

Assignment 1

posted Friday 8 April 2016

due Friday 15 April 2016 at midnight

Submission policy: Report all plots (Question II parts 2, 3, 4, 5.1 and Question IV part 2) and your code in [this iPython notebook](#). Print your notebook as a PDF and attach it to the rest of your assignment. Turn in your assignment on the 2nd floor of Packard in the EE372 bin next to the kitchen area.

Question I: Sanger sequencing

1. In a Sanger sequencing experiment, a bio-chemist observes that the masses of molecules that are terminated by ddATP are 400, 1200, 1351. The masses of molecules terminated by ddCTP are 531, 1482. The masses of the molecules terminated by ddGTP are 671, 813, 961. The masses of the molecules terminated by ddTTP are 1093, 1657. The primer used here is AGC. What is the molecule being sequenced?
2. Assuming that A, G, C, T have the same molecular weight, and the masses measured have a tolerance of $\pm 0.05\%$. Give a bound on the maximum length that can be sequenced without error (assuming all measurements are within $\pm 0.05\%$ of the true value). How does this change when the molecular weights are different?

Question II: Base calling for Illumina sequencing

Consider the following base calling model studied in the class. We will focus on the A channel. Let $s(1), \dots, s(L)$ be the binary sequence, obtained by setting $s(t) = 1$ if the t -th base is an A and 0 otherwise, and $y(1), y(2), \dots, y(L)$ be the sequence of intensities observed in the A channel. Now, in the case of a large number N of strands in a cluster, the intensities and the DNA sequence can be related approximately as follows

$$y(t) = \sum_j Q_{jt} s(j) + n(t), \quad t = 1, 2, \dots, L,$$

where Q_{jt} is the probability that after t -cycles, the template has synthesized j nucleotides. Further assume that there is no possibility of a template leading, only lagging, *i.e.*, in the notation of the class $q = 0$. Assume $n(t)$ is Gaussian noise with zero mean and variance σ^2 (we are neglecting the effect that $y(t)$ is forced to be a positive real number, and we are also ignoring the amplitude a as that factor can be absorbed into σ^2 with a rescaling).

1. Calculate Q_{jt} in terms of j , t , and p . Given t and p , at which value of j is Q_{jt} maximized? Is this intuitive?
2. Simulate and plot $y(1), \dots, y(L)$ according to the probability model (for $s(t)$ being i.i.d. equally probable to be 0 or 1). Do this for various values of $p = 0, 0.01, 0.05, 0.1$, and 0.2 with $\sigma^2 = 0.1$.
3. Write down the zero-forcing equalizer (*i.e.* matrix inversion) and the decoding rule. Simulate this rule and for different values of p plot its quality score as a function of position along the DNA sequence. Here the quality score Q of a base is defined to be:

$$Q = -10 \log_{10} p_e$$

where p_e is the probability of error of detecting the base. Do the quality scores increase or decrease with the position of the base? Give an intuitive explanation.

4. Write down the formula for the MMSE equalizer and the corresponding decoding rule. Simulate this rule and for different values of p plot the resulting quality score as a function of position along the DNA sequence. Compare it to the rule above.
5. (Matched filter bound): In this section, we will try to calculate a lower bound on the probability of error for any rule. To do so, we invoke a bound called as the matched filter bound in communication. Consider the following system. Suppose you want to decode $s(m)$ for a particular m . If there were no interference from any other symbol but you observe the intensities at all possible times, then we have

$$\tilde{y}(t) = Q_{mt} s(m) + n(t), \quad t = 1, 2, \dots, L.$$

Given these observations, the optimal combining rule is called the matched filter rule in which a weighted average of the intensities

$$y_m = \sum_t Q_{mt} \tilde{y}(t)$$

is calculated and followed by an appropriate detection rule to perform base calling. The probability of error of this rule is a lower bound to the probability of error of the

optimal rule in the original problem because ignoring interference from other symbols will only improve performance.

1. Find the appropriate detection rule and give an expression for the probability of error and the quality score of the optimal combining rule. Plot the quality score for a fixed $p = 0.05$ as a function of the position m . Compare this to the performance of the base calling rules in parts 3. and 4.
2. What happens to this probability of error as a function of position in this case? What does this say about why the read length in Illumina sequencing is limited?

Question III: Playing around with reads

Download the reference genome for *E. coli* [here](#). Download a set of reads obtained from an *E. coli* experiment [here](#). You can right click each link and select "Save Link As".

- What is the length of the reference?
- What is the length of each read?
- How many reads are there?
- What is the maximum number of times a read is repeated?
- What is the sequencing depth of each base in the reference for this experiment?


Question IV: Read alignment with Bowtie

For this question, use the *E.coli* genome and reads used in Question III.

1. We are given N reads of length L and a reference genome of length ℓ . Assuming reads were sampled uniformly from the entire genome, what is the expected number of times a base at a particular position will be sequenced? In other words, what is the *sequencing depth* of each base in the genome? What is the probability that we see the exact same read twice? You can assume that if a length- L sequence appears in the genome, it appears exactly once.
2. How many distinct 20-length substrings do you see across all reads? These substrings are commonly referred to as k -mers ($k = 20$ in this case). Count how often each distinct 20-mer appears and generate a histogram of the counts. *Hint*: Note that initializing a length- 4^{20} array may not be a viable approach. Consider using dictionaries!
3. [Bowtie](#) is a popular read aligner optimized for aligning large amounts of short reads to long references. Bowtie is preinstalled on Stanford's Corn cluster, but you can also install Bowtie on your local machine by downloading the [binary](#).

- Build a Bowtie index from the *E. coli* reference genome (“bowtie-build” command). You can copy the downloaded files from your computer to Corn using the [scp](#) command.
 - Using the default settings, use Bowtie to align the *E. coli* reads to the newly built Bowtie index. Use Bowtie’s “-t” option to obtain the runtime. How many reads have at least 1 reported alignment? What was the runtime?
4. [BONUS] Visually prove or disprove whether the reads are uniformly distributed across the reference. *Hint:* Use a sam/bam visualiser like [IGV](#) or [Bamview](#).
-

[This assignment as a pdf.](#)

 [data-science-sequencing](#)
{ [gkamath](#), [jessez](#), [dntse](#) }
[@stanford.edu](#)