Краткий конспект

Лекция 4. Суффиксные деревья

версия 0.1(draft)

Д. Ищенко* Б. Коварский* И. Алтухов* Д. Алексеев*

7 марта, 2016

*МФТИ

1 Задача поиска k мотивов в геноме

Вернемся к задаче поиска мотива m в геноме g. В лекции о Z-алгоритме мы выяснили, что ее можно решить за линейное время $O(|m|+|g|)$, где $|m|$ - длина мотива, $|g|$ - длина генома. Представим, что необходимо найти k различных мотивов в геноме g. Используя предыдущий метод, мы будем конструировать «Z-ящики» для каждой объединенной строки $m\$g$ и производить поиск. Соответственно временная сложность алгоритма вырастет до $O(k(|m|+|g|))$, в виду того, что $|m| \ll |g|$, сложность - $O(k|g|)$.

В реальных задачах k может иметь достаточно большое значение, например, при картировании ридов (коротких последовательностей) на референсный геном, значения k могут быть порядка $10^6 - 10^8$. Возникает необходимость оптимизации алгоритма. В этой лекции мы рассмотрим такую конструкцию, как суффиксное дерево и покажем, как с помощью него решить задачу поиска мотива и несколько других задач, существенно уменьшив временную сложность вычислений относительно Z-алгоритма.

2 Суффиксное дерево

По аналогии с ранее введенными префиксами строки S длиной n, назовем **k-ым суффиксом** строки S ее подстроку, начинающуюся с k-го и заканчивающуюся последним символом: $S[k..n]$. Например, для строки TATATG все множетсво суффиксов будет представлять из себя:

```
1: TATATG
2: ATATG
3: TATG
4: ATG
5: TG
6: G
```

Отметим, что количество всех суффиксов строки S равно ее длине, а также тот факт, что полная строка также является суффиксом самой себя (первый суффикс).

Введем понятие *суффиксного дерева*. В общем виде – это древовидный граф с набором вершин и ребер, создающийся из всех суффиксов последовательности S. В графе к каждому ребру приписана «метка», соответствующая некоторой подстроке из S. Двигаясь по ребрам от выделенной вершины суффиксного дерева, называемой корнем, к одному из листьев дерева (вершине, к которой ведет только одно ребро) и соединяя последовально «метки» ребер в одну последовательность, в конечном итоге мы получаем один из суффиксов исходной последовательности S. Неформально процесс создания суффиксного дерева можно описать следующим образом:

(i) допишем в конец последовательности S символ, не встречающийся в самой последовательности, например: **$**, тем самым получим последовательность $S^* = S\$$

(ii) возьмем все суффиксы последовательности S^*, «закрепим» их начала в одной вершине, называемой корнем дерева (Рис. 1), в «концы» суффиксов поместим вершины и соединим их ребрами с корнем дерева. Вершины пронумеруем от 1 до n, ребрам припишем «метки» равные соответствующим суффиксам.

(iii) «склеим» ребра с совпадающими началами, объединив их общую часть в одно ребро и «ответвив» от новой внутренней вершины несовпадающие части. Образованный граф представляет из себя суффиксное дерево для последовательности S^*.

