Nate Evans

Christina Zheng & Shannon McWeeney

BMI665

11/5/18

Midterm, Fall 2018

Reminder: This is a take-home exam. Students are expected to develop, write up, and hand in their own individual solutions and, in doing so, develop a sufficient understanding of the problem and solution so as to be able to explain it adequately to the instructor. Under no circumstances should a student discuss the exam with others while taking it, copy or consult the solution of another student, or copy a solution from any other source, including the Internet. You must show all work.

1. Provide the pseudo-code for an algorithm that will determine the **length** of the shortest substring that occurs exactly once in a genomic sequence of length N. You must provide all assumptions made for your algorithm.

Assumptions

* Case sensitive
* Spaces (gaps) are treated as a distinct character
* Input string needs no parsing, what is provided is used to find shortest substring
* If there is no shortest substrings found, the length of the full string will be returned
* The given string is taken as is, it does not represent DNA/RNA/AAs and therefore the only substring match is the substring. e.g. ‘ATG’ != ‘TAC’

**FOR CODE, SEE “evans\_midterm.py”**

Given a string S, to find the shortest substring that occurs exactly once, the following algorithm can be employed.

1. Generate a BWT from S
2. Using the F,L backtracking method, build out each substring (row)
   1. At each additional character, compare each substring (row) with all other substrings (rows).
      1. If there are any unique substrings (a substring that has no other matches do not include substrings with terminal character),
         1. Return the number of characters in substring (iteration +1)
3. If the entire BW matrix is reconstructed, return the length of S (OR iteration +1)

Example: S = “ACTGTC”

Algorithm search visualized (red=unknown, black=known)

<iteration = 0>

[['$' 'A' 'C' 'A' 'C' 'T' 'G' 'G' 'T' 'C']

['A' 'C' 'A' 'C' 'T' 'G' 'G' 'T' 'C' '$']

['A' 'C' 'T' 'G' 'G' 'T' 'C' '$' 'A' 'C']

['C' '$' 'A' 'C' 'A' 'C' 'T' 'G' 'G' 'T']

['C' 'A' 'C' 'T' 'G' 'G' 'T' 'C' '$' 'A']

['C' 'T' 'G' 'G' 'T' 'C' '$' 'A' 'C' 'A']

['G' 'G' 'T' 'C' '$' 'A' 'C' 'A' 'C' 'T']

['G' 'T' 'C' '$' 'A' 'C' 'A' 'C' 'T' 'G']

['T' 'C' '$' 'A' 'C' 'A' 'C' 'T' 'G' 'G']

['T' 'G' 'G' 'T' 'C' '$' 'A' 'C' 'A' 'C']]

<iteration = 1>

[['$' 'A' 'C' 'A' 'C' 'T' 'G' 'G' 'T' 'C']

['A' 'C' 'A' 'C' 'T' 'G' 'G' 'T' 'C' '$']<non-unique

['A' 'C' 'T' 'G' 'G' 'T' 'C' '$' 'A' 'C']<non-unique

['C' '$' 'A' 'C' 'A' 'C' 'T' 'G' 'G' 'T']<terminal

['C' 'A' 'C' 'T' 'G' 'G' 'T' 'C' '$' 'A']<unique! <<

['C' 'T' 'G' 'G' 'T' 'C' '$' 'A' 'C' 'A']<NA

['G' 'G' 'T' 'C' '$' 'A' 'C' 'A' 'C' 'T']<NA

['G' 'T' 'C' '$' 'A' 'C' 'A' 'C' 'T' 'G']<NA

['T' 'C' '$' 'A' 'C' 'A' 'C' 'T' 'G' 'G']<NA

['T' 'G' 'G' 'T' 'C' '$' 'A' 'C' 'A' 'C']]<NA

Compare each substring (first column) with all other substrings (other first column, rows)

<<< return substring length (iteration + 1)

2. As reported in the HG38 version of the human genome, please answer the following questions about the gene, CD8B.

a. What chromosome is this gene on?

Chromosome 2

b. What is the start and end coordinate of this gene?

[86815557, 86861915]

But since it’s on the minus strand, the gene 5’ end is at 86861915 and the 3’ end is at 86815557.

c. What strand of the DNA contains this gene?

Minus (-) strand

d. How many transcripts are in this gene as reported by the RefSeq Consortium?

5

e. How many distinct exons are reported in the RefSeq transcripts?

9, Total Exon Count: 6 - reported in Description and page index

f. How many distinct introns are reported in the RefSeq transcripts?

10

g. List the genomic coordinates of 3 introns (start-end) from above which contains a canonical splice sites.

???

h. List the genomic coordinates of each intron (start-end) from above which contains a noncanonical splice site.

???

i. What is the official gene symbol of the nearest gene to CD8B? What strand of the DNA contains this nearest neighboring gene? How would the location of these genes be described in relation to one another?

ANAPC1P1

This is on the positive (+) strand. This gene is on the complementary DNA strand to CD8B, immediately upstream. Eg. it starts at 86861915 on chr2 but on the other strand of DNA and going the other direction.

3. Describe the minimum information needed to unambiguously define the location of a gene? Provide an example using a gene of your choice.

Assuming that this must apply to genes whose symbols are not in the UCSC browser, and thus cannot be accessed via gene symbol. e.g. how to define the location of a new gene?

