Оглавление

Введение

Исследование геномов организмов — одна из областей биоинформатики. Геном — это совокупность наследственного материала, заключенного в ДНК или РНК, хранящемся в ядре клетки. Геном содержит информацию, необходимую для построения и поддержания жизни организма. Для исследования и анализа содержания геном представляют в виде строки над алфавитом $\{A, C, G, T\}$. Процесс этого представления включает секвенирование (формальное описание первичной структуры линейной макромолекулы в виде последовательности мономеров в текстовом виде), сборку и аннотацию генома.

Задачи, связанные с технологиями и процессом секвенирования, лежат на биологах, физиках и инженерах. Биоинформатика занимается анализом, данных, полученных в ходе секвенирования, разработкой алгоритмов для повышения качества данных, основываясь на анализе стратегий и физических особенностей исследуемых технологий.

В процессе секвенирования получают данные, называемые чтениями (“прочитанный“ участок ДНК размером 100-300 нуклеотидов), которые представляют собой маленькую часть изначального генома (размером 10^5 — 10^7 нуклеотидов). Эти данные используются для сборки генома, что является отдельной непростой задачей.

Существуют разные технологии секвенирования – Illumina, PacBio, Ion Torrent, Sanger, отличающихся подходами к чтению молекул ДНК и РНК. Получаемые данные содержат ошибки (которые могут отличаться в разных подходах), что затрудняет задачу верной сборки генома. Исправить это обстоятельство можно двумя способами:

1. Исправить причину, по которой ошибки происходят. Это может быть невозможно, так как причиной могуть быть конкретные явления физической природы — например, невозможность получить достаточное количество молекул ДНК для исследования;

2. Исследовать характер ошибок и создать алгоритмы, которые могут находить и исправлять такие ошибки.

В последние годы в связи с удешевлением процесса методы секвенирования нового поколения стали использоваться повсеместно, не только в научных и лабораторных исследованиях, но и в клинической и фармакологических областях. Поэтому есть необходимость в разработке новых и усовершенствовании старых алгоритмов моделирования и поиска ошибок.

Одной из таких технологий является Ion Torrent [1], метод “секвенирования через синтез“, в ходе которого комплементарная цепь строится на основе последовательности исходной цепи. Метод основан на определении последовательности ДНК с помощью детектирования ионов водорода, выделяемых при полимеризации ДНК — построении комплементарной цепи.

Для чтений, полученных с помощью Ion Torrent, характерны ошибки типа вставок и удалений [2], [3]. Большое количество ошибок такого рода приводит к сдвигам в выравнивании и появлению неверных ошибок типа замен. Это может быть принципиально важно, например, в задачах определения ОНП (однонуклеотидных полиморфизмов).

Так как такие ошибки являются следствием особенности технологии, они типичны и воспроизводимы, и поэтому могут быть проанализированы и исправлены. Сначала следует обратить внимание на то, как работает метод.

Секвенирование методом Ion Torrent проходит следующим образом. Есть чип с лунками, в каждой из которой содержится шарик, на котором присутствует ДНК-полимераза и множество копий одного фрагмента одной цепочки ДНК длиной до 600 нуклеотидов. Происходит наращивание цепей ДНК. В каждую лунку последовательно зпомещаются нуклеотиды A, C, G, T в предопределенном порядке. Если вводимый нуклеотид комплементарен следующим нуклеотидам цепи, то он включается в растущую цепь с помощью ДНК-полимеразы, идет реакция связывания, в результате которого выделяются ионы водорода, изменяющие pH раствора, что детектируется датчиком, находящемся на дне лунки. Если связывания не происходит, сигнал не обнаруживается, то не прореагировавшие нуклеотиды удаляются. Процесс повторяется до окончания наращивания цепей.

В случае, когда в ДНК несколько одинаковых нуклеотидов идут подряд (то есть, в случае так называемого “гомополимера“ длины более 1), происходит множественное связывание, и, соответственно, детектируется сигнал большей величины. Проблема состоит в том, что с ростом длины гомополимера растет ошибка в распознавании его длины на основании данных о силе сигнала. Например, на настоящий момент гомополимер длины 5 читается верно только в 98% случаев [4].

Имея представление о том, в чем состоят ошибки секвенирования и о том, какая информация доступна после секвенирования, ставится задача построения оценивающей функции для выравнивания, выбирающей оптимальное выравнивание, в котором исключены ошибки, вызванные особенностями данной технологии секвенирования.

Основные идеи основываются на том, что процедура выравнивания должна происходить в пространстве гомополимеров (обычно выравнивание идет в пространестве нуклеотидов), что реализовано в PanGEA [4], а так же на том, что нужно использовать информацию о интенсивности синала (реализовано в FAAST [4]), и на том, что параметры для оценивающей функции можно выбирать, опираясь на некоторые реальные данные, полученные конкретной машиной.

Стоит отметить еще одну вещь относительно выравнивания и исправления ошибок. Задача исправления ошибок в технологии Ion Torrent состоит в том, чтобы найти ошибки типа вставки и удаления, вызывающие ложные ошибки типа замены. Все ошибки в чтениях определяются относительно некоторого эталонного генома. В качестве эталонного генома может быть взят геном, построенный ранее на этих же чтениях, или тот же организм, но собранный на других чтениях или другой организм той же таксономической группы. В последних двух случаях нужно отличать ОНП от ошибок секвенирования.

Для моделирования процесса чтения и ошибок используются скрытые марковские модели (Hidden Markov Model, далее HMM) [5]. Скрытыми состояниями являются события, происходящие при чтении (нуклеотид прочитан верно, произошла замена, вставка или удаление), наблюдаемой последовательностью — то, что прочитано. В статье [4] было предложено использовать HMM, построенную для строк из гомополимеров, и была предложена параметрическая модель для оценивания эмиссионных вероятностей. Но построенная HMM не учитывала то, что качество чтения падает к концу последовательности, что должно отразиться на эмиссионных вероятностях: они должны меняться со временем.

Структура работы:

- Глава 1: Hidden Markov Model для задачи выравнивания. В этой главе дается описание существующих методов применения HMM для выравнивая строк из нуклеотидов.
- Глава 2: Hidden Markov Model для задачи выравнивания в пространстве гомополимеров. В этой главе сделано обобщение HMM из предыдущей главы на пространство строк из гомополимеров, описаны особенности получаемой модели.
- Глава 3: Оценивание параметров. В этой главе описано построение процедуры оценки параметров для модели, полученной в Главе 2.

