ELSEVIER

Discrete Applied Mathematics 71 (1996) 95–109

DISCRETE APPLIED MATHEMATICS

Analyzing and visualizing sequence and distance data using SPLITSTREE

A. Dress, D. Huson*,[1], V. Moulton[2]

FSPM-Strukturbildungsprozesse, University of Bielefeld, Postfach 10 01 31, 33501, Bielefeld, Germany

Received 10 June 1995; revised 31 May 1996

Abstract

In this paper, we describe and illustrate a tool for analyzing and visualizing sequence and distance data, called the *splits-graph*. The construction of this graph is based upon the *split-decomposition* technique which is a procedure to decompose a given metric defined on a finite set in a canonical way into a sum of simpler metrics. In a way, this technique is comparable to Fourier analysis which also decomposes a given object under consideration (that is a periodic signal) into a sum of simpler such objects, in a canonical way. The splits-graph and the theory behind it have been developed mainly in Bielefeld over the last 5 years. The procedure for producing splits-graphs implemented in the SPLITSTREE program is also described and it is available from the authors.

1. Introduction

One of the main problems in phylogenetic analysis is to find a good method for *analyzing* and *visualizing* a phylogenetic distance data set, in order to understand better the phylogenetic relationships that exist between the taxa within this set. The aim of this paper is to describe one such method which produces what we have come to call the *splits-graph*, a data anlaysis technique that has been developed over the last 5 years in Bielefeld and which is based on the *split-decomposition* method, a method for decomposing metrics canonically into a sum of simpler metrics, developed jointly with Bandelt in [7].

The mathematical field devoted to structuring and/or visualizing data sets according to pregiven (or readily deduced) similarity relationships is often called *cluster theory*.

* Corresponding author. E-mail: huson@mathematik.uni-bielefeld.de.

[1] Supported by the Deutsche Forschungsgemeinschaft and in part by the European Union's "Algebraic Combinatorics" project.

[2] Supported by a scholarship from the European Union's "Algebraic Combinatorics" project.

PII S0166-218X(96)00059-5

More precisely, cluster theory aims at structuring a set X by specifying a system $\mathscr{C}(X)$ of subsets of X, called *clusters*, subject to the following conditions (see [5]):

- The clusters should collect *similar* objects, that is objects in a given cluster $C \in \mathscr{C}(X)$ should somehow be more similar to each other than to objects outside C.
- The clustering procedure should be reasonably *stable*, that is, addition, elimination, and/or small changes of a few aspects (e.g. positions of sequences) or even small changes of X should not result in a drastically different system of clusters.
- The set of clusters $\mathscr{C}(X)$ should be *informative*, that is it should be reasonably small (e.g. it should grow at most polynomially if not linearly with the size of X) and, simultaneously, it should be reasonably large (e.g. not equal to just $\{\emptyset, X\}$) and, if possible, it should in addition contain reasonably sized subsets (not only very small or very large subsets).
- $\mathscr{C}(X)$ should be *computable*: without a reasonably fast algorithm to compute the clusters of $\mathscr{C}(X)$, even the best theory could not be used in practice.
- Finally, sometimes (for example in evolutionary biology) the clusters should be non-overlapping and thus form a *hierarchy*, that is, for all $C, C' \in \mathscr{C}(X)$, with $C \cap C' \neq \emptyset$, the intersection $C \cap C'$ should equal either C or C'.

This could be achieved easily if it were not for the notorious *intransitivity* of similarlity, the crux of cluster theory. Many attempts have been made and many clever schemes have been designed to overcome this problem. While some of the most popular classification procedures aim directly at constructing hierarchical classification schemes (or *tree-like structures*) which approximate as accurately as possible a given *scheme of diversity* – usually a *metric space* – others are less restrictive and allow the detection of parallel and convergent evolutionary events, as well as hybridization effects due to gene exchange in addition to phylogenetic kinship relations, leading to trees only if the data set unambiguously supports a unique tree. These less restrictive methods include the *spectral analysis* of phylogenetic data sets, introduced by Hendy and Penny [16], the analysis of *weak hierarchies* associated with distance data sets [5, 9], and the *split decomposition* method which we describe in this paper.

In general, it is impossible to reconstruct unambiguously the true phylogenetic tree structure for any given phylogenetic data set, independently of whether one uses morphological, fossil, or molecular records. A case in point, for example, is the ongoing debate concerning the mutual phylogenetic relationship between sponges, fungi, plants, and animals.

