# GENOME-SCALE ALGORITHM DESIGN 

by Veli Mäkinen, Djamal Belazzougui, Fabio Cunial and Alexandru I. Tomescu
Cambridge University Press, 2nd edition, 2023
www.genome-scale.info

## Exercises for Chapter 17. Metagenomics

17.1 Show that Problem 17.1 can be reduced to a minimum-cost network flow problem, by extending the reduction constructed in Section 5.3.3 for Problem 5.7.
17.2 Show that you can reduce Problem 17.1 to one in which every read vertex in $A$ has at least two incident edges.
17.3 Discuss how to modify the statement of Problem 17.1 if the probability of sequencing a read from a certain location inside each genome is no longer uniform for all locations, but follows a different distribution function, which is known beforehand.
17.4 Explain what changes you need to make to the reduction constructed in Exercise 17.1 if the objective function in Problem 17.1 is

$$
\begin{equation*}
(1-\alpha) \sum_{(x, y) \in M} c(x, y)+\alpha \sum_{i=1}^{m} \sum_{j=1}^{s_{i}}\left(\frac{c_{i}}{s_{i}}-d_{M}\left(y_{i, j}\right)\right)^{2} . \tag{1}
\end{equation*}
$$

Hint. Recall Exercise 5.16.
17.5 What can you say about the complexity of Problem 17.2 for $\alpha=0$ ? What about $\alpha=1$ ?
17.6 Describe an algorithm to compute the BWT of the reads that are kept and of the reads that are filtered out in Section 17.2.1, by reusing the BWT of the original file. Recall that reads in the input file $R$ are separated by distinct binary strings on the artificial characters $\#_{0}$ and $\#_{1}$ which do not belong to [1...].
17.7 Given the concatenation $R$ of all reads in a metagenomic sample as defined in Section 17.2, consider the interval of a $k_{2}$-mer $W$ in $\mathrm{BWT}_{R}$. Section 17.2.2 claims that, during the creation of pre-clusters, we can discard any $k_{2}$-mer $V$ whose interval in $\mathrm{BWT}_{R}$ coincides with the interval of $a W$ for some $a \in[1 . . \sigma]$. Prove this claim, and describe an algorithm that implements it in linear time using just one scan of the bitvector intervals.
17.8 With reference to Section 17.2.2, describe an algorithm to create pre-clusters in which two reads are merged if they share at least two distinct $k_{2}$-mers. Assume that you have $|R| \log r$ bits of main memory in addition to the space required by Lemma 17.5, and assume that you have an algorithm that sorts tuples in external memory as a black box. Describe how to implement additional conditions on top of your algorithm, for example that the two shared $k_{2}$-mers are in the same order in both reads, that they do not overlap, or that they are offset by exactly one position.
17.9 With reference to Section 17.2.2, describe an algorithm to create pre-clusters in which two reads are merged if they share a $k_{2}$-mer or its reverse complement. Note that in this case we cannot discard $k_{2}$-mers that occur exactly once in $R$, since they
could occur twice in $R \cdot \underline{R}$. Describe a solution that takes $K+K^{\prime} \log r$ bits of space in addition to the bidirectional BWT index of $R$, where $K$ is the number of distinct $k_{2}$-mers in $R$, and $K^{\prime}$ is the number of distinct $k_{2}$-mers that occur at least twice in $R \cdot \underline{R}$. This algorithm should perform $k_{2}+1$ operations for every position of the input file. Adapt your solution to filter out reads such that all their $k_{1}$-mers occur less than $\tau_{1}$ times in $R \cdot \underline{R}$, as described in Section 17.2.1. Finally, describe another solution for creating pre-clusters that uses string $R \cdot \underline{R}$ as input and that takes approximately twice the time and the space of Lemma 17.5.
17.10 Describe a way to parallelize the algorithm in Lemma 17.5 by using samples of $\mathrm{SA}_{R}$ (see Section 9.2.3).
17.11 Adapt the algorithm for document counting described in Section 8.4.2 such that, given a reference taxonomy $\mathcal{T}$, it computes the markers of every taxon $v \in \mathcal{T}$.
17.12 Let $\mathrm{MS}_{S, T, \tau}$ be the following generalization of the matching statistics vector described in Section 11.2.3: $\mathrm{MS}_{S, T, \tau}[i]$ is the longest prefix of suffix $S[i . .|S|] \#$ that occurs at least $\tau$ times in $T$. Similarly, let $\mathrm{SUS}_{T, \tau}[i]$ be the shortest prefix of $T[i . .|T|] \#$ that occurs at most $\tau$ times in $T$. Describe an algorithm that computes the markers of every taxon $v$ in a reference taxonomy $\mathcal{T}$ using the shortest unique substring vector $\operatorname{SUS}_{S, \tau}$, the matching statistics vector $\mathrm{MS}_{S, T, \tau+1}$, and a bitvector $f \operatorname{lag}_{S, T, \tau}[1 . .|S|]$ such that $\mathrm{flag}_{S, T, \tau}[i]=1$ if and only if $S\left[i . . i+\mathrm{MS}_{S, T, 1}[i]-1\right]$ occurs at most $\tau$ times in $T$, for every genome $T \neq S$ in the database.
17.13 Prove that the number of $k$-submaximal repeats of a string $S$ is upper-bounded by the number of distinct $k$-mers that occur at least twice in $S$.
17.14 With reference to Insight 17.1, prove that attaching the root of the smallest subtree to the root of the largest subtree guarantees $O(\log n)$ time for find.
17.15 With reference to Section 17.2.3, give the pseudocode of an algorithm that extracts all the maximal repeats of length at least $k_{4}$ from all pre-clusters, using a single enumeration of all the internal nodes of the suffix tree of $R$.
17.16 With reference to Section 17.3.2, describe a way to approximate $\mathcal{R} \otimes \mathcal{Q}$ that is more accurate than just computing $\mathcal{R} \odot \mathcal{Q}$. Hint. Use the operator $\odot$ iteratively on suitable subsets of $\mathcal{R}$ and $\mathcal{Q}$. Then, give the pseudocode of an algorithm that, given vector $\mathrm{MS}_{R^{i}}$ for a read $R^{i}$, finds a set of nonoverlapping substrings of length $k$ of maximum size.

