EXON ASSEMBLY
Given scoring function σ with $\sigma(x,-) \leq 0$ and $\sigma(-,x) \leq 0$ for all characters x (as it is usually the case for standard scoring functions), genome string $G = G[1 \ldots n]$, list of candidate exons, i.e. substrings $E_1, \ldots, E_b$ of G, and target string $T = T[1 \ldots m]$, compute the best possible optimal alignment score for any concatenation Γ^* of a chain Γ of candidate exons with T:

$$\max_{\text{chains } \Gamma} \sigma_{\text{opt}}(\Gamma^*, T).$$

The following notions are used in the sections below. For a candidate exon $E = G[i \ldots j]$ define $\text{first}(E) = i$ and $\text{last}(E) = j$. For candidate exons E and F let $E < F$ stand for $\text{last}(E) < \text{first}(F)$.

3.5.2 Parameterization and Conditioning

As usual, parameterization is done via truncation of strings G and T. Concerning target string T, we explicitly truncate it at some position j with $0 \leq j \leq m$. Concerning genome string G, truncation is done indirectly by fixing some exon candidate E_k as the last one to be used in any chain of candidate exons, and truncating E_k at some position i with $\text{first}(E_k) \leq i \leq \text{last}(E_k)$. Note that truncation of T may lead to the empty string (in case $j = 0$), whereas truncation of E_k always gives a non-empty string (containing at least base $G(i)$). For a chain Γ of candidate exons $F_1 < F_2 < \ldots < F_p$ and index i with $\text{first}(F_p) \leq i \leq \text{last}(F_p)$, we use the suggestive notation $\Gamma^*[\ldots i]$ to denote the concatenation $F_1 F_2 \ldots F_{p-1} G[\text{first}(F_p) \ldots i]$, with the last candidate exon truncated at absolute position i.

Now the parameterized and conditioned problem can be stated as follows: for all combinations of parameters j, k, i with $0 \leq j \leq m$, $1 \leq k \leq b$, and $\text{first}(E_k) \leq i \leq \text{last}(E_k)$ compute the following value $\text{chain}(j,k,i)$:

$$\text{chain}(j,k,i) = \max_{\Gamma \text{ ending with } E_k} \sigma_{\text{opt}}(\Gamma^*[\ldots i], T[1 \ldots j]). \tag{3.4}$$

After computation of all these values, the original problem is then solved by returning

$$\max_{1 \leq k \leq b} \text{chain}(m, k, \text{last}(E_k)). \tag{3.5}$$

3.5.3 Bellman Principle

... is obviously fulfilled.

3.5.4 Recursive Solution

The solution presented here is taken from [64]. For the computation of terms $\text{chain}(j,k,i)$ for any admissible combination of parameters j,k,i we distinguish several cases:

Case 1. $j=0$
Case 2. $j>0$, there is no index r with $E_r < E_k$
Case 3. $j>0$, $\text{first}(E_k) < i$, there is at least one index r with $E_r < E_k$
Case 4. $j>0$, $\text{first}(E_k) = i$, there is at least one index r with $E_r < E_k$

In case 1, we have to optimally align a suitable string $\Gamma^*[\ldots i]$ with the empty string $T[1\ldots 0]$. Since inserts and deletes are scored zero or negative, the optimal alignment is achieved by using no other candidate exons than the prescribed last E_k. We obtain:

$$\text{chain}(0,k,i) = \sum_{p=\text{first}(E_k)}^{i} \sigma(G(p),-). \tag{3.6}$$

In case 2, there are no further candidate exons left of E_k that can be used in $\Gamma^*[\ldots i]$. Thus we obtain:

$$\text{chain}(j,k,i) = \sigma_{\text{opt}}(G[\text{first}(E_k)\ldots i], T[1\ldots j]). \tag{3.7}$$

In case 3, we look at what may happen at the right end of an optimal alignment of $\Gamma^*[\ldots i]$ and $T[1\ldots j]$. As usual, either $G(i)$ is aligned with $T(j)$, or $G(i)$ is aligned with spacing symbol – , or spacing symbol – is aligned with $T(j)$. In any case, in the recursive step we do not leave E_k as last used exon candidate. Thus we obtain:

$$\text{chain}(j,k,i) = \max \begin{cases} \text{chain}(j-1,k,i-1) + \sigma(G(i),T(j)) \\ \text{chain}(j,k,i-1) + \sigma(G(i),-) \\ \text{chain}(j-1,k,i) + \sigma(-,T(j)). \end{cases} \tag{3.8}$$

In case 4, after aligning $G(i)$ with $T(j)$ or $G(i)$ with spacing symbol – , we have consumed the last available symbol of E_k. Every candidate exon E_q with $E_q < E_k$ may be the segment used in Γ left of E_k; alternatively E_k was the only segment used in Γ. Thus we obtain:

$$\text{chain}(j,k,i) = \max \begin{cases} \sigma(G(i),T(j)) + \max\limits_{q \text{ with last}(E_q)<i} \text{chain}(j-1,q,\text{last}(E_q)) \\ \sigma(G(i),-) + \max\limits_{q \text{ with last}(E_q)<i} \text{chain}(j,q,\text{last}(E_q)) \\ \sigma(G(i),T(j)) + \sum_{p<j} \sigma(-,T(p)) \\ \sigma(G(i),-) + \sum_{p\le j} \sigma(-,T(p)) \\ \sigma(-,T(j)) + \text{chain}(j-1,k,i). \end{cases} \tag{3.9}$$