Supplemental Material

Amir Joudaki[1,2,*], Alexandru Meterez[1,*], Harun Mustafa[1,2,3],
Ragnar Groot Koerkamp[1], André Kahles[1,2,3,†], and Gunnar Rätsch[1,2,3,4,†]

[1] Department of Computer Science, ETH Zurich, Zurich 8092, Switzerland
[2] University Hospital Zurich, Biomedical Informatics Research, Zurich 8091, Switzerland
[3] Swiss Institute of Bioinformatics, Lausanne 1015, Switzerland
[4] ETH AI Center, 8092 Zurich, Switzerland
`{firstname}.{lastname}@inf.ethz.ch`
[*] Equal contribution.
[†] To whom correspondence should be addressed.

A Formal statement and proof of Example 1

In this section, we provide a slightly more general and more formal proof for example 1. We are given a reference sequence S consisting of N uniformly sampled elements from $\Sigma := \{A, C, G, T\}^N$, and a mutation rate $0 \leq r \leq 1$. Given the arbitrary offset $\Delta \in \{0, \ldots, N - |q|\}$, we create a query q by copying a substring of the reference sequence and substituting each position independently with probability r:

$$\text{for } j = 1 \text{ up to } |q| : \begin{cases} q_j \leftarrow S_{\Delta+j} & \text{copy with } 1-r \text{ probability} \\ q_j \text{ draw uniformly } \Sigma \setminus \{[S_{\Delta+j}]\} & \text{substitute with } r \text{ probability.} \end{cases}$$

We define a hit as a pair of indices that correspond to matching k-mers, and a true positive hit as pairs that are aligned to the offset:

$$\text{Hits} := \{(i,j) \in [N-k+1] \times [|q|-k+1] : S_{i\ldots i+k-1} = q_{j\ldots j+k-1}\} \quad \text{TP-Hits} := \{(i,j) \in \text{Hits} : i - j = \Delta\}$$

We define recall as the event that there is at least one true positive $\text{Recall} := \Pr(|\text{TP-Hits}| > 0)$.

Expected number of hits

We first consider the expected number of hits due to chance. Let $\mathcal{E}_{i,j}$ denote the event where the k-mers at indices i and j in the reference and query, respectively, are a match, i.e., $\mathcal{E}_{i,j} = \mathbb{1}[(i,j) \in \text{Hits}]$, where $\mathbb{1}$ is the indicator variable. For values of i that are not in the range $[\Delta - k, \Delta + |q| + k]$, the probability of a hit takes the form $|\Sigma|^{-k}$, corresponding to every position matching randomly. Observe that $\mathcal{E}$ represents the number of hits is the sum of these events $|\text{Hits}| = \sum_{i \leq N-k+1, j \leq |q|-k+1} \mathcal{E}_{i,j}$, and based on linearity of expectation and assuming N and $|q|$ are greater than k, we have $\mathbb{E}|\text{Hits}| = \Theta\left(|q|N|\Sigma|^{-k}\right)$.

Recall

In order to have recall, at least one of the k-mers in the query must remain un-mutated. Let $\mathcal{E}_i$ denote the event that the k-mer starting at position i in the query sequence, $q_{i,\ldots,i+k-1}$, is not mutated. We are only interested in $i \in \{1, \ldots, |q|-k+1\}$ because indices $i \geq |q|-k$ do not correspond to a valid k-mer. We can use the union bound to upper bound the probability that any of these events occur: $\text{Recall} = \Pr(\bigvee_{i=1}^{|q|-k+1} \mathcal{E}_i) \leq (|q|-k+1)(1-r)^k$. This follows from the fact that the probability of any individual event is $(1-r)^k$.

B Commands used for experiments

We use an assembled version of the base graph to evaluate GraphAligner (GA). For the vg methods, we transform the base graph into a variant graph.

