Version dated: January 16, 2014

FlatNJ: A novel network-based approach to visualize evolutionary and biogeographical relationships

Monika Balvočiūtė[1], Andreas Spillner[1], and Vincent Moulton[2]

[1]*Department of Mathematics and Computer Science, University of Greifswald, Germany;*

[2]*School of Computing Sciences, University of East Anglia, Norwich, UK*

Corresponding author: Vincent Moulton, School of Computing Sciences, University of East Anglia, Norwich, NR4 7TJ, United Kingdom; E-mail: vincent.moulton@cmp.uea.ac.uk.

Abstract.— Split networks are a type of phylogenetic network that allow visualization of conflict in evolutionary data. We present a new method for constructing such networks called FlatNetJoining (FlatNJ). A key feature of FlatNJ is that it produces networks that can be drawn in the plane in which labels may appear inside of the network. For complex data sets that involve, for example, non-neutral molecular markers, this can allow additional detail to be visualized as compared to previous methods such as split decomposition and NeighborNet. We illustrate the application of FlatNJ by applying it to whole HIV genome sequences, where recombination has taken place, fluorescent proteins in corals, where ancestral sequences are present, and mitochondrial DNA sequences from gall wasps, where biogeographical relationships are of interest. We find that the networks generated by FlatNJ can facilitate the study of genetic variation in the underlying molecular sequence data and, in particular, may help to investigate

processes such as intra-locus recombination. FlatNJ has been implemented in Java and is freely available at `www.uea.ac.uk/computing/software/flatnj`.

(Keywords: phylogenetic network, split, split network, flat split system, NeighborNet, QNet)

Phylogenetic networks are useful for representing evolutionary scenarios that are not described sufficiently well by a single phylogenetic tree. There are several types of phylogenetic networks and various methods have been proposed for their construction (for an overview, see Huson et al. 2010). Here we are concerned with *split networks* (also known as a "data-display networks"), a type of phylogenetic network that is commonly used in exploratory data analysis (Huson and Bryant 2006; Morrison 2010). Split networks are designed to represent character (or tree) conflict in a data set, without making prior assumptions about the causes of those conflicts. Such conflicts might be caused, for example, by horizontal gene transfer or recombination, homoplasy or methodological issues in data collection or analysis.

Split networks have been used in various applications including the evolutionary analysis of viruses (Tugume et al. 2010), plants (Goremykin et al. 2013), microbes (Octavia and Lan 2006), animals (The STAR Consortium 2008), and even languages (Dunn et al. 2005). As an illustrative example, consider the split networks in Figure 1, which we generated from subcollections of a *Simian immunodeficiency virus* (SIV) data set published in Pelletier et al. (1995) and analyzed using split networks in Wain-Hobson et al. (2003). Each network in this figure represents a collection of *splits* or bipartitions of the taxa that label the network. In particular, each split is represented by a band of parallel edges that all have the same length. For example, the band of bold edges in network N1 represents the split that groups taxa 104 and 119 together versus the remaining taxa. This is much the same as the way in which each edge of an unrooted phylogenetic tree represents a split of its leaf set.

The boxes that appear in many of the networks in Figure 1 indicate pairs of splits that are *incompatible*, that is, pairs of groupings that cannot be represented simultaneously in a single phylogenetic tree. Therefore, such boxes indicate that the data are not treelike. In this particular example, some of the boxes are probably the result of intra-locus recombination. For example, the box in network N1 with one vertex labeled 119 indicates that taxon 119 shares similarities with both taxon 203 and taxon 104. This suggests that taxon 119 could be a recombinant, although a more detailed recombination analysis would have to be performed to verify this possibility.

The two methods for generating split networks that are most relevant to this paper are *split decomposition* (Bandelt and Dress 1992b) and *NeighborNet* (Bryant and Moulton 2004). Both are implemented in the SplitsTree package (Huson and Bryant 2006), and the networks generated by them for the SIV data set are depicted in the top two rows of networks in Figure 1. Split decomposition networks are useful for analysis of small data sets, but have two disadvantages in general. First, for large data sets, they tend to be very unresolved (cf. network N3 in Fig. 1, see also Winkworth et al. 2005). Second, split decomposition may yield networks where edges cross (cf. network N2 in Fig. 1), which can make it difficult to produce a layout for these networks that can easily be interpreted. NeighborNet overcomes both of these issues as it can generate quite resolved networks even for much larger data sets (see, e.g., Beiko 2011), and it is guaranteed to produce a network that is *planar*, that is, that can be drawn without crossing edges. Even so, NeighborNet networks are constrained to be *outer-labeled*, that is, all labels must lie on the outside of the network. This may lead to situations where potentially useful information can be lost (the split represented by the bold edges in network N1 in Fig. 1, for example, is is not displayed by network N4).

In this paper, we present a new method to infer phylogenetic networks. Our new method, FlatNJ, helps to rectify the difficulties with split decomposition and NeighborNet as it does not force labels to the outside of the network (cf. network N7 in Fig. 1), it avoids crossings between edges as much as possible (cf. network N8 in Fig. 1) and it can yield informative splits even when the number of taxa increases (cf. network N9 in Fig. 1). As with the QNet method (Grünewald

et al. 2007) for generating outer-labeled planar split networks, FlatNJ takes quartet data as input. In addition, FlatNJ employs an agglomerative approach to construct networks similar to that used in Neighbor-Joining (Saitou and Nei 1987) and NeighborNet, that involves repeatedly identifying taxa that are "neighbors".

After presenting the FlatNJ method in the next section, we illustrate its potential applications in the Results section by applying it to three data sets that each involve molecular sequences sampled from within a single species or from closely related species. The first two data sets (whole HIV genome sequences and fluorescent proteins in corals) were selected to shed some light on the potential of FlatNJ to identify potential candidates for statistical inference in the analysis of intra-locus recombination. Note that biologists increasingly study patterns of genetic variation that may be caused by intra-locus recombination and introgression of genes (see e.g. Gonthier and Garbelotto 2011; Bahr and Wilson 2012; Zhao et al. 2013) as well as the potential role of these processes in the adaptation of species to environmental change (see e.g. Becker et al. 2013) and the impact of them on the accuracy of species tree reconstruction methods (see e.g. Lanier and Knowles 2012). The last data set (gall wasps) was chosen to give an impression of the performance of FlatNJ in displaying the biogeographic structure of a somewhat larger collection of non-recombining mitochondrial DNA. We conclude with a discussion, where we also mention some possible future directions.

Methods

In the following, X will always denote a set of $n \geq 4$ taxa, and any split of X that groups a non-empty subset A of X against the remaining taxa $B = X - A$ will be denoted by $A|B$. Note that $A|B$ and $B|A$ both denote the same split.

As mentioned in the Introduction, a split network is a graphical representation of a collection of splits of X. For convenience, we call any such collection Σ a *split system* on X; the collection of all possible splits of X is denoted by $\Sigma(X)$. The length of the edges representing a split S in a split network is proportional to a non-negative real number $\omega(S)$, also called the

weight of S. The pair (Σ, ω), consisting of the split system and its weighting, is called a *weighted split system* on X. We now describe the special kind of split systems underlying FlatNJ networks.

Flat split systems

Flat split systems first appeared in Bryant and Dress (2007) and can be formally defined in several equivalent ways. In the following, we describe them, in a way that is sufficiently general for the purpose of explaining our new method, but omit some technical details that are not of great importance here. The reader interested in these details is referred to the definition of flat split systems given in Spillner et al. (2011) which, for the convenience of the reader, is also briefly explained in the supplementary material (doi:10.5061/dryad.q80f6).

First we recall that any outer-labeled split network, such as a NeighborNet network, represents a *circular split system*. This is a split system Σ on X, for which there exists an ordering $x_1, x_2, \ldots, x_n$ of the n taxa in X such that every split in Σ is of the form $\{x_i, x_{i+1}, \ldots, x_j\}|X - \{x_i, x_{i+1}, \ldots, x_j\}$ for some $1 \leq i \leq j < n$ (cf. Fig. 2a). We can also view the splits in any such split system as arising in the following way: The taxa in X are arranged as points along a circle and a split is represented by a straight line separating the set of points into two non-empty subsets (cf. Fig. 2b). Note that NeighborNet networks are constructed from circular split systems: The fact that NeighborNet networks are outer-labeled is essentially a consequence of the points representing the taxa in X being constrained to lie on a circle.

When we adopt the view that circular split systems arise from taxa arranged on a circle, it is natural to wish to remove this constraint and to arrange the taxa in X arbitrarily in the plane, but to still represent splits by straight lines (cf. Fig. 2c). In particular, a split system Σ of X is called *flat* if X can be arranged in the plane so that every split in Σ can be represented by some straight line (Technically speaking, this is actually an *affine split system*, see, e.g., Spillner et al. 2011, but for simplicity we shall just call such split systems flat in this paper.). To simplify the subsequent description, we shall assume that no three points in X lie on a common straight line. This will not, however, restrict the split systems that can be obtained.

Note that the freedom of being able to place the taxa in X anywhere in the plane is the reason why we can have interior labels in the corresponding split networks. Also note that, as we represent splits by straight lines, it can be shown (just as with any circular split system on X) that any flat split system Σ of X contains at most $\binom{n}{2}$ splits (Spillner et al. 2011). If Σ contains precisely $\binom{n}{2}$ splits, we call it *full*. In addition, if Σ is weighted we require that $\omega(S) > 0$ holds for all $S \in \Sigma$.

Systems of 4-splits

When we developed FlatNJ, we at first considered taking a matrix of pairwise distances as input, as with the NeighborNet method. However, this has the disadvantage that there can be more than one flat split system representing such a matrix (see, e.g., Fig. 3a-c). Intuitively, the problem is that pairwise distances cannot distinguish between the two fundamentally different geometric configurations of four points in the plane (cf. Fig. 3d and e): Either none of the points is inside the triangle formed by the other three, or precisely one of the points is inside the triangle formed by the other three. To discriminate between these two configurations, we decided to consider quartet-like input data like that used for the QNet method (Grünewald et al. 2007). In particular, using a link between flat split systems and the theory of oriented matroids (Bryant and Dress 2007), it can be formally shown that four-element subsets are sufficient to discriminate a pair of full flat split systems.

We now present some definitions that are necessary for us to describe our method. For any four distinct elements a, b, c and d in X, a 4-*split* is either of the form $\{a, b\}|\{c, d\}$ or of the form $\{a\}|\{b, c, d\}$. As with splits of the whole taxon set, $\{a, b\}|\{c, d\}$ and $\{c, d\}|\{a, b\}$ (and, similarly, $\{a\}|\{b, c, d\}$ and $\{b, c, d\}|\{a\}$) denote the same 4-split. Note that 4-splits that group two taxa versus two other taxa are usually referred to as *quartets*. Thus, 4-splits can be viewed as a straight-forward generalization of quartets where also groupings of one taxon versus three other taxa are considered. Also note that there are precisely seven distinct 4-splits for any set of four taxa. In the following, we denote the collection of all possible 4-splits that can be formed from the

taxa in X by $\mathcal{F} = \mathcal{F}(X)$ and we will usually also consider a weighting λ that assigns to every 4-split in $\mathcal{F}$ a non-negative real number. The pair $(\mathcal{F}, \lambda)$ will then be referred to as a *(weighted) system of* 4-splits and our method takes such a system as its input.

