4M8 Dr K Johnstone (Plant Sciences)
keith.johnstone@plantsci.cam.ac.uk

Engineering Tripos -

Bioinformatics 4M8

1)

(*a*) What are the differences between PatternHunter, BLAST, Smith-Waterman and Needleman-Wunsch algorithms? [4 marks]

The Needleman-Wunsch algorithm finds the best global alignment between two sequences, whereas Smith-Waterman, Blast and PatternHunter find the best local alignments. Smith-Waterman assigns a negative score/weight to mismatches (e.g. it requires a gap penalty); this has the effect of maximizing locally optimal alignments at the expense of the global alignment. Needleman-Wunsch usually loses subsequence matches. Both Smith-Waterman and Needleman-Wunsch compare all bases in one sequence against all bases in another sequence, a time consuming method which doesn't lend itself well to large DNA sequences. BLAST and PatternHunter index one sequence and use this to quickly find short exact "seed" matches (hits), which are then extended into longer alignments. PatternHunter is many times faster than Blast and is more sensitive. PatternHunter uses a patented spaced seed technology and algorithm for handling hit generation, hit extension and gap extension. PatternHunter has been proven to be more sensitive and is two orders of magnitude faster than BLAST when processing large sequences and requires only a fraction of the memory. With the optimal multiple spaced seed technology, PatternHunter achieves Smith-Waterman sensitivity at a speed 3000 times faster.

(*b*) Discuss the use of affine the gap penalty with respect to the constant gap penalty. [4 marks]

The affine gap cost model penalizes insertions and deletions using a linear function in which one term is length independent, and the other is length dependent. Gap = Gapopen + Len * Gapextend. A constant gap extension method would assign a fixed cost per gap. An affine gap penalty encourages the extension of gaps rather than the introduction of new gaps.

(*c*) Describe the UPGMA algorithm [4 marks]

UPGMA (Unweighted Pair Group Method with Arithmatic Mean) is an ultrametric tree building algorithm. UPGMA proceeds by inferring one ancestral sequence per step. In the first round UPGMA selects the least distant pair of sequences (or one of them), summarizes their distance as the first branches of a new tree, and recalculates the entire matrix with the pair as one entity (taking the mean of distances). After N - 1 steps (where N is the number of sequences) the matrix is reduced to just one element. The last inferred ancestor is taken as the root of the tree.

(*d*) What does the ultrametric property of a tree tell us about the evolutionary

2)

a. Differentially-expressed gene: gene which has different expression level in two conditions.

For each gene, we compute a stat (e.g. t-statistic, SAM d-statistic, fold-change (Rank Products)) and order them.

Problems: (1) noise – need to average over many chips, as one chip may be unreliable; but cannot average over too many chips as chips are expensive. (2) p value calculations: traditional techniques are too conservative, so need more modern correction methods. (3) DE can create a list of DE genes, but how far down the list do we go? Too long a list =¿ false positives; too short a list =¿ false negatives.

b. Bootstrapping: use existing data to generate distributions typically by resampling existing data (either with or without replacement). Good when we can't use statistical theory to assume a particular distribution of responses, or indeed can now be non-parametric.

Example applications: (1) bootstrap to compute Null distribution of Rank products (or sam). (2) Generate confidence intervals on clustering results (either dendrograms or partitions). (3) Classifier confidence intervals.

c. LDA: Compute p(k—x) for an input vector to be classfied into class k; pick k s.t. p(k—x) maximised. Uses Bayes rule to compute posterior, typically assuming e.g. Normal distribution for each p(k—x), params estimated by data. K-nn very simple in contrast: label to majority of K nearest neighbours. LDA has sound theory behind it, but requires estimation of many parameters; Knn gives little explanation. How do we choose K? When classifying microarray data, both can be used and compared. Empirical results comparing several datasets suggests that choice of classifier might not be so important for microarray data.

3 a)

~~$\dot{p}_k = \lambda p_{k+1} + \beta p_{k-2}$~~

$$\dot{p}_k = \lambda p_{k-1} + \beta(k+2)^2 p_{k+2} - \lambda p_k - \beta k^2 p_k \quad ; k \geq 2$$

$p_0 = 0$

$$\dot{p}_1 = \beta (3)^2 p_3 - \lambda p_1$$

2 marks

b) Sample next event from exponential distribution with parameter $\lambda + \beta n^2$

this is a bird with prob $\frac{\lambda}{\lambda + \beta n^2}$ etc 2 marks

c) $$\frac{d\langle n \rangle}{dt} = \lambda \langle n \rangle - \beta \langle n^2 \rangle$$

$\langle n^2 \rangle - \langle n \rangle^2 = \sigma_n^2$ assumed small

$$\Rightarrow \frac{d\langle n \rangle}{dt} \approx \lambda \langle n \rangle - \beta \langle n \rangle^2$$

2 marks

d) Linearising about $\langle n \rangle = N \qquad \lambda = \beta N$

$$\dot{x} = \lambda x - 2\beta N x + \frac{1}{2}\lambda \eta$$

$\Rightarrow$ (after a lot of working)

$$\eta = \frac{\sigma^2}{\langle n \rangle} = 0.5$$ whereas $\frac{\sigma^2}{\langle n \rangle} = 1$ for linear case

need to check this