

## The MAP Test for Multimodality

Gregory Paul M. Rozál

Yale University

J. A. Hartigan

Yale University

**Abstract:** We introduce a test for detecting multimodality in distributions based on minimal constrained spanning trees. We define a Minimal Ascending Path Spanning Tree (MAPST) on a set of points as a spanning tree that has the minimal possible sum of lengths of links with the constraint that starting from any link, the lengths of the links are non-increasing towards a root node. We define similarly MAPSTs with more than one root. We present some algorithms for finding such trees. Based on these trees, we devise a test for multimodality, called the MAP Test (for Minimal Ascending Path). Using simulations, we estimate percentage points of the MAP statistic and assess the power of the test. Finally, we illustrate the use of MAPSTs for determining the number of modes in a distribution of positions of galaxies on photographic plates from a rich galaxy cluster.

**Keywords:** Minimal constrained spanning trees; Nearest neighbor density estimates; Minimal ascending path spanning trees; Tests for modes; The MAP test.

### 1. Introduction

There have been several proposals for detecting multimodality in a distribution. In the one-dimensional case, Kruskal proposes a test of multimodality based on successive order statistics, when the modes have been specified in advance (Giacomelli, Wiener, Kruskal, Pomeranz, and Loud 1971).

Silverman (1981) describes a test for multimodality based on the smallest window width  $h$  for a kernel density estimate  $\hat{f}$  to be unimodal. When the

underlying distribution is bimodal, the true  $h$  will be large because a considerable amount of smoothing is necessary to yield a unimodal estimate from samples exhibiting bimodality. The significance level of the test may be estimated by the proportion of bootstrap resamples from  $\hat{f}$  that yield multimodal density estimates when using the sample window width  $h$ .

To test for multimodality in one dimension, Hartigan and Hartigan (1985) suggest using the dip statistic:

$$dip = \inf_{F \in U} \sup_x |F(x) - F_n(x)|,$$

the maximum difference between the empirical distribution function  $F_n$  and the closest distribution  $F$  from the class of all unimodal distributions  $U$ . A large dip indicates multimodality. The dip, however, does not generalize readily to more than one dimension because the measure of distance between distribution functions is meaningful only in one dimension.

With the SPAN test, Hartigan (1988) extends the idea of the dip to more than one dimension. For the SPAN test, Hartigan uses distribution functions defined on rooted minimal spanning trees instead of the usual multivariate distribution functions. With respect to the root node, a rooted tree induces a partial order on its nodes. The SPAN statistic is the maximum difference between the empirical distribution function and its closest unimodal approximant, minimized over all choices of root. The computation of the SPAN statistic is quite involved.

Continuing with the use of minimal spanning trees, Hartigan and Mohanty (1992) devised the RUNT test, a simpler test based on the asymptotics of single linkage clusters. Single linkage clustering may be viewed as a hierarchical division of the set of all points into two or more clusters, and then the division of each of these subclusters into two or more subclusters, and so on, until each cluster consists of a single point. The RUNT statistic is the number of points that belong to the smallest of the subclusters obtained by such a division, maximized over all possible divisions. Under bimodality, the RUNT is expected to be large.

In this paper, we develop a test based on minimal spanning trees with the constraint that the link lengths are non-increasing on the path to the root node starting from any link. We shall refer to this constraint on the link lengths as the *ascending path constraint*. For each choice of root node, a tree with the smallest possible weight that satisfies the ascending path constraint is a Minimal Ascending Path Spanning Tree (MAPST). Because we would like to accommodate the possibility of multimodality, we also define MAPSTs with more than one root. A multiple root MAPST is a spanning tree of minimal weight constrained so that starting from any link there is a path to

one of the root nodes satisfying the ascending path constraint. Computed through such trees, we named our proposed test statistic for multimodality the MAP, for Minimal Ascending Path.

First, we shall present algorithms to find MAPSTs. Then, appealing to notions from nearest neighbor density estimates, we develop the MAP statistic for testing multimodality. Using simulations from some null distributions, we estimate the 95% point of the distribution of the MAP statistic. Then, we compare the power of the MAP statistic against the RUNT statistic in detecting bimodality for some alternative distributions. Finally, we illustrate the technique of using MAPSTs to test for multimodality by an application to a data set on the positions of galaxies on photographic plates in a rich galaxy cluster.

## 2. MAPSTs and Algorithms

Although the minimum spanning tree (MST), i.e., a tree connecting all points with the minimal total length, may be used to identify high density clusters (Zahn 1971), it does not correspond consistently to any particular feature of the density. In particular, the gradient lines of the density are the theoretical analogs of the minimum spanning tree, but the MST does not converge to them. If the links of a sample tree were able to trace gradients of the underlying density, then we would expect to recover similar looking trees for different samples from the same distribution. By forcing the links to ascend the gradients of the underlying density, we should be able to consistently identify disconnected high density regions corresponding to different modes in the true density. With this in mind, we define a spanning tree having the following characteristics:

**Definition.** A MAPST with root node  $x_0$  is a spanning tree of minimal total length constrained so that the link lengths are non-increasing in the path from any link to  $x_0$ .

The MAPST is a rooted tree and is usually, but not always, different for different choices of the root, for the same set of points. A MAPST is not a directed graph since the links follow no fixed directional flow, i.e., it can either flow away from or towards a root node, but for convenience in describing and proving the algorithms, we shall adopt the following convention. If  $l(x,y)$  is a link in a MAPST connecting the nodes  $x$  and  $y$ , then either  $x$  is a root node, or  $x$  is passed on the way to a root node from  $y$ . To find MAPSTs, we modified the following algorithm, due to Prim (1957) for finding minimal spanning trees.

**Algorithm (Prim 1957):**

1. Choose any starting point. Call this initial tree  $T_1$ .
2. At the  $k$ -th step, connect to the tree  $T_{k-1}$ , a point which is not yet in  $T_{k-1}$  that is nearest in Euclidean distance to  $T_{k-1}$ . This gives the tree  $T_k$ .
3. Repeat Step 2 until all points are added to the tree.

Although we shall not prove this theorem, we have included it here for the sake of completeness.

**Theorem 1** *A spanning tree constructed using the Prim algorithm is a minimal spanning tree, that is, a tree connecting all points of minimal total length.*

Realizing that a MAPST is really just a MST with a constraint on the kind of links that are allowed to form, we arrived at the MAPST algorithm by modifying the second step of the Prim algorithm to include the ascending path constraint.

**MAPST Algorithm:**

1. Choose any point as a root node. Call this initial tree  $T_1$ .
2. At the  $k$ -th step, connect to  $T_{k-1}$  a point  $x \notin T_{k-1}$  that will be connected with the shortest possible link provided that this new link is at least as long as the link to which it will be connected (Ascending Path Property).
3. Repeat Step 2 until all points have been included in the tree.

Figure 1 illustrates the algorithm. In this example, the root is 1 (encircled in Figure 1a.) As shown in Figure 1b, the closest node to root 1 is 2. The succeeding figures show the points being connected in order, according to the MAPST algorithm. Figure 1f shows the completed MAPST with root at 1.

The root node is considered to be connected to itself; hence, the initial link has zero length. Thus, it is always possible to obtain a spanning tree with the ascending path property. Starting from the root node, each point in the sample is joined into the tree in a certain order, assuming that no two inter-point distances are equal. Then, with respect to the root node, the sample points are partially ordered, so that if  $x_i$ ,  $x_j$ , and  $x_k$  were joined in that order when  $x_0$  is the root, with  $x_j$  joined to  $x_i$  and  $x_k$  to  $x_j$ , then the lengths of the links, denoted by  $d(x_i, x_j)$  and  $d(x_j, x_k)$ , respectively, are related by  $d(x_i, x_j) \leq d(x_j, x_k)$ .

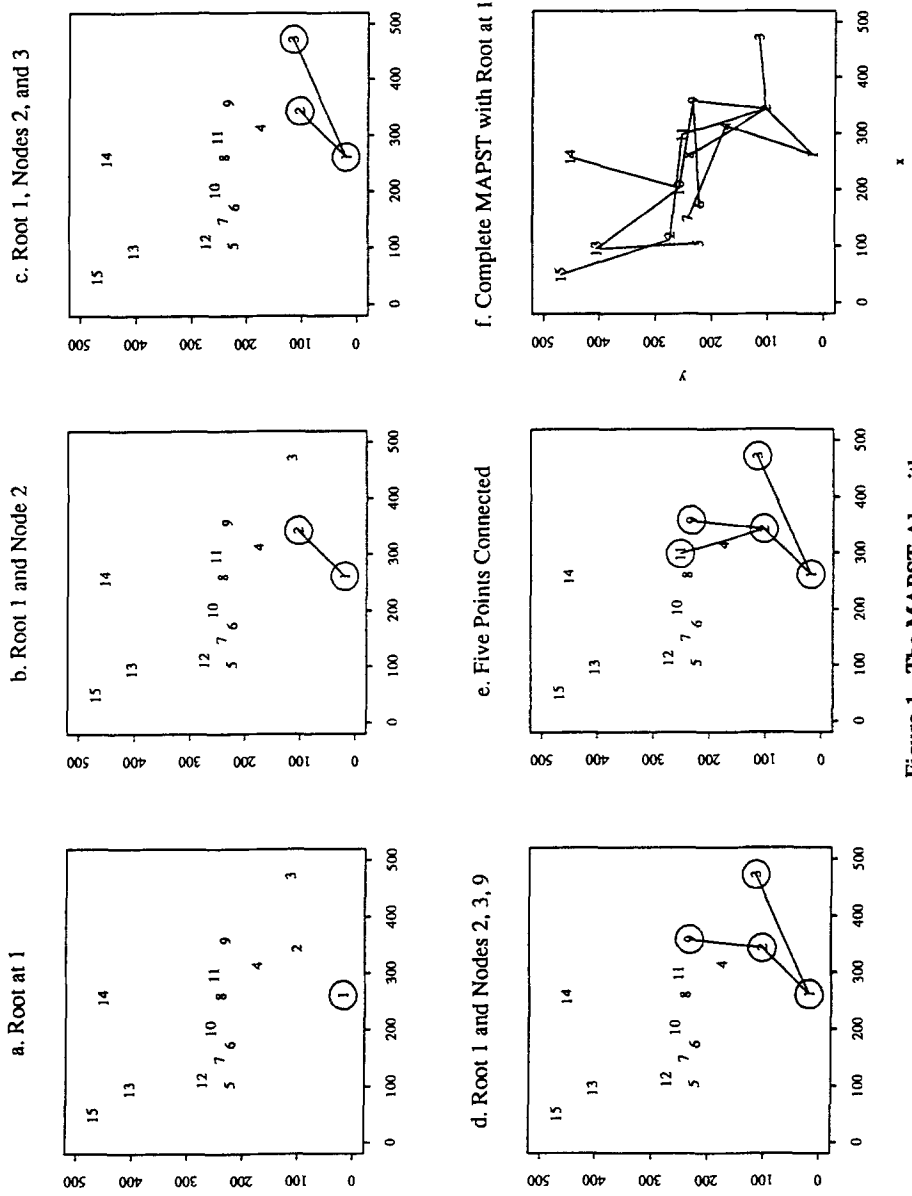


Figure 1. The MAPST Algorithm.

