Topological representation of metric data: the ABC Theorem

L. Ridgway Scott
University of Chicago, Chicago, Illinois, USA

August 23, 2019

Abstract

Tree representations of metric data provide a powerful way to understand relationships. We illustrate this with data drawn from protein biophysics. We review the connection between metrics for data relationships and corresponding tree representations, including some standard algorithms currently used. We suggest a new approach in which only topological features of trees are of interest. We introduce the ABC Theorem as the simplest possible realization of this approach.

The advent of data science requires new data analytics tools. Here we introduce one such tool that resolves a conundrum regarding the relationship between data metrics and corresponding tree representations. The *four-point condition* is a barrier to understanding data via trees, and we suggest a way that this condition can be relaxed in a useful way. The relaxation is realized in what we call the ABC Theorem.

Data science has arisen from various domains, each of which has its own data formats and corresponding analytical challenges. The domain driving the presentation here is protein biophysics [38]. We use data from this area to motivate the study.

The material here is designed to be tutorial in nature, so we have included exercises to amplify its study. This material can be viewed as a companion to the book [38] in which protein biophysics is studied in depth.

1 Tree representations of data

The process of evolution is naturally expressed via a tree [9, 41]. For this reason, tree representations of data are widely used in biology. We have used a tree representation of relationships among different myoglobin molecules in various species in Figure 1. In that representation, the distances along the tree are intended to represent the evolutionary distance, as measured by sequence similarity, between different versions of myoglobin. Myoglobin is found in many species, and thus it is a natural object of study [21].

Trees are not the only way to represent relationships in biology [25, 26, 27, 35, 40]. Simple clustering techniques [4, 35] are often used. On the other hand, one could represent distances by a general graph [25]. A compromise between a tree and a general graph is represented by 'reticulated' representations [27]. However, tree representations are very widely used in biology

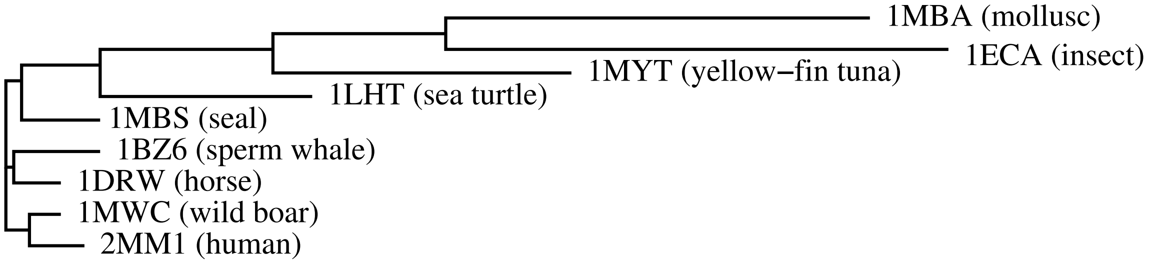

Figure 1: The protein myoglobin found in various species [18], which are presented in an evolutionary tree determined by sequence alignment distances.

and thus deserve significant attention. In many cases, only the topology of the representation is significant [26], and we discuss some issues related to the uniqueness of the topology of tree representations.

We explain here some of the basic issues with tree representations from a mathematical point of view, mainly that they rarely represent the data exactly. There is a precise condition that must be satisfied for a set of distances to be representable by a tree, and generically this would not be satisfied. Moreover, different algorithms [4, 13, 16, 23, 24, 36] are commonly used, and they do not give the same trees in many cases. Although we are not able to explain the differences that may arise in general, we do give one hint of how this may be tolerated in practice. We also give various examples of interest as illustration.

Determining tree representations also suffers from challenges that are purely biological [6, 42, 43]. These mainly have to do with the difficulty of determining relationships between different biological entities for which the data is likely incomplete. This gets reflected in the mathematical problem of tree determination in that the definition of distance may be imprecise. Unfortunately, the compounding of two types of fuzziness does not tend to cancel the fuzziness, but perhaps the one makes the other irrelevant. We will derive at least one result that is robust with respect to uncertainties in the distances.

The objects represented by trees in biology can vary, from individual proteins to entire species. Our main application will be to compare different proteins. The notion of distance used to define the trees will vary correspondingly, but in different directions. Even for a particular set of objects, such as proteins, it is possible to define potentially different notions of distance between individual proteins. For example, although the frequently used definition of distance derives from sequence similarly, we show that using particular features can also provide useful information. For completeness, we will begin with basic information about distance metrics and review the basic concepts of distance based on sequence alignment.

2 Distance metrics

The concept of distance is formalized mathematically in a **metric space**. Finite metric spaces are quite simple. They are characterized by having objects that can be labeled by integers $i = 1, \ldots, N$, and corresponding distances between the points given by a **distance matrix** $\mathcal{D}(i, j)$ where the individual values of the entries of the matrix are the distances in the space between objects i and j. For the time being, think of the objects as being different proteins and the distances some measure of similarity among them.

Several of the key properties of distance matrices are both easy to verify by inspection and easy to motivate. By definition, the diagonal of $\mathcal{D}$ is always zero, since the distance from one point to itself should be zero. Also, since they are distances, all of the entries are positive. Similarly, the matrix is assumed to be symmetric, reflecting the presumption that the cost of going from A to B is the same as going from B to A. In some cases, this might not be true, so a different kind of mathematics would need to be used.

The most important condition on a distance matrix is the **triangle inequality**. This condition encapsulates an important geometric condition and is not easily verified by inspection.

2.1 Triangle inequality

The triangle inequality for a distance matrix $\mathcal{D}$ requires that

$$\mathcal{D}(i,j) \leq \mathcal{D}(i,k) + \mathcal{D}(k,j) \tag{1}$$

for all $i, j, k = 1, \ldots, N$, where N is the total number of objects in the metric space. For example, if there are only three objects in the metric space (three proteins, say), then the general form of a distance matrix is

$$\begin{pmatrix} 0 & a & b \\ a & 0 & c \\ b & c & 0 \end{pmatrix} \tag{2}$$

where a, b, and c are positive parameters. Suppose that, without loss of generality, that c is the smallest:

$$c \leq \min\{a, b\}. \tag{3}$$

(If not, re-label the names of the elements of the metric space.) Then we leave as Exercise 9.2 that the triangle inequality implies that

$$|b - a| \leq c. \tag{4}$$

We leave as Exercise 9.3 to show that any coefficients a, b, and c satisfying the bounds (3) and (4) generate a distance matrix of the form (2) satisfying the triangle inequality. Thus the set of possible values for a, b, and c occupy a non-convex cone (with non-empty interior) in three dimensions defined by the bounds (3) and (4) (cf. Exercise 9.4).

2.2 Non-metric measurements

In many cases in biology, there are values $\widetilde{\mathcal{D}}(i,j)$ that represent relationships between objects i and j that are all positive but do not necessarily satisfy the triangle inequality (1). Thus they do not faithfully represent the 'cost' or 'distance' in going from i to j. For example, if $\widetilde{\mathcal{D}}(i,j) > \widetilde{\mathcal{D}}(i,k) + \widetilde{\mathcal{D}}(k,j)$, then the path

$$i \to k \to j \tag{5}$$

represents a cheaper way (with cost $\widetilde{\mathcal{D}}(i,k)+\widetilde{\mathcal{D}}(k,j)$) to get from i to j, rather than going directly. Thus $\widetilde{\mathcal{D}}$ no longer satisfies our general notion of a collection of direct distances. However, it is possible to define a metric related to $\widetilde{\mathcal{D}}$ that represents the relations faithfully, as follows.

Using the metaphor of distance, we can think of progressing from i to j via a finite sequence of steps like (5), i.e.,

$$i = k_0 \to k_1 \to \cdots \to k_r = j. \tag{6}$$

Then we define $\mathcal{D}(i,j)$ as the minimum cost over all possible paths of the type (6), viz.,

$$\mathcal{D}(i,j) = \min_{i=k_0,k_1,\ldots,k_r=j} \sum_{\ell=1}^{r} \widetilde{\mathcal{D}}(k_{\ell-1},k_\ell). \tag{7}$$

Without loss of generality, we can assume that all paths satisfy $k_{\ell-1} \neq k_\ell$, since if they are equal there is no contribution to the sum in (7), and the corresponding paths with the repetitions of terms eliminated lead to the same value.

Lemma 2.1 *Suppose that the matrix $\widetilde{\mathcal{D}}$ is symmetric, has all off-diagonal entries positive and all diagonal entries zero. Then the matrix $\mathcal{D}$ defined by (7) is a metric (in particular, satisfies the triangle inequality) and*

$$\mathcal{D}(i,j) \leq \widetilde{\mathcal{D}}(i,j) \tag{8}$$

for all i and j.