Глава 1

Hidden Markov Model для задачи выравнивания

1.1. Выравнивание

Пусть Σ — алфавит, Σ^+ — пространство строк над алфавитом. Пусть $\widetilde{\Sigma} = \Sigma \cup \prime-\prime$, где $\prime-\prime$ — специальный символ.

Обозначения:

- $s,\ r \in \Sigma^+$ — пара строк над алфавитом Σ^+;
- $s[i]$ — i-я буква строки s, $1 \leqslant i \leqslant |s|$;
- $[i, j]$, где $i \leqslant j$ — интервал целых чисел $\{i, i+1, \ldots, j\}$. Поэтому $s = s[0, |s| - 1]$.

Определение 1.1.1. *Под редакционным расстоянием будем понимать количество вставок символа $\prime-\prime$ в s и r, сделанных так, чтобы максимизировать количество позиций с совпадающими символами в получающихся строках. Обозначение: $\rho_{ed}(s,\ r)$*

Определение 1.1.2. *Выравниванием для $s,\ r \in \Sigma^+$ называется представление в виде $s',\ r' \in \widetilde{\Sigma}^+$ такое, что $\rho_{ed}(s', r')$ минимально. $s',\ r'$ определяются неоднозначно.*

Имея $s',\ r'$, будем считать, что $|s'| = |r'|$. Иначе, дополним меньшую строку символами $\prime-\prime$ для тех позиций, которые в ней не представлены.

Пример 1.1.1. $s = GAATTCA,\ r = GCATCGA$

```
GAATTC-A
|  ||  |  |
GCAT-CGA
```

Рис. 1.1. Выравнивание между $s = GAATTCA,\ r = GCATCGA$

1.1.1. Типы ошибок в выравнивании

Определение 1.1.3. *$s',\ r' \in \widetilde{\Sigma}^+,\ |s'| = |r'|$. Будем говорить, что на позиции i произошла ошибка, если $s'[i] \neq r'[i],\ 0 < i < |s'|$.*

Будем выделять три типа ошибок:

- Вставки (Insertion), если $r'[i] \neq$ ‘ $-$ ‘ при том, что $s'[i] =$ ‘ $-$ ‘;

```
Вставка на позиции 6:
position:     0  1  2  3  4  5  6  7
s':           G  A  A  T  T  C  -  A
r':           G  C  A  T  -  C  G  A
```

- Удаления (Deletion), если $r'[i] = \text{`}-\text{`}$ при том, что $s'[i] \neq \text{`}-\text{`}$;

```
Удаление на позиции 4:
position:     0  1  2  3  4  5  6  7
s':           G  A  A  T  T  C  -  A
r':           G  C  A  T  -  C  G  A
```

- Замены (Mismatch), если $r'[i] \neq s'[i]$, и $r'[i],\ s'[i] \neq \text{`}-\text{`}$.

```
Замена на позиции 1:
position:     0  1  2  3  4  5  6  7
s':           G  A  A  T  T  C  -  A
r':           G  C  A  T  -  C  G  A
```

Отметим, что ошибки несимметричны — вставка в s' является удалением в r' и наоборот. Здесь и далее будем считать, что ошибки происходят в r' относительно s'.

Обозначим совпадение позиций через M (Match). Тогда $\forall i : 0 \leq i \leq |s| - 1$, $s'[i]$ в зависимости от $r'[i]$ можно сопоставить одно из событий: $\{Match,\ Mismatch,\ Insertion,\ Deletion\} \Leftrightarrow \{M,\ D,\ I\}$.

1.2. Интерпретация задачи

Рассмотрим строку s над алфавитом Σ. Пусть происходит последовательное чтение строки s (с ошибками), в результате которого получается строка r. Процесс чтения можно описать с помощью выравнивания для пары строк s, r. И, наоборот, имея две строки, можно построить для них выравнивание, то есть попытаться воспроизвести те события, которые происходили в процессе чтения. Тем не менее, эти процессы не эквивалентны:

- Процедура чтения однозначно задает выравнивание;
- Для двух строк выравнивание задается неоднозначно. Его можно рассматривать как аппроксимацию процесса чтения.

Таким образом, возникает задача построения модели, имитирующей процесс чтения, т.е. производящей выравнивание двух строк. Причем, оно должно быть наиболее вероятным в заданных условиях.

Эта модель должна:

- Зависеть от параметров, характеризующих чтение;
- Давать оценку выравниванию;

- Строить для заданных строк наилучшее выравнивание.

В качестве такой модели можно использовать скрытые марковские модели [5] (Hidden Markov Model, далее HMM).

1.3. HMM: определение

Пусть Ω — некоторое множество. $(\Omega, \mathcal{F}, \mathcal{P})$ — вероятностное пространство, где $\mathcal{F}$ — сигма-алгебра на множестве подмножеств Ω, $\mathcal{P}$ — вероятность. Рассмотрим случайные величины $\xi_n : \Omega \to X$, $n \in \mathbb{N}_{\cup}\{0\}$, где $X = \{x_1, x_2, \ldots, x_k, \ldots\}$ — конечное или счетное множество.

Определение 1.3.1. *Последовательность случайных величин $\xi_0, \xi_1, \ldots, \xi_n, \ldots$ называется дискретной марковской цепью, если выполняется марковское свойство:*

$$\mathcal{P}(\xi_n = x_{i_n} | \xi_{n-1} = x_{i_{n-1}}, \ldots, \xi_0 = x_{i_0}) = \mathcal{P}(\xi_n = x_{i_n} | \xi_{n-1} = x_{i_{n-1}}),$$

где $x_{i_j} \in X,\ j = 0, 1, \ldots,\ n \in \mathbb{N}$, X — множество состояний марковской цепи.

Пусть $(K, \mathcal{A}, \mathcal{P}')$ — другое вероятностное пространство, $Y = \{y_1, y_2, \ldots, y_k, \ldots\}$ — конечное или счетное множество, $\eta_1,\ \eta_2, \ldots : K \to Y$ — независимые случайные величины.

