trees that are discussed in this chapter are unrooted. However, by using this *outgroup trick* much of what we discuss concerning unrooted trees can also be adapted to rooted phylogenetic trees.

Let $T = (V, E)$ be an unrooted phylogenetic tree.

The **split encoding** of $T$ is the set of all splits represented by its edges

$$\mathcal{S}(T) := \{\, \sigma_T(e) : e \in E \,\}$$

A tree can be uniquely reconstructed from its split encoding.

A tree $T$ **represents** a set of splits $\mathcal{S}$ if $\mathcal{S}(T) = \mathcal{S}$ .

**Algorithm 5.2.2** (Splits from tree) *The set $\mathcal{S}(T)$ of all splits associated with an unrooted phylogenetic tree $T$ on $\mathcal{X}$ can be computed as follows:*

(i) *Choose a start leaf $\rho$ and assume that all edges of $T$ are directed away from $\rho$.*
(ii) *In a postorder traversal of $T$, for each node $v$ compute the set $L(v)$ of taxon labels that are encountered in the subtree rooted at $v$.*
(iii) *For each edge $e = (u, v)$ of $T$, add the split $\sigma(e) = \frac{L(v)}{\mathcal{X} - L(v)}$ to $\mathcal{S}(T)$.*

This algorithm is quadratic in time.

**Compatibility**

Two splits $S_1 = A_1 \mid B_1$ and $S_2 = A_2 \mid B_2$ are **compatible** if one of the following intersections is empty

$$A_1 \cap A_2\,, \qquad A_1 \cap B_2\,, \qquad A_2 \cap B_1\,, \qquad B_1 \cap B_2$$

Two splits that are not compatible are **incompatible**.
A set of splits is **compatible** if all the splits are (pairwise) compatible.

**Fact** Let $\mathcal{S}$ be a set of splits on $\mathcal{X}$ that contains all trivial splits on $\mathcal{X}$.

There exists a unique unrooted phylogenetic tree $T$ that realizes $\mathcal{S}$ if and only if $\mathcal{S}$ is compatible.

We can drop the hypothesis on trivial splits if we consider $\mathcal{X}$-trees.

**Definition 5.3.3** (Incompatibility graph) *The* incompatibility graph $IG(\mathcal{S})$ *of a set of splits* $\mathcal{S}$ *is the graph* $(V, E)$ *that has node set* $V = \mathcal{S}$ *and edge set* $E = \{\{S_1, S_2\} \mid S_1 \text{ and } S_2 \text{ are incompatible}\}$.

A split $S \in \mathcal{S}$ is compatible with all other splits in $\mathcal{S}$, if and only if it is an isolated node in the incompatibility graph. In consequence, a set of splits $\mathcal{S}$ is compatible if and only if the incompatibility graph $IG(\mathcal{S})$ has no edges.

## Splits and clusters

splits -> clusters

To obtain a set of clusters $\mathcal{C}$ from a set of splits $\mathcal{S}$ on $\mathcal{X}$, we must first choose an *outgroup* taxon $o \in \mathcal{X}$. Then, for each split $S = \frac{A}{B}$ in $\mathcal{S}$, we define the cluster $C$ associated with $S$ to be the split part that does not contain $o$, that is, we set $C = \bar{S}(o)$. We usually also consider $\{o\}$ as a cluster, to ensure that all trivial clusters are present in $\mathcal{C}$.

**Teo.** $\mathcal{S}$ is compatible if and only if $\mathcal{C}$ is compatible.

Let $\mathcal{S}$ be a set of splits on $\mathcal{X}$. If $\mathcal{S}$ is compatible, then there exists an unrooted phylogenetic tree $T$ that represents $\mathcal{S}$. In this case, an alternative method to define the associated set of clusters $\mathcal{C}$ is to choose a root $\rho$ for $T$ and then to let $\mathcal{C}$ be the set of all clusters represented by the rooted version of $T$. If we choose a node of $T$ to be the root, then there is a simple one-to-one correspondence between the splits and clusters. On the other hand, if the root is chosen so as to subdivide some edge $e$ of $T$, then the split $S = \frac{A}{B}$ associated with $e$ gives rise to precisely two clusters, namely $A$ and $B$.

clusters -> splits

Now let us look at the opposite problem of defining a set of splits $\mathcal{S}$ for a given set of clusters $\mathcal{C}$ on $\mathcal{X}$. For a given cluster $C$, we could simply define the associated split $S$ as $C$ versus the complement of $C$, that is, as $S = \frac{C}{(\mathcal{X}-C)}$. Unfortunately, this assignment does not preserve incompatibilities. For example, the clusters $\{a, b\}$ and $\{b, c\}$ on $\mathcal{X} = \{a, b, c\}$ are incompatible, whereas the two splits associated in the manner suggested, $\frac{\{a,b\}}{\{c\}}$ and $\frac{\{a\}}{\{b,c\}}$, are not. To address this problem, we add a new (formal) *outgroup* taxon $o \notin \mathcal{X}$ that is then always placed in the split part that contains the complement of a cluster. In other words, for every cluster $C$ we define the associated split as $S = \frac{C}{(\mathcal{X}-C)\cup\{o\}}$ on $\mathcal{X}' = \mathcal{X} \cup \{o\}$.

In the special case that the set of clusters $\mathcal{C}$ is compatible, and thus corresponds to some rooted phylogenetic tree $T$, we can obtain the set of associated splits directly from $T$ as $\mathcal{S}(T)$, after first unrooting the tree.

The number of all splits on a set of $n$ taxa is $2^{n-1} - 1$
(the number of clusters is the same, off by one).

Let $\mathcal{S}$ be a set of splits on $\mathcal{X}$. If $\mathcal{S}$ is incompatible, then there are two basic computational problems that are sometimes of interest. The first problem is to remove a minimum number of splits such that the remaining set of splits is compatible:

**Problem 5.4.4** (Maximum compatibility problem) *Determine a maximum-size subset of splits $\mathcal{S}' \subset \mathcal{S}$ that is compatible.*

The second problem is to remove a minimum number of taxa such that the set of splits induced on the remaining taxa is compatible:

**Problem 5.4.5** (Maximum compatible subset problem) *Determine a maximum-size subset of taxa $\mathcal{X}' \subset \mathcal{X}$ such that the set of splits $S|_{\mathcal{X}'}$ induced on $\mathcal{X}'$ is compatible.*

It follows from the NP-completeness of the two analogous problems formulated for clusters in Section 6.2.1 that these two problems are NP-complete.

## Split networks

As we have seen, any set of compatible splits (containing all trivial splits) corresponds to an unrooted phylogenetic tree on $\mathcal{X}$. In this section, we introduce a mathematical generalization of the concept of an unrooted phylogenetic tree, called a *split network*, which can be used to represent an arbitrary, in particular, incompatible, set of splits.

In a split network, we use one or more edges to represent a split. The set of edges used to represent a given split $S$ has the property that deletion of all these edges produces exactly two connected components, and, as in the case of phylogenetic trees, the two parts of $S$ are given by the sets of taxa that occur as labels of one component, or the other, respectively.

**Lemma 5.5.4** (Split networks and compatiblity) *A set of splits $\mathcal{S}$ on $\mathcal{X}$ is compatible if and only if there exists a split network $N$ representing $\mathcal{S}$ that is a tree.*

the split network associated with a set of splits is not uniquely defined and also a split network representation of a compatible set of splits need not necessarily be a tree, although a tree representation for a compatible set of splits always exists. **Convex hull algorithm and circular network algorithm** for constructing a split network for a given set of splits produce a tree, when run with a compatible set of splits as input.

## Canonical split network

For any set of splits $\mathcal{S}$ there exists a unique split network that represents $\mathcal{S}$, called the **canonical split network** or **Buneman graph** associated with $\mathcal{S}$.

If $\mathcal{S}$ is compatible, then the Buneman graph is an unrooted phylogenetic tree.

The number of nodes and edges of the Buneman graph is exponential in the number of splits, in the worst case.

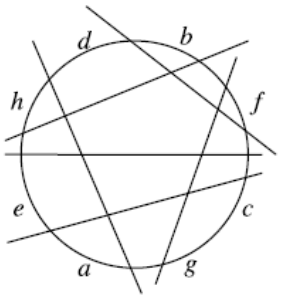
The network computed by the convex hull algorithm is the Buneman graph.

## Circular splits

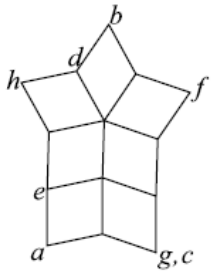
Informally, a set of splits $\mathcal{S}$ on $\mathcal{X}$ is called *circular*, if the taxa in $\mathcal{X}$ can be placed around a circle in such a way that each split $S = \frac{A}{B}$ can be realized by a line through the circle that separates the plane into two half-planes, one containing all taxa in $A$ and the other containing all taxa in $B$ (see Figure 5.9). An example of a non-circular set of splits and the corresponding split network is shown in Figure 5.10.

$\{a,b,d,e,h\} \mid \{c,f,g\}$
$\{a,c,d,e,g,h\} \mid \{b,f\}$
$\{a,c,e,g\} \mid \{b,d,f,h\}$
$\{a,c,g\} \mid \{b,d,e,f,h\}$
$\{a,c,e,f,g\} \mid \{b,d,h\}$
$\{a,e,h\} \mid \{b,c,d,f,g\}$

(a) Circular splits




(b) Circular ordering

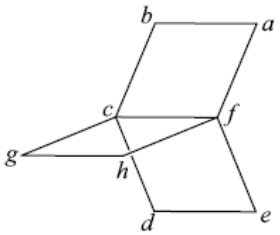



(c) Planar network

Figure 5.9 (a) A set of six circular splits $\mathcal{S}$ on $\mathcal{X} = \{a, b, \ldots, h\}$. (b) An arrangement of the taxa around a circle such that every split $S = A \mid B \in \mathcal{S}$ can be realized by a straight line through the circle that separates the two split parts $A$ and $B$. A circular ordering is given by $(a, g, c, f, b, d, h, e)$. (c) An outer-labeled planar split network representing $\mathcal{S}$.

