Introduction to Bioinformatics — Hints for solving problems

Chapter 1.

Problem 1.1 Keep as much of the core of the code as possible. Apply it to the original sequence as in the original program, and to the original sequence with 1 character removed from the beginning, and with 2 characters removed from the beginning. Create the reverse complement by processing the original string character by character and repeat the steps in the previous sentence.

Problem 1.2 You will need to write a short program to generate the fragments. Then try them using the original program! Obviously (c) is the hardest. The main difficulty is to understand why the given program does not work in cases where it fails. Do a flowchart of the program. The problem with example (c) is that the program is designed under the assumption that there is only one way to assemble partial overlaps at any stage.

Problem 1.3 You will need to write a short program to generate the fragments; this will be similar but not identical to the one needed for Problem 1.2.

Problem 1.4 Use the same logic and flowchart, but beware of trying to translate PERL directly into PYTHON.

Problem 1.5 Use the same logic and flowchart, but beware of trying to translate PERL directly into PYTHON.

Problem 1.6 (a) test each subject in each cohort for correlation between phenotype and presence/absence of each SNP. (b) Calculate how many tests you would then need, and assume that the time required simply scales with the number of tests. Can you think of any clever ways to speed up the calculation?

Problem 1.7 For example, the first two executable statements in the concise version read the data and create the array of fragments. They correspond to lines 4-9 of the long version.

Reasonable comments might include:

```
# input of data.  create array each element of which contains one line of data.
     $/ = {"}";  # signal to read in full paragraph at a time
     @fragments =
        split("\n",<DATA>);  # split lines in paragraph into successive
                             # elements of an array
```

Problem 1.8 Ask: what skills are required, efficiently to create correct and syntactically acceptable annotations? The answer should include components from biology and database organization.

Chapter 2.

Problem 2.1 This is an application of the Hardy-Weinberg principle. Calculate the overall allelic composition of the equilibrium population and distribute the alleles randomly among individuals.

Problem 2.2 (a) 1 cM = 1% recombination frequency per generation. (b) probability of recombination + probability of non-recombination = 1. (c) Probability of no recombination in 2 generations = (probability of no recombination in one generation)^2. Generalize this, and evaluate for part (d).

Problem 2.3 (a) Substitute appropriate numbers into formulas. For eight-fold coverage, c = 8 (b) G = 2×10^6 (c) L = 500 (d) Solve for $Ne^c = 4$.

Problem 2.4 The sequence begins: `CCTTATC`...

Problem 2.5 Use Figure 2.19 to create the generalized suffix tree for

```
gacata#atagac$
```

Problem 2.6 Form the generalized suffix tree from the sequence plus its reverse complement and look for matches, or run the sequence against its reverse complement through the dotplot program (see Chapter 4.)

Problem 2.7 Draw a figure for the new sequence analogous to the procedure in Box 2.11.

Problem 2.8 How many errors would be tolerable? Convert to Phred score.

Problem 2.9 Leaving aside the possibility that the reader is considering a career as a violent criminal, relevant considerations involve both personal aspects and aspects of public policy. One condition to be considered would be to retain anonymity in applications of the sequence data to research projects.

Problem 2.10 There are 365 possible birthdays. The probability that anyone will have a particular birthday is 1/365. There are 4^N possible DNA sequences of length N. The probability that any N-mer in a genome of length 3.2×10^9 will appear at some position is $1/3.2\times10^9$; ignore the correction to $1/(3.2\times10^9 - N)$.

Chapter 3.

Problem 3.1 (a) For instance, for *M. genitalium*, the gene density is 468 genes/580.07 kb = 0.81 genes/kb.

Problem 3.2. Estimate the total number of bases required and compare with the size of the human genome.

Chapter 4.

Problem 4.1 (a) Given a probe string `$motif` and a sequence to be searched `$sequence,` finding exact matches is easy in PERL: `$sequence` =∼/`$motif/e` where the `e` suffix means that the pattern to be matched is an expression. (b) To find matches with one allowed error, replace each position in the `$motif` with a 'wild card' ? character. (? matches any single character.) That is, if you want to find the word match with one error, search for /?atch/, /m?tch/, /ma?ch/, /mat?h/, and /matc?/.

Problem 4.2 Use dotplot program from text.

Problem 4.3 (a) The difference is in the initialization. (b,c) The solution with no gaps internal to the motif `atg` would score the highest.

Problem 4.4 First identify the positions of the turns by inspection of Figure 4.7, or by checking the secondary-structure assignments in the wwPDB file 2TRX, or by using other software; for instance, SST: `http://lcb.infotech.monash.edu.au/sstweb2/Submission_page.htm` An example of a turn not corresponding to a region containing insertions or deletions appears near residue 30.

