Submission Phase

1. Do assignment ☑ (/algorithmicthink2-004/human_grading/view/courses/976305/assessments/34/submissions)

due in 5day 5h

Evaluation Phase

2. Evaluate peers ☑ (/algorithmicthink2-004/human_grading/view/courses/976305/assessments/34/peerGradingSets)

Results Phase

3. See results 🔒 (/algorithmicthink2-004/human_grading/view/courses/976305/assessments/34/results/mine)

← Return to list (/algorithmicthink2-004/human_grading/view/courses/976305/assessments/34/peerGradingSets/261)

Save draft Submit evaluation

Submission from: Student 6

Overview

In Project 4, you implemented dynamic programming algorithms for determining both global and local alignments of pairs of sequences. In this Application, we will demonstrate the utility of these algorithms in two domains. In the first part of the Application, we examine an interesting problem from genomics. (This is based on "Introduction to Computational Genomics", by Nello Cristianini and Matthew W. Hahn). We will compare two sequences that have diverged from a common ancestor sequence due to mutation. (Mutation here includes base-pair substitution, which changes the sequence content, and insertion/deletion, which change the sequence lengths.) In the second part of the Application, we consider words that have spelling mistakes.

For the genomics part of the Application, you will load several protein sequences and an appropriate scoring matrix. For the spelling correction part of the Application, you will load a provided word list. To simplify these tasks, you are welcome to use this provided code (http://www.codeskulptor.org/#alg_application4_provided.py).

Important: Please use Coursera's "Attach a file" button to attach your plots/images for this Application as required. For each question you can attach more than one image as well as including text and math (LaTeX) in the same answer box. In particular, please do not post your solution plots/images on 3rd party sites. This practice exposes your peers to extra security risks and has the potential for abuse since the contents of a link to an external site may be modified after the hard deadline. Failure to follow this policy may lead to your plots/images being counted as "not submitted".

Comparing two proteins

In 1994, Walter Gehring and colleagues at the University of Basel carried out an "interesting" experiment: they were able to turn on a gene called eyeless in various places on the body of the fruit fly, *Drosophila melanogaster*. The result was astonishing - fruit flies developed that had whole eyes sprouting up all over their bodies. It turned out that the eyeless is a master regulatory gene - it controls a cascade that contains more than 2000 other genes. Turning it on anywhere in the body activates the cascade and produces a fully formed, but non-functioning, eye. Humans, as well as many other animals, have a slightly different version of the eyeless gene (that is, a similar, yet not identical sequence of the same gene).

This observation suggests that about 600 million years ago (the estimated time of divergence between humans and fruit flies) there was an ancestral organism that itself used some version of eyeless, and that throughout the evolution of humans and fruit flies this gene continued to be maintained, albeit while accumulating mutations that did not greatly affect its function. In particular, a substring of the eyeless protein of about 130 amino acids, known as the PAX domain, whose function is to bind specific sequences of DNA, is virtually identical between the human and fruit fly versions of eyeless.

In following questions, we compute the similarity between the human and fruit fly versions of the eyeless protein and see if we can identify the PAX domain.

Question 1 (2 pts)

First, load the files HumanEyelessProtein (http://storage.googleapis.com/codeskulptor-alg/alg_HumanEyelessProtein.txt) and FruitflyEyelessProtein (http://storage.googleapis.com/codeskulptor-alg/alg_FruitflyEyelessProtein.txt) using the provided code. These files contain the amino acid sequences that form the eyeless proteins in the human and fruit fly genomes, respectively. Then load the scoring matrix PAM50 (http://storage.googleapis.com/codeskulptor-alg/alg_PAM50.txt) for sequences of amino acids. This scoring matrix is defined over the alphabet `{A,R,N,D,C,Q,E,G,H,I,L,K,M,F,P,S,T,W,Y,V,B,Z,X,-}` which represents all possible amino acids and gaps (the "dashes" in the alignment).

Next, compute the local alignments of the sequences of HumanEyelessProtein and FruitflyEyelessProtein using the PAM50 scoring matrix and enter the score and local alignments for these two sequences below. Be sure to clearly distinguish which alignment is which and include any dashes (`'-'`) that might appear in the local alignment. This problem will be assessed according to the following two items:

- Is the score of the local alignment correct? (Hint: The sum of the decimal digits in the score is 20.)
- Are the two sequences in the local alignments (with dashes included if inserted by the algorithm) clearly distinguished and correct?