$\{a,b\} \mid \{c,d,e,f,g,h\}$
$\{a,b,c,d,e,f\} \mid \{g,h\}$
$\{a,b,c,f,g,h\} \mid \{d,e\}$
$\{a,e,f,h\} \mid \{b,c,d,g\}$

(a) Non-circular splits




(b) Non-planar network

Figure 5.10 (a) A set of four non-circular splits $\mathcal{S}$ on $\mathcal{X} = \{a, b, \ldots, h\}$. (b) A non-planar split network representing $\mathcal{S}$.

**Definition 5.7.1** (Circular splits) *A set of splits $\mathcal{S}$ on $\mathcal{X}$ is called* circular, *if there exists a linear ordering $(x_1, \ldots, x_n)$ of the elements of $\mathcal{X}$ for $\mathcal{S}$ such that each split $S \in \mathcal{S}$ has the form*

$$S = \frac{\{x_p, x_{p+1}, \ldots, x_q\}}{\mathcal{X} - \{x_p, x_{p+1}, \ldots, x_q\}},$$

*for appropriately chosen $1 < p \leq q \leq n$.*

We call such an ordering $(x_1, \ldots, x_n)$ a *circular ordering* for $\mathcal{S}$

**Definition 5.7.4** (Outer-labeled planar) *Let $G$ be a graph in which some of the nodes are labeled. We call $G$* outer-labeled planar, *if there exists a drawing of $G$ in the plane such that no two edges intersect and all labeled nodes lie on the outside of the graph.*

With this definition we have [62]:

**Theorem 5.7.5** (Circular implies outer-labeled planar) *A set of splits $\mathcal{S}$ on $\mathcal{X} = \{x_1, \ldots, x_n\}$ is circular if and only if it can be represented by a split network $N$ that is outer-labeled planar.*

**Exercise 5.7.6** (Splits from one or two trees) *the set of splits obtained from a single unrooted phylogenetic tree is always circular.*

*the set of splits taken from two unrooted phylogenetic trees is not necessarily circular.*

**Lemma 5.7.7** (Circular splits have quadratic-size network) *If $\mathcal{S}$ is a set of $m$ circular splits on $\mathcal{X}$, then the number of nodes and edges in an outer-labeled planar split network $N$ representing $\mathcal{S}$ is at most quadratic in $m$.*

the canonical split network (Buneman graph) $N$ associated with a set of splits $\mathcal{S}$ on $\mathcal{X}$ is not necessary planar when $\mathcal{S}$ is circular.

## Weak compatibility

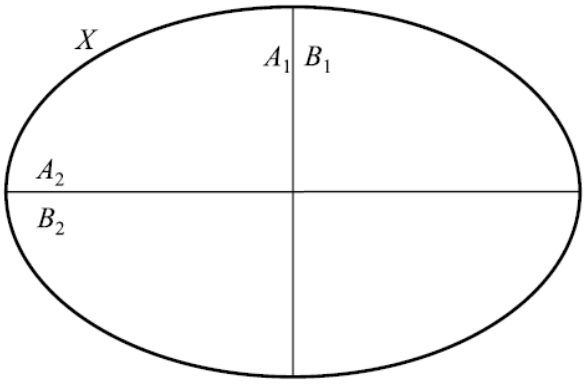
Three splits $S_1 = \frac{A_1}{B_1}$, $S_2 = \frac{A_2}{B_2}$, $S_3 = \frac{A_3}{B_3}$ are **weakly compatible**
if at least one of the following intersections is empty

$A_1 \cap A_2 \cap A_3$ , $A_1 \cap B_2 \cap B_3$ , $B_1 \cap A_2 \cap B_3$ , $B_1 \cap B_2 \cap A_3$

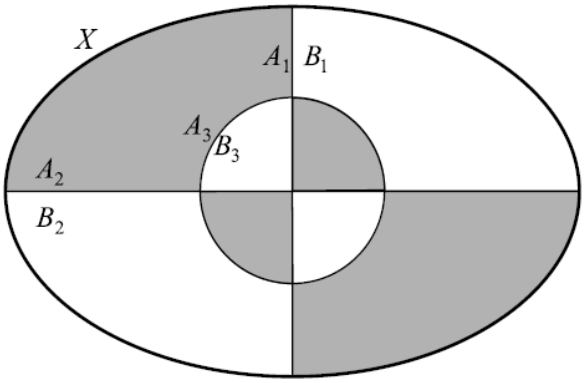
and at least one of the following intersections is empty

$B_1 \cap B_2 \cap B_3$ , $B_1 \cap A_2 \cap A_3$ , $A_1 \cap B_2 \cap A_3$ , $A_1 \cap A_2 \cap B_3$

A set of splits is weakly compatible if all its splits are (triplewise) weakly compatible.




(a) Venn diagram for compatibility




(b) Venn diagram for weak compatiblity

(a) Two splits $S_1 = \frac{A_1}{B_1}$ and $S_2 = \frac{A_2}{B_2}$ on $\mathcal{X}$ are compatible, if and only if one of the four regions in this Venn diagram is empty. (b) Three splits $S_1 = \frac{A_1}{B_1}$, $S_2 = \frac{A_2}{B_2}$ and $S_3 = \frac{A_3}{B_3}$ on $\mathcal{X}$ are weakly compatible if and only if at least one of the gray regions and one of the white regions in this Venn diagram are empty.

If three splits are not weakly compatible, then it can happen that every possible intersection of any three split parts is non-empty and the split network required to represent the three splits consists of the eight edges of a cube.

Thus the split networks associated with weakly compatible splits are often quite close to being planar, as they usually have only a few edges crossing over each other and do not contain any “high-dimensional cubes”, which may occur for completely unrestricted sets of splits.

The set of splits obtained by two unrooted phylogenetic trees is weakly compatible.

**Lemma 5.8.3** (Circular implies weakly compatible) *Let $\mathcal{S}$ be a set of splits on $\mathcal{X}$. If $\mathcal{S}$ is circular, then $\mathcal{S}$ is weakly compatible.*

## Buneman tree

One way to determine whether a given set of splits $\mathcal{S}$ on $\mathcal{X}$ is compatible is to check for each quadruple of taxa $Q \subseteq \mathcal{X}$ and every pair of splits $S_1$ and $S_2$ whether the splits induced by $S_1$ and $S_2$ on $Q$ are compatible:

**Lemma 5.9.1** (Quadruple condition for compatibility) *Two distinct splits $S_1$ and $S_2$ on $\mathcal{X}$ are compatible if and only if the set $\{S_1|_Q, S_2|_Q\}$ contains at most one non-trivial split for every quadruple $Q$ on $\mathcal{X}$.*

In other words, two splits $S_1$ and $S_2$ on $\mathcal{X}$ are compatible, unless there exists a quadruple $Q$ on $\mathcal{X}$ that induces one non-trivial split $S_1|_Q$ on $S_1$ and a different non-trivial split $S_2|_Q$ on $S_2$. A non-trivial split $\frac{\{w,x\}}{\{y,z\}}$ on a quadruple $Q = \{w, x, y, z\}$ is called a *quartet topology*. The lemma suggests the following strategy for constructing a set of compatible splits on $\mathcal{X}$:

- For each quadruple $Q$ on $\mathcal{X}$ choose one of the three possible quartet topologies and denote it by $\hat{Q}$.
- Construct the set of all splits $\mathcal{S}$ that *respect* the chosen quartet topologies, that is, determine every split $S = \frac{A}{B}$ on $\mathcal{X}$ for which $S|_Q = \hat{Q}$ holds for all quadruples $Q$ on $\mathcal{X}$ with $|Q \cap A| = |Q \cap B| = 2$.

**Exercise 5.9.2** (Compatibility) *Show that the set of splits $\mathcal{S}$ computed in this way is compatible.*

Assume that we are given a distance matrix $D$ on $\mathcal{X}$. How to select a quartet topology for each quadruple $Q \subseteq \mathcal{X}$? For a given quadruple $Q = \{w, x, y, z\} \subseteq \mathcal{X}$, consider the three possible sums of pairs of distances on $Q$:

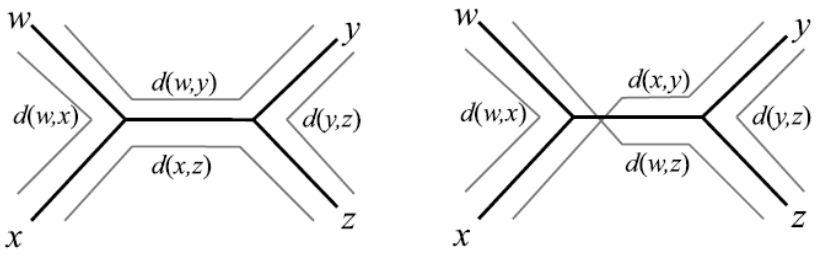
$$d(w, x) + d(y, z), \quad d(w, y) + d(x, z) \quad \text{and} \quad d(w, z) + d(x, y). \tag{5.20}$$

We choose the quartet topology corresponding to the smallest of these three values. For example, if $d(w, x) + d(y, z)$ is smallest, then we set $\hat{Q} = \frac{\{w,x\}}{\{y,z\}}$. If more than one quartet topology takes on the smallest value, then no topology is chosen for the quadruple. The motivation for this choice is the desire to minimize the distances between taxa on the same side of a split.

To describe this more formally, consider any four taxa $w$, $x$, $y$ and $z$ with $\{w, x\} \cap \{y, z\} = \emptyset$, but not necessarily $w \neq x$ or $y \neq z$. We define the (quartet) *Buneman index* of $\hat{Q} = \frac{\{w,x\}}{\{y,z\}}$ with respect to $D$ as:

$$\hat{\beta}_D\left(\tfrac{\{w,x\}}{\{y,z\}}\right) = \tfrac{1}{2}\big(\min\{d(w, y) + d(x, z), d(w, z) + d(x, y)\} - d(w, x) - d(y, z)\big). \quad (5.21)$$

It follows from this definition that the Buneman index is positive for, at most, one quartet topology on any given quadruple; and this is the one that we choose. To understand the quartet Buneman index, note that it measures the length of the central edge in an edge-weighted phylogenetic tree on four taxa, as illustrated in




A phylogenetic tree on four taxa $\{w, x, y, z\}$. (a) Here we see that $d(w, y) + d(x, z) - d(w, x) - d(y, z)$ covers the length of the central edge $e$ exactly twice, and thus that the Buneman index $\hat{\beta}_D\left(\frac{\{w,x\}}{\{y,z\}}\right)$ equals the length of $e$. (b) Similarly, in this case $d(w, z) + d(x, y) - d(w, x) - d(y, z)$ does so, too.

