Разработка метода сравнения нуклеотидных последовательностей путём разбиения на фрагменты

Капун Евгений Дмитриевич

Научный руководитель: д. т. н., проф. А. А. Шалыто

Санкт-Петербургский государственный университет информационных технологий, механики и оптики

2010

Постановка задачи

Задано две последовательности нуклеотидов. Необходимо определить степень их сходства.
Сходство — число от нуля до единицы. Сходство последовательности самой с собой равно единице.
Мутации не должны значительно изменять сходство. Чем чаще происходят мутации некоторого типа, тем они меньше должны влиять на сходство.

Виды мутаций

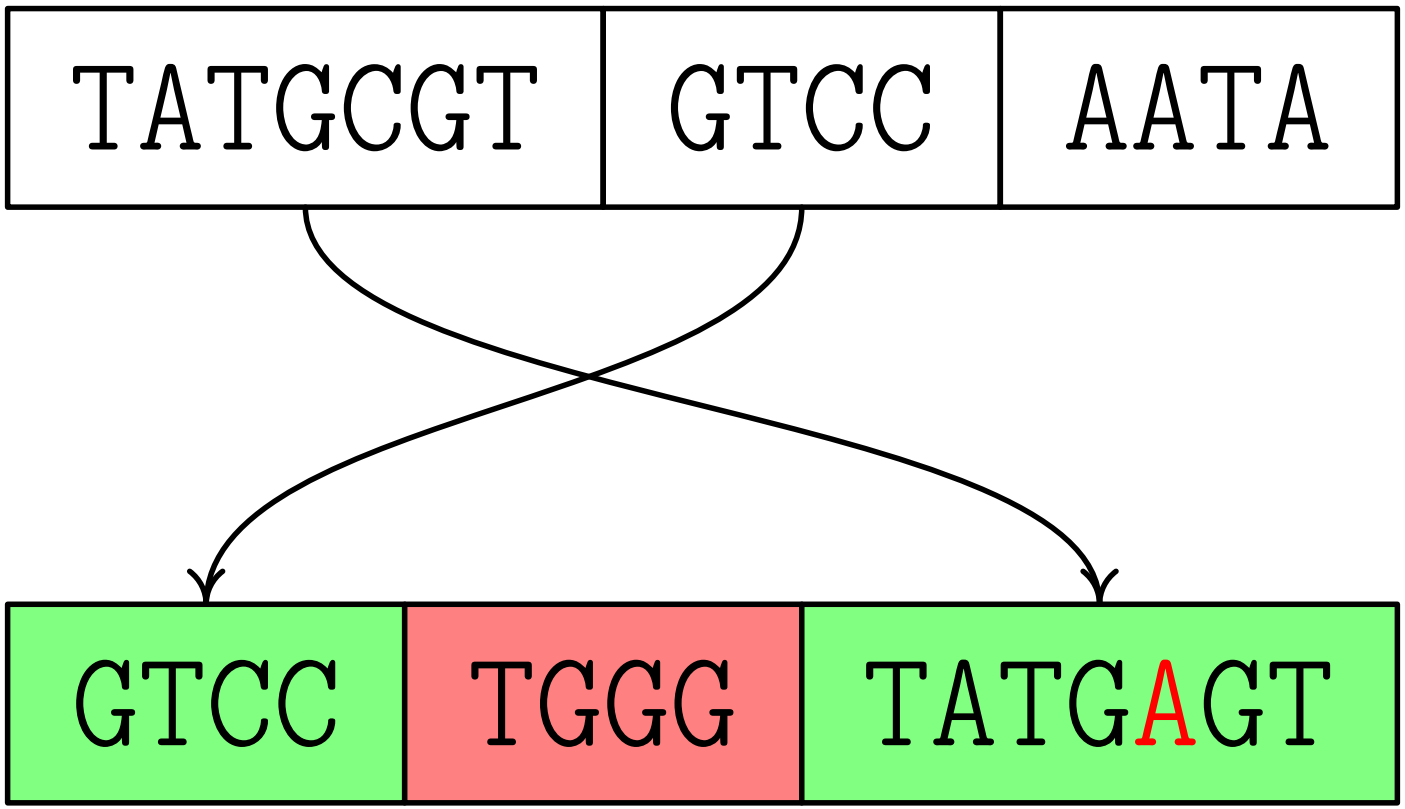
- Локальные мутации:
  - Замена — один нуклеотид заменяется на другой, делятся на *транзиции* и *трансверсии*.
  - Добавление нуклеотида.
  - Удаление нуклеотида.
- Нелокальные мутации:
  - *Дупликация* — копирование участка ДНК. Копия может появиться в произвольном месте, да ещё и потом мутировать независимо от оригинала.
  - *Делеция* — удаление участка ДНК.
  - *Транслокация* — перемещение участка ДНК.
  - *Инверсия* — замена участка ДНК на комплементарный. Нуклеотиды заменяются на комплементарные и переставляются в обратном порядке.