Problem 4.5 Hints are contained in the statement of the problem: Two possibilities are (a) determine the sequence similarities of all pairs of sequences, and retain only one example of any set of sequences for which all pairs have high mutual similarity. (b) determine the sequence similarities of all pairs of sequences, and weight each sequence by 1 divided by the number of sequences that have high mutual similarity.

Introduction to Bioinformatics — Hints for solving problems

Problem 4.6 Basic steps: (1) Read alignment table. (2) Process alignment table column by column, making inventory. (3) To score a distribution of amino acids in the inventory against a single amino acid in a query sequence, either take the minimum score of the query amino acid with any amino acid that appears in that column, or take a weighted average of the scores.

Problem 4.7 (a) One way is to take a running 5-character window from the first sequence and make an associative array with the five-character sequence as argument. Then take a running 5-character window from the second sequence and check whether each 5-character sequence appears as an argument in the associative array. (b) For this you will have to record where in the first sequence each 5-character substring appeared.

Problem 4.8 You are being asked to produce something like Figure 4.3, with specific character strings.

Problem 4.9 Produce individual frames, convert to gif format, and collect into a movie using software such as gifsicle. `https://www.lcdf.org/gifsicle/`

Problem 4.10 Draw a structure analogous to a dotplot, except that the original sequence should appear along the rows, and the reverse complement should appear down the columns. A diagonal series of matches in this plot corresponds to a pair of complementary regions. (Compare Problem 2.6.)

Problem 4.11 One approach is to run simulations. For each position of each die, there are only four possible successor positions. Use the perl *rand* function to choose the successor, and keep track of the 'trajectory' of the system until the statistics converge.

Problem 4.12 Basic idea: if there are two distinct ways to get from node A to node B, then go from A to B along one path and back from B to A along the reverse of the other.

Problem 4.13 (a) Just draw them out. Not as bad as it seems. Remember that you can't retrace any step. (b) For instance, there are 3 paths from Start to B (Northeast-Northeast-Southeast, Northeast-Southeast-Northeast, and Southeast-Northeast-Northeast). (d) There are $6\times5\times4$ ways to make successive choices that assign 3 left turns to six steps, For each assignment, it could have been arrived at by choosing those three steps in six different orders.

Problem 4.14 (a) The basic idea is that two or more nodes with the same parent are enclosed in parentheses, and then are replaced by the parent node. The innermost pair in this example is (BC). Let X be the parent node of BC. This transforms the expression to ((AX)D). Define a parent node of AX as Y and continue. (b) Reverse the process. The left tree in Exercise 4.24 would be written: (A(((EC)D)(BF)))

Problem 4.15 Give both an introductory statement of what is going on and what the method is. Also end each line with a comment, beginning with #, stating what that line accomplishes.

Problem 4.17 For the data in Example 4.7, the UPGMA tree gives the distance from A→B as the sum of the labels on the segments of the path from A→B: 2.66 + 2.66 + 1.25 + 0.5 = 7.07. The split decomposition diagram gives 4.

Problem 4.18 Most obvious is that you lose the observation (visible in the split decomposition) that A is not too much farther from G than it is from B.

Problem 4.19 Read the sequences and append successive portions of the sequences that appear in different blocks to give one variable per sequence, containing the entire sequence, including the gap characters. Then for each pair of sequences compare character-by-character to count mismatches. An attempt to be tricky, using PERL, would be to count mismatches between two strings `$a` and `$b` by taking the exclusive or of the strings (PERL operator ∧) and counting non-null characters: `$count = ($a ∧ $b ) =∼ tr/\0//;` But what would happen if some position in each of the two

sequences contained a gap character? This could not happen for only two strings but might happen for more than two strings.

Problem 4.20 Use the same logic and flowchart, but beware of trying to translate PERL directly into PYTHON.

Problem 4.21 Use the same logic and flowchart, but beware of trying to translate PERL directly into PYTHON. The generalization required is to extract the branch lengths from the input, and to draw the tree to reflect the specified branch lengths.

Chapter 5.

Problem 5.1 Consider (a) allowing mismatches from the PPHHPPHHPP pattern, (b) looking for periodicities of 4 in the hydrophobicity.

Problem 5.2 To change colours in postscript insert the statement `r g b setrgbcolor` where `r`, `g` and `b` are the relative values of red, green and blue. Thus `1 0 0` = red, `0 1 0` = green, `0 0 1` = blue, `0 1 1` = cyan, `1 0 1` = magenta. (There must be a space or spaces between the r, g, b values and before the word setrgbcolor.) Each time the protein prints an amino acid one-letter abbreviation, look up the appropriate colour and print the appropriate setrgbcolor statement.

