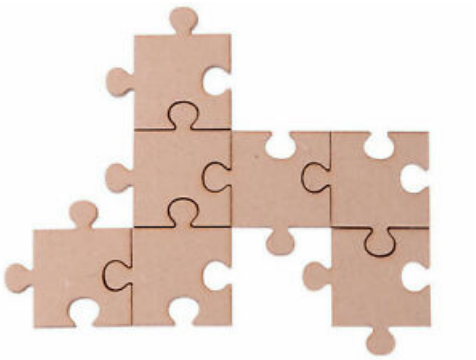
Algorithms in Computational Biology (INFO-F438)
Assignment 2: Shortest Common Superstring

Jean Cardinal (jcardin@ulb.ac.be)
Sofia Papadimitriou (Sofia.Papadimitriou@vub.be)

March 25, 2020



The shortest common superstring problem

Given a collection $\mathcal{C}$ of strings on an alphabet Σ, a common superstring of $\mathcal{C}$ is a string s on the same alphabet such that every string in $\mathcal{C}$ is a substring of s. Here, a substring is a contiguous subsequence, hence the symbols of each string in $\mathcal{C}$ must appear consecutively in s.

We consider the following computational problem, called the SHORTEST COMMON SUPERSTRING problem:

Input: A collection $\mathcal{C}$ of strings on an alphabet Σ.

Output: A shortest common superstring of $\mathcal{C}$

In what follows, we suppose that no string in $\mathcal{C}$ is a substring of another string in $\mathcal{C}$ (since in that case we can forget about the smaller one). The SHORTEST COMMON SUPERSTRING problem is known to be intractable unless $\mathsf{P} = \mathsf{NP}$. Hence it is most likely that there does not exist any polynomial-time algorithm to solve this problem exactly. One can, however, design an exhaustive search algorithm that checks all possible permutations of the strings in $\mathcal{C}$, and for each of them construct a common superstring in which the strings of $\mathcal{C}$ appear in that order.

Another idea is to give up on optimality and use a heuristic algorithm. A natural greedy algorithm is the following:

1. While $|\mathcal{C}| > 1$:
   (a) Pick two strings a, b in $\mathcal{C}$ with maximum overlap.
   (b) Remove a and b from $\mathcal{C}$ and replace them by their shortest common superstring
2. Return the unique string in $\mathcal{C}$.

Here the *overlap* between two strings a and b (in this order) is the length of the longest suffix of a that is also a prefix of b. The shortest common superstring of a, b is simply obtained by overlapping the suffix of a with the prefix of b.

Example: Let $\mathcal{C} = \{TCCC, CCCT, CCACC\}$. The unique shortest common superstring of $\mathcal{C}$ is $TCCCACCCT$, of length 9. The greedy algorithm starts by replacing the first two strings, having overlap 3, by $TCCCT$, and then add the last one to form either $TCCCTCCACC$ or $CCACCTCCCT$, of length 10.

Your Work

Write a program that takes a collection of strings in $\{A, C, T, G\}$ as input and computes two feasible solutions of the SHORTEST COMMON SUPERSTRING problem:

1. a shortest common superstring computed using **an exact algorithm**;
2. a common superstring computed using **the greedy algorithm**.

The goal will be to compare the output of the two algorithms, and empirically evaluate the performance of the greedy algorithm.

Requirements:

1. The source code of a program performing the above task, preferably in Python.
2. A number of relevant, non-redundant comments and explanations about your code, either in the form of comment lines, or in a separate report
3. The collection of instances on which you run your tests, with the design rationale.
4. A detailed comparison between the outputs of the two algorithms and an empirical evaluation of the greedy algorithm.

Further Readings

The problem is discussed in Chapter 8 of the course textbook. Problem 8.13, at the end of the Chapter, hints at a theoretical analysis of the greedy algorithm.

Evaluation

The evaluation will be based on the following criteria:

1. the general understanding of the instructions,
2. the proper use of the programming language,
3. the efficiency and correctness of the implemented algorithms,
4. the clarity and relevance of the comments and explanations,
5. the report on the comparison between the two algorithms.

Ethics

Plagiarism will be severely sanctioned. Plagiarism cases include reusing someone else's written or drawn material, or any kind of work, without an explicit quote or reference.
`http://www.plagiarism.org/`.

Deadline

Send your work to sofia.papadimitriou@vub.be until April 17, 2020 at midnight.