**Theorem 2** *A spanning tree constructed using the MAPST algorithm is a minimal ascending path spanning tree. (A proof is included in the Appendix.)*

The MAPST with one root node may be appropriate for unimodal distributions, but since we are allowing for the possibility of multimodality in the underlying distribution of the data, we shall define MAPSTs with more than one root.

**Definition.** *A MAPST with  $r$  root nodes or a multiple root MAPST is a spanning tree of minimal total length with the property that from any link, there is a path to one of the  $r$  root nodes made up of links whose lengths are non-increasing.*

The MAPST algorithm can also be used to find MAPSTs with more than one root. Two changes are necessary: (1), start with a chosen set of  $r$  root nodes instead of just a single one; and (2), connect the subtrees that will be formed into a single spanning tree using the appropriate additional links to satisfy the ascending path constraint. Presented here are two algorithms to find multiple root MAPSTs. The First Algorithm is an extension of the MAPST algorithm for single roots to an algorithm for several roots. The Second Algorithm uses the tree information, i.e., links and link lengths, from the single root MAPSTs. The First Algorithm is easier to describe and understand; the Second Algorithm is more complicated but more efficient.

### First Algorithm:

1. Select  $r$  points as root nodes. Call this initial graph  $T_1$ .
2. At the  $k$ -th step, connect to  $T_{k-1}$  a point  $x \notin T_{k-1}$  that will be connected with the shortest possible link under the condition that this new link formed is at least as long as the link to which the new link will be connected (Ascending Path Property).
3. Repeat Step 2 until all points have been included in the graph.
4. When all points have been connected to the graph, join the disconnected subtrees into a single spanning tree by a set of links between subtrees such that the ascending path constraint is satisfied and such that the total length is minimal.

Since each root node is considered to be connected to itself, the initial links have zero length. Thus, it is always possible to find a spanning tree with  $r$  root nodes satisfying the ascending path property.

Initially, each root node is connected to itself. So, with  $n$  points and  $r$  roots, there will be  $n - r$  links with distinct end points, after all points have been included in the graph. At this stage, the graph consists of  $r$  ascending

path subtrees — each subtree with its own root. At the final step of the algorithm, the subtrees are connected by links from a point in one subtree to a point in another. The links are chosen to form a spanning tree having minimal total length and satisfying the ascending path constraint.

**Theorem 3** *A spanning tree constructed by the First Algorithm is a minimal ascending path spanning tree with  $r$  root nodes. (A proof is given in the Appendix.)*

In a MAPST with  $r$  root nodes, every node  $x$  can be associated with the root node of the subtree in which  $x$  is connected before the subtrees are joined. We define formally this concept.

**Definitions:**

1. A node  $x$  belongs to root node  $x_0$  if  $x$  belongs to the subtree  $T(x_0)$  that contains  $x_0$ , i.e.,  $x \in T(x_0)$ .
2. A bridge link is any link whose end points belong to different root nodes.

The final step in the preceding algorithm involves searching for all bridge links that satisfy the ascending path constraint, and finding a set of bridge links that is both minimal and spans all the subtrees. It takes  $O(n^2)$  comparisons to accomplish this task. We shall follow another route to decrease the required number of operations. Define the distance between two subtrees with root nodes  $x$  and  $y$ ,  $D(x,y)$ , by the length of the shortest bridge link between the subtrees that would satisfy the ascending path constraint. By using these bridge distances between subtrees and treating the  $r$  ascending path subtrees as points, we can find a minimal spanning tree to join the subtrees into a single spanning tree with minimal weight.

**Second Algorithm:**

1. Let  $v_x(y)$  be the node to which  $y$  is attached when  $x$  is the root node in a single root MAPST, and let  $\rho_x(y)$  be the length of this link,  $l(v_x(y), y)$ .
2. To construct a MAPST with  $r$  root nodes,  $x_1, \dots, x_r$ , attach  $y$  to  $v_{x_i}(y)$  where  $x_i$  is a root node that minimizes  $\rho_{x_i}(y)$ . For example, to construct the MAPST when the root nodes are  $x_1$  and  $x_2$ , attach  $y$  to  $v_{x_1}(y)$  if  $\rho_{x_1}(y) \leq \rho_{x_2}(y)$ ; otherwise, attach  $y$  to  $v_{x_2}(y)$ .
3. Compute the distance  $D(x_s, x_t)$  between two subtrees with roots  $x_s$  and  $x_t$  as follows:

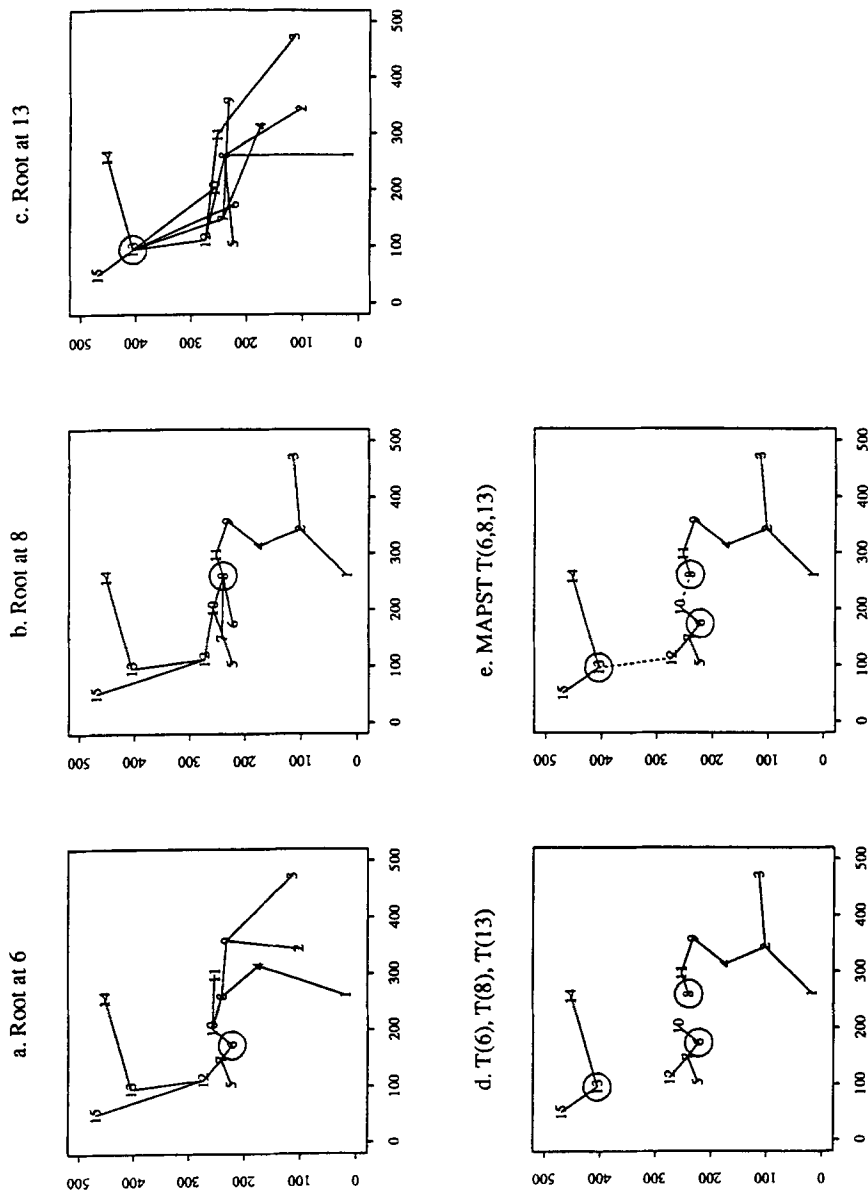


Figure 2. The Second Multiple Root MAPST Algorithm. Single Root MAPSTs, Subtrees and MAPST with Three Roots.



Table 1: The Second Algorithm for Multiple Root MAPSTs

Point	$\rho_6(\cdot)$	$\nu_6(\cdot)$	$\rho_8(\cdot)$	$\nu_8(\cdot)$	$\rho_{13}(\cdot)$	$\nu_{13}(\cdot)$	$T(6, 8, 13)$
1	164	4	117	2	221	8	2
2	131	9	78	4	160	8	4
3	164	9	130	2	222	11	2
4	84	8	73	9	178	7	9
5	48	7	104	10	157	8	7
6	0	6	89	8	199	13	6
7	32	6	112	8	171	13	6
8	61	10	0	8	153	12	8
9	98	8	63	11	210	7	11
10	46	6	61	8	182	13	6
11	96	10	40	8	188	12	8
12	49	7	92	10	132	13	7
13	132	12	132	12	0	13	13
14	169	13	169	13	169	13	13
15	204	12	204	12	78	13	13

- (a) Among all the nodes  $y$  that belong to  $x_s$ , find a node  $y^*$  that minimizes  $\rho_{x_i}(y)$ , i.e.,

$$\rho_{x_i}(y^*) = \min_{y \in T(x_s)} \rho_{x_i}(y).$$

- (b) If  $\nu_{x_i}(y^*)$  belongs to  $x_t$ , then  $D(x_s, x_t) = \rho_{x_i}(y^*)$ ; otherwise, set  $D(x_s, x_t) = \infty$ .

4. Using the distances  $D(x_s, x_t)$  between subtrees, find a minimal spanning tree (treating each subtree as a point) that will connect the subtrees into a single spanning tree.

To illustrate the Second Algorithm, consider the points shown in Figures 2a-e. As indicated, Figures 2a-c are single root MAPSTs with roots at 1, 8, and 13, respectively. We shall use these trees to make the multiple root MAPST  $T(6,8,13)$  of Figure 2e.

Table 1 contains all the necessary information from the single root MAPSTs to make the  $T(6,8,13)$ . The first column indexes the points in the sample. Following our notation, the  $\rho_i(\cdot)$  columns are the lengths of the links when  $i$  is the root, and the  $\nu_i(\cdot)$  columns are the points to which each point in the first column is connected to when  $i$  is the root. For example, when the root is 6, point 1 is connected to point 4; the length of the link  $l(4,1)$  is 164. Hence,  $\rho_6(1) = 164$  and  $\nu_6(1) = 4$ .

Following Step 2 of the Second Algorithm, for each row  $j$ , choose the  $v_i(j)$  that has the smallest corresponding  $\rho_i(j)$ . We have written the minimizing  $v_i(j)$ s in the  $T(6,8,13)$  column. Now, if we connect each point  $i$  to the corresponding entry in the last column, we get three subtrees whose elements belong to the roots 6, 8, and 13. We show these subtrees  $T(6)$ ,  $T(8)$ , and  $T(13)$  in Figure 2d.