To use the same graph in VG MAP and VG MPMAP we “bluntify” it with the GETBLUNTED tool to remove overlaps between nodes (Eizenga et al. 2021). Finally, we call `vg autoindex` on the blunted graph to obtain a GCSA index used by VG.

We evaluate GA using unlimited `tangle effort` by enabling the parameter `-C -1`. While in this setting GA uses 13GB of RAM, disabling this parameter resulted in an imperfect recall even at 0% mutation. This is due to the heuristic used in tangled graph regions, which can drop the correct path in case the extender starts exploring a false positive path first. This is a common case in our evaluation since our graphs have many similar paths.

We initially generate the De Bruijn graph in METAGRAPH. The graph is then assembled into unitigs and blunted using `get_blunted` to accommodate for VG’s format. This removes the overlap between the nodes. The GCSA index required by VG is generated using `vg autoindex`. In practice we ran the following set of commands:

```
metagraph build -k 80 --parallel 20 -o graph.dbg sequence.fa
metagraph assemble --to-gfa --compacted --unitigs -o graph.gfa graph.dbg
get_blunted --input_gfa graph.gfa > blunted_graph.gfa
vg autoindex -g blunted_graph.gfa -V 2 -w map
vg convert index.xg -p > index.vg
```

To evaluate each baseline we use the following set of commands:

GraphAligner

```
GraphAligner -g blunted_graph.gfa -f input.fa
-a output.gaf -x dbg -C -1
```

MetaGraph Sketching

```
metagraph align --seeder sketch --embed-dim 14
--num-neighbours 10 --align-end-bonus 0 -i graph.dbg in.fa
```

MetaGraph Exact

```
metagraph align --seeder default --align-min-seed-length 15
--align-xdrop 15 -i graph.dbg input.fa
```

vg map

```
vg map -z 1 -o 1 -x index.xg -g index.gcsa -f input.fa --gaf
```

vg mpmap

```
vg mpmap -n DNA -F GAF -z 1 -o 1 -x index.xg -g index.gcsa
-f input.fa --gaf
```

To measure peak memory usage, we used the `/usr/bin/time -f %M` command. To evaluate VG MAP and VG MPMAP, we extracted the path spelling of the GAF output using `vg find` and `vg view`. To evaluate GRAPHALIGNER, we extracted the obtained path spellings from the input GFA file.

C Proof of Tensor Embedding bound

Notation The alphabet is Σ, of size $|\Sigma| = \sigma$. The set of indices is $\mathcal{I}^t := \{(i_1, \ldots, i_t) \in [n]^t : i_1 < \cdots < i_t\}$. Given a string $a_1 \ldots a_n = a \in \Sigma^n$, we define the I-index as $a_I = (a_{i_1}, \ldots, a_{i_t})$. We write $[X]$ for the indicator variable of event X, which is 1 when X holds and 0 otherwise.

Definition 1: Tensor Embedding Given $a \in \Sigma^n$, the Tensor Embedding T_a is the σ^t tensor given by $T_a[s] = \sum_{I\in\mathcal{I}^t}[A_I = s]$ for each $s \in \Sigma^t$.

The normalized Tensor Embedding distance d_{te} between two sequences a and b is defined as

$$d_{te}(a,b) := \frac{1}{2}\binom{n}{2t-1}^{-1}\|T_a - T_b\|_2^2. \tag{1}$$

Lemma 1: Tensor embedding preserves Hamming distance under ℓ^2 norm Let a be a uniform random sequence of length n in Σ^n, and for a fixed mutation rate $r \in [0,1]$ let b be a sequence where a_i is substituted by a new character $b_i \in Unif(\Sigma \backslash a_i)$ with probability r and $b_i = a_i$ otherwise. Then:

$$\mathbb{E}_{a,b}[d_{te}(a,b)] = \Big(1 + O(2t\sigma/4^{t-1}n)\Big)\cdot(4/\sigma)^{t-1}\cdot r, \tag{2}$$

which for DNA with $\sigma = 4$ gives $\mathbb{E}[d_{te}(a,b)] = (1 + O(n^{-1}))\cdot r$.

