Minimal Absent Words in Plasmids
-
Software Report

Veronika Hendrychová, Nazar Misyats

December 2023

1 Program structure

- `maw/`: contains the core MAW calculation API.
  - `naive.py`, `better.py` and `fast.py`: implements the Naive, Extension and Suffix-Array methods respectively. They all expose a `find_maws(sequences: set[str], kmax: int) -> dict[int, set[str]]` function which returns a dictionary `maws` such that `maws[k]` is the set of all MAWs of length `k` in the set of sequences.
  - `fmt.py`: utilities for formatting the output of `find_maws`.
  - `readfa.py`: parse and read FASTA files.
  - `utils.py`: generic utilities and global constants when dealing with DNA strings.
  - `karkkainen_sanders.py`: linear-time suffix array implementation.
  - `main.py`: command line entrypoint.
- `tests/`: Unit test folders.
- `benchmark.py`: A script for automatic benchmarking and comparison of the different algorithms.

2 Implementation details

2.1 Naive approach

All possible strings for a given length are generated in lexicographic order. The `increment_lexicographic` function increments an integer in a given base represented in an array by one, and returns `False` if this increment exceeded the maximum value representable. The function `generate_lexicographic` then iterates through all numbers representable on some number of digits, in a base the size of the given alphabet. The number is mapped backed to a string when yielded.

2.2 Extensions approach

Possible candidates for MAWs are generated as one-letter extensions (left or right) of all substrings from the given sequences. The function `get_all_maws` then checks the necessary condition for the given candidate to be MAW, and creates a set of MAWs for the `find_maws` function.

2.3 Suffix-Array approach

A class `Sequence` is implemented to more easily manipulate a string and its associated SA and LCP array. The global method `build_lcp` implements Kasai's algorithm, and is called in the constructor of `Sequence`. The substring generation is implemented in the `substrings` member method.

A logarithmic time search of substrings in a sequence is implemented (overload of the `__contains__` operator), but didn't end up being used has Python's `set` type showed better results.