Supplementary Material for:

Algorithms for the Hard Pre-Image Problem of String Kernels and the General Problem of String Prediction

Sébastien Giguère [1,2*] GIGUERE.SEBASTIEN@GMAIL.COM
Amélie Rolland [2*] AMELIE.ROLLAND.1@ULAVAL.CA
François Laviolette [2] FRANCOIS.LAVIOLETTE@IFT.ULAVAL.CA
Mario Marchand [2] MARIO.MARCHAND@IFT.ULAVAL.CA
[1] Institute for Research in Immunology and Cancer, University of Montreal, Montreal, Canada
[2] Department of Computer Science and Software Engineering, Laval University, Quebec, Canada
[*] These authors contributed equally to this work.

1. Algorithms

The details of the branch and bound algorithm are given in Algorithm 1. The best first search is detailed in Algorithm 2.

Algorithm 1 Branch and bound for finding $\mathbf{y}^\star \in \mathcal{A}^\ell$

function branch_and_bound($\ell, n, \mathcal{A}$)
  $\mathcal{Q}$: empty priority queue ordering (string, bound) pairs in descending order of bound values
  $best \leftarrow$ *Node*(*empty_string*, 0)
  for all $s \in \mathcal{A}^n$ **do**
    $\mathcal{Q}$.push(*Node*($s, F(s,\ell)$))
  end for
  while $node \leftarrow \mathcal{Q}$.pop() & $node.bound > best.bound$ **do**
    $node \leftarrow$ bf_search($node, best, \mathcal{Q}, \ell, \mathcal{A}$)
    if $node.bound > best.bound$ **then**
      $best \leftarrow node$
    end if
  end while
  return $best.string, best.bound$
end function

Algorithm 2 Best first search

function bf_search($node, best, \mathcal{Q}, \ell, \mathcal{A}$)
  while $|node.string| < \ell$ & $node.bound > best.bound$ **do**
    $best\_child \leftarrow$ *Node*(*empty-string*, 0)
    for all $a \in \mathcal{A}$ **do**
      $s' \leftarrow$ concatenate($a, node.string$)
      if $F(s',\ell) > best.bound$ **then**
        if $F(s',\ell) > best\_child.bound$ **then**
          $best\_child \leftarrow$ *Node*($s', F(s',\ell)$)
        end if
        $\mathcal{Q}$.push(*Node*($s', F(s',\ell)$))
      end if
    end for
    $node \leftarrow best\_child$
    $\mathcal{Q}$.remove($best\_child$)
  end while
  return $node$
end function

2. Bounds

2.1. Details of f when $\sigma_p > 0$ and $\sigma_c = 0$

Let us recall that f must lower bound $K_{\mathcal{Y}}(\mathbf{y}, \mathbf{y})$, given that string $\mathbf{y}$ has $\mathbf{y}'$ as suffix:

$$f(\mathbf{y}', \ell) \leq \min_{\mathbf{y} \in \mathcal{A}^{\ell-p} \times \{\mathbf{y}'\}} K_{\mathcal{Y}}(\mathbf{y}, \mathbf{y}) \,. \tag{1}$$

The main idea behind the bound is that each l-gram in string $\mathbf{y}$ is indexed by a starting position that varies between 0 and $\ell - l$, where $\ell = |\mathbf{y}|$. The kernel value is obtained by summing the comparison of all l-gram pairs. To get the bound, we decide to split the comparison of the l-gram pairs in three groups that depend on their starting positions. Each group is bounded independently. The final bound is defined as

$$\begin{aligned} f(\mathbf{y}', \ell) &\stackrel{\text{def}}{=} GS(\mathbf{y}', \mathbf{y}', n, \sigma_p, \sigma_c) + 2YY'(\mathbf{y}', \ell, n, \sigma_p, \sigma_c) \\ &\quad + YY(\ell - |\mathbf{y}'|, n, \sigma_p, \sigma_c) \,. \end{aligned} \tag{2}$$

The first group of l-grams are those in position $\ell - |\mathbf{y}'|$ to $\ell - l$. All these l-grams belong to the suffix $\mathbf{y}'$ and are compared with themselves using the GS kernel function. This part of the bound is exact.