To complete the MAPST, we need to find the appropriate bridge links to connect the subtrees. By Step 3, we compute the distance between each pair of subtrees as follows. Mark each row of Table 1 with the number of the root of the tree to which the row belongs. In our example, rows 1, 2, 3, 4, 8, 9, and 11 belong to  $T(8)$ , rows 5, 6, 7, 10, and 12 belong to  $T(6)$ , and rows 13, 14, and 15 belong to  $T(13)$ . Among all rows that belong to  $T(8)$ , find the row with the smallest  $\rho_6(\cdot)$ ; this would be row 8 in Table 1. The corresponding  $v_6(8) = 10 \in T(6)$ . Then, the distance between  $T(6)$  and  $T(8)$  is  $D(8,6) = 61$ , i.e., the length of the link  $l(10,8)$ . Note that the smallest entry in column  $\rho_8(\cdot)$ , among all rows belonging to 6, is also 61, i.e., the length of  $l(8,10)$ . Then,  $D(6,8) = 61$ . Using the same procedure, the distance between  $T(6)$  and  $T(13)$  is  $D(13,6) = D(6,13) = 132$ , using bridge link  $l(12,13)$ .

Of the rows belonging to  $T(13)$ , the minimal entry in  $\rho_8(\cdot)$  is 132, corresponding to  $v_8(13) = 12 \notin T(8)$ . In this case, we set  $D(8,13) = \infty$ . If we take the minimum of the entries in  $\rho_{13}(\cdot)$  in rows belonging to  $T(8)$ , the minimum would be at row 8, with  $\rho_{13}(8) = 153$  and  $v_{13}(8) = 12 \notin T(13)$ . Then,  $D(13,8) = \infty$ . Therefore, the complete tree includes the bridge links  $l(8,10)$  and  $l(12,13)$  as shown in Figure 2e.

The Second Algorithm is a more efficient method for constructing multiple root MAPSTs. Since it only uses the information from the single root MAPSTs, several multiple root MAPSTs can be computed in parallel.

Since  $\rho_{x_i}(x_i) = 0 < \rho_{x_i}(x_j)$  for  $x_i \neq x_j$ , a root node will be connected to itself as in the previous algorithm. Except for the root nodes, each node is attached to another node by the second step of the algorithm. Thus, there will be a path defined by such attachments from each node to just one of the roots. These paths are ascending because they are derived from connections in the single root MAPSTs. The subtree of each root consists of all nodes connected to that root by the attachments.

For the MAPST with two roots at  $x_1$  and  $x_2$ , the distance between the subtrees  $T(x_1)$  and  $T(x_2)$  is the smallest among the larger of  $\rho_{x_1}(y)$  and  $\rho_{x_2}(y)$ , the lengths of the links that were not used in making the subtrees.

The reason for item (b) in Step 3 is to handle the case, for more than two roots, when the  $y^*$  chosen in (a) connects to a node  $v_{x_i}(y^*)$  not belonging to  $T(x_i)$ ; when this situation occurs, the link  $l(v_{x_i}(y^*), y^*)$  of length  $\rho_{x_i}(y^*)$  is ineligible to be in the final MAPST. Thus, whenever  $D(x_s, x_t) = \infty$ , the

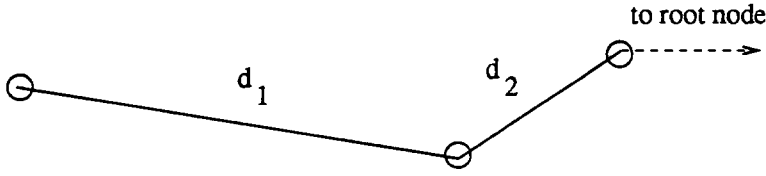


Figure 3. The Generalized Ascending Path Algorithm ( $d_1 \geq Sd_2 + R$ , for  $S \geq 1$  and  $R \geq 0$ .)

optimal MAPST will not include any direct link between objects in  $T(x_s)$  and  $T(x_t)$ . (This case occurred between subtrees  $T(8)$  and  $T(13)$  in the example.)

The final step in the Second Algorithm finds the set of links that will connect the subtrees to have a spanning tree with the least possible overall weight, subject to the ascending path constraint.

The following lemmas are used to prove that the Second Algorithm finds a multiple root MAPST. They are stated here because they are interesting in and of themselves. The proofs are included in the Appendix.

**Lemma 4** *Paths formed by the Second Algorithm starting from any link, ending at a root node, and containing no bridge link are ascending.*

**Lemma 5** *The bridge links formed by the Second Algorithm cannot be shorter than the adjacent non-bridge links.*

**Theorem 6** *The ascending path spanning tree with  $r$  root nodes obtained by the Second Algorithm is minimal.*

The ascending path constraint requires that links decrease on a path from any node to a root. It is possible to generalize this constraint without making the tree harder to compute. For the following, refer to Figure 3. If  $d_1$  and  $d_2$  are lengths of successive links on a path to a root, we generalize the ascending path constraint and require that  $d_1 \geq Sd_2 + R$  for some fixed  $S \geq 1$ ,  $R \geq 0$ . We shall call such a tree a MAPST( $S, R$ ).

**Definition:** *For  $S \geq 1$  and  $R \geq 0$ , a MAPST( $S, R$ ) with root node  $x_0$  is a spanning tree of minimal total length constrained so that the path from any link to  $x_0$  satisfies the generalized ascending path constraint.*

#### Generalized MAPST Algorithm:

1. Fix  $S \geq 1$  and  $R \geq 0$ . Choose any point as a root node. Call this initial tree  $T_1$ .
2. At the  $k$ -th step, connect to  $T_{k-1}$  a point  $x \notin T_{k-1}$  that will be connected with the shortest possible link provided that this new link satisfies the generalized ascending path constraint.

3. Repeat Step 2 until all points have been included in the tree.

The root node is considered to be connected to itself; hence, the initial link has zero length. Thus, it is always possible to obtain a spanning tree satisfying the generalized ascending path constraint.

**Theorem 7** *A spanning tree constructed using the generalized MAPST algorithm is a MAPST( $S, R$ ).*

We will not prove this theorem in detail; instead, we shall point out the key elements in proving the theorems on the MAPST algorithms. In proving the earlier theorems, we started with the following assumption — among all the spanning trees that satisfied the appropriate ascending path property, there exists at least one  $T^*$  that has minimal weight. We proved that the initial tree  $T_1$  obtained by the algorithm is contained either in  $T^*$  or if not, in another spanning tree satisfying the ascending path property and having the same weight as  $T^*$ , i.e., another MAPST. Assuming that  $T_k \subset T^*$ , we showed that  $T_{k+1} \subset T^*$  or some other MAPST. By mathematical induction, we claimed that  $T_n \subset T^*$ .

To prove that  $T_{k+1} \subset T^*$ , assuming that  $T_k \subset T^*$ , we added the link  $l(x_k, x_{k+1})$  to  $T^*$ , if it is not already there. Because  $T^*$  is a spanning tree, adding  $l(x_k, x_{k+1})$  induces a cycle  $C$  in  $T^*$ . Consider Figure 4 below. In cycle  $C$ , there exists a point  $z$  that is included in all paths in  $T^*$  to the root node from any point in  $C$ . We call such a node  $z$ , the *maximal node in the cycle*. As an aside, we proved the following theorem about the maximal node in a cycle.

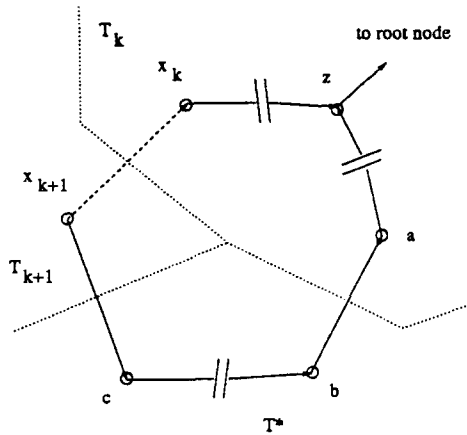
**Theorem 8** *There exists a unique maximal node in every cycle induced by adding a link to a MAPST. (A proof is in the Appendix.)*

In Figure 4 points  $a$  and  $b$  are such that link  $l(a, b)$  is in  $T^*$ ,  $a \in T_k$ , but  $b \notin T_k$ . In  $T^*$ ,  $x_{k+1}$  is connected to  $c$ .

By the ascending path property in  $T^*$ ,  $d(c, x_{k+1}) \geq S \times d(a, b) + R$ . By construction,  $d(a, b) \geq d(x_k, x_{k+1})$ ; otherwise,  $l(a, b)$  would have formed before  $l(x_k, x_{k+1})$ . Therefore,

$$d(c, x_{k+1}) \geq S \times d(a, b) + R \geq S \times d(x_k, x_{k+1}) + R \geq d(x_k, x_{k+1})$$

for  $S \geq 1$  and  $R \geq 0$ . If  $d(c, x_{k+1}) > d(x_k, x_{k+1})$  then  $T^*$  cannot be minimal; otherwise,  $d(c, x_{k+1}) = d(x_k, x_{k+1})$  and obtain another MAPST from  $T^*$  that now includes  $T_{k+1}$ .


 Figure 4. Cycle  $C$  and Portions of Trees  $T^*$ ,  $T_k$ , and  $T_{k+1}$ .

When constructing  $\text{MAPST}(S, R)$ s that have more than one root, the second algorithm is still applicable, using  $\rho_{x_i}(y)$ 's and  $v_{x_i}(y)$ 's derived from the single root  $\text{MAPST}(S, R)$ s. Having established that multiple root  $\text{MAPST}(S, R)$ s can be found, we now proceed to develop a test of multimodality from these trees.

### 3. The MAP Test for Multimodality

#### 3.1 The Nearest Neighbor Density Estimate

Following Silverman (1986, p. 96), let  $x_1, \dots, x_n \in R^d$  be sample points from a density  $f$  with an unknown number of modes. Let  $r_k(x)$  be the Euclidean distance from  $x$  to the  $k$ -th nearest point. If  $c_d$  is the volume of the unit sphere in  $d$  dimensions, i.e.,  $c_1 = 2$ ,  $c_2 = \pi$ ,  $c_3 = 4\pi/3$ , etc., then  $V_k(x) = c_d r_k(x)^d$  is the ( $d$ -dimensional) volume of the  $d$ -dimensional sphere  $S_k(x)$  of radius  $r_k(x)$  centered at  $x$ . Of the  $n$  sample points, it is expected that about  $nf(x)V_k(x)$  fall inside  $S_k(x)$ . Equating this expected number with the number  $k$  actually observed gives the nearest neighbor density estimate of  $f$  at  $x$ ,  $\hat{f}(x)$ ,

$$\hat{f}(x) = \frac{k}{nV_k(x)} = \frac{k}{nc_d r_k(x)^d}.$$

Then, for fixed  $n$  and  $d$ , the  $d$ -th root of the density estimate at  $x$  is inversely proportional to the distance of the  $k$ -th nearest point from  $x$ . Thus, the density estimate is highest where the points are closest together indicating the

possible existence and locations of modes. By taking logarithms of both sides, we find that the log density estimate at  $x$  is linear in the minus log distance of the  $k$ -th nearest point from  $x$ .