Score: 875

Human:

HSGVNQLGGVFVNGRPLPDSTRQKIVELAHSGARPCDISRILQVSNGCVSKILGRYYETGSIRPRAIGGSKPRVATPEVVSKIAQYKRECPSIFAWEIRDRLLSEGVC

QQ

Fruit Fly:

HSGVNQLGGVFVGGRPLPDSTRQKIVELAHSGARPCDISRILQVSNGCVSKILGRYYETGSIRPRAIGGSKPRVATAEVVSKISQYKRECPSIFAWEIRDRLLQENVC

flag (https://accounts.coursera.org/i/zendesk/courserahelp?return_to=https://learner.coursera.help/hc/articles/201212339-Peer-Graded-Assignments)

Evaluation/feedback on the

above work

Item a (1 pt) Is the score of the local alignment correct? (Hint: The sum of the decimal digits in the score is 20.)

The score for the local alignment is 875.

Item b (1 pt) Are the two sequences in the local alignments (with dashes included if inserted by the algorithm) clearly distinguished and correct?

The local alignment has the following two sequences:

The sequence for the HumanEyelessProtein is:
`"HSGVNQLGGVFVNGRPLPDSTRQKIVELAHSGARPCDISRILQVSNGCVSKILGRYYETGSIRPRAIGGSKPRVATPEVVSKIAQYKRECPSIFAWEIRDRLLSEGVCTNDNIPSVSSINRVLRNLASEK-QQ"`.

The sequence for the FruitflyEyelessProtein is:
`"HSGVNQLGGVFVGGRPLPDSTRQKIVELAHSGARPCDISRILQVSNGCVSKILGRYYETGSIRPRAIGGSKPRVATAEVVSKISQYKRECPSIFAWEIRDRLLQENVCTNDNIPSVSSINRVLRNLAAQKEQQ"`.

Note that the local alignment is unique and the two submitted sequences should match exactly. Submitted answers that do not include the '-' in the human alignment should be counted as incorrect.You may wish to copy and paste the answer above in CodeSkulptor or desktop Python to do an exact comparison.

Comments: Please enter an explanation for your scoring, especially if you deducted any points for one of the rubric items for this question.

You've written 0 words

Question 2 (2 pts)

To continue our investigation, we next consider the similarity of the two sequences in the local alignment computed in Question 1 to a third sequence. The file ConsensusPAXDomain (http://storage.googleapis.com/codeskulptor-alg/alg_ConsensusPAXDomain.txt) contains a "consensus" sequence of the PAX domain; that is, the sequence of amino acids in the PAX domain in any organism. In this problem, we will compare each of the two sequences of the local alignment computed in Question 1 to this consensus sequence to determine whether they correspond to the PAX domain.

Load the file ConsensusPAXDomain. For each of the two sequences of the local alignment computed in Question 1, do the following:

- Delete any dashes `'-'` present in the sequence.
- Compute the **global alignment** of this dash-less sequence with the ConsensusPAXDomain sequence.
- Compare corresponding elements of these two globally-aligned sequences (local vs. consensus) and compute the percentage of elements in these two sequences that agree.

To reiterate, you will compute the global alignments of local human vs. consensus PAX domain as well as local fruitfly vs. consensus PAX domain. Your answer should be two percentages: one for each global alignment. Enter each percentage below. Be sure to label each answer clearly and include three significant digits of precision.

Human vs Consensus agree = 72.932%
Fly vs Consensus agree = 70.149%

flag (https://accounts.coursera.org/i/zendesk/courserahelp?return_to=https://learner.coursera.help/hc/articles/201212339-Peer-Graded-Assignments)

Evaluation/feedback on the

above work

Item a (2 pts) Your answer should be two percentages: one for each sequence in the local alignment (human and fruitfly) computed in Question 1. Enter each percentage below. Be sure to clearly label each answer and include three significant digits of precision.

When globally-aligned, the HumanEyelessProtein sequence from the local alignment agrees with 72.9% of the ConsensusPAXDomain. The FruitflyEyelessProtein sequence from the local alignment agrees with 70.1% of the ConsensusPAXDomain when the two are globally aligned.