We define the *Buneman index* of a split $S = \frac{A}{B}$ on $\mathcal{X}$ as the minimum quartet Buneman index of all the quadruples induced by $S$:

$$\beta_D(S) = \min\left\{\hat{\beta}_D\left(\tfrac{\{w,x\}}{\{y,z\}}\right) \mid w, x \in A,\ y, z \in B\right\}. \quad (5.22)$$

The set of splits with positive Buneman index is compatible and the corresponding unrooted phylogenetic tree is called the **Buneman tree** for $D$.

**Lemma 5.9.3** (Buneman tree on additive distances) *Let $D$ be a distance matrix on $\mathcal{X}$. The distances in the Buneman tree $T$ equal the distances in $D$, if and only if $D$ is additive.*

Although this result shows that the Buneman tree method computes the correct tree for any distance matrix $D$ that satisfies the four-point condition, the method is rarely used in practice. This is because violations of the four-point condition usually lead to a loss of splits and the resulting tree is often very unresolved.

## Refined Buneman tree

from [BM99], [MS99]

An important problem in phylogenetic analysis is to approximate distances (such as those arising from biomolecular data) by tree metrics, and many various methods have been found for attacking this (see [3,4] for surveys). We investigate this problem by looking for a *tree construction map*, that is, a map $\phi : \mathcal{D}(X) \to \mathcal{D}(X)$, with $\phi(\mathcal{D}(X)) \subseteq \mathcal{T}(X)$, which satisfies the following properties.

(R1) $\phi|_{\mathcal{T}(X)} = Id|_{\mathcal{T}(X)}$.
(R2) The map $\phi$ is continuous.
(R3) The map $\phi$ is *homogeneous*, i.e., $\phi(\lambda d) = \lambda\phi(d)$, for $d \in \mathcal{D}(X)$, and $\lambda > 0$.

(R4) The map $\phi$ is *equivariant*, i.e., $\phi(d^\tau) = (\phi \circ d)^\tau$ for all $\tau$ in the permutation group of $X$ and $d \in \mathcal{D}(X)$, where $d^\tau(x, y) = d(\tau(x), \tau(y))$ for all $x, y \in X$.
(R5) If $d \in \mathcal{D}(X)$, then $\phi(d)$ can be computed in time that is polynomial in $|X|$.

Requirements (R1)–(R5) are chosen since they are desirable in biological applications. For example, (R4) can be rephrased as requiring that the tree construction method does not depend on the order in which the taxa set $X$ is processed—a property that does not hold for the popular Neighbor Joining method, for example. (See [5–7] for more details.)

In [8], Buneman gives a method for tree construction that satisfies (R1)–(R5). However, "the price paid for continuity", as Buneman puts it, is that the resulting tree is often highly unresolved. In [5], the Buneman construction is modified in an attempt to address this problem. The resulting construction is called the *refined Buneman tree* and is shown to satisfy (R1)–(R4). However it is not shown whether (R5) holds for this construction or not[1]. In this note, we fill in this gap and present an algorithm for computing the refined Buneman tree in polynomial time.

An important problem in applications (such as in biology) is how to take an arbitrary distance function, which is in some sense an estimate of (but not itself) a tree metric, and recover a"nearby" tree metric, and thereby the associated (edge weighted) $S$-tree. As Buneman [8] pointed out, it is desirable that such a map, from distance functions onto tree metrics, should be *continuous*. That is, a small change in the input distance function should not result in a drastically different edge-weighted tree. This is important for applications where distances are merely estimates obtained from imperfect data, often subject to stochastic effects (in biology, random mutations in DNA sequences). Surprisingly, one of the most popular methods currently in use in phylogenetic analysis – neighbor joining – fails on this count, as we show below in Section 4.2. Some earlier methods which attempt to find a closest tree metric to a given distance function are also discontinuous.

This prompted Buneman [8] to construct a continous map from metrics onto tree metrics, which we recall in Section 4. Buneman (and others subsequently, see [4]) have noticed that such a map applied to real data (particularly when $S$ is large) often leads to highly unresolved "star-like" trees, with few internal edges. Such trees tell a biologist little about the underlying evolutionary relationships. This has led to a preference by practitioners for other (discontinuous) methods as these methods generally construct fully resolved trees, which therefore appear to provide more information about the underlying evolutionary history. Yet, as pointed out in [8], such methods will construct fully resolved trees even if fed completely random data. In this case the evolutionary "information" contained in the tree is completely phantom, and liable to change completely under a small perturbation. Buneman suggests that the non-resolution observed in his tree building method is "the price paid for continuity".

One escape from this dilemma has been to modify Buneman's construction so as to output a graph, rather than necessarily a tree, via the elegant split decomposition theory of Bandelt and Dress [4]. Here we adopt a slightly different approach – by modifying Buneman's construction in an alternative way (see Section 5) we are able to ensure that the output is always a tree, but it will, in general, give a more highly resolved output tree than Buneman's method.

Regarding homogeneity, we note that in biological applications distance functions are frequently transformed by non-linear (typically logarithmic) functions before being used to reconstruct trees. Clearly such functions are not homogeneous, so the requirement of homogeneity is meant to apply simply to the transformed distances, not to the input distances. Homogeneity is desirable for applications to transformed distances as these distances generally estimate the expected number of mutations that have occurred between pairs of species for sequences that have been undergoing site mutations at some rate over a period of time (see [14]) – homogeneity thus becomes the requirement that the edge weights on the output trees should be proportional to the expected number of mutations on that edge (and so proportional to time, in case the rate is constant).

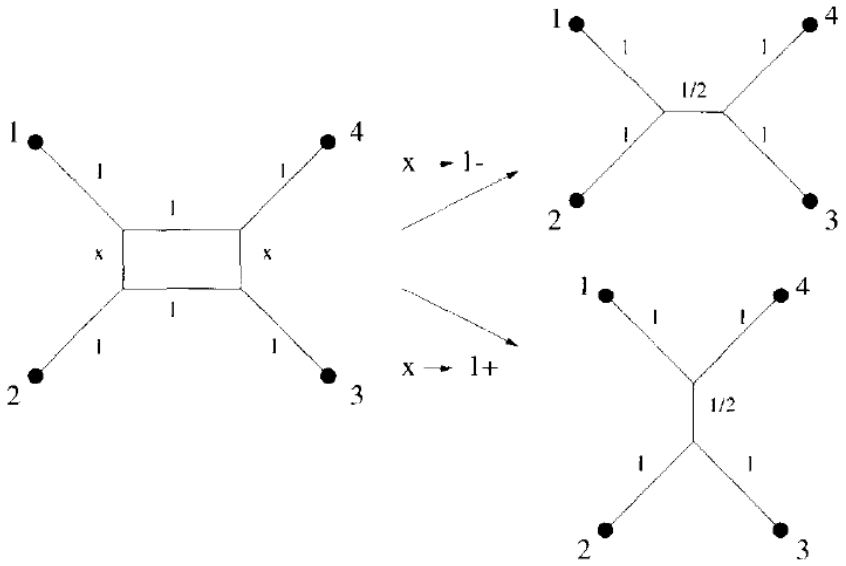



Fig. 1. An example where NJ is not continuous.

## Split decomposition

In the previous section, we saw that if a set of splits $\mathcal{S}$ on $\mathcal{X}$ has the property that it gives rise to at most one quartet topology for each quadruple $Q$ on $\mathcal{X}$, then this property suffices to ensure that the given set of splits $\mathcal{S}$ is compatible. In this section, we show that prescribing *up to two* of the three possible quartet topologies for every quadruple $Q$ on $\mathcal{X}$ ensures that a given set of splits is *weakly* compatible. We start with the following observation [9]:

**Lemma 5.9.4** (Quadruple condition for weak compatibility) *Three distinct splits $S_1$, $S_2$ and $S_3$ on $\mathcal{X}$ are weakly compatible if and only if the set $\{S_1|_Q, S_2|_Q, S_3|_Q\}$ contains at most two different non-trivial splits on $Q$ for every quadruple $Q$ on $\mathcal{X}$.*

This result suggests the following strategy for constructing a set of weakly compatible splits on $\mathcal{X}$:

- For each quadruple $Q$ on $\mathcal{X}$ we make a fixed choice of up to two of the three possible quartet topologies.
- Construct the set of all splits $\mathcal{S}$ that respect the chosen quartet topologies, that is, determine every split $S = \frac{A}{B}$ on $\mathcal{X}$ for which $S|_Q$ equals one of the chosen quartet topologies on $Q$, for all quadruples $Q$ on $\mathcal{X}$ with $|Q \cap A| = |Q \cap B| = 2$.

**Lemma 5.9.6** (Result is weakly compatibility) *The set of splits $\mathcal{S}$ computed by the described strategy is weakly compatible.*

Assume we are given a distance matrix $D$ on $\mathcal{X}$. How to select up to two quartet topologies for each quadruple $Q \subseteq \mathcal{X}$? For a given quadruple $Q = \{w, x, y, z\} \subseteq \mathcal{X}$, consider the three possible sums of distances on $Q$:

$$d(w, x) + d(y, z), \;\; d(w, y) + d(x, z) \text{ and } d(w, z) + d(x, y). \tag{5.23}$$

Generalizing the method discussed in the previous section, we choose all (that is, zero, one or two) quartet topologies for which the corresponding sum is smaller than the largest sum. More precisely, if the three sums are all different, or if two are equal and smaller than the third, then we choose the two topologies associated with the two smaller sums. If two are equal and larger than the third, then we choose the topology associated with the third value. If all three values are the same, then none of the three topologies is chosen. For example, if $d(w, x) + d(y, z) < d(w, y) + d(x, z) = d(w, z) + d(x, y)$, then we (only) choose the quartet topology $\frac{\{w,x\}}{\{y,z\}}$.