Предлагаемое решение

Решение — покрывать одну последовательность фрагментами другой.

- Называем одну последовательность исходной, а другую — целевой.
- Разбиваем целевую последовательность на фрагменты. Некоторые фрагменты называем свободными, а остальные — связанными.
- Каждому связанному фрагменту сопоставляем подстроку в исходной последовательности.

Пример разбиения



Степень различия

При подсчёте степени различия учитывается:

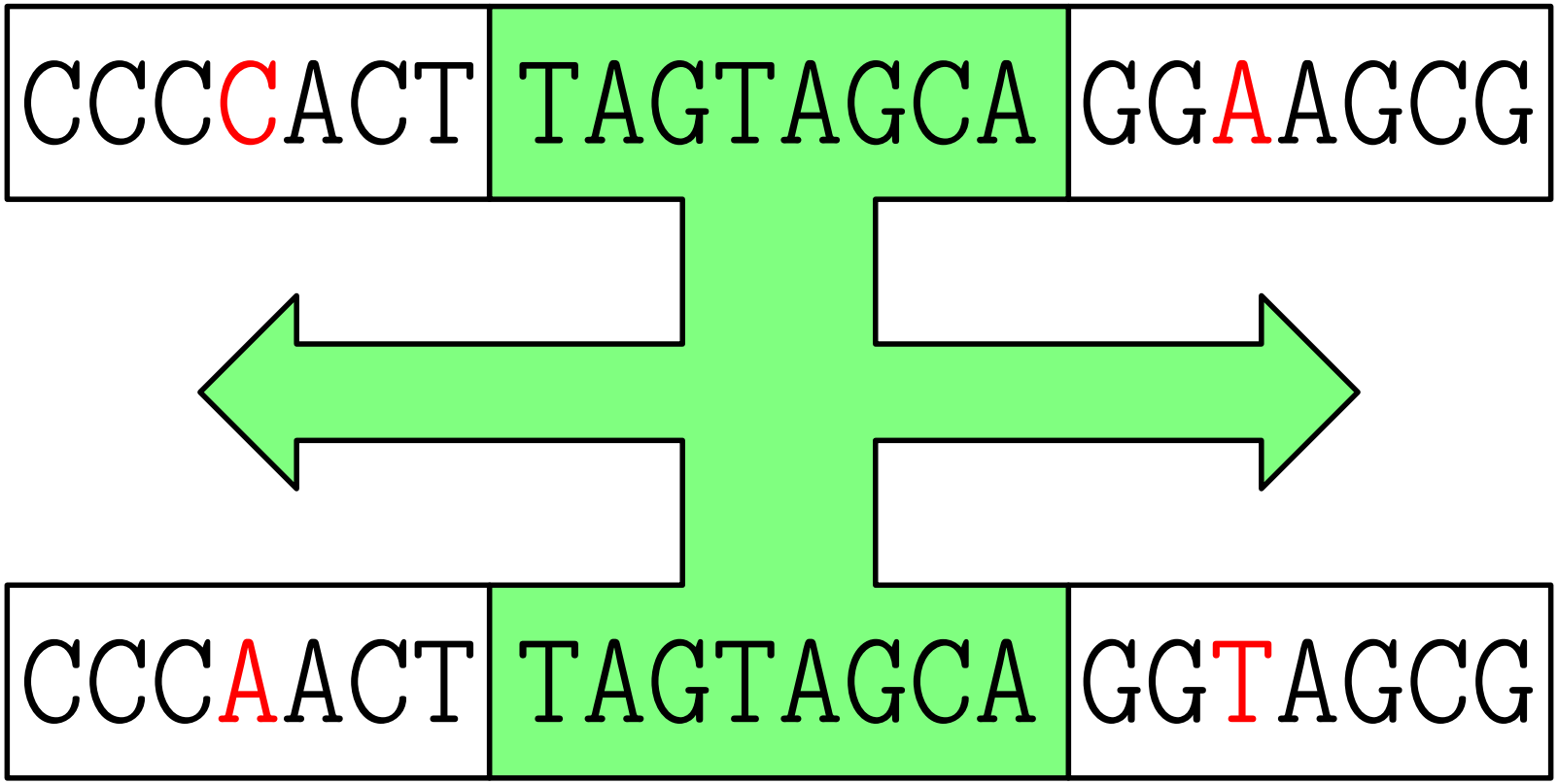
- Суммарная длина свободных последовательностей.
- Суммарное редакционное расстояние связанных последовательностей.
- Общее количество последовательностей.

