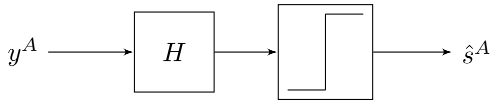
Recap

Last time, we introduced a model for the errors introduced by the sequencing process as:

$$y^A = Qs^A + n$$

where s^A is the 0-1 vector describing the locations of A's in the DNA sequence, Q is a matrix representing errors in the read process, and n is $\mathcal{N}(0, \sigma^2)$ Gaussian noise. y^A is the data we observe, in the form of intensities of light captured from photos of the assay plate.

We considered models of the form:



But the problem with these models is that they ignore the fact that errors propagate and therefore symbol-by-symbol decoding is suboptimal.

Maximum Likelihood

To find an optimal decoder, we appeal to the ideal of **maximum likelihood**, which asks us to find the sequence s^A that is the solution of the problem:

$$\max_{s^A \in \{0,1\}^L} p(y^A | s^A)$$

Since $y^A | s^A \sim \mathcal{N}(Qs^A, \sigma^2)$, we have:

$$p(y^A | s^A) = \frac{1}{(2\pi)^{n/2}\sigma} \exp\left(-\frac{1}{2\sigma^2} \|y^A - Qs^A\|^2 \right)$$

Our optimization problem then reduces to:

$$\min_{s^A \in \{0,1\}^L} \|y^A - Qs^A\|^2$$

This is a discrete optimization problem with brute-force complexity $O(2^L)$, which is intractable even for relatively small L (~ 200).

However, we can exploit the following fact about the matrix Q. Remember that the values of the off-diagonals of Q get exponentially smaller as we move away from the diagonal - this is essentially because for a value of 1 at $s^A(i)$ to influence the reading of $y^A(j)$, there would need to be $|i - j|$ independent failures in the sequencing process, each with its own (small) probability.

Therefore, it is reasonable to approximate the matrix Q as **band-diagonal**. We consider the case when only the main diagonal and the first lower diagonal Q (i.e. $Q_{i(i-1)}$, for $i = 2..., L$) are nonzero, with the rest assumed to be zero.

For simplicity, in the following section, we drop the superscript A from y^A and s^A.

Given this band-diagonal assumption, the equation

$$y = Qs + n$$

can be expanded as:

$$\begin{aligned} y(1) &= Q_{11}s(1) + n(1) \\ y(2) &= Q_{22}s(2) + Q_{21}s(1) + n(2) \\ y(3) &= Q_{33}s(3) + Q_{32}s(2) + n(2) \\ &\vdots \end{aligned}$$

Now, we can express the objective $\|y - Qs\|$ as:

$$\sum_{i=2}^{L} \left[y(i) - Q_{ii}s(i) - Q_{i(i-1)}s(i-1)\right]^2 + (y(1) - Q_{11}s(1))^2$$

The magical thing is that we can minimize this objective efficiently!

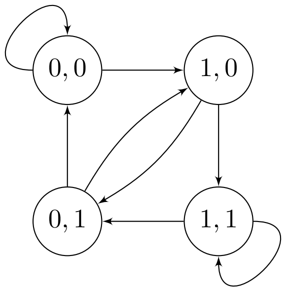
The Key Idea: State

To solve this problem, we introduce the idea of a **state**. Noticing that in the objective above, each term of the sum only depends on the value of $(s(i), s(i-1))$ - we will use this pair as our state. Alternatively, we can think of the state

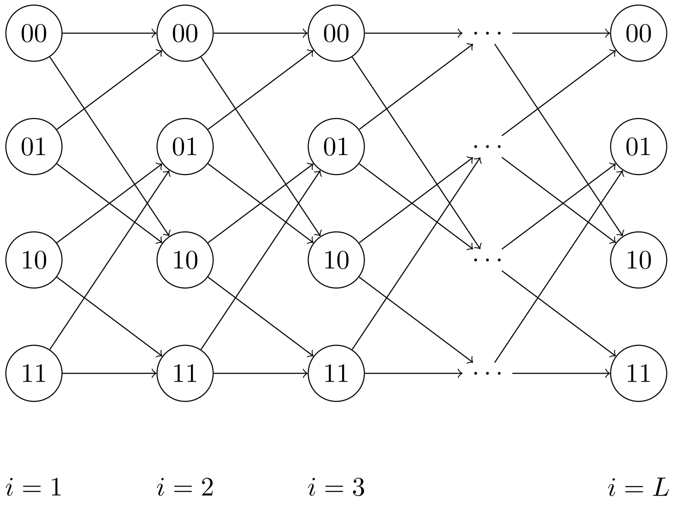
$$(s(i), s(i-1))$$

as encapsulating all the information we need to produce a single (noisy) measurement $y(i)$ (because of the band-diagonal structure of our matrix Q).