Proof. Since one of the possible paths is $i = k_0, k_1 = j$, we have $\mathcal{D}(i,j) \leq \widetilde{\mathcal{D}}(i,j)$ for all i and j, which proves (8).

Let $\epsilon = \min\left\{\widetilde{\mathcal{D}}(i,j) \mid i,j = 1,\ldots N\right\}$ and $\mu = \max\left\{\widetilde{\mathcal{D}}(i,j) \mid i,j = 1,\ldots N\right\}$. Then

$$\sum_{\ell=1}^{r} \widetilde{\mathcal{D}}(k_{\ell-1},k_\ell) \geq \epsilon r. \tag{9}$$

In view of (8),

$$\mathcal{D}(i,j) \leq \widetilde{\mathcal{D}}(i,j) \leq \mu \quad \text{for all } i,j.$$

Thus, we can restrict to paths such that

$$\sum_{\ell=1}^{r} \widetilde{\mathcal{D}}(k_{\ell-1},k_\ell) \leq \mu, \tag{10}$$

because paths with a larger sum will not contribute to a lower value of the minimum in (7). So we can assume that $r \leq \mu/\epsilon$. Thus the number of paths can be assumed to be finite and the minimum in (7) is thus positive for all i and j. Moreover, the minimum must be attained by some (not necessarily unique) path in (7):

$$\mathcal{D}(i,j) = \sum_{\ell=1}^{r} \widetilde{\mathcal{D}}(k_{\ell-1},k_\ell), \tag{11}$$

for some path of the form (6).

The matrix $\mathcal{D}$ defined by (7) is symmetric because any path from i to j given via steps $i = k_0, k_1, \ldots, k_r = j$ gives a path from j to i given by $j = k_r, k_{r-1}, \ldots, k_1 = i$, and conversely. And the inequality (8) guarantees that $\mathcal{D}$ is zero on the diagonal.

Now let us confirm that the triangle inequality (1) holds for the matrix $\mathcal{D}$ defined by (7). Let i, j, and m be arbitrary. Let $i = k_0, k_1, \ldots, k_r = m$ be a path such that

$$\mathcal{D}(i, m) = \sum_{\ell=1}^{r} \widetilde{\mathcal{D}}(k_{\ell-1}, k_\ell). \tag{12}$$

Similarly, let $m = k'_0, k'_1, \ldots, k'_{r'} = j$ be a path such that

$$\mathcal{D}(m, j) = \sum_{\ell=1}^{r'} \widetilde{\mathcal{D}}(k'_{\ell-1}, k'_\ell). \tag{13}$$

Using the path $i = k_0, k_1, \ldots, k_r = m = k'_0, k'_1, \ldots, k'_{r'} = j$ as a possible path in (7) proves that

$$\mathcal{D}(i, j) \leq \sum_{\ell=1}^{r} \widetilde{\mathcal{D}}(k_{\ell-1}, k_\ell) + \sum_{\ell=1}^{r'} \widetilde{\mathcal{D}}(k'_{\ell-1}, k'_\ell) = \mathcal{D}(i, m) + \mathcal{D}(m, j). \tag{14}$$

QED

2.3 Finding the right metric

In studying any objects mathematically, it is important to cast them into a space in which they can be compared via some metric. To study proteins and their interactions, it is natural to look for ways to compare proteins. The similarity of proteins can be measured in various ways. The most common way is via sequence similarity. We will briefly review the main concepts in Section 4.

Another way to measure similarity is based on structure. Protein folds provide a way to compare proteins. The structure is often related to function, and it is natural to use function as a measure of similarity. It might well be that all of these measures are equivalent in some sense. But what is striking about these notions of similarity is that two sequences can have similar function and fold similarity while having only fractional sequence similarity. That is, two proteins with, say, 30% sequence similarity are often of biological interest for their potential biological similarity. How can proteins that differ in the vast majority of their sidechains nevertheless be biologically similar?

We address this question by studying measures of similarity based on key features that determine protein function. This illustrates that only key parts of protein sequence are crucial for determining protein function, e.g., by governing protein-ligand interaction. For example, we explain a distance between proteins based on comparing their dehydrons. In some cases, it is possible to show that this measure of distance alone is sufficient both to characterize evolutionary relationships among classes of proteins but also to explain pharmacological relationships.

3 Distance metrics, similarity trees and dendrograms

One way to define metrics for biological entities is by comparing feature differences [28]. Different features can indicate a propensity for protein-protein interaction, such as surface curvature, wrapping (dehydrons), hot spots, etc. These features can be compared on different proteins

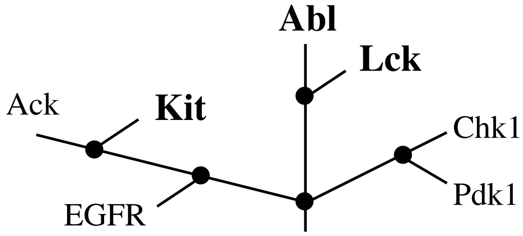

Figure 2: Packing similarity tree (PST) for the seven structurally aligned paralogs of Bcr-Abl, listed as 'Abl' in the diagram, based on the dehydron distances in Table 1. The paralogs in **bold** have the most similar packing in the region that aligns with the Imatinib-wrapped region in Bcr-Abl and are also primary targets of this inhibitor [14].

	Ack	Kit	Abl	Lck	Chk1	Pdk1
EGFR	2	2	3	3	4	4
Ack		1	4	4	5	5
Kit			4	4	5	5
Abl				1	4	4
Lck					4	4
Chk1						1

Table 1: Dehydron distances for seven homologous tyrosine kinase proteins. PDB files corresponding to the abbreviations in the table are: Abl (1FPU), Ack1 (1U54), Chk1 (1IA8), C-kit (1T45), EGFR (1M17), Lck (3LCK), Pdk1 (1UVR).

to see if they are similar or not. Different features may not be conserved across paralogs. A numerical measure can help to quantify relationships. This can be done in a variety of ways, but one way is to first form a standard structural alignment[1] and then to trim the protein sequences by restricting to those residues that have been structurally aligned in a one-to-one correspondence.

3.1 HB Hamming distance

One metric that can be useful is to compare hydrogen bonds (HBs) in the different, aligned structures. An indicator matrix $A(i,j)$ for each protein is constructed that is indexed by the residues involved in hydrogen bonds. That is, $A(i,j)=1$ if i and j are linked by a hydrogen bond, and zero otherwise. Then, a Hamming-type distance [17] can be defined based on the number of disagreements between two such indicator matrices:

$$d(A,B)=\sum_{i,j}|A(i,j)-B(i,j)|. \tag{15}$$

The Hamming distance (15) will depend on the alignment chosen. Thus it is possible to minimize over all such alignments. But for now, we consider the computation for just one fixed alignment.

[1] http://www.ncbi.nlm.nih.gov/structure/CN3D/cn3d.shtml

3.2 Packing distance

Metrics based on comparison of hydrogen bond structures can also be refined by limiting to certain classes of hydrogen bonds. A very successful use of this idea is to restrict to the underwrapped hydrogen bonds (the dehydrons) [19]. Since this metric reflects defects in the packing of the hydrogen bonds by hydrophobic groups, it is called the 'packing distance.' The packing distance for seven homologous tyrosine kinase proteins [30] is depicted in Table 1.

For dehydron analysis, an indicator matrix for each protein is constructed that is indexed by the residues involved in dehydrons. It consists only of zeros unless two residues are paired by a dehydron, in which case the corresponding entry is one. Again, a Hamming-type distance (15) can be defined based on the number of disagreements between two such indicator matrices. We refer to this as **dehydron distance** or **packing distance**.

3.3 Other metrics

Another example of restricting to special hydrogen bonds is depicted in Table 2. Here the distance is based on the intermolecular hydrogen bonds formed between the antibody and the antigen in homologous antibody-antigen structures.

A distance based on nonpolar regions on proteins (a.k.a., hydrophobic patches) can similarly be defined [8]. A pharmacological distance matrix can also be constructed based on affinity profiling against specified drugs [14, 17].

Many other features could potentially be employed to define useful metrics. For example, geometric [3] features such as surface curvature can be measured by

$$\mathcal{D}(m,n) = \sum_i |K_i(m) - K_i(n)| \tag{16}$$

where K_i is the curvature of the protein-ligand interface at the i-th position of aligned proteins m, n.

