Algorithm 1 Quality Score Line Partition Method

Input:
- File \( Q \) with \( N \) quality score lines, \( Q^N \);
- Predefined number parameter of sampling quality score lines of input file \( Q \), \( M \) (Default: 10);
- Predefined size parameter of substrings of quality scores within one quality score line, \( k \);
- Predefined partition threshold, \( \alpha \); // Data are partitioned based on this threshold

Output:
- Occurrence frequency table of substrings of length \( k \), \( F \); // \( F(q) \) means the frequency of string \( q \)
- Weight table of each entry in Frequency table \( F \), \( S \); // \( S(q) \) means the weight of string \( q \)
- Output quality score stream 0, \( C_0 \);
- Output quality score stream 1, \( C_1 \);

1: // Note*: All indexes shown in this algorithm start from 1, not 0.

2: Phase 1: Collect statistical information among the first \( M \) sampling quality score lines.

3: \( \text{Count}_F \leftarrow 0 \) // Record the total occurrence frequency of all k-mers
4: for \( i = 1 \) to \( M \) do
5: \( \text{len} = \text{length}(Q_i) \) // \( Q_i \) means the \( i \)th line in \( Q^N \). \( \text{length}(Q_i) \) means the length of \( Q_i \).
6: for \( j = 1 \) to \( \text{len} - k + 1 \) do
7: if \( Q_i[j\!+:+k\!-1] \) not in \( F \) // \( Q_i[m,n] \) means the substring of \( Q_i \) from index \( m \) to \( n \) then
8: \( F[Q_i[j\!+:+k\!-1]] \leftarrow 1 \)
9: else
10: \( F[Q_i[j\!+:+k\!-1]] \leftarrow F[Q_i[j\!+:+k\!-1]] + 1 \)
11: end if
12: \( \text{Count}_F \leftarrow \text{Count}_F + 1 \)
13: end for
14: end for

15: Phase 2: Weight assignment for each k-mer among the first \( M \) sampling quality score lines.

16: for \( q \) in \( F \) do
17: \( S(q) = F(q)/\text{Count}_F \);
18: end for

19: Phase 3: Obtain Max Weight among sampling lines in store for subsequent normalization.

20: \( \text{Max}_\text{Weight} \leftarrow 0 \) // \( \text{Max}_\text{Weight} \) is used to normalize the weight for each line in \( Q^N \).
21: for \( i = 1 \) to \( M \) do
22: \( \text{Line}_\text{Weight} \leftarrow 0 \) // \( \text{Line}_\text{Weight} \) is used to record the weight for each line in \( Q^N \).
23: \( \text{len} = \text{length}(Q_i) \)
24: for \( j = 1 \) to \( \text{len} - k + 1 \) do
25: \( \text{Line}_\text{Weight} \leftarrow \text{Line}_\text{Weight} + S(Q_i[j\!+:+k\!-1]) \)
26: end for
27: \( \text{Line}_\text{Weight} \leftarrow \text{Line}_\text{Weight}/(\text{len} - k + 1) \)
28: if \( \text{Max}_\text{Weight} \leq \text{Line}_\text{Weight} \) then
29: \( \text{Max}_\text{Weight} \leftarrow \text{Line}_\text{Weight} \)
30: end if
31: end for

32: Phase 4: Calculate the total weight for each quality score line in \( Q^N \)

33: // And partition them into two streams based on predefined threshold \( \alpha \).
34: for \( i = 1 \) to \( N \) do
35: \( \text{Line}_\text{Weight} \leftarrow 0 \)
36: \( \text{len} = \text{length}(Q_i) \)
37: for \( j = 1 \) to \( \text{len} - k + 1 \) do
38: \( \text{Line}_\text{Weight} \leftarrow \text{Line}_\text{Weight} + S(Q_i[j\!+:+k\!-1]) \)
39: end for
40: \( \text{Line}_\text{Weight} \leftarrow \text{Line}_\text{Weight}/(\text{len} - k + 1) \)
41: if \( \text{Line}_\text{Weight}/\text{Max}_\text{Weight} \geq \alpha \) then
42: \( C_0 \leftarrow Q_i \)
43: else
44: \( C_1 \leftarrow Q_i \)
45: end if
46: end for