For scoring purposes, count any answer for the human/consensus that is between 72% and 73% as being correct, and count any answer for the fruitfly/consensus that is between 70% and 71% as being correct. Answers in decimal form are also acceptable.

Comments: Please enter an explanation for your scoring, especially if you deducted any points for one of the rubric items for this question.

You've written 0 words

Question 3 (1 pt)

Examine your answers to Questions 1 and 2. Is it likely that the level of similarity exhibited by the answers could have been due to chance? In particular, if you were comparing two random sequences of amino acids of length similar to that of HumanEyelessProtein and FruitflyEyelessProtein, would the level of agreement in these answers be likely? To help you in your analysis, there are 23 amino acids with symbols in the string (`"ACBEDGFIHKMLNQPSRTWVYXZ"`). Include a short justification for your answer.

The similarities exhibited in the answers received in Question 1 and 2 are not by chance.

After creating a short function which generated 2 random protein strings, checking the agreement between the two, then store the results iterated 1000 times, and returning the average of the results, returned an average consensus of 0.649%

flag (https://accounts.coursera.org/i/zendesk/courserahelp?return_to=https://learner.coursera.help/hc/articles/201212339-Peer-Graded-Assignments)

Evaluation/feedback on the above work

Item a (1 pt) Examine your answers to Questions 1 and 2. Is it likely that the level of similarity exhibited by the answers could have been due to chance? Include a short informal justification for your answer.

The level of agreement of each local alignment to the ConsensusPAXDomain is much too high for this situation to have arisen by chance. Each alignment agrees on 90+ amino acids for sequences of length 130+. The chance that this many elements in each local alignment would agree with ConsensusPAXDomain due to chance would be vanishingly small for an alphabet of size 23, especially given so few dashes.

For the local human vs. consensus PAX domain alignment, the chance (http://www.codeskulptor.org/#alg_min_matches.py) of 97 corresponding elements matching between two 133 element sequences whose elements are chosen at random from a 23 character alphabet is less than 10^{-100}!.

In evaluating the plausibility of the provided justification, examine the answer for a mention of the lengths of the sequences, the size of the alphabet, and the level of agreement. An estimate of the actual probability of agreement is not necessary. However, the answer should state the chance of agreement due to random matching is very, very small. If in doubt on this item, be generous.

Comments: Please enter an explanation for your scoring, especially if you deducted any points for one of the rubric items for this question.

You've written 0 words

Hypothesis testing

One weakness of our approach in Question 3 was that we assumed that the probability of any particular amino acid appearing at a particular location in a protein was equal. In the next two questions, we will consider a more mathematical approach to answering Question 3 that avoids this assumption. In particular, we will take an approach known as statistical hypothesis testing (http://en.wikipedia.org/wiki/Statistical_hypothesis_testing) to determine whether the local alignments computed in Question 1 are statistically significant (that is, that the probability that they could have arisen by chance is extremely small).

Question 4 (2 pts)

Write a function `generate_null_distribution(seq_x, seq_y, scoring_matrix, num_trials)` that takes as input two sequences `seq_x` and `seq_y`, a scoring matrix `scoring_matrix`, and a number of trials `num_trials`. This function should return a dictionary `scoring_distribution` that represents an un-normalized distribution generated by performing the following process `num_trials` times:

