1 Introduction

Our biological problem is the distribution of Guanine and Cytosine in a genome sequence which is formed also from Adenine and Thymine. Example of the biological importance of this question is the relatively high proportions of these two nucleotides in organisms of high temperatures. In complex organisms there is a variety of this distribution, for example the ninth chromosome of the human is rich with GC. We would like to draw a graph like Figure 1.

Our assumption is that $\frac{A}{T} = \frac{C}{G} = 1$ and we use two different ways to solve this problem:

1.1 The Uniform Model

We assume that the same coin was used to choose every base, $HEAD = GC$ and $TAIL = AT$. We would like to estimate the ratio:

$$\Theta = \frac{\%GC}{\%AT} = \frac{HEADS}{TAILS}$$

We shall use the principle of maximum likelihood, assuming we know $\Theta$, what is the probability to get the sequence we got. The flips of the coin are independently and identically distributed (i.i.d.). Let $D$ be a group of results, the Data. $D = <H, H, T, \ldots>$

$$P(D : \theta) = P(H : \theta)P(H : \theta)P(T : \theta)\ldots = \theta\theta(1-\theta)\ldots = \theta^#H\theta^#T = \theta^{\sum_n 1\{x_n=H\}}\theta^{\sum_n 1\{x_n=T\}}$$

We define:

$$1\{\ldots\} = \begin{cases} 1 & \text{if } \{\ldots\} \text{ is true} \\ 0 & \text{otherwise} \end{cases}$$

And we estimate $\hat{\theta} = \arg\max_\theta P(D : \theta)$ For example when $D = (H, H, T, H)$ than $P(D) = \theta^3(1-\theta)$ examine figure 2 to see a visualization of this. In order to find the maximum we would use:

$$\frac{d}{d\theta} L(\theta : D) = 0$$

and we use the monotony of log and look at $\log(L)$ which get to its maximum correlated with $L()$. We mark:

$$N_T = \sum_n 1\{x_n = T\}$$

$$N_H = \sum_n 1\{x_n = H\}$$
Figure 1: CG presentage in the Genome

\[ l(\theta) = \log(\theta^N (1 - \theta^N)) = N_H \log \theta + N_T \log(1 - \theta) \]

\[ l'(\theta) = \frac{N_H}{\theta} - \frac{N_T}{1 - \theta} \]

\[ \theta = \frac{N_H}{N_H + N_T} \]

\( N_H \) and \( N_T \) are called statistics which is a function of \( D \), \( S(D) \rightarrow \mathbb{R}^k \) so if \( S(D) = S(D') \) than \( P(D : \theta) = p(D' : \theta) \) for every \( \theta \). If the if is iff (if and only if) than the statistic is minimal. Intuitively the function preserves statistical information of the data.

1.2 The Piecewise Model

Assumptions: we deal with step function like before, and the \( X \) axis is divided to \( k \) sections where for each section there is a different \( \theta \). Our parameter than is:

\[ \{ \hat{\theta}_i \} = \left\{ \begin{array}{l} n_i \quad \text{the region index} \\ \theta_i \quad 1 \leq i \leq k \quad \text{- the same } \theta \text{ as before, unique to each region} \end{array} \right. \]

What is the likelihood function?

\[ P(D : \theta) = \prod_{i=1}^{k} \prod_{n=n_{i-1}+1}^{n_i} P(x_n : \theta_i) = \prod_{i=1}^{k} P_{\text{uniform}}(<x_{n_{i-1}+1}, \ldots, x_{n_i} : > \theta_i) \]

Where \( P_{\text{uniform}} \) is the joined probability as we know it. If the series \( (n_i) \) is given, finding \( \hat{\theta}_i \) is explicitly dealing with \( k \) different problems, independent of each other, identical to the problem in the section above.

\[ \hat{\theta}_i = \frac{N_H(n_{i-1}, n_i)}{n_i - n_{i-1}} \]

\[ N_{H(i,j)} = \sum_{n=i+1}^{j} 1 \{ x_n = H \} \]
But our function is of the type:

\[
\arg\max_{n_1,\ldots,n_k} \left[ \max_{\theta_1,\ldots,\theta_k} l(\theta_1,\ldots,\theta_k, n_1,\ldots,n_k : D) \right]
\]

The problem with this function is the huge number of opportunities, that the \(\max\) function has to operate on. So we define a new function:

\[
f(m, d) = \max_{\theta, n_i} l(\theta_1,\ldots,\theta_d, n_1,\ldots,n_d : <x_1,\ldots,x_m>)
\]

which is the maximum of dividing over the \(m\) first letters to \(d\) sections, Yielding dynamic programing approach.

\[
f(m, d) = \max_{n_d-1} \max_{\theta_d} f(n_d-1, d-1) + \log[\theta_d^{N_H(n_d-1,m)}(1-\theta_d)^{N_T(n_d-1,m)}] = \\
= \max_{n_d-1} [f(n_d-1, d-1) + N_H(n_d-1,m) \log \frac{N_H(n_d-1,m)}{m-n_d-1} + N_T(n_d-1,m) \log \frac{N_T(n_d-1,m)}{m-n_d-1}]
\]

which is the solution of the last section. The algorithm is described in the next page.
func uniforml($N_H, N$)
    return $N_H \log \frac{N_H}{N} + (N - N_H) \log \frac{N - N_H}{N}$

initialization:
    for $m = (1, \ldots, n)$
        $f[m, 1] = \text{uniforml}(N_H(0,m),m)$ # counts number of heads from 0 to m
    end
    for $d = (2, \ldots, k)$
        for $m = (2, \ldots, N)$
            $\text{Bestm} = 0$
            $\text{Bestl} = -\infty$
            for $n = (1, \ldots, m-1)$
                $l = f[n,d-1] + \text{uniforml}(N_H(n,m),m-n)$
                if $l > \text{Bestl}$
                    $\text{Bestl} = l$
                    $\text{Bestm} = n$
                end
            end
            $f[m,d] = \text{Bestl}$
        end
    end
    $\text{fhelper}[m,d] = \text{BestM}$ # fhelper is a matrix used to store the division of the sequence, in term of position, nucleotide number
end
end
$T[k] = N$
for $i = (k-1, \ldots, 1)$
    $T[i] = \text{fhelper}[T(i+1),i+1]$ # extracting the exact positions of the best division in maximum likelihood terms
end
return $f[N,k]$

The time complexity of this algorithm is $N^2k$