Consider any four taxa $w$, $x$, $y$ and $z$ with $\{w, x\} \cap \{y, z\} = \emptyset$, but not necessarily $w \neq x$ or $y \neq z$. We define the (quartet) *isolation index* of $\hat{Q} = \frac{\{w,x\}}{\{y,z\}}$ with respect to $D$ as:

$$\hat{\alpha}_D\left(\tfrac{\{w,x\}}{\{y,z\}}\right) = \tfrac{1}{2}\Big( \max\Big\{ d(w, x) + d(y, z),\, d(w, y) + d(x, z),\, d(w, z) + d(x, y) \Big\} - d(w, x) - d(y, z) \Big). \tag{5.24}$$

Note that the quartet isolation index is always non-negative, because the quantity that is subtracted also occurs as an argument of the maximum function. Also, because one of the three sums has to be largest, the quartet isolation index can only be positive for at most two quartet topologies, and these are the ones that we choose.

We define the *isolation index* $\alpha_D(S)$ *of a split* $S = \frac{A}{B}$ on $\mathcal{X}$ as the minimum isolation index of all the quartets induced by $S$:

$$\alpha_D(S) = \min\{\hat{\alpha}_D\left(\tfrac{\{w,x\}}{\{y,z\}}\right) \mid w, x \in A,\; y, z \in B\} \geq 0. \tag{5.25}$$

A split $S$ whose isolation index $\alpha_D(S)$ is greater than 0 is called a *D-split*. By construction, we have:

**Lemma 5.9.7** (D-splits are weakly compatible) *Let $D$ be a distance matrix on $\mathcal{X}$. The set of D-splits on $\mathcal{X}$ is weakly compatible.*

**Algorithm 5.9.10** (Split decomposition) *Given a distance matrix $D$ on $\mathcal{X} = \{x_1, \ldots, x_n\}$, compute the set of all weighted $D$-splits on $\mathcal{X}$ as follows:*

*Initially, set $\mathcal{X}_1 = \{x_1\}$ and $\mathcal{S}_1 = \emptyset$. Now, assume that we have computed the set of all $D$-splits $\mathcal{S}_i$ on the first $i$ taxa $\mathcal{X}_i = \{x_1, \ldots, x_i\}$. To obtain $\mathcal{S}_{i+1}$ on $\mathcal{X}_{i+1} = \{x_1, \ldots, x_{i+1}\}$, for each split $\frac{A}{B} \in \mathcal{S}_i$ do:*

- *If $\alpha_D\left(\frac{A \cup \{x_{i+1}\}}{B}\right) > 0$, then add $\frac{A \cup \{x_{i+1}\}}{B}$ to $\mathcal{S}_{i+1}$.*
- *If $\alpha_D\left(\frac{A}{B \cup \{x_{i+1}\}}\right) > 0$, then add $\frac{A}{B \cup \{x_{i+1}\}}$ to $\mathcal{S}_{i+1}$.*
- *If $\alpha_D\left(\frac{\mathcal{X}_i}{\{x_{i+1}\}}\right) > 0$, then add $\frac{\mathcal{X}_i}{\{x_{i+1}\}}$ to $\mathcal{S}_{i+1}$.*

*The result is given by $\mathcal{S}_n$.*

This algorithm works because extending a partial split to more taxa can only maintain or decrease its isolation index. The worst-case time requirement of this algorithm is $O(n^6)$, because the number of iterations of the main loop is $n$, the number of splits present in the $i$-th iteration is $O(n^2)$ (as we show below), and the time required to compute the isolation index of a split in the main loop is $O(n^3)$. In practice, the run-time of the algorithm is usually much better than this analysis suggests, since the number of non-trivial $D$-splits tends to drop quite rapidly as the number of taxa increases.

We can also use this algorithm to efficiently compute all splits of the Buneman tree, by simply replacing the isolation index $\alpha_D$ by the Buneman index $\beta_D$.

Let $D$ be a distance matrix on $\mathcal{X}$ and let $\mathcal{S}$ be the set of all $D$-splits on $\mathcal{X}$. We define the *residue* distance matrix $D_0$ as

$$d_0(x, y) = d(x, y) - \sum_{S \in \mathcal{S}(x,y)} \alpha_D(S) \geq 0, \tag{5.40}$$

where $\mathcal{S}(x, y)$ denotes the set of all splits that separate $x$ and $y$, as usual.

**Theorem 5.9.12** (Residue is split prime) *If $D_0$ is the residue of some distance matrix $D$, then $D_0$ does not admit any $D_0$-split. In this case $D_0$ is called* split prime.

**Lemma 5.9.15** (Split decomposition) *Let $D$ be a distance matrix on $\mathcal{X}$ and let $\mathcal{S}$ be the set of all $D$-splits on $\mathcal{X}$. The* split decomposition *of $D$ is given by the unique decomposition*

$$d(x, y) = \sum_{S \in \mathcal{S}(x,y)} \alpha_D(S) + d_0(x, y) \tag{5.53}$$

*for all $x, y \in \mathcal{X}$, where $D_0$ is the split prime residue of $D$.*

In applications, a distance matrix $D$ on $\mathcal{X}$ is represented by the split network $N$ of all its $D$-splits, computed as described in Chapter 7. If the residue $D_0$ of $D$ is zero for all $x, y \in \mathcal{X}$, then $D$ is *totally decomposable*. In this case the split network $N$ provides an exact representation of $D$, as the length of a shortest path between any two taxa $x$ and $y$ in $N$ equals $d(x, y)$.

To measure how well the distances in $N$ (or, more precisely, in the split decomposition of $D$) represent the distances in $D$, we define the *percent fit* as

$$\text{fit}(N, D) = 100 \times \frac{\sum_{x,y \in \mathcal{X}} \sum_{S \in \mathcal{S}(x,y)} \alpha_D(S)}{\sum_{x,y \in \mathcal{X}} d(x, y)} \geq 0, \tag{5.54}$$

making use of the fact that distances based on the $D$-splits never exceed the distances given in $D$. In practice, a fit of 80% or more is usually required for a split network $N$ based on split decomposition to be considered a reliable representation of a distance matrix $D$.

A main attraction of the split decomposition method is that the number of splits that it can produce is larger than the number of splits produced by a tree-building method (such as the Buneman tree), while not being too large:

**Lemma 5.9.16** (Size of the split decomposition) *Let $D$ be a distance matrix on $\mathcal{X} = \{x_1, \ldots, x_n\}$. The number of $D$-splits on $\mathcal{X}$ is at most $\frac{n(n-1)}{2}$.*

**Lemma 5.9.17** (Split decomposition on additive distances) *Let $D$ be a distance matrix on $\mathcal{X}$. Then $D$ is additive if and only if the set of $D$-splits on $\mathcal{X}$ is compatible and the residue $D_0$ of $D$ is zero.*

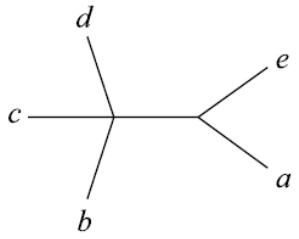
The split decomposition method has been used in numerous publications to represent distance matrices that show a significant amount of non-tree-likeness, or in situations in which there is reason to believe that the best representation of the data is not a phylogenetic tree. In practice, the method is limited to relatively small datasets of less than 100 taxa, or even much smaller, depending on the actual data. This is because the isolation index of a split $S$ is defined as the minimum quartet isolation index over all quartets that span $S$ and so, for a larger number of taxa, or for more divergent taxa, it is quite likely that one of the quartets will have an isolation index of 0, thus preventing $S$ from being a $D$-split.

In Section 10.4 we discuss the neighbor-net method, an alternative method for constructing a set of (not necessarily compatible) splits from a distance matrix, which does not suffer from this practical limitation.

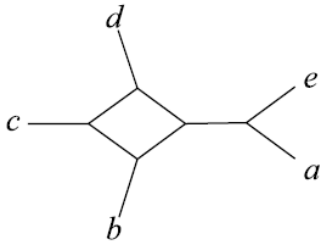
Because the isolation index is a relaxation of the Buneman index, it follows that the set of splits computed by the Buneman tree method is always a subset of the set of splits computed by the split decomposition method. In consequence, the Buneman tree can always be obtained from a split network that represents the split decomposition by *contracting* certain splits, as defined in the next section.

| | $a$ | $b$ | $c$ | $d$ | $e$ |
|---|---|---|---|---|---|
| $a$ | 0 | 4 | 5 | 4 | 2 |
| $b$ | 4 | 0 | 3 | 4 | 4 |
| $c$ | 5 | 3 | 0 | 3 | 5 |
| $d$ | 4 | 4 | 3 | 0 | 4 |
| $e$ | 2 | 4 | 5 | 4 | 0 |

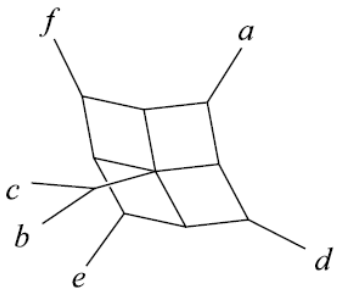
(a) Distance matrix



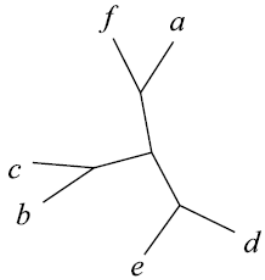

(b) Buneman tree



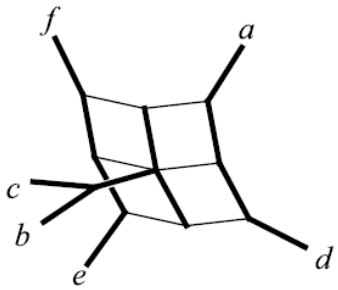

(c) Split decomposition




(a) Split network $N$




(b) Tree $T$




(c) $T$ embedded in $N$

(a) A split network $N$ on $\mathcal{X} = \{a, b, c, d, e, f\}$ and (b) a tree $T$ on $\mathcal{X}$. In (c) the edges of $N$ that represent the splits from $T$ are highlighted. One sees clearly that contraction of the other edges gives rise to the tree $T$ and thus that $N$ contains $T$.