Определение 1.3.2. *Семейство пар $\{\xi_t, \eta_t\}_{t=1,2,\ldots T}$, где $\{\xi_t\}$ — марковская цепь, $\{\eta_t\}$ — последовательность независимых случайных величин такая, что η_t зависит только от ξ_t, называется скрытой марковской моделью.*

Распределение скрытой марковской модели задается:

1. Вектором начального распределения $\pi = \{\pi_i\}$, где $\pi_i = p(\xi_0 = x_i)$;
2. Матрицей переходных вероятностей $P_{\xi_n|\xi_{n-1}} = \{p_{ij}^n\}$, где $p_{ij}^n = p(\xi_n = x_j | \xi_{n-1} = x_i)$;
3. Распределением вероятностей наблюдаемых переменных для скрытых состояний: $p_{\eta|\xi_i},\ i = 1, 2, \ldots$

1.4. HMM для задачи выравнивания

Рассмотрим $s,\ r \in \Sigma^+$ — две строки над Σ, $s',\ r' \in \widetilde{\Sigma}^+$ — соответствующее им выравнивание. Будем моделировать процесс чтения — скрытую марковскую цепь. Определим скрытые состояния как $X = \{B, E, M_i, D_i, I_i\},\ i \in \mathbb{N}$. $\xi : \Omega \to X$ — случайные величины, образующие марковскую цепь.

Скрытые состояния можно интерпретировать как события, происходящие в процессе чтения строки s. А именно:

- M_i: $s[i]$ прочитан верно или заменен на другой элемент Σ;
- I_i: после прочтения $s[i]$ в r произошла вставка элемента из Σ;
- D_i: произошло удаление элемента $s[i]$;
- $Begin, End$ $(B,\ E)$ — скрытые состояния для обозначения начала и конца выравнивания.

$B,\ E,\ D$ — такие состояния, что для них $p(\eta\ =\ '-') = 1$.

Наблюдаемыми результатами будут $Y = \Sigma,\ \eta : \Omega \to Y$ — элементы пространства гомополимеров.

1.4.1. Конструкция HMM

Полученная скрытая марковская модель задается:

1. Вектором начального распределения: p_{ξ_0}, где $p(\pi_0 = B) = 1,\ p(\pi_0 \neq B) = 0$;
2. Матрицей переходных вероятностей: $P = \{p_{ij}\}$, где $p_{ij} = p(\xi_n = x_j | \xi_{n-1} = x_i)$. Причем, не все переходы возможны;
3. Распределением вероятностей наблюдаемых переменных для скрытых состояний: $P_{\eta|\xi} = p(\eta|\xi),\ i = 1, 2, \ldots$

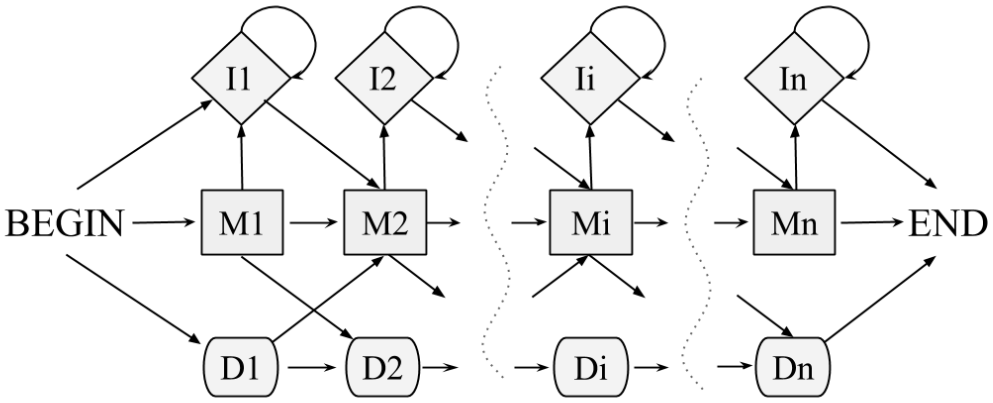

Рис. 1.2. HMM для выравнивания

Особенности полученной HMM для выравнивания:

- Марковская цепь будет однородной,. Это достигается за счет расширения множества скрытых состояний. Можно было бы определить $X = \{B, E, M, I, D\}$, и тогда МЦ была бы неоднородной — в каждый момент времени матрица переходов в скрытые состояния была бы новой. Это отражает то, что качество чтения падает к концу. Но так не получится сделать, так как нельзя моделировать ситуации вставки. Почему - в силу определения скрытых состояний — они определяются для $s[i]$, вставке соответствует пустой гомополимер, не представленный в s;
- В матрице переходных вероятностей возникают ограничения. Так, например, запрещены переходы:
  - $M_i, I_i, D_i \to M_{i-j}, I_{i-j}, D_{i-j},\ 0 < j < i$;
  - $D_i \to I_j,\ I_i \to D_j, \forall i,\ j \in \mathbb{N}$, поскольку эти переходы эквивалентны состоянию M (замены элемента из s).

  Эти ограничения являются естественными и необходимы потому, что конструируемая марковская цепь является аналогией процесса выравнивания, которое в каждый момент времени продвигается вперед по строкам s, r и не возвращается в предыдущие состояния;

- В матрице переходов ненулевыми будут только диагональные элементы и элементы, находящиеся около диагонали.

1.4.2. HMM: решаемые задачи

Обозначим за Θ совокупность параметров, определяющих HMM, π — скрытую последовательность состояний.

Есть три основные типа задач, связанные с HMM [6], [5]:

1. Даны параметры модели, скрытая и наблюдаемая последовательности. Требуется вычислить вероятность появления наблюдаемой последовательности.

   В контексте данной задачи: имея s, r, π, Θ, оцениваем вероятность данного выравнивания, т.е. $p(\pi|(s,r),\Theta)$. Таким образом, можем сравнивать пару выравниваний между собой. Будем считать лучшим то, что имеет наибольшую вероятность появления.

2. Даны параметры модели и наблюдаемая последовательность. Требуется найти наиболее вероятную последовательность скрытых состояний.

   Имея s, r, Θ и HMM, имитируется процесс чтения и строится наиболее вероятное выравнивание π, т.е. $p(\pi|(s,r),\Theta) = \max\limits_{\pi'} p(\pi'|(s,r),\Theta)$. Для этого используется алгоритм Витерби.

