# Assignment 1

posted Friday 8 April 2016
due Sunday 17 April 2016 at midnight

## 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

Consider the following base calling model studied in the class. Let $s_1, \cdots, s_L$ be the binary sequence corresponding to a base (obtained, for example, by setting $s_t = 1$ if the $t$-th base is an $A$ and 0 otherwise) and $I_1, \ldots, I_L$ be the sequence of intensities observed (for example, in the $A$ channel). Now, the intensities and the DNA sequence can be related as follows:

$$I_t = \sum_{j=1}^{L} Q_{jt} s_j + n_t, \quad t = 1, 2, \cdots, L,$$

where $Q_{jt}$ is the probability that the 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 $q = 0$. Recall that this implies

$$Q_{jt} = \binom{t}{j}(1-p)^j p^{t-j}.$$

Assume $n_t$ is Gaussian noise with zero mean, and variance $\sigma^2$ (we are neglecting the effect that the observed intensity $I_t$ is forced to be a positive real number).

1. Simulate $N$ strands of DNA each of length $L$ with each base in each strand chosen *i.i.d.* to be $\{0,1\}$ with probability $\{1/4,3/4\}$. Let $N=100, L=100$ For each of these $L$ length sequences, generate $I_1,\dots,I_L$ according to the probability model. Do this for various values of $p = 0, 0.01, 0.05, 0.1, 0.2$
2. Write down the zero-forcing equalizer (*i.e.* the matrix inversion decoder) and the decoding rule. Using the proposed rule, decode the bases. Calculate the probability of error as a function of $p$.
3. Write down the formula for the MMSE equalizer and the corresponding decoding rule. Using the proposed rule, decode the bases. Calculate the probability of error as a function of $p$.
4. (Bounds on the optimal rule): 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 the symbol $s_m$. If there were no interference from any other symbol but you observe the intensities at all possible times, then we have

   $$I_t = Q_{mt}s_m + n_t.$$

   Given these observations, the optimal combining rule is called the matched filter rule, in which is the following,

   $$y_m = \frac{1}{A_m}\sum_t Q_{mt}I_t = \frac{1}{A_m}\sum_t \left(Q_{mt}^2 s_m + Q_{mt}n_t\right),$$

   where $A_m = \sqrt{\left(\sum_t Q_{mt}^2\right)}$. Thus the overall noise variance is still $\sigma^2$ and the overall signal power is $A_m$ times the original signal power.
   1. Calculate $A_m$ as a function of $m$ for $m = 1, 2, \cdots, L$ and plot this for a fixed $p = 0.1$.
   2. Calculate the implied probability of error for decoding the signal under this matched filter model. This is a lower bound on the probability of error for any signal.

## Question III: Read alignment

Python programming is required for all of part a) and question 3 in part b). Please write your code in this iPython notebook and email the completed assignment to ee372-spr1516-staff@lists.stanford.edu. Save your completed notebook as "ee372_assignment1q3_FIRSTNAME_LASTNAME.ipynb".

### a) Naive aligner

1. Write a function that computes the edit distance between two strings. You can use standard libraries such as editdistance to check the correctness of your function. The input should be two strings each of at least length 2. The output should be a float representing the edit distance between the two input strings.
2. Write a function that generates random reads. The input should be the number of reads generated $N$ and the length $L$ of each read generated. The output should be $N$ random length-$L$ sequences of nucleotides.
3. How can you use the above two functions to perform alignment? What issues do you see with this approach? (3-4 sentences).
4. For $L$ = 10, generate two reads of length $L$ and compute their edit distance. Average your result over 100 runs to obtain an estimate of the average edit distance $\hat{d}_L$ of two randomly generated reads of length $L$. Repeat for $L$ = 1, 100, 1000, and 10000. Plot $\hat{d}_L$ as a function of $L$ with error bars. Do you observe any trends? What can you say about how well random strings align? (1-2 sentences).

### b) Bowtie

1. We are given $N$ reads of length $L$ and a reference genome of length $\ell$. 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. 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?

3. How many distinct 20-length substrings do you see across all reads? These substrings are commonly referred to as $k$-mers where $k$ = 20. 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!)
4. 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?
5. [BONUS] Can you give a profile of the reads on the reference genome? *Hint:* Use a sam/bam visualiser like IGV or Bamview.

data-science-sequencing
{ gkamath, jessez, dntse }
@stanford.edu