BCB BCB/GDCB/STAT/COM S 568 Spring 2011

Homework 5

February 24, 2011

1. For a bifurcating tree with n leaves, determine the number of pairwise distances between the leaves and the number of branches in the tree. Prove that the number of topologically different trees is $\frac{(2n-5)!}{2^{n-3}(n-3)!}$ for $n \geq 3$.

   Solution:

   The number of distances is $\binom{n}{2}$. The number of branches is $2n-3$. All results are easily proved by induction, starting with $n = 3$.

2. A conventional measure of sequence divergence is the number of substitutions observed for two aligned sequences. Because multiple substitutions and back substitutions at a given site cannot be observed, the observed number of substitutions may under-estimate the actual number of substitutions if the two sequences derive from a common ancestor.

   Derive the "Poisson correction" for the mean number of substitutions per site as a function of the observed fraction of identical residues.

   Solution:

   Let λ be the substitution rate, here assumed to be equal for all sites and amino acids (see N. Grishin [1995] J. Mol. Evol. 41: 675-679 for an elegant and comprehensive treatment under less stringent assumptions). Let $P_0(t)$ be the probability of no substitution occuring at a site in time t. With the conventional assumptions of independence between non-overlapping time intervals and $1 - P_0(h) = \lambda h$ for small h, it follows

$$P_0(t+h) = P_0(t)P_0(h) = P_0(t)(1 - \lambda h),$$

   or

$$P_0'(t) = -\lambda P_0(t)$$

   with solution

$$P_0(t) = e^{-\lambda t}.$$

   For a sequence of length n, let Δr represent the number of substitutions accumulated in time Δt. We can estimate $\Delta r = n(1 - e^{-\lambda \Delta t}) \simeq n\lambda\Delta t$, i.e. $\Delta d = \frac{\Delta r}{n} = \lambda \Delta t$ with solution $d = \lambda t$. Here d is the distance measure (substitutions per site). Finally, q, the fraction of identical sites, can be set equal to $P_0(t)$ as long as q is not too small, and thus

$$q \simeq e^{-\lambda t} = e^{-d}$$

   or

$$d = -\ln q.$$

3. (Review Problem) Given two sequences $A = a(1)a(2)\dots a(m)$ and $B = b(1)b(2)\dots b(n)$, an alignment of A and B can be represented as an ordered pair of integers $(i_1, j_1), (i_2, j_2)\dots(i_k, j_k)$, where $1 \leq i_1 < i_2 < i_3 < \cdots < i_k \leq m$ and $1 \leq j_1 < j_2 < j_3 < \cdots < j_k \leq n$.

   In this representation, $a(i_x)$ is matched with $b(j_x)$, and gaps occur when matched letters are not consecutive in either sequence. Zuker (1991, J. Mol. Biol. 221: 403) assigned substitution scores $s(a(i), b(j))$ and gap penalties $w_k = -\alpha - \beta k$, where $k > 0$ is the size of a gap defined as the sum of unaligned residues in either sequence between matched letters; i.e., the gap between aligned pairs $a(i_{x-1})$, $b(j_{x-1})$ and $a(i_x)$, $b(j_x)$ is of size $i_x - i_{x-1} + j_x - j_{x-1} - 2$.

   The optimal score S_{ij} for aligning the prefixes $a(1)a(2)\dots a(i)$ and $b(1)b(2)\dots b(j)$ can be calcuated recursively as follows:

$$S_{ij} = \max\{D_{ij}, G_{ij}\},$$

   where

$$D_{ij} = S_{i-1,j-1} + s(a(i), b(j))$$

   and

$$G_{ij} = \max\{S_{i-k,j-l} + w_{k+l}\}, \qquad k = 0, 1, 2, \dots, i; l = 0, 1, 2, \dots, j; \max\{k, l\} > 0,$$

   with appropriate initial conditions.

   Show that the algorithm can produce the optimal score S_{mn} in $O(mn)$ operations.

Solution:

Write

$$G_{ij} = \max \left\{ \begin{array}{ll} S_{i-1,j} + w_1 & \\ S_{i,j-1} + w_1 & \\ \max\{S_{i-k,j-l} + w_{k+l}\} & k = 1, 2, \dots, i; l = 0, 1, 2, \dots, j \\ \max\{S_{i-k,j-l} + w_{k+l}\} & k = 0, 1, 2, \dots, i; l = 1, 2, \dots, j \end{array} \right\}$$

By re-indexing, this is equivalent to

$$G_{ij} = \max \left\{ \begin{array}{ll} S_{i-1,j} + w_1 & \\ S_{i,j-1} + w_1 & \\ \max\{S_{i-1-k,j-l} + w_{k+1+l}) & k = 0, 1, 2, \dots, i; l = 0, 1, 2, \dots, j; \max\{k, l\} > 0 \\ \max\{S_{i-k,j-1-l} + w_{k+l+1}) & k = 0, 1, 2, \dots, i; l = 0, 1, 2, \dots, j; \max\{k, l\} > 0 \end{array} \right\}$$

where $w_{k+1+l} = w_{k+l+1} = w_{k+l} - \beta$. Thus,

$$G_{ij} = \max \left\{ \begin{array}{l} S_{i-1,j} + w_1 \\ S_{i,j-1} + w_1 \\ G_{i-1,j} - \beta \\ G_{i,j-1} - \beta \end{array} \right\}$$

Thereby, foreach S_{ij} and G_{ij} the update depends only on the adjacent cells in the optimal score matrix, making the complexity of the algorithm of the order of the sequence lengths product.