Problem 5.3 That is, associate x and y coordinates with each residue, by converting cylindrical polar coordinates to x and y. In fact, associate two sets of x and y coordinates with each residue corresponding to the two copies of the helix surface network. Connect neighbouring hydrophobic residues. Find large connected subsets. One way to do this is to assign a different 'colour' to each residue, and whenever two residues are connected, change the colour of one so that the entire connected set has one colour. Then for each colour count the number of residues with that colour.

Problem 5.4 For example, residues 27 and 28 are predicted correctly because they are within the experimental helix 27-29. Residue 29 is not predicted correctly because it is in fact helical.

Problem 5.5 Bonneau, Tsai, Ruczinski and Baker also predicted a helix starting at residue 22. They got residue 29 right.

Problem 5.6 All points with coordinates $0 \leq x \leq 1, 0 \leq y \leq 1$ lie in the unit square. Points for which $x^2 + y^2 < 1$ lie in the circle. Generate points and count the fraction that lie within the circle.

Problem 5.7 (a) Note that the sixth column has all large non-polar residues (F, with one Y). Consider the distribution of positively-charged residues, such as R and K. Many columns have a few R and K residues but some, notably near the C-terminus, are rich in R and K. (b) For example, L is absolutely conserved in the third column. Column 7 also has conserved L but with one exception. (c, d) Look for periodicities of approximately 4, suggesting an $\alpha-$helix, or 2, suggesting a strand of $\beta-$sheet. (e) A cluster of positively-charged residues should bind a negatively-charged ligand. A nucleic acid would be a reasonable suggestion.

Problem 5.8 Right approach: study the pictures of the structures. Wrong approach: look them up in SCOP or CATH.

Problem 5.9 A challenge in this problem is to relate the mutated sidechains in Figure 5.30 with positions in the sequence.

Problem 5.10 (a) $3\times4 - 6\times1$. (b) Solve $3x - 6 = 0$.

Introduction to Bioinformatics — Hints for solving problems

Chapter 6.

Problem 6.1 Profit = 5800×price - (sum of costs) = 1.05×(sum of costs); add up costs and solve for price.

Problem 6.2 Identify all amino acids that have volumes over the stated threshold. Identify all amino acids that have distal carboxy or amide groups on their sidechains. Identify which amino acids are common to both lists, and draw as Venn diagram.

Problem 6.3 (a, b) In the correct parsing, British is an adjective modifying left, which is a noun referring to a group of persons with a liberal political outlook; and Waffles is a verb. (c) The problem is that there are two potential verbs in the sentence: left, and waffles. Design a sentence without a second possible verb. For instance:(Name of any politician [perhaps excluding Margaret Thatcher]) waffles on (any topic).

Problem 6.4 The problem is 'that croaks the fatal entrance of Duncan' is a unit, introduced by the word that. It is as if the sentence had been punctuated (misleadingly) as follows: The raven himself is hoarse, that croaks the fatal entrance of Duncan, under my battlements.

Problem 6.5 A 'conveniently-readable format' would have the characteristic of being not cluttered up with markup material.

Chapter 7.

Problem 7.1 $\Phi(x)\cdot\Phi(y)$ is the dot product of $(1, \sqrt{2}\times x, x^2)$ and $(1, \sqrt{2}\times y, y^2)$.

Problem 7.2 The line $-bx + ay = 0$, equivalent to: $y = (b/a)x$ has slope (b/a). The line $ax + by = c$, equivalent to: $y = -(a/b)x + c$ has slope (a/b). If the product of the slopes of two lines is -1, the lines are perpendicular.

Problem 7.3 Hint contained in problem.

Problem 7.4 You could do this with a network with 8 nodes in the input layer, each feeding into a node in a hidden layer that emits 1 if the input is H and 0 if the input is P (for nodes 1, 2, 5 and 6) or which emits 1 if the input is P and 0 if the input is H (for nodes 3, 4, 7 and 8). All eight nodes feed into an output node with a very simple output criterion.

Problem 7.5 The design of a neural network that could achieve an exclusive or of two inputs was a classic problem in the field. In a seminal book from 1969, Minsky and Papert showed that a neural network without hidden layers could not compute an exclusive or.

Problem 7.6 Assign a variable to each node, and give a formula converting the inputs to each node to the outputs.

Problem 7.7 Modify the program from Problem 7.6 to use the smoothed function. Then compare the outputs of the two programs with a set of carefully chosen inputs.

Problem 7.8 The problem with trying to do this computationally is that you don't have the analytic representations of the individual curves. It could be done by image processing but that would be a much more difficult approach.

Problem 7.9 Look for STOP codons in all reading frames, including the reverse complement sequence.

Problem 7.10 Use formula following Figure 7.21.

Problem 7.11 Evaluate formula in caption to Figure 7.17.

Problem 7.12 This will require some access to matrix diagonalization software. I recommend the

language R.

Chapter 8.