## Neighbor-net

The neighbor-net method takes as input a distance matrix $D$ on $\mathcal{X}$ and produces as output a collection of weighted splits $\mathcal{S}$ on $\mathcal{X}$ [32]. As we shall see, the produced set of splits is circular, that is, there exists a circular ordering $(x_1, x_2, \ldots, x_n)$ of the taxa $\mathcal{X}$ such that every split $S \in \mathcal{S}$ is of the form

$$S = \frac{\{x_p, x_{p+1}, \ldots, x_q\}}{\mathcal{X} - \{x_p, \ldots, x_q\}} \tag{10.5}$$

for some pair of indices $p$ and $q$ with $1 < p \leq q \leq n$. The resulting set of splits $\mathcal{S}$ is then given to the circular network algorithm described in Section 7.2 to compute an actual split network $N$.

As we saw in Section 5.7, circularity implies that $\mathcal{S}$ can be represented by an outer-labeled planar split network, that is, a split network drawn in the plane in such a way that no two edges cross and all taxon labels occur around the outside of the network. The resulting networks are not overly complicated and that is one reason why neighbor-net is a popular method for constructing phylogenetic networks.

A distance matrix $D$ on $\mathcal{X}$ is called *circular*, if it equals the weighted split metric of some circular set of splits $\mathcal{S}$ on $\mathcal{X}$. The relationship between the input distance matrix $D$ and the splits produced by neighbor-net is given by the following result:

**Theorem 10.4.1** (Consistency of neighbor-net) *Let $D$ be a distance matrix on $\mathcal{X}$. The set of weighted splits $\mathcal{S}$ computed by neighbor-net represent $D$ (exactly) if and only if $D$ is circular.*

Because compatible splits are always circular, it follows that the neighbor-net method always computes the correct phylogenetic tree, given a distance matrix that is additive.

On real biological data, the obtained distance matrices usually do not fulfill the four-point condition and are also not circular. Nevertheless, just as the neighbor-joining method is used to compute trees from distance matrices even when the four-point condition is not satisfied, it is also common practice to apply the neighbor-net algorithm even when circularity is not given. The hope is that deviations from the required conditions do not distort the result too much.

Given a distance matrix $D$ on $\mathcal{X}$, the *neighbor-net method* proceeds in two steps:

(i) The main computation is to determine a circular ordering $Z$ of $\mathcal{X}$, using an iterative algorithm similar to neighbor-joining.

(ii) Then a set of weighted splits $\mathcal{S}$ is computed that respect the circular ordering $Z$.

Neighbor-net is an attractive method for computing split networks for the following reasons: First, the resulting networks are outer-labeled planar and thus easy to draw and to read. Secondly, the algorithm is quite fast. Thirdly, it produces resolved networks even for quite large numbers of taxa, unlike the split decomposition method, which rapidly loses resolution as the number of taxa increases.

see also [BM2004, BH2023]

## T-theory

From a mathematically more abstract point-of-view, we can interpret an unrooted tree as a compact, simply connected, one-dimensional polytope, and this leads to the following question posed in T-theory: If $D$ is *not* additive, can we characterize an appropriate low-dimensional compact polytope into which we can embed $(\mathcal{X}, D)$ *isometrically*, that is, preserving all distances between elements of $\mathcal{X}$? This polytope can then serve as a mathematical generalization of a phylogenetic tree.

Another motivation for the development of T-theory is the desire to give a direct characterization of the phylogenetic tree $T$ associated with an additive distance matrix $D$ that is not based, explicitly or implicitly, on a tree construction algorithm.

The key concept in T-theory is a construction called the *tight span* that provides a (at most $\lfloor \frac{n}{2} \rfloor$-dimensional) compact polytope that contains a given finite metric space. The main aim of T-theory is to investigate and understand the combinatorial structure of the tight span.

Let $D$ be a distance function on $\mathcal{X} = \{x_1, \ldots, x_n\}$. We first define an unbounded $n$-dimensional polytope associated with $(\mathcal{X}, D)$:

$$P(\mathcal{X}, D) = \left\{ \mathbf{v} = \begin{pmatrix} v_1 \\ \ldots \\ v_n \end{pmatrix} \in \mathbb{R}^n \mid v_i + v_j \geq d(x_i, x_j),\ i, j = 1, \ldots, n \right\}, \tag{5.57}$$

in particular with $v_i \geq 0$ for all $i$, which follows from $v_i + v_i \geq d(x_i, x_i) = 0$.

For two $n$-dimensional vectors $\mathbf{u}$ and $\mathbf{v}$ we define $\mathbf{u} \leq \mathbf{v}$ to mean $u_i \leq v_i$ for all $i = 1, \ldots, n$. We now come to the main definition:

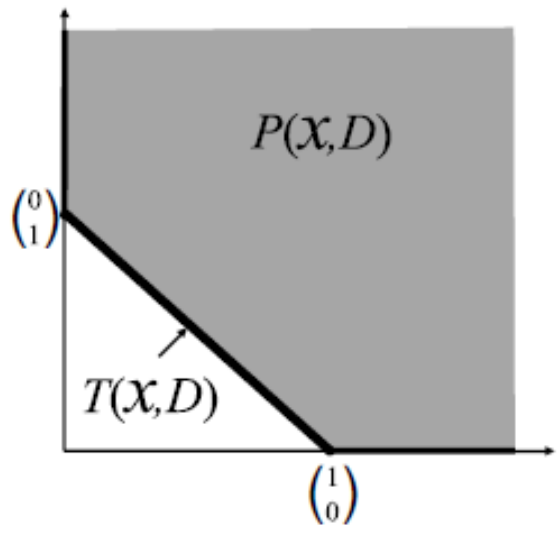
**Definition 5.12.1** (Tight span) *For a finite metric space $(\mathcal{X}, D)$ we define the* tight span $T(\mathcal{X}, D)$ *as the (compact) polytope consisting of all minimal points in $P(\mathcal{X}, D)$, given by:*

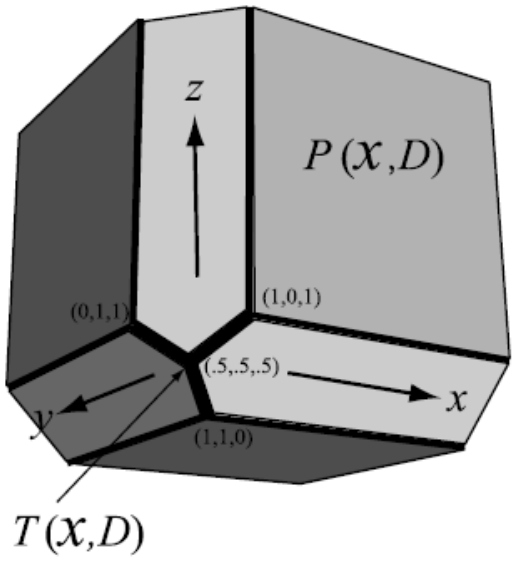
$$T(\mathcal{X}, D) = \left\{ \mathbf{v} \in P(\mathcal{X}, D) | \mathbf{w} \in P(\mathcal{X}, D) \text{ and } \mathbf{w} \leq \mathbf{v} \text{ implies } \mathbf{w} = \mathbf{v} \right\}. \tag{5.58}$$

We shall consider the tight span $T(\mathcal{X}, D)$ together with the distance function

$$||\mathbf{v}, \mathbf{w}||_\infty = \max \{|v_i - w_i| : i = 1, \ldots, n\} \tag{5.59}$$

as a metric space.



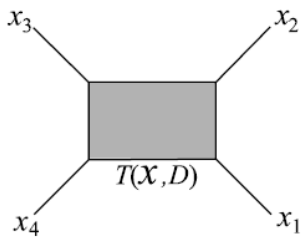
**Lemma 5.12.3** (Isometric embedding) *Let $(\mathcal{X}, D)$ be a finite metric space on $\mathcal{X} = \{x_1, \ldots, x_n\}$. The* Kuratowski map $\kappa : \mathcal{X} \to \mathbb{R}^n$, *defined for all taxa $x_t$ in $\mathcal{X}$ as*

$$x_t \mapsto \kappa(x_t) = \begin{pmatrix} d(x_1, x_t) \\ \ldots \\ d(x_t, x_t) \\ \ldots \\ d(x_n, x_t) \end{pmatrix}, \tag{5.62}$$

*maps the metric space $(\mathcal{X}, D)$ isometrically into the tight span $T(\mathcal{X}, D)$.*

Let $P$ be a compact $n$-dimensional polytope. We call $P$ an $\mathbb{R}$-*tree*, if it is the set of all points contained in a graph-theoretical tree embedded in $n$-dimensional space.