3. Дан набор наблюдаемых последовательностей. Требуется найти параметры для HMM такие, что вероятность появления данного набора будет максимальной.

   То есть, пусть $D = \{(s^d, r^d)\}_{d=1}^N$ — набор пар строк, задача — найти такой набор параметров Θ^*, что правдоподобие D: $p(D|\Theta^*) = \sum\limits_{d=1}^{N} p((s^d, r^d)|\Theta^*)$ будет максимально, т.е. $p(D|\Theta^*) = \max\limits_{\Theta} p(D|\Theta)$.

Глава 2

Hidden Markov Model для задачи выравнивания в пространстве гомополимеров

2.1. Гомополимеры

Определение 2.1.1. *Гомополимер — максимальная последовательность одинаковых символов в строке, идущих подряд. Обозначение:* $AAA \to \langle A, 3\rangle$.

Для любой строки$s \in \Sigma^+$ существует эквивалентное единственное представление s^h, где s^h — последовательность гомополимеров.

Пример 2.1.1. $s = AAAGCTTGG \Leftrightarrow s^h = \langle A,3\rangle\langle G,1\rangle\langle C,1\rangle\langle T,2\rangle\langle G,2\rangle$.

$\mathbb{H} = \{\langle\alpha, k\rangle | \alpha \in \Sigma,\ k \in [1 \dots 15]\}$ — пространство гомополимеров, $\widetilde{\mathbb{H}} = \mathbb{H} \cup \langle'-', 0\rangle$.

2.2. Специфика технологии Ion Torrent

В процессе чтения строка S разбивается на подстроки длиной до 600 символов, и происходит последовательное чтение каждой строки по отдельности. Особенности Ion Torrent [1], [2]:

- Чтение идет не побуквенно, а по гомополимерам. То есть, на каждом шаге чтения определяется, какой символ из Σ следует далее в строке, и определяется длина последовательности, состоящей из этого символа;
- Длина гомополимера определяется следующим образом. С оборудования приходит сигнал, пропорциональный длине читаемого гомополимера. Величина сигнала может быть нецелой, поэтому она определенным образом округляется до целого, и полученное значение считается длиной прочтенного гомополимера.

Проблемы, возникающие у чтений Ion Torrent [3]:

- С ростом длины гомополимера хуже распознается его длина. Например, гомополимер длиной 5 читается верно только в 98% случаев;
- Гомополимер длиной более 15 невозможно прочитать верно;
- Качество чтения падает к концу последовательности: есть тенденция к увеличению количества ошибок с приближением к концу чтения.

Применяя HMM для имитации процесса чтения технологии Ion Torrent, нужно учесть его особенности: определить строки над другим пространством и изменить эмисионные вероятности.

2.3. HMM для строк из $\mathbb{H}$

$\mathbb{H} = \{\langle\alpha, k\rangle | \alpha \in \Sigma,\ k \in [1 \dots 15],\ k \in \mathbb{N}\}$ — пространство гомополимеров. $\widetilde{\mathbb{H}} = \mathbb{H} \cup \langle'-', 0\rangle$.

$\widetilde{\mathbb{H}} = \{\langle\alpha, k\rangle | \alpha \in \Sigma,\ k \in (0 \dots 15],\ k \in \mathbb{R}\} \cup \langle'-', 0\rangle$ — пространство, соответствующее сигналу.

Пусть $s,\ r \in \mathbb{H}^+,\ s = s_1 \dots s_{|s|},\ r = r_1 \dots r_{|r|},\ s_i,\ r_j \in \mathbb{H}, 1 \leqslant i \leqslant |s|,\ 1 \leqslant j \leqslant |r|$ — две строки над пространством гомополимеров. $s',\ r' \in \widetilde{\mathbb{H}}^+$ — соответствующее им выравнивание.

2.3.1. Состояния HMM

Скрытые состояния: $X = \{B, E, M_i, D_i, I_i\},\ i \in \mathbb{N}$. $\xi : \Omega \to X$ — случайные величины, образующие марковскую цепь.

Интерпретация состояний:

- M_i: $s[i] = \langle\alpha, k\rangle,\ r[j] = \langle\alpha, l\rangle,\ k,\ l > 0,\ \alpha \in \Sigma$. То есть, M соответствует состоянию, когда основание гомополимера прочитано верно (полученная длина может отличаться от исходной);
- I_i: после прочтения $s[i]$ в r произошла вставка элемента из $\mathbb{H}$;
- D_i: произошло удаление элемента $s[i]$;
- $Begin, End$ $(B,\ E)$ — скрытые состояния для обозначения начала и конца выравнивания.

$B,\ E,\ D$ — такие состояния, что для них $p(\eta = \langle'-', 0\rangle) = 1$.

В данной модели нет скрытого состояния, обозначающего замену гомополимера. Это объясняется тем, что для Ion Torrent характерны ошибки вставки-удаления (которые здесь учитываются), а не ошибки замены (которыми можно пренебречь).

Наблюдаемые состояния: $Y = \mathbb{H},\ \eta : \Omega \to Y$.

2.3.2. Конструкция HMM

Полученная скрытая марковская модель задается:

1. Вектором начального распределения:

   p_{ξ_0}, где $p(\pi_0 = B) = 1,\ p(\pi_0 \neq B) = 0$;

2. Матрицей переходных вероятностей:

   $P = \{p_{ij}\}$, где $p_{ij} = p(\xi_n = x_j | \xi_{n-1} = x_i)$. Причем, не все переходы возможны.

3. Распределением вероятностей наблюдаемых переменных для скрытых состояний: $P_{\eta|\xi} = p(\eta|\xi)$.