A pharmacological distance matrix can also be constructed based on affinity profiling against specified drugs [17, 14]. The dehydron distance for a set of kinases displays remarkable similarity to pharmacological distance defined using these kinases and a set of available drugs [17, 14]. Tree representations familiar in phylogenetic analysis [34, 24] provide a useful, albeit not unique, representation of relationships among drug targets. The direct correlation of distance matrices provides a unique comparison and confirms the efficacy of wrapping technology in predicting selectivity [17].

3.4 Distances via trees

Distance matrices can be used to construct dendrograms (or trees) and are implicitly behind the dendrograms commonly seen [14, 20]. Such trees are easily constructed by using standard algorithms for phylogenetic analysis [24]. Distance in the tree reflects distance as encoded in the distance matrix, with the proviso that certain restrictions must apply to the distance matrix to get an exact representation [24]. For example, with dehydrons, we refer to this tree as a **packing similarity tree** (PST). The resulting tree gives a visual indication of closeness for the various proteins, cf. Figure 2 which depicts one such tree. The PST allows the assessment of possible effects of targeting given features in drug design to determine potential specificity. Nearby proteins (Abl and Lck, for example) have similar features in our simple example.

	1DQJ	1C08	1NDG
1NDM	3	7	6
1DQJ		6	5
1C08			5

Table 2: Distance matrix for intermolecular hydrogen bond interactions between antibody and antigen in the PDB complexes: a=1NDM, b=1DQJ, c= 1C08, d= 1NDG. Zeros on the diagonal and the redundant lower triangular part of the matrix have been omitted.

Although visual inspection and comparison of trees is useful, the lack of uniqueness of trees is a cause for concern. A direct comparison of distance matrices provides a more rigorous comparison of measured properties, such as a comparison of pharmacological distance and packing distance [17]. In general, if a distance matrix is to be represented faithfully by a tree, it must satisfy a **four-point condition** [24] that includes the familiar triangle inequality but is substantially more restrictive. This requirement implies that typical biological trees will not uniquely represent a given biological distance matrix. Thus direct comparison of distance matrices is a more reliable technique [17].

3.5 Relating distances to trees

In some cases, the relationship between distance data and a tree is easy to comprehend, as is the case for Table 1 and Figure 2. The latter is a particularly simple tree in which the distance between the leaves and the parent node is exactly half the distance between the parent nodes themselves. If we take that distance to be one, then we obtain the distances given in Table 1. Correspondingly, we can relate the distances given in Table 1 to the number of parent nodes on the shortest path between leaves of the tree.

The key concept of tree representation of similarity is that distance in the tree is intended to reflect distance as encoded in the distance matrix, as is the case in Table 1 and Figure 2. However, we will see that in general this does not easily hold. There are certain restrictions that apply to the distance matrix to get an exact representation [24]. There is often no tree that exactly represents the distance data, and the set of trees that closely approximate it is often not well defined. We will refer to this as a lack of uniqueness in the tree representation, although the situation is really more complex than that. Thus the general use of trees to represent distance data is problematic.

4 Sequence distance metrics

The simplest form of sequence comparison is based on editing strings and counting the number of edits required to get from one sequence to the other. The formulation of string-edit distance d_e balances two different types of edits. The simplest is replacement of a single letter by another letter. To start with, we need a metric on the set of letters in the alphabet Σ for our sets of sequences. Let D_Σ be a metric on the alphabet Σ. Then $D_\Sigma(\xi, \eta)$ measures the distance between the two letters ξ and η in Σ, that is, the cost of changing from ξ to η at a single point in the string. Now we show how to extend this to a metric d_e on the set of all finite strings Σ^*.

If two strings x and y differ only in the k-th position, then we set $d_e(x,y) = D_\Sigma(x_k, y_k)$. In general, when there are multiple replacements, string edit distance is based on just summing the effects. However, string-edit distance also allows a different kind of change as well: insertion and deletion. For example, we can define $x_{\hat{k}}$ to mean the string x with the k-th entry removed. It might be that $x_{\hat{k}}$ agrees perfectly with the string y, and so we assign $d(x,y) = \delta$ where δ is the deletion penalty. Similarly, insertions of characters are allowed to determine edit distance. Clearly, if $y = x_{\hat{k}}$, then adding x_k to y at the k-th position yields x. Again, the effect of multiple insertions/deletions is additive, and this allows strings of different lengths to be compared.

The use of both replacements and insertion/deletions to determine edit distance means that an edit path from x to y is not unique. Edit distance is therefore defined by taking the minimum over all possible representations, as we define formally in (20). But this will not in general define a metric unless appropriate conditions on δ and D_Σ are satisfied. These conditions can be defined by extending the alphabet Σ and metric D_Σ to include a "gap" as a character, say "_" (let $\widetilde{\Sigma}$ denote the extended alphabet), and by assigning a distance $D_{\widetilde{\Sigma}}(x, \_)$ for each character x in the original alphabet. Theorem 9.4 of [39] says that d_e is a metric on strings of letters in Σ whenever $D_{\widetilde{\Sigma}}$ is a metric on the extended alphabet.

4.1 Two-letter alphabets

The simplest non-trivial example is an alphabet with two letters, say x and y, when there is only one distance $D_\Sigma(x,y)$ that is non-zero. The requirement that the triangle inequality hold for $D_{\widetilde{\Sigma}}$ reduces to three inequalities that can be expressed as

$$\left|D_{\widetilde{\Sigma}}(x,\_) - D_{\widetilde{\Sigma}}(y,\_)\right| \le D_\Sigma(x,y) \le D_{\widetilde{\Sigma}}(x,\_) + D_{\widetilde{\Sigma}}(y,\_). \tag{17}$$

Together with the condition that all distances be non-negative, we see that (17) characterizes completely the requirement for $D_{\widetilde{\Sigma}}$ to be a metric in the case of a two-letter alphabet Σ.

4.2 General alphabets

For a general alphabet Σ, if there exists an $\alpha > 0$ such that

$$\alpha \le D_\Sigma(x,y) \le 2\alpha \tag{18}$$

for all $x \ne y$ (including _), then D_Σ is a metric (that is, the triangle inequality holds). This is because

$$D_\Sigma(x,y) \le 2\alpha \le D_\Sigma(x,z) + D_\Sigma(z,y) \tag{19}$$

for any $z \in \Sigma$. One simple choice for a metric on letters is to choose $D_\Sigma(x,y) = 1$ for all $x \ne y$, and then to take $D_{\widetilde{\Sigma}}(x,\_) = 2$; the resulting $D_{\widetilde{\Sigma}}$ satisfies (18) for $\widetilde{\Sigma}$. However, condition (18) is far from optimal as the example (17) shows.

The edit distance d_e is derived from the extended alphabet distance $D_{\widetilde{\Sigma}}$ as follows. We introduce the notion of *alignment* $\mathcal{A}$ of sequences $(x^*, y^*) = \mathcal{A}(x,y)$ where $x*$ has the letters of x in the same order but possibly with gaps _ inserted, and similarly for y^*. We suppose that $x*$ and y^* have the same length even if x and y did not, which can always be achieved by adding gaps at one end or the other. Then

$$d_e(x,y) = \min_{\mathcal{A}} \sum_i D_{\widetilde{\Sigma}}(x_i^*, y_i^*). \tag{20}$$

The minimum is over all alignments $\mathcal{A}$ and the sum extends over the length of the sequences. Fortunately, string-edit distance d_e, and even more complex metrics involving more complex gap penalties, can be computed efficiently by the dynamic programming algorithm [39].

The simple string-edit distance d_e described here is useful in many contexts. However, more complex metrics would be required in other applications.

4.3 Distance versus score

Typically biologists prefer to work with a "score" that is large when two proteins are close as opposed to a distance which is small in such a case. The dynamic programming algorithm can equivalently be used to minimize the distance or maximize a score. There is a formal correspondence that can be made between scores and distances, as follows [39].

Since we see that distance and score are simple duals of each other, we will continue to work with the notion of distance rather than the notion of score.

4.4 Sequence space geometry

Let us try to acquire some intuition about what it means for sequences to be close in edit distance. For concreteness, we take the simple example presented in Section 4.1, namely, $D_\Sigma(x, y) = 1$ for all $x \neq y$, and $D_{\bar{\Sigma}}(x, \_) = 2$. Then two sequences are at distance one apart if and only if they differ in exactly one place in the sequence, and there are no gaps. Suppose that we restrict to sequences which are all of length exactly n. Then for any sequence, it has exactly n neighbors which are a distance one away. This means that the sequences of size n appear somewhat like an n-dimensional space. Note that there are no fractional distances. Either the distance is zero, in which case the sequences are identical, or the distance has to be at least one.