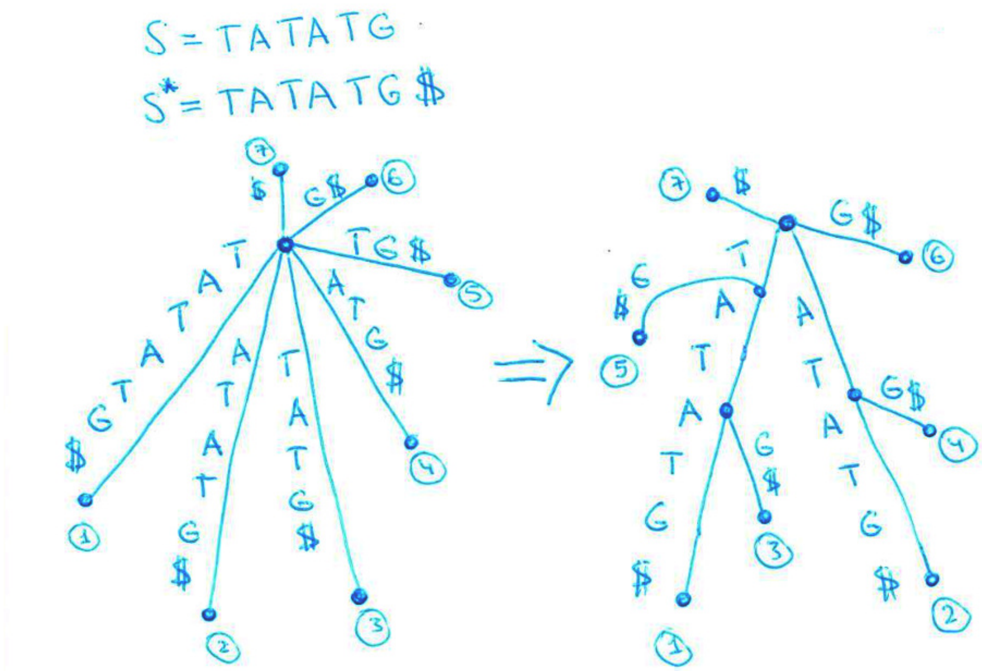

Рис. 1: Создание суффиксного дерева для последовательности S.

Перечислим некоторые свойства суффиксного дерева:

(i) Количество листьев дерева равно количеству суффиксов последовательности S^*.

(ii) Каждый узел в дереве, кроме корневого, имеет ровно один родительский узел.

(iii) Двигаясь по ребрам от корня дерева к одному из *листьев* и объединяя метки соответствующих ребер в одну последовательность, последняя будет соответствовать одному из *суффиксов* последовательности S^*.

(iv) Двигаясь по ребрам от корня дерева к одной из *внутренних вершин* дерева и объединяя метки соответствующих ребер в одну последовательность, последняя будет соответствовать некоторой *подстроке* из S^*.

(v) Для любой подстроки S^* можно найти соответсвующий ей путь от корня дерева, причем, путь не обязательно завершается в вершине дерева, а может быть заверешен в середине одного из ребер.

(vi) Метки ребер выходящих из корня или любой внутренней вершины отличаются первыми символами (иначе они были бы склеены в одно ребро). Следовательно из любой вершины не может выходить больше $\sigma + 1$ ребер, где σ – размер алфавита (количество разных символов, для нуклеотидной последовательности $\sigma = 4$).

(vii) Если общее число вершин в дереве равно N, то число ребер: $(N - 1)$.

Предположим, что мы построили для последовательности S суффиксное дерево, как с помощью него определить количество вхождений мотива p в строку S?

3 Алгоритм посроения $O(n^2)$

```
procedure TreeCreation(S)
    S = S + "$"
    nodes ← [[1], [−1]]
    edges ← ["", S]

    for i ← 1, len(S) do
        suf ← S[i..len(S)]
        j ← 0
        cur_node ← 0
        to_node ← −1
        isAdded ← False

        while not isAdded do
            for k in nodes[cur_node] do
                if suf[j] = edges[k][0] then
                    to_node ← k
                end if
            end for

            if to_node = −1 then
                ...                                  ▷ Добавляем новый лист и ребро
                isAdded ← True
            else
                for p ← 1, len(edges[to_node]) do
                    if suf[j + p]! = edges[to_node][p] then
                        ...            ▷ Добавляем новый лист и внутреннюю вершину
                        isAdded ← True
                        break
                    end if
                end for
                j ← j + len(edges[to_node])
                cur_node = to_node
            end if
        end while
    end for

    return [nodes, edges]
end procedure
```

4 Поиск в глубину

Решаем задачу поиска мотива, как посчитать количество вхождений?

```
procedure LeavesCount(i, nodes)
    if nodes[i][0] = −1 then
        return 1
    end if

    lCount ← 0
    for k in nodes[i] do
        lCount = lCount + LeavesCount(k, nodes)
    end for

    return lCount
end procedure
```

5 Какие еще задачи можно решить с помощью дерева

Поиск повтора, поиск максимальной общей подстроки, нечеткий поиск.

6 Ссылки