### 3.2 The MAP Statistic

For  $x_1, \dots, x_n$ , let  $T_U(x_i)$  ( $U$  for unimodal) be a MAPST with a single root at  $x_i$ ; let  $T_B(x_{i_1}, x_{i_2})$  ( $B$  for bimodal) be a MAPST with roots  $x_{i_1}$  and  $x_{i_2}$ . Let  $\rho_{x_i}(x_j)$  be the length of the link from  $x_j$  to the next node  $v_{x_i}(x_j)$  in the direction towards the root in  $T_U(x_i)$ . Define the functions  $L_U(x_i)$  and  $L_B(x_{i_1}, x_{i_2})$ , which are the sum of the logarithms of the lengths of the links in  $T_U(x_i)$  and  $T_B(x_{i_1}, x_{i_2})$  respectively, by

$$L_U(x_i) = \sum_{j \neq i} \log \rho_{x_i}(x_j),$$

$$L_B(x_{i_1}, x_{i_2}) = \sum_{j \neq i_1, i_2} \log (\min_{i \in i_1, i_2} \rho_{x_i}(x_j)) + \log D(x_{i_1}, x_{i_2}),$$

where  $D(x_{i_1}, x_{i_2})$  is the distance between the subtrees belonging to the roots  $x_{i_1}$  and  $x_{i_2}$  in  $T_B(x_{i_1}, x_{i_2})$ , i.e., the length of the shortest bridge link in  $T_B(x_{i_1}, x_{i_2})$ . In practice, we need to protect against very small or zero distances  $\rho_{x_i}(x_j)$  which will produce  $\log \rho_{x_i}(x_j) = -\infty$ . There is no compelling cure for this problem; what we have done is set a threshold at the 5% point of the set of MAPST links and set any distance smaller than this threshold to be the threshold.

The nearest neighbor density estimate suggests that  $-\log \rho_{x_i}(x_j)$  is roughly proportional to the log density at  $x_j$ . Thus, when  $f$  is unimodal,  $-L_U(x_i)$  should be roughly proportional to the log likelihood  $\sum_{j \neq i} \log f(x_j)$ . When  $f$  is bimodal, however, some of the distances  $\rho_{x_i}(x_j)$  will be substantially larger than the nearest neighbor distances. Then,  $L_U(x_i)$  should be larger than minus the log likelihood.

Under bimodality, the links appearing in  $T_B(x_{i_1}, x_{i_2})$  will be close to the nearest neighbor links and so,  $-L_B(x_{i_1}, x_{i_2})$  will be close to the log likelihood

$$\sum_{j \neq i_1, i_2} \log f(x_j) + \log D(x_{i_1}, x_{i_2}).$$

The term  $D(x_{i_1}, x_{i_2})$  is necessary so that the same number of links appear in both expressions; otherwise, the test statistic that we shall now describe would depend on the scaling of the data. The bridge link that joins the

subtrees in  $T_B(x_{i_1}, x_{i_2})$  was chosen to reflect the density around the saddle point between two possible modes.

If the interpoint distances around a root  $x_i$  are small, the links of  $T_U(x_i)$  are small, and  $L_U(x_i)$  is small. Thus, we expect  $L_U(x_i)$  to be minimized whenever  $x_i$  is near a mode of  $f$  where the density is highest and the points are closest together. In the same way, we expect  $L_B(x_{i_1}, x_{i_2})$  to be minimized whenever either root is near the mode of  $f$ , if  $f$  is unimodal, or each root is near some mode of  $f$ , if  $f$  has two or more modes. Therefore, we define the MAP statistic for bimodality by

$$MAP = \min_i L_U(x_i) - \min_{i_1, i_2} L_B(x_{i_1}, x_{i_2}).$$

If  $f$  were unimodal, then the interpoint distances should be smaller around the sample points that are near the mode than those for points that are far from the mode. So, both  $L_U$  and  $L_B$  are minimized when the roots are chosen to be those close to the mode of  $f$ . Except for the small links near the roots, the unimodal and bimodal trees will be nearly identical and the value of the MAP will be near zero. On the other hand, if  $f$  were bimodal,  $L_B$  is minimized when one root is near one mode and the other root is near the other mode;  $L_U$  is minimized when the root is near one of the modes. With the root of the minimal unimodal tree  $T_U$  (minimizing  $L_U$ ) being near one of the modes, the links of  $T_U$  that connect points belonging to the other mode will be considerably longer than those links in the minimal bimodal tree  $T_B$  that connect those same points into  $T_B$ . There will be a considerable difference between the minima of  $L_U$  and of  $L_B$ ; the MAP will be bounded away from zero.

Figure 5 shows what happens typically to the minimal unimodal and bimodal MAPSTs when the data exhibit bimodality. In this example, the minimal unimodal MAPST has its root at 1, and the minimal bimodal MAPST has roots at 1 and 7.

The test for bimodality can be easily extended to a test of multimodality by computing the difference in the sum of the log lengths of the best fitting  $k$ -modal MAPST  $T_k$ , i.e., the MAPST with  $k$  roots having the smallest sum of log link lengths, and the best fitting  $(k+1)$ -modal MAPST  $T_{k+1}$ . The multimodal MAP statistic that tests for  $k+1$  modes is given by:

$$MAP_k = \min L_k(x_{i_1}, \dots, x_{i_k}) - \min L_{k+1}(x_{j_1}, \dots, x_{j_{k+1}}),$$

where  $L_k(x_{i_1}, \dots, x_{i_k})$  is the sum of log link lengths of the MAPST when the root nodes are  $x_{i_1}, \dots, x_{i_k}$ .

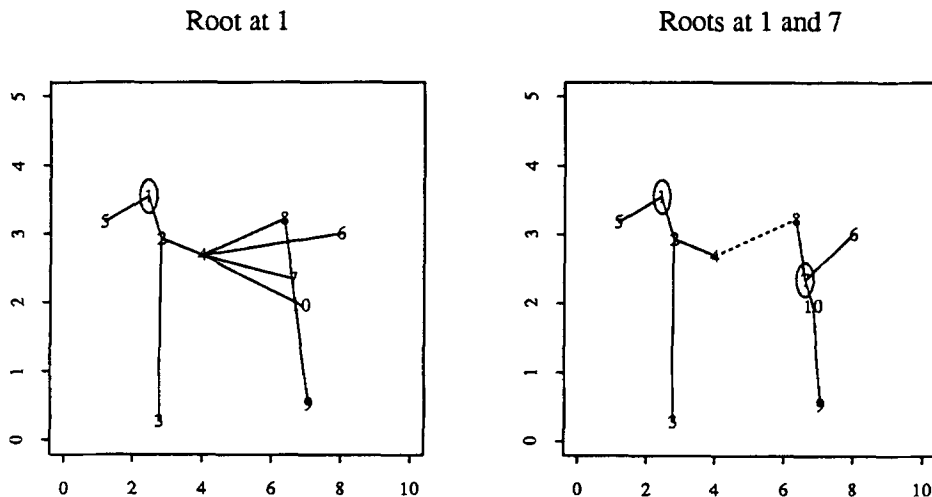


Figure 5. Unimodal and Bimodal MAPSTs on a Bimodal Sample.

The idea behind the MAP test for bimodality and the extension to a test for multimodality is the same. For any given set of points, the weight of the best fitting MAPST decreases with the number of roots; all the  $MAP_k$  statistics are non-negative. However, when the correct number of roots is reached, the decrease in weight is considerable. This situation results in a large value for the  $MAP_k$  statistic.

The significance of a  $MAP_k$  statistic is determined by comparing values from resamples taken from a reference or null distribution. In samples where a satisfactory reference distribution cannot be guessed, we propose a resampling method to determine significance levels. In Section 5, we discuss this resampling method in more detail and give an example that illustrates our method of using MAPSTs to determine multimodality.

#### 4. 95% Point of the MAP Distribution and Power of the MAP Test

##### 4.1 Percentage Points for Multivariate Normal and Spherical Uniform Distributions

We present in Table 2 the 95% points of the distribution of the MAP statistic under the multivariate normal distribution  $N(0, I)$  and under the uniform distribution on the unit sphere  $U(0, 1)$ , in 1, 2, 5, and 20 dimensions, for sample sizes of 10 and 100. Each estimate was based on 999 simulations. Throughout the table, the standard errors are less than 5% of the values. For the normal distribution, aside from the ordinary MAP, we also computed the 95% point of the MAP statistic using MAPST(2,0)s. Under the same



Table 2: 95% Points of the MAP Distribution under some Distributions

Distribution	Sample Size	Dimensions			
		1	2	5	20
$N(0, I)$	10	7.00	3.09	1.33	0.45
	100	44.57	16.20	6.23	1.88
MAP(2,0) on $N(0, I)$	10	7.35	3.85	1.83	0.78
	100	65.66	29.91	13.95	7.03
$U(0, 1)$	10	7.87	4.12	1.68	0.52
	100	56.85	23.30	8.60	2.51

distribution, the percentage points should be greater for  $S > 1$  than those for  $S = 1$ . To satisfy the steeper ascending path constraint the links in the spanning trees have to be longer as  $S$  gets larger. Thus, the entries in the third and fourth rows should be greater than the corresponding entries in the first two rows.

The MAP statistic increases with sample size because it is not scaled by sample size. As more links are added to the trees, the weights of the trees increase. The MAP tends to decrease with the number of dimensions because, in higher dimensions, path links do not increase so much as low density points are added to the trees.

The percentage points are greater for the uniform distribution than for the normal because it is easier to have an appearance of bimodality in a sample from a uniform than from a normal.

#### 4.2 Power of the MAP Test

We computed the power of the MAP Test for detecting bimodality in mixtures of unit normals and in log normals, taking the unit normal as the null. We exhibit in Table 3 the probability of rejecting the null hypothesis using a 5% significance level

By  $\frac{1}{3} N(0, I) + \frac{2}{3} N(3, I)$ , we mean 1:2 mixture of normals with  $I$ , the identity matrix, as covariance matrix, but with mean vectors that are three units apart. The first distribution in the mixture has its mean at the origin while the second has its mean three units away from the origin. In the other mixture,  $\frac{1}{3} N(0, I) + \frac{2}{3} N(5, I)$ , the means are five units apart. The log normal alternative was tested against the unit normal to examine whether the MAP test confused long tails with modes. We estimated the power by the

proportion of the MAP statistics (from samples taken from the alternative distributions) that exceeded the 95% point under the null (cf. Table 2). For each alternative, 999 simulations were performed. For comparison, we have included the powers computed for the RUNT Test of Multimodality (Hartigan and Mohanty 1992).

When the alternative was  $\frac{1}{3} N(0, I) + \frac{2}{3} N(3, I)$ , the highest powers were observed in two dimensions for the ordinary MAP, and in five dimensions for the MAP(2,0). The powers increased with sample size. The MAP is consistently more powerful than the RUNT in detecting bimodality. When the alternative was  $\frac{1}{3} N(0, I) + \frac{2}{3} N(5, I)$ , the MAP performed even better against the RUNT. The MAP performed best in two and five dimensions for this alternative.

We experimented with using different values of  $S$ , for various combinations of dimensions and of sample sizes. We noticed that the MAP test has the highest powers for detecting bimodality in samples from normal mixtures when  $S = 2$ . In this case, the MAP performs best around three to five dimensions. Using  $S = 2$  gave a test that had superior power both to the case when  $S = 1$  and to the RUNT; we have not explored how to set  $R$  and  $S$  in general, but the preliminary experiments suggest that  $S > 1$  is helpful.

For the multivariate log normal  $e^{N(0, I)}$  alternative, low values in this section of Table 3 are an encouraging sign that the MAP does not detect the long tails of the log normal alternative as modes. As sample size increases, the MAP makes fewer errors in indicating non-existent modes. Overall, however, the RUNT appears to be more resistant to mistakenly identifying long tails as modes.

We computed the power of the MAP Test in detecting bimodality in a 1:2 mixture of uniforms on spheres with unit radii but with centers that are three units apart ( $\frac{1}{3} U(0, 1) + \frac{2}{3} U(3, 1)$ ) against a uniform on the unit sphere ( $U(0, 1)$ ). In the mixture, the first uniform is centered at the origin, while the second one is centered at the vector with the first component at three units and the other components at zero. We estimated the probability of rejecting the null hypothesis of unimodality under the alternative by the proportion of samples from  $\frac{1}{3} U(0, 1) + \frac{2}{3} U(3, 1)$  that had a MAP value greater than the 95% point of the MAP under  $U(0, 1)$  (shown in Table 2). The results of 999 simulations for sample sizes 10 and 100, and in 1, 2, 5 and 20 dimensions, are shown in Table 4. In this table the power of the MAP increases with sample size and dimension. In fact, when the sample size is 100, for dimensions 5 and 20 all the samples from the alternative distribution had MAP values exceeding the corresponding 95% point.

Table 3: Power of the MAP and RUNT Tests:  $N(0, I)$  against some Alternatives

Alternative Distribution	Sample Size	Test	Dimensions			
			1	2	5	20
$\frac{1}{3}N(0, I) + \frac{2}{3}N(3, I)$	10	MAP	.072	.149	.126	.094
		MAP(2,0)	.077	.128	.151	.092
		RUNT	.061	.093	.073	.063
	100	MAP	.091	.254	.213	.133
		MAP(2,0)	.112	.362	.368	.160
		RUNT	.084	.144	.159	.085
$\frac{1}{3}N(0, I) + \frac{2}{3}N(5, I)$	10	MAP	.157	.407	.441	.344
		RUNT	.082	.128	.148	.143
	100	MAP	.805	.950	.905	.655
		RUNT	.102	.684	.841	.525
$e^{N(0, I)}$	10	MAP	.032	.070	.076	.087
		RUNT	.069	.043	.017	.005
	100	MAP	.009	.037	.010	.015
		RUNT	.003	.004	.003	.001

Table 4: Power of the MAP and RUNT Tests:  $U(0, 1)$  against  $\frac{1}{3}U(0, 1) + \frac{2}{3}U(3, 1)$

Sample Size		Dimensions			
		1	2	5	20
10	MAP	.103	.334	.646	.902
	RUNT	.050	.061	.089	.096
100	MAP	.895	.995	.999	.999
	RUNT	.002	.011	.162	.801

### 5. An Example: A Cluster of Galaxies}

The data come from a study of 55 rich clusters of galaxies which included positions, morphological types, estimated total magnitudes, bulge sizes, and ellipticities for about 6000 galaxies determined from high scale photographic plates (Dressler 1980). From the 55 clusters, we have chosen the DC 0559-40 cluster which includes 116 galaxies. Using the X and Y position measured in millimeters on the plate, we applied our technique of using MAPSTs to test whether this cluster of galaxies exhibits multimodality.

We obtained the optimal (in terms of minimal sum of log link lengths) one, two, up to six-root MAPSTs,  $T_1, T_2, \dots, T_6$ , (see Figure 6) and computed the differences in the sum of the log link lengths as more roots are added. That is, if  $L_k$  is the sum of the log link lengths of the optimal MAPST with  $k$  root nodes over all possible choices of sets of roots, let  $MAP_k = L_k - L_{k+1}$  for  $k = 1, \dots, 5$ . We present the  $MAP_k$  statistics for the data in the first row of Table 5.

We obtained an appropriate null distribution for the  $MAP_1, \dots, MAP_5$  statistics through resampling via the following scheme. First, we obtained resamples of size 116 from  $T_1$  (shown in Figure 6a). Each point  $x$  on the tree  $T_1$  generates a resample point  $x'$ , taken at random from the circle of radius  $d(y, x)$  and center  $y$ , where  $y$  is the next point towards the root and  $d(y, x)$  is the length of the link  $l(y, x)$  from  $x$  to  $y$ . See Figure 7 for an illustration. Notice that when  $x$  is a root, only  $x$  falls in the resampling region; in this case,  $x' = x$ . The series of MAPs,  $MAP_1, \dots, MAP_5$ , were then computed for each set of 116 resample points. This procedure was repeated for 999 sets of resamples. The p-value for the null hypothesis of unimodality over bimodality is then estimated by the fraction of the  $MAP_1$  taken from the resamples that exceeded the  $MAP_1$  statistic that had been computed for the data.

Resamples of size 116 were obtained in the same fashion from trees  $T_2, \dots, T_5$ . The MAPs were computed for each resample. This was repeated 999 times. Out of 999 repeats, the fraction of the MAP statistics from the resamples that were greater than the corresponding MAPs from the DC 0559-40 galaxy cluster is presented in Table 5. Estimated p-values for the null of  $k$  modes against the alternative of  $k+1$  modes may be read off the diagonal in this table.

From looking at Table 5, it appears that the  $MAP_2$  value of 12.53 is large compared to the  $MAP_2$  of most of the resamples from  $T_2$ . Only 47 of the 999 resamples from  $T_2$  had  $MAP_2$ s bigger than 12.53. This result indicates that there was a relatively large improvement in the fit to the data, from the best fitting bimodal tree  $T_2$  to the best fitting trimodal tree  $T_3$ .

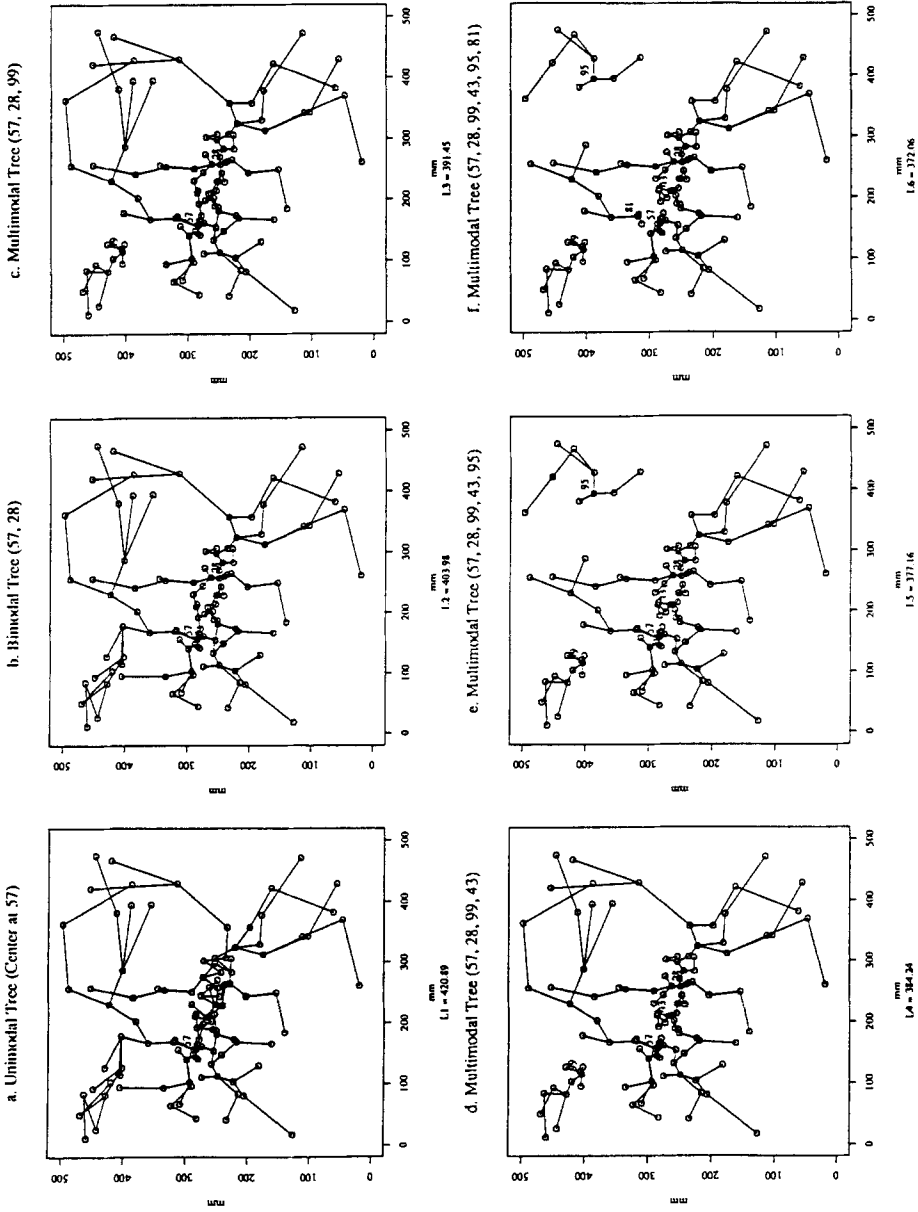


Figure 6. Optimal MAPSTs for Galaxy Clusters DC 0559-40. (Roots are marked by black squares.)

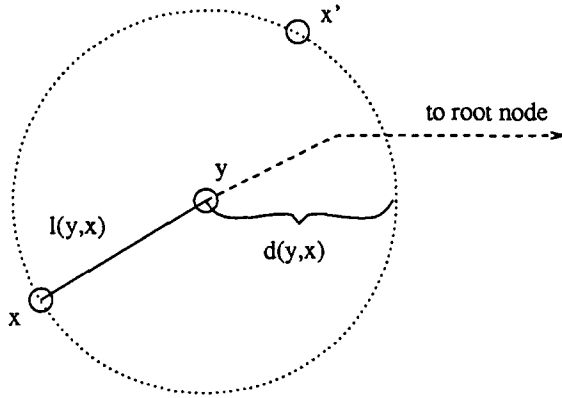


Figure 7. Resampling from Trees.

Table 5: MAP Statistics for DC 0559-40 Galaxy Cluster and Fractions of the MAP Statistics from  $T_1$ ,  $T_2$ ,  $T_3$ ,  $T_4$  and  $T_5$  Resamples Exceeding the Corresponding Galaxy Cluster MAPs

k	1	2	3	4	5
DC 0559-40 MAP <sub>k</sub>	16.90	12.53	7.21	7.08	5.10
$T_1$	0.173	0.044	0.245	0.091	0.263
$T_2$	0.912	0.047	0.221	0.094	0.222
$T_3$	0.966	0.763	0.379	0.118	0.256
$T_4$	0.912	0.800	0.782	0.278	0.343
$T_5$	0.897	0.840	0.886	0.584	0.526

The small fraction (0.044) of  $MAP_2$ s from the  $T_1$  resamples that exceeded the galaxy  $MAP_2$  only highlights the fact that the 12.53 is a large value for the difference in the weight of the best fitting two-root MAPST from the weight of the best fitting three-root MAPST, when sampling from a unimodal tree. After all, we should not expect a large improvement in the fit, from trees with two roots to trees with three roots, when fitting such trees to resamples from unimodal trees. This reasoning also explains the smaller fractions in the upper triangle compared to the lower triangle in the p-value table.

Thus, it seems that the DC 0559-40 galaxy cluster may have three modes. From looking at  $T_3$ , it appears that one mode may be around the upper left corner, while two other modes appear to be in the main group of galaxies. In  $T_5$ , it appears that there could be another mode in the upper right corner. It is conceivable that this group of galaxies may not really be a subcluster, but rather the tail sections of the two cluster groups in the middle region. This speculation is actually suggested in Figure 6c. Therefore, we feel that there is no contradiction between our numerical and graphical results.

## 6. Appendix

### Proof of Theorem 2 (MAPST Algorithm)

Let  $T_k$  be a tree obtained by our algorithm after the  $k$ -th step, and let  $T^*$  be a MAPST for a set of points  $x_1, \dots, x_n$  in some Euclidean space. For two points or less, the MAPST and the MST are the same. The two algorithms give the same tree, and there is no need to prove these trivial cases. Assume that  $n \geq 3$ . Let  $x_1$  be the root node.

Let  $l_{i,j} = l(x_i, x_j)$  be the link joining point  $x_j$  to  $x_i$ , and let  $d_{i,j} = d(x_i, x_j)$  be its length. Suppose  $l_{1,2}$  is the first non-trivial link added by the algorithm. Is  $l_{1,2} \in T^*$ ? If not, add  $l_{1,2}$  to  $T^*$ , forming a cycle  $C$ . Suppose  $l_{1,3}$  is the other link in cycle  $C$  connected to  $x_1$ , while  $l_{2,4}$  is the other link to  $x_2$  in  $C$ . Note that  $x_3$  and  $x_4$  may not be distinct. Since  $l_{1,2}$  is not in  $T^*$ ,  $l_{2,4}$  is connected to  $x_1$  by a path leading through  $l_{1,3}$  and so, by the ascending path property of  $T^*$ ,  $d_{2,4} \geq d_{1,3}$ . By the rule of the algorithm,  $d_{1,2} \leq d_{1,3}$ . Therefore,  $d_{2,4} \geq d_{1,2}$ .

Now, if  $d_{2,4} > d_{1,2}$  then  $T^*$  can be reduced in weight by removing  $l_{2,4}$  contradicting the minimality of  $T^*$ . The ascending path constraint will still be satisfied, and  $T^*$  cannot be a MAPST. Otherwise,  $d_{2,4} = d_{1,2}$  and we can construct a new tree with the same weight as  $T^*$  by replacing  $l_{2,4}$  with  $l_{1,2}$ . This new tree is a MAPST because it has minimal weight and satisfies the ascending path property. For simplicity, we shall name this new tree  $T^*$ .

After the  $k$ -th step of construction, we have  $T_k$  by the algorithm. The algorithm now adds  $x_{k+1} \notin T_k$  at the  $(k+1)$ st step. We shall now show that there exists a MAPST that contains  $T_k$  and  $l_{k,k+1}$ . Assume that the MAPST

includes  $T_k$ , i.e.,  $T_k \subset T^*$ . Let  $C$  be the cycle induced by adding  $l_{k,k+1}$  to  $T^*$ . Let  $z$  be the maximal node in  $C$ . Since  $x_k \in T_k$  has a path to the root in  $T_k$ , necessarily  $z \in T_k$ . Consider Figure 4 again and introduce the points  $a$ ,  $b$ , and  $c$  as positioned in relation to  $x_k$ ,  $x_{k+1}$ , and  $z$  in cycle  $C$ .

Points  $a$  and  $b$  are such that  $l(a,b) \in T^*$ ,  $a \in T_k$ , but  $b \notin T_k$ . Assume that the paths to  $z$  from  $x_k$  and from  $a$  are contained in  $T_k$ . In fact, points  $x_k$ ,  $z$ , and  $a$  may not necessarily be distinct. Point  $c$  can be any point to which  $x_{k+1}$  is connected in  $T^*$ . It is possible that  $b = c$  or even  $a = c$ ; in this case,  $b$  does not exist and  $a \neq x_k$ .

Because  $T^*$  has the ascending path property,  $d(a,b) \leq d(c, x_{k+1})$ . By construction,  $d(a,b) \geq d(x_k, x_{k+1})$ ; otherwise,  $l(a,b)$  would have formed first instead of  $l_{k,k+1}$ . Therefore,  $d(x_k, x_{k+1}) \leq d(c, x_{k+1})$ . If the last inequality were strict, then the weight of  $T^*$  can be further minimized by removing  $l(c, x_{k+1})$  and replacing it by  $l_{k,k+1}$ , which contradicts the minimality of  $T^*$ . So,  $d(x_k, x_{k+1}) = d(c, x_{k+1})$ , and we can swap the corresponding links from  $T^*$ , and obtain a new tree with the same weight as  $T^*$ , but that now includes  $T_k$  and  $l_{k,k+1}$ .

If a point  $b \notin T_k$  does not exist, then  $x_{k+1}$  is connected directly to  $a$  in our illustration. Although it is possible that  $x_k = z$  or  $a = z$ , in this case,  $x_k$  must be distinct from  $a$ . By the algorithm,  $d(x_k, x_{k+1}) \leq d(a, x_{k+1})$ . A strict inequality contradicts the minimality of  $T^*$ . So,  $d(x_k, x_{k+1}) = d(a, x_{k+1})$ , and we can then remove  $l(a, x_{k+1})$  from cycle  $C$  to obtain a MAPST that includes  $l_{k,k+1}$ . Thus, eventually we can find a MAPST that contains all the links in  $T_n$ . ■

### Proof of Theorem 3 (First Algorithm)

We shall consider only the case when  $r = 2$ ; for other values of  $r$ , the proofs are similar. Let  $T_k$  be a graph obtained by our algorithm after the  $k$ -th step, and let  $T^*$  be a MAPST with two roots for a set of points  $x_1, \dots, x_n$  in some space where non-negative distances are defined. Assume that  $n \geq 3$ . Let  $x_1$  and  $x_2$  be the root nodes.

Let  $l_{i,j} = l(x_i, x_j)$  be the link joining point  $x_j$  to  $x_i$ , and let  $d_{i,j} = d(x_i, x_j)$  be its length. Note that  $l_{1,2}$  cannot be formed at this time because both  $x_1$  and  $x_2$  are in the graph already and there are still points that have not been included in the graph. Suppose  $l_{1,3}$  is the first non-trivial link added by the algorithm. Is  $l_{1,3} \in T^*$ ?

If not, add  $l_{1,3}$  to  $T^*$ , forming a cycle  $C$ . If  $x_3$  belongs to the root node  $x_1$  in  $T^*$ , then we are in a situation that is not different from the MAPST with a single root node  $x_1$ . We have shown for this case that  $l_{1,3}$  may replace some link in  $T^*$  without changing its weight.



However, if  $x_3$  belongs to  $x_2$  in  $T^*$ , then cycle  $C$  includes the bridge link  $B^*$  in  $T^*$ . The reason is that introducing  $l_{1,3}$  to  $T^*$  would give two different paths from  $x_1$  to  $x_2$ ; one path includes  $B^*$ , while the other includes  $l_{1,3}$ . Now,  $x_3$  is connected to  $x_2$  in  $T^*$  by some non-increasing path  $p$ . Let  $l_{2,t}$  and  $l_{s,3}$  be the end links of  $p$  from  $x_3$  to  $x_2$ . It is possible that  $s = 2$  and  $t = 3$ . By the ascending path property of  $T^*$ ,  $d_{s,3} \geq d_{2,t}$ . Since  $l_{1,3}$  is the first non-trivial link,  $d_{2,t} \geq d_{1,3}$ . Therefore,  $d_{s,3} \geq d_{1,3}$ . If the last inequality were strict, then removing  $l_{s,3}$  from cycle  $C$  would produce an ascending path spanning tree that weighs less than  $T^*$ , contradicting the minimality of  $T^*$ . So,  $d_{s,3} = d_{1,3}$ , and we can replace  $l_{s,3}$  with  $l_{1,3}$  in  $T^*$  to produce an ascending path spanning tree with root nodes  $x_1$  and  $x_2$  having the same weight as  $T^*$ , but with  $l_{1,3}$ . Consider this new tree as the current  $T^*$ .

After the  $k$ -th step of construction, we have  $T_k$  by the algorithm. The algorithm now adds  $x_{k+1} \notin T_k$  at the  $(k+1)$ -st step. We shall show there exists a MAPST that contains  $T_k$  and  $l_{k,k+1}$ . Assume MAPST  $T^*$  includes  $T_k$ , i.e.,  $T_k \subset T^*$ . Without loss of generality, let  $x_k \in T_k$  belong to  $x_1$ . Let  $C$  be the cycle induced by adding  $l_{k,k+1}$  to  $T^*$ . Let  $z_1$  be the maximal point in  $C$  with respect to the root  $x_1$ , and  $z_2$ , the maximal point in  $C$  with respect to the root  $x_2$ . Recall that the maximal point in a cycle  $C$  with respect to a root node  $x_0$  is the point in  $C$  included in all paths from any point in  $C$  to the root  $x_0$ .

