2008 IIB 4G1 COMPUTATIONAL AND SYSTEMS BIOLOGY DR J GONCALVES/ DR S EGLEN

Question 1

a) Assume we know that x and y are very similar

- If the optimal alignment of x and y has few gaps, then path of the alignment will be close to diagonal

Assumption: # gaps(x, y) < k(N) (say N>M)

xi
| implies | i - j | < k(N)
yj

We can align x and y more efficiently:

Time, Space: O(N × k(N)) << O(N2)

Algorithm:

Initialization:

F(i,0), F(0,j) undefined for i, j > k

Iteration:

For i = 1…M
  For j = max(1, i - k)...min(N, i+k)

$$F(i,j) = \max \begin{cases} F(i-1, j-1) + s(x_i, y_j) \\ F(i, j-1) - d, \text{ if } j > i - k(N) \\ F(i-1, j) - d, \text{ if } j < i + k(N) \end{cases}$$

Termination: same

Easy to extend to the affine gap case

b) The Smith-Waterman algorithm provides an exact algorithmic solution to the problem of computing optimal local alignments. However, its quadratic time complexity has motivated the creation of rapid heuristic local alignments tools. A basic idea behind all heuristic algorithms is to focus only on those regions which share some patterns, assumed to witness (or to *hit*) a potential similarity.

In doing so, it is close to the Banded Algorithm for pairwise alignment focuses on finding patterns in some diagonal bands. Those patterns are formed by *seeds* which are small strings (usually up to 25 nucleotides) that appear in both sequences. FASTA and BLAST are well-known examples of such methods. BLAST is currently the most commonly used sequence alignment tool.

c)

- O(mnk)
  - m characters, n leaves, k possible values for a character

Input includes a $k * k$ scoring matrix describing the cost of transformation of each of k states into another one

- Calculate and keep track of a score for every possible label at each vertex
  - $st(v)$ = minimum parsimony score of the **subtree** rooted at vertex v if v has character t
- The score at each vertex is based on scores of its children:
  - $st(\textbf{\textit{parent}}) = \min i \{si(\textbf{\textit{ left child }}) + \delta_{i,t}\} + \min j \{sj(\textbf{\textit{ right child }}) + \delta_{j,t}\}$
- The scores at the root vertex have been computed by going up the tree
- After the scores at root vertex are computed the Sankoff algorithm moves down the tree and assign each vertex with optimal character.

Question 2

(a) Key steps: extract RNA from samples 1 and 2 reverse transcription -> cDNA, called "target" label cDNA with green (sample 1) or red (sample 2). Mix cDNA from both samples and apply to array allow for hybridisation of target with probes on array. Wash off excess target cDNA from array scan array with laser twice, once at "green" wavelength, once at red. [35%]

(b) Proceed on gene-by-gene basis; several (N) microarray images collected, that compare expression levels in the two types of sample. Replicates are needed to help average out noise in expression levels.

For each gene we then have N values for expression level in sample 1, and N values for sample 2. We then apply stat. test (e.g. modified t-test, e.g. SAM technique) to see if mean of N values from sample 1 is statistically different from mean of sample 2 values.

Problem of multiple-comparisons appears when we have many (» 1000) genes, since null-hypothesis is rejected 1/20 times at 5by chance. e.g. with 1000 genes we could get 50 false positives. Handle by correction procedures, such as FDR.

Non-parametric methods (e.g. rank products) also useful. [35%]

(c) Can apply clustering in two different ways to gene expression data, depending on what you call the input vectors:

(1) clustering genes – each row of the gene expression matrix represents the expression of one gene across different arrays. Used to see which genes have similar gene expression profiles (e.g. genes that have similar expression during cell cycle)

(2) clustering samples – each column of the gene expression matrix represents the expression profile of one microarray sample. Clustering the samples allows for quality control, e.g. similar samples should cluster together. Could then be used to predict category of an unknown sample (e.g. ALL vs AML cancer example). [30%]

Question 3

(a)

(i) $\frac{d}{dt}P(k,t) = \lambda P(k-r,t) + \beta(k+1)P(k+1,t) - (\lambda+\beta k)P(k,t)$

(ii) $\frac{d}{dt}\langle x\rangle = \sum_k k\frac{d}{dt}P(k,t)$

$= \sum\left[k\lambda P(k-r,t) + \beta k(k+1)P(k+1,t) - k(\lambda+\beta k)P(k,t)\right]$

$= \sum (k+r)\lambda P(k,t) + \sum \beta(k-1)k\,P(k,t) - \sum k(\lambda+\beta k)P(k,t)$

$= \sum r\lambda P(k,t) - \sum \beta k P(k,t)$

$= r\lambda - \beta\langle x\rangle$ //

(iii) $\sigma_x^2 = \langle x^2\rangle - \langle x\rangle^2$

$\frac{d\sigma_x^2}{dt} = \frac{d}{dt}\langle x^2\rangle - \left(\frac{d}{dt}\langle x\rangle\right)2\langle x\rangle$

$= \lambda r^2 - 2\beta\langle x^2\rangle + (2\lambda r+\beta)\langle x\rangle - 2(r\lambda - \beta\langle x\rangle)\langle x\rangle$

$= \lambda r^2 + \beta\langle x\rangle - 2\beta(\langle x^2\rangle - \langle x\rangle^2)$

$= -2\beta\sigma_x^2 + \lambda r^2 + \beta\langle x\rangle$

equilibrium: $\langle x\rangle = \frac{r\lambda}{\beta}$

$\sigma_x^2 = \frac{\lambda r^2 + \beta\langle x\rangle}{2\beta} = \frac{\lambda(r^2+r)}{2\beta}$

$\therefore \frac{\sigma_x^2}{\langle x\rangle} = \frac{r+1}{2}$ //

(b) (i) $\frac{d}{dt} P(ON) = k_1 P(OFF) - k_2 y P(ON)$

$= k_1 [1 - P(ON)] - k_2 y P(ON)$

at equilibrium

$P(ON) = \frac{k_1}{k_1 + k_2 y} = \frac{k_1/k_2}{k_1/k_2 + y}$ so $K = \frac{k_1}{k_2}$

(ii) $\frac{d}{dt} \langle y \rangle = \left\langle \frac{K}{K+y} \right\rangle - B \langle y \rangle$,

but $\left\langle \frac{K}{K+y} \right\rangle \neq \frac{K}{K + \langle y \rangle}$