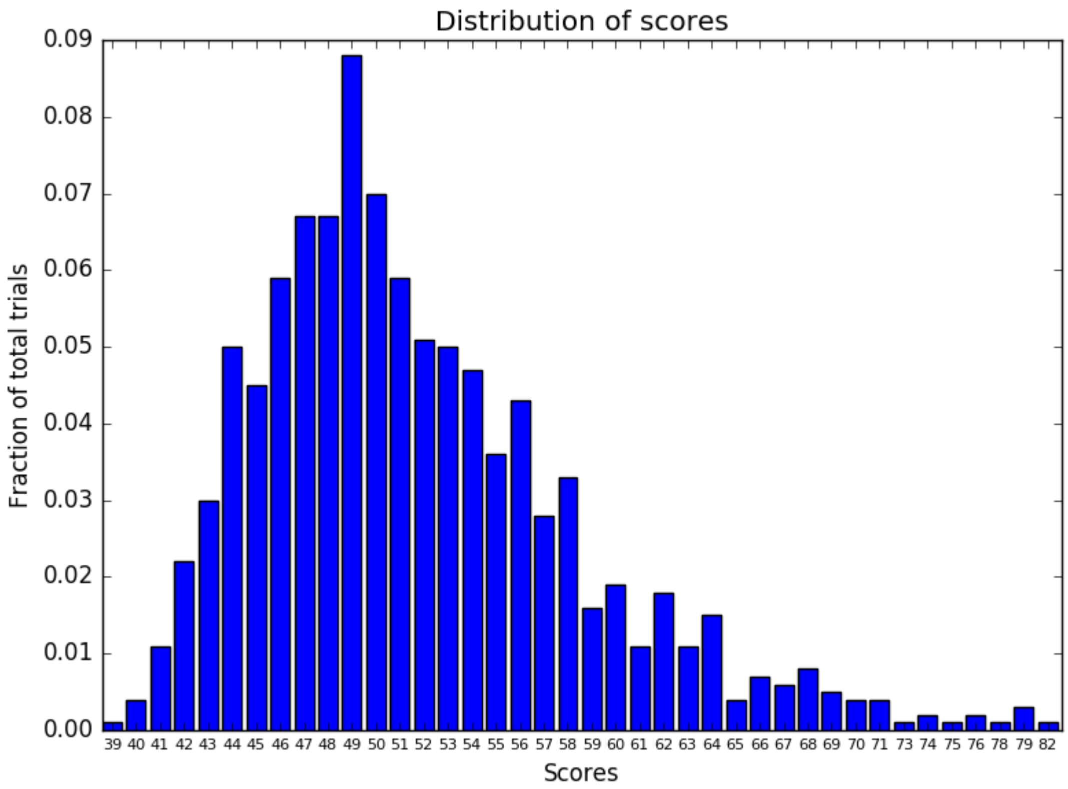
- Generate a random permutation `rand_y` of the sequence `seq_y` using `random.shuffle()`.
- Compute the maximum value `score` for the **local alignment** of `seq_x` and `rand_y` using the score matrix `scoring_matrix`.
- Increment the entry `score` in the dictionary `scoring_distribution` by one.

Use the function `generate_null_distribution` to create a distribution with 1000 trials using the protein sequences HumanEyelessProtein and FruitflyEyelessProtein (using the PAM50 scoring matrix). **Important:** Use HumanEyelessProtein as the first parameter `seq_x` (which stays fixed) and FruitflyEyelessProtein as the second parameter `seq_y` (which is randomly shuffled) when calling `generate_null_distribution`. Switching the order of these two parameters will lead to a slightly different answers for question 5 that may lie outside the accepted ranges for correct answers.

Next, create a bar plot of the **normalized** version of this distribution using `plt.bar` in `matplotlib` (or your favorite plotting tool). (You will probably find CodeSkulptor too slow to do the required number of trials.) The horizontal axis should be the scores and the vertical axis should be the fraction of total trials corresponding to each score. As usual, choose reasonable labels for the axes and title. **Note:** You may wish to save the distribution that you compute in this Question for later use in Question 5.

Once you have created your bar plot, upload your plot in the box below using "Attach a file" button (the button is disabled under the 'html' edit mode; you must be under the 'Rich' edit mode for the button to be enabled). Please review the class guidelines for formatting and comparing plots on the "Creating, formatting, and comparing plots" class page. These guidelines cover the basics of good formatting practices for plots. Your plot will be assessed based on the answers to the following two questions:

- Does the plot follow the formatting guidelines for plots?
- Is the shape of the plot correct?



flag (https://accounts.coursera.org/i/zendesk/courserahelp?return_to=https://learner.coursera.help/hc/articles/201212339-Peer-Graded-Assignments)

Evaluation/feedback on the

above work

Item a (1 pt) Does the plot follow the formatting guidelines for plots?