If  $x_{k+1}$  belongs to  $x_1$  in  $T^*$ , then all nodes in  $C$  belong to  $x_1$  in  $T^*$ , and we are again in the case similar to a MAPST with one root node. We have shown that in this situation, we can replace some link in  $C$  with  $l_{k,k+1}$ , and obtain a MAPST containing  $l_{k,k+1}$ . If  $x_{k+1}$  belongs to  $x_2$ , consider Figure 8, and introduce the points  $a$ ,  $b$ , and  $c$  as positioned in relation to  $x_k$ ,  $x_{k+1}$ ,  $z_1$ , and  $z_2$  in cycle  $C$ . That is, let  $p \subset T^*$  be a path from  $x_{k+1}$  to  $x_2$  that includes  $l(c, x_{k+1})$  and  $l(a, b)$  such that  $a \in T_k$  but  $b \notin T_k$ . Point  $c$  is any point to which  $x_{k+1}$  is attached in  $T^*$ .

Points  $x_k$ ,  $z_1$ , and the nearer end point of the bridge link may not necessarily be distinct points. The same is true for points  $a$ ,  $z_2$ , and the other end point of the bridge link. Assume that the paths from  $x_k$  to  $z_1$  and from  $a$  to  $z_2$  are in  $T_k$ . If point  $b$  exists, then it may be true that  $b = c$ ; otherwise,  $a = c$ . At this point, the argument is the same as the last part of the proof for the case of a single root node.

Because  $T^*$  has the ascending path property,  $d(a, b) \leq d(c, x_{k+1})$ . By construction,  $d(a, b) \geq d(x_k, x_{k+1})$ ; otherwise, the link between  $a$  and  $b$  would have formed first instead of  $l_{k,k+1}$ . Therefore,  $d(x_k, x_{k+1}) \leq d(c, x_{k+1})$ . If the last inequality were strict, then the weight of  $T^*$  can be further minimized by removing  $l(c, x_{k+1})$ , and replacing it by  $l_{k,k+1}$ , contradicting the minimality of  $T^*$ . So,  $d(x_k, x_{k+1}) = d(c, x_{k+1})$ , and we can swap the corresponding links from  $T^*$ , and obtain a new tree with the same weight as  $T^*$ , but that now includes  $T_k$  and  $l_{k,k+1}$ .



If no point  $b$  exists, then  $x_{k+1}$  is connected to  $a$  in the diagram. By the algorithm,  $d(x_k, x_{k+1}) \leq d(a, x_{k+1})$ . A strict inequality leads to a contradiction of the minimality of  $T^*$ . Therefore,  $d(x_k, x_{k+1}) = d(x_{k+1}, a)$ . We can then replace  $l(a, x_{k+1})$  in  $T^*$  by  $l(x_k, x_{k+1})$  to obtain a new MAPST that contains  $l_{k,k+1}$ . Thus, eventually we can find a MAPST that includes all the first  $n - 2$  non-bridge links.

Finally, suppose  $B$  is the bridge link found by the algorithm; i.e.,  $B$  is a bridge link of minimal length that satisfies the ascending path constraint, and  $B^*$  is the bridge link in  $T^*$ . If  $B \neq B^*$  then add  $B$  to  $T^*$ , forming a cycle. Let  $|B|$  be the length of  $B$ ; define  $|B^*|$  likewise. By the rule of the algorithm,  $|B| \leq |B^*|$ , because  $B^*$  is a bridge link connecting the subtrees  $T(x_1)$  and  $T(x_2)$  that satisfies the ascending path property. If the inequality were strict then  $T^*$  cannot be minimal. Therefore,  $|B| = |B^*|$ . Thus, we can replace  $B^*$  in  $T^*$  by  $B$ , and manufacture an ascending path spanning tree that includes all the links obtained through our First Algorithm with the same weight as  $T^*$ . ■

#### Proof of Lemma 4 (Ascending Paths Lemma)

Let  $T$  be a tree constructed through the Second Algorithm. Let  $T(x_1), \dots, T(x_m)$  be single root MAPSTs with root nodes  $x_1, \dots, x_m$ , respectively, that were used for deriving  $T$ . For some path in  $T$  that contains no bridge link, let  $r, s$ , and  $t$  be successive points on the way to a root node. Is  $d(s, r) \geq d(t, s)$ ? If both  $l(s, r)$  and  $l(t, s)$  are in the same tree  $T(x_i)$ , for some  $i \in \{1, \dots, m\}$ , then the ascending path property of  $T(x_i)$  guarantees that  $d(s, r) \geq d(t, s)$ .

Suppose  $l(s, r) \in T(x_i)$  but  $l(t, s) \notin T(x_i)$ . Then,  $l(t, s) \in T(x_{i'})$  for some  $i' \neq i$ . By the ascending path property of  $T(x_i)$ ,  $d(s, r) = \rho_{x_i}(r) \geq \rho_{x_i}(s)$ . Since  $l(t, s) \in T(x_{i'})$ ,  $\rho_{x_i}(s) \geq \rho_{x_{i'}}(s) = d(t, s)$ . Therefore,  $d(s, r) \geq d(t, s)$ . ■

#### Proof of Lemma 5 (Bridge Links Lemma)

Let  $B$  be a bridge link connecting the subtrees  $T(x_1)$  and  $T(x_2)$  with roots  $x_1$  and  $x_2$ , respectively. Suppose that the ends of  $B$  are  $y \in T(x_1)$  and  $v_{x_2}(y) \in T(x_2)$ ,  $v_{x_1}(y) \neq v_{x_2}(y)$ . By Lemma 4 (Ascending Paths Lemma),  $T(x_2)$  satisfies the ascending path constraint. Then,

$$|B| = \rho_{x_2}(y) \geq \min_{i=1,2} \rho_{x_i}(v_{x_i}(y)).$$

By Step 2 of the Second Algorithm,  $|B| = \rho_{x_2}(y) \geq \rho_{x_1}(y)$ . See Figure 9. ■

### First Part of Proof of Theorem 6 (Second Algorithm: Non-bridge Links)}

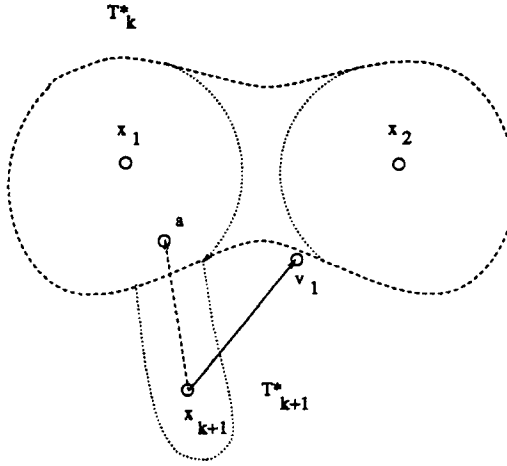
Consider only the non-bridge links for now; we shall prove the optimality of the choice for bridge links in the second part of the proof. For the non-bridge links, we shall only prove the case when  $r = 2$ ; for  $r > 2$ , the proof is similar. Starting with  $x_1$  and  $x_2$  as roots, let  $T_k^*$  be a graph obtained by the First Algorithm after  $k$  points have been included; denote by  $T^*$  the completed MAPST. Let  $T$  be an ascending path spanning tree by the Second Algorithm. Let  $T(x_1)$  and  $T(x_2)$  be the MAPSTs with roots  $x_1$  and  $x_2$ , respectively, that were used for deriving  $T$ . Assume that all distances between points,  $d_{i,j} = d(x_i, x_j)$ , are different for  $i \neq j$ ; the equal distance cases may be handled by a continuity argument. We know that the first two links in  $T^*$  and  $T$  will be the same, i.e.,  $x_1$  and  $x_2$  are connected to themselves. In both  $T^*$  and  $T$ , each root node is connected to itself.

Suppose the first non-trivial link in  $T^*$  were  $l_{1,3}$ . By the rule of the First Algorithm,  $d_{1,3} \leq d_{1,j}$  whenever  $j \neq 1, 2$ . First, consider the case when  $d_{1,3} < d_{1,2}$ . Then in  $T(x_1)$ ,  $v_{x_1}(x_3) = x_1$  and  $\rho_{x_1}(x_3) = d_{1,3}$ . In  $T(x_2)$ ,  $x_3$  is connected to the root  $x_2$  by a path whose end links are  $l_{s,3}$  and  $l_{2,t}$ . Since  $d_{1,3} < d_{1,2}$ ,  $x_s$  cannot be  $x_1$  as the ascending path property will not be satisfied in  $T(x_2)$ . In this path,  $\rho_{x_2}(x_3) = d_{s,3} \geq d_{2,t}$ . From  $T^*$ ,  $d_{1,3} \leq d_{2,j}$  whenever  $j \neq 2$ , then,  $\rho_{x_1}(x_3) = d_{1,3} \leq d_{2,t} \leq d_{s,3} = \rho_{x_2}(x_3)$ . Therefore,  $\rho_{x_1}(x_3) \leq \rho_{x_2}(x_3)$ , and  $x_3$  is connected to  $v_{x_1}(x_3) = x_1$  by the Second Algorithm. Thus,  $l_{1,3} \in T$ .

Now, consider the case when  $d_{1,3} > d_{1,2}$ . In making  $T(x_1)$ , first,  $x_2$  would be joined with starting node  $x_1$ . With  $x_2$  included, according to  $T^*$ ,  $l_{1,3}$  would form next. Then,  $v_{x_1}(x_3) = x_1$ . In  $T(x_2)$ , starting with  $x_2$ ,  $l_{2,1}$  would be the first link formed followed by  $l_{1,3}$ . So,  $v_{x_2}(x_3) = x_1$ . This result means that  $\rho_{x_1}(x_3) = \rho_{x_2}(x_3)$ . Therefore,  $l_{1,3} \in T$ .

Suppose  $T_k^* \subset T$ ; is  $T_{k+1}^* \subset T$ ? Let  $x_{k+1}$  be the  $k+1$ -st point included by the First Algorithm. Suppose  $\rho_{x_1}(x_{k+1}) \leq \rho_{x_2}(x_{k+1})$  such that in  $T$ ,  $x_{k+1}$  is connected to  $v_1 = v_{x_1}(x_{k+1})$  by link  $l(v_1, x_{k+1})$ . Is  $l(v_1, x_{k+1}) \in T_{k+1}^*$ ? If not, then in  $T_{k+1}^*$ ,  $x_{k+1}$  is connected to some point  $a \in T_k^*$ , where  $a \neq v_1$ .

We will now show that  $d(a, x_{k+1}) < d(v_1, x_{k+1})$ . Refer to Figure 10. The First Algorithm chooses the shortest link from a point  $x \notin T_k^*$  to a point  $x_i \in T_k^*$  that satisfies the ascending path property. Then (a),  $d_{i,j} < d(a, x_{k+1})$ , for all links  $l_{i,j} \in T_k^*$ , and (b),  $d(a, x_{k+1}) < d(x_i, x)$ , for all  $x_i \in T_k^*$  and for all  $x \notin T_k^*$ , whenever  $l(x_i, x)$  satisfies the ascending path constraint. By Lemma 4,  $l(v_1, x_{k+1})$  satisfies the ascending path property. Thus, if  $v_1 \in T_k^*$ , then  $d(a, x_{k+1}) < d(v_1, x_{k+1})$ . Otherwise, if  $v_1 \notin T_k^*$ , then  $v_1$  is connected in  $T$  to a root node by some path  $p$ . Starting from  $v_1$  and going toward a root node, let  $l(b, c)$  be the first link in  $p$  such that  $c \notin T_k^*$  but  $b \in T_k^*$ . It is possible that


 Figure 10.  $T_k^* \subset T$  and  $T_{k+1}^*$ .

