Лабораторная работа № 4

Выравнивание биологических последовательностей

1 Глобальное и локальное выравнивание последовательностей

Выравнивание последовательностей используется для решения следующих задач.

- Наблюдать закономерности сохранения (или изменчивости).
- Найти общие мотивы, присутствующие в обеих последовательностях.
- Оценить вероятность того, что две последовательности произошли от одной и той же последовательности.
- Чтобы узнать, какие последовательности из базы данных похожи на имеющуюся последовательность.

Есть два пути для сравнения последовательностей

- точечная матрица сходства (dotplot) — визуальный, качественный метод,
- выравнивание последовательности (sequence alignment) — точный, количественный метод.

Выравнивание последовательности включает:

- Построение наилучшего выравнивания между последовательности.
- Оценка подобия по выравниванию.

Существует два различных типа выравнивания последовательностей:

- Парное выравнивание,
- Множественное выравнивание последовательностей.

Парное выравнивание бывает глобальным и локальным.

- *Глобальное выравнивание* применяется к двум входным последовательностям целиком. Более удобно в случае двух гомологичных последовательностей.

- *Локальное выравнивание* применяется, если последовательности содержат как родственные , так и неродственные участки. Результатом локального выравнивания является выбор участка в каждой из последовательностей и выравнивание между этими участками.

Пример.

Глобальное выравнивание:

```
_  _  T  _  _  C  C  _  C  _  A  G  T  _  _  T
      |        |  |     |        |  |        |
A  A  T  T  G  C  C  G  C  C  _  G  T  C  G  T
```

Локальное выравнивание:

```
_  _  _  _  A  C  G  T  C  C  A  G  T  A  A  T
            |  |  |  |  |  |
C  A  T  T  A  C  G  T  C  C  C  T  _  _  _  _
```

Парное выравнивание последовательности — это установление соответствие буквы букве между двумя последовательности такое, что порядок букв в каждой последовательности сохраняется.

Пробел, указывающий на то, что буква не совпадает ни с чем, может быть введены в любой последовательности. Совпадение двух пробелов не имеет смысла и не допускается. Даны две последовательности, нужно найти число, соответствующее каждому возможному выравниванию. То есть оценка выравнивания соответствует качеству выравнивания.

Схема подсчета очков представляет собой набор правил, по которым осуществляется оценка выравнивания для любого заданного выравнивания двух последовательностей. Есть два подхода:

- Максимизировать сходство.
- Минимизировать различие.

Простая схема начисления очков.

- $+1$ очко в качестве награды за совпадение,
- -1 очко в качестве штрафа за несоответствие,

- пробелы игнорируются.

Примеры:

1.

ATGGCGT

ATG−AGT

Очки: $+1+1+1+0-1+1+1=4$

2.

ATGGCGT

A−TGAGT

Очки: $+1+0-1+1-1+1+1=2$

Здесь мы максимизировали сходство.

Краткий способ выражения стоимости замещения остатков может быть достигнут с помощью матрицы $N \times N$ (N равно 4 для ДНК и 20 для белков).

Матрица замещения для простой схемы подсчета очков:

	a	t	g	c
a	1	-1	-1	-1
t	-1	1	-1	-1
g	-1	-1	1	-1
c	-1	-1	-1	1

Рассмотрим возможность нахождения наилучшего выравнивания полным перебором всех выравниваний.

Подход Brute-force включает в себя:

- Создать список всех возможных выравниваний между двумя последовательности и оценить их.
- Выбрать выравнивание с лучшим результатом.

Количество возможных глобальных выравниваний между двумя последовательности длины N

$$\frac{2^{2N}}{\sqrt{\pi N}}$$

Для двух последовательностей из 250 букв это число 10^{149}.

2 Алгоритмы динамического программирования для парного выравнивания последовательностей

Для получения парного выравнивания используются разновидности метода динамического программирования: для глобального выравнивания — алгоритм *Нидлмена–Вунша*, для локального — алгоритм *Смита–Ватермана.*

Алгоритмы Смита-Ватермана и Нидлмана-Вунша служат разным целям, но оба алгоритма используют понятия матрицы замен, функции штрафа за пропуски, матрицы подсчета очков и штраф за делецию.

Схема, содержащая все возможные выравнивания, может быть построена в виде матрицы наподобие той, которая используется для изображения точечной матрицы сходства. Остатки одной последовательности индексируют строки, а остатки другой последовательности – столбцы. Любой путь по матрице, начинающийся в левом верхнем углу и заканчивающийся в правом нижнем, соответствует одному выравниванию. Задача – найти путь наименьшего веса, и трудность состоит в том, что таких путей нужно рассмотреть очень много.