Now let us characterize sequences at a distance of two. This can occur in two ways. There could be a simple gap in one, with an otherwise perfect match in the two sequences. Or there could be no gap but instead two individual letter mismatches. Two sequences of length n cannot agree exactly with just one gap, since the lengths would not match. But a sequence of length $n - 1$ could match, with a gap, a sequence of length n. In fact, there are n such sequences of length $n - 1$. But there are many more sequences with two letter mismatches, namely $\mathcal{O}(n^2)$. This means that the space does not really look n dimensional, because the number of points in an n dimensional space of distance c from a given point would be of the order c^n, not n^c. But in any case, for long sequences (proteins would have n in the hundreds), there are a lot of neighbors at the closest possible distances.

The maximum distance between any two sequences of length n is at most $4n$: each sequence is matched to a gap. Another instructive case is a pair of sequences with no letters in common but also not gaps, so the distance is n. Two sequences that agree exactly for half of the sequences and align without gaps, the distance is $n/2$. But the number sequences at a distance $n/2$ is $\mathcal{O}(n^{n/2})$. Including gaps only increases the number of neighbors at distance $n/2$. So how could we say that agreement of only half of the sequence is a good match? That is, if we declare two protein sequences with 50% identity to be similar, then what about the other $\mathcal{O}(n^{n/2})$ proteins which are the same distance away?

The answer undoubtedly is that most of these other sequences have no biological significance. For example, they would likely not fold into a stable, three-dimensional structure [15, 11]. Rather than being concerned that there are so many possible neighbors, we might instead think that

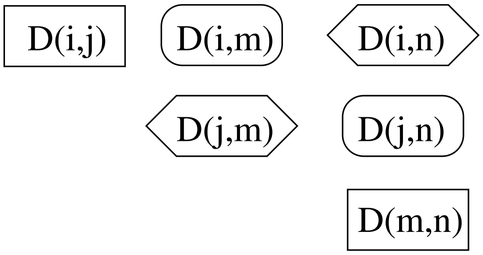

Figure 3: Entries in the distance matrix which are constrained by the four point condition.

biological proteins make up a rather sparse (and randomly distributed) set of sequences in Σ^*. In this scenario, it is remarkable to find sequences that are close at all. This seems to be the biologically relevant point of view. However, it is still interesting to ask what determines that proteins are close. We will see that a much simpler metric can provide some insight.

5 Tree representation of metrics

A tree naturally defines a metric space for the leaves of the tree, where the distance between leaves is the shortest path in the tree. We now invert this observation and ask which metrics can be represented by trees.

5.1 Three point metrics

A distance matrix $\mathcal{D}(i,j)$ for three points can always be represented by a tree. There are only three unique values in the distance matrix: $\mathcal{D}(1,2)$, $\mathcal{D}(2,3)$ and $\mathcal{D}(1,3)$. A tree that connects the points 1, 2, and 3 via a central node (call it 0) has only three distances to define it: the distance δ_i from 0 to node i for $i = 1,2,3$. The relationship between the two representations of distance is $\mathcal{D}(i,j) = \delta_i + \delta_j$. It is easy to see that this 3×3 system of equations is invertible.

However, in general a distance matrix $\mathcal{D}(i,j)$ for $k > 3$ points cannot be exactly represented by the distances in a tree. We will see this explicitly in the case $k = 4$, but it is easy to see why there is a difficulty in this case. The tree of interest will take the form in Figure 5. There are only five internal distances available in this tree. But there are six different distinct values in a distance matrix $\mathcal{D}(i,j)$ for $k = 4$. Even if these six values are chosen to satisfy the triangle inequality, there is one too many of them to be able to be matched by five parameters.

Different algorithms are used in practice to produce a tree from a metric, and they have the property that given a distance matrix $\mathcal{D}(i,j)$ they will produce an answer. It is not known in general what the relationship is between the products of different popular algorithms. However, for $k = 4$ we can at least give an interpretation of the extent of ambiguity.

5.2 Four point condition

In general, if a distance matrix is to be represented faithfully by a tree, it must satisfy the following *four-point condition* [5, 24, 37]

$$\mathcal{D}(i,j) + \mathcal{D}(m,n) \leq \max\{\mathcal{D}(i,m) + \mathcal{D}(j,n), \mathcal{D}(j,m) + \mathcal{D}(i,n)\}, \tag{21}$$

for all i, j, m, and n. The four-point condition generalizes the triangle inequality (take $m = n$). For a distance matrix satisfying the triangle inequality, it suffices to take i, j, m, and n distinct. The relationship between the various values are depicted in Figure 3. In the case of only four points, the entire (upper-triangular part of the) distance matrix is depicted in Figure 3, and the entries making up the three terms in (21) are enclosed in like enclosures (rectangle, oval, hexagon). In the case of a general distance matrix, the same picture obtains after eliminating all rows except i, j and m, and all columns except j, m and n.

We will see that generically most distance matrices do not satisfy the four-point condition. This implies that typical biological trees will not rigorously represent a given biological distance matrix.

Definition 5.1 *A matrix that satisfies the four point condition is called* **additive***.*

The following theorem may be found in [24].

Theorem 5.1 *A distance matrix can be represented by distances in a tree if and only if it satisfies the four point condition (i.e., it is additive).*

There is a simple interpretation of the four point condition, as follows. Given distinct values of i, j, m, and n, define parameters A, B, and C by

$$A = \mathcal{D}(i,j) + \mathcal{D}(m,n), \quad B = \mathcal{D}(i,m) + \mathcal{D}(j,n), \quad \text{and} \quad C = \mathcal{D}(j,m) + \mathcal{D}(i,n). \tag{22}$$

When we have three values A, B, and C, it is natural to expect that there are three distinct values given by

$$\min\{A,B,C\} \le \text{mid}\{A,B,C\} \le \max\{A,B,C\} \tag{23}$$

Here, the 'mid' function picks out the value between the min and the max. The four-point condition is the requirement that

$$\text{mid}\{A,B,C\} = \max\{A,B,C\}. \tag{24}$$

Otherwise said, of the three values A, B, and C, the two largest must be equal for the four-point condition to hold.

As an example, consider the intermolecular hydrogen bond distance data presented in Table 2. For a metric space with only four points, there is only one choice for the distinct values of i, j, m, and n, although relabeling is possible. For the data in Table 2, we find that

$$\{A,B,C\} = \{8,12,12\}, \tag{25}$$

so Table 2 is an additive matrix.

It is possible to motivate the four-point condition by considering a simple algorithm for reducing the size of the metric space. This algorithm is only suggestive, but it does make clear why the condition arises, and it also motivates one of the widely used algorithms for determining trees from data.

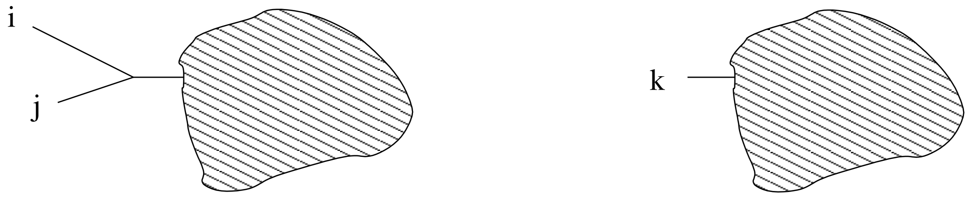

Figure 4: Original tree (left) and reduced tree (right) where nodes i and j have been removed and replaced by node k. The corresponding distances are defined via (26).

5.3 A reduction algorithm

Consider the following algorithm for constructing a tree from a distance matrix. Loop over all pairs i, j. Suppose there is a graph representation with nodes i and j appearing as leaves of a parent node as depicted on the left-hand side of Figure 4. Call the parent node k, where k is an index not being used in the current indexing scheme. Then define

$$\mathcal{D}(m,k) = \mathcal{D}(k,m) = \tfrac{1}{2}\left(\mathcal{D}(i,m) + \mathcal{D}(j,m) - \mathcal{D}(i,j)\right) \quad \forall m \neq i, j. \tag{26}$$

Create a new discrete space by eliminating i and j and adding k; in terms of the distance matrix, we eliminate the i and j rows and add the new information defined by (26). By the triangle inequality, this new matrix is non-negative. Moreover, if we find $\mathcal{D}(k,m) = 0$ we can take $k = m$ and avoid the addition to the discrete space, so we can assume that this new matrix is non-degenerate.