The second group compares those in positions 0 to $\ell - |\mathbf{y}'| - 1$ with those of the suffix $\mathbf{y}'$ in position $\ell - |\mathbf{y}'|$ to $\ell - l$. A lower bound on the comparison of these l-gram pairs is given by the function

$$\begin{aligned} & YY'(\mathbf{y}', \ell, n, \sigma_p, \sigma_c) \\ &= \sum_{l=1}^{n} \sum_{i=0}^{\ell-|\mathbf{y}'|-1} \min_{\mathbf{y} \in \mathcal{A}^l} \sum_{j=0}^{|\mathbf{y}'|-l} \exp\left(\tfrac{-(i-(j+\ell-|\mathbf{y}'|))^2}{2\sigma_p^2}\right) \\ &\qquad \times I(y_1, .., y_l = y'_{j+1}, .., y'_{j+l}) \,. \end{aligned} \tag{3}$$

This function effectively lower bounds the contribution of l-grams in positions 0 to $\ell - |\mathbf{y}'| - 1$ when compared to the suffix $\mathbf{y}'$ as it always selects the l-gram in $\mathcal{A}^l$ minimising the contribution. Observe that the index j refers to positions in $\mathbf{y}'$. For that reason, in the position penalty term $\exp\left(\frac{-(i-j)^2}{2\sigma_p^2}\right)$, j was offset by $\ell - |\mathbf{y}'|$ to correspond to positions in $\mathbf{y}$.

The last group compares those in positions 0 to $\ell - |\mathbf{y}'| - 1$ with themselves. A lower bound on the comparison of these l-gram pairs is given by the function

$$YY(d, n, \sigma_p, \sigma_c) = \sum_{l=1}^{n} \sum_{i=0}^{d-1} \sum_{j=0}^{d-1} S(i, j, l, d, \sigma_p, \sigma_c), \tag{4}$$

where

$$\begin{aligned} & S(i, j, l, d, \sigma_p, \sigma_c) \\ &= \begin{cases} 1 & \text{if } i = j, \\ exp\left(\dfrac{-d^2}{2\sigma_p^2}\right) & \text{if } |i-j| \in \{|\mathcal{A}|^l, 2|\mathcal{A}|^l, 3|\mathcal{A}|^l, ...\}, \\ 0 & \text{otherwise,} \end{cases} \end{aligned}$$

which accounts for the minimum number of l-gram repetitions at largest distance.

Finally, since all l-gram pairs were considered in one of the three functions, this makes $f(\mathbf{y}', \ell)$ a valid lower bound.

2.2. Details of f when $\sigma_p > 0$ and $\sigma_c > 0$

The strategy to obtain a bound in this case is the same as in the previous case: the l-gram pairs are divided in the same groups but the functions YY' and YY that lower bound their contributions are modified to take into account ψ^l.

For YY', the identity function between l-grams is replaced by the second exponential term of the GS kernel. Hence,

$$\begin{aligned} & YY'(\mathbf{y}', \ell, n, \sigma_p, \sigma_c) \\ &= \sum_{l=1}^{n} \sum_{i=0}^{\ell-|\mathbf{y}'|-1} \min_{\mathbf{y} \in \mathcal{A}^l} \sum_{j=0}^{|\mathbf{y}'|-l} \exp\left(\tfrac{-(i-j+|\mathbf{y}'|-\ell)^2}{2\sigma_p^2}\right) \\ &\qquad \times \exp\left(\frac{-\|\boldsymbol{\psi}^l(y_1,..,y_l) - \boldsymbol{\psi}^l(y'_{j+1},..,y'_{j+l})\|^2}{2\sigma_c^2}\right) . \end{aligned} \tag{5}$$

For YY, only the function S needs to be redefined as

$$S(i, j, l, d, \sigma_p, \sigma_c) = \exp\left(\tfrac{-(i-j)^2}{2\sigma_p^2}\right) \exp\left(\tfrac{-l(D(i,j))}{2\sigma_c^2}\right)$$

where

$$D(i, j) = \begin{cases} 0 & \text{if } i = j, \\ \max\limits_{(a,a') \in \mathcal{A}^2} ||\boldsymbol{\psi}(a) - \boldsymbol{\psi}(a')||^2 & \text{otherwise.} \end{cases}$$

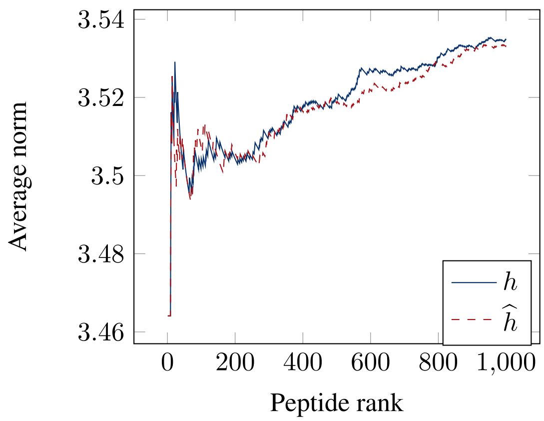

Figure 1. The cumulative moving average of norm $||\boldsymbol{\phi}_{\mathcal{Y}}(\mathbf{y})||$ for the $1,000$ peptides having the highest predicted bioactivities for the BPPs dataset.