Алгоритм глобального оптимального выравнивания двух последовательностей Нидлмена–Вунша, дает максимальное количество баллов (максимизирующий сходство двух последовательностей).

Алгоритм Нидлмана–Вунша (Сол Нидлман и Кристиан Вунш, 1970) предлагает способ уменьшить огромное количество возможностей, которые необходимо учитывать, но при этом гарантировать, что будет найдено лучшее выравнивание.

Основная идея состоит в том, чтобы создать наилучшее выравнивание, используя оптимальное выравнивание меньших подпоследовательностей. Алгоритм Нидлмана-Вунша является примером динамического программирования, дисциплины, изобретенной Ричардом Беллманом (американским математиком) в 1953 году.

Алгоритм Смита–Ватермана был предложен Т. Ф. Смитом и М. Ватерманом в 1981. Подобно алгоритму Нидлмана–Вунша, алгоритм Смита–Ватермана использует принцип динамического программирования. Он гарантирует нахождение оптимального, относительно используемой им меры оценки качества, локального выравнивания. Эта мера оценки — так называемый вес, или счёт (Score) выравнивания, предусматривающий использование матрицы замен и штрафов за пропуски.

Динамическое программирование работает по стратегии «разделяй и властвуй»:

- Проблему разбивается на более мелкие подзадачи.
- Находится оптимальное решение каждой небольшой подзадачи.
- Решения подзадач используется для построения оптимального решение исходной задачи.

Динамическое программирование можно применять только к задачам биоинформатики.

Примеры:

- Задача коммивояжера.
- Поиск лучшего шахматного хода.

3 Алгоритм глобального выравнивания Нидлмана-Вунша

Алгоритм Нидлмана-Вунша решает задачу глобального выравнивания двух последовательностях, применяя матрицу сходства $M_{i,j}$, где i и j изменяются от нуля до длины сравниваемых строк. Это позволяет учитывать различные замены, вставки и удаления отдельно взятых букв.

Входными параметрами метода являются две последовательности S_1 и S_2 длины n и m соответственно и матрица замещения, определяющая штрафы разного рода несоответствия для каждой комбинации элементов выравнивания.

Алгоритм глобального выравнивания Нидлемана-Вунша заключается в заполнении матрицы M. На начальном этапе верхний левый элемент матрицы считается равным нулю, для вычисления значений прочих элементов верхней строки $M_{1,j}$ и левого столбца $M_{i,1}$ используются, соответственно, формулы

$$M_{i,j} = M_{i,j-1} + g$$

и

$$M_{i,j} = M_{i-1,j} + g,$$

где g — число, прибавляемое к весу выравнивания за каждый из пробелов.

Значение каждого из остальных элементов$M_{1,j}$ i-й строки j-го столбца матрицы переходов M вычисляются по формуле

$$M_{i,j} = \max\{M_{i-1,j} + g, M_{i,j-1} + g, M_{i-1,j-1} + w\}, \qquad i, j > 1,$$

где g— штраф за пробел, w — штраф за несовпадение в соответствии с матрицей замен.

Если максимум достигается в $M_{i-1,j} + g$, то переход в элемент $M_{i,j}$ осуществляется из элемента $M_{i-1,j}$; если в $M_{i,j-1} + g$, то из элемента $M_{i,j-1}$; если в $M_{i-1,j-1} + w$ то из элемента $M_{i-1,j-1}$.

Результатом выравнивания будет являться число, полученное в правом нижнем углу матрицы M. Для построения самого выравнивания необходимо совершить обратный обход матрицы, начиная с ячейки правого нижнего угла, по ячейкам, имеющих максимальный вес. Данный алгоритм имеет квадратичные затраты памяти и времени.

Пример глобального выравнивания последовательностей

Даны две последовательности:

ACTGATTCA

ACGCATCA

Найти глобальное выравнивание, используя алгоритм Нидлмана-Вунша.

Использовать

-2 в качестве штрафа за пробел,

-3 в качестве штрафа за несоответствие и

2 за совпадение.

Решение:

Шаг 1: Нарисуйте матрицу. Для двух последовательностей длиной m и длиной n размер матрицы должен быть $(m + 1) \times (n + 1)$.

Шаг 2: Заполним матрицу M.

Шаг 3: Найдем оптимальный путь по матрице прослеживается, начиная с правого нижнего угла:

Шаг 4: Выпишем выравнивание: и найдем вес (оценку) этого выравнивания:

Положительная особенность рассматриваемого алгоритма состоит в том, что метод гарантирует *глобальный* оптимум: наилучший результат выравнивания при заданном наборе параметров (матрицы замен и штрафов для пропусков).

		A	C	T	G	A	T	T	C	A
	0	-2	-4	-6	-8	-10	-12	-14	-16	-18
A	-2	2	0	-2	-4	-6	-8	-10	-12	-14
C	-4	0	4	2	0	-2	-4	-6	-8	-10
G	-6	-2	2	1	4	2	0	-2	-4	-6
C	-8	-4	0	-1	2	1	-1	-3	0	-2
A	-10	-6	-2	-3	0	4	2	0	-2	2
T	-12	-8	-4	0	-2	2	6	4	2	0
C	-14	-10	-6	-2	-4	0	4	2	6	4
A	-16	-12	-8	-4	-5	-2	2	1	4	8

		A	C	T	G	A	T	T	C	A
	0	-2	-4	-6	-8	-10	-12	-14	-16	-18
A	-2	2	0	-2	-4	-6	-8	-10	-12	-14
C	-4	0	4	2	0	-2	-4	-6	-8	-10
G	-6	-2	2	1	4	2	0	-2	-4	-6
C	-8	-4	0	-1	2	1	-1	-3	0	-2
A	-10	-6	-2	-3	0	4	2	0	-2	2
T	-12	-8	-4	0	-2	2	6	4	2	0
C	-14	-10	-6	-2	-4	0	4	2	6	4
A	-16	-12	-8	-4	-5	-2	2	1	4	8

```
ACTG-ATTCA
|| | || ||
AC-GCAT-CA
```

```
Score = (AA) + (CC) + (T-) + (GG) + (-C) +
        (AA) + (TT) + (T-) + (CC) + (AA)
      = 2+2-2+2-2+2+2-2+2+2
      = 8
```

Отрицательная особенность состоит в том, что многие выравнивания могут привести к оптимальному, но одному и тому же числу баллов. И совершенно необязательно, что хотя бы одно из них имеет отношение к биологически корректному выравниванию.

Пример. При сравнении последовательностей $\alpha-$ и $\beta-$цепей гемоглобина курицы ученые В. Фитч и Т. Смит нашли 17 выравниваний с одинаковыми оптимальными баллами, из которых корректно только одно (на основании сравнения трехмерных структур).

4 Алгоритм локального выравнивания Смита–Ватермана

Алгоритм Смита–Ватермана решает задачу поиска гомологичных участков в двух последовательностях, применяя матрицу сходства $M_{i,j}$, где i и j изменяются от нуля до длины сравниваемых строк. Это позволяет учитывать различные замены, вставки и удаления отдельно взятых букв.

Входными параметрами метода являются две последовательности S_1 и S_2 длины n и m соответственно и матрица замещения, определяющая штрафы разного рода несоответствия для каждой комбинации элементов выравнивания.

На момент начала построения значения элементов верхней строки и левого столбца матрицы M считаются равными нулю. В процессе построения значение каждого из остальных элементов $M_{i,j}$ i-й строки j-го столбца матрицы M вычислялось по формуле

$$M_{i,j} = \max\{M_{i-1,j} + g, M_{i,j-1} + g, M_{i-1,j-1} + w, 0\}, \qquad i, j > 1,$$

где g, w принимают различные значения в зависимости от того, совпадают ли соответствующие элементу $M_{i,j}$ буквы двух последовательностей.

Находится наибольший из аргументов $M_{i-1,j} + g$, $M_{i,j-1} + g$, $M_{i-1,j-1} + w$, 0

Для каждого $M_{i,j} > 0$ направление перехода определяется следующим образом: если максимум достигается в $M_{i-1,j} + g$, то переход в элемент $M_{i,j}$ осуществляется из элемента $M_{i-1,j}$; если в $M_{i,j-1} + g$, то из элемента $M_{i,j-1}$; если в $M_{i-1,j-1} + w$, то из элемента $M_{i-1,j-1}$.

Максимальный элемент матрицы будет равен максимальной степени сходства.

Для получения значения веса для каждого элемента матрицы динамического программирования добавлена возможность нулевого веса, в случае, если все остальные варианты выбора дают отрицательные значения.

Для получения картины выравнивания необходимо осуществить обратный обход матрицы, начиная с элемента с наибольшим количеством очков в матрице подсчёта и заканчивая в ячейке матрицы, имеющей оценку 0.