For most reconstruction methods used in phylogenetic analysis, these facts are reflected in the highly *unstable* solutions of the tree construction problem which may easily switch to another one upon deletion of a few characters (or positions when DNA or amino-acid sequences are analyzed) or upon adding other taxa; in the worst case, these solutions may even strongly depend upon the consecutive labeling of the taxa under consideration. Hence, people have to check their results by all sorts of bootstrap methods and to trust only those phylogenetic groupings for which a high *concensus* is reached. Thus, it is desirable to have a method at hand which does not even try to

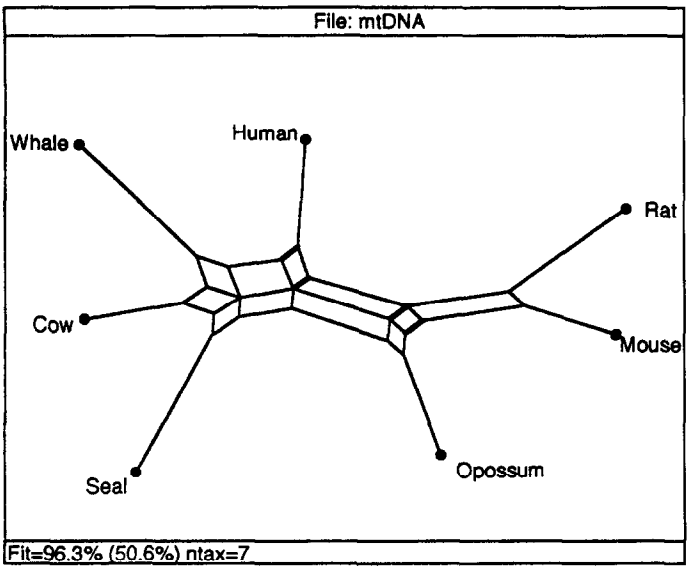

Fig. 1. Splits-graph for six mammals and one marsupial based on mitochondrial DNA. Splits are represented by single edges or by bands of parallel edges. For example, the four bold edges represent the split that separates human, rat and mouse from the other four species.

construct a tree-like branching pattern whenever such a pattern is not clearly supported by the data set.

It is the purpose of this paper to discuss one such technique – the above mentioned split decomposition method – that has been fully implemented in the SplitsTree program [17, 26] and has proved useful in many different contexts. We explain some of the theory behind split decomposition in the next section and describe briefly our implementation in Section 3. Finally, we illustrate the split-decomposition method with an assortment of examples in Section 4.

Before we proceed, we give a brief example to illustrate split decomposition. Consider the *splits-graph* depicted in Fig. 1. This graph is a visualization of the phylogenetic distance data set obtained by analyzing mitochondrial DNA from the taxa whale, mouse, seal, rat, man, opossum, and cow (see [19, 26] for more details). It is built up of parallelograms (sometimes also, more generally, from *zonotopes*, that is, center-symmetric polygons) and individual edges. Consequently, the geometric structure of the graph gives rise to *bands of parallel* edges, where by a band we mean a minimal set of edges which, with any edge e, also contains every edge e' which is opposite to e in some parallelogram (or zonotope) containing e.

We interpret this graph as follows. The sum of the lengths of all the edges along a shortest path from one taxon to another is proportional to (a canonically defined approximation of) the actual distance between those two taxa in the data set. Recall that in a tree, any edge partitions the tree into two connected components and, consequently, it partitions the set of taxa into two nonempty, disjoint subsets, thus forming what is

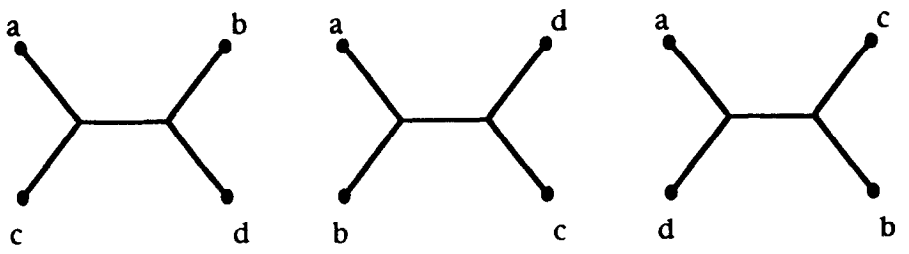

Fig. 2. The three nondegenerate additive tree topologies on four objects.

called a *split*. In our more general setting, such a split cannot always be represented by just a single edge, but will give rise to a band of parallel, equally long edges as described above, whenever the given data is not "tree-like". The length of any one of these parallel edges is called the *isolation index* of the split. In essence, this index tells us how far apart the two subsets are. Thus, in this example, we see that mouse and rat form one cluster which is separate from the other taxa. We also see that the graph gives us a hierarchical way to cluster the data set. For example, even though the taxa not equal to either rat or mouse belong in a separate cluster, we can subdivide this into three subclusters, namely: whale, cow, and seal; man; and opossum.

In fact, this example also illustrates that split decomposition usually behaves very well with respect to most of the above requirements for a good clustering technique. As we have seen, the number of clusters as well as their sizes appear to be reasonable, and the clusters form a hierarchical structure. Moreover, they are clearly informative in the sense described above. Split decomposition is generally quite stable (for this example see [26]), and the computation is fast for reasons that we describe below. Another important feature is the *splittability index* which is a "goodness of fit estimate" for the splits-graph, and which gives us a measure of how accurate the representation of the data set is. In this example, the fit is particularly good, being 96.3%.

2. How split decomposition works

The theory behind the *splits-graph* technique was developed jointly with Bandelt in [7]. Algorithms and techniques for visualizing splits-graphs were developed jointly with Bandelt and Wetzel (cf. [26]).

