

I519 Lab 2 Fall 2008
School of Informatics
Indiana University

Prof. M.M. Dalkilic
Associate Instructor Chao Ji

September 25, 2008

Introduction

- Be thorough in your answers.
- You are free to discuss abstractly the problems—no other kind of assistance is acceptable

R

1. Create and read the file below into R.

X	Z	Y	W
1	0	3	2
-1	2	3	4
4	3	2	1
4	5	6	4
2	3	2	4

Are the variances of Y and W significantly different?

2. The geometric mean $\hat{y} = \sqrt[n]{\prod y}$ was given as `exp(mean(log(.)))` where `.` is the formal parameter. Show mathematically the R code calculates the geometric mean.
3. Write a function in R that calculates the entropy of a discrete probability distribution—the distribution will be in the form of a list *e.g.* `c(.2 .3 .1 .4)`. The function should return an error message if the actual parameter is not a probability distribution. To remind you, for $P = (p_1, p_2, \dots, p_n)$, then the entropy is $\mathcal{H}(P) = \sum_{i=1}^n p_i \log \frac{1}{p_i}$
4. Define “degrees of freedom”. Why is this quantity important?
5. Confidence intervals (CI) are necessary, since we cannot guarantee the value of an outcome. Explain first what CI are. What is the relationship between CI and interval size?

General Concepts

1. A fair coin is flipped until the first tail appears. Let X be a random variable whose values are the number of flips required. Find $\mathcal{H}(X)$. (*hint*: the handout I gave earlier in the week might help)
2. Prove or disprove $n \log n \in O(n^2)$

3. Prove or disprove $\log_{10} n \notin O(\log_4 n)$
4. Using R plot the pairs of functions in the two questions above from 1 to 100 in intervals of 10. Would this empirical procedure be proof of correctness?

Programming

1. Here is a recurrence: $\binom{N}{k} = \binom{N-1}{k} + \binom{N-1}{k-1}$, where $\binom{N}{N} = \binom{N}{0} = 1$. In the language of your choice, write (1) a recursive solution (2) a Top-down DP and (3) a Bottom-up DP.
2. Find $\binom{2000}{1000}$ (*hint*: that's enough of a hint)
3. Program Smith-Waterman algorithm for proteins. The program should be called SW and take four arguments: -m scoring_matrix, -f fasta_file, -g opening_gap_penalty, -e extension_gap_penalty. The FASTA file contains two proteins in FASTA format. We will learn about scoring matrices next week. But for now you should know they are simple grids in which a cell contains a match score indexed by row letter and column letter. The output should displayed as an HTML page and give one of the highest scoring alignments as its value. A sample output would look like this:

Score = 21

```

P X W Z - A
| | * | *
P X Y Z N N

```

Observe if two letters are identical, then a bar is placed between them; if there is a non-identical match then an asterisk; finally, for gaps nothing. For **Big Bonus** extra credit add an extra argument to the function that indicates the number of local alignments you'd like to see. For instance, if you have 3 as an actual parameter, the program will produce the top three local alignments.