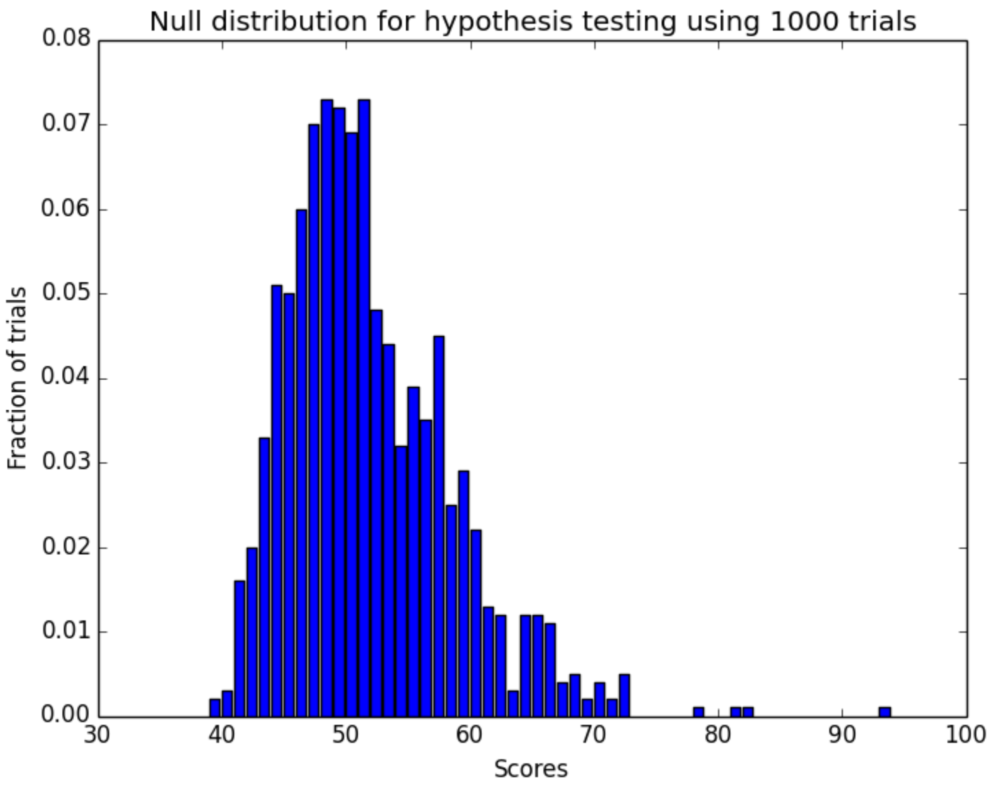
The formatting guidelines include the following items:

- The plot is an image and not a text file.
- The plot is appropriately trimmed. Showing the boundary of the plot's window is fine. However, the plot should not include part of the desktop.
- The elements of the plot are of the correct type. Line plots are not the same as point plots.
- Both axes should have tick marks labeled by regularly-spaced coordinate values.
- Both axes have appropriate text labels that describe the quantities being plotted.
- The plot has an appropriate title that describes the content of the plot.
- The plot has an appropriate legend (when required) that distinguishes the various components of the plot.

Assess the submitted plot based on these guidelines. Note that the submitted plot should be a bar plot that represents a normalized (vertical range between 0 and 1) distribution. If not, give no credit for this item.

Item b (1 pt) Is the shape of the plot correct?

Below is a bar plot created using 1000 trials with `generate_null_distribution`. Note that there will be some variation due to the use of random trials.



When scoring the submitted plot, check for plots that are roughly bell-shaped in a manner similar to that of a normal distribution, but slightly asymmetric. The left-hand side of the distribution should drop to zero much more quickly than the right-hand side of the distribution, which trails off towards zero more slowly. The vast majority of the distribution should span a range from approximately 35 to 90 with a peak roughly centered at 50.

If the plot appears to have used substantially fewer trials (as indicated by a very noisy distribution), you may score this item as incorrect at your discretion.

Comments: Please enter an explanation for your scoring, especially if you deducted any points for one of the rubric items for this question.

You've written 0 words

Question 5 (2 pts)

Given the distribution computed in Question 4, we can do some very basic statistical analysis of this distribution to help us understand how likely the local alignment score from Question 1 is. To this end, we first compute the *mean* μ and the *standard deviation* σ of this distribution via:

$$\mu = \frac{1}{n} \sum_i s_i ,$$

$$\sigma = \sqrt{\frac{1}{n} \sum_i (s_i - \mu)^2} ,$$

where the values s_i are the scores returned by the n trials. If s is the score of the local alignment for the human eyeless protein and the fruitfly eyeless protein, the z-score (http://en.wikipedia.org/wiki/Standard_score) z for this alignment is

$$z = \frac{s-\mu}{\sigma} .$$

The z-score helps quantify the likelihood of the score s being a product of chance. Small z-scores indicate a greater likelihood that the local alignment score was due to chance while larger scores indicate a lower likelihood that the local alignment score was due to chance.