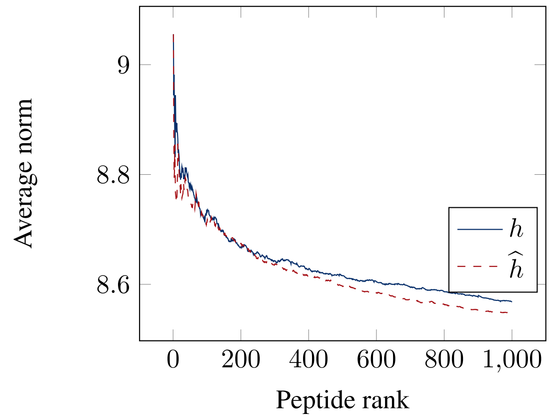

Figure 2. The cumulative moving average of norm $||\boldsymbol{\phi}_{\mathcal{Y}}(\mathbf{y})||$ for the $1,000$ peptides having the highest predicted bioactivities for the CAMPs dataset.

3. Additional Figures

Figure 1 and Figure 2 present additional results for the cumulative moving average of the norm of the 1000 top peptides maximizing the un-normalized predictor h and the normalized predictor $\widehat{h}$. For both BPPs and CAMPs datasets, these results show that the norms of the peptides maximizing h and $\widehat{h}$ are similar. This was expected since the chosen σ_p values are small ($\sigma_p = 0.2$ for BPPs and $\sigma_p = 0.8$ for CAMPs).

Figure 3 and Figure 4 compare the ability of each method to handle strings of different lengths when $\sigma_p = \infty$. On both datasets, the peptide with the highest bioactivity for the un-normalized predictor $h_{\sigma_p=\infty}$ has the longest length. For the normalized predictor $\widehat{h}_{\sigma_p=\infty}$, the best length is 7 for the BPPs dataset and 20 for the CAMPs dataset. As explained previously, the longer a string $\mathbf{y}$ is, the larger $||\boldsymbol{\phi}_{\mathcal{Y}}(\mathbf{y})||$ generally is, which effectively influences $h(\mathbf{y})$. Consequently, if the length of the output string is not constrained, $\mathbf{y}^h$ will be biased toward long strings, especially when the value of σ_p is large.

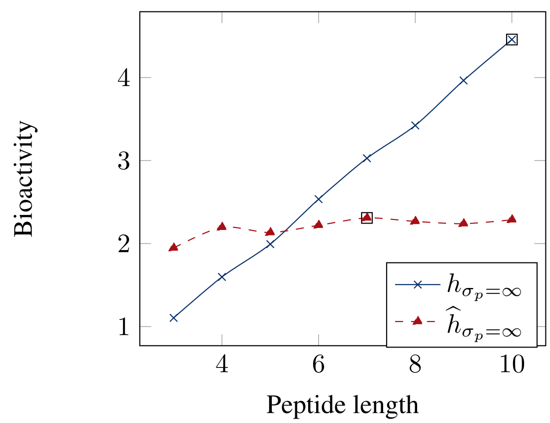

Figure 3. Maximum predicted bioactivity as a function of the peptide length for the BPPs dataset.

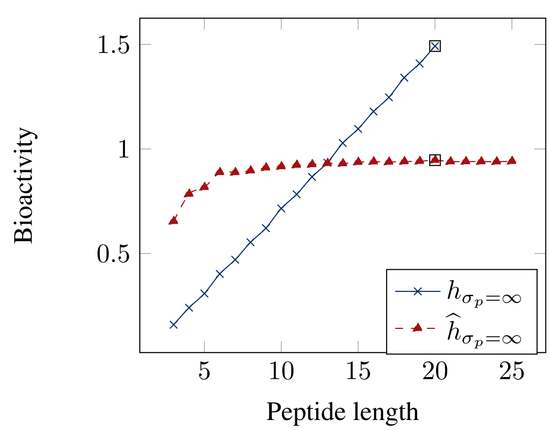

Figure 4. Maximum predicted bioactivity as a function of the peptide length for the CAMPs dataset.