Given a finite set X, a *split* of X is a bipartition of X into two nonempty subsets A, B. The main idea behind split decomposition is to construct, for a phylogenetic data set X, *global* phylogenetic splits $X = A \cup B$ which (hopefully) separate one monophyletic group A from all other organisms in question, using given *local* information in a rather relaxed way only. To understand this statement more fully, consider the three possibly nondegenerate, additive tree topologies definable on the set $\{a, b, c, d\}$ depicted in Fig. 2.

First, note that one way to construct a *tree* structure on a data set X is to specify, for each quadruple $\{a,b,c,d\}$ of X as above, the most probable of the three nondegenerate tree topologies for this quartet. Then, using this information, a set of global splits is constructed, consisting precisely of those partitions of the total set X into two disjoint subsets A and B which never place two organisms a,a' into A and two others b,b' into B unless, in the list of the local data, the tree topology considered for the quartet $\{a,a',b,b'\}$ is the one which separates a,a' from b,b'.

In contrast, split decomposition works on the following principle. Instead of proceeding as above, we just exclude, for any four organisms, the most improbable of the three tree topologies. We then accept as candidates for potentially relevant evolutionary splits all those global splits which never realize, for any given quartet, the excluded most improbable grouping. In this situation, the resulting system of splits may not fit into a tree since we may encounter pairs of *incompatible splits*, i.e. pairs of splits A,B and A',B' with $U \cap V \neq \emptyset$ for all $U \in \{A,B\}$ and $V \in \{A',B'\}$ (see [4] for a more detailed discussion of this concept). However, as we have indicated in Section 1, the resulting system can be represented by an associated, canonically defined network, which we call the *splits-graph*. In addition, and even more importantly, the fact that for any quartet $\{a,b,c,d\}$ no three splits can simultaneously realize all three tree topologies implies that there cannot be more than $\binom{n}{2}$ global splits if the total set X has cardinality n (see [7, p. 62]).

At first glance, one might expect this procedure to be even worse than the standard tree reconstruction methods: while the artefacts resulting from the construction principles in standard methods usually consist of single unreliable (or completely missing) edges in the suggested tree, here we may have a whole network of such edges. Such pronounced nettedness can, however, be taken as evidence that none of the involved edges have strong phylogenetic support in the data set (see Example 3 in Section 4) and so, it can contain highly valuable phylogenetic information.

We now briefly summarize one possible method by which a split system and a corresponding splits-graph can be constructed. Assume that we are given a distance matrix $d = (d_{ij})_{1 \leqslant i,j \leqslant n}$ of dissimilarities between pairs of taxa $X = \{1,\ldots,n\}$. We call a bipartition of X into two disjoint, nonempty subsets A,B a *d-split*, and represent it simply as the pair A,B, if, for all $i,j \in A$, and $k,l \in B$, we have

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

This amounts to excluding for any quartet i,j,k, and l, the tree topology which separates, say, the taxa labeled i and j from those labeled k and l in case one has $d_{ij}+d_{kl} \geqslant d_{ik}+d_{jl}$ and $d_{ij}+d_{kl} \geqslant d_{il}+d_{jk}$, and accepting as d-splits all those splits A,B of X which for any given quartet as above, never induce the excluded tree topology.

Each d-split then receives a positive weight, namely, the quantity

$$\alpha_{A,B} := \alpha^d_{A,B} := \tfrac{1}{2} \min_{i,j \in A,\ k,l \in B} \{\max(d_{ik} + d_{jl}, d_{il} + d_{jk}) - d_{ij} - d_{kl}\},$$

which is called the *isolation index* of A,B. Note that if, independently of whether or not a given split A,B is a d-split, we define the isolation index for every split A,B of X by

$$\alpha^d_{A,B} := \tfrac{1}{2} \max_{i,j\in A,\ k,l\in B} \{\max(d_{ik}+d_{jl}, d_{il}+d_{jk}, d_{ij}+d_{kl}) - d_{ij} - d_{kl}\},$$

then all d-splits will receive the same isolation index as before, while all bipartitions of the taxa which are not d-splits will have their isolation index equal to 0.

For each split $X = A \dot\cup B$, we next define the *split metric* $\delta_{A,B}$ associated with A,B, to be the (pseudo-)metric which assigns distance 1 to any two taxa in different parts of the split, and distance zero otherwise. In [7], it is shown that the inequality

$$\sum_{\text{splits } A,B} \alpha_{A,B} \cdot \delta_{A,B}(x,y) \leqslant d(x,y)$$

holds for all $x, y \in X$, and that the resulting map $d^0 : X \times X \to \mathbb{R}_{\geqslant 0}$, defined by

$$d^0(x,y) := d(x,y) - \sum_{\text{splits } A,B} \alpha_{A,B} \cdot \delta_{A,B}(x,y),$$

is a (pseudo-)metric which does not admit any further splits with positive isolation index. Hence, the metric d^0 is also called the *split-prime residue* (*of* d). By its very definition, it can be used to decompose[3] the metric d in the form

$$d = d^0 + \sum_{\text{splits } A,B} \alpha_{A,B} \cdot \delta_{A,B}.$$

In most cases, the split-prime residue is nonzero. Yet, when the data set is fairly tree-like, it is small in comparison to $d^1 := d - d^0$. Intuitively, this reflects the fact that, when each quartet in the data set satisfies the so-called *four-point condition*, the split-prime residue vanishes, and (as shown in [7]) the split decomposition is exactly the same as the decomposition one would get by summing the set of weighted split metrics associated to the splits obtained from deleting single edges in the unique tree fitting the data set (see [4] and the references quoted there for a discussion of the relationship between trees and metrics satisfying the four-point condition). Hence, in practice, the splits-graph also tends to exhibit tree-like features for tree-like data sets (see Example 1, Section 4).

