

Правительство Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Санкт-Петербургский государственный университет»

Кафедра Системного Программирования

Атаманова Анна Михайловна

Скрытые Марковские модели переменного порядка для анализа данных ChIP-seq

Бакалаврская работа

Допущена к защите.
Зав. кафедрой:
д. ф.-м. н., профессор Терехов А. Н.

Научный руководитель:
д. ф.-м. н., профессор Терехов А. Н.

Рецензент:

Санкт-Петербург
2015

SAINT-PETERSBURG STATE UNIVERSITY

Chair of Software Engineering

Anna Atamanova

Variable-length hidden Markov models for ChIP-seq data analysis

Bachelor's Thesis

Admitted for defence.

Head of the chair:
professor Andrey Terekhov

Scientific supervisor:
professor Andrey Terekhov

Reviewer:

Saint-Petersburg
2015

Оглавление

Введение	4
1. Постановка задачи	6
2. Обзор существующих решений	7
2.1. Основные понятия и определения	7
2.2. Скрытые Марковские модели	9
2.3. Обучение модели СММПП	10
2.4. Обучение на нескольких выборках	13
2.5. Сравнение	14
3. Реализация	15
4. Применение	16
4.1. Применение к симмулированным данным	16
4.1.1. Смесь	16
4.1.2. СММ	17
4.1.3. Более интересный случай, СММПП	18
4.2. Применение к реальным данным	18
Заключение	21
Список литературы	22

Введение

Предметная область

Наш организм состоит из огромного количества клеток. Клетки постоянно движутся, что-то строят, что-то разрушают. Вся жизнь наша заключается в их функционировании. Одна из интереснейших частей клетки — это ее память, ДНК (дезоксирибонуклеиновая кислота), которая хранит в себе просто невероятное количество информации, в том числе «рецепты» построения необходимых веществ. Своеобразным строительным материалом клетки является белок. Белок также выполняет структурные, сигнальные, механические и другие функции. Соединения ДНК с конкретным белком могут играть роль в структуре ДНК, во внутренних механизмах управления клетки. Поэтому изучение ДНК-белковых взаимодействий крайне важно и актуально.

Однако перед самым изучением взаимодействий, необходимо обнаружить/распознать места, где они случились.

Данная работа посвящена изучению нахождения позиций связывания конкретного белка и ДНК, то есть нахождения позиций ДНК-белковых взаимодействий при заранее выбранном белке.

ChIP-seq

ChIP-seq (chromatin immunoprecipitation sequencing) — биологический эксперимент, позволяющий получить информацию о наличии или отсутствии взаимодействия ДНК с заданным белком вдоль всего генома.

Утрировано эксперимент можно описать следующим образом. Сначала, в клетки заливается специальный раствор, который приклеивает белки к ДНК. Потом, с помощью ультразвука, ДНК разрезаются на более мелкие фрагменты. Далее специальным антителом, подобранным к данному белку, вылавливаются те фрагменты, которые были связаны с исследуемым белком. Затем специальный прибор — секвенатор считывает концы фрагментов (целый фрагмент слишком велик для считывания). Считанный кусок фрагмента называется прочтением или ридом. Так продолжается, пока каждый фрагмент не будет с высокой вероятностью считан несколько раз.

Далее для каждого полученного рида ищется соответствующий ему участок последовательности генома (рис. 1). Обычно риды, которым может соответствовать более одного участка в геноме, исключают из рассмотрения.

Результаты эксперимента представляют в виде вектора длины генома, в котором стоит 1, если в соответствующей позиции генома начиналось хотя бы одно прочтение и 0 в обратном случае.