Problem 8.1 For instance, the number of neighbours of Oxford Circus is 6. To create a graph of the London Underground (see Weblem 8.3) see:
`https://commons.wikimedia.org/wiki/London_Underground_geographic_maps/CSV` or
`https://lasttrain.co.uk/tube-train-lines/london-underground-tube-line-names/`

Problem 8.2 Either just count them, or derive a graph from one of the web sites listed in the previous problem and analyse the result computationally.

Problem 8.3 With 5 yes-or-no questions one could distinguish 32 different items. So 5 is enough to distinguish 26 letters, but 4 yes-or-no questions would not be sufficient.

Problem 8.4 For each text in a known language, create a file containing the known language example followed by the text in the unknown language. Compress the result, using some standard algorithm such as gzip, and compare the sizes of the compressed files.

Problem 8.5 There exist specialized algorithms for determining shortest paths in graphs, but a rough-and-ready approach would be to work outwards from each station, creating a list of neighbours at different distances. Cyclic paths create a difficulty.

Problem 8.6 The Transport for London route planner is at `https://tfl.gov.uk/plan-a-journey/`

Problem 8.7 (b)

```
M1y1c1a1r1e1i1s1l1o1s2o1f1c1a1r1e1b1y1o1l1d1c1a1r1e1d1o1n1e1
Y1o1u1r1c1a1r1e1i1s1g1a1i1n1o1f1c1a1r1e1b1y1n1e1w1c1a1r1e1w1o1n1
```

Original string: 63 characters. RLE string 124 characters. In this case, no compression.

(d) Expand the RLE version to the entire string and then invert the Burrows-Wheeler transform.

Problem 8.8 Given the complete set of suffixes of a string, sort them. In order of the suffixes, form a string BWT by appending the character that precedes the suffix to BWT. (If the suffix is equal to the entire string, append the terminator character $ to BWT.) Then BWT is the Burrows-Wheeler transform of the original string.

Problem 8.9 This is intended as a pencil-and-paper problem, but it would be possible to write a program to carry out the steps and print perspicuous intermediate results.

Chapter 9.

Problem 9.1 Ask yourself: what will be the phenotype of a heterozygote?

Problem 9.2 For last question: if not, what would be the effect on the reaction rate?

Problem 9.3 Treating the longest linear pathways, all but two of the metabolites align.

Problem 9.4 Metabolites in central vertical column align.

Problem 9.5 For instance, from the entry 7 5 in row 2 (second and third columns) it would be possible to generate 9 2, 5 7, or 4 8. To generate 5 7 would not be in the shortest path because it generates a combination that already occurs.

Introduction to Bioinformatics — Hints for solving problems

Chapter 10.

Problem 10.1 (a) Number of positions = number of rows × number of columns. (b, c) Count number of red and green spots. (d) Count number of yellow spots. (e) Count positions that are not red and not yellow and not green.

Problem 10.2 (a) $k_{\text{off}} = k_{\text{on}} \times K_{\text{D}}$. (b) $t_{\frac{1}{2}} = 0.693/k_{\text{off}}$

Problem 10.3 Strains (a) and (c) do not feel effects of isoniazid. (c) in strain c the drug is not activated.

Problem 10.4 (a) $k_{\text{off}} = k_{\text{on}} \times K_{\text{D}}$. (b) You should draw conclusions from the values calculated in (a). Note however that the k_{on} rates are all equal to within a factor of 10, but the differences in K_{D} are much greater.

Problem 10.5

(a) The sum of C $\exp(\alpha k)$ = C[$\exp(\alpha)$ + $\exp(\alpha)^2$ + $\exp(\alpha)^3$ + ...] = C$\exp(\alpha)$/[1 – $\exp(\alpha)$] = 1. Solve for C.

(b) Knowing C from part (a), compute successive values of C$\exp(\alpha k)$ for $k = 1, 2, \ldots$ and stop when the value becomes < 0.01

(d) Hint given in text.

(e) Substitute $\alpha = 0.8$ into answer to (d).

(f) If M is the median, then the sum of C$\exp(\text{-}\alpha k)$ from 1 to M is ≤ 0.5 but the sum from 1 to M+1 is ≥ 0.5. $\sum_1^M C \exp(-\alpha k) = C \exp(-\alpha)(1 - \exp(-\alpha M))/(1 - \exp(-\alpha))$. Evaluate for $M = 1, \ldots$ until sum exceeds 0.5.

Problem 10.6 Two forks in succession would do it.

Problem 10.7 For instance, CII activates CI. Therefore add a node corresponding to CII and an arrow from it to CI.

Problem 10.8 For instance, if mutant cro failed to bind OR3, the repression indicated by the link between cro and CI would be lost.

Problem 10.9 (a) Determine by inspection. (b) Possible edit operations: add node, delete node, add edge, delete edge. Work out a sequence of edit operations that convert one network to the other.