In general, to measure the effectiveness of the split decomposition procedure, the *splittability index*,

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

was introduced, which can be viewed as an indication of the amount of the original distance information that is still present in the weighted system of splits.

[3] This decomposition can be characterized abstractly by certain structural requirements relating to concepts from category theory, applied to the category of metric spaces and nonexpanding maps, cf. [11, 13, 18].

The d-splits can be found simply and efficiently since, as mentioned above, the number of all d-splits is bounded by $\binom{n}{2}$. The procedure is recursive, and we describe it here. Suppose that, as above, $X := \{1,\ldots,n\}$ and that the d-splits restricted to the subset $\{1,\ldots,i-1\}$ are already determined; then, for each d-split A,B of this subset, check whether $A \cup \{i\},B$ or $A,B \cup \{i\}$ is a d-split on the set $\{1,\ldots,i\}$. Also, check whether or not $\{1,\ldots,i-1\}$, $\{i\}$ is a d-split. Clearly, this procedure ends when we have included all n taxa. As stated in [7], the total number of steps is bounded by a polynomial in n of degree 6, with a small leading coefficient. In addition, as experience has shown, the average computation time is considerably lower.

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. To illustrate this process, we discuss an example. Suppose we have a taxa set $\{a,b,\ldots,g\}$, with the following set of splits: $S_1 := \{a,b,f,g\}, \{c,d,e\}$, $S_2 := \{a,f,g\}, \{b,c,d,e\}$, $S_3 := \{a,b,c,d\}, \{e,f,g\}$, and $S_4 := \{a,b,c,g\}, \{d,e,f\}$. We start with a vertex labeled $\{a,b,\ldots,g\}$, as depicted in Fig. 3(a). Then, choosing the first split S_1, we "pull apart" this node to produce a two vertex graph as pictured in Fig. 3(b), whose vertices are labeled by two sets in S_1. Now "pull apart" this two vertex graph according to the way in which S_2 divides the parts of the split S_1, to get Fig. 3(c) and then again, using S_3, to get Fig. 3(d). Finally, "pull apart" the graph in Fig. 3(d), using the split S_4 to get the graph in Fig. 3(e).

In general, if a graph $\Gamma = (V,E)$ together with a labeling $\varphi : X \to V$ representing some given splits $S_1,\ldots,S_{k-1}$ of X has already been constructed, and if $S = \{A,B\}$ is an additional split of X to be represented along with $S_1,\ldots,S_{k-1}$, then consider the two induced subgraphs Γ_A and Γ_B of Γ which are defined as follows. The graph Γ_A has vertex set V_A, which consists of all those vertices $v \in V$ such that there exist $a,a' \in A$ and a shortest path in Γ from $\varphi(a)$ to $\varphi(a')$, which passes through v, and the graph Γ_B has vertex V_B which is defined in the same way using elements from B. Let

$$V' := \{(v,T) \in V \times \{A,B\} \mid v \in V_T\},$$

$$E' := \left\{ \{(v_1,T_1),(v_2,T_2)\} \in \binom{V'}{2} \;\middle|\; v_1 = v_2, \text{ or } T_1 = T_2 \text{ and } \{v_1,v_2\} \in E \right\},$$

$\Gamma' := (V',E')$, and define a new labeling $\varphi' : X \to V'$ by $\varphi'(x) := (\varphi(x),S_x)$, where S_x denotes the subset A or B which contains x. A repeated application of this procedure, starting with the trivial graph $\Gamma_0 := (\{*\},\emptyset)$ and the trivial labeling $\varphi_0 : X \to \{*\}$, which maps each element in X onto $*$, incorporating a given family $S_1,\ldots,S_N$ of distinct splits consecutively, always produces a subgraph Γ_N of the "N-dimensional hypercube", whose vertices consist of *all* maps

$$v : \{1,\ldots,N\} \to \{A \subseteq X \mid \{A,X-A\} \in \{S_1,\ldots,S_N\}\}$$

satisfying the condition $v(i) \in S_i$, and whose edges consist of all (unordered) pairs of such maps which differ at precisely one index $i \in \{1,\ldots,N\}$. A map v is labeled by