$c = v_1$ . Then,  $d(a, x_{k+1}) < d(b, c) \leq d(v_1, x_{k+1})$ .

Now consider  $T(x_1)$ . In  $T(x_1)$ ,  $x_{k+1}$  is connected to  $v_1 \neq a$ . Since  $d(a, x_{k+1}) < d(v_1, x_{k+1})$  then  $x_{k+1}$  cannot be connected to  $a$ ; otherwise, the construction of  $T(x_1)$  would require that  $v_1 = a$ . So,  $l(a, x_{k+1}) \notin T(x_1)$ . Then, adding  $l(a, x_{k+1})$  to  $T(x_1)$  forms a cycle  $C$ . First, assume that  $l(a, x_{k+1})$  satisfies the ascending path constraint in  $T(x_1)$ , i.e., there is a point  $e \neq x_{k+1}$  and a link  $l(e, a) \in T(x_1)$  such that point  $a$  is between  $e$  and  $x_{k+1}$  on the path to the root node  $x_1$  and, therefore,  $d(e, a) \leq d(a, x_{k+1})$ . Because  $d(a, x_{k+1}) < d(v_1, x_{k+1})$ , then  $T(x_1)$  cannot be minimal; we can produce a MAPST with root  $x_1$  that weighs less than  $T(x_1)$  by removing  $l(v_1, x_{k+1})$  from  $C$ . Therefore,  $a = v_1$ , and  $T_{k+1}^* \subset T$ .

Suppose  $l(a, x_{k+1})$  does not satisfy the ascending path property in  $T(x_1)$ ; that is,  $a$  is connected to some point  $e \neq x_{k+1}$  by  $l(e, a) \in T(x_1)$  such that  $d(e, a) > d(a, x_{k+1})$ . Since  $a \in T_k^* \subset T$ , then there exists an  $e^* \in T_k^*$  and  $l(e^*, a) \in T_k^*$  such that  $d(e^*, a) < d(a, x_{k+1})$  in  $T^*$ . Then,  $e^* \neq e$ , and  $a$  is connected to  $e^*$  in  $T(x_2)$  by  $l(e^*, a)$ . Let  $v_2 = v_{x_2}(x_{k+1})$ . So far, we know that

$$\begin{aligned} d(e^*, a) &< d(a, x_{k+1}) < d(v_1, x_{k+1}) = \rho_{x_1}(x_{k+1}) \\ &\leq \rho_{x_2}(x_{k+1}) = d(v_2, x_{k+1}). \end{aligned}$$

Then,  $T(x_2)$  cannot be minimal by the following argument. Since  $a$  is connected to  $e^* \neq x_{k+1}$  ( $e^* \in T_k^*$ ,  $x_{k+1} \notin T_k^*$ ) and  $x_{k+1}$  is connected to  $v_2 \neq a$  (because  $d(a, x_{k+1}) < d(v_2, x_{k+1})$ ), then adding  $l(a, x_{k+1})$  to  $T(x_2)$  forms a cycle. In this cycle,  $d(e^*, a) < d(a, x_{k+1}) < d(v_2, x_{k+1})$ . If we remove  $l(v_2, x_{k+1})$  from this cycle, then we have an ascending path spanning tree

with root  $x_2$  that weighs less than  $T(x_2)$ . So,  $a = v_2$ , which implies that  $\rho_{x_1}(x_{k+1}) > \rho_{x_2}(x_{k+1})$  by the following inequalities:

$$\begin{aligned}\rho_{x_1}(x_{k+1}) &= d(v_1, x_{k+1}) \\ &> d(a, x_{k+1}) = d(v_2, x_{k+1}) = \rho_{x_2}(x_{k+1}).\end{aligned}$$

This result contradicts the assumption that  $\rho_{x_1}(x_{k+1}) \leq \rho_{x_2}(x_{k+1})$ , and so,  $a = v_1$  and  $T_{k+1}^* \subset T$ . ■

### Second Part of Proof of Theorem 6 (Second Algorithm: Bridge Links)

So far, we have shown that  $T_n^* \subset T$ ; i.e., all the non-bridge links in  $T^*$  (First Algorithm tree) are in  $T$  (Second Algorithm tree). Consider the case of two roots; let  $B$  and  $B^*$  be the bridge links in  $T$  and  $T^*$ , respectively. If  $B \neq B^*$ , add  $B$  to  $T^*$  forming a cycle. Now,  $|B^*| \leq |B|$ ; otherwise,  $T^*$  is not minimal.

By the First Algorithm,  $B^*$  satisfies the ascending path constraint. Then, there exists an ascending path from  $B^*$  to a root node, say  $x_1$ . Let  $T(x_1)$  be the MAPST with root at  $x_1$  from which  $T$  is partly based. If  $|B^*| < |B|$ , then  $B^* \in T(x_1)$  for  $T(x_1)$  to be minimal. This finding contradicts the choice of the bridge link  $B$  in the Second Algorithm;  $B$  is one of the shortest bridge links. Therefore,  $|B^*| = |B|$ . By Lemma 5,  $B$  satisfies the ascending path constraint with respect to the non-bridge links already in place. Then, we can replace  $B^*$  by  $B$  to obtain a spanning tree with the same total weight as  $T^*$ , satisfying the ascending path property.

The Second Algorithm search for bridge links must be shown to meet the global minimum weight of the tree as required by the First Algorithm. The case  $r > 2$  is a bit different from the preceding case  $r = 2$ . We shall consider only the case when  $r = 3$ , with subtrees  $T(x_1)$ ,  $T(x_2)$ , and  $T(x_3)$ . By Lemma 5, all the bridge links satisfy the ascending path property with respect to all the non-bridge links already in place. Thus, we only have to worry about the minimality of the links, and not the ascending path constraint, when choosing among bridge links to connect subtrees.

We defined the distance between  $T(x_1)$  and  $T(x_2)$  to be the length of the smallest of the links from  $x \in T(x_2)$  to  $v_{x_1}(x)$ , and from  $y \in T(x_1)$  to  $v_{x_2}(y)$ . Similarly, the distance between  $T(x_2)$  and  $T(x_3)$  is the length of the shortest link among the links between a point  $x \in T(x_2)$  and  $v_{x_3}(x)$ , and between  $z \in T(x_3)$  and  $v_{x_2}(z)$ , and so on. In the Second Algorithm, we would base a MST on these three distances between pairs of  $T(x_1)$ ,  $T(x_2)$ , and  $T(x_3)$ . The two smallest such distances would then appear in the MST. Thus,

the MST connects the subtrees by the two distances with smallest total length. This condition is accomplished in Step 4 of the Second Algorithm. There is a possible difficulty in that the shortest link found may not be a bridge link between two subtrees. For example, the shortest link found between  $x \in T(x_2)$  and  $v_{x_1}(x)$  might not connect  $T(x_1)$  to  $T(x_2)$ , if  $v_{x_1}(x)$  belongs  $T(x_3)$ . When this situation arises, however, we set the distance between  $T(x_1)$  and  $T(x_2)$  to infinity. The two smallest bridge links are those that connect  $T(x_1)$  to  $T(x_3)$ , and  $T(x_2)$  to  $T(x_3)$ . The nonexistent link is unnecessary. ■

### Proof of Theorem 8 (Existence and Uniqueness of the Maximal Node)

The *maximal node*  $z$  in a cycle  $C$  is the node included in all paths in  $T^*$  to the root node from any point in  $C$ . To show that  $z$  exists, consider the intersection of all paths in  $T^*$  from any node in  $C$  to the root; we need to show that there is a unique node  $z \in C$  in that intersection. Let  $p_x$  denote a path from  $x \in C$  to the root node. Suppose  $x' \in C$ ,  $x' \in p_x$ , and  $p_{x,x'} \subset C$  is the path from  $x$  to  $x'$  in  $C$ . Then  $p_x = p_{x'} \cup p_{x,x'}$  and the intersection of all paths  $p_x$  from all points  $x \in C$  is contained in the intersection of all such paths  $p_x$  omitting  $p_{x,x'}$  but not  $x'$ . Therefore, the intersection of all such paths is the intersection of all paths  $p_x$  for which there exists no  $x' \in C$ ,  $x' \in p_x$ , and  $p_{x,x'} \subset C$ . Suppose there are two distinct points  $x$  and  $x'$  in  $C$  such that  $p_x$  and  $p_{x'}$  satisfy the condition that there exists no point  $y \in C$  in either  $p_x$  or  $p_{x'}$  such that  $p_{x,y} \subset C$  or  $p_{x',y} \subset C$ . Since  $x \in C$  and  $x' \in C$ , they are connected by points in  $C$ , and either  $p_x$  or  $p_{x'}$  will include some of these points, which contradicts the condition that paths  $p_x$  and  $p_{x'}$  have no other points in  $C$ . It follows that  $x = z$  is uniquely determined. ■

### References

- DRESSLER, A. (1980), "A Catalog of Morphological Types in 55 Rich Clusters of Galaxies," *Astrophysical Journal Supplement Series*, 42, 565--609.
- GIACOMELLI, F., WIENER, J., KRUSKAL, J. B., POMERANZ, J. W., and LOUD, A. V. (1971), "Subpopulations of Blood Lymphocytes as Demonstrated by Quantitative Cytochemistry," *Journal of Histochemistry and Cytochemistry*, 19, 426--433.
- HARTIGAN, J. A., and HARTIGAN, P. M. (1985), "The Dip Test of Unimodality," *Annals of Statistics*, 13, 70--84.
- HARTIGAN, J. A. (1988), "The Span Test for Unimodality," in *Classification and Related Methods of Data Analysis*, Ed. H. H. Bock, Amsterdam: North-Holland, 229--236.
- HARTIGAN, J. A. and MOHANTY, S. (1992), "The RUNT Test for Multimodality," *Journal of Classification*, 9, 63--70.
- PRIM, R. C. (1957), "Shortest Connection Networks and some Generalizations," *Bell System Technical Journal*, 36, 1389--1401.

- SILVERMAN, B. W. (1981), "Using Kernel Density Estimates to Investigate Multimodality," *Journal of the Royal Statistical Society, Series B*, 43, 97--99.
- SILVERMAN, B. W. (1986), *Density Estimation for Statistics and Data Analysis*, New York: Chapman and Hall.
- ZAHN, C. T. (1971), "Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters," *IEEE Transactions on Computers*, C-20, 68--86.