Now, instead of a sequence of 0's and 1's $s(1), s(2), ..., s(L)$, we have a sequence of states $(s(1), s(2)), (s(2), s(3)), ...$ In the first sequence, the value of $s(i)$ does not depend on the value of $s(i-1)$; however in the second sequence, not all state transitions are allowed. They are summarized in the following diagram:



The idea of state is powerful because it allows us to reformulate our problem as a shortest-path problem. Let the **cost of a state** $(s(i), s(j))$ be the value of $(y(i) - Q_{ii}s(i) - Q_{i(i-1)}s(i-1))^2$ (or $(y(1) - Q_{11}s(1))^2$ for $i = 1$). Minimizing our objective corresponds to finding the minimum cost path (from left to right) through the **trellis**:



Each column represents a state $s(i)$, and each state within that column is associated with a cost $c_i(s(i), y(i))$ that is a function of our estimate of the state $s(i)$ and our observation $y(i)$. The **minimum cost path** is the path through the graph from left to right that minimizes the sum of the costs of the states through which it passes. This minimum cost path corresponds exactly to the sequence s that minimizes our objective function!

This problem can be solved using dynamic programming (the **Viterbi Algorithm**). in $O(L2^b)$ time, where b is the number of nonzero diagonals of our Q matrix (and hence, 2^b is the number of states at each time step of the trellis). For more info, see `https://en.wikipedia.org/wiki/Convolutional_code`.

Long Read Technologies

We now turn to the next generation of sequencing technologies, which are capable of producing much longer reads (hence the name, **long read** technologies), albeit with higher (10-20%) error rates. These methods are generally capable of quick sequencing rates, and do not require PCR (i.e. they do **single molecule sequencing**).

Nanopore

See video.

Each double-stranded DNA fragment is bound to a special enzyme. The enzyme binds to a nanopore, which is affixed to a substrate. The enzyme then unzips and advances the DNA through the nanopore, one base at a time. A voltage across the nanopore produces a small electrical current - the difference in resistances of different bases allows the DNA sequence to be determined based on this varying current signal.

There are several issues with this technology that make it a challenging (read: interesting) signal processing problem.

- The enzyme's job is to advance the DNA forward at a relatively constant rate. However, sometimes the enzyme moves the DNA forward faster or slower, or even moves it backwards!
- The resistance across the nanopore is a function of the **DNA context** - $\sim$ 4 or so bases - rather than an individual nucleotide.

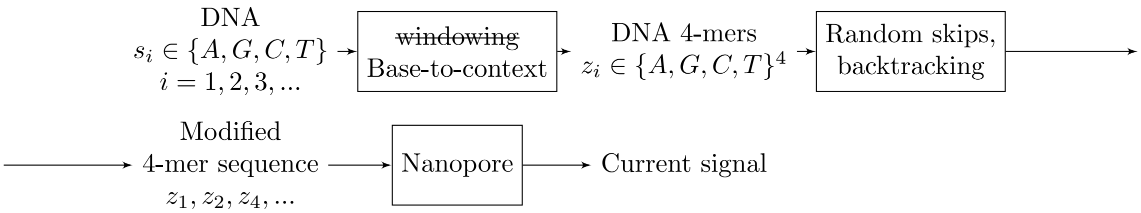
For example, given a sequence $s =$ AGCTTCA..., we're actually looking at a current signal that is the function of the **4-mer sequence**

AGCT, GCTT, CTTC, TTCA,...

possibly with skipping and/or backtracking, so the actual current signal might be a function of

AGCT, GCTT, ~~CTTC~~, TTCA, ...

We will model this channel as follows:



How can we reconstruct the 4-mer sequence from a current signal? We return to the idea of state, discussed previously, except that this time, our state z_i takes values in the set of the 16 possible 4-mers $\{A, G, C, T\}^4$. At each timestep, we collect a noisy measurement of this state z_i, and we can produce a cost that allows us (just like before) to use the Viterbi algorithm to find the maximum likelihood decoding of our sequence of 4-mers.

Zero-mode Waveguide (PacBio)

See video.