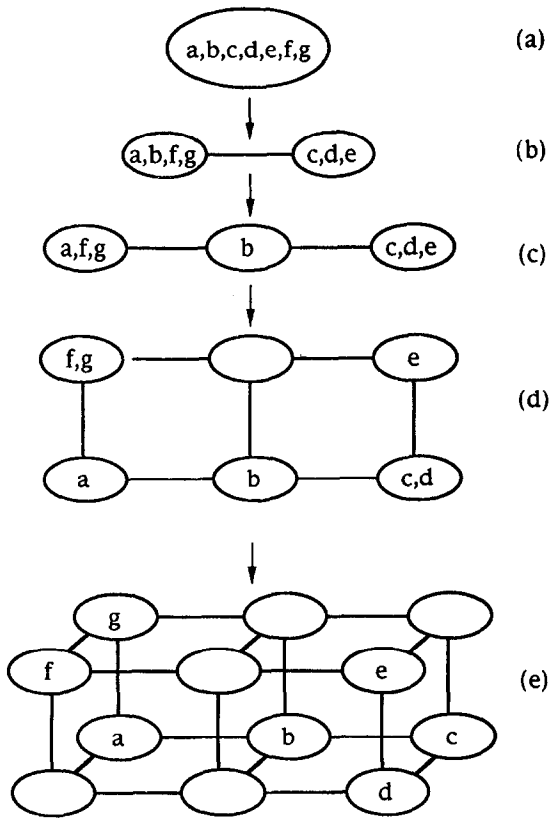

Fig. 3. Producing a subgraph of the four-dimensional hypercube.

some $x \in X$ if and only if $v(i) = S_i(x)$ for all $i \in \{1,\ldots,N\}$. More precisely, it can be shown that – independently of the indexing of the involved splits – the graph Γ_N is necessarily isomorphic to the induced subgraph of that hypercube graph whose vertex set consists of all maps v with $v(i) \cap v(j) \neq \emptyset$, for all $i,j \in \{1,\ldots,N\}$.

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. For example, if the splits S_1, S_2, S_3, S_4 had weights 5, 3, 1, and 2, respectively, then we would produce the graph in Fig. 4.

In the case of d-splits, this graph represents the d^1 summand of the split decomposition $d = d^0 + d^1$ since, by removing any set of parallel edges, we obtain one of the original d-splits and, by looking at the length of the removed parallel edges, we obtain the isolation index for that particular split.

If this graph is planar, then it is exactly the splits-graph. Unfortunately, this procedure will rarely produce a planar result; hence we need other techniques to find a better, and (if possible) a planar representation of the data set.

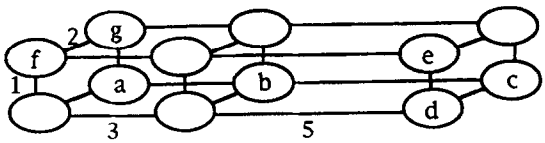

Fig. 4. The weighted version of the graph in Fig. 3(e).

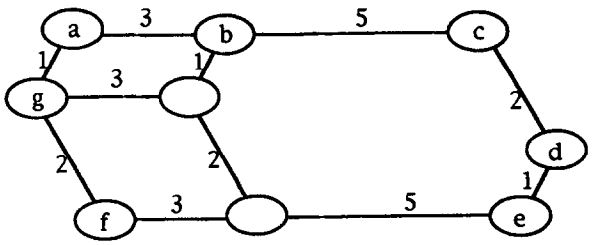

Fig. 5. The weighted graph with redundant edges removed.

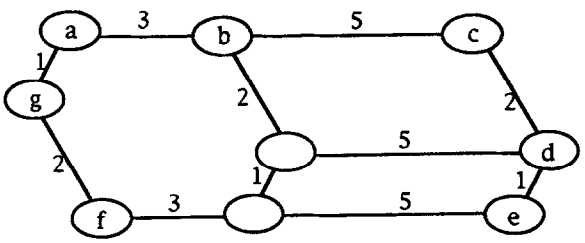

Fig. 6. A different representation of the same four splits.

We illustrate one method used for producing a more transparent splits-graph as follows. 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. For example, the graph depicted in Fig. 6 is a different representation of the same four given splits.

Finally, we note that when our splits form a *cyclic system* of d-splits, i.e. a system of splits that can be represented by splits obtained by dividing a set of points which are spaced equally on a circle in the plane by intersecting this circle with appropriately chosen straight lines, then it is always possible to construct a planar splits-graph

(see [26]). In addition, there are particularly efficient ways to detect cyclic split systems and, once one is found, to produce a planar splits-graph for this system, all of which have been implemented in SplitsTree.

3. The program SplitsTree

The program SplitsTree (see [17] for availability) is a C++ implementation of the split decomposition method, that runs on any unix or macintosh computer. The application is based on algorithms and code developed jointly with Bandelt and Wetzel [26]. In general terms, the program takes as input any number of taxa in terms of aligned sequences or pairwise distances and produces as output a graph, the *splits-graph*, which indicates how the different taxa are related to each other. In particular, the graph gives an immediate indication of which possible phylogenies are supported by the data set, and to what degree. A nice feature of this program is that the more "tree-like" the input data set is, the more tree-like the graph becomes (see Section 2). Deviations from this ideal lead either to more numerous and more boxlike polygons in the splits graph, or – in the worst case – just to something like a bush.