Пример локального выравнивания последовательностей

Даны две последовательности:

ATGCATCCCATGAC

TCTATATCCGT

Найти локальное выравнивание, используя алгоритм Смита–Ватермана.

Использовать

−2 в качестве штрафа за пробел,

−3 в качестве штрафа за несоответствие и

2 за совпадение.

Решение:

Заполним матрицу M и найдем путь от наибольшего элемента до нуля. $(m+1) \times (n+1)$.

		A	T	G	C	A	T	C	C	C	A	T	G	A	C
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T	0	0	2	0	0	0	2	0	0	0	0	2	0	0	0
C	0	0	0	0	2	0	0	4	2	2	0	0	0	0	2
T	0	0	2	0	0	0	0	2	1	0	0	2	0	0	0
A	0	2	0	0	0	2	0	0	0	0	2	0	0	2	0
T	0	0	4	2	0	0	2	0	0	0	0	4	2	0	0
A	0	2	0	0	0	2	0	0	0	0	2	0	0	2	0
T	0	0	4	2	0	0	4	2	0	0	0	4	0	0	0
C	0	0	2	0	4	0	0	6	4	2	0	0	0	0	2
C	0	0	0	0	2	0	0	4	8	6	4	2	0	0	2
G	0	0	0	2	0	0	0	2	6	5	3	1	4	2	0
T	0	0	2	0	0	0	2	0	4	3	2	5	3	1	0

Выпишем выравнивание: и найдем вес (оценку) этого выравнивания:

```
ATCC
||||
ATCC
```

```
Score = (AA) + (TT) + (CC) + (CC)
      = 2 + 2 + 2 + 2
      = 8
```

Отличия алгоритма Смита–Ватермана от алгоритма Нидлмана-Вунша

	Алгоритма Смита–Ватерман	Алгоритма Нидлмана-Вунша
Инициализация	Первый столбик и первая строка заполняются нулями	Первый столбик и первая строка заполняются в соответствии со штрафами за пробел
Оценка	Отрицательная оценка зануляется	Оценка может быть отрицательной
Путь по матрице	Начинается с наибольшего числа и заканчивается в клетке с нулем	Начинается в нижней правой клетке и заканчивается в левой верхней

5 Задания

Задание 1.

Найти наилучшее глобальное выравнивание между двумя строками нуклеотидных последовательностей при заданной матрицы весов. Файлы для примера брать в соответствии с вариантом из Лабораторной работы 2 из базы данных GenBank.

Входные данные. Строки V и W и матрица весов.

Выходные данные. Глобальное выравнивание между V и W вес которого (определенный по матрице весов) является максимальным среди всех возможных выравниваний V и W.

Задание 2.

Найти наилучшее локальное выравнивание между двумя строками нуклеотидных или аминокислотных последовательностей. Файлы для примера брать в соответствии с вариантом из Лабораторной работы 2 из базы данных GenBank.

Входные данные. Две строки V и W и матрица весов.

Выходные данные. Локальное выравнивание, определяемое подстроками строк V и W, глобальное выравнивание которых определенное матрицей весов, является наилучшим среди всех глобальных выравниваний всех подстрок V и W.

Варианты матриц замен.

Вариант 1.

	a	t	g	c
a	1	-1	-1	-1
t	-1	1	-1	-1
g	-1	-1	1	-1
c	-1	-1	-1	1

Штраф за пробел -1.

Вариант 2.

	a	t	g	c
a	10	-1	-3	-4
t	-1	7	-5	-3
g	-3	-5	9	0
c	-4	-3	0	8

Штраф за пробел -2.

Вариант 3.

	a	t	g	c
a	5	-4	-4	-4
t	-4	5	-4	-4
g	-4	-4	5	-4
c	-4	-4	-4	5

Штраф за пробел -1.

Вариант 4.

	a	t	g	c
a	1	0	0	0
t	0	1	0	0
g	0	0	1	0
c	0	0	0	1

Штраф за пробел 0.

Вариант 5.

	a	t	g	c
a	1	-5	-5	-1
t	-5	1	-1	-5
g	-5	-1	1	-5
c	-1	-5	-5	1

Штраф за пробел -1.

Вариант 6.

	a	t	g	c
a	1	-5	-5	-1
t	-5	1	-1	-5
g	-5	-1	1	-5
c	-1	-5	-5	1

Штраф за пробел -2.

Вариант 7.

	a	t	g	c
a	30	-3	-20	-3
t	-3	30	-6	-20
g	-20	-6	30	-10
c	-3	-20	-10	30

Штраф за пробел -2.