1. A Reference Genome
2. Chromosome ID/number
3. Start and end coordinates
4. DNA strand (+/-)

To unambiguously define the location of CD8B as if it was an unknown gene, we must define which reference genome we will use. For this example, we will choose the human genome **HG38 version.** Next, we must define which chromosome it’s on, define it as **chromosome 2.** The next step is to define where in the chromosome it’s located, we will define the start end coordinates as **[86815557, 86861915]** meaning that the gene starts at the 86815557th base pair from the 5’ end of the positive strand. Lastly, we have to define which strand of DNA the gene is located on, positive (5’ end is at 0th base) or negative (3’ end is at the 0th base). Negative and positive are complementary strands and the displayed bases are representative of the positive strand, so if the gene is on the negative strand, the complement of the gene shown should be considered. In this example, we will define the gene as being on the **negative strand.**

4. Align the protein sequences S1 and S2 with a **gap opening penalty of 2** and a **gap extension penalty of 1** and the **“H2O” substitution cost defined as**:

For amino acids x and y,

H(x,y) =

+5, if x=y

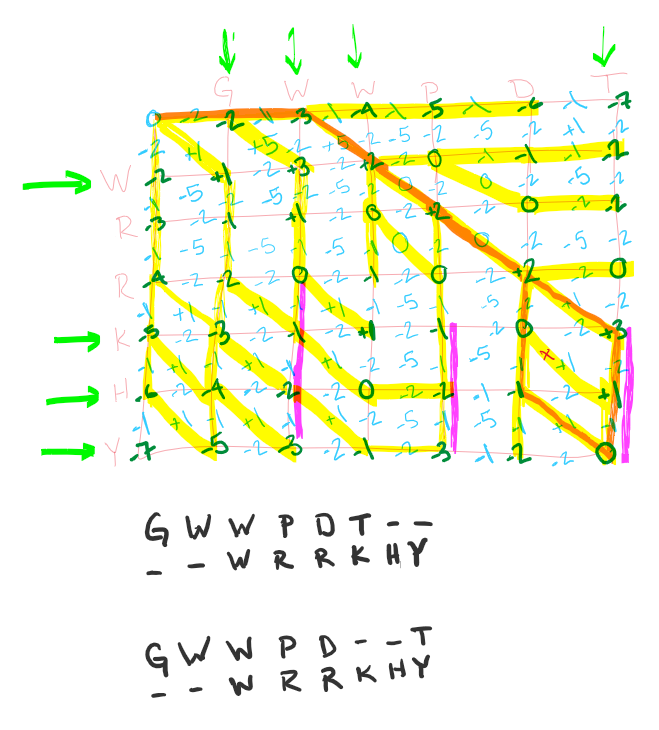
+1, if x≠y, but x,y are both hydrophobic

‐5, if x≠y, and only one is hydrophobic

0, if x≠y, neither are hydrophobic

S1 = GWWPDT

S2 = WRRKHY



In the table above:

Blue : score for specific transition

Dark Green : score at that point

Yellow: Transition resulting in node score

Purple: Transition dependant on gap extension scoring, think of it as an overpass; If traveling along purple route, you can only get off at the end.

Orange: Highest scoring alignment route

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| G | W | W | P | D | T | - | - |
| G | W | W | P | D | - | - | T |
| - | - | W | R | R | K | H | Y |

5. Describe a scenario where a researcher would be interested in investigating both the local and global alignment of two sequences.

When comparing the genomes of two distinct organisms, local alignment of a gene may show highly conserved regions, which may pertain to functional regions of RNA or Protein. However, global alignment would be useful in measuring the genetic time scale that separates the two organisms; The global alignment can be used to measure how long ago the two species diverged.

6. Bacterial genomes are often circular. To transform to a linear form, some genome assembly programs will pick a random location in the genome to break the circle. Thus, it is possible that running the same program multiple times we would get different answers, corresponding to different circular rotations of the same string. Provide the psuedo-code that will determine if two DNA strings are circular rotations of each other. For example TTGATC is a circular rotation of ATCTTG. You must state all assumptions.

Given string S1, S2

Asserting that S1 and S2 are the same length

Asserting that it only contains characters {ATCG}

Assumes that the genome assembly program always outputs the same strand and directionality, eg. S1 and S2 will never be complements or reverse complements of each other

Assumes S1 and S2 represent only DNA, no mRNA or Residue sequences

**Please see evans\_midterm.py for code**

7. We can define a set of distinct substrings of a string S that includes all substrings. However, each repeat is only represented once. For example, for the string S = AATATT, this set is:

{A, T, AA, AT, TA, TT, AAT, ATA, TAT, ATT, AATA, ATAT, TATT, AATAT, ATATT, AATATT}

You are given a suffix tree of S. Provide the pseudo-code for an algorithm that counts the number of distinct substrings of S. For full credit, this should run in O(n) time.

This is a recursive algorithm to count the number of tips of the tree.

-------------------------------------------------------------------------------

Define a function such that, given a specific node and access to the entire tree, it will:

1. Define a count variable, set to zero

IF there are no child nodes

2. Return 1

ELSE

1. for each child node of the given node
   1. Call this function again, passing it the child node, and add the returned value to our count variable.
2. Return the count variable

Provide the function above the root node of your tree.

-------------------------------------------------------------------------------

More codified- Pseudocode may actually make it more clear.

-------------------------------------------------------------------------------

Def rec\_search(node, tree):

count = 0

If (node.has\_no\_children()):

return 1

For child in node.get\_children\_nodes():

count += rec\_search (child, tree)

return count

-------------------------------------------------------------------------------

num\_distinct\_substrings = rec\_search(root, tree)

-------------------------------------------------------------------------------