SplitsTree allows you to open a file containing either a number of aligned sequences, a distance matrix describing the distances between some given taxa, or a system of splits. The application is based on the NEXUS file format [22]. Upon opening a file of sequences, the application first computes the corresponding distance matrix, using one of the following transformations specified by the user: Hamming distances, Kimura 3ST, Jukes-Cantor, or the LogDet transformation, recently introduced by Mike Steel [23]. Moreover, you can specify which of the sequences in the input file should be used, the range of sites (or "positions"), and whether to consider gap sites, non-parsimony sites, or constant sites in the computation (in connection with LogDet, it may for instance make sense to ignore a certain percentage of constant sites).

Once a distance matrix d for a set of taxa X has been computed, or is given as input, SplitsTree first computes the split decomposition of d and then computes and draws the corresponding splits-graph $G = (V, E)$. A subset of the vertex set V is labeled by the elements of X. Additionally, V contains unlabeled interior vertices. Each split A, B is represented by a set of parallel edges $S \subseteq E$, that separate two connected components in G, one containing all vertices representing the objects in A and the other containing all vertices representing those in B. The length of each edge in S is proportional to the isolation index $\alpha_{A,B}$ and thus indicates how significant the split is. Alternately, the program can draw the graph with all edges having equal length, emphasizing the combinatorial structure of the graph.

The program offers basic editing facilities such as zooming, rotating, flipping, and reshaping of the graph. Moreover, the computed graph can be copied and pasted into a drawing or writing program. Given a set of sequence, SplitsTree can be asked to label the vertices of the graph by the characters of the sequences at any chosen site. This is useful for determining which sites support the indicated splits.

Additionally, the program contains two other methods for computing splits from a set of given sequences, namely spectral analysis [16, 24] followed by a *greedy* selection of a weakly compatible system of splits, and the calculation of *p-splits* [8]. Bootstrapping can also be performed on all calculations.

4. Examples of applications of SplitsTree

Split decomposition has been applied successfully to numerous data sets mostly from biology and psychology. For example, it has been applied to the evolution of the foot and mouth disease virus [10], genetic relationships in human populations [3], and distinguishing fish populations [1]. Here, we give three brief examples, two from biology, and one from psychology, in order to illustrate the application of SplitsTree. For further and more detailed examples, see also [1, 3, 6, 8, 21, 25, 26].

The first example, depicted in Fig. 7, is the splits-graph obtained from the 23S ribosomal RNA sequences of 6 archaebacteria, 6 eubacteria (including 2 chloroplasts), and 4 eukaryotes, studied by Leffers et al. [20]. Biological data sets typically gives rise to slightly more splits than can be fitted into a tree. This example illustrates that a large portion of these fit together on a tree. In addition, the split-prime residue is rather small (the splittability index in this case is 87.9%). In contrast, randomly generated distance data sets tend to have a rather large residue (in practice, the splittability index of randomly generated sequence families consisting of 10 or more sequences is considerably smaller than 50%) and to produce mostly *trivial* splits which separate one

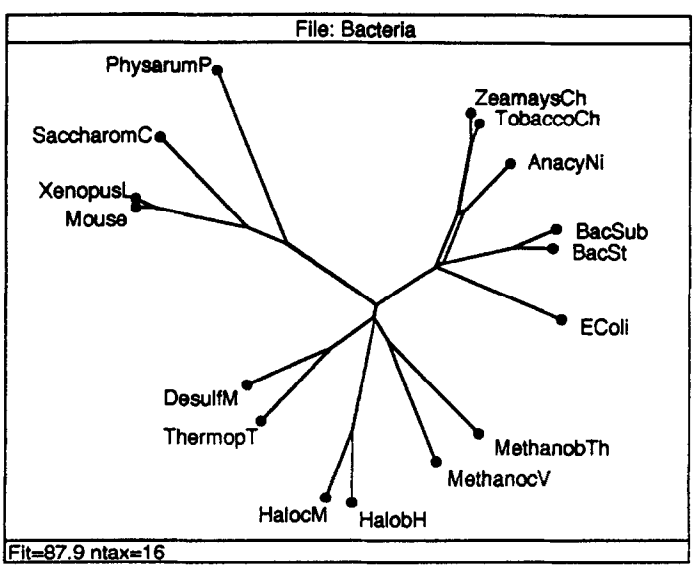

Fig. 7. Splits-graph obtained from the 23S ribosomal RNA sequences of 6 archaebacteria, 6 eubacteria (including 2 chloroplasts), and 4 eukaryotes.

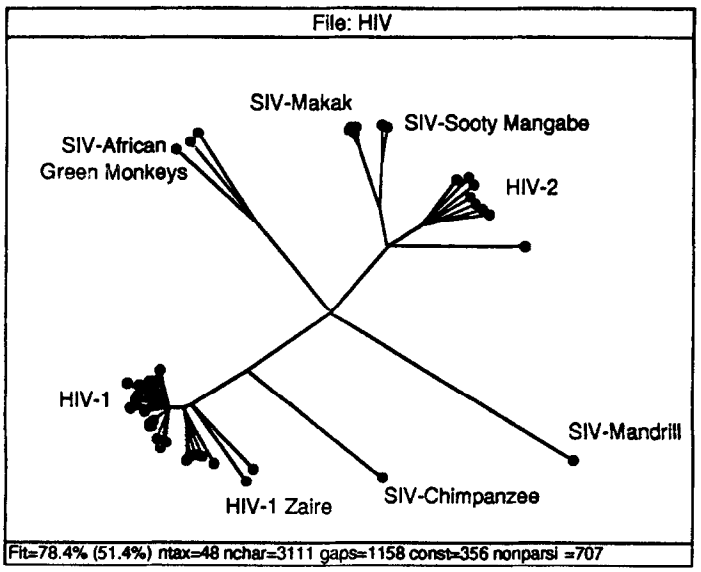

Fig. 8. The evolution of the AIDS-virus.

taxon from all of the others, and almost none which separate more than two taxa from the rest (thus producing an almost *bush-like* structure).

The second example is an application to a data set arising from the AIDS-virus (for more details see [14]). The splits-graph in Fig. 8 clearly shows the evolutionary history of the AIDS-virus. While it seemingly coevolved with the immune system of apes and monkeys, adapting to the evolutionary pressures that it experienced there, the diagram suggests that there must have been two independent events by which humans were infected with these viruses, giving rise to the HIV-1 and HIV-2 family. This example is particularly interesting since it shows how the splits-graph can be used to identify "explosive" evolutionary events. Also, it should be noted that in this example the data set is again quite tree-like, which is reflected in the nature of the splits-graph.

The final example comes from a data set obtained in cognitive psychology [15], see also [26]. In Helm's experiment, 10 people with normal eyesight and 4 color-blind people were each asked to rank the similarity of 10 colors. The experiment went as follows: For any three colors, the test subject was first asked to decide which two were least similar. She then had to estimate the distance of the third color to the other two, using colored counters on a board. From this set of data (120 triplets per test subject), Helm computed a distance measure on the set of 10 colors. Fig. 9 shows the splits graph corresponding to the distance measure obtained from the ten persons with normal eyesight, whereas the distance measure produced by one of the four color-blind persons is depicted in Fig. 10.

Note that the splittability index in Fig. 9 is 97%, hence the graph very closely represents the given distance measure. We see that the split that has the largest isolation index separates the two "warmest" colors *yellow* and *red* from the others. Moreover,

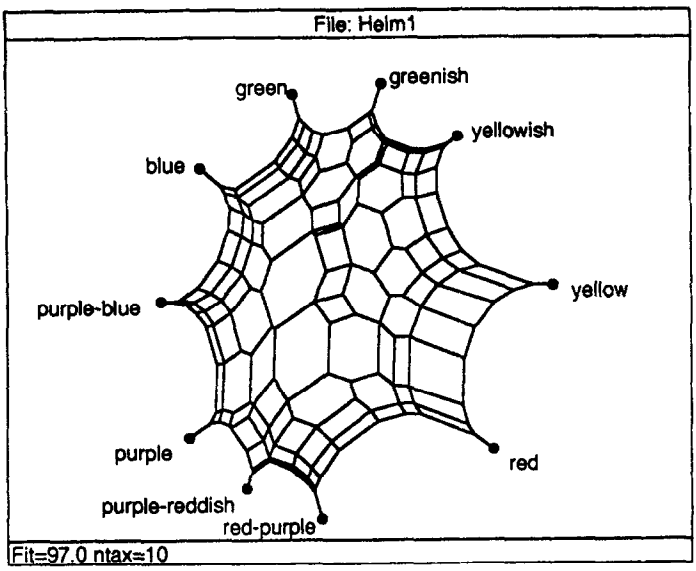

Fig. 9. Splits-graph showing distances between different colors, estimated by 10 persons with normal eyesight. Bold lines indicate the only non-trivial splits that also occur in Fig. 10.

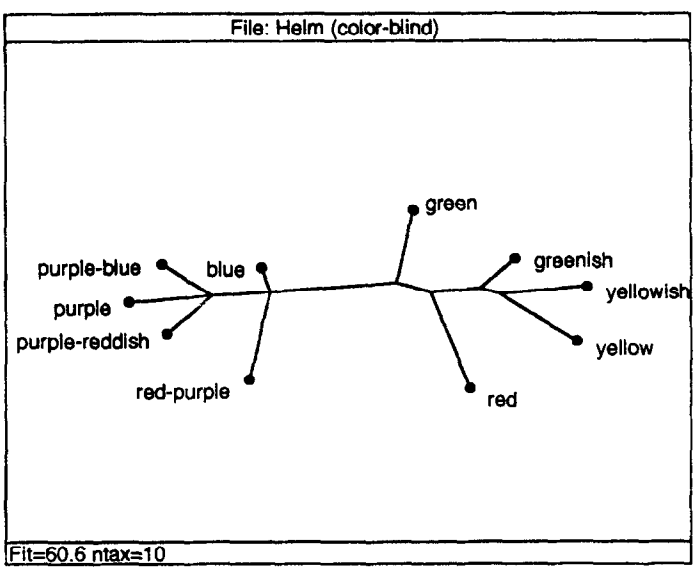

Fig. 10. Splits-graph showing distances between different colors, estimated by a color-blind person.

the colors *purple*, *purple-reddish* and *red-purple* all lie close to each other, as do *green*, (*green-yellow*) *greenish* and (*green-yellow*) *yellowish*. Moreover, the graph clearly approximates the well-known color-circle and the distances between pairs of diametrically opposite colors are very similar to each other.