Особенности HMM для имитации чтения строк из $\mathbb{H}$:

- Отсутствие скрытого состояния, соответствующего замене. Теперь M означает только Match (правильное прочтение символа гомополимера, но длина может быть определена неверно);
- Изменение эмиссионных вероятностей. В случае строк над Σ наблюдается только элемент Σ, т.е. нужно знать вероятность для $|\Sigma|$ исходов. В случае гомополимера нужно знать не только символ, но и то, сколько таких символов шло в строке подряд;

```
s' = <A,1><C,1><G,1><T,1>
r' = <A,1><C,1><-,0><T,2>
```

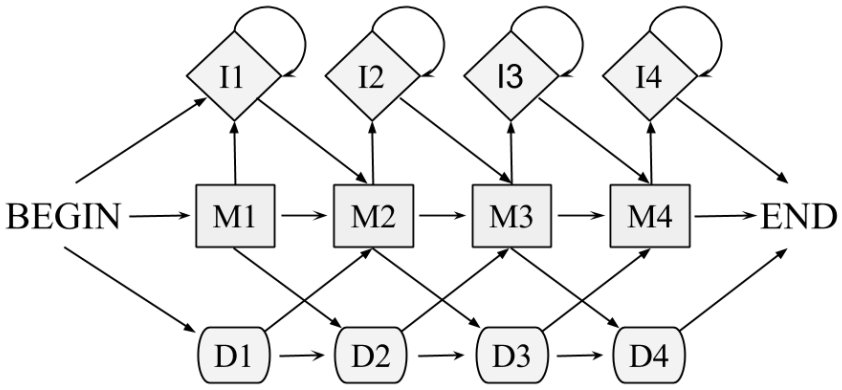

Рис. 2.1. HMM для выравнивания строк из $\mathbb{H}^+$

- Запрет части переходов, таких как:
  - $M_i, I_i, D_i \to M_{i-j}, I_{i-j}, D_{i-j},\ 0 < j < i$;
  - $D_i \to I_j,\ I_i \to D_j, \forall i,\ j \in \mathbb{N}$, поскольку эти переходы эквивалентны состоянию замены, которым пренебрегаем в данной модели.

Полученная скрытая марковская модель решает те же задачи, что и HMM для строк из Σ.

2.4. Упрощение полученной HMM

2.4.1. Мотивация

Посчитаем количество параметров, которые нужно оценить для применения сконструированной модели. Пусть $|s| \sim 400$.

1. Вектор начального распределения: поскольку введено специальное скрытое состояние *Begin* для обозначения начала выравнивания, то $p(\xi_0 = B) = 1$. То есть, вектор начального распределения оценивать не нужно;

2. Матрица переходных вероятностей P. В силу того, что большинство переходов запрещены, ненулевых элементов в матрице будет примерно $9 \cdot |s| = 3600$;

3. Распределение вероятностей для наблюдаемых гомополимеров. Пусть наблюдается $\langle \beta, k \rangle,\ k > 0,\ \beta \in \Sigma$.

   $p_{\eta|\xi} = p(\eta = \langle \beta, k \rangle \mid \xi) = p(k|l, \alpha, \xi_n) p(\beta|\alpha, \xi)$.

   Тогда:

- $\xi_n = D_n$: $p(\eta = \langle{}'-', \ 0\rangle | D_n) = 1$, т.о. нет параметров для оценки;
- $\xi_n = I_n$: $p_{\eta|I_n} = p(k|0,'-', I_n)p(\beta|'-', I_i)$, где $p(\beta|'-', I_i)$ — вероятность того, что произошла вставка элемента $\beta \in \Sigma$ ($|\Sigma| = 4$ значений для оценки), $p(k|0,'-', I_n)$ — вероятность, что вставилось k таких символов ($k = 15$ значений для оценки). То есть, всего 19 значений;
- $\xi_n = M_n$: $f_{\eta|M_n} = p(k|l, \alpha, M_n)I_{\alpha=\beta}$, где $p(k|l, \alpha, M_n)$ — вероятность наблюдать гомополимер длины k при том, что в исходной последовательности был гомополимер длины l ($15 \cdot 15 = 225$ значений).

То есть, $\forall i : 1 \leq i \leq |s|$ необходимо оценить $19 + 225 = 244$ параметров.

Всего получается $3600 + 244 \cdot 400 > 100000$ — слишком параметров для оценки.

Кроме того, хочется учесть процесс определения длины в технологии Ion Torrent — то, что он включает в себя получение сигнала и его последующую интерпретацию.

2.4.2. Парамерическая модель для определения длин гомополимеров

Пусть $s[i] = \langle\alpha, l\rangle \in \mathbb{H}$. Тогда распределение вероятностей наблюдения $\langle\beta, k\rangle \in \mathbb{H}$:

$$p_{\eta|\xi} = p(\eta = \langle\beta, k\rangle \,|\, \xi) = p(k|l, \alpha, \xi)p(\beta|\alpha, \xi_i).$$

В статье [4] предложена двухуровневая параметрическая модель для подсчета $p(k|l, \alpha, \pi), \ \pi \in X$. На первом этапе определяется интенсивность сигнала из длины l (в случае, когда эта информация дана со входными данными, эту стадию можно опускать), на втором — длина k на основе полученной интенсивности: $l \to f \to k$.

Первый этап — моделирование величины сигнала, $p(f|l)$:

Возможны два случая:

1. $l = 0$, т.е. $\xi = I_i$. Для моделирования распределения длины наблюдаемого гомополимера при вставке используется лог-нормальное распределение с $\mu = 0$:

$$p(f|0) = \frac{1}{f\sigma\sqrt{2\pi}} e^{-\frac{(\ln f)^2}{2\sigma^2}}.$$

Тогда нужно оценить только параметр распределения σ.

2. $l \neq 0$, т.е. $\xi = M_i$. . Тогда $k, \ l > 0$. В качестве модели для $p(f|l)$ используется распределение Лапласа:

$$p(f|l) = \frac{1}{2b_l} e^{-\frac{|f-l|}{b_l}}.$$

Где b — степенная функция от l: $b = c_0 + c_1 \times l^{c_2}$, а c_0, c_1, c_2 — константы, удовлетворяющие уравнению: $\int_f p(f|l) = 1$.

Тогда нужно оценить константы $c_0, \ c_1, \ c_2$. Можно было бы оценивать $b_l, \ l = 1, \ldots, 15$, но из-за того, что не все пары $(l, \ k)$ встречаются достаточное количество раз, берутся только часто встречающиеся значения, вычисляются $c_0, \ c_1, \ c_2$, и затем вычисляются b_l.