**Lemma 5.12.4** (Additivity and tight span) *Let $(\mathcal{X}, D)$ be a finite metric space. The distance matrix $D$ is additive if and only if the tight span $T(\mathcal{X}, D)$ is an $\mathbb{R}$-tree.*

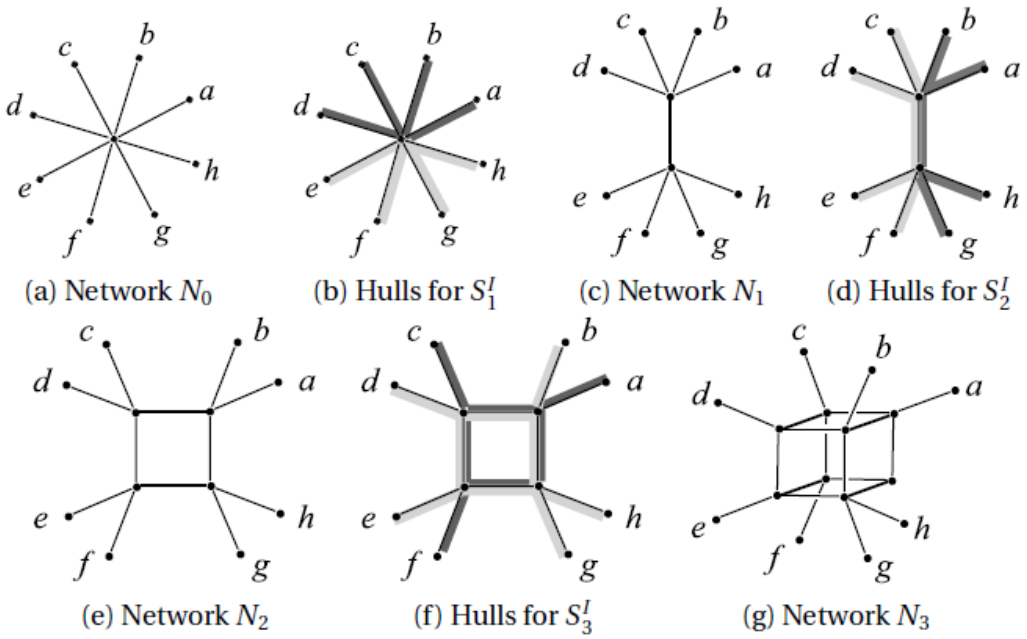



The tight span $T(\mathcal{X}, D)$ of a distance matrix $D$ on $\mathcal{X} = \{x_1, \ldots, x_4\}$ that is not additive. The polytope consists of all points contained in the lines and in the gray area. The labels $x_1, \ldots, x_4$ indicate the locations of taxa under the Kuratowski embedding.

**Lemma 5.12.5** (Split decomposition and tight span) *Let $D$ be a distance matrix on $\mathcal{X}$. If $D$ is totally decomposable then the canonical split network $N$ representing the split decomposition of $D$ is contained in the* 1-skeleton *(set of 1-dimensional faces) of $T(\mathcal{X}, D)$.*

# Phylogenetic networks from splits

## Convex hull algorithm




Construction of the canonical split network for three non-trivial splits $S_1^I$, $S_2^I$ and $S_3^I$ on $\mathcal{X} = \{a, \ldots, h\}$. (a) The split network $N_0$ representing all trivial splits on $\mathcal{X}$. (b) The two convex hulls $H(A_1)$ and $H(B_1)$ for the split $S_1^I = \frac{A_1}{B_1} = \frac{\{a,b,c,d\}}{\{e,f,g,h\}}$ (highlighted in dark gray and light gray, respectively). The intersection $H(A_1) \cap H(B_1)$ consists of the central node. (c) The resulting network $N_1$ obtained using the convex hull algorithm. (d) The two convex hulls for $S_2^I = \frac{A_2}{B_2} = \frac{\{a,b,g,h\}}{\{c,d,e,f\}}$. (e) The resulting network $N_2$. (f) The two convex hulls for $S_3^I = \frac{A_3}{B_3} = \frac{\{a,c,f\}}{\{b,d,e,g,h\}}$. (g) The resulting network $N_3$.

The convex hull algorithm computes the Buneman graph.

from [DHM96]

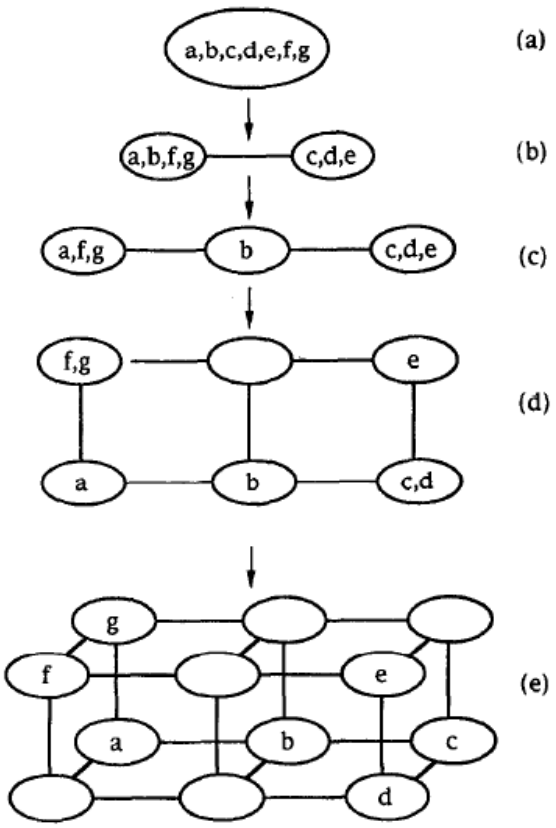



Fig. 3. Producing a subgraph of the four-dimensional hypercube.

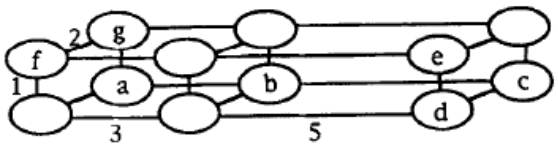



Fig. 4. The weighted version of the graph in Fig. 3(e).

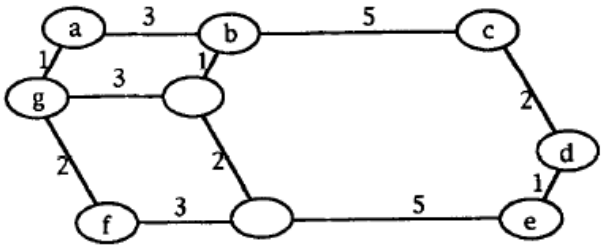



Fig. 5. The weighted graph with redundant edges removed.

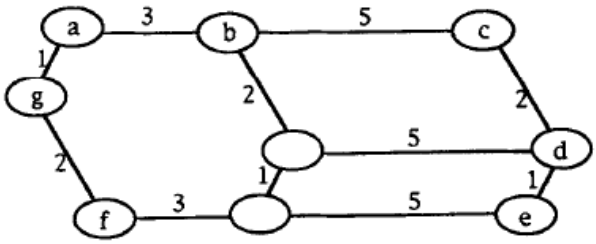


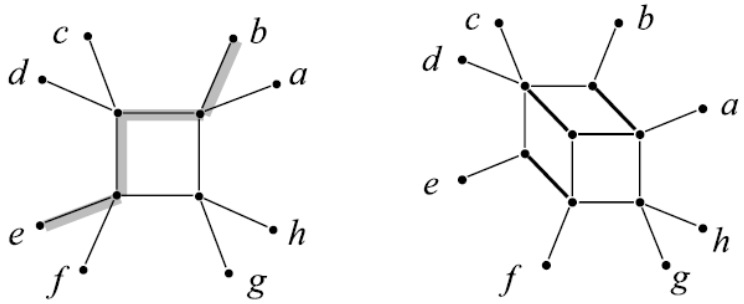

Fig. 6. A different representation of the same four splits.

We now indicate how to produce the splits-graph from a given family $\mathscr{S}$ of, say, $N$ splits, e.g. the $d$-splits for a metric $d$. The first step is to produce a graph from the splits in $\mathscr{S}$ that is a subgraph of an $N$-dimensional hypercube.

Once we have obtained this graph for a system of weighted splits, for each split we expand or contract all the edges in the band of parallel edges that represents the split by the same amount so that their lengths become proportional to the given weight (e.g. the isolation index, if we are dealing with $d$-splits) to obtain a weighted graph.

Consider the weighted graph obtained in Fig. 4. Clearly, some of its edges are redundant in representing the data set, since their removal does not affect the distance between the labeled vertices in the graph, and it also preserves the splits defined by the collections of parallel edges. The graph depicted in Fig. 5 is obtained by the removal of such redundant edges. It contains the same information as the original weighted graph, whilst having the advantage of being planar. In general, by carefully removing all such edges in the original weighted graph and by changing the slopes of the families of parallel edges representing the various splits, it is often possible to get an almost planar splits-graph (see [26]). It should be noted, however, that even though the planar representation obtained in this way contains all of the original data, it is not unique.

## Circular network algorithm

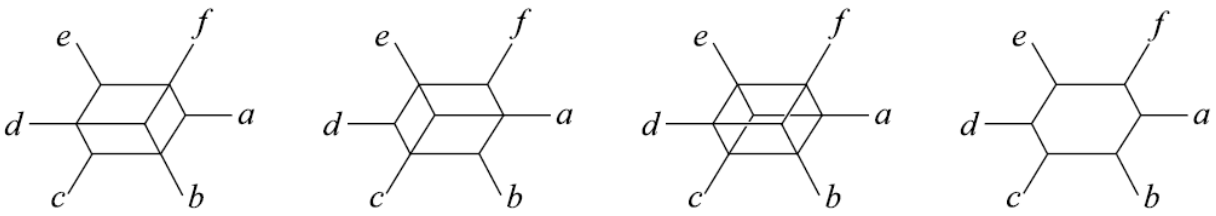



(a) Network $N_2$ (b) Network $N_3$

To add the split $\frac{A}{B} = \frac{\{a,f,g,h\}}{\{b,c,d,e\}}$ to the split network $N_2$ shown in (a), with $x_1 = a$, $x_2 = b$, etc., the circular network algorithm duplicates the internal edges of the path $M(b, e)$ leading from the $b$ to $e$ (highlighted in gray) to obtain the network $N_3$ shown in (b), in which the three edges representing $\frac{A}{B}$ are highlighted in bold.

The number of nodes and edges produced by this algorithm is at most quadratic in the number of splits $m$.

The requirement that the non-trivial splits must be processed in order of non-increasing size of the split parts not containing $x_1$ is not only important for minimizing the number of edges in the network. If we disregard this requirement, then the resulting split network may fail to have the property that edges corresponding to the same split can be drawn as parallel line segments of the same length:




All four different split networks shown here represent the same set of splits. The split networks shown in (a) and (b) were computed using the circular network algorithm processing the splits and taxa in two different orders. The one shown in (c) was constructed using the convex hull algorithm. The split network shown in (d) can be obtained by deleting superfluous edges in any of the first three.