- What are the mean and standard deviation for the distribution that you computed in Question 4?
- What is the z-score for the local alignment for the human eyeless protein vs. the fruitfly eyeless protein based on these values?

mean = 51.000000
standard deviation = 6.782330
z-score = 121.492172

flag (https://accounts.coursera.org/i/zendesk/courserahelp?return_to=https://learner.coursera.help/hc/articles/201212339-Peer-Graded-Assignments)

Evaluation/feedback on the

above work

Item a (1 pt) What are the mean and standard deviation for the distribution that you computed in Question 4?

The mean for our distribution in Question 4 is approximately 52. The standard deviation is approximately 6.8. Since the distribution may vary due to the use of random trials, score any mean in the range $[50, 55]$ as correct and any standard deviation in the range $[5.7, 8.0]$ as being correct. (These are very generous bounds.)

Item b (1 pt) What is the z-score for the local alignment for the human eyeless protein vs. the fruitfly eyeless protein based on these values?

For our computed distribution, the z-score was approximately 122. (Remember that s was 875.) Since this z-score may vary some due to the use of random trials, please score any z-score in the range $[100, 155]$ as being correct.

Comments: Please enter an explanation for your scoring, especially if you deducted any points for one of the rubric items for this question.

You've written 0 words

Question 6 (1 pt)

For bell-shaped distributions such as the normal distribution (http://en.wikipedia.org/wiki/Normal_distribution), the likelihood that an observation will fall within three multiples of the standard deviation for such distributions is very high (http://en.wikipedia.org/wiki/Standard_deviation#Rules_for_normally_distributed_data).

Based on your answers to Questions 4 and 5, is the score resulting from the local alignment of the HumanEyelessProtein and the FruitflyEyelessProtein due to chance? As a concrete question, which is more likely: the similarity between the human eyeless protein and the fruitfly eyeless protein being due to chance or winning the jackpot in an extremely large lottery (http://en.wikipedia.org/wiki/Lottery_jackpot_records)? Provide a short explanation for your answers.

It is due to chance.

"Meaningful words and sentences are rare among the set of possible combinations of English letters,particularly as the sequence grows. The ratio of meaningful 12-letter words to 12-letter sequences is 1/1014, the ratio of 100-letter sentences to 100-letter strings is 1/10100. Meaningful sentences are highly isolated from one another in the space of possible combinations, so that random substitutions degrade meaning. While some closely clustered sentences may be accessible by random substitution, the overwhelming majority of meaningful sentences lie beyond the reach of random search" (http://www.darwinsmaths.com/#ch10 (http://www.darwinsmaths.com/#ch10))[http://www.darwinsmaths.com/#r47 (http://www.darwinsmaths.com/#r47)] flag (https://accounts.coursera.org/i/zendesk/courserahelp?return_to=https://learner.coursera.help/hc/articles/201212339-Peer-Graded-Assignments)

Evaluation/feedback on the

above work

Item a (1 pt) Based on your answers to Questions 4 and 5, is the score resulting from the local alignment of the HumanEyelessProtein and the FruitflyEyelessProtein due to chance? As a concrete question, which is more likely: the similarity between the human eyeless protein and the fruitfly eyeless protein being due to chance or winning the jackpot in an extremely large lottery (http://en.wikipedia.org/wiki/Lottery_jackpot_records)?

The distribution of scores is close enough to being bell-shaped that we will assume that 99% of the scores are withing three standard deviations of the mean for this distribution. Based on the z-score, the actual score for the Human/Fruitfly alignment is actually more than 100 standard deviations away from the mean of the distribution. If we assume that each multiple of three standard deviations reduces the likelihood of this score arising randomly by a factor of 10^{-2}, the resulting probability is on the order of approximately 10^{-67}. (In reality, the probability is much, much smaller (http://www.wolframalpha.com/input/?i=Probability+of+120+standard+deviations).)

As a comparison, the odds of winning even the largest lottery are certainly more than one in a trillion (i.e; 10^{-12}). So, winning the jackpot in the world's largest lottery is much more likely.