Второй этап — моделирование итоговой длины при данном сигнале $p(k|f)$:

Запишем, что такое $p(k|f)$:

$$p(k|f) = \frac{p(k,f)}{p(f)} = \frac{p(f|k)p(k)}{p(f)}.$$

Нужно, чтобы $\sum\limits_{k'} p(k'|f) = 1$, поэтому поделим на нормировочную константу $Z = \sum\limits_{k'} p(f|k')p(k')$.

Тогда, вычисление $p(k|f)$ производится следующим образом:

$$p(k|f) = \frac{1}{Z}p(f|k)p(k), \tag{2.1}$$

где:

- $p(k)$ — распределение длин гомополимеров в строке S;
- $p(f|k)$ — правдоподобность наблюдения данного сигнала f при наблюдаемой длине k;
- $Z = \sum\limits_{k'} p(f|k')p(k')$ — нормирующая величина.

В случае, когда есть информация о сигнале, достаточно формулы (2.1). В ином случае, будем использовать формулу:

$$p(k|l) = \int\limits_{f} p(k|f)p(f|l).$$

Глава 3

Оценивание параметров

Для простоты опишем алгоритмы для оценки параметров сначала для обычной HMM [5], а затем укажем то, как они модифицированы для HMM, использующейся в этой работе.

3.1. Общий случай

Обозначения:

1. $k = 1, 2, \ldots$ — скрытые состояния;
2. Θ^0 — набор начальных параметров модели;
3. Θ^t — набор параметров для модели на шаге t.
4. $\pi = \{\pi_1, \pi_2, \ldots, \pi_n, \}$ — последовательность скрытых состояний;
5. $x = \{x_1, x_2, \ldots, x_n\}$ — последовательность наблюдений;
6. $A = a_{kl}$ — матрица перехода, $a_{kl} = p(\pi_i = l | \pi_{i-1} = k)$ — вероятность перехода между состояниями k, l;
7. $e_k = p(x_i = b | \pi_i = k)$ — эмиссионные вероятности для скрытого состояния k;
8. $p(x, \pi|\Theta)$ — вероятность наблюдаемой последовательности для заданной последовательности скрытых состояний;
9. $p(x)$ — вероятность наблюдаемой последовательности.

Для подсчета $p(x)$ определяются две переменные: $f_k(i)$ — вероятность находиться в k-ом скрытом состоянии в момент времени i, когда наблюдаются первые i элементов последовательности:

$$f_k(i) = p(x_1, x_2, \ldots, x_i, \pi_i = k), \tag{3.1}$$

и $b_k(i)$ — вероятность наблюдать элементы последовательности, начиная с $i-$го, при условии, что $i-$тым скрытым состоянием является k:

$$b_k(i) = p(x_{i+1}, \ldots, x_n | \pi_i = k). \tag{3.2}$$

Значения переменных находятся рекурсивно. Инициализация:

$$f_0(0) = 1, \ f_k(0) = 0, k > 0,$$
$$b_k(n) = a_{k0}, \ \forall k.$$

Формулы для подсчета $f_k(i)$, $b_k(i)$:

$$f_l(i) = e_l(x_i) \cdot \sum_k f_k(i-1)a_{kl}$$

$$b_k(i) = \sum_l a_{kl} e_l(x_{i+1}) b_l(i+1).$$

Тогда, $p(x) = \sum_k f_k(n)a_{k0} = \sum_l a_{0l} e_l(x_1) b_l(1)$.

3.1.1. Алгоритм Витерби, общая идея

Пусть известна последовательность наблюдаемых значений x и параметры модели Θ. Алгоритм состоит в поиске скрытой последовательности состояний, наилучшим образом описывающей наблюдаемые значения. То есть, нужно найти $\hat{\pi}$ такое, что:

$$\hat{\pi} = \underset{\pi}{\operatorname{argmax}} p(x, \pi|\Theta).$$

Эту задачу можно сформулировать как поиск оптимального пути на решетке из всех возможных скрытых состояний для каждого момента времени. Поиск производится рекуррентно, то есть, на каждом шаге оптимальный путь высчитывается на основании данных, полученных с предыдущего шага. Обозначим $v_k(i)$ — вероятность для самого вероятного пути, заканчивающегося в скрытом состоянии k, при том, что наблюдается i.

Инициализация:

$$v_0(0) = 1,\ v_k(0) = 0,\ \forall k > 0.$$

Формула для пересчета:

$$v_l(i+1) = e_l(x_{i+1}) \max_k \{v_k(i)a_{kl}\}.$$

Запоминая на каждом шаге состояние, для которого значение вероятности было наибольшим, можно получить наиболее вероятную последовательность скрытых состояний.

3.1.2. Алгоритм Баума-Уэлча, общая идея

Этот алгоритм (частный случай EM-алгоритма) используется для оценки параметров модели. Пусть $D = \{x^i\}_{i=1}^N$ — набор последовательностей наблюдений, $\Pi = \{\pi^i\}_{i=1}^N$ — соответствующие им последовательности скрытых состояний. Задача — найти такой набор параметров Θ^*, что правдоподобие набора:

$$p(D|\Theta^*) = \sum_{i=1}^N p(x^i|\Theta^*),$$

будет максимально, т.е. $p(D|\Theta^*) = \max_\Theta p(D|\Theta)$.

За A_{kl} обозначим количество раз, которое переход $k \to l$ возник в Π, за $E_k(b)$ — сколько раз для состояния k наблюдалось b.

Тогда параметры модели можно посчитать как:

$$a_{kl} = \frac{A_{kl}}{\sum_i A_{ki}}, \tag{3.3}$$

$$e_k(b) = \frac{E_k(b)}{\sum\limits_c E_k(c)}. \tag{3.4}$$

Пусть считаем, что выравнивание неизвестно и знаем только $\{x^i\}_{i=1}^N$. Пусть A_{kl} — ожидаемое количество раз, которое переход $k \to l$ возник в Π, $E_k(b)$ - ожидаемое число наблюдений b из k.

