Notes for Bioinformatics I, Oct. 4, 2007

Sorting by Reverals; Breakpoints

We want an example where no reversal decreases the number of breakpoints. Note that if a reversal decreases the number of breakpoints, then after the reversal at least one end of the reversed segment does not have a breakpoint (why?); in other words, the reversal must bring together a pair of numbers that differ by 1. Also note that if n and $n+1$ are not adjacent in a permutation, then there are precisely two ways to bring them together with a single reversal, namely "bring n to $n+1$" or "bring $n+1$ to n". (The reversed segments have the same length and are offset by 1 position.) For the permutation 0156723489 (3 breakpoints), we can consider all reversals that bring together two separated numbers that differ by 1. The possibiliites are:

bring 1 to 2: $0\underline{1567}23489 \Rightarrow 0765123489$ (3 breakpoints)
bring 2 to 1: $01\underline{5672}3489 \Rightarrow 0127653489$ (3 breakpoints)
bring 4 to 5: $015\underline{67234}89 \Rightarrow 0154327689$ (3 breakpoints)
bring 5 to 4: $01\underline{56723}489 \Rightarrow 0132765489$ (3 breakpoints)
bring 7 to 8: $0156\underline{7234}89 \Rightarrow 0156432789$ (3 breakpoints)
bring 8 to 7: $01567\underline{2348}9 \Rightarrow 0156784329$ (3 breakpoints)

Weight Matrices

Specific Plan:

Given: N DNA sequences, each of length L.
Method:

1. Form an array of numbers, with four rows labeled $\texttt{A}$, $\texttt{C}$, $\texttt{G}$ and $\texttt{T}$, and L columns numbered 1 to L. The value in row r of column i is the number of times that nucleotide r appears as the i^{th} entry in one of the sequences.
2. Divide each of the $4L$ numbers by N. Then the sum of entries in any specified column is 1.0. Let $f_i(r)$ denote the value in row r of column i. For example, $f_3(\texttt{A})$ intuitively estimates the probability that the 3rd entry of the sequence is $\texttt{A}$.
3. Replace each $f_i(r)$ by $\log_2(f_i(r)/p(r))$, where $p(\texttt{A}) = p(\texttt{T}) = 0.29$, and $p(\texttt{C}) = p(\texttt{G}) = 0.21$. (But what if $f_i(r) = 0$?)

Given any DNA sequence x of length L, we can use this *weight matrix* to associate a number $w(x)$, where $w(x) > 0$ if and only if x looks more like one of the original N sequences than like a typical mammalian N-mer.

General Plan:

Given: Two training sets of DNA sequences, a "positive set" P and a "negative set" Q.
Goal: Discriminate P and Q. That is, develop a method that can decide whether an arbitrary DNA sequence is more P-like or more Q-like.
Method:

1. Select a variety of probabilistic sequence model that is appropriate for P and one appropriate for Q.
2. Estimate model parameters for the two models. For any DNA sequence x, let $\Phi_P(x)$ and $\Phi_Q(x)$ denote the probability that the models for P and Q (respectively) will generate x, assuming that they generate a sequence of the same length as x.
3. For any x, the number $\log(\Phi_P(x)/\Phi_Q(x))$ exceeds 0 if and only if x is more P-like than Q-like.