Proof of Lemma 1 By definition we have

$$2\binom{n}{2t-1}d_{te}(a,b) = \|T_a - T_b\|_2^2 = \sum_{s\in\Sigma^t}\left(\sum_{I\in\mathcal{I}^t}[a_I = s] - \sum_{I\in\mathcal{I}^t}[b_I = s]\right)^2 \tag{3}$$

$$= \sum_{s\in\Sigma^t}\sum_{I,J\in\mathcal{I}^t}\Big([a_I = s][a_J = s] - [a_I = s][b_J = s] - [b_I = s][a_J = s] + [b_I = s][b_J = s]\Big). \tag{4}$$

By symmetry between a and b, the first and last term, and second and third term are equal in expected value, reducing this to

$$\mathbb{E}_{a,b}\left(\|T_a - T_b\|_2^2\right) = \mathbb{E}\left(2\sum_{s\in\Sigma^t}\sum_{I,J\in\mathcal{I}^t}\Big([a_I = s][a_J = s] - [a_I = s][b_J = s]\Big)\right) \tag{5}$$

$$= \mathbb{E}\left(2\sum_{I,J\in\mathcal{I}^t}\sum_{s\in\Sigma^t}\Big([a_I = s \wedge a_J = s] - [a_I = s \wedge b_J = s]\Big)\right) \tag{6}$$

$$= 2\sum_{I,J\in\mathcal{I}^t}\mathbb{E}\Big([a_I = a_J] - [a_I = b_J]\Big). \tag{i}$$

Define the *overlap* q as the number of positions where I and J are equal, $q(I,J) := |\{x \in [t] : I_x = J_x\}|$. We will show using induction on t that $\mathbb{E}[a_I = b_J] = (\sigma(1-r))^q\sigma^{-t}$. For $t = 0$ we have $I = J = \varnothing$ and trivially $\mathbb{E}[a_I = b_J] = 1$. For $t > 0$, write I' and J' for the tuples $(I_1, \dots, I_{t-1})$ and $(J_1, \dots, J_{t-1})$. When $I_t = J_t$, the characters a_{I_t} and b_{J_t} are independent of the earlier characters and equal with probability $1-r$, and $q(I', J') = q - 1$, so that

$$\mathbb{E}[a_I = b_J] = (1-r)\mathbb{E}[a_{I'} = b_{J'}] \tag{7}$$

$$= (1-r)\cdot(\sigma(1-r))^{q-1}\sigma^{-(t-1)} \tag{8}$$

$$= (\sigma(1-r))^q\sigma^{-t}. \tag{9}$$

When $I_t \neq J_t$, assume without loss of generality that $I_t < J_t$. Then $I_x < J_t$ for all $x \in [t]$, resulting in b_{J_t} is independent from the characters seen so far. This implies that $[a_{I_t} = b_{J_t}]$ is independent from $[a_{I'} = b_{I'}]$.

$$\mathbb{E}[a_I = b_J] = \mathbb{E}[a_{I_t} = b_{J_t}]\mathbb{E}[a_{I'} = b_{J'}] \tag{10}$$

$$= (\sigma)^{-1}\cdot(\sigma(1-r))^q\sigma^{-(t-1)} \tag{11}$$

$$= (\sigma(1-r))^q\sigma^{-t}. \tag{12}$$

We conclude that

$$\mathbb{E}_{a,b}\big([a_I = a_J] - [a_I = b_J]\big) = \sigma^{-t+q}\big(1-(1-r)^{q(I,J)}\big). \tag{13}$$