Алгоритм:

- **Expectation step**

  1. Для каждого $x \in D$ с помощью алгоритма Витерби строится оптимальная последовательность скрытых состояний;
  2. Подсчитываются A_{kl}, $E_k(b)$ с использованием формул из forward-backward алгоритма:

$$A_{kl} = \sum_{j=1}^{N} \sum_{i=1}^{n-1} p(\pi_i = k, \pi_{i+1} = l | x^j, \Theta) = \tag{3.5}$$

$$= \sum_{j=1}^{N} \frac{1}{p(x^j)} \sum_{i=1}^{n-1} f_k^j(i) a_{kl} e_l(x_{i+1}^j) b_l^j(i+1), \tag{3.6}$$

- **Maximization step** На этом шаге обновляются параметры модели, то есть подсчитываются a_{kl}, $e_k(b)$ по формулам (3.3), (3.4). в полученной модели определяется правдоподобие D.

Особенностями алгоритма является то, что это итеративная процедура. На каждом шаге правдоподобие модели не уменьшается, алгоритм сходится к максимуму (возможно, локальному). У функции правдоподобия может быть несколько локальных максимумов, куда алгоритм может сойтись. В случае, когда изначальные параметры модели не заданы, его можно запустить несколько раз с разными параметрами.

3.2. Оценивание параметров для HMM в пространстве гомополимеров

Обозначения:

1. s, r — строки из $\mathbb{H}^+$, $s', r' \in \widetilde{\mathbb{H}}^+$;
2. $X = \{B, E, M_i, D_i, I_i\}$, $i \in \mathbb{N}$ — множество скрытых состояний, $\pi \in X$ — скрытые состояния;
3. Θ^0 — набор начальных параметров модели;
4. Θ^t — набор параметров для модели на шаге t.
5. $\Pi = \{\pi_1, \pi_2, \ldots, \pi_n, \}$ — последовательность скрытых состояний, выравнивание для s, r;
6. $A = a_{\pi_k, \pi_l}$ — матрица перехода, $a_{\pi_k \pi_l} = p(\pi_k | \pi_l)$ — вероятность перехода между состояниями $\pi_k, \pi_l \in X$;
7. $p_{\eta|\xi} = p(\eta = \langle \beta, k \rangle \,|\, \xi) = p(\beta|\alpha, \xi) p(k|l, \alpha, \xi)$ — эмиссионные вероятности;
8. $p(s, r|\Theta)$ — вероятность прочесть строку r из s;

Пусть дан тренировочный набор $D = \{(s^d, r^d)\}_{d=1}^N$. Считаем, что все пары в наборе независимы друг от друга.

3.2.1. Специфика HMM для гомополимеров

До этого говорилось, что для $s, r \in \mathbb{H}^+$ существует единственное эквивалентное представление в Σ^+, и наборот. Строки над пространством гомополимеров рассматриваются потому, что чтение в технологии Ion Torrent происходит по гомополимерам, что позволяет рассматривать исходную строку как строку над $\mathbb{H}$. Моделируя процесс чтения, между гомоплимерами строки может быть вставлен гомополимер нулевой длины $\langle '-', 0 \rangle$.

Но для строки чтения нельзя сразу рассматривать ее представление в пространстве гомополимеров. Рассмотрим пример, иллюстрирующий причину:

Пример 3.2.1. *Пусть* $s = \langle A, 2\rangle\langle C, 1\rangle\langle A, 1\rangle \ = AAACA.$

Пусть $\pi = (M, D, M) \Rightarrow r = \langle A, 2\rangle\langle '-', 0\rangle\langle A, 1\rangle = \langle A, 3\rangle \ = AAAA.$

Рассматривая r над $\mathbb{H}$, даже с помощью построенной модели не сможем восстановить правильное выравнивание. Поэтому в описанных ниже алгоритмах r рассматривается над Σ^+.

3.2.2. Алгоритм Витерби

Определим переменную $V_j^{\pi}(i)$ как вероятность оптимального выравнивания между префиксами $r_{[1:i]}$ и $s_{[1:j]}$, заканчивающегося в скрытом состоянии π при условии, что суффиксом $r_{[1:i]}$ является элемент длины k.

$$V_j^M(i) = e_{M_j}(i) \cdot \begin{cases} V_{j-1}^M(i-1) \cdot a_{M_{j-1}M_j}; \\ V_{j-1}^I(i-1) \cdot a_{I_{j-1}M_j}; \\ V_{j-1}^D(i-1) \cdot a_{D_{j-1}M_j}. \end{cases}$$

$$V_j^I(i) = e_{I_j}(i) \cdot \begin{cases} V_j^M(i-1) \cdot a_{M_jI_j}; \\ V_j^I(i-1) \cdot a_{I_jI_j}; \\ V_j^D(i-1) \cdot a_{D_jI_j}. \end{cases}$$

$$V_j^D(i) = \begin{cases} V_{j-1}^M(i) \cdot a_{M_{j-1}D_j}; \\ V_{j-1}^I(i) \cdot a_{I_{j-1}D_j}; \\ V_{j-1}^D(i) \cdot a_{D_{j-1}D_j}. \end{cases}$$

Инициализация переменных для нулевого шага:

$$V_0^B(0) = 1, \ \ V_0^{\pi}(0) = 0, \forall \pi \neq B,$$
$$V_0^D(0) = 0, \ V_j^M(0) = 0 \ j > 0,$$
$$V_0^M(i > 0) = 0.$$

3.2.3. Forward-Backward алгоритм

Пусть $s \in \mathbb{H}^+, r \in \Sigma^+$.

Forward-переменная $F(i,j,k,\pi)$ — это сумма вероятностей по всем возможным выравниваниям между префиксами $r[1:i],\ s[1:j]$, в которых гомополимер $\langle\beta,k\rangle$, заканчивающийся в r на позиции i, выровнен в гомополимер $\langle\alpha,l\rangle$ последовательности s на позиции j.

$$F(i,j,k,\pi) = \sum_{\pi'} F(i-k, j-I, \pi')p(\pi|\pi')p(\beta|\alpha,\pi)p(k|l,\beta,\pi);$$
$$F(i,j,\pi) = \sum_{k,l} F(i,j,k,l,\pi).$$

Распишем вид для каждого π.

$$F(i,j,k,l,\pi) = \begin{cases} \sum\limits_{\pi'} F(i-k,j-1,\pi')p(\pi|\pi')I_{\beta=\alpha}p(k|l,\beta,\pi),\ \pi = Match; \\ \sum\limits_{\pi'} F(i-k,j,\pi')p(\pi|\pi')p(\beta\,|\,'-',\pi)p(k|l,\beta,\pi),\ \pi = Insertion; \\ \sum\limits_{\pi'} F(i,j-1,\pi')p(\pi|\pi'),\ \pi = Deletion. \end{cases}$$

