

Traversing the k -mer Landscape of NGS Read Datasets for Quality Score Sparsification

Jacopo Notarstefano
`jacopo.notarstefano [at] gmail.com`

May 19, 2014

Next-generation sequencing (NGS)

The problem with next-generation sequencing

"In the past two decades, genomic sequencing capabilities have increased exponentially, outstripping advances in computing power and storage."

- Moore's law predicts that the number of transistors on integrated circuits doubles every 24 months.
- Kryder's law predicts that hard drive density doubles approximately every 13 months.
- Sequencing output has doubled every 9 months.

How to deal with storage

	foo	bar
foo	bar	baz
foo	bar	baz

The FASTQ format

Phred quality score

Definition (Phred quality score)

Let P_e be the estimated probability that a base call is incorrect. Then we define its *Phred quality score* Q as

$$Q = -10 \log_{10} P_e.$$

This quantity is encoded in the Sanger FASTQ format as a single byte, where the character '!' represents the lowest quality while '~' is the highest.

The RQS algorithm, in brief

The DICT algorithm

Algorithm 1 DICT

Input: C, k, r

Output: D

```
1:  $D \leftarrow \{\}$ 
2:  $A \leftarrow [0, \dots, 0] \in \mathbb{N}^{4^k}$ 
3: for  $x \in C_k$  do
4:    $A[x]++$ 
5: for  $x \in [4^k]$  do
6:   if  $A[x] \geq r$  then
7:      $D.append(x)$ 
8: return  $D$ 
```

The MARKMER algorithm

Algorithm 2 MARKMER

Input: x, D

Output: M

```
1: if  $\Delta(x, D) > 1$  then
2:    $M \leftarrow [\text{false}, \dots, \text{false}] \in \{\text{true}, \text{false}\}^k$ 
3: else
4:    $M \leftarrow [\text{true}, \dots, \text{true}] \in \{\text{true}, \text{false}\}^k$ 
5:   for  $y \in D \mid \Delta(x, y) = 1$  do
6:     for  $i \in [k]$  do
7:       if  $x_i \neq y_i$  then
8:          $M_i \leftarrow \text{false}$ 
9: return  $M$ 
```

The MARKREAD algorithm

Algorithm 3 MARKREAD

Input: γ, D

Output: \mathcal{M}

- 1: // Let x^a be the k -mer in γ starting at a .
 - 2: // Cover γ by k -mers $\{x^{a_1}, \dots, x^{a_n}\}$.
 - 3: **for** $i \in [n]$ **do**
 - 4: $M^i \leftarrow \text{MARKKMER}(x^{a_i}, D)$
 - 5: $\overline{M}^i \leftarrow [\text{false}, \dots, \text{false}] \in \{\text{true}, \text{false}\}^{\text{length}(\gamma)}$
 - 6: **for** $j \in [k]$ **do**
 - 7: $\overline{M}_{j+a_i-1}^i \leftarrow M_j^i$
 - 8: $\mathcal{M} \leftarrow \overline{M}^1 \vee \dots \vee \overline{M}^n$
 - 9: **return** \mathcal{M}
-

The SPARSIFYRQ algorithm

Algorithm 4 SPARSIFYRQ

Input: $\gamma, Q, D, Q_{\text{threshold}}$

Output: Q'

- 1: $Q' \leftarrow Q$
 - 2: $\mathcal{M} \leftarrow \text{MARKREAD}(\gamma, D)$
 - 3: **for** $i \in [\text{length}(\gamma)]$ **do**
 - 4: **if** $(Q_i > Q_{\text{threshold}})$ **or** $(\mathcal{M}_i = \text{true})$ **then**
 - 5: $Q'_i \leftarrow Q_{\text{threshold}}$
 - 6: **return** Q'
-

Results