In contrast, we find that the splittability index is only 60.6% in Fig. 10. This low value indicates that the data set is either quite noisy or, more probably, that the distance measure does not fit too well into the framework defined by split decomposition theory. Indeed, Helm stated that these distances can only be visualized in a higher dimensional space.

Finally, this example illustrates the fact that the splits-graph can be an effective data analysis tool even when the graph is rather grid-like (i.e. has no definite tree-like structure).

References

[1] H.-J. Bandelt, Phylogenetic networks, Verhandl, Naturwiss (Vereins, Hamburg, 1994).
[2] H.-J. Bandelt, Combination of data in phylogenetic analysis, Plant Syst. Evol., to appear.
[3] H.-J. Bandelt, Reticulate diagrams displaying genetic relationships between human populations, Ann. Human Biol., to appear.
[4] H.-J. Bandelt and A. Dress, Reconstructing the shape of a tree from observed dissimilarity data, Adv. Appl. Math. 7 (1986) 309–343.
[5] H.-J. Bandelt and A. Dress, Weak hierarchies associated with similarity measures – an additive clustering technique, Bull. Math. Biol. 51 (1989) 133–166.
[6] H.-J. Bandelt and A. Dress, A new and useful approach to phylogenetic analysis of distance data, Mol. Phylogen. Evol. 1 (1992) 242–252.
[7] H.-J. Bandelt and A. Dress, A canonical decomposition theory for metrics on a finite set, Adv. Math. 92 (1992) 47–105.
[8] H.-J. Bandelt and A. Dress, A relational approach to split decomposition, in: O. Opitz, B. Lausen, R. Klar, eds., Information and Classification (Springer, Berlin, 1993) 123–131.
[9] H.-J. Bandelt and A. Dress, An order theoretic framework for overlapping clustering, Discrete Math. 136 (1994) 21–37.
[10] J. Dopazo, A. Dress and A.v. Haeseler, Split decomposition: a new technique to analyse viral evolution, PNAS 90 (1993).
[11] A. Dress, Trees, tight extensions of metric spaces, and the cohomological dimension of certain groups: a note on combinatorial properties of metric spaces, Adv. Math. 53 (1984) 321–402.
[12] A. Dress, Some mathematical problems arising in molecular bioinformatics, in: C. Colbourn and E. Mahmoodian, eds., Combinatorics Advances (Kluwer Academic Publishers, Dordrecht, to appear).
[13] A. Dress, V. Moulton and W. Terhalle, T-Theory, Eur. J. Combin. 17 (1996) 161–175.
[14] A. Dress and R. Wetzel, The human organism – a place to thrive for the immuno-deficiency virus, in: Proceedings of IFCS, Paris (1993).
[15] C.E. Helm, Multidimensional ratio scaling analysis of perceived color relations, J. Opt. Soc. Amer. 54 (1964) 256–262.
[16] M. Hendy and D. Penny, Spectral analysis of phylogenetic data, J. Classification 10 (1993) 5–24.
[17] D.H. Huson, SPLITSTREE2 – A tool for analyzing and visualizing evolutionary data, FSPM, Bielefeld University, Germany (1996); Available from: ftp::/ftp.uni-bielefeld.de/pub/math/splits.
[18] J. Isbell, Six theorems about metric spaces, Comment. Math. Helv. 39 (1964) 65–74.
[19] A. Janke, G. Feldmaier-Fuchs, W. Kelly, A.v. Haeseler and S. Pääbo, The marsuipal mitochondrial genome and the evolution of placental mammals, Genetics 137 (1994) 243–256.
[20] H. Leffers et al., J. Mol. Biol. 195 (1987) 43–61.
[21] P.J. Lockhart, A.C. Barbrook, D.H. Huson, M.A. Charleston and C.J. Howe, Are systematic biases in plastid sequences phlyogentically informative? (1996) in preparation.
[22] D.R. Maddison, D.L. Swofford and W.P. Maddison, NEXUS: an extendible file format for systematic information (1995).
[23] M.A. Steel, Recovering a tree from the leaf colorations it generates under a Markov model, Appl. Math. Lett. 7 (1995) 19–24.

[24] L.A. Szekely, M.A. Steel and Peter Erdös, Fourier calculus on evolutionary trees, Adv. Appl. Math. 14 (1993) 200–216.

[25] J. Wägele and R. Wetzel, Opinion: nucleic acid sequence data are not per se reliable for inference of phylogenies, J. Natural History 28 (1993) 749–761.

[26] R. Wetzel, Zur Visualisierung abstrakter Ähnlichkeitsbeziehungen, Dissertation, Universität Bielefeld (1995).