Как и в общем случае,

$$p(s,r) = \sum_{\pi} F(|r|,|s|,\pi).$$

Backward-переменная: вероятность, просуммированная по всем возможным выравниваниям между суффиксами $r[i+1:n],\ s[j+1:n]$, начинающимся в скрытом состоянии π, в котором гомополимер длины k, заканчивающийся в r на позиции i, выровнен в гомополимер длины l последовательности s на позиции j. Пусть $s[j+1] = \langle\alpha_{j+dj}, dj\rangle,\ r[i+1,\ i+di] = \langle\beta_{i+di}, di\rangle$.

$$B(i,j,k,l,\pi) = \sum_{\pi',di,dj} p(\pi'|\pi)p(\beta_{i+di}|\alpha_{j+dj},\pi')p(di|dj,\beta_{i+di},\pi')B(i+di, j+I_{dj>0}, di, dj, \pi'), \tag{3.7}$$

$$0 \le di \le k_{max},\ 0 \le dj \le 1,$$

где k_{max} — максимальная длина гомополимера, начинающегося в r на позиции $i+1$.

Для разных скрытых состояний в (3.7) могут отсутствовать некоторые множители. Удобно написать представление переменной для каждого скрытого состояния:

$$B(i,j,k,l,\pi) = \sum_{\pi'} \begin{cases} B(i+di, j+1, di, dj, \pi')p(\pi'|\pi)p(\beta|\alpha,\pi')p(k|l,\beta,\pi'),\ \pi' = M_{j+1},\ \alpha,\ \beta \in \Sigma; \\ B(i+di, j, di, 0, \pi')p(\pi'|\pi)p(\beta\,|\,'-',\pi')p(k|l,\beta,\pi'),\ \pi' = I_j,\ \beta \in \Sigma; \\ B(i, j+1, 0, dj, \pi')p(\pi'|\pi),\ \pi' = D_{j+1},\ \alpha \in \Sigma. \end{cases}$$

Причем для $B(i,j,k,l,\pi)$:

1. Если $\pi = M_j$, то $k,\ l \neq 0$;
2. Если $\pi = I_j$, то $l = 0, k \neq 0$;
3. Если $\pi = D_j$, то $k = 0,\ l \neq 0$.

3.2.4. Алгоритм Баума-Уэлча

Пусть $r \in \Sigma^+,\ s \in \mathbb{H}^+$.

Определим две переменные:

1. $\gamma_{i,j,k,l,\pi}$: вероятность того, что гомополимер $r[i, i+k]$ выровнен на гомополимер $s[j]$ длиной l в скрытом состоянии π;

2. $\zeta_{i,j,k,l,\pi,\pi'}$: вероятность того, что гомополимер $r[i, i+k]$ выровнен на гомополимер $s[j]$ длиной l в скрытом состоянии π, при том, что предыдущим скрытым состоянием было π'.

- **Expectation step**

  Для каждого $(s^d, r^d) \in D$ вычисляются $\gamma^d_{i,j,k,l,\pi}$, $\zeta^d_{i,j,k,l,\pi,\pi'}$ с помощью forward-backward алгоритма:

  $$\gamma^d_{i,j,k,l,\pi} = \frac{F^d(i,j,k,l,\pi)B^d(i,j,k,l,\pi)}{p(s^d, r^d)};$$
  $$\zeta^d_{i,j,k,l,\pi,\pi'} = \frac{\sum_{k',l'} F^d(i-k, j-I_{l>0}, k, l', \pi)p(\pi|\pi')p(\beta_i|\alpha_j, \pi)p(k|l, \beta_i, \pi)B^d(i,j,k,l,\pi)}{p(s^d, r^d)};$$
  $$1 \le d \le N,\ 1 \le i \le |r^d|, 1 \le j \le |s^d|.$$

- **Maximization step**

  На этом шаге обновляются параметры модели. Для этого заполняются три таблицы:

  $$T_{\pi,\pi'} = \sum_{i,j,k,l,d} \zeta^d_{i,j,k,l,\pi,\pi'};$$
  $$L_{k,l} = \sum_{i,j,k',l',d,\pi} \gamma^d_{i,j,k',l',\pi}\delta(k,k')\delta(l,l');$$
  $$B_\alpha = \sum_{i,j,k,d,\pi} \gamma^d_{i,j,k,0,\pi}\delta(r[i-k,i],\alpha);$$

$p(\alpha|'-', \pi')$ соответствует отнормированный вектор B, матрице переходных вероятностей — отнормированная по строкам матрица $T_{\pi,\pi'}$.

Для обновления параметров, соответствующих $p(k|l, \beta, \pi')$, нужно обновить параметры для логнормального распределения и распределения Лапласа, что делается с использованием EM-алгоритма.

Список литературы

1. Ion torrent community. — URL: https://www.lifetechnologies.com/ru/ru/home/brands/ion-torrent.html.
2. Ion torrent technology. — URL: http://www.lifetechnologies.com/ru/ru/home/life-science/sequencing/next-generation-sequencing/ion-torrent-next-generation-sequencing-technology.html.
3. Problems in Ion Torrent technology. — URL: http://mendel.iontorrent.com/ion-docs/Technical-Note---TMAP-Alignment_9012907.html.
4. Feng Z., Rui J., Ting C. PyroHMMsnp: an SNP caller for Ion Torrent and 454 sequencing data // Nucleic Acids Research. — 2013. — Vol. 41, no. 13.
5. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids / R. Durbin, Sean R. E., A. Krogh, G. Mitchison. — Cambridge University Press, 1998.
6. Lawrence R. A tutorial on hidden markov models and selected applications in speech recognition // IEEE. — 1989. — February. — Vol. 77, no. 2.
7. Mann T. P. Numerically stable hidden markov model implementation. — 2006.
8. Merriman B., Rothenberg J. M. Progress in Ion Torrent semiconductor chip based sequencing // Electrophoresis. — 2012. — Vol. 33, no. 23. — P. 3397—3417.
9. TMAP. — URL: https://github.com/nh13/TMAP.