This new matrix satisfies the triangle inequality, that is $\mathcal{D}(k,m) \leq \mathcal{D}(k,n) + \mathcal{D}(n,m)$ for all n. This is equivalent to

$$\left(\mathcal{D}(i,m) + \mathcal{D}(j,m) - \mathcal{D}(i,j)\right) \leq \left(\mathcal{D}(i,n) + \mathcal{D}(j,n) - \mathcal{D}(i,j)\right) + 2\mathcal{D}(n,m), \tag{27}$$

or equivalently

$$\mathcal{D}(i,m) + \mathcal{D}(j,m) \leq \mathcal{D}(i,n) + \mathcal{D}(j,n) + 2\mathcal{D}(n,m). \tag{28}$$

But the triangle inequality for the original matrix implies this: we just add

$$\mathcal{D}(i,m) \leq \mathcal{D}(i,n) + \mathcal{D}(n,m) \tag{29}$$

to

$$\mathcal{D}(j,m) \leq \mathcal{D}(j,n) + \mathcal{D}(n,m) \tag{30}$$

to prove (28).

The reduction to a smaller metric space just described motivates the popular UPGMA algorithm. The algorithm proceeds by clustering nodes i and j for which $\mathcal{D}(i,j)$ is smallest. The heuristic is that smaller $\mathcal{D}(i,j)$ values should mean that i and j are closer in the tree (which is correct) and (which is not correct) the closest points must be children of the same parent node in the tree. We give an example to show how this fails in Section 7.2.

5.4 Obstruction to reduction

The difficulty arises in the assignment of the distances between the new point and the deleted points. If all were well, we would have

$$\mathcal{D}(i,k) = \mathcal{D}(i,m) - \mathcal{D}(m,k) = \tfrac{1}{2}\left(\mathcal{D}(i,m) - \mathcal{D}(j,m) + \mathcal{D}(i,j)\right), \tag{31}$$

for any m, and similarly for $\mathcal{D}(j,k)$:

$$\mathcal{D}(j,k) = \mathcal{D}(j,m) - \mathcal{D}(m,k) = \tfrac{1}{2}\left(\mathcal{D}(j,m) - \mathcal{D}(i,m) + \mathcal{D}(i,j)\right). \tag{32}$$

Since m is arbitrary in (31), it must hold as well for any other node n replacing m. Since the left-hand side of (31) remains unchanged, we must have

$$\mathcal{D}(i,m) - \mathcal{D}(j,m) + \mathcal{D}(i,j) = \mathcal{D}(i,n) - \mathcal{D}(j,n) + \mathcal{D}(i,j) \tag{33}$$

for any m and n, which is the same as saying

$$\mathcal{D}(i,m) + \mathcal{D}(j,n) = \mathcal{D}(i,n) + \mathcal{D}(j,m), \tag{34}$$

which we will see is equivalent to the 'mid=max' interpretation of the four-point condition.

To complete the derivation of the four-point condition from (34), we note that the common value in (34) may be written as (see Exercise 9.10)

$$\mathcal{D}(i,m) + \mathcal{D}(j,n) = \mathcal{D}(i,n) + \mathcal{D}(j,m) = \mathcal{D}(i,j) + \mathcal{D}(k,m) + \mathcal{D}(k,n). \tag{35}$$

By the triangle inequality,

$$\mathcal{D}(i,j) + \mathcal{D}(m,n) \le \mathcal{D}(i,j) + \mathcal{D}(m,k) + \mathcal{D}(k,n). \tag{36}$$

Combining (36) and (35) completes the derivation of the four-point condition. Moreover, it proves that $\mathcal{D}(i,j) + \mathcal{D}(m,n)$ has to be the 'min' value of these ABC values, for all m and n. This appears at first to be stronger than the four point condition, so we state it formally.

Lemma 5.1 *Suppose that the matrix $\mathcal{D}$ can be represented as a tree. Then there are indices i and j such that*

$$\mathcal{D}(i,j) + \mathcal{D}(m,n) \le \min\{\mathcal{D}(i,m) + \mathcal{D}(j,n), \mathcal{D}(i,n) + \mathcal{D}(j,m)\} \tag{37}$$

for all m and n.

The problem of identifying indices i and j such that the reduction algorithm can be applied is not simple, and there are different ways to identify them for an additive distance matrix. Moreover, the different methods lead to different approximation algorithms in the case of non-additive distances. However, one algorithm derives directly from Lemma 5.1 [22] (also see page 326 [37]). Define the expression

$$P_{ij} = \#\big\{m, n \ne i, j \;\big|\; \mathcal{D}(i,j) + \mathcal{D}(m,n) \le \min\{\mathcal{D}(i,m) + \mathcal{D}(j,n), \mathcal{D}(i,n) + \mathcal{D}(j,m)\big\}, \tag{38}$$

where # means the cardinality of the set. The ADDTREE algorithm [37] consists of choosing i, j such that P_{ij} is maximized. In the definition of P_{ij} it is sufficient to assume that $m < n$. For an additive distance, Lemma 5.1 guarantees that there is some pair i, j such that the maximum possible value of P_{ij} is attained, so the maximization algorithm will always yield a pair satisfying (37). For non-additive distances, the ADDTREE algorithm also provides good approximations, as we will discuss shortly. An alternative way to express the ADDTREE algorithm is to define

$$R_{ij} = \#\left\{m, n \ne i, j \;\middle|\; \mathcal{D}(i,j) + \mathcal{D}(m,n) > \mathcal{D}(i,m) + \mathcal{D}(j,n)\right\}. \tag{39}$$

Then i, j are suitable neighbors if and only if $R_{ij} = 0$; the ADDTREE algorithm corresponds to minimizing R_{ij} for $i \neq j$ (cf. Exercise 9.11). Note that the computation of P can be simplified by assuming $m < n$, but the computation of R requires all m, n to be considered.

There is another condition that is known to determine suitable indices i and j for additive distances. The condition is to minimize the expression

$$\begin{aligned} Q_{ij} =& (N-2)\mathcal{D}(i,j) - \sum_{k=1}^{N} \mathcal{D}(i,k) - \sum_{k=1}^{N} \mathcal{D}(j,k) \\ =& (N-4)\mathcal{D}(i,j) - \sum_{k \neq i,j; k=1}^{N} \mathcal{D}(i,k) + \mathcal{D}(j,k), \end{aligned} \tag{40}$$

where N is the total number of points in the metric space. Although discovered some time ago [36], the condition based on minimizing (40) has received a great deal of attention [23] and continues to be of interest [1, 2, 12, 13, 32]. The exact form (40) is due to Studier and Kepler [22]. The condition (40) is known to be unique [7] among certain functionals for determining indices i and j.

The algorithm we have described for constructing a tree by recursively reducing the size of the metric space using the condition (40) is called **neighbor joining** (NJ). The neighbors being joined are i and j. The choice given by minimizing (40) is widely used, but not universal. The UPGMA algorithm uses the far simpler heuristic of minimizing $\mathcal{D}(i, j)$. We explain why this approach leads to the wrong answer in Section 7.2 even for an additive metric.

The leaves i and j identified as above not only allow the reduction algorithm to construct a tree for an additive metric, they also are good choices in the case that a metric is not additive [1, 2, 7, 12, 13, 23, 32, 37]. Although we will see that there are many interesting biological metrics that are additive, in general this is not the case. Thus it is of interest to understand to what extent we can approximate a general distance matrix via an additive metric [4].

It is possible to assess theoretically the approximation quality for both neighbor joining, using the criterion of minimizing (40) [7, 12, 13, 23, 32], and the ADDTREE algorithm [37], using the criterion of maximizing (38). It is known that both of these algorithms are stable in maximum norm, as we now explain.

Define the ℓ^∞-norm for distance matrices:

$$\|\mathcal{D}\|_{\ell^\infty} = \max_{i<j} |\mathcal{D}(i,j)|. \tag{41}$$

Let f denote the mapping from a distance matrix $\mathcal{D}$ to an additive tree $\mathcal{T}$ given by either of the algorithms based on (38) or (40). For any such tree, let $\epsilon(\mathcal{T})$ denote the length of the shortest edge in the tree.