Unlike in the case of the convex hull algorithm, the split network constructed by this algorithm is not uniquely defined, as the resulting network depends on the order in which the splits are processed.

This example also illustrates that the convex hull algorithm does not necessarily produce a planar network, even when given a circular set of splits.

In many cases, direct application of the convex hull algorithm leads to an over-complicated network, as indicated in Figure 7.3. In practice, a useful heuristic is to first choose an order of the taxa such that a large subset of the given set of splits is circular. This subset of splits is then processed using the circular network algorithm to obtain an outer-labeled planar network. The remaining splits are then processed using the convex hull algorithm, which will add some non-planar parts to the network. This is the default approach used in SplitsTree [125].

In the following chapters we describe a number of different methods for computing a set of splits from sequences, distances, trees or quartets. All these methods are used in conjunction with the convex hull and circular network algorithms to obtain a split network from biological data.

## *Metric Spaces in Pure and Applied Mathematics*

**John Isbell's contribution**

We consider the category of metric spaces with non-expansive maps.

**Def.** Let $(A, d_A)$ and $(B, d_B)$ be metric spaces.

A **non-expansive map** from $A$ to $B$ is a function $f : A \to B$ such that $d_B\big(f(a), f(a')\big) \leq d_A(a, a'), \quad \forall\, a, a' \in A$

> Isbell then went on to show that a unique *injective hull* exists in this category for every one of its objects, providing an explicit construction of this hull for all spaces and noting that it comes endowed, at least for finite spaces, with an intrinsic polytopal structure.

**Facts**

- There exist **injective metric spaces**, that is
  a metric space $(X, d)$ such that,
  for every isometric embedding $\alpha : X \hookrightarrow X'$
  of $(X, d)$ into another metric space $(X', d')$ ,
  there exists a **non-expansive retract** $\alpha' : X' \to X$ , that is
  a non-expansive map from $X'$ to $X$ such that $\alpha' \circ \alpha = id_X$ .
- Every metric space $(X, d)$ can be embedded isometrically
  into an injective metric space $(\hat{X}, \hat{d})$ .
- Given an isometric embedding $\alpha : X \hookrightarrow \hat{X}$ of a metric space $(X, d)$
  into an injective metric space $(\hat{X}, \hat{d})$ ,
  there exists a unique smallest injective subspace $(\bar{X}, \bar{d})$ of $(\hat{X}, \hat{d})$
  containing $\alpha(X)$ .
  This subspace depends, up to isometry, only on $(X, d)$.

$$X \subseteq \bar{X} \subseteq \hat{X}$$

- The map

$$\begin{aligned} \bar{X} &\to \mathbb{R}^X \\ \bar{x} \mapsto h_{\bar{x}} : X &\to \mathbb{R} \\ x &\mapsto \bar{d}(\alpha(x), \bar{x}) \end{aligned}$$

  is an isometric embedding of $(\bar{X}, \bar{d})$ into $(\mathbb{R}^X, \|\cdot,\cdot\|_\infty)$ where

$$\|f, g\|_\infty \coloneqq \sup_{x \in X} |f(x) - g(x)|$$

- Its image coincides with

$$T(X,d) \coloneqq \left\{ f \in \mathbb{R}^X \,\middle|\, f(x) = \sup_{y \in X} \big(d(x,y) - f(y)\big),\ \forall\, x \in X \right\}$$

  that is called the **tight span** of $(X, d)$ .

- This embedding identifies $X$ with the set

$$\left\{ \begin{matrix} h_x : X \to \mathbb{R} \\ y \mapsto d(y,x) \end{matrix} \right\}_{x \in X}$$

  and hence with the subset of $T(X,d)$

$$T^0(X,d) \coloneqq \{\, f \in T(X,d) \mid f(X) \ni 0 \,\}$$

- This construction also identifies $T(X,d)$ with a subset of the convex set

$$P(X,d) \coloneqq \{\, f \in \mathbb{R}^X \mid f(x) + f(y) \geq d(x,y),\ \forall\, x, y \in X \,\}$$

  more precisely, it identifies it with the set of minimal maps in $P(X,d)$ , relative to the partial order inherithed from $\mathbb{R}^X$

$$f \leq g \quad \Leftrightarrow \quad f(x) \leq g(x),\ \ \forall x \in X$$

- $T(X,d)$ is contractible.
  There exists a continous family of non-expansive maps

$$f_t : T(X,d) \to T(X,d), \qquad t \in [0,1]$$

  such that $f_0$ is the identity and the image of $f_1$ is a point.

Although these notions may appear to be somewhat strange at first, the tight span of small metric spaces $(X, d)$ can be described in simple geometric terms as follows: In case $X$ consists of just two points of distance $c$, its tight span is exactly the interval of length $c$, its end points being just the two points from $X$ (thus the name "tight span"). In case $X$ consists of just three points of distance $c_1, c_2, c_3$, its tight span is the union of three intervals of length $(c_1+c_2-c_3)/2$, $(c_1+c_3-c_2)/2$, and $(c_2+c_3-c_1)/2$, respectively, all identified at one end point while the other three end points are the three points from $X$.

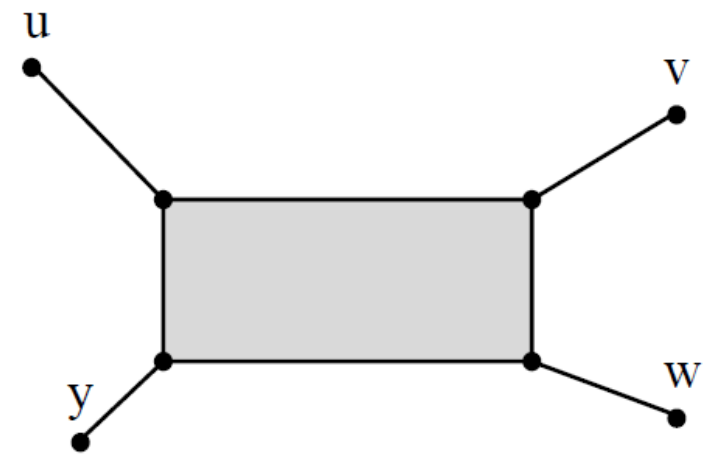



Figure 2: *The tight span of a generic metric $d$ on the set $\{u, v, w, y\}$ for which $d(u, w) + d(v, y)$ is the largest of the three sums $d(u, w) + d(v, y)$, $d(u, v) + d(w, y)$, and $d(u, y) + d(v, w)$; it consists of eight 0-cells, eight 1-cells, and one 2-cell.*

In general, the tight span of a *finite* metric space $(X, d)$ coincides exactly with the union of all compact faces of the polytope $P(X, d)$.

The construction works not only for metrics,
but also for maps $D$ from $\mathcal{P}_{fin}(X)$ to $\underline{\mathbb{R}} := \mathbb{R} \cup \{-\infty\}$

$$P(X,D) := \left\{ f \in \mathbb{R}^X \ : \sum_{x \in Y} f(x) \geq D(Y), \ \ \forall\, Y \in \mathcal{P}_{fin}(X) \right\}$$

$$T(X,D) := \left\{ f \in \mathbb{R}^X \ : \ f(x) = \sup_{Y \in \mathcal{P}_{fin}(X-\{x\})} \left( D(Y \cup \{x\}) \right) - \sum_{y \in Y} f(y) \right\}$$

In the case of a metric $d$, we can define $D_d : \mathcal{P}_{fin}(X) \to \underline{\mathbb{R}}$ as

$$D_d(Y) := \begin{cases} d(x,y)\,, & Y = \{x,y\} \\ -\infty\,, & \text{otherwise} \end{cases}$$

and we observe that the new definitions are consistent

$$P(X, D_d) = P(X, d)\,, \qquad T(X, D_d) = T(X, d)$$

Perhaps a bit surprisingly, this generalization can be used to construct affine buildings of $GL$-type.

## Phylogenetic analysis

Clearly, the obvious idea any tree-reconstruction algorithm must use is that, given any three sequences that have been derived by the process of replication, mutation, and selection from one common ancestral sequence, the last common ancestral sequence of the two more similar among those three sequences should have existed later than the last common ancestral sequence of all three sequences. This suggests the following tree-construction algorithm: First, identify each sequence $S$ from the set $X$ of sequences in question with the corresponding one-element clade $\{S\}$ consisting of $S$, only. Then, using any appropriately defined dissimilarity measure $d : X \times X \to \mathbb{R}$ (e.g. the mismatch or *Hamming* distance employed by Fitch and Margoliash), search for those two sequences $S_1, S_2$ that have minimal dissimilarity and, supposing that no other sequence in $X$ can be an offspring of the last common ancestral sequence of $S_1$ and $S_2$, fuse $S_1$ and $S_2$ into one larger *d-clade* $\{S_1\} \cup \{S_2\}$. Then replace the set $X$ by a smaller set $X'$ representing all maximal, presently identified ($d$-)clades (that is, the one $d$-clade of cardinality 2 and the additional, not yet processed single-element clades at that stage) and define a new dissimilarity measure on those clades by defining the distance $d(Y_1, Y_2)$ of any two such clades $Y_1, Y_2$ to be some function of the dissimilarities $d(y_1, y_2)$ with $y_1 \in Y_1$ and $y_2 \in Y_2$. And then, repeat the above process to identify the next two clades that are to be fused into one new, larger $d$-clade, and so on. Obviously, if $d(Y_1, Y_2)$ is defined by $d(Y_1, Y_2) := \min\{d(y_1, y_2) | y_1 \in Y_1, y_2 \in Y_2\}$ for any two $d$-clades $Y_1, Y_2$, this will lead exactly to the tree $T_{F\&M}(X, d)$ described above.