Из всех разбиений выбирается такое, при котором степень различия минимальна.

Алгоритм

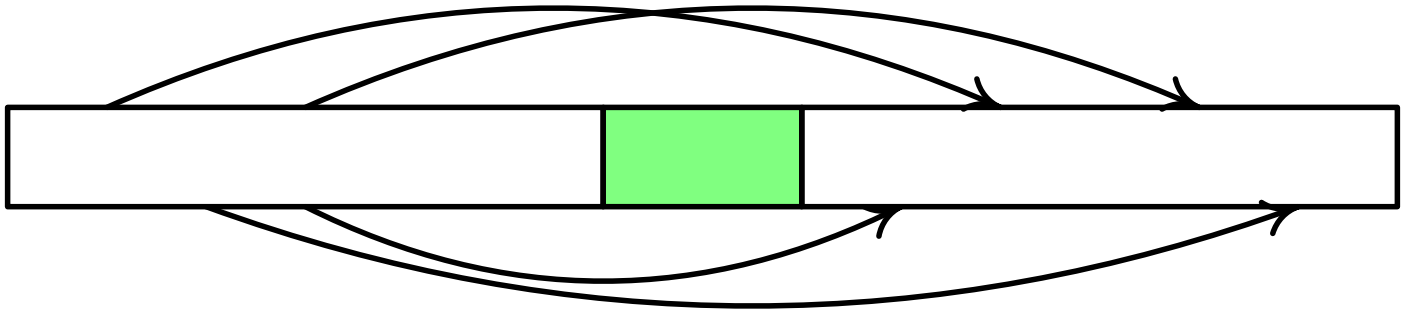
- Составить словарь: каждой строки заданной длины, встречающейся в исходной строке, сопоставляется список вхождений в исходную строку.
- Перебрать все подстроки такой же длины в целевой строке и запросить её в словаре. Каждую пару (вхождение в целевую строку, вхождение в исходную строку) попытаться расширить до потенциального фрагмента.

Расширение вхождений



Вычисление результата

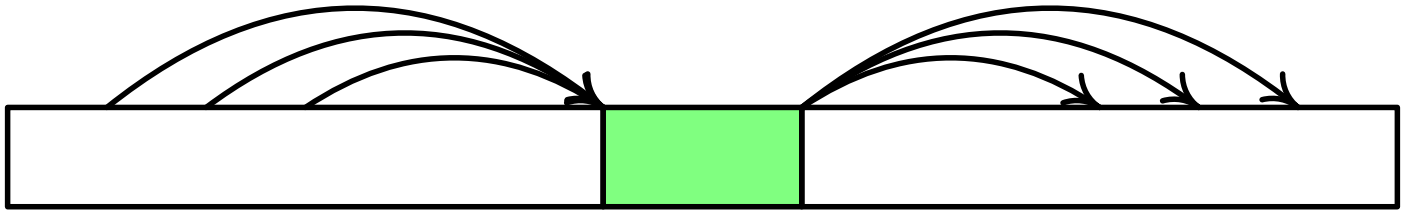
Для вычисления результата используется динамическое программирование. Результат вычисляется для каждого префикса целевой строки.



В каждый момент времени значения слева от ключа уже посчитаны, а справа ещё нет.

Вычисление результата

На самом деле, вместо одного квадратичного пересчёта используется два линейных:



Результаты

	NC_012759	NC_012947	NC_012967	NC_010067	NC_002952
NC_012759	100%	90%	88%	11%	0,1%
NC_012947	90%	100%	96%	11%	0,1%
NC_012967	89%	97%	100%	11%	0,1%
NC_010067	11%	11%	11%	100%	0,1%
NC_002952	0,1%	0,1%	0,1%	0,1%	100%