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 [Moo65].
- Kryder's law predicts that hard drive density doubles approximately every 13 months [Wal05].
- Sequencing output has doubled every 9 months [Kah11].

How to deal with storage

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

```
D ← {}
A ← [0, ..., 0] ∈ ℕ^(4^k)
for x ∈ C_k do
    A[x]++
for x ∈ [4^k] do
    if A[x] ≥ r then
        D.append(x)
return D
```

The MarkKmer algorithm

Algorithm 2 MarkKmer

Input: x, D

Output: M

if $\Delta(x, D) > 1$ **then**
    $M \leftarrow [\text{false}, \ldots, \text{false}] \in \{\text{true}, \text{false}\}^k$
else
    $M \leftarrow [\text{true}, \ldots, \text{true}] \in \{\text{true}, \text{false}\}^k$
    for $y \in D \mid \Delta(x, y) = 1$ **do**
        for $i \in [k]$ **do**
            if $x_i \neq y_i$ **then**
                $M_i \leftarrow \text{false}$
return M

The MarkRead algorithm

Algorithm 3 MarkRead

Input: γ, D

Output: $\mathcal{M}$

// Let x^a be the k-mer in γ starting at a.
// Cover γ by k-mers $\{x^{a_1}, \ldots, x^{a_n}\}$.
for $i \in [n]$ **do**
    $M^i \leftarrow \textsc{MarkKmer}(x^{a_i}, D)$
    $\overline{M}^i \leftarrow [\text{false}, \ldots, \text{false}] \in \{\text{true}, \text{false}\}^{\text{length}(\gamma)}$
    for $j \in [k]$ **do**
        $\overline{M}^i_{j+a_i-1} \leftarrow M^i_j$
$\mathcal{M} \leftarrow \overline{M}^1 \vee \cdots \vee \overline{M}^n$
return $\mathcal{M}$

The SparsifyRQ algorithm

Algorithm 4 SparsifyRQ

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

Output: Q'

$\mathsf{Q}' \leftarrow \mathsf{Q}$

$\mathcal{M} \leftarrow \textsc{MarkRead}(\gamma, \mathsf{D})$

for $\mathsf{i} \in [\text{length}(\gamma)]$ **do**

  if $(\mathsf{Q}_{\mathsf{i}} > \mathsf{Q}_{\text{threshold}})$ **or** $(\mathcal{M}_{\mathsf{i}} = \text{true})$ **then**

    $\mathsf{Q}'_{\mathsf{i}} \leftarrow \mathsf{Q}_{\text{threshold}}$

return Q'

An example

Results

The implementation of DICT

An alternative implementation

Bibliography

- Scott D. Kahn, *On the future of genomic data*, Science **331** (2011), 728–729.
- Gordon E. Moore, *Cramming more components onto integrated circuits*, Electronics (1965), 114–117.
- Chip Walter, *Kryder's law*, Scientific American (2005).