When assessing the submitted answer check that the submitted answer mentions the fact that the score distribution has approximately bell-shaped (looks like a normal distribution) and that the z-score indicates that the true score is many standard deviations away from the mean.

Comments: Please enter an explanation for your scoring, especially if you deducted any points for one of the rubric items for this question.

You've written 0 words

Spelling correction

Up to now, we have measured the similarity of two strings. In other applications, measuring the dissimilarity of two sequences is also useful. Given two strings, the edit distance (http://en.wikipedia.org/wiki/Edit_distance) corresponds to the minimum number of single character insertions, deletions, and substitutions that are needed to transform one string into another. In particular, if x and y are strings and a and b are characters, these edit operations have the form:

- **Insert** - Replace the string $x + y$ by the string $x + a + y$.
- **Delete** - Replace the string $x + a + y$ by the string $x + y$.
- **Substitute** - Replace the string $x + a + y$ by the string $x + b + y$,

Question 7 (3 pts)

Not surprisingly, similarity between pairs of sequences and edit distances between pairs of strings are related. In particular, the edit distance for two strings x and y can be expressed in terms of the lengths of the two strings and their corresponding similarity score as follows:

$$|x| + |y| - score(x, y)$$

where $score(x, y)$ is the score returned by the global alignment of these two strings using a very simple scoring matrix that can be computed using `build_scoring_matrix`.

Determine the values for `diag_score`, `off_diag_score`, and `dash_score` such that the score from the resulting global alignment yields the edit distance when substituted into the formula above. Be sure to indicate which values corresponds to which parameters. Finally, as a side note, note that there are alternative formulations of edit distance as a dynamic programming problem using different scoring matrices. For this problem, please restrict your consideration to the formulation used above.

diag_score = 2

off_diag_score = 1
dash_score = 0

flag (https://accounts.coursera.org/i/zendesk/courserahelp?return_to=https://learner.coursera.help/hc/articles/201212339-Peer-Graded-Assignments)

Evaluation/feedback on the

above work

Item a (3 pts) Determine the values for `diag_score`, `off_diag_score`, and `dash_score` such that the score from the resulting global alignment can be used to compute the edit distance via the formula above.

The correct values for the three types of entries in the scoring matrix are:

- `diag_score` is exactly 2,
- `off_diag_score` is exactly 1,
- `dash_score` is exactly 0.

The key to understanding the correctness of these values is to score the global alignments produced by this distance matrix on an character-by-character basis and compare this score to $|x| + |y|$. If two corresponding non-dash characters agree, the scoring matrix scores that match as 2. Note that these two matching characters also increase the size of $|x| + |y|$ by exactly two, leading to no increase in the edit distance.

If two corresponding non-dash characters disagree, the scoring matrix scores the match as 1. Since these two non-matching characters also increase the size of $|x| + |y|$ by exactly two, the edit distance is increased by one corresponding to the fact that a substitution is necessary. Finally, if a non-dashed character matches a dash, the scoring matrix scores this match as 0. Since the single non-dash character increases the size of size of $|x| + |y|$ by exactly one, the edit distance is increased by one corresponding to the fact that an insertion or deletion is necessary.

Comments: Please enter an explanation for your scoring, especially if you deducted any points for one of the rubric items for this question.

You've written 0 words

Question 8 (2 pts)

In practice, edit distance is a useful tool in applications such as spelling correction and plagiarism detection where determining whether two strings are similar/dissimilar is important. For this final question, we will implement a simple spelling correction function that uses edit distance to determine whether a given string is the misspelling of a word.

To begin, load this list (http://storage.googleapis.com/codeskulptor-assets/assets_scrabble_words3.txt) of 79339 words. Then, write a function `check_spelling(checked_word, dist, word_list)` that iterates through `word_list` and returns the set of all words that are within edit distance `dist` of the string `checked_word`.

Use your function `check_spelling` to compute the set of words within an edit distance of one from the string `"humble"` and the set of words within an edit distance of two from the string `"firefly"`. (Note this is not `"fruitfly"`.)

Enter these two sets of words in the box below. As quick check, both sets should include eleven words.

```
(11, set(['bumble', 'humbled', 'tumble', 'humble', 'rumble', 'humbler', 'humbles', 'fumble', 'humbly', 'jumble', 'mumble']))
(11, set(['firefly', 'tiredly', 'freely', 'fireclay', 'direly', 'finely', 'firstly', 'liefly', 'fixedly', 'refly', 'firmly']))
```

flag (https://accounts.coursera.org/i/zendesk/courserahelp?return_to=https://learner.coursera.help/hc/articles/201212339-Peer-Graded-Assignments)

Evaluation/feedback on the

above work

Item a (2 pts) Use your function `check_spelling` to compute the set of words with an edit distance of one from the string `"humble"` and the set of words with an edit distance of two from the string `"firefly"`. (Note this is not `"fruitfly"`.)

The set of words within edit distance one from the string `"humble"` is `set(['bumble', 'fumble', 'humble', 'humbled', 'humbler', 'humbles', 'humbly', 'jumble', 'mumble', 'rumble', 'tumble'])`.

The set of words within an edit distance of two for the string `"firefly"` is `set(['direly', 'finely', 'fireclay', 'firefly', 'firmly', 'firstly', 'fixedly', 'freely', 'liefly', 'refly', 'tiredly'])`.

When evaluating this item, each submitted word should exactly correspond to one of the words in the solution set. However, note the ordering of the submitted words is unimportant. The submitted answer does not have to appear in a particular format such as a set in Python.

Comments: Please enter an explanation for your scoring, especially if you deducted any points for one of the rubric items for this question.

You've written 0 words

Question 9 (1 pt Extra Credit)

As you may have noted in Question 8, your spelling correction function is not particularly responsive. In particular, the function may require several seconds to compute a set of possible corrections. This slow performance is due to the need to iterate through the entire list of provided words.

Reconsider the formulation of question 8 from a more general point of view and design a spelling correction tool that would provide real-time (almost instantaneous) correction of spelling errors within an edit distance of two. To guide you in the correct direction, we will provide two hints. First, you should convert your list of provided words to a set of words to enable a fast check for whether a string is a valid word. Second, you do not need to use dynamic programming to solve this problem. However, you will need to focus on the structure of the three editing operations described in Question 7.

Please provide an English description of your approach to this problem. You may also include pseudo-code if you so desire. You do not need to implement your algorithm in Python.

I would use a probability approach.
Creating a dictionary of how many times each words occurs in a text.
I would then process the word being checked by creating a set of words that have an edit distance of 1 and 2.
I would then cross-reference the set with the dictionary of probabilities, and return the word with the highest probability.

flag (https://accounts.coursera.org/i/zendesk/courserahelp?return_to=https://learner.coursera.help/hc/articles/201212339-Peer-Graded-Assignments)

Evaluation/feedback on the

above work

Item a (1 pt) Reconsider the formulation of question 8 from a more general point of view and design a spelling correction tool that would provide real-time (almost instantaneous) correction of spelling errors. To guide you in the correct direction, we will provide two hints. First, you should convert your list of provided words to a set of words to enable a fast check for whether a string is a valid word. Second, you do not need to use dynamic programming to solve this problem. However, you will need to focus on the structure of the three editing operations describes in Question 7.

The key idea behind this improved spell checker is that we can detect whether a particular string is a valid word in $O(1)$ time by representing the word list as a set. Now, given a string that needs to be spell checked, we can first test whether the string is a valid word quickly. If so, we are done. If the string is not a valid word, we can use the editing operations in question 7 to generate all possible strings that are one edit away from the given string. Then, we can iterate through these strings to check whether any of these strings are valid words. This approach is also fast since the number of possible strings that are one edit away is still relatively small.

For words that are two edits away, we take the input string and enumerate all strings that are one edit away. Then, we take those strings and enumerate all strings that are one edit away from those strings. The result is the set of strings which are two edits away. We can then check whether those strings are in the provided word list. This page (http://norvig.com/spell-correct.html) gives more details on how this approach is used by Google to do spelling correction very efficiently. The algorithm that Google uses also accounts for the fact that some words are used more often than others.

In scoring this question, the key observation to check for is whether the answer suggested building the set of all strings that are two edits away from the given string. If this observation is present in the explanation, give credit. If another method is proposed, use your judgment in deciding whether to award credit. Please add a comment describing your analysis of the proposed solution.

Comments: Please enter an explanation for your scoring, especially if you deducted any points for one of the rubric items for this question.

You've written 0 words

Save draft

Submit evaluation