However, this procedure is obviously bound to make mistakes: Assume, we have four sequences $S_1, S_2, S_3, S_4$ and that, during the evolution of those four sequences from their common ancestor sequence $S$, there were first two distinct offsprings sequences $S', S''$ of $S$ so that $S_1$ and $S_2$ were later derived from $S'$ and $S_3$ and $S_4$ from $S''$. Assume furthermore that $S_1$ remained very similar to $S'$ and $S_3$ remained very similar to $S''$ and $S_2$ as well as $S_4$ diverged very far from their respective ancestor sequences. Then, the above algorithm will inevitably form a wrong clade $\{S_1, S_3\}$ (see Figure 4).

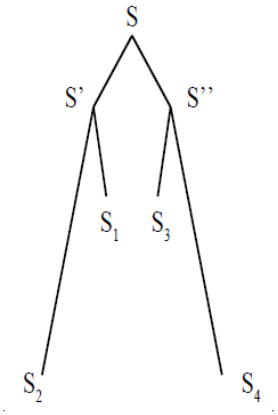



Figure 4: *As explained in the text, the incorrect clade $\{S_1, S_3\}$ is formed by the agglomeration algorithm and the 'true topology' of the tree is not found.*

Many algorithms have therefore been designed to deal with this particular problem. And quite a few of them accept the dissimilarities computed from the input sequences as a starting point, yet they search for a tree that provides the best global approximation of the given dissimilarity pattern, i.e. a tree

whose leaves are labeled by the elements from $X$, and to whose branches appropriate edge lengths are attached so that the resulting induced *tree metric* (that associates to any pair of elements $x, y$ from $X$ the total length of the unique path from the two leaves labeled with $x$ and $y$) matches the given dissimilarities *in toto* as closely as possible.

To imagine the task one has to perform using the approach it is worthwhile to observe that the *space* of all possible dissimilarities that can be defined on an $n$-set $X$ has dimension $\binom{n}{2}$ while the subspace of *tree-like* dissimilarities that can be defined on $X$ has dimension $2n-3$ (the maximal number of branches in a tree with $n$ leaves) and forms a rather complex low-dimensional network of large codimension $\binom{n}{2} - 2n + 3$ within this cone. Consequently, while trying to identify the best global 'tree-like' approximation of the given dissimilarity pattern, there may be many rather distinct, yet essentially equally good tree-like approximations to a given arbitrary dissimilarity $d$ and to find the best one will naturally be very hard

## Tree reconstruction and the tight span

Nevertheless, this approach suggests a number of interesting, purely mathematical questions which to pursue might still be helpful in this context: E.g., it leads to the question which dissimilarities are *tree like* dissimilarities, i.e. which dissimilarities would fit exactly into a tree, and whether that tree would be completely determined by those dissimilarities. Fortunately, these two questions have simple answers that have been discovered in the sixties and seventies of the last century independently by various mathematicians (cf. [5, 29, 30]):

(i) A dissimilarity $d$ is tree like if and only if

$$d(x,y) + d(u,v) \leq \max\{d(x,u) + d(y,v), d(x,v) + d(y,u)\}$$

holds for all $x, y, u, v$ from $X$.

(ii) If this condition is fulfilled, there is only one tree that fits the given dissimilarity (up to isomorphism, and except for additional branches not involved with the given data).

Remarkably, once we define a metric on *all* points of that tree (whether a branching point, an end point, or just a point somewhere on some branch) by associating again to any two such points $x, y$ the total length of the unique path from $x$ to $y$, the resulting metric space, necessarily an $\mathbb{R}-$tree (by the very definition of $\mathbb{R}-$trees) actually coincides with the injective hull of the metric defined on its leaves. This establishes not only the uniqueness of the tree in question; it can also be used to study the structure of that tree in terms of the metric defined on its leaves. More importantly, it suggests to use the injective hull in any case, whether or not the input dissimilarities satisfy the above four-point condition, as a good substitute for the tree in question — at least, it is always simply connected (though not always of dimension one).

In particular, if there exists some subset $K$ of small diameter within this injective hull $T$ not containing any leaf, yet such that its complement $T - K$ has several connected components, the (labels of the) leaves in at least all but one of these components have a good chance to form one of those clades within $X$ that phylogenetic analysis is designed to find.

In particular, the analysis of injective hulls of finite metric spaces made it obvious that the injective hull of a sum $d = d_1 + d_2 + \ldots + d_k$ of $k$ metrics $d_1, d_2, \ldots, d_k$ defined on a finite set $X$ is closely related to that of the summands $d_1, d_2, \ldots, d_k$ provided these metrics form a *coherent decomposition* of the metric $d$, i.e. provided there exist, for every map $f : X \to \mathbb{R}$ with $f(x) + f(y) \geq d_1(x,y) + d_2(x,y) + \ldots + d_k(x,y)$ for all $x, y \in X$, some maps $f_1, f_2, \ldots, f_k : X \to \mathbb{R}$ such that $f_i(x) + f_i(y) \geq d_i(x,y)$ holds for all $x, y \in X$ and for all $i = 1, 2, \ldots, k$ (cf. [2, 24, 25, 26]).

Moreover, defining a metric $d$ to be

- a *split* — or a *cut* — metric if there are exactly two subsets of $X$ in the set $X/d$ of equivalence classes of elements of $X$ relative to the equivalence relation $\simeq$ defined on $X$ by $x \simeq y \Leftrightarrow d(x,y) = 0$, and
- a *split-prime* metric if it cannot be decomposed into a coherent sum of a split metric and another metric,

it could be shown that

- every metric $d$ defined on a finite set $X$ has a unique coherent decomposition — also called the *canonical split decomposition* of $d$ — into a sum $d = d_1 + d_2 + \ldots + d_k + d_0$ of pairwise linearly independent split metrics $d_1, d_2, \ldots, d_k$ and a split-prime metric $d_0$ (possibly the 0-metric),
- the metrics $d_1, d_2, \ldots, d_k$ occurring in this decomposition are always linearly independent (as elements in the vector space of all maps from $X \times X$ into $\mathbb{R}$) — and so are $d_1, d_2, \ldots, d_k, d_0$ if $d_0 \neq 0$ holds,
- the metrics $d_1, d_2, \ldots, d_k$ occurring in this decomposition are – up to scaling – exactly those split metrics $d'$ defined on $X$ for which $d - d'$ is also a metric and the two metrics $d', d - d'$ form a coherent decomposition of $d$,
- if $d$ is a tree-like metric, then the split-prime metric $d_0$ in the corresponding canonical coherent decomposition $d = d_1 + d_2 + \ldots + d_k + d_0$ of $d$ into a sum of pairwise linearly independent split metrics $d_1, d_2, \ldots, d_k$ and a split-prime metric $d_0$ vanishes while the split metrics $d_1, d_2, \ldots, d_k$ correspond in a one-to-one fashion to the branches of the associated tree

This was of considerable interest within the context of phylogenetic analysis: If a split metric $d'$ occurs as a summand in a coherent component of a metric $d$ derived from a family of phylogenetically related sequences, there is a good chance that at least one of the two equivalence classes in $X/d'$ is one of those clades within $X$ that we want to find.

In particular, given any metric $d$ defined on a set $X$ of cardinality $n$, the linear independence of the split metrics occurring in the canonical decomposition of $d$ implies that there exist, up to scaling, at most $\binom{n}{2}$ split metrics $d'$ such that (i) $d - d'$ is also a metric and (ii) the two metrics $d', d - d'$ are coherent, – clearly too many to fit into a tree (because a tree with $n$ leaves has at most $2n - 3$ edges), but surely much less than $2^{n-1} - 1$, the number of all split metrics that, up to scaling, can be defined on an $n$-set.

Consequently, algorithms were developed to compute, given any metric $D$, all split metrics $d$ for which the above conditions are fulfilled as well as to visualize the resulting *split network* (cf. [3, 12, 22]). The resulting SplitsTree program has proven useful in diverse phylogenetic applications. Moreover, as Figure 6 shows, it can as well be applied to all sorts of distance data: The split networks in Figure 6(left) was computed for the distances between the towns of Wellington on the North Island, and Christchurch, Greymouth etc. on the South Island of New Zealand that were taken from a mileage chart. If one compares this graph with a map of New Zealand a good correlation between the distribution of vertices and the geographical locations of the towns is observed. It has also been applied to analyze the perceived similarity of colors and — in *stemmatology* — the "kinship" relations between the various hand-written versions of Chaucer's *Canterbury tales* written by Geoffrey Chaucer about 100 years before book printing was invented (in central Europe) (cf. [4]).

These examples illustrate that split networks can give meaningful representations of data even if they are not necessarily tree-like in character. Within biology, non tree-like distances often arise when analyzing viral data sets, a phenomenon that is probably caused by more complex evolutionary processes such as recombination.

## Back to mathematics and quadratic forms

In addition to these applications, there are also striking analogies between split-decomposition theory and the theory of positive semi-definite quadratic forms as developed by the Russian school: In both fields, one considers a large convex cone (either consisting of all metrics defined on a finite set or consisting of all positive semi-definite quadratic forms defined on some finite-dimensional vector space), one has good reasons to decompose this cone — in one way or the other — into a family of finitely generated convex subcones, and one wants to understand the combinatorics of the resulting stratification of the large cone. In split-decomposition theory, it is the concept of *coherence* that gives rise to the stratification in question: given any two metrics $d$ and $d'$, defined on a fixed finite set $X$, one may define the metric $d'$ to be a *coherent specialization* of the metric $d$ if there exists some positive real number $\rho$ such that $d'' := \rho d - d'$ is also a metric and the two metrics $d', d''$ form a coherent decomposition of $d$. One can show that, given any metric $d$ defined on $X$, the collection of metrics $d'$ that are coherent specializations of $d$ forms a finitely generated convex subcone $C(d)$ of the cone of all metrics defined on $X$. Moreover, some (not at all obvious) conditions on $d$ are known from split-decomposition theory which imply that $C(d)$ is a simplicial cone while this does not seem to hold in general for every metric $d$.

Very similar problems have been (and still are being) studied in the theory of positive semi-definite quadratic forms while trying to understand the process of reduction of quadratic forms (cf. [17, 18]). And in both areas, the extremals of the convex cones in question — the positive semi-definite quadratic forms of rank one on the one hand and the split metrics as well as some further, not yet well understood metrics on the other — appear to be of special significance.