Note that, unlike the QNet method, FlatNJ also assigns weights to the *trivial splits* (i.e., splits that separate one taxon from all of the rest) in the resulting flat split system. These splits correspond to "pendant" edges in the final split network. In our first experiments we found that the split systems we generated from systems of 4-splits tended to be almost circular. On investigating this phenomenon, we realized that this was probably due to the fact that any flat split system that contains all of the possible trivial splits must in fact be circular. Moreover, the presence of many 4-splits of the form $\{x\}|\{a, b, c\}$ with large weights in the input will naturally lead to flat split systems that contain the trivial split $\{x\}|X - \{x\}$, thus blocking the option of x being placed inside the resulting split network. For this reason, given a system of 4-splits $(\mathcal{F}, \lambda)$, we first compute the quantity

$$\beta(x) = \min_{\{x\}|\{a,b,c\}\in\mathcal{F}} \lambda(\{x\}|\{a, b, c\}) \tag{1}$$

for every x in X, that is, the smallest weight over all 4-splits of the form $\{x\}|\{a, b, c\}$ in $\mathcal{F}$. Then we adjust λ by subtracting $\beta(x)$ from the weight of every 4-split of this form because this amount of weight will definitely be represented in the resulting split network independently of whether x is placed inside the network or not (see the section below describing the final step of our method for more details on how this is achieved).

Generating systems of 4-splits

We now present two possible methods to generate systems of 4-splits: the first produces such systems from multiple sequence alignments using statistical geometry (Eigen et al. 1988), and the second directly from distances between points in the plane.

For the first method, let $\mathcal{A}$ be a sequence alphabet, and let $\mathcal{D}$ denote a measure of pairwise dissimilarity between the letters in $\mathcal{A}$. Here we use $\mathcal{D}(L, L) = 0$ and $\mathcal{D}(L, L') = 1$ for any

two distinct letters L and L' in $\mathcal{A}$ (see, e.g., Nieselt-Struwe and von Haeseler 2001). Then, for a multiple sequence alignment with ℓ columns $c_1, c_2, \ldots, c_\ell$, each column c_i, $1 \leq i \leq \ell$, yields a distance matrix D_i on X by putting $D_i(x, x') = \mathcal{D}(L, L')$, where L and L' are the letters in column c_i in the sequence corresponding to taxon x and taxon x', respectively. To obtain a weight for each 4-split in $\mathcal{F}$, we put

$$\lambda(\{a,b\}|\{c,d\}) = \frac{1}{\ell} \sum_{1 \leqslant i \leqslant \ell} \frac{1}{2} \big(\max \left\{ \begin{smallmatrix} D_i(a,c)+D_i(b,d) \\ D_i(a,d)+D_i(b,c) \\ D_i(a,b)+D_i(c,d) \end{smallmatrix} \right\} - D_i(a,b) - D_i(c,d) \big) \text{ and}$$

$$\lambda(\{a\}|\{b,c,d\}) = \frac{1}{\ell} \sum_{1 \leqslant i \leqslant \ell} \frac{1}{2} \min \left\{ \begin{smallmatrix} \max\{D_i(a,b)+D_i(a,c)-D_i(b,c),0\} \\ \max\{D_i(a,c)+D_i(a,d)-D_i(c,d),0\} \\ \max\{D_i(a,b)+D_i(a,d)-D_i(b,d),0\} \end{smallmatrix} \right\}$$

for any four distinct taxa a, b, c and d in X. Note that the i-th summand in both formulae corresponds to the so-called *isolation index* (Bandelt and Dress 1992a) of the 4-split with respect to the distance matrix D_i.

We also developed a second method for generating systems of 4-splits from distances between points in the plane (coming from, e.g., geographical coordinates for sampling locations of taxa) since we are also interested in the possibility of incorporating such information into our analyses. Recall that, for any four distinct taxa a, b, c and d, there are essentially two different ways in which the corresponding taxa locations can be arranged (cf. Fig. 3d and e). In each case, only six 4-splits (out of the seven possible 4-splits) are suggested by the relative position of the locations and these are exactly those 4-splits represented in the corresponding split network in Figure 3b and c, respectively.

To assign weights to the 4-splits, we apply the formula in Moulton and Spillner (2012, Thm. 3) to the Euclidean distances D^E between the locations. This formula will yield the unique weights for the 4-splits such that the shortest path lengths in the corresponding split network equal the given Euclidean distances. In particular, if the four taxa are arranged as in Figure 3d this is equivalent to weighting each 4-split by its isolation index with respect to D^E as given above (which immediately implies $\lambda(\{a,c\}|\{b,d\}) = 0$). Otherwise, if the four taxa are arranged

as in Figure 3e, we put $\lambda(\{d\}|\{a,b,c\}) = 0$ and set

$$\lambda(\{a\}|\{b,c,d\}) = \frac{1}{2}(D^E(a,b) + D^E(a,c) - D^E(b,d) - D^E(c,d)),$$
$$\lambda(\{b\}|\{a,c,d\}) = \frac{1}{2}(D^E(a,b) + D^E(b,c) - D^E(a,d) - D^E(c,d)),$$
$$\lambda(\{c\}|\{a,b,d\}) = \frac{1}{2}(D^E(a,c) + D^E(c,b) - D^E(a,d) - D^E(b,d)),$$
$$\lambda(\{a,b\}|\{c,d\}) = \frac{1}{2}(D^E(a,d) + D^E(b,d) - D^E(a,b)),$$
$$\lambda(\{a,c\}|\{b,d\}) = \frac{1}{2}(D^E(a,d) + D^E(c,d) - D^E(a,c)), \text{ and}$$
$$\lambda(\{a,d\}|\{b,c\}) = \frac{1}{2}(D^E(b,d) + D^E(c,d) - D^E(b,c)).$$

It is easy to verify that these weights will always be non-negative.

Neighbors in flat split systems

FlatNJ constructs a flat split system from a system of 4-splits using an agglomerative approach similar to the ones used in Neighbor-Joining and NeighborNet. One of the key steps in both of these previous approaches is the selection of "neighbors". As mentioned above, the splits displayed in the networks produced by NeighborNet can be represented by arranging points on a circle (cf. Fig. 2b). Two distinct taxa x and x' are considered to be neighbors if they correspond to consecutive points along the circle (e.g., b and c are neighbors in Fig. 4a). Note, however, that this is equivalent to the following condition (cf. Fig. 4a and b):

(Nb) The straight line segment with end points x and x' does not intersect any of the straight lines through any pair of distinct elements in $X - \{x, x'\}$.

The advantage of condition (Nb) is that it can readily be applied to any set of points in the plane not necessarily arranged around a circle (cf. Fig. 4c and d). More precisely, given a flat split system Σ, two taxa x and x' in X are *neighbors relative to* Σ if there exists some arrangement of X in the plane so that every split in Σ can be represented by a straight line and also x and x' satisfy condition (Nb). Note that there exist flat split systems for which no pair of taxa form neighbors (cf. Fig. 4e). Such split systems will not be generated by FlatNJ.

The overall approach

Given a system of 4-splits $(\mathcal{F}, \lambda)$, FlatNJ essentially works in the following four stages, in a similar way to the NeighborNet method. (i) A pair of neighbors x and x' in X is selected. (ii) The neighbors x and x' are removed from X and replaced by a new element z representing both x and x' (i.e., x and x' are *agglomerated* into a new element z). The system of 4-splits $(\mathcal{F}, \lambda)$ is then updated to give a new system on $X' = (X - \{x, x'\}) \cup \{z\}$. This selection and agglomeration procedure is repeated until only four elements remain. (iii) The whole agglomeration process is reversed to create a full flat split system Σ. (iv) The split weights are estimated for Σ relative to $(\mathcal{F}, \lambda)$, and a corresponding planar split network is then drawn. We describe each of (i)–(iv) in more detail in the following four sections.

Choosing neighbors

As with Neighbor-Joining and NeighborNet, we choose neighbors by assigning scores to pairs of elements in X. In particular, we use two scoring functions that have been chosen to ensure that the algorithm is "consistent" (see below). The first function is based on the following observation. Let (Σ, ω) be a weighted flat split system and x and x' be two taxa that are neighbors in Σ. Then, for any two distinct taxa y and y' in $X - \{x, x'\}$, for at least one of the 4-splits $\{x\}|\{x', y, y'\}$, $\{x, y\}|\{x', y'\}$, $\{x, y'\}|\{x', y\}$ and $\{x, y, y'\}|\{x'\}$ that separate x and x' the sum of the weights of all the splits in Σ that extend the 4-split is 0 (a split $S = A|B$ of X *extends* a 4-split $\{a, b\}|\{c, d\}$ if either $\{a, b\} \subseteq A$ and $\{c, d\} \subseteq B$ or $\{a, b\} \subseteq B$ and $\{c, d\} \subseteq A$; the extension of a 4-split of the form $\{a\}|\{b, c, d\}$ is defined in the same way). For example, for the full flat split system Σ on the set $X = \{a, b, c, d, e\}$ represented by the arrangement in Figure 4c, the taxa c and d are neighbors and there is no split in Σ that extends the 4-split $\{c\}|\{a, b, d\}$.

This suggests defining the following score for any pair x and x' of taxa in X:

$$\sigma_{min}(x, x') = \sum_{\substack{y, y' \in X - \{x, x'\} \\ y \neq y'}} \min \begin{Bmatrix} \lambda(\{x\}|\{x', y, y'\}) \\ \lambda(\{x'\}|\{x, y, y'\}) \\ \lambda(\{x, y\}|\{x', y'\}) \\ \lambda(\{x, y'\}|\{x', y\}) \end{Bmatrix}.$$

Intuitively, the score $\sigma_{min}(x, x')$ captures the total amount of 4-split weight, over all 4-splits in $\mathcal{F}$, that will definitely *not* be represented in the resulting flat split system if we make x and x' neighbors. Hence, good candidates for neighbors are taxa x and x' for which $\sigma_{min}(x, x')$ is minimized.

Once we have found the pairs that minimize the score $\sigma_{min}(x, x')$, we employ a second scoring function that aims to capture the total amount of 4-split weight, over all 4-splits in $\mathcal{F}$ of the form $\{x, x'\}|\{y, y'\}$, that *will* be represented in the resulting flat split system if we make x and x' neighbors. More formally, we put

$$\sigma_{max}(x, x') = \sum_{\substack{y,y' \in X - \{x,x'\} \\ y \neq y'}} \lambda(\{x, x'\}|\{y, y'\}).$$

Note that this function is also used in the selection of neighbors in the QNet algorithm (Grünewald et al. 2007).