Вариант 8.

	a	t	g	c
a	2	-1	-2	-2
t	-1	2	-2	-2
g	-2	-2	2	-1
c	-2	-2	-1	2

Штраф за пробел -1.

Вариант 9.

	a	t	g	c
a	6	-1	-4	-2
t	-1	10	-6	-2
g	-4	-6	6	-1
c	-2	-2	-1	11

Штраф за пробел -2.

Вариант 10.

	a	t	g	c
a	1	0	-1	-2
t	0	1	0	-1
g	-1	0	1	0
c	-2	-1	0	1

Штраф за пробел 0.

Вариант 11.

	a	t	g	c
a	1	-5	-5	-1
t	-5	1	-1	-5
g	-5	-1	1	-5
c	-1	-5	-5	1

Штраф за пробел -1.

Вариант 12.

	a	t	g	c
a	1	-5	-5	-1
t	-5	1	-1	-5
g	-5	-1	1	-5
c	-1	-5	-5	1

Штраф за пробел -1.

Вариант 13.

	a	t	g	c
a	1	-1	-1	-1
t	-1	1	-1	-1
g	-1	-1	1	-1
c	-1	-1	-1	1

Штраф за пробел 0.

Вариант 14.

	a	t	g	c
a	10	-1	-3	-4
t	-1	7	-5	-3
g	-3	-5	9	0
c	-4	-3	0	8

Штраф за пробел -2.

Вариант 15.

	a	t	g	c
a	5	-4	-4	-4
t	-4	5	-4	-4
g	-4	-4	5	-4
c	-4	-4	-4	5

Штраф за пробел -1.

Вариант 16.

	a	t	g	c
a	1	0	0	0
t	0	1	0	0
g	0	0	1	0
c	0	0	0	1

Штраф за пробел 0.

Вариант 17.

	a	t	g	c
a	1	-5	-5	-1
t	-5	1	-1	-5
g	-5	-1	1	-5
c	-1	-5	-5	1

Штраф за пробел -1.

Вариант 18.

	a	t	g	c
a	1	-5	-5	-1
t	-5	1	-1	-5
g	-5	-1	1	-5
c	-1	-5	-5	1

Штраф за пробел -2.

Вариант 19.

	a	t	g	c
a	1	-2	-2	-2
t	-2	1	-2	-2
g	-2	-2	1	-2
c	-2	-2	-2	1

Штраф за пробел -1.

Вариант 20.

	a	t	g	c
a	8	-1	-3	-4
t	-1	6	-5	-3
g	-3	-5	7	0
c	-4	-3	0	8

Штраф за пробел -2.

Вариант 21.

	a	t	g	c
a	5	-4	-4	-4
t	-4	5	-4	-4
g	-4	-4	5	-4
c	-4	-4	-4	5

Штраф за пробел -1.

Вариант 22.

	a	t	g	c
a	1	0	0	0
t	0	1	0	0
g	0	0	1	0
c	0	0	0	1

Штраф за пробел 0.

Вариант 23.

	a	t	g	c
a	1	-5	-5	-1
t	-5	1	-1	-5
g	-5	-1	1	-5
c	-1	-5	-5	1

Штраф за пробел -1.

Вариант 24.

	a	t	g	c
a	1	-5	-5	-1
t	-5	1	-1	-5
g	-5	-1	1	-5
c	-1	-5	-5	1

Штраф за пробел -2.

Вариант 25.

	a	t	g	c
a	30	-3	-20	-3
t	-3	30	-6	-20
g	-20	-6	30	-10
c	-3	-20	-10	30

Штраф за пробел -2.

Вариант 26.

	a	t	g	c
a	2	-1	-2	-2
t	-1	2	-2	-2
g	-2	-2	2	-1
c	-2	-2	-1	2

Штраф за пробел -1.

Вариант 27.

	a	t	g	c
a	6	-1	-4	-2
t	-1	10	-6	-2
g	-4	-6	6	-1
c	-2	-2	-1	11

Штраф за пробел -2.

Вариант 28.

	a	t	g	c
a	1	0	-1	-2
t	0	1	0	-1
g	-1	0	1	0
c	-2	-1	0	1

Штраф за пробел 0.

Вариант 29.

	a	t	g	c
a	1	-5	-5	-1
t	-5	1	-1	-5
g	-5	-1	1	-5
c	-1	-5	-5	1

Штраф за пробел -1.

Вариант 30.

	a	t	g	c
a	1	-5	-5	-1
t	-5	1	-1	-5
g	-5	-1	1	-5
c	-1	-5	-5	1

Штраф за пробел -1.