Theorem 5.2 *[37] For all distance matrices $\mathcal{D}'$ such that*

$$\|\mathcal{D} - \mathcal{D}'\|_{\ell^\infty} < \tfrac{1}{2}\epsilon(\mathcal{T}), \tag{42}$$

then the algorithms NJ and ADDTREE form the same tree that is,

$$f(\mathcal{D}') = f(\mathcal{D}). \tag{43}$$

This theorem says that the tree formation algorithms are continuous with respect to small perturbations in the data. It is known that Theorem 5.2 is sharp, in that there are $\mathcal{D}$ and $\mathcal{D}'$ such that $\|\mathcal{D}-\mathcal{D}'\|_{\ell^\infty} = \frac{1}{2}\epsilon(\mathcal{T})$ and $f(\mathcal{D}') \neq f(\mathcal{D})$ [37]. However, the set of distance matrices leading to the same additive tree according to a given algorithm is a more complex set [12].

Operationally, one may not know a priori whether a given distance matrix $\mathcal{D}'$ is close to being additive or not. But applying ADDTREE or NJ [29], we can obtain an additive tree $\mathcal{T} = f(\mathcal{D}')$, and we can form the corresponding additive matrix $\mathcal{D}$ that matches this tree. If $\|\mathcal{D}-\mathcal{D}'\|_{\ell^\infty} < \frac{1}{2}\epsilon(\mathcal{T})$ then we can have good confidence that this tree well represents the data $\mathcal{D}'$. There are also external measures to determine which trees should be picked to represent non-additive distance matrices [10, 31].

In many cases, what is of most interest is not the lengths of the edges in the tree, but rather just the topology of the tree [4]. We now consider one simple case in which we can understand this approximation problem in some detail.

6 The ABC theorem

It is possible to show that the topology of the tree representation of four points is essentially unique under very mild conditions, as follows. Consider the three independent quantities that figure in the four point condition:

$$\begin{aligned} A &= \mathcal{D}(i,m) + \mathcal{D}(j,n) \\ B &= \mathcal{D}(i,n) + \mathcal{D}(j,m) \\ C &= \mathcal{D}(i,j) + \mathcal{D}(n,m) \end{aligned} \tag{44}$$

based on the three ways to partition the index set $\{i,j,m,n\}$ into distinct pairs. These quantities determine the topology of the tree representations, as follows. There are four distinct cases. Three of them involve two internal nodes and one internal edge, and are categorized by the following three distinct possibilities for additive matrices: $A = B > C$, $B = C > A$, and $C = A > B$. The fourth tree corresponds to $A = B = C$. Note that when $A = B = C$, the tree representing the distance matrix is a star. That is, there is one internal node k, and four edges joining the four indices to k.

We will show that even in the case that a matrix is not additive, a unique assignment of one of these topology classes is possible in most cases.

Suppose $\mathcal{D}$ is a general distance matrix that is not necessarily additive. Without loss of generality, by renaming the indices if necessary, we can assume that the terms are ordered:

$$A \geq B \geq C. \tag{45}$$

The four-point condition can now be stated simply: $A = B$. In this case, the distance matrix can be represented exactly by a tree. Now we consider the other case, that $A > B$. First, we define the ℓ^1-norm for distance matrices:

$$\|\mathcal{D}\|_{\ell^1} = \sum_{i<j} |\mathcal{D}(i,j)| \tag{46}$$

Note that we allow for negative entries, as we intend to apply the norm to differences of distance matrices.

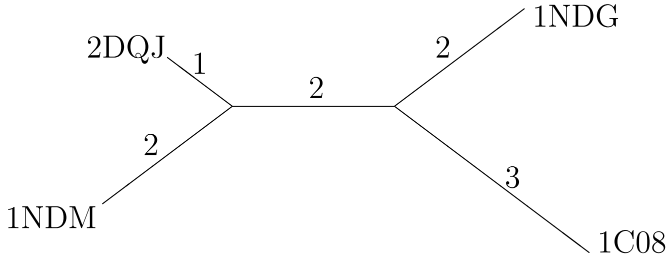

Figure 5: Tree representation of the distance matrix in Table 2.

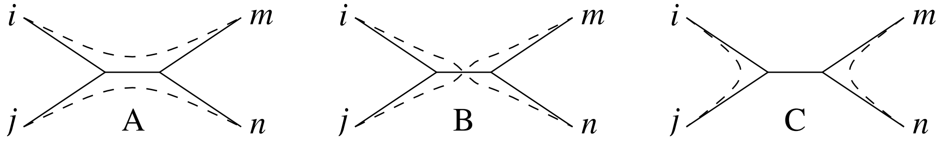

Figure 6: Relationship between the values A, B and C and the corresponding additive tree in the case that $C < \min\{A, B\}$. The dashed lines indicate the distances that correspond to A, B and C.

Theorem 6.1 *Suppose that $A > B$. Then*

$$\inf\left\{\|\mathcal{D} - \mathcal{D}'\|_{\ell^1} \;\middle|\; \mathcal{D}' \textit{ satisfies the four-point condition}\right\} = A - B. \tag{47}$$

Moreover, if $B > C$, then all additive distance matrices $\mathcal{D}'$ which satisfy

$$\|\mathcal{D} - \mathcal{D}'\|_{\ell^1} = A - B \tag{48}$$

have trees with the same topology, cf. Figure 6.

First of all, we note that there can be multiple 'best' distance matrices $\mathcal{D}'$ in (47). That is, we cannot define the 'best' tree in a simple way by just minimizing the ℓ^1 distance. However, we do assert in the second part of the theorem that the topology of all these trees is the same, provided that $B > C$.

When $B = C$, there is an ambiguity in representing $\mathcal{D}$ since there are additive matrices $\mathcal{D}'$ all equally close in ℓ^1 norm with different topology types. We leave as an exercise that there is a matrix $\mathcal{D}'$ with $A' = B' = C' = B = C$, as well as two others: one with $A' = B' = A$ and $C' = B = C$ and the other with with $A' = C' = A$ and $B' = B = C$.

Proof. To prove these assertions, we first show that

$$\inf\left\{\|\mathcal{D} - \mathcal{D}'\|_{\ell^1} \;\middle|\; \mathcal{D}' \text{ satisfies four-point condition}\right\} \le A - B. \tag{49}$$

To so so, we simply need to exhibit a $\mathcal{D}'$ which satisfies the four-point condition and $\|\mathcal{D}-\mathcal{D}'\|_{\ell^1} = A - B$. We can do this if we keep $A' = A$ and increase B' to be equal to A. For example, we can set

$$\mathcal{D}'_{in} = \mathcal{D}_{in} + A - B, \tag{50}$$

leaving all other entries of $\mathcal{D}'$ the same as for $\mathcal{D}$. Thus by explicit construction, we have $\|\mathcal{D} - \mathcal{D}'\|_{\ell^1} = A - B$. Similarly, since we also have $A' = A = B'$, $\mathcal{D}'$ satisfies the four point condition.

Now we demonstrate the other inequality. We can clearly write

$$|A - A'| + |B - B'| + |C - C'| \leq \|\mathcal{D} - \mathcal{D}'\|_{\ell^1} \tag{51}$$

for any distance matrices. Now suppose it were the case that for some $\mathcal{D}'$ we have $\|\mathcal{D} - \mathcal{D}'\|_{\ell^1} < A - B$. Then by (51) we have

$$\begin{aligned} B' - A' + A - B =& A - A' + B' - B \\ \leq& \|\mathcal{D} - \mathcal{D}'\|_{\ell^1} < A - B \end{aligned} \tag{52}$$

from which we conclude that $B' < A'$, so $\mathcal{D}'$ cannot satisfy the four point condition. This completes the proof of the equality (47).

Now suppose that $\mathcal{D}'$ is additive and satisfies (48). Then we want to show that

$$A \geq A' = B' \geq B. \tag{53}$$

Suppose that $A < B'$. Then $B' - B > A - B$ and applying (51) we find that $\|\mathcal{D} - \mathcal{D}'\|_{\ell^1} > A - B$. On the other hand, if $A' < B$ then $A - A' > A - B$, and again (51) yields a contradiction. Thus (53) has to hold.

Applying (53) in (51), we get

$$\begin{aligned} A - B =& A - A' + B' - B \\ =& |A - A'| + |B - B'| \\ \leq& |A - A'| + |B - B'| + |C - C'| \\ \leq& \|\mathcal{D} - \mathcal{D}'\|_{\ell^1} = A - B \end{aligned} \tag{54}$$

which means that equality holds throughout the expression (54), so we must have $C = C'$. In particular, we conclude that $A' = B' \geq C'$. If $B > C$, then we also have $B' > C'$.

Now it is easy to show that all additive matrices with $A' = B' > C'$ have the same topology. **QED**

7 Approximate algorithms

Since we see that there is a unique topology for trees representing generic four-point metric spaces, it is of interest to ask what various approximation algorithms compute in this simple case. We will see that neighbor joining always yields a tree with the correct topology. However, we will see that UPGMA does not, even in the case of an additive distance matrix.

7.1 What neighbor joining does

Neighbor joining chooses vertices based on the expression (40). So it makes sense to relate those quantities to the determinants of topology. We use the notation of Section 6, and in particular that of Figure 6. Noting that $n = 4$, we can use the second form of (40) to see that

$$Q_{ij} = Q_{mn} = -(\mathcal{D}(i,m) + \mathcal{D}(i,n)) - (\mathcal{D}(j,m) + \mathcal{D}(j,n)) = -(A + B). \tag{55}$$

Similarly, it is easy to compute that

$$Q_{im} = Q_{jn} = -(B + C) \qquad \text{and} \qquad Q_{in} = Q_{jm} = -(A + C). \tag{56}$$

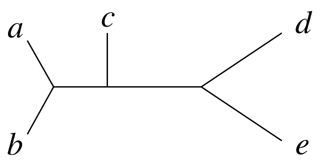

Figure 7: Tree for additive distance matrix $\mathcal{D}$ in (57) [33].

When $C \leq \min\{A, B\}$, then the indices that minimize the expression (40) are (i, j) and (m, n). Thus neighbor joining will choose the tree shown in Figure 6.

It is also clear that ADDTREE will also give the same tree, since it is based on ordering A, B and C. However, ADDTREE and neighbor joining will not give the same results [33] with five points. Consider the data in (57); $\mathcal{D}$ is additive, and $\mathcal{D}'$ is not:

$$\mathcal{D} = \begin{pmatrix} \begin{array}{ccccc} & b & c & d & e \\ a & 2 & 3 & 6 & 6 \\ b & & 3 & 6 & 6 \\ c & & & 5 & 5 \\ d & & & & 4 \end{array} \end{pmatrix} \qquad \mathcal{D}' = \begin{pmatrix} \begin{array}{ccccc} & b & c & d & e \\ a & 2 & 3 & 6 & \mathbf{3} \\ b & & 3 & 6 & 6 \\ c & & & 5 & 5 \\ d & & & & 4 \end{array} \end{pmatrix}. \tag{57}$$

The only difference between $\mathcal{D}$ and $\mathcal{D}'$ is that $\mathcal{D}(a, e) \neq \mathcal{D}'(a, e)$, as shown in bold in (57). The resulting tree for $\mathcal{D}$ is depicted in Figure 7.

The Q matrices (40) for $\mathcal{D}$ and $\mathcal{D}'$ in (57) are

$$Q(\mathcal{D}) = \begin{pmatrix} \begin{array}{ccccc} & b & c & d & e \\ a & -25 & -21 & -17 & -23 \\ b & & -24 & -20 & -17 \\ c & & & -22 & -19 \\ d & & & & \mathbf{-27} \end{array} \end{pmatrix} \qquad Q(\mathcal{D}') = \begin{pmatrix} \begin{array}{ccccc} & b & c & d & e \\ a & -28 & -24 & -20 & -20 \\ b & & -24 & -20 & -20 \\ c & & & -22 & -22 \\ d & & & & \mathbf{-30} \end{array} \end{pmatrix}, \tag{58}$$

indicating the same choice for coalescence of neighbors, d and e. The R matrices for $\mathcal{D}$ and $\mathcal{D}'$ in (39) are

$$R(\mathcal{D}) = \begin{pmatrix} \begin{array}{ccccc} & b & c & d & e \\ a & \mathbf{0} & 2 & 3 & 3 \\ b & & 2 & 3 & 3 \\ c & & & 2 & 2 \\ d & & & & \mathbf{0} \end{array} \end{pmatrix} \qquad R(\mathcal{D}') = \begin{pmatrix} \begin{array}{ccccc} & b & c & d & e \\ a & 1 & 3 & 5 & 2 \\ b & & 1 & 3 & 6 \\ c & & & 2 & 4 \\ d & & & & \mathbf{0} \end{array} \end{pmatrix}. \tag{59}$$

Note that there are two choices for $\mathcal{D}$ for coalescence of neighbors, (a, b) and (d, e).

7.2 What UPGMA does

The UPGMA algorithm [24] applied to a distance matrix will coalesce the closest points, that is, the two for which the entry in the distance matrix is smallest, say $\mathcal{D}(i, j)$. One might hope that UPGMA would find the correct tree for an additive matrix. But this is not so.

The nearest neighbors in the tree for an additive matrix are the indices that combine to form the term C that is the smallest of the three terms involved in the four point condition: $C < B = A$. However, even though it may hold that $\mathcal{D}(i, j)$ is the smallest entry in the distance matrix, $C = \mathcal{D}(i, j) + \mathcal{D}(m, n)$ is not smaller than A or B. In this case, UPGMA finds the wrong

	b	c	d
a	3	5	4
b		4	5
c			7

Table 3: Additive distance matrix for which UPGMA gives the wrong tree.

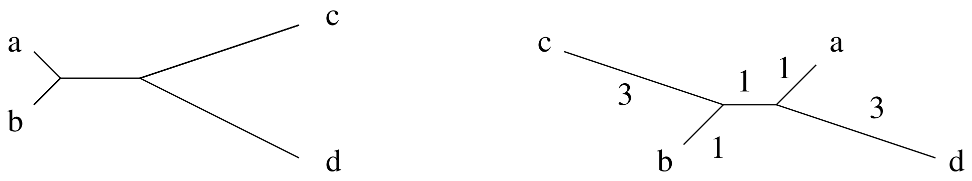

Figure 8: UPGMA tree (left) and additive tree for distance matrix in Table 3.

tree. For example, consider the distance matrix in Table 3 with $A = 5 + 5 = 10$, $B = 4 + 4 = 8$, and $C = 3 + 7 = 10$. In Figure 8 we see both the tree that UPGMA will generate (left) for the data together with the tree (right) that precisely represents this additive matrix.

8 Conclusions

We have presented various tree representations of data, drawn from protein biophysics, as a way to motivate diverse ways in which this data analytics tool is used to understand relationships. We reviewed the relationship between a metric for data relationships and corresponding tree representations, including giving some details regarding several standard algorithms. We suggested a new approach based on topological features of trees, with the ABC Theorem being the central new idea providing possible realization of this approach. This topological approach to trees is a new data analytics tool for data science. It resolves a conundrum regarding the relationship between data metrics and corresponding tree representations caused by the four-point condition.

9 Exercises

Exercise 9.1 *Show that the distance matrices in Table 1, Table 2, and Table 3 satisfy the triangle inequality.*

Exercise 9.2 *Prove that any distance matrix (2) must satisfy* $|b - a| \leq c$ *provided that* $c \leq \min\{a, b\}$. *(Hint: apply the triangle inequality (1).)*

Exercise 9.3 *Suppose that a distance matrix (2) satisfies (3) and (4). Prove that it satisfies the triangle inequality (1). (Hint: reverse the derivation in Exercise 9.4.)*

Exercise 9.4 *Suppose that a distance matrix (2) satisfies* $c \leq \min\{a, b\}$. *Sketch the cone of values of* a, b, *and* c *that satisfy the triangle inequality (1). (Hint: apply Exercise 9.2 and use the fact that* $a \geq c$ *and* $b \geq c$.*)*

Exercise 9.5 *Determine the set of the (six) allowable values for a distance matrix for a four-element metric space (cf. Exercise 9.4).*

Exercise 9.6 *Does that the matrix*

$$\begin{pmatrix} 0 & 4 & 2 \\ 4 & 0 & 1 \\ 2 & 1 & 0 \end{pmatrix} \tag{60}$$

satisfy the triangle inequality (1)? (Hint: see Exercise 9.2.)

Exercise 9.7 *PAM and Blossum matrices attempt to encode evolutionary distance between sequence elements. Examine these to see if they satisfy the triangle inequality (1).*

Exercise 9.8 *Prove that (15) defines a metric on stings of length n, provided that D_Σ is a metric on the alphabet Σ.*

Exercise 9.9 *Compute the Q matrix for the data in Table 1.*

Exercise 9.10 *Prove (35).*

Exercise 9.11 *Show that the matrices $P(\mathcal{D})$ and $R(\mathcal{D})$ defined in (38) and (39) satisfy*

$$P(i,j) + R(i,j) = c$$

for all $i,j = 1,\ldots,N$, where c is a constant depending only on the size N of the distance matrix $\mathcal{D}$. Determine the value of the constant c as a function of N.

References

[1] K. Atteson. The performance of neighbor-joining methods of phylogenetic reconstruction. *Algorithmica*, 25:251–278, Jun 1999. 10.1007/PL00008277.

[2] Kevin Atteson. *The performance of neighbor-joining algorithms of phylogeny reconstruction*, pages 101–110. LNCS Volume 1276. Springer, 1997. 10.1007/BFb0045077.

[3] Y.-E. A. Ban, H. Edelsbrunner, and J. Rudolph. Interface surfaces for protein-protein complexes. In *Proceedings of the 8th Annual International Conference on Research in Computational Molecular Biology (RECOMB)*, pages 205–212, 2004.

[4] Hans-Jurgen Bandelt and Andreas Dress. Reconstructing the shape of a tree from observed dissimilarity data. *Advances in Applied Mathematics*, 7:309–343, Sep 1986.

[5] Jean-Pierre Barthelemy and Alain Guenoche. *Trees and Proximity Representations*. John Wiley & Sons, New York, 1991.

[6] Arturo Becerra, Luis Delaye, Sara Islas, and Antonio Lazcano. The very early stages of biological evolution and the nature of the last common ancestor of the three major cell domains. *Annual Review of Ecology, Evolution, and Systematics*, 38(1):361–379, 2007.

[7] David Bryant. On the uniqueness of the selection criterion in neighbor-joining. *Journal of Classification*, 22:3–15, Jun 2005. 10.1007/s00357-005-0003-x.

[8] Jianping Chen, Xi Zhang, and Ariel Fernández. Molecular basis for specificity in the druggable kinome: sequence-based analysis. *Bioinformatics*, 23(5):563–572, 2007.

[9] Francesca D. Ciccarelli, Tobias Doerks, Christian von Mering, Christopher J. Creevey, Berend Snel, and Peer Bork. Toward automatic reconstruction of a highly resolved tree of life. *Science*, 311(5765):1283–1287, 2006.

[10] Richard Desper and Olivier Gascuel. The minimum evolution distance-based approach to phylogenetic inference. In Olivier Gascuel, editor, *Mathematics of evolution and phylogeny*, pages 1–32. World Scientific, 2005.

[11] Christopher M. Dobson. Protein folding and misfolding. *Nature*, 426:884–890, 2003.

[12] Kord Eickmeyer, Peter Huggins, Lior Pachter, and Ruriko Yoshida. On the optimality of the neighbor-joining algorithm. *Algorithms for Molecular Biology*, 3(1):5, 2008.

[13] Isaac Elias and Jens Lagergren. *Fast Neighbor Joining*, pages 1263–1274. LNCS Volume 3580. Springer, 2005. 10.1007/11523468_102.

[14] Miles A. Fabian, William H. Biggs, Daniel K. Treiber, Corey E. Atteridge, Mihai D. Azimioara, Michael G. Benedetti, Todd A. Carter, Pietro Ciceri, Philip T. Edeen, Mark Floyd, Julia M. Ford, Margaret Galvin, Jay L. Gerlach, Robert M. Grotzfeld, Sanna Herrgard, Darren E. Insko, Michael A. Insko, Andiliy G. Lai, Jean-Michel Lelias, Shamal A. Mehta, Zdravko V. Milanov, Anne Marie Velasco, Lisa M. Wodicka, Hitesh K. Patel, Patrick P. Zarrinkar, and David J. Lockhart. A small molecule-kinase interaction map for clinical kinase inhibitors. *Nature Biotechnology*, 23(3):329–336, 2005.

[15] Marcus Fandrich and Christopher M. Dobson. The behaviour of polyamino acids reveals an inverse side chain effect in amyloid structure formation. *EMBO J.*, 21(21):5682–5690, 2002.

[16] Joseph Felsenstein. *Inferring Phylogenies.* Sinauer Associates, second edition, 2003.

[17] Ariel Fernández and Sridhar Maddipati. A priori inference of cross reactivity for drug-targeted kinases. *Journal of Medicinal Chemistry*, 49(11):3092–3100, 2006.

[18] Ariel Fernández and L. Ridgway Scott. Dehydron: a structurally encoded signal for protein interaction. *Biophysical Journal*, 85:1914–1928, 2003.

[19] Ariel Fernández and L. Ridgway Scott. Modulating drug impact by wrapping target proteins. *Expert Opinion on Drug Discovery*, 2:249–259, 2007.

[20] Anton F. Fliri, William T. Loging, Peter F. Thadeio, and Robert A. Volkmann. Biological spectra analysis: Linking biological activity profiles to molecular structure. *Proceedings of the National Academy of Sciences, USA*, 102(2):261–266, 2005.

[21] Christopher M. Fraser and L. Ridgway Scott. Candidate dehydron identification in high resolution myoglobin structures. Research Report UC/CS TR-2015-12, Dept. Comp. Sci., Univ. Chicago, 2015.

[22] O Gascuel. A note on Sattath and Tversky's, Saitou and Nei's, and Studier and Keppler's algorithms for inferring phylogenies from evolutionary distances. *Mol Biol Evol*, 11(6):961–963, 1994.

[23] Olivier Gascuel and Mike Steel. Neighbor-joining revealed. *Mol Biol Evol*, 23(11):1997–2000, 2006.

[24] Dan Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology.* Cambridge University Press, January 1997.

[25] K. C. Klauer and J. D. Carroll. A mathematical programming approach to fitting general graphs. *Journal of Classification*, 6:247–270, Dec 1989. 10.1007/BF01908602.

[26] Karl Klauer. Ordinal network representation: Representing proximities by graphs. *Psychometrika*, 54:737–750, Sep 1989. 10.1007/BF02296406.

[27] C. R. Linder, B. M. E. Moret, L. Nakhleh, and T. Warnow. Network (reticulate) evolution: biology, models, and algorithms. In Russ B. Altman, A. Keith Dunker, Lawrence Hunter, Tiffany A. Jung, and Teri E. Klein, editors, *Biocomputing 2004, Proceedings of the 9th Pacific Symposium on Biocomputing, Hawaii, USA, 6–10 January 2004*. World Scientific, 2004.

[28] S. Maddipati and A. Fernández. Feature-similarity protein classifier as a ligand engineering tool. *Biomolecular engineering*, 23(6):307–315, 2006.

[29] Vladimir Makarenkov. T-REX: reconstructing and visualizing phylogenetic trees and reticulation networks. *Bioinformatics*, 17(7):664–668, 2001.

[30] G. Manning, D. B. Whyte, R. Martinez, T. Hunter, and S. Sudarsanam. The protein kinase complement of the human genome. *Science*, 298(5600):1912–1934, 2002.

[31] Peter McCullagh. Marginal likelihood for distance matrices. *Statistica Sinica*, 19:631–650, 2009.

[32] Radu Mihaescu, Dan Levy, and Lior Pachter. Why neighbor-joining works. *Algorithmica*, 2008. 10.1007/s00453-007-9116-4.

[33] Radu Horia Mihaescu. *Distance Methods for Phylogeny Reconstruction.* PhD thesis, UC Berkeley, 1996.

[34] D. W. Mount. *Bioinformatics.* Cold Spring Harbor Laboratory Press, 2001.

[35] Sandra Pruzansky, Amos Tversky, and J. Carroll. Spatial versus tree representations of proximity data. *Psychometrika*, 47:3–24, Mar 1982. 10.1007/BF02293848.

[36] N. Saitou and M. Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol Biol Evol*, 4(4):406–425, 1987.

[37] Shmuel Sattath and Amos Tversky. Additive similarity trees. *Psychometrika*, 42:319–345, Sep 1977. 10.1007/BF02293654.

[38] L. Ridgway Scott and Ariel Fernández. *A Mathematical Approach to Protein Biophysics.* Springer-Verlag, 2017.

[39] Michael Waterman. *Introduction to Computational Biology.* Chapman & Hall/CRC Press, 1995.

[40] Stephen Willson. Unique reconstruction of tree-like phylogenetic networks from distances between leaves. *Bulletin of Mathematical Biology*, 68:919–944, May 2006. 10.1007/s11538-005-9044-x.

[41] Yuri I. Wolf, Igor B. Rogozin, Nick V. Grishin, and Eugene V. Koonin. Genome trees and the tree of life. *Trends in Genetics*, 18(9):472 – 479, 2002.

[42] Anne D. Yoder and Ziheng Yang. Estimation of primate speciation dates using local molecular clocks. *Mol Biol Evol*, 17(7):1081–1090, 2000.

[43] Natalya Yutin, Kira S. Makarova, Sergey L. Mekhedov, Yuri I. Wolf, and Eugene V. Koonin. The deep archaeal roots of eukaryotes. *Mol Biol Evol*, 25(8):1619–1630, 2008.