Hence, in summary, we choose neighbors by first computing all pairs $\{x, x'\}$ that minimize $\sigma_{min}(x, x')$ and then, out of these pairs, selecting some pair $\{x, x'\}$ that maximizes $\sigma_{max}(x, x')$.

Agglomeration

We now explain how to update the system of 4-splits $(\mathcal{F}, \lambda)$ on X to form one on the set $X' = (X - \{x, x'\}) \cup \{z\}$ once a pair of neighbors x and x' has been selected in X. First, all those 4-splits that involve neither x nor x' remain the same in the updated system of 4-splits on X'. Otherwise, let a, b, c be any three distinct elements in $X - \{x, x'\}$ and put $Y_x = \{x, a, b, c\}$ and $Y_{x'} = \{x', a, b, c\}$. Then the 4-splits involving precisely the four taxa in $\{z, a, b, c\}$ are assigned the

average of the weights of the corresponding 4-splits of Y_x and $Y_{x'}$, that is, we put:

$$\lambda(\{z\}|\{a,b,c\}) = \frac{1}{2}\big(\lambda(\{x\}|\{a,b,c\}) + \lambda(\{x'\}|\{a,b,c\})\big),$$
$$\lambda(\{a\}|\{z,b,c\}) = \frac{1}{2}(\lambda(\{a\}|\{x,b,c\}) + \lambda(\{a\}|\{x',b,c\})),$$
$$\lambda(\{b\}|\{z,a,c\}) = \frac{1}{2}(\lambda(\{b\}|\{x,a,c\}) + \lambda(\{b\}|\{x',a,c\})),$$
$$\lambda(\{c\}|\{z,a,b\}) = \frac{1}{2}(\lambda(\{c\}|\{x,a,b\}) + \lambda(\{c\}|\{x',a,b\})),$$
$$\lambda(\{z,a\}|\{b,c\}) = \frac{1}{2}(\lambda(\{x,a\}|\{b,c\}) + \lambda(\{x',a\}|\{b,c\})),$$
$$\lambda(\{z,b\}|\{a,c\}) = \frac{1}{2}(\lambda(\{x,b\}|\{a,c\}) + \lambda(\{x',b\}|\{a,c\})), \text{ and}$$
$$\lambda(\{z,c\}|\{a,b\}) = \frac{1}{2}(\lambda(\{x,c\}|\{a,b\}) + \lambda(\{x',c\}|\{a,b\})).$$

Reversing the agglomeration process

Once all possible agglomerations have been performed, a set with four elements, which we denote by X^*, and a system of 4-splits $(\mathcal{F}^*, \lambda^*)$ on X^* is left. Note that, since X^* contains precisely four taxa, every split of X^* can be viewed as a 4-split. Moreover, note that $\Sigma(X^*)$ is not a flat split system since it contains seven splits. Therefore, to obtain a full flat split system on X^* (which must contain precisely $\binom{4}{2} = 6$ splits), we need to select one split in $\Sigma(X^*)$ that will be removed. Following again the idea that we want to minimize the amount of 4-split weight not represented in the output, we choose a split $S \in \Sigma(X^*)$ that, when viewed as a 4-split, is assigned minimum weight by λ^* over all splits of X^* and put $\Sigma^* = \Sigma(X^*) - \{S\}$. Put differently, we choose Σ^* since it covers as much of the weight as possible of the 4-splits in $\mathcal{F}^*$. In addition, we construct an arrangement of X^* in the plane such that all the splits in Σ^* are represented by straight lines through this arrangement.

Next, starting with X^* we reverse the agglomerations one by one. For simplicity, we only describe how this is done for the last reversal that replaces z in X' by x and x' to obtain the set X since the other reversals are performed in a completely analogous way. To this end, assume

that we have a full flat split system Σ' on X' arranged in the plane. From this we want to find a suitable arrangement of X in the plane that corresponds to a full flat split system Σ on X (see Fig. 5a). In particular, the arrangement of X is obtained by replacing the point representing z in X' by two points representing x and x', respectively (cf. Fig. 5b). These two points are placed in such a way that, for each split $S = A|B \in \Sigma'$ with $z \in A$, we have the split $(A - \{z\}) \cup \{x, x'\}|B \in \Sigma$. This is achieved by placing x and x' close enough to the original position of z. In the situation depicted in Figure 5b it suffices, for example, to place x and x' inside the shaded region.

In addition to those splits that arise from the splits in Σ', the split system Σ also contains $n-1$ splits that separate x and x'. The splits of this type that are contained in Σ depend on the position of x' relative to x. Note that there is some freedom in choosing the precise coordinates of x and x'. Topologically, there are, however, only $2(n-2)$ different configurations that can be described as follows. We place a suitably small disk centered at the original position of z (cf. Fig. 5c). At the center of this disk we place x. Then we partition the disk into $2(n-2)$ sectors by drawing straight lines that contain x and any of the points in $X' - \{z\}$. For each of these sectors, placing x' anywhere within that sector yields the same flat split system on X, and placing x' in a different sector yields a different flat split system (cf. Fig. 5d and e). Let $\mathcal{C}$ denote the resulting collection of $2(n-2)$ different full flat split systems.

We now use the input system of 4-splits $(\mathcal{F}, \lambda)$ again to select one of the flat split systems in $\mathcal{C}$. More specifically, we select some Σ in $\mathcal{C}$ for which

$$\sum_{\substack{y,y' \in X - \{x,x'\} \\ y \neq y'}} \quad \sum_{\substack{S' \text{ a 4-split of } \{x,x',y,y'\} \\ \text{and some } S \text{ in } \Sigma \text{ extends } S'}} \lambda(S') \tag{2}$$

is maximum. In other words, we consider all 4-splits in $\mathcal{F}$ that involve both x and x' and select a split system Σ for which the total weight of those 4-splits that are extended by some split in Σ is maximum. Note that there can be more than one Σ in $\mathcal{C}$ that maximizes (2). In this case we select, among those maximizing (2), one for which Σ contains the two trivial splits $\{x\}|X - \{x\}$

and $\{x'\}|X - \{x'\}$, if such a Σ exists, and an arbitrary one otherwise. This ensures that if there is a simpler way to accommodate the input data (i.e., a phylogenetic tree or a circular split system) then we choose this.

Weighting and drawing

Once we have computed a full flat split system Σ on X whose structure reflects that of the input system of 4-splits $(\mathcal{F}, \lambda)$, it only remains to compute non-negative weights for the splits in Σ. To do this, we use an approach similar to the one used in QNet. More specifically, split weights ω are computed so that the system of 4-splits $(\mathcal{F}, \lambda_{(\Sigma,\omega)})$ on X (which is defined by setting, for every 4-split $S' \in \mathcal{F}$, $\lambda_{(\Sigma,\omega)}(S')$ to be the total weight of those splits S of X that extend S') is as close as possible to the input system of 4-splits $(\mathcal{F}, \lambda)$ in the least squares sense, that is, we minimize

$$\sum_{S' \in \mathcal{F}} \big(\lambda(S') - \lambda_{(\Sigma,\omega)}(S')\big)^2. \tag{3}$$

To minimize this last expression we solve a quadratic program (see, e.g., Lawson and Hanson 1974). The user can then filter the resulting weighted flat split system (Σ, ω) if desired using the method described in Grünewald et al. (2007) to suppress splits with very low weights. In particular, the user provides a real-valued threshold t, $0 \leq t \leq 1$, which suppresses any split S in Σ for which there exists some other split S' in Σ such that S and S' are incompatible and the weight of S is less than a fraction t of the weight of S'.

The resulting weighted flat split system (Σ, ω) is represented by a planar split network $\mathcal{N}$, which is drawn using the algorithm presented in Spillner et al. (2011). It can then be displayed using the SplitsTree package (Huson and Bryant 2006). Note that the filtering mentioned above can help ease interpretation of this network by reducing the number of small boxes that appear in it. At this stage, the values $\beta(x)$, defined in (1) for all x in X, are also taken into account as follows. If $\mathcal{N}$ already contains a pendant edge representing the trivial split $\{x\}|X - \{x\}$ then the length of this pendant edge is just increased by $\beta(x)$. Otherwise (i.e., $\mathcal{N}$ does not contain a pendant edge representing the trivial split $\{x\}|X - \{x\}$ and $\beta(x) > 0$), a new pendant edge of

length $\beta(x)$ is added to $\mathcal{N}$. Note that this last step can potentially produce pendant edges that must cross some other edges in the planar network $\mathcal{N}$.

Implementation of FlatNJ

We have implemented FlatNJ in Java. For analyzing the examples below, we ran the program on a PC with Intel i5-2300(4) CPU, with 6 GB of main memory and with the operating system Ubuntu 12.04. The run time of our implementation is superpolynomial in the worst case due to the fact that the computation of the weights for the splits involves solving a quadratic program (for this we use algorithms in the Gurobi Optimizer, version 5.0, `www.gurobi.com`), although in practice we have not found this to be a limitation for sets of up to 100 taxa. Note that the entire agglomeration process and its reversal can be done in polynomial time. More specifically, in our implementation we take $O(n^4)$ time, which is optimal since the input consists of $7 \cdot \binom{n}{4}$ 4-splits on the set X.

Consistency of FlatNJ

An important property that any method for constructing a split network should ideally satisfy is *consistency*. This means that if the method is designed to produce a split system with a certain special property (e.g., compatible or circular), then if such a split system (or associated data) is taken as input, the same split system should result. For example, if a compatible/circular weighted split system corresponding to a phylogenetic tree/outer-labeled planar network is taken as input to Neighbor-Joining/NeighborNet, then it can be shown that the split system will be reproduced (Atteson 1999; Bryant et al. 2007).

By construction, FlatNJ always generates a flat split system Σ on X with the following special recursive property: Σ contains at least one pair of taxa that are neighbors, and if any pair of neighbors in Σ is agglomerated then a new flat split system results that has at least one pair of neighbors and that has the same property. We call such flat split systems *neighborly* (note that there are flat split systems that are not neighborly). If (Σ, ω) is a weighted flat split system, and

FlatNJ is given $(\mathcal{F}, \lambda_{(\Sigma,\omega)})$ as input system of 4-splits, then it can be shown that it will reproduce (Σ, ω) if any of the following hold: (a) Σ is compatible, (b) Σ is circular, or (c) (Σ, ω) is a neighborly, full flat split system. Note that both of the scoring functions σ_{min} and σ_{max} are necessary to achieve consistency of FlatNJ in (a)–(c). In particular, when used on its own, the scoring function σ_{min} can fail to select neighbors even in circular split systems. Similarly, even though σ_{max} will always select neighbors in circular split systems, used alone it can fail to select neighbors in neighborly flat split systems (see supplementary material for more details).

In general, although we have found that there are many non-full, neighborly flat split systems for which FlatNJ is consistent, there are also such split systems that FlatNJ cannot reproduce. Ideally, we would like to give a complete and concise description of those flat split systems for which FlatNJ is consistent. However, we expect that there might not be one since such a description would probably pave the way for a polynomial time algorithm to decide whether or not an arbitrary split system is flat, a problem that we strongly suspect to be NP-complete.

Results

We now illustrate some potential uses of FlatNJ by presenting its application to data sets involving recombination, ancestral sequences and biogeographical features. In each of these cases we shall see how labeling the inside of a network can be useful for understanding the specific structure of the data.

A circulating recombinant form of HIV

The first data set involves the study of recombination in viruses, for which split networks have been commonly used. In this example, we applied FlatNJ to analyze the circulating recombinant form *CRF49* of HIV reported in de Silva et al. (2010). We aligned the three whole genome sequences representing *CRF49* published in de Silva et al. (2010) together with reference sequences for the collection $Sub = \{A, B, C, D, F, G, H, J, K\}$ of known subtypes of HIV (see

supplementary material for details). In Figure 6 we present the networks produced by NeighborNet and FlatNJ for this data set.

It was found in de Silva et al. (2010) that *CRF49* is composed of the known subtypes A (23% of total sequence length), C (18%), J (48%), K (5%) and, in addition, also contains a region (6%) that could not be assigned to any of the known subtypes. This composition is reflected by the fact that both the NeighborNet and FlatNJ networks contain the splits $S_J = \{CRF49, J\}|Sub - \{J\}$ and $S_C = \{CRF49, C\}|Sub - \{C\}$ (the split decomposition network, included in the supplementary material, only contains the split S_J). Moreover, the weight assigned to these splits in both networks is quite similar to the relative contribution of subtypes J and C to *CRF49*.

Note that the FlatNJ network also contains the split $S_{A,G} = \{CRF49, A, G\}|Sub - \{A, G\}$, which indicates that subtypes A and/or G could have contributed to *CRF49*. According to de Silva et al. (2010), subtype A contributed to *CRF49*, but this cannot be easily deduced from the NeighborNet network. In fact, it is impossible to display the three splits S_J, S_C and $S_{A,G}$ together in *any* outer-labeled split network. Hence, the FlatNJ network provides a more complete visualization of the composition of *CRF49* inferred in de Silva et al. (2010).

Ancestral forms of fluorescent proteins

We now consider a data set presented in Ugalde et al. (2004) to investigate the evolution of fluorescent proteins in corals. This data set consists of previously published proteins and reconstructed ancestral sequences presented in Ugalde et al. (2004). Here we focus on those groups of proteins for which an ancestral sequence was presented in Ugalde et al. (2004): *Red* = {*Kaede*,*mc1*,*R1_2*}, *pre-Red* = {*G1_2*}∪*Red*, *Red/Green* = {*R2*,*mc2*,*mc3*,*mc4*}∪*pre-Red* and *ALL* = {*G5_2*,*mc5*}∪*Red/Green*. The sequences were aligned (see supplementary material for details) and the networks produced by NeighborNet and FlatNJ are depicted in Figure 7. We use the same labels as in Ugalde et al. (2004) and the names of the groups above are used to indicate the corresponding ancestral sequences.

As can be seen, both networks group sequences emitting the same color (red, green or cyan) together. However, the networks also contain many pairs of incompatible splits suggesting a complex, non-treelike evolution of fluorescent proteins in corals. This is in agreement with the findings in Kelmanson and Matz (2003), suggesting that intra-locus recombination could be one of the mechanisms that produced the sequence diversity we see today. This data set illustrates that it could be useful to allow internal labels when ancestral sequences are present. Indeed, in contrast to the NeighborNet, FlatNJ places all four ancestral sequences inside the network. Moreover, their placement relative to one another also better reflects the groups of proteins given above.

Biogeography of gall wasps

In our final example we consider a data set of 80 mitochondrial DNA sequences (see supplementary material for details) sampled from individuals of the species *A. kollari* (oak gall wasp) for which geographic coordinates for the sampling locations corresponding to each sequence are known (see Fig. 8). This data set was also used in Spillner et al. (2011) to illustrate how, in a somewhat ad-hoc fashion, flat split systems can also be generated using multi-dimensional scaling.

A. kollari is native to regions at the latitude of the Mediterranean from Portugal to Iran. Stone et al. (2001, 2007) studied the colonization of Northern Europe, in particular the British Isles, by this species and concluded that the data suggest that a large number of individuals of *A. kollari* that came originally from the Eastern Mediterranean were introduced to Britain by human trade. One step taken to reach this conclusion was the generation of a NeighborNet network for the sequences, which suggested that a tree-based analysis was not sufficient to fully assess the data.

The networks produced by NeighborNet and FlatNJ from the sequence alignment are presented in Figure 9. Overall, the networks are quite similar with 45% of the total weight of all splits in the FlatNJ network corresponding to splits that are also represented in the NeighborNet. This is somewhat reassuring as we feel that it is desirable for FlatNJ to not behave too differently from the well-established NeighborNet method, at least for data that are planar in nature. In

contrast, the network produced by the split decomposition method (included in the supplementary material) is again much less resolved.

We next explored a way to visualize the relationship between the geographic and genetic data using split networks. More specifically, we generated the flat split system Σ_{geo} from Euclidean distances between the sampling locations and, to investigate which of the splits in Σ_{geo} are supported by the genetic distances, we reassigned weights to the splits in Σ_{geo} by minimizing the objective function (3) for the 4-split weights obtained from the sequence alignment. The split network representing the resulting weighted flat split system is depicted in Figure 10. The network displays a clear-cut geographic structure, although it is quite different from the FlatNJ network in Figure 9. Even so, the split highlighted in bold is present (up to sequence (80)) in both networks, which might represent a signal for a geographical divide between the sequences from Iberia and South-Western France and the other sequences. Note that such a major divide has not only been observed for *A. kollari* but also for other species from the genus *Andricus* (cf. Stone et al. 2007).

Discussion

We have introduced and implemented a new method called FlatNJ for generating split networks. As with NeighborNet and QNet, the method generates planar split networks. Unlike QNet, FlatNJ permits the estimation of pendant edge lengths, and, in contrast to both NeighborNet and QNet, FlatNJ allows internal vertices in the network to be labeled. Note that, although split decomposition also allows internal labels, it does not necessarily produce a planar network.

We emphasize that FlatNJ does *not* force the data to be represented using many boxes or internal labels: If the data are perfectly treelike then FlatNJ will return a tree and this will be the same tree that Neighbor-Joining, split decomposition and NeighborNet will all necessarily return. More generally, if the data are perfectly represented by some weighted circular split system then FlatNJ will return this circular split system including the weights and, thus, agree with the output of NeighborNet. It is only when encountering data that are neither treelike nor circular that

FlatNJ provides the *option* to produce a network that is not a tree or an outer-labeled network. Thus, FlatNJ offers an excellent opportunity for exploratory data analysis (Morrison 2010).

To illustrate some of FlatNJ's potential uses, we applied it to three data sets. In the analysis of recombination the added value of having labels inside the network is mainly the flexibility gained by representing collections of splits that cannot be displayed with outer-labeled networks. We also saw that ancestral sequences can be naturally placed by FlatNJ in the interior of the network, which not only helps avoid unnecessary distortion in the representation, but might potentially help to identify candidate ancestral sequences in situations where these are not known. In the last data set geographic considerations were of interest and we demonstrated that FlatNJ could also be useful for analyzing and visualizing such data. As a further potential application, it should also be noted that the taxon selection problem described in Minh et al. (2009) can be solved efficiently for the weighted split systems produced by FlatNJ (using, e.g., the algorithm presented in Spillner et al. (2008)).

Even though we have found that FlatNJ is able to visualize more information than NeighborNet this can come at a price: To avoid distortion FlatNJ sometimes uses more pairs of incompatible splits to represent the data than NeighborNet (see, e.g., the networks N5 and N8 in Fig. 1). Moreover, we have found that producing a suitable layout of the labels of interior vertices can be quite challenging, especially for data where large groups of taxa label the inside of the network (see, e.g., Fig. 9). Developing alternative ways to draw the network that address this would be desirable. More generally, although having a planar network can be useful for interpreting data, as noted in Bryant and Moulton (2004), some data sets are intrinsically better represented by "high-dimensional", non-planar networks such as the ones that can be generated using split decomposition. It is therefore still an interesting challenge to develop methods to help effectively construct and visualize such networks.

As with some other quartet-based methods (such as QNet), the applicability of FlatNJ can be somewhat limited by the fact that its input consists of a system of 4-splits, whose size grows with a polynomial of degree 4 in the number of taxa. In particular, compared with

NeighborNet, the split construction phase in FlatNJ and QNet is one order of magnitude slower ($O(n^3)$ vs. $O(n^4)$ for n taxa). Even so, in practice we have found that data sets with up to 100 taxa can usually be processed within a few minutes using the current implementation. Using 4-splits also has consequences for memory usage. As with QNet, with careful manipulation of matrices, FlatNJ is implemented using $O(n^4)$ memory.

A subtle point, that is tightly linked with generating split networks, is the interpretation of the weights assigned to their edges (see, e.g., Levy and Pachter 2011). If we use estimates of pairwise evolutionary distances, for example, then these distances are decomposed according to the type of split system (e.g., circular or flat) underlying the network, just like the edges in a tree decompose the distance between two leaves into a sum of branch lengths. In a similar way, the weights in a quartet or, more generally, a 4-split are decomposed when such input data are used. We designed FlatNJ to be adaptable and modular: For example, the statistical geometry method for estimating the 4-split weights presented above could be replaced by any alternative approach (e.g., likelihood-based as in the Tree-Puzzle method by Schmidt et al. 2002). Moreover, using extensions of probabilistic models from trees to split networks such as those proposed in Bryant (2005), the whole agglomerative approach could, in principle, be replaced by a procedure that estimates a flat split system, including the weights, using a Bayesian or maximum likelihood approach. How this could be done in a practically useful way remains a challenging direction for future research.

In conclusion, FlatNJ is a flexible new method to analyze and visualize data. It should provide a useful complementary approach to methods such as split decomposition, NeighborNet and QNet, and could also be used as the basis for developing new methods to better understand data sets involving specific considerations such as ancestral sequences and biogeographical information.

Supplementary material

An online-only appendix can be found in the Dryad data repository

(doi:10.5061/dryad.q80f6).

Funding

MB was supported through a scholarship funded by the federal state of Mecklenburg-Western Pomerania.

Acknowledgments

We thank David Bryant for many stimulating discussions related to the work presented in this paper. We also thank Lars Jermiin and Andy Anderson for their helpful input, as well as the referees.

*

References

Atteson K. 1999. The performance of the Neighbor-Joining methods of phylogenetic reconstruction. Algorithmica 25:251–278.

Bahr A., Wilson A. 2012. The evolution of MHC diversity: Evidence of intralocus gene conversion and recombination in a single-locus system. Gene 497:52–57.

Bandelt H.-J., Dress A. 1992a. A canonical decomposition theory for metrics on a finite set. Adv. Math. 92:47–105.

Bandelt, H.-J. and A. Dress. 1992b. Split decomposition: a new and useful approach to phylogenetic analysis of distance data. Mol. Phylogenet. Evol. 1:242–252.

Becker M., Gruenheit N., Steel M., Voelckel C., Deutsch O., Heenan P., McLenachan P., Kardailsky O., Leigh J., Lockhart P. 2013. Hybridization may facilitate *in situ* survival of endemic species through periods of climate change. Nat. Clim. Change In press.

Beiko R. 2011. Telling the whole story in a 10,000-genome world. Biol. Direct 6.

Bryant D. 2005. Extending tree models to split networks. Pages 322–334 *in* Algebraic Statistics for Computational Biology (L. Pachter and B. Sturmfels, eds.). Cambridge University Press.

Bryant D., Dress A. 2007. Linearly independent split systems. Eur. J. Combin. 28:1814–1831.

Bryant D., Moulton V. 2004. Neighbor-net: an agglomerative method for the construction of phylogenetic networks. Mol. Biol. Evol. 21:255–265.

Bryant D., Moulton V., Spillner A. 2007. Consistency of the Neighbor-Net algorithm. Algorithms Mol. Biol. 2.

de Silva T., Turner R., Hué S., Trikha R., van Tienen C., Onyango C., Jaye A., Foley B., Whittle H., Rowland-Jones S., Cotten M. 2010. HIV-1 subtype distribution in the Gambia and the significant presence of CRF49_cpx, a novel circulating recombinant form. Retrovirology 7.

Dunn M., Terrill A., Reesink G., Foley R., Levinson S. 2005. Structural phylogenetics and the reconstruction of ancient language history. Science 309:2072–2075.

Eigen M., Winkler-Oswatitsch R., Dress A. 1988. Statistical geometry in sequence space: a method of quantitative comparative sequence analysis. Proc. Natl. Acad. Sci. USA 85:5913–5917.

Gonthier P., Garbelotto M. 2011. Amplified fragment length polymorphism and sequence analyses reveal massive gene introgression from the European fungal pathogen *Heterobasidion annosum* into its introduced congener *H. irregulare*. Mol. Ecol. 20:2756–2770.

Goremykin V., Nikiforova S., Biggs P., Zhong B., Delange P., Martin W., Woetzel S., Atherton R., McLenachan P., Lockhart P. 2013. The evolutionary root of flowering plants. Syst. Biol. 62:50–61.

Grünewald S., Forslund K., Dress A., Moulton V. 2007. QNet: an agglomerative method for the construction of phylogenetic networks from weighted quartets. Mol. Biol. Evol. 24:532–538.

Huson D., Bryant D. 2006. Application of phylogenetic networks in evolutionary studies. Mol. Biol. Evol. 23:254–267.

Huson D., Rupp R., Scornavacca C. 2010. Phylogenetic networks. Cambridge University Press.

Kelmanson I., Matz M. 2003. Molecular basis and evolutionary origins of color diversity in great star coral montastraea cavernosa (scleractinia: Faviida). Mol. Biol. Evol. 20:1125–1133.

Lanier H., Knowles L. 2012. Is recombination a problem for species-tree analyses? Syst. Biol. 61:691–701.

Lawson C., Hanson R. 1974. Solving least squares problems. Prentice Hall.

Levy D., Pachter L. 2011. The neighbor-net algorithm. Adv. Appl. Math. 47:240–258.

Minh B., Klaere S., von Haeseler A. 2009. Taxon selection under split diversity. Syst. Biol. 58:586–594.

Morrison D. 2010. Using data-display networks for exploratory data analysis in phylogenetic studies. Mol. Biol. Evol. 27:1044–1057.

Moulton V., Spillner A. 2012. Optimal algorithms for computing edge weights in planar split networks. J. Appl. Math. Comput. 39:1–13.

Nieselt-Struwe K., von Haeseler A. 2001. Quartet-mapping, a generalization of the likelihood-mapping procedure. Mol. Biol. Evol. 18:1204–1219.

Octavia S., Lan R. 2006. Frequent recombination and low level of clonality within *Salmonella enterica* subspecies I. Microbiology 152:1099–1108.

Pelletier E., Saurin W., Cheynier R., Letvin N., Wain-Hobson S. 1995. The tempo and mode of SIV quasispecies development *in vivo* calls for massive viral replication and clearance. Virology 208:644–652.

Saitou N., Nei M. 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol. Biol. Evol. 4:406–425.

Schmidt H., Strimmer K., Vingron M., von Haeseler A. 2002. TREE-PUZZLE: maximum likelihood phylogenetic analysis using quartets and parallel computing. Bioinformatics 18:502–504.

Spillner A., Nguyen B., Moulton V. 2008. Computing phylogenetic diversity for split systems. IEEE/ACM Trans. Comput. Biol. Bioinf. 5:235–244.

Spillner A., Nguyen B., Moulton V. 2011. Constructing and drawing regular planar split networks. IEEE/ACM Trans. Comput. Biol. Bioinf. 9:395–407.

Stone G., Atkinson R., Rokas A., Csóka G., Nieves-Aldrey J. 2001. Differential success in northwards range expansion between ecotypes of the marble gallwasp *Andricus kollari*: a tale of two lifecycles. Mol. Ecol. 10:761–778.

Stone G., Challis R., Atkinson R., Csóka G., Hayward A., Melika G., Mutun S., Preuss S., Rokas A., Sadeghi E., Schönrogge K. 2007. The phylogeographical clade trade: tracing the impact of human-mediated dispersal on the colonization of northern europe by the oak gallwasp *Andricus kollari*. Mol. Ecol. 16:2768–2781.

The STAR Consortium. 2008. SNP and haplotype mapping for genetic analysis in the rat. Nat. Genet. 40:560–566.

Tugume A., Mukasa S., Kalkkinen N., Valkonen J. 2010. Recombination and selection pressure in the ipomovirus sweet potato mild mottle virus (*Potyviridae*) in wild species and cultivated sweetpotato in the centre of evolution in East Africa. J. Gen. Virol. 91:1092–1108.

Ugalde J., Chang B., Matz M. 2004. Evolution of coral pigments recreated. Science 305:1433.

Wain-Hobson S., Renoux-Elbé C., Vartanian J., Meyerhans A. 2003. Network analysis of human

and simian immunodeficiency virus sequence sets reveals massive recombination resulting in shorter pathways. J. Gen. Virol. 84:885–895.

Winkworth R., Bryant D., Lockhart P., Havell D., Moulton V. 2005. Biogeographic interpretation of split graphs: least squares optimization of edge lengths. Syst. Biol. 54:56–65.

Zhao M., Wang Y., Shen H., Li C., Chen C., Luo Z., Wu H. 2013. Evolution by selection, recombination, and gene duplication in MHC class I genes of two *Rhacophoridae* species. BMC Evol. Biol. 13.

Figure captions

Figure 1: Split networks for SIV sequences (Pelletier et al. 1995). Sequence labels are the same as those used in Wain-Hobson et al. (2003). Networks in a row are generated with the method specified in front of the row.

Figure 2: a) An outer-labeled phylogenetic network $\mathcal{N}$ representing a circular split system. A corresponding circular ordering of the taxa is a, b, c, d, e. b) The taxa are represented by points arranged along a circle respecting the circular ordering. The split $S = \{a, b\}|\{c, d, e\}$ corresponding to the bold edges in $\mathcal{N}$ is drawn as a straight line separating the points on the circle. c) Five points representing taxa. They are not constrained to lie on a circle. The straight line represents the split $S' = \{a, c\}|\{b, d, e\}$.

Figure 3: a) A matrix of pairwise distances on the set of taxa $X = \{a, b, c, d\}$. b), c) Two split networks representing weighted flat split systems in which the shortest path distance perfectly matches the distances given in subfigure a. For clarity, the lengths of the edges are also given as the number next to each edge. d) A configuration of four points in the plane that corresponds to the structure of the flat split system represented in subfigure b: No taxon is inside relative to the other three. e) A configuration of four points in the plane that corresponds to the structure of the flat split system represented in subfigure c: Taxon d is inside relative to the other three.

Figure 4: a) Any two consecutive elements along the circle are considered neighbors. None of the bold gray straight lines intersects the dotted straight line segment whose end points are the neighbors b and c. b) The bold gray straight line through b and e intersects the dotted straight line segment with end points a and c, indicating that a and c are not neighbors. c) Taxa c and d are neighbors because none of the bold gray straight lines intersects the dotted straight line segment with end points c and d. d) Taxa c and e are not neighbors because the bold gray

straight line through a and d intersects the dotted straight line segment with end points c and e. e) An arrangement of the taxa $X = \{a, b, \ldots, f\}$ in the plane for which there is no pair of neighbors relative to the corresponding full flat split system.

Figure 5: a) A full flat split system Σ' on the set $X' = \{a, b, c, z\}$ with z representing two agglomerated elements x and x'. The black straight lines depict the $\binom{4}{2} = 6$ splits in Σ'. b) Replacing z by two points representing x and x'. c) The disk sectors representing the options for placing x' relative to x. d) A placement of x' that yields the four splits $\{x, a\}|\{x', b, c\}$, $\{x, a, b\}|\{x', c\}$, $\{x, a, c\}|\{x', b\}$ and $\{x, c\}|\{x', a, b\}$ separating x and x'. e) An alternative placement of x' that yields again the splits $\{x, a\}|\{x', b, c\}$ and $\{x, a, b\}|\{x', c\}$ but also two different splits, namely, $\{x, b\}|\{x', a, c\}$ and $\{x, b, c\}|\{x', a\}$.

Figure 6: Split networks for three sequences of a circulating recombinant form (*CRF49*) of HIV reported in de Silva et al. (2010) and representatives of HIV subtypes *A*-*K*. The edges that represent the split $S_J = \{CRF49, J\}|Sub - \{J\}$ mentioned in the text are drawn bold.

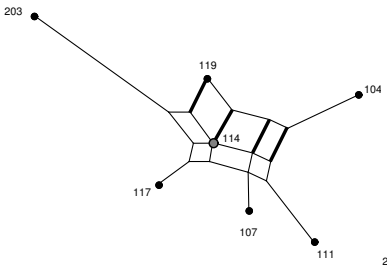
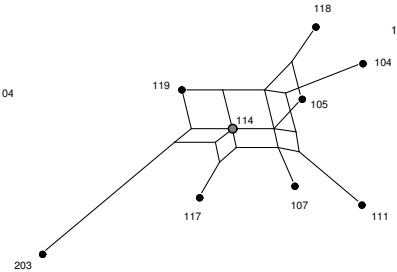
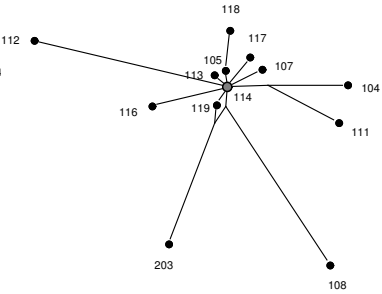
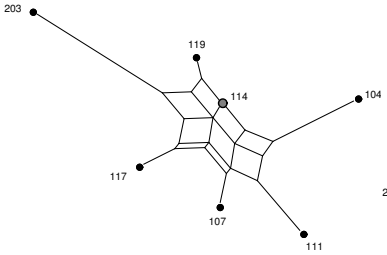
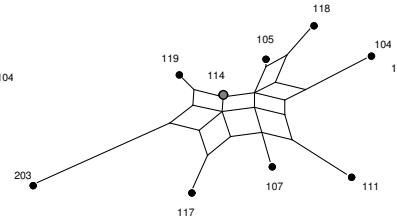
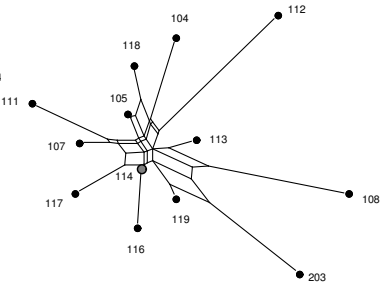
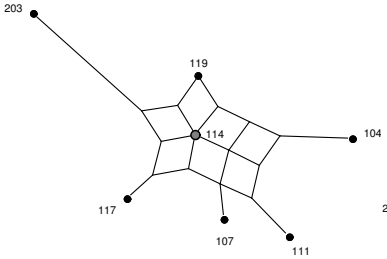
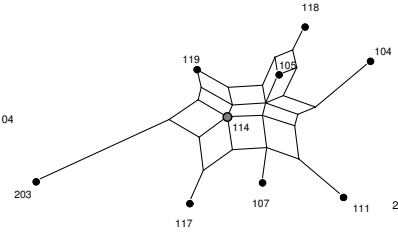
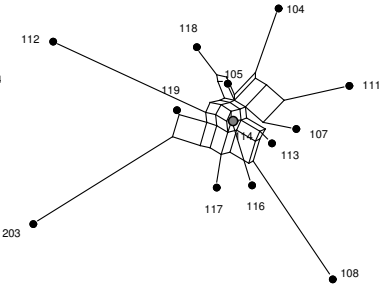
Figure 7: Split networks for fluorescent protein sequences including reconstructed ancestral sequences. The labels correspond to the color emitted by the protein as follows: cyan (*G5_2*, *mc5*), green (*G1_2*, *R2*, *mc2*, *mc3*, *mc4*) and red (*R1_2*, *Kaede*, *mc1*). The labels *Red*, *pre-Red*, *Red/Green* and *ALL* correspond to reconstructed ancestral sequences from Ugalde et al. (2004) for the four groups of proteins mentioned in the text.

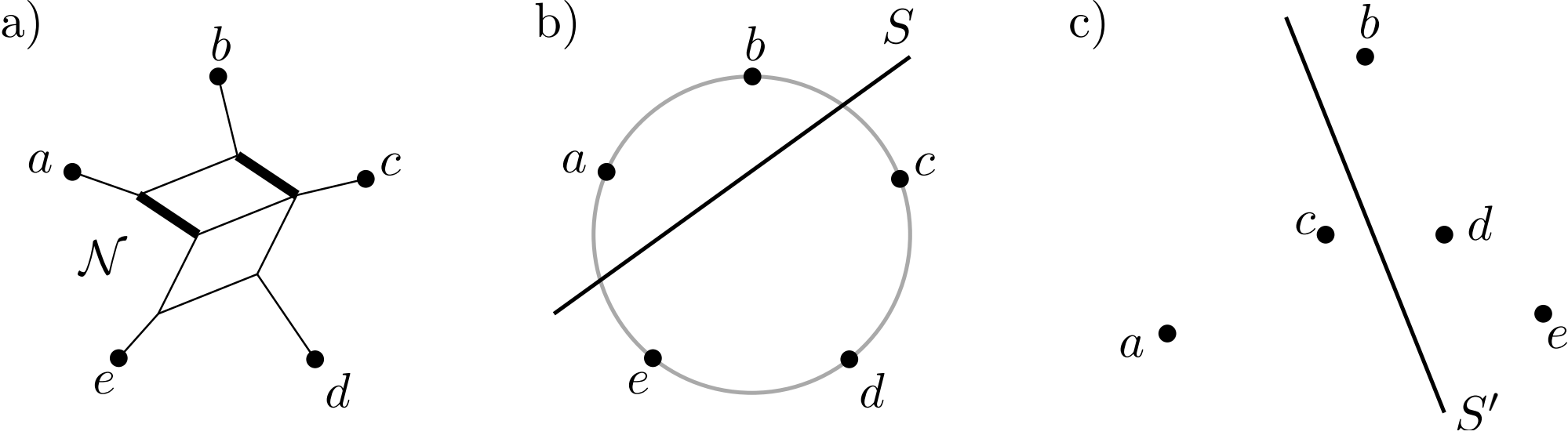
Figure 8: A map with the sampling locations of the sequences in the gall wasp data set. The accession numbers corresponding to the labels used in the map and the networks can be found in

the supplementary material.

Figure 9: Split networks produced from the sequence alignment for the gall wasp data set. The coloring/shading scheme is the same as in Figure 8.

Figure 10: Split network produced by FlatNJ for the gall wasp data set from geographic coordinates with split weights computed using the weighted 4-splits generated from the sequence alignment as described in the text. The split highlighted in bold separates the sequences from Iberia and Southern France from the other sequences. The coloring/shading scheme is the same as in Figure 8.

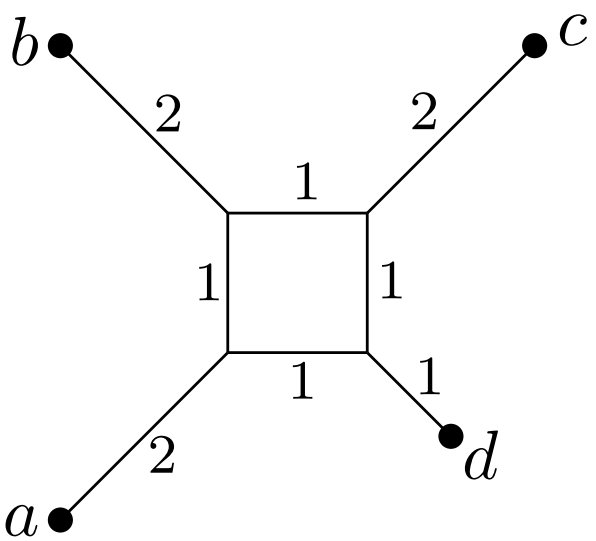
	7 taxa	9 taxa	13 taxa
Split decomposition	**N1** 	**N2** 	**N3** 
NeighborNet	**N4** 	**N5** 	**N6** 
FlatNJ	**N7** 	**N8** 	**N9** 


a)
b
a
c
$\mathcal{N}$
e
d
b)
b
S
a
c
e
d
c)
b
c
d
e
a
S'

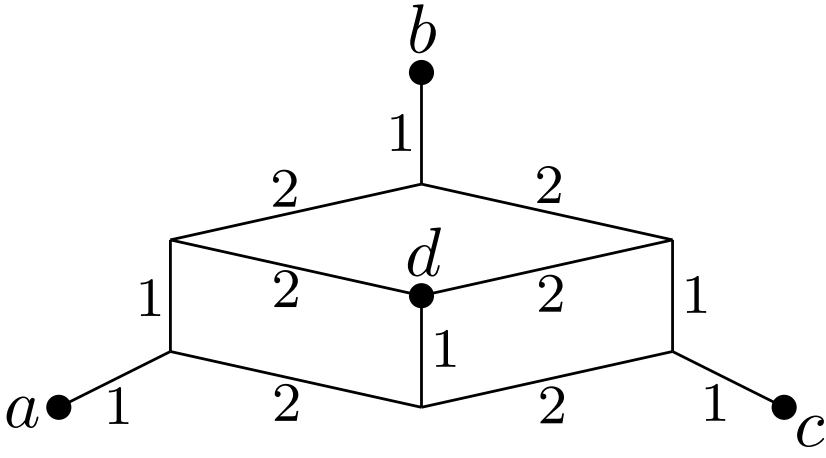
a)

D	a	b	c	d
a	0	5	6	4
b	5	0	5	5
c	6	5	0	4
d	4	5	4	0

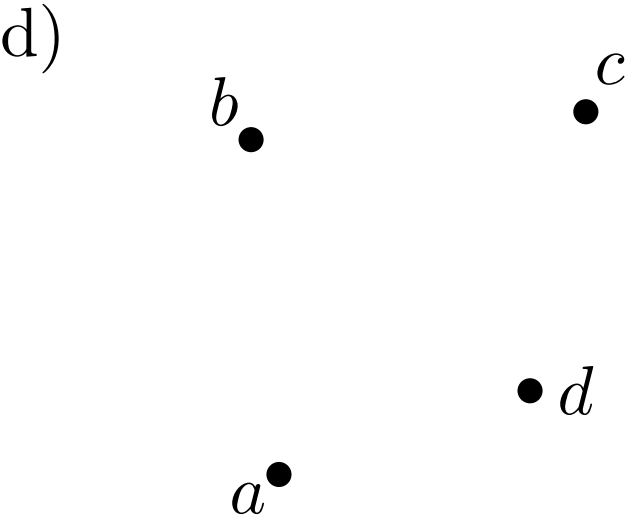
b)



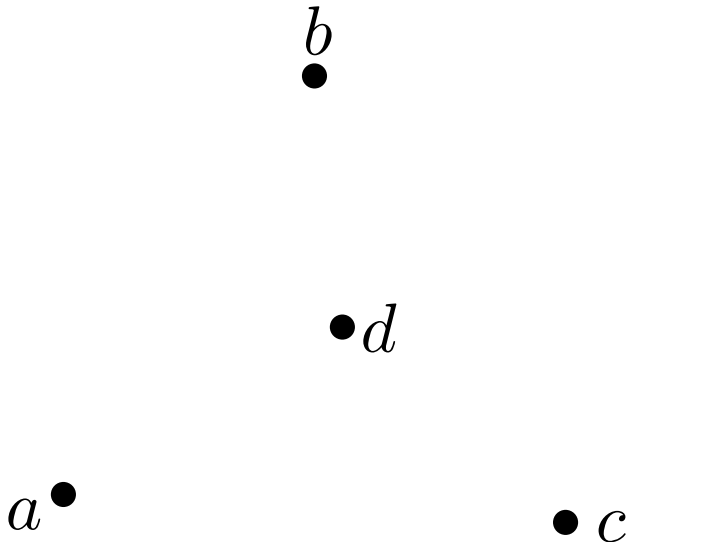
c)

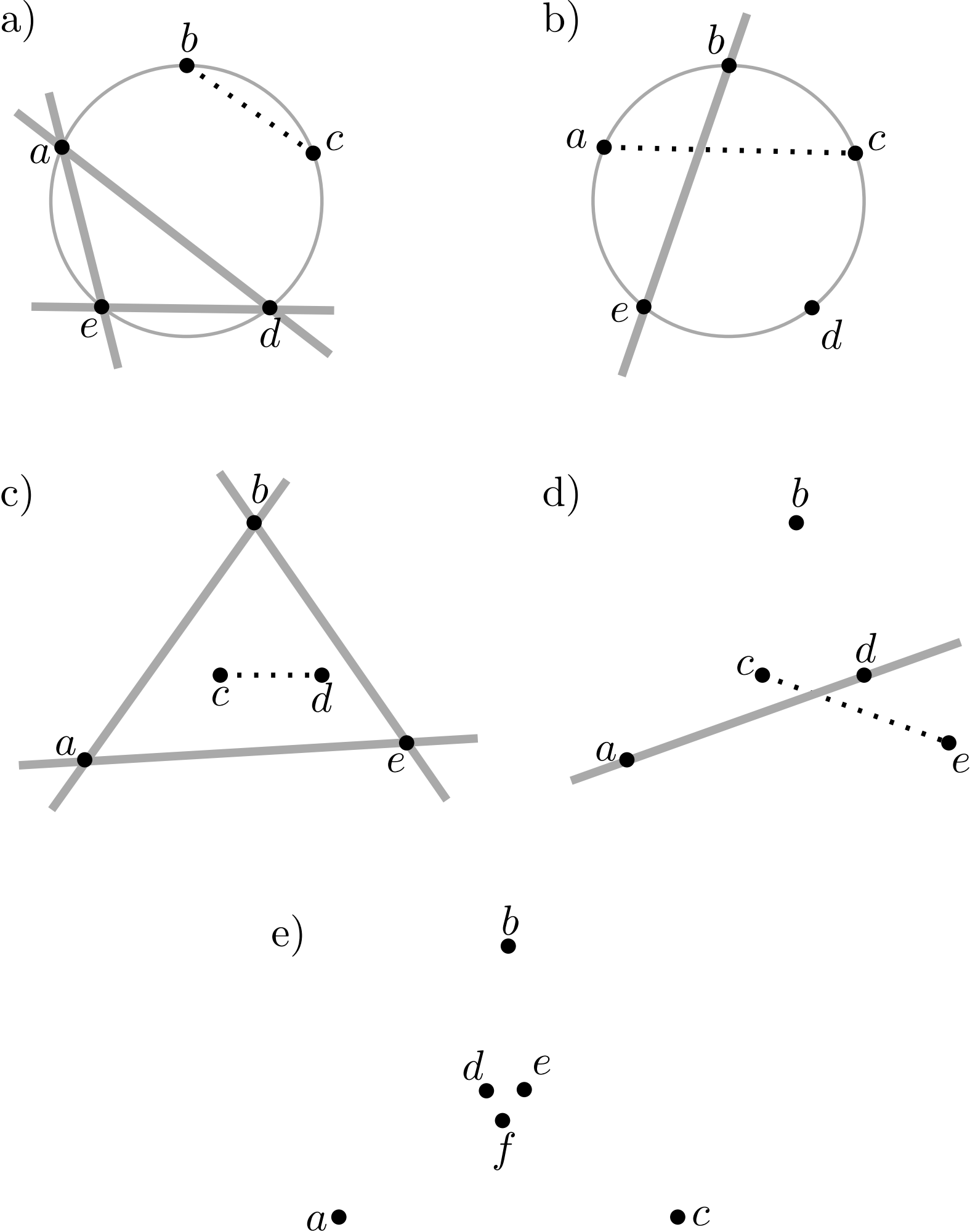


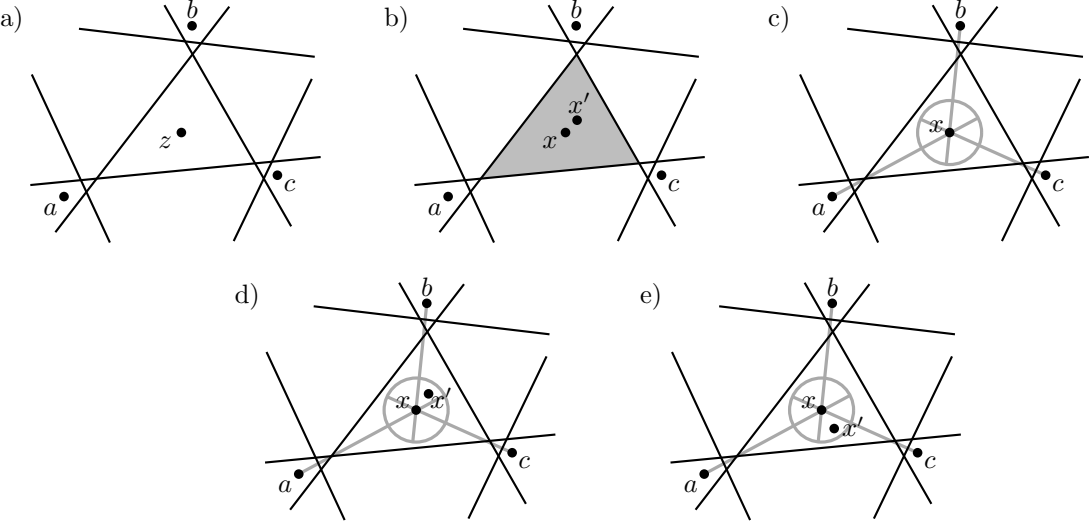
d)

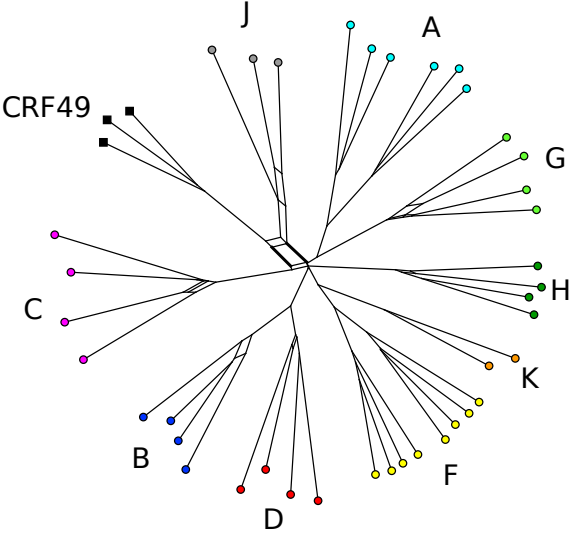


e)

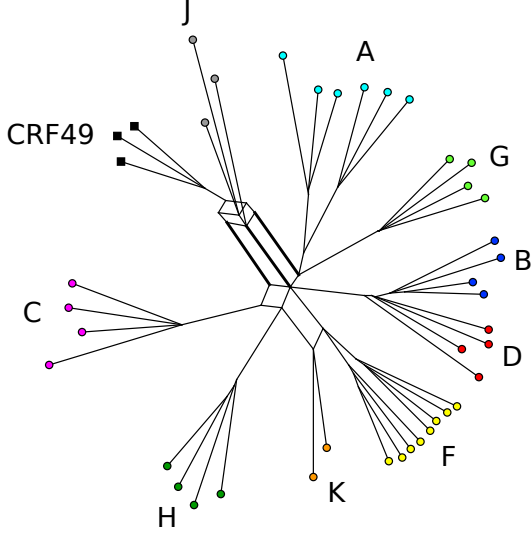



a)
b
a
c
e
d
b)
b
a
c
e
d
c)
b
c
d
a
e
d)
b
c
d
a
e
e)
b
d
e
f
a
c

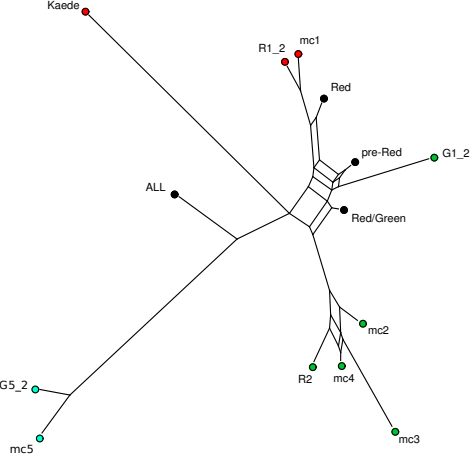

a)
b
z
c
a
b)
b
x′
x
c
a
c)
b
x
c
a
d)
b
x
x′
c
a
e)
b
x
x′
c
a



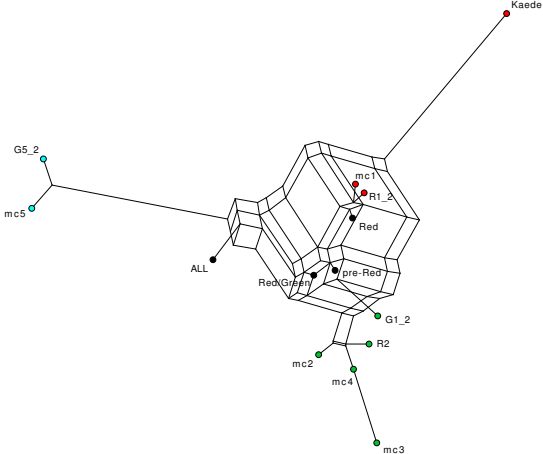
NeighborNet



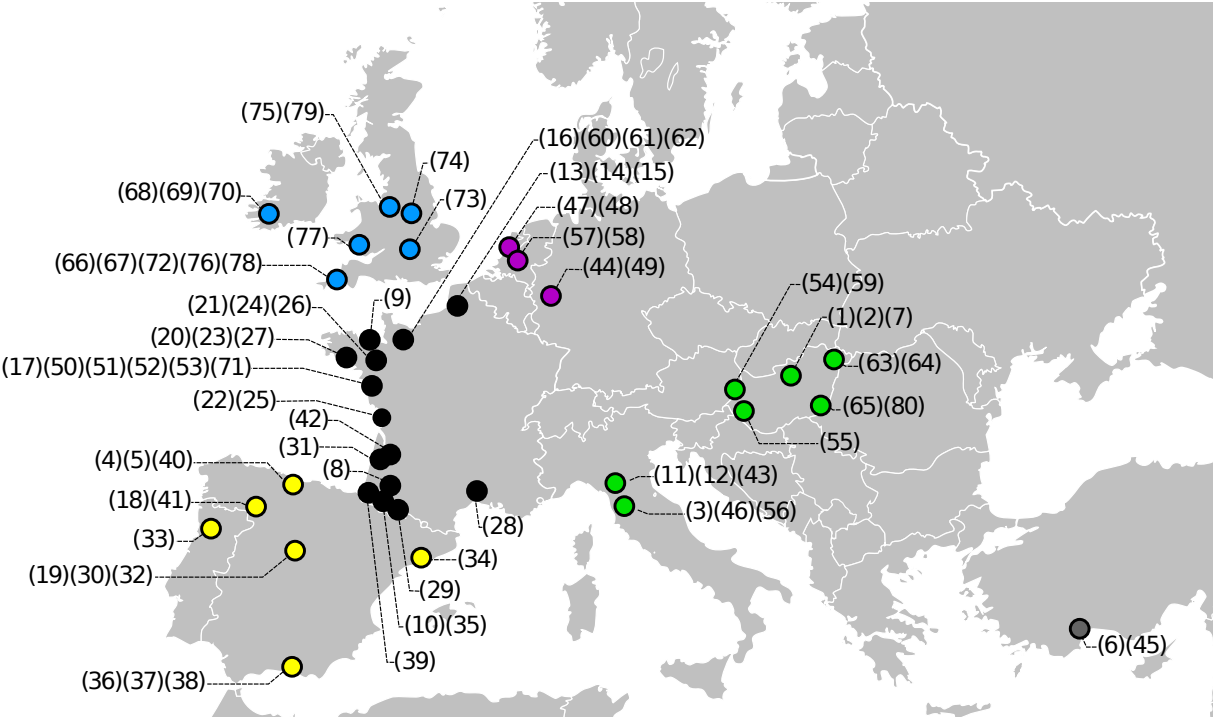
FlatNJ

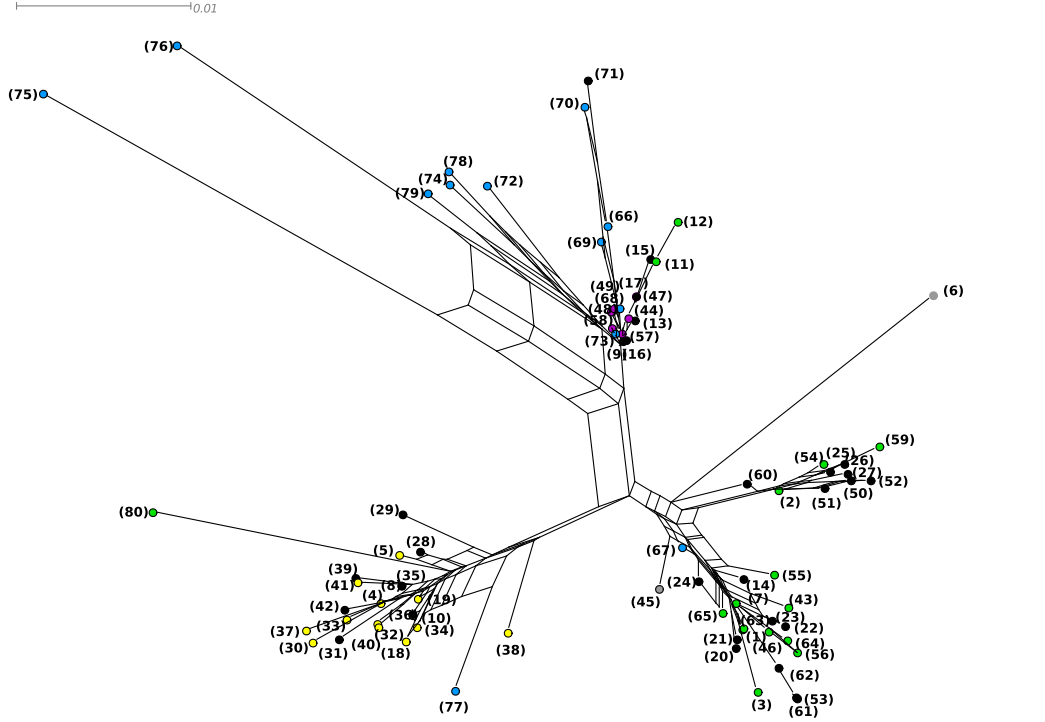


NeighborNet

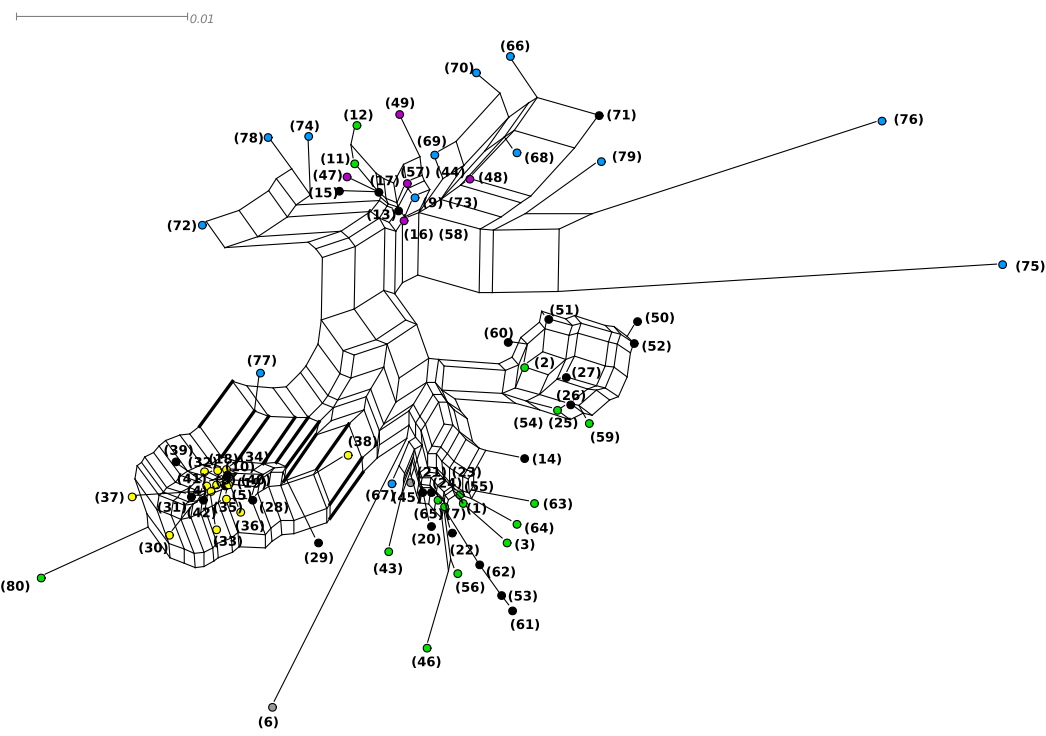


FlatNJ

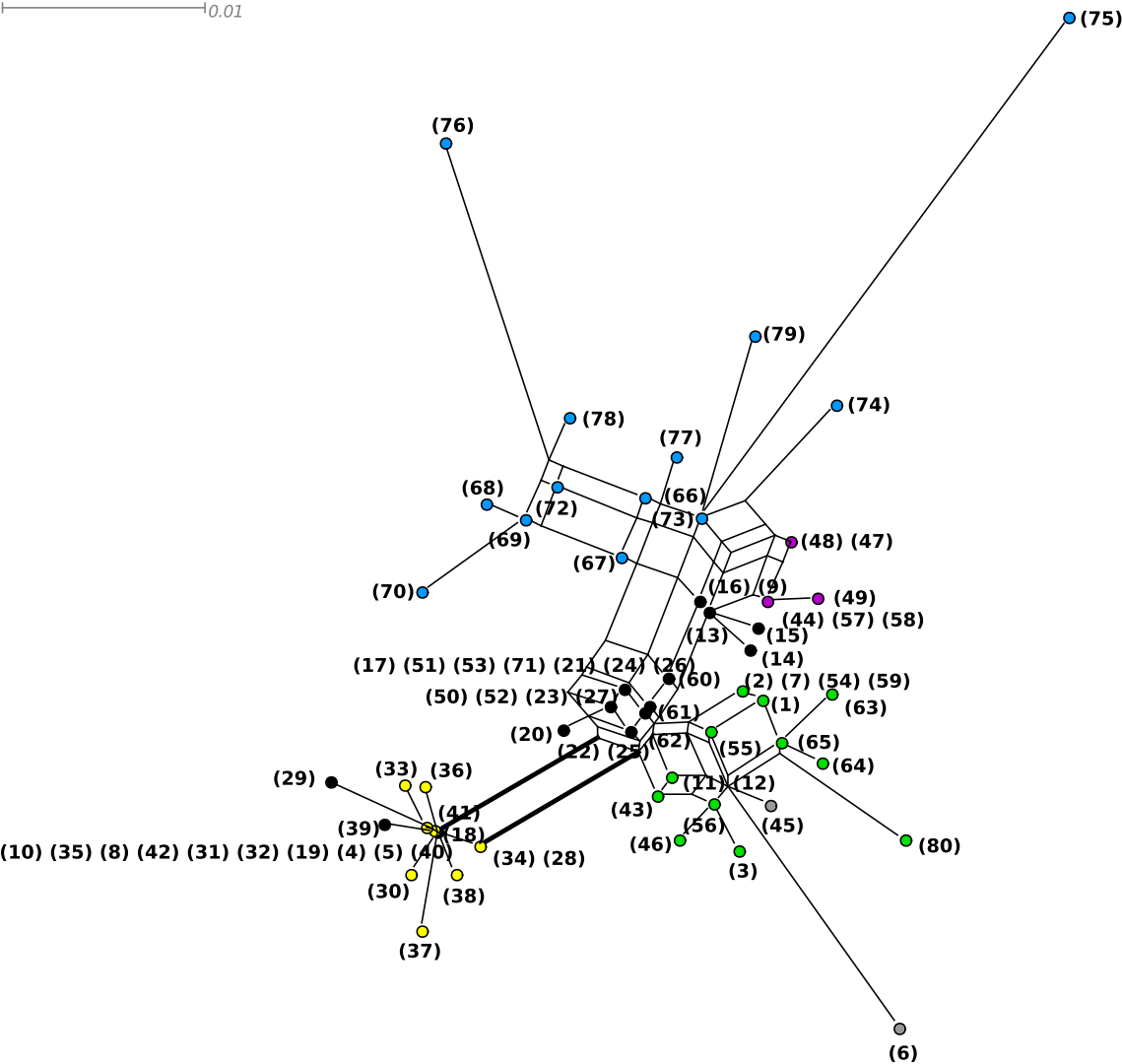

(75)(79)
(74)
(68)(69)(70)
(73)
(77)
(66)(67)(72)(76)(78)
(16)(60)(61)(62)
(13)(14)(15)
(47)(48)
(57)(58)
(44)(49)
(21)(24)(26)
(9)
(20)(23)(27)
(17)(50)(51)(52)(53)(71)
(22)(25)
(42)
(31)
(4)(5)(40)
(8)
(18)(41)
(33)
(19)(30)(32)
(28)
(34)
(29)
(10)(35)
(39)
(36)(37)(38)
(54)(59)
(1)(2)(7)
(63)(64)
(65)(80)
(55)
(11)(12)(43)
(3)(46)(56)
(6)(45)



NeighborNet



FlatNJ


0.01
(75)
(76)
(79)
(74)
(78)
(77)
(68)
(66)
(72)
(73)
(69)
(48) (47)
(67)
(70)
(16) (9)
(49)
(44) (57) (58)
(13)
(15)
(14)
(17) (51) (53) (71) (21) (24) (26)
(60)
(2) (7) (54) (59)
(50) (52) (23) (27)
(1)
(63)
(61)
(20)
(62)
(65)
(22) (25)
(55)
(64)
(29)
(33)
(36)
(11) (12)
(43)
(41)
(56)
(45)
(39)
(18)
(46)
(80)
(10) (35) (8) (42) (31) (32) (19) (4) (5) (40)
(34) (28)
(3)
(30)
(38)
(37)
(6)