CS195Z: Problem Set 3

Due: Wednesday, 4/22/09
Questions: email Crystal (clkahn)

April 16, 2009

1. Give a lower bound on the reversal distance between the following two signed permutations using (a) the number of breakpoints and (b) the number of cycles in the cycle decomposition of the breakpoint graph.

$$\begin{array}{l} +1\ +2\ +3\ +4\ +5\ +6\ +7\ +8\ +9 \\ +1\ -3\ +2\ +4\ +9\ +7\ +5\ +6\ +8 \end{array} \tag{1}$$

2. A permutation on a set $\{1, 2, \ldots, n\}$ has a number of breakpoints, with respect to the identity permutation, equal to the number of adjacent pairs of numbers that are not successive in the identity permutation, i.e. are not of the form i and $i + 1$. For a fixed n, let $\Gamma(b)$ be the number of unsigned permutations on $\{1, 2, \ldots, n\}$ with exactly b breakpoints. How does $\Gamma(b)$ change as b increases? Give some intuition for your answer.

3. Characterize an example for which the CAST (Cluster Affinity Search Technique) algorithm does not converge.

4. Consider the trees shown in Figure 1.

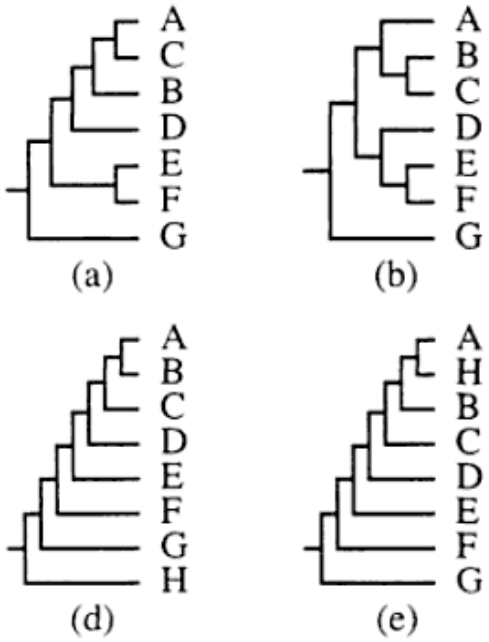

Figure 1:

(a) Give the strict consensus tree for rival trees (a) and (b).

(b) Give the strict consensus tree for rival trees (d) and (e).

5. You have been introduced to several types of consensus trees in class. For a tree on a set of taxa, two groups of taxa are *combinable* if either (1) they have no taxa in common, (2) they are identical, or (3) one group is a proper subset of the other. The combinable-component consensus tree is defined by the set of all combinable groups (i.e., each group retained in the consensus is equal to or combinable with all groups of every rival tree). Under what conditions are the strict and combinable-component consensus methods equivalent?

6. Use Aho's BUILD algorithm to find the Adams consensus tree of the three rival trees shown in Figure 2.

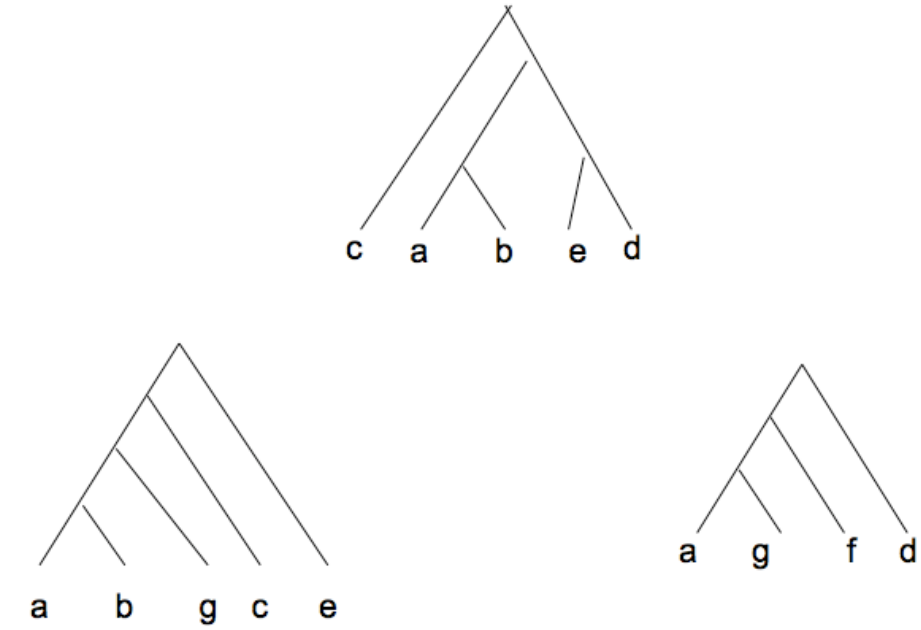

Figure 2:

7. The Threshold Number of Misclassifications (TNoM) score was defined by Ben-Dor et. al as a measure for selecting informative genes for binary classification. Suppose we have the following gene expression data for a set of genes $G = \{g_1, g_2, \ldots, g_n\}$ and a set of samples $S = \{s_1, s_2, \ldots, s_m\}$ where s_1 and s_3 are afflicted by some disease and s_2 and s_4 are not:

	s_1	s_2	s_3	s_4
g_1	0.1	0.9	0.2	0.6
g_2	0.5	0.6	0.7	0.1
g_3	0.2	0.5	0.3	0.7
g_4	0.1	0.4	0.5	0.8
g_5	0.5	0.5	0.2	0.7

We define the rank vector v_{g_1} for g_1 by sorting its row in the matrix in increasing order: s_1, s_3, s_4, s_2 and replacing the diseased samples with a '-' and the healthy samples with a '+'. Thus, $v_{g_1} = (-, -, +, +)$. For such a vector, the TNoM score is determined by the split that maximizes the number of -'s on one side of the split and the number of +'s on the other side of the split. See class notes for the precise definition. In our example, the best split is between the second and third positions and the TNoM score is zero.

(a) Compute the rank vector and TNoM score for genes g_2 through g_5.

(b) We assess the statistical significance of TNoM(v_g) by comparing it to vectors with the same number of +'s and -'s. For a rank vector v_{g_i} of a gene g_i, let p be the number of +'s in the vector and q be the number of -'s. We wish to compute

$$Pr[TNoM(L) \leq TNoM(v_g)], \tag{2}$$

where L is drawn uniformly from the set of all vectors in $\{+, -\}^m$ with p +'s and q -'s. How many such vectors L are there?

8. (BONUS) $Pr[TNoM(L) \leq s]$ can be computed by considering a random walk in R^2 that begins at $(0, 0)$ and moves one position to the right and one position up/down for each entry in L. Thus, each

entry i in L corresponds to a point $(x(i), y(i)) = (i, y(i-1) + \operatorname{sign} L(i))$ Note that for given (p, q) all paths are bounded by the paths of the two perfect classifiers ($L = (+, +, \ldots, +)$ or $L = (-, -, \ldots, -)$). In addition, we have the following lemma:

Lemma 1. *A vector has a TNoM score $\leq s$ iff the corresponding path crosses either of the lines $y = p - s$ or $y = s - q$.*

(a) Prove this lemma.

(b) To compute a p-value of a TNoM score, we can count how many of the paths with the same (p, q) have at most the same TNoM score. Using the reflection principle for random walks, calculate this number of paths. (Hint: to count exactly, you might have to use the inclusion-exclusion principle. Partial credit will be given for a good upper bound on the count.)