Tropin Nikolay

tropinnikolay@gmail.com

4 March 2021

1 Выравнивание на многие последовательности

В нашей задаче будем учитывать, что алфавит у нас небольшой (либо 4 нуклеотида, либо около 20 аминокислот) - поэтому опускаем данную константу при расчёте асимптотики.
Построение бора на заданных последовательностях имеет время $O(|genomes|)$, где

$$|genomes| = \sum_i length(genome_i)$$

Т.е. в худшем случае, когда все N последовательностей имеют длину L и не имеют общего префикса, то время построения будет равно $O(N \cdot L)$, памяти потребуется ровно столько же.
Ответ на запрос для строки длины M исходя из алгоритма получается обходом всех вершин бора, при этом считаем, что в каждой вершине мы совершаем выравнивание нашей строки. Т.о. посещение всех $N \cdot L$ вершин и выравниваение нашей последовательности с единственным символом в каждой вершине даёт итоговое время ответа на запрос $O(M \cdot N \cdot L)$

2 Поиск подстроки

Для использования BWT и поиска вхождений подстроки длины M в текст длины N нам необходимо построить суффиксный массив - за $O(N \cdot \log N)$ по времени и за $O(N)$ по памяти. Тут мы снова пренебрегли константой алфавита, иначе было бы $O((N+|alphabet|) \cdot \log N)$ по времени и $O(N+|alphabet|)$ по памяти.
По сути мы используем первый и последний столбцы матрицы циклических перестановок (которая ещё и была отсортирована), поэтому индексирование можно выполнить за время $2 * N$. Далее восстановление исходного текста (по переходам между столбцами по индексам) тоже производится за линию, потому что каждый переход - константа.
При добавлении оптимизации с чекпоинтами мы ничего не портим по асимптотике и получаем те же $O(1)$ по времени для переходов и $O(n)$ памяти, но

уже с меньшей константой.
Теперь разберёмся с ответом на запрос для строки длины M: переходя между столбцами F и L мы в конечном итоге найдём полное вхождение строки в текст (если таковое существует), а индекс вхождения получим из суффиксного массива. При множественном вхождении нашей строки алгоритм по сути не изменяется - мы аналогично будем прыгать по столбцам, но уже учитывая все вхождения.
Т.о. найти все вхождения строки в текст можем за $O(M)$.