Однако, белок мог находиться не в самом начале фрагмента, и, кроме того, соединение белка с ДНК происходит не точно, а на некотором участке ДНК. По этому,

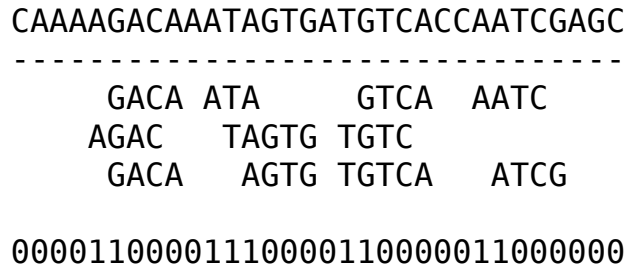


Рис. 1: Схематическое изображение выравнивания прочтений секвенатора (под чертой) на известную последовательность генома (над чертой).

для дальнейшего анализа, полученный вектор разбивается на отрезки заранее выбранной длины, называемые окнами (обычно 200 пн (пар нуклеотидов)). Значение в окне определяется, как сумма единиц в нем.

Эксперимент ChIP-seq (как и большинство биологических экспериментов) не исключает наличие ошибок в результатах. Недостаточная специфичность антитела, наличие ошибок секвенирования, нестабильность положения белка на ДНК приводят к возникновению сигнала не зависящего от наличия взаимосвязи. Поэтому, для дальнейшего анализа результатов эксперимента, требуется построение вероятностной модели, способной отделять ошибки, а также выявлять зависимости соединений и, по возможности, описывать их структуру. Большинство существующих моделей ([7], [5]) для данных хроматин-иммунопреципитации основано на аппарате скрытых Марковских моделей (СММ) [4] первого порядка с Пуассоновскими испусканиями. Использование распределения Пуассона для покрытия опирается на предположение о том, что в каждой позиции генома в среднем начинается одинаковое количество прочтений. Марковский процесс, как правило, имеет два состояния «1» — сигнал есть и «0» — сигнала нет. Первый порядок модели означает, что состояние некоторого окна зависит только от состояния его прямого предшественника. Использование моделей первого порядка объясняется тем, что количество параметров модели, а также сложность её обучения и использования экспоненциально зависят от порядка. Так СММ порядка m для каждой цепочки из m состояний содержит распределение на следующее состояние (2^m вероятностных распределений). В связи с этим неправильный выбор m в обучении сильно усложняет модель и способствует ее переобучению.

Скрытые Марковские модели переменного порядка избегают такого эффекта, т.к. они не фиксируют длину строки порождающей следующее состояние и стараются ее уменьшить.

1. Постановка задачи

Цель данной дипломной работы — построение скрытой Марковской модели переменного порядка для анализа данных ChIP-seq.

Для достижения цели были определены следующие задачи.

1. Реализация скрытой Марковской модели переменного порядка.
2. Анализ эффективности работы модели на синтетических данных, сравнение с более простыми моделями (СММ первого порядка)
3. Применение к данным ChIP-seq.

2. Обзор существующих решений

Марковские модели переменного порядка (не скрытые) обучаются путем построения контекстного дерева переходов [2]. Скрытые Марковские модели фиксированного порядка обучаемы алгоритмом Баума-Велша [4]. Совмещение этих двух идей дает возможность обучить скрытые Марковские модели переменного порядка (СММП). Такой подход обучения был предложен в [6].

Итоговым алгоритмом обучения СММП был выбран модифицированный под поставленную задачу алгоритм из [6], дополненный недостающей информацией об обучении контекстных деревьев из статей [2], [3]. Модификация заключается в следующем: наблюдения итоговой модели будут порождаться не из контекстов, а из соответствующих состояний, т.е. распределение значений для каждого окна будет задаваться скрытым состоянием, которое определяет, была ли там взаимосвязь с белком или нет.

2.1. Основные понятия и определения

Путь $S = \{0, 1\}$ — множество состояний (в нашем случае 1 будет обозначать связь, 0 — обратное, и состояния будут находиться в скрытом слое), X_0, X_1, \dots — последовательность случайных величин (дискретный случайный процесс), значения которых лежат в S , а x_0, x_1, \dots — некоторая реализация случайных величин X_0, X_1, \dots

Определение 1. $\{X_i\}_{i \in \mathbb{Z}_+}$ называется *Марковским процессом порядка m* , если

$$\begin{aