ELSEVIER

Discrete Applied Mathematics 71 (1996) 297–309

DISCRETE APPLIED MATHEMATICS

Alignment networks and electrical networks

Martin Vingron [a,*], Michael S. Waterman [b]

[a] *GMD- German National Research Center for Information Technology, Institute for Algorithms and Scientific Computing, D-53757 St. Augustin, Germany*

[b] *Department of Mathematics, University of Southern CA, Los Angeles, CA 90089-1113, USA*

Received 26 May 1995; revised 8 May 1996; accepted 20 May 1996

Abstract

In the analysis of biological sequences there arises the question of attributing weights to each element of a set of objects in such a way that similar objects do not overly influence deductions based on the entire set of objects. Since there is no known precise statement of this problem, we introduce two axioms that these weights should fulfill. It is then easy to see that this formulation includes one commonly used method. Further we apply the axiomatic framework to the new problem of attributing weights to alignments between two sequences. Viewing a sequence alignment as a directed network, an analogy to electrical networks is developed. This connection is used to prove the existence of the weights and develop other characteristics of an alignment network.

Keywords: Sequence alignment; Weighting methods; Electrical networks; Least squares estimation

1. Introduction

In several contexts in molecular biology the problem arises of attributing weights to species, sequences or other objects [3]. However, the given objects may not be independent descriptors of what one seeks to describe and thus a set of objects is not as informative as it would seem. One example of this situation arises in the problem of attributing weights to sequences in a multiple sequence alignment. The researcher's idea is that each sequence provides information about the residues that are admissible in a position of the sequence. At the same time, some of the sequences are closely related (in evolution) and therefore cannot be viewed as independent pieces of information. This problem is reviewed by Vingron and Sibbald [10].

Another problem of this type is the weighting of alignment paths in cases where a set of alignments is being considered. For our purpose, we are given a list of possible alignments, without further distinction between them, and we believe that the correct

* Corresponding author. Present address: Deutsches Krebsforschungszentrum, Abt. Theoretische Bioinformatik, INF 280, D-69120 Heidelberg, Germany

0166-218X/96/$15.00 © 1996 Elsevier Science B.V. All rights reserved
PII S0166-218X(96)00070-4

solution is an alignment from the list. How much do we then know about a single residue pair? A residue pair that is not part of any alignment will not be part of the correct alignment, while a residue pair that is contained in all alignments must be correct. We are led to the idea of assigning weights to the residue pairs such that the weight reflects the "confidence" that we have in a residue pair based on its representation in the alignment paths. But conflicting forces have to be considered: When many alignment paths contain a residue pair it should raise our confidence in this pair. If these alignments are all very similar, however, they should not have as much importance as totally distinct alignments that share only this one residue pair. Fig. 1 shows some situations that illustrate the problems that arise. A different viewpoint on this issue resulting in an alternative formalization is presented in [5].

In this paper, we will first give a formulation of the general problem of attributing weights to overlapping sets and their elements. In the form given, it applies to the above cases and possibly to several more in the realm of biological applications. Furthermore, we will make precise the connection to electrical circuits and linear least squares that has been mentioned in the literature [1, 3]. Application to the weighting of alignment paths will be discussed in detail.

2. Axioms for point-set-weighting

Suppose we are given a finite set $X = \{x_1, x_2, \ldots, x_n\}$ and a finite family $\mathscr{B} = \{B_i\}_{i=1..N}$ of subsets of X. Weights on the elements of X can be written as a vector $v \in \mathbb{R}^n$ and weights on the sets as $w \in \mathbb{R}^N$. We consider first a weight vector v on the points x_i such that the sum over the weights of the points in a set is equal (to some constant c, say) for all sets.

Axiom 1.

$$\sum_{i:x_i \in B_k} v_i = c \quad \text{for all} \quad k = 1, \ldots, N.$$

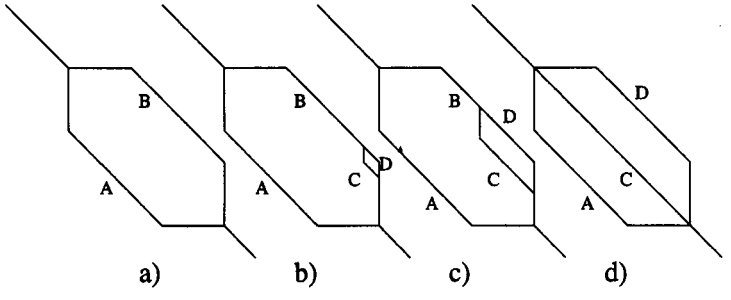

Fig. 1. Weights on paths should reflect the overlap between them. While the weights for the two paths in (a) should be equal, there should be only a tiny change in (b), gradually leading to equal weights for all three paths in (d).

While the sum over the point-weights in a set is constant, the sets will also have associated weights $w = (w_i)_{i=1,\ldots,N}$. Given the vector v, we define w by requiring that the weight of a point is expressible as the sum of the weights w_i of the sets of which it is an element.

Axiom 2 (Additivity).

$$v_i = \sum_{k : x_i \in B_k} w_k \quad \text{for all} \quad i = 1, \ldots, n.$$

The weight of a point is thus inherited from the sets of which this point is a member.

We abbreviate the N-element column vector that contains only 1 as $\mathbf{1}$. Let $M = (m_{ij})_{i,j=1\ldots N}$ be the matrix defined by $m_{kl} = |B_k \cap B_l|$. One can think of M as an "overlap matrix" that counts the number of points which two sets share.

Theorem 1. *Assume M is non-singular. Then the above weights satisfy*

$$w = cM^{-1} \cdot \mathbf{1}$$

with v given by the axiom of additivity.

Proof. Let A be the $N \times n$ matrix which contains a 1 at position (k, l) whenever $x_l \in B_k$ and 0 otherwise. Axiom 2 then may be written as $v = A^t w$. Axiom 1 becomes $Av = c\mathbf{1}$. Together this implies $AA^t w = c\mathbf{1}$. It is easy to see that $AA^t = M$: One can think of A as the adjacency matrix of a bipartite graph. One component of this bipartite graph is the family $\mathscr{B}$ and the other component is X. There is an edge in this graph when a set B_k contains a point x_l. Entry (i, j) of the product AA^t then counts the number of paths that go from B_i first to some point and then on to a set B_j. The number of such paths is exactly the cardinality of $B_i \cap B_j$ which is m_{ij}. Therefore $Mw = c\mathbf{1}$ and $w = cM^{-1} \cdot \mathbf{1}$. □

We adopt the convention that c should be chosen such that $\sum_{i=1}^{N} w_i = 1$. The proof also shows that M is positive semidefinite because it is the product of a matrix A and its transpose. Furthermore, it will be positive definite, and thus invertible, when A has maximal rank. When the rank of A is not maximal, there may be either a manifold of solutions or no solution. For the application we give below, it will be shown that there always exists a solution. In any case, a convenient way to calculate w according to Theorem 1 is by singular value decomposition [6].

3. Two applications

3.1. Alignment networks I

Assume now we are given an "alignment network", by which we mean a directed graph representing a set of alignments in an edit graph comparing two sequences

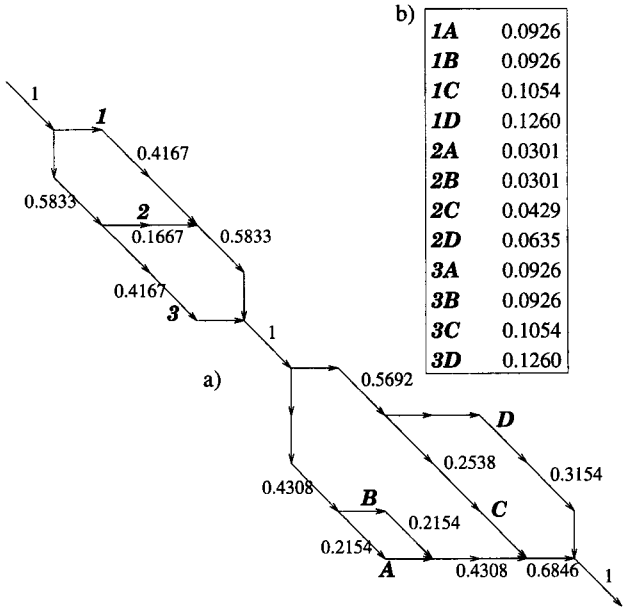

Fig. 2. (a) An alignment network with edge-weights. (b) Paths (1A–3D) are identified by the region through which they pass. Next to the path identification the path-weight is given.

(Fig. 2, for an introduction to this view of sequence alignment see [2]). In this paper we show paths in edit graphs; the alignments themselves are implicit. We assume all alignments are global, i.e. the paths link a source with a sink. We want to assign weights to the alignment-paths that reflect how "unique" a path is (i.e. how little overlap with other paths it has), and we want to assign weights to the edges in the network. An edge inherits the weights from the paths it is on, and the edge weight is equal to the sum of these path weights. Considering a simple example like the one in Fig. 1a, we would give each of the two possible alignment paths a weight of 0.5 ($\boldsymbol{w}^t = (0.5, 0.5)$) and assign the edges where the paths coincide a weight of 1 ($v_i = 1$), while the other edges receive a weight of 0.5 ($v_j = 0.5$). This might suggest a general rule to attribute to an edge the number of alignment paths using that edge divided by the number of paths in the network. The remainder of Fig. 1 shows that this rule would fail. In Fig. 1(b), the path passing through region B is split into two by a small deviation, creating two regions C and D. Clearly this should not have much effect on the weights of the edges in region B which should still be close to 0.5. However, the edges in region B now receive $\frac{2}{3}$ as their weight because two out of three paths pass through those edges. Edges in regions C and D would receive $\frac{1}{3}$ each. What we wanted to achieve, however, was that as region B shrinks and C and D increase (Figs. 1(c) and (d)), the weights on edges in C and D should approach $\frac{1}{3}$.

The example in Fig. 1 shows that the degree of overlap between alignment paths must be taken into account. With paths corresponding to sets and edges to elements of

the sets, we can employ the point-set-weighting of the previous section. By counting the overlaps between the paths in terms of the number of edges they share, we obtain the matrix M. We can apply the scheme described in Section 2 and obtain a vector $\boldsymbol{w} = cM^{-1} \cdot \mathbf{1}$ of weights for the alignment paths. Weights $\boldsymbol{v}$ for the edges then are the sum of the path-weights of the paths going through an edge. Applying this scheme to the situation of Fig. 1 does indeed result in weights which adequately reflect our intuition (as one can readily check with paper and pencil). Fig. 2 shows a less schematic application.

One consequence of the scheme described does not conform with the intuition about alignment weighting: Two paths which do not overlap but use different numbers of edges receive unequal path weights. We will return to this phenomenon in Section 5.

3.2. Weighting of sequences related by a tree

We now show that the scheme of Section 2 can also be used to explain a commonly used method of attributing weights to sequences related by a rooted evolutionary tree due to Altschul et al. [1]. The problem there is to attribute weights to the labels of a rooted binary tree that has lengths associated to its edges. The method of Altschul et al. introduces a matrix that they call a covariance matrix and that is analogous to the overlap matrix. It measures the sum of the length of the tree edges shared by paths connecting two different species to the root of the tree. Weights are then assigned as the row-sums of the inverse of the covariance matrix. When one thinks of every edge in the tree as made up of points and the branches as sets containing the points of the edges, then the analogy to the overlap matrix becomes clear. The weighting scheme by Altschul et al. can thus be interpreted in the framework of Section 2.

Altschul et al. name two analogous frameworks that yield the same weighting scheme: electrical networks and least squares estimation. We will discuss both these topics and prove the existence of weights for alignment networks.

4. Electrical networks

We proceed by translating an alignment network into an electrical network. Consider an electrical network with the same topology as the given alignment network. Every edge in the edit graph corresponds to a wire. For the moment, we set the resistance of each wire to 1. We show that if we attach a unit current to this network, the currents in the wires are exactly the weights $\boldsymbol{v}$ on the edges. This will subsequently be used to demonstrate the existence of weights $\boldsymbol{w}$ on the paths.

We briefly summarize some facts about directed graphs and electrical networks as given e.g. by Bollobas [4]. The cycles of the underlying undirected graph of a directed graph can be written as vectors with as many entries as there are edges. Edges that

are not used in the cycle correspond to a 0 entry, edges that are traversed in their proper orientation are $+1$, others are -1. These "cycle-vectors" form a vector space, the cycle space. A convenient basis for the cycle space can be constructed from a spanning tree of the oriented network. Every chord (an edge of the graph that is not part of the spanning tree) then closes exactly one cycle using edges from the tree and this one chord. Such a cycle is called a fundamental cycle. The vectors associated to the fundamental cycles form a basis of the cycle space. By counting the chords, one obtains the dimension of the cycle space: The spanning tree has #{vertices} $-$ 1 edges. Thus, the number of chords is #{edges in the graph} $-$ #{vertices} $+$ 1, which is the dimension of the cycle space. Recall also Kirchoff's laws: the sum of incoming and outgoing currents in a vertex is 0 (current law) and the sum of the potential differences along a cycle is 0 (voltage law). Ohm's law says that the potential difference along an edge is the product of current and resistance in that edge.

There appears to be a problem in the correspondence between alignment networks and electrical networks since paths have no natural meaning in an electrical network. However, given an alignment network D, we can define a new network $\tilde{D}$ which is the network D plus one additional edge from the sink to the source of the network. We will call this new edge the closing edge. Each path from the source to the sink of D now defines exactly one cycle in $\tilde{D}$, namely the cycle made up of the path and the closing edge. We will call the set of cycles created from the paths by closing the network in this way the closed paths. Since $\tilde{D}$ has one edge more than D, the dimension of its cycle space is one higher than the dimension of the cycle space of D. For later reference we summarize this in the following lemma. Let d be the dimension of the cycle space of D and $\tilde{d}$ be dimension of the cycle space of $\tilde{D}$

Lemma 1. $d = \#\{\textit{edges in } D\} - \#\{\textit{vertices}\} + 1.$

$$\tilde{d} = \#\{\textit{edges in } \tilde{D}\} - \#\{\textit{vertices}\} + 1 = 1 + d.$$

Lemma 2. *The closed paths span the cycle space of $\tilde{D}$.*

Proof. We will show that a fundamental cycle of $\tilde{D}$ can be written as the difference of two closed paths. Let e be the chord defining the fundamental cycle. Let a and b be the branches from the root of the tree to the beginning and end of e. a and b separate at a vertex x. Let us call a' the part of a from x to the beginning of e, and call b' the part of b from x to the end of e. The fundamental cycle contains a', e and b'. There also exists a closed path from the root along a and e to the sink, as well as one along b. The difference between these two paths is exactly the fundamental cycle containing e. □

Let A be the path–edge matrix of D: $a_{ij} = 1$ when path i contains edge j and 0 otherwise. Analogously we define the closed path–edge matrix of $\tilde{D}$ as $\tilde{A}$. Compared to A, $\tilde{A}$ has one additional column containing all 1s because the closing edge is contained in all closed paths. Notice that the set of edges in the network can be identified with

the set X from Section 2, and the set of paths through the network can be identified with the family $\mathscr{B}$ from Section 2. The path–edge matrix introduced now is therefore identical to the matrix A used in Section 2.

Lemma 3. *rank*$(\tilde{A}) = \tilde{d}$.

Proof. The rows of $\tilde{A}$ contain the vectors corresponding to the closed paths. From Lemma 2, these vectors span a vector space of dimension $\tilde{d}$. □

Lemma 4. *rank* $(A) =$ *rank*$(\tilde{A})$.

Proof. Select a set of edges in D that form a cut. Every path through the network contains exactly one edge from this cut. There correspond columns of A to these edges. The sum of these columns is the vector $\mathbf{1}^t$ and this is exactly the column of $\tilde{A}$ corresponding to the closing edge. Because this column is a linear combination of columns of A, the two matrices must have the same rank.

Corollary 1. *rank* $(M) = \tilde{d}$.

Proof. M is AA^t, and this product has the same rank as A. □

We now turn to the question of calculating the current in the electrical network when a unit current is attached at the closing edge of the network. Let m be the number of edges in D, n be the number of vertices in D, and N be the number of paths from the source to the sink of D. Identifying the set of paths with the family $\mathscr{B}$ from Section 2 and the edges with the set X, we can search for vectors $\boldsymbol{v} \in \mathbb{R}^m$ and $\boldsymbol{w} \in \mathbb{R}^N$ that fulfill Axioms 1 and 2. The following two theorems establish that knowing $\boldsymbol{v}$ and $\boldsymbol{w}$ corresponds to calculating the current through the electrical network.

Theorem 2. *If there exist vectors* $\boldsymbol{v} \in \mathbb{R}^m$ *and* $\boldsymbol{w} \in \mathbb{R}^N$ *fulfilling Axioms* 1 *and* 2, *then Kirchoff's laws hold for currents* $\boldsymbol{v}$ *on the edges of the network.*

Proof. We first show that Axiom 2 implies Kirchoff's current law. Axiom 2 says that the weights $\boldsymbol{w}$ of the paths add up to give the weights of edges. Therefore, when a set of edges enter a vertex, all incoming paths will also exit the vertex. This means that the sum of edge-weights leading into a vertex is the same as the sum of edge-weights coming out of a vertex. Thus, by Axiom 2, the edge-weights fulfill Kirchoff's current law. Axiom 1 says that the sum of edge-weights along a path from source to sink is constant. Under the assumption of unit resistance, this also means that the sum of potential differences along any path from source to sink is constant. As a consequence, the potential difference along any cycle is 0. □

To formulate the converse, we assume that we are given a network and attach a unit current to it. A current flow always exists (for a mathematical proof see [4]).

We make the currents on the edges the vector v. Kirchhoff's potential law implies that the potential difference between source and sink is the same along each path through the network. Thus Axiom 1 is fulfilled by the currents v. We need to show that given v there exists a vector w obeying Axiom 2, i.e. that solves the equation $A^t w = v$.

Theorem 3. *If v is the vector of currents on the edges of a network, $A^t w = v$ is solvable.*

Proof. Let $\tilde{v}$ be the vector of currents on $\tilde{D}$. $\tilde{v}$ is identical to v in all its entries except the one corresponding to the closing edge. This entry is 1 because we are attaching a unit current to the network. We show that $\tilde{A}^t w = \tilde{v}$ is solvable. From this it easily follows that the theorem holds. Let $\tilde{B}$ be the signed vertex–edge incidence matrix of the directed graph. The signed vertex–edge matrix of a directed graph has entries $+1$ (when an edge points to a vertex), -1 (when an edge leads out of a vertex) or 0 (when a vertex is not incident to an edge). $\tilde{B}\tilde{v}$ then will be a vector with all entries 0 (Kirchoff's current law). Viewing $\tilde{B}$ as a map $R^{m+1} \to R^n$, $\tilde{v} \in ker\,\tilde{B}$ due to Kirchhoff's current law. To prove the theorem we need to show that $ker\,\tilde{B} \subseteq Im\,\tilde{A}^t$. In fact equality holds: We show first that $Im\tilde{A}^t \subseteq ker\,\tilde{B}$. Let $x \in Im\,\tilde{A}^t$, i.e. $\exists y : \tilde{A}^t y = x$. Applying $\tilde{B}$ we get $\tilde{B}\tilde{A}^t y = \tilde{B}x$. However, $\tilde{B}\tilde{A}^t$ is a matrix with all entries 0 because an entry in $\tilde{B}\tilde{A}^t$ is the sum along a path of the edges incident to one vertex. And this sum is always 0, because there is one edge entering and one edge leaving a vertex. Therefore $x \in ker\tilde{B}$. To prove $ker\,\tilde{B} = Im\,\tilde{A}^t$ we show that $dim(Im\,\tilde{A}^t) = dim(ker\,\tilde{B})$. $dim(Im\,\tilde{A}^t)$ is $(m+1)-n+1$ from the Lemmas 1 and 3. To calculate $dim(Im\,\tilde{B})$, choose a spanning tree for the network $\tilde{D}$ and select those rows of $\tilde{B}$ which correspond to edges in the tree. This matrix has full rank [4], namely $n-1$ since the tree has $n-1$ edges. Thus $dim(Im\,B)$ is $n-1$ and from the equation $dim(Im\,B) + dim(ker\,B) = m+1$ it follows that $dim(ker\,B) = (m+1) - n + 1$. Therefore $ImA^t = kerB$. □

Apart from their existence, another attribute of the weights is of interest: Are they positive? This question turns out to be related to another question that arises from interpreting an alignment network as an electrical network. An alignment network is a directed graph and a path through this network is defined as obeying the directions of the edges. Electrical current, however, does not ask for permission to flow in a certain direction. Its direction is determined by the resistances. In fact, it is possible to design an alignment network that would result in negative current in one of its edges. Fig. 3 shows such an example. It can easily be checked that the current in the edge that is printed in bold flows in reverse direction. Albeit, this phenomenon does not contradict any of the above observations. The (directed) paths through the alignment network are only one of many ways to span the cycle space. Other paths, forbidden as alignments, might be used otherwise. At the same time it is easy to see that when assigning weights to species related by a tree, the weights will always be positive because the current has no alternative routes.

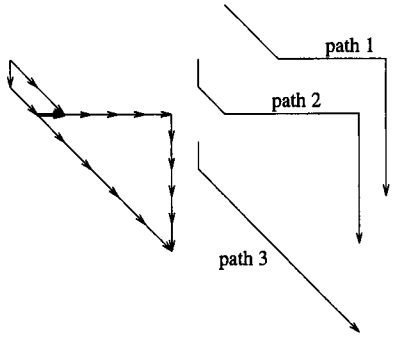

Fig. 3. A simple network that contains an edge where the current would flow against the direction of the alignment network. Paths 1 and 2 have a much higher resistance than path 3.

It is well known that an electrical network minimizes the energy dissipation ($\sum_{edges}$ current^2 * resistance, see [8]). The following observation will be referred to in the next section.

Lemma 5. *The energy dissipation of the electrical network is the normalizing constant c.*

Proof. $\sum v_i^2 = \langle v, v\rangle = \langle A^t w, A^t w\rangle = \langle w, AA^t w\rangle = \langle w, Mw\rangle = \langle w, c\mathbf{1}\rangle = c\sum w_i = c.$ □

5. Linear least squares

Consider a set of measurements from which we want to estimate the mean. When the measurements are independent normally distributed, the minimum variance linear unbiased estimator is the average of the measurements. If they are assumed to be correlated and this correlation described in the form of a variance–covariance matrix Σ, then the estimator is the weighted average. The weights are the normalized row-sums of the inverse of the variance-covariance matrix Σ^{-1}. Because the matrix M is non-negative definite, it can be interpreted as a variance-covariance matrix. The following observation connects the error that is minimized in linear least squares problem with $\Sigma = M$ to electrical networks.

Lemma 6. *The energy dissipation of the electrical network is the variance in a least squares problem.*

Proof. The quantity minimized in linear least squares is $E(e^t e)$ where e is the vector of errors. This can be rewritten as $\langle \mathbf{1}, M^{-1}\mathbf{1}\rangle^{-1}$ [8]. Since $w = cM^{-1}\mathbf{1}$ and $\sum w_i = 1$, we have $\langle \mathbf{1}, M^{-1}\mathbf{1}\rangle^{-1} = \langle \mathbf{1}, (1/c)w\rangle^{-1} = c$, which was shown to be the energy dissipation minimized by the currents through the electrical network. □

6. Alignment networks II

An additional modification of the model is required. So far we identified currents and potential differences by assuming that every edge had unit resistance. However, this also implies that a long alignment receives less weight than a shorter one. For two disjoint paths, one of which has more indels than the other, this disagrees with our intuition. The situation can be remedied by using non-identical resistances to make all global alignments equally long.

All our earlier observations hold also with resistances. The resistances can be summarized in a square matrix R that on its diagonal has the edge resistances and is 0 elsewhere. Recall that when one abbreviates the vector of all potential differences in the edges as $\boldsymbol{p}$, Ohm's law becomes $\boldsymbol{p} = R\boldsymbol{v}$. Axiom 1 is then rephrased as $AR\boldsymbol{v} = c\mathbf{1}$. Axiom 2 remains unchanged, namely $\boldsymbol{v} = A^{\mathrm{t}}\boldsymbol{w}$. The overlap matrix M becomes ARA^{t}, but with R a diagonal matrix ($r_{ii} \neq 0$) the rank of M is unchanged. It is easily checked that Theorems 2 and 3 still hold.

Theorem 4. *Assume that a network is made up of edge-disjoint paths. When one assigns resistance* 0.5 *to horizontal and vertical edges and resistance* 1 *to diagonal edges, all paths will receive equal weight.*

Proof. From the study of sequence alignments it is well known that for an alignment between two sequences of length l_1 and l_2 the following holds: $2 * \#\{\text{matches}\} + \#\{indels\} = l_1 + l_2$. This alignment-invariant is a special case of the invariant used in [7]. In a network on an edit graph, this means that if one assigns resistances of 0.5 to all horizontal or vertical edges (corresponding to indels) and a resistance of 1 to diagonal edges, the sum of the resistances along any path connecting two vertices will be constant. We need to show that all entries of $\boldsymbol{w} = cM^{-1}\mathbf{1}$ are equal. M is now ARA^{t} with r_{ii} either 1 or 0.5. Multiplying A^{t} with R makes the entries of A either 1 or 0.5, depending on the resistance of the corresponding edge. Multiplying again by A adds up the resistances along a path and since the paths are disjoint, the result will be a diagonal matrix. Due to the alignment invariant, all diagonal elements of this matrix are equal, namely to the sum of the resistances along a path. The inverse of M will therefore also be a diagonal matrix with constant main diagonal and all path-weights w_i are equal. □

Fig. 2(a) shows an example of a more complicated alignment network. Horizontal and vertical edges have resistance 0.5, diagonal edges have resistance 1. Note that due to the "bottleneck" in the middle of the graph, the matrix M is not invertible. Corollary 1 allows us to calculate the rank of M as 6: 35 edges + 1 edge to close all paths − 31 vertices +1. Due to the bottleneck, M is easily decomposed into two smaller matrices, one for the upper part and one for the lower part of the network. The weights for the paths are listed in Fig. 2(b) and the weights of the edges (the currents) are shown adjacent to the respective parts of the alignment network in Fig. 2(a).

With the above choice of resistances we tried without success to devise networks that contain edges where the current flows in the opposite direction to the alignment. Therefore we suspect that these resistances are sufficient to keep edge currents positive. An even more challenging question is whether there are necessary and sufficient criteria for the absence of negative edge weights.

7. Calculating the overlap matrix for sets of optimal alignments

The above analysis yields two ways to calculate the weights w and v. The first option is to calculate the currents in the network thus obtaining v. Solving the equation from Axiom 2 will then yield w. Alternatively, as pointed out at the end of Section 2, from Theorem 1 one can calculate w without knowing v. Axiom 2 then yields v. This procedure requires knowledge of the overlap matrix M. We thus proceed to give an algorithm to calculate the overlap matrix M for the set of all optimal alignments of two sequences. Naively, one would calculate all alignments and then compare them and count their agreement. M can, however, be constructed very efficiently upon backtracking through the dynamic programming matrix. We describe this for the unit resistance case.

Suppose the forward pass of calculating a sequence alignment has been performed. The algorithm to calculate the overlap matrix M needs to keep track of the numbering of the paths. We will start by explaining how to label the edges that are part of some optimal alignment with an array that contains pointers to path identifiers. Let the identifiers be an array of numbers, one for each path. The backtracking starts at the sink of the network. Label this vertex with a pointer to the first path. We do not follow each alignment path but examine each vertex in the order described next. We will proceed backwards through the edit graph from one anti-diagonal to the next. Suppose we are in an anti-diagonal. We start at one end and move towards the other end of the anti-diagonal, examining each vertex on the anti-diagonal. When inspecting the edges adjacent to a certain vertex, we first take the union of the labels of the incoming paths from the prior anti-diagonal. This gives us the set S of all labels of paths that pass through this vertex. Say there are k, $k \in \{1,2,3\}$, possible continuations of these paths, depending on how many edges are optimal in the backtrack from the vertex. The number of indices for paths must now be increased by $k-1$ times the cardinality of S, since each path must be divided into k paths. Set the label of the first of the k edges to the same indices as the incoming paths. The labels of the next possible edge are set to the next set of $|S|$ indices, etc. At the source of the network, this procedure has generated a set of labels that contains as many labels as there are paths through the network.

At the same time that one calculates the labels, it is possible to generate the overlap matrix M. Initially M is a 1×1 matrix with entry 0. As one changes the labels of the edges one changes the size of the matrix and updates it in in the following way: Suppose a subset S of the set T of paths generated up to this point of the computation

enters a vertex. For simplicity, think of the matrix M re-ordered in such a way that the paths in S form a consecutive set of rows and edges. Then the overlap-counts among the paths in S form a block of the matrix. We will call this block the $S \times S$-block. Add 1 to each entry in the $S \times S$ block. This represents extending the paths by the one edge that we are currently inspecting. If this is the only way to extend the paths in S, proceed along the anti-diagonal. Otherwise, there is at least one other edge that can continue the paths in S. Extend the matrix by adding as many rows and columns as there are elements in S. Say the new rows (columns) are members of a set S'. Initialize all entries of the $S' \times T$ sub-matrix to the same values as the $S \times T$ sub-matrix. Analogously for $T \times S'$. The $S' \times S'$-block is set to the same values as occur in the $S \times S$-block. When one reaches the source of the network, the matrix will contain exactly the overlaps between all the paths.

The matrix rows can be used as the identifiers. At this point of the algorithm, however, it is not clear which edge belongs to which path. Thus the edges cannot be assigned the weights that result from inverting the matrix. This information can be generated by, upon backtracking, building a tree where an edge describing a certain path branches into the new paths to which it gives rise. This tree will have as many leaves as there are paths through the network. After completion of the the above algorithm, if one goes from the leaves to the root, one can assign to each edge of the network a list of pointers to all paths in which it is contained.

8. Conclusions

Motivated by biological applications, we presented axioms describing a weighting scheme that may be applied to sequences connected by a phylogenetic tree or to different alignments of two sequences. It turns out that the axioms describe the current flow through an electrical network. Using this analogy, it is not difficult to prove the existence of the weights. The electrical networks are, of course, not necessary but simply helpful in understanding what the weights capture.

When this weighting scheme was first applied to biological problems by Altschul et al., the motivation was drawn from linear least squares estimation. In the more involved case of alignment path weighting it is hard to see how to extend that treatment, while the electrical networks form a natural framework for both problems. The question why the optimization problem in least squares estimation and the electrical networks lead to the same answer has, on a formal level, been answered. In more depth this has been well explained by G. Strang [8] in his book "Introduction to Applied Mathematics". His entire book is based on this duality between solving linear equations and optimizing quadratic forms. The phenomenon we observed in our application is only a special case of this.

A more systematic application of numerical methods combined with the analysis of the topology of a network should make the weights computable even for large sets of alignments. This in turn will allow a dot-plot representation of a large set of alignments

where edges are colored according to their weight. Visual analysis of such a dot-plot will thus be made significantly easier. Furthermore, applications in the assessment of alignment reliability [9] can be envisaged. With respect to the biological applications several open problems remain, however. We have not characterized the situations where weights will be positive. This question becomes increasingly harder as one applies the axioms for weighting to other objects, for example to RNA structures. First results show that negative weights are common in this case.

Acknowledgements

Most of this work was done while M.V. was a postdoctoral research associate at the University of Southern California and was supported by grants from the National Science Foundation (DMS 90-05833, DMS 87-20208) and the National Institutes of Health (GM-36230).

References

[1] S. F. Altschul, R. J. Carroll and D. J. Lipman, Weights for data related by a tree, J. Mol. Biol. 207 (1989) 847–653.
[2] S. F. Altschul and B. W. Erickson, Optimal sequence alignment using affine gap costs, Bull. Math. Biol. 48 (1986) 603–616.
[3] S. F. Altschul and D. J. Lipman, Equal animals, Nature 348 (1990) 493–494.
[4] B. Bollobás, Graph Theory (Springer, Berlin, 1979).
[5] S. Miyazawa, A reliable sequence alignment method based on probabilities of residue correspondences, Protein Eng. 8 (1995) 999–1009.
[6] W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, Numerical Recipes in C. (Cambridge University Press, Cambridge, 1988).
[7] T. F. Smith, M. S. Waterman, and W. Fitch, Comparative biosequence metrics, J. Mol. Evol. 18 (1981) 38–46.
[8] G. Strang, Introduction to Applied Mathematics, (Wellesley-Cambridge Press, Cambridge, 1986).
[9] M. Vingron and P. Argos, Determination of reliable regions in protein sequence alignments, Protein Eng. 3 (1990) 565–569.
[10] M. Vingron and P. R. Sibbald, Weighting in sequence space: A comparison of methods in terms of generalized sequences, Proc. Natl. Acad. Sci. USA 90 (1993) 8777–8781.