This difference vanishes for $q = 0$, and thus in (i) we only have to consider (I,J) with $q(I,J) \geq 1$. The summation can now be rewritten as

$$\begin{aligned}
\mathbb{E}_{a,b}\left(\|T_a - T_b\|_2^2\right) &= 2\sum_{q=1}^{t} \sum_{\substack{I,J\in\mathcal{I}^t:\\ q(I,J)=q}} \mathbb{E}\Big([a_I = a_J] - [a_I = b_J]\Big) && (14)\\
&= 2\sum_{q=1}^{t} \sum_{\substack{I,J\in\mathcal{I}^t:\\ q(I,J)=q}} \sigma^{-t+q}\big(1-(1-r)^q\big) && (15)\\
&= 2\sum_{q=1}^{t} \sigma^{-t+q}\big(1-(1-r)^q\big)\cdot f_q, && \text{(ii)}
\end{aligned}$$

where f_q counts the number of pairs (I,J) with $q(I,J) = q$. Since $|I\cap J| \geq q$, the total number of distinct indices is bounded by $|I\cup J| \leq 2t-q$. This directly implies that $f_q \leq \binom{n}{2t-q}$, which for $q \geq 2$ gives

$$\binom{n}{2t-1}^{-1}\binom{n}{2t-q}\cdot\sigma^{-t+q}\big(1-(1-r)^q\big) = O((2t\sigma/n)^{q-1})\cdot\sigma^{1-t}r. \tag{16}$$

When $q = 1$ but $|I\cup J| < 2t-1$ a similar argument applies, and we are left with the case where $q = 1$ and $|I\cup J| = 2t-1$. We can first choose the $2t-1$ distinct values for $I\cup J$ in $\binom{n}{2t-1}$ ways, and then assume that $I\cup J = [2t-1]$. The overlap can be at any odd position $2k+1 \in \{1,3,\dots,2t-1\}$, since I and J must both have an equal number of distinct elements smaller (resp. larger) than $2k+1$. Given $2k+1$, the $2k$ smaller positions can be split into two halves in $\binom{2k}{k}$ ways, and similarly for the right half, leading to the following number of (I,J) pairs with $q = 1$ and $|I\cup J| = 2t-1$:

$$\binom{n}{2t-1}\cdot\sum_{k=0}^{t-1}\binom{2k}{k}\binom{2(t-1-k)}{t-1-k} = \binom{n}{2t-1}\cdot 4^{t-1}, \tag{17}$$

a well-known equality (Lyons 1998). Finally, splitting (ii) into the cases $q = 1$ (with $|I\cap J| = 1$ and $|I\cap J| > 1$) and $q > 2$, and assuming that $n \gg 2t\sigma$, we get:

$$\begin{aligned}
\mathbb{E}(d_{te}(a,b)) &= (4/\sigma)^{t-1}\cdot r + O(2t\sigma/n)\cdot\sigma^{-t}r + \sum_{q=2}^{t} O((2t\sigma/n)^{q-1})\cdot\sigma^{1-t}r && (18)\\
&= (4/\sigma)^{t-1}\cdot r + O(2t\sigma/n)\cdot\sigma^{1-t}r && (19)\\
&= \Big(1 + O(2t\sigma/4^{t-1}n)\Big)\cdot(4/\sigma)^{t-1}\cdot r && (20)
\end{aligned}$$

■

D Implementation details for MG-SKETCH

We provide an overview of the implementation of TENSOR SLIDE SKETCHING.

Algorithm 2: TensorSlideSketch ϕ_{TSS}

```
Input      : Query $q \in \Sigma^k$
Output     : Sketch $\Phi \in \mathbb{R}^{\lceil \frac{k-w+1}{s} \rceil D}$
Parameter: Dimension $D \in \mathbb{N}^+$
             Tuple size $t \in \mathbb{N}^+$
             Stride $s \in \mathbb{N}^+$
             Window size $w \in \mathbb{N}^+$
for $j \leftarrow 0$ to $\lceil \frac{k-w+1}{s} \rceil$ do
    $\Phi_j \leftarrow$ TensorSketch$\big(q[js : js + w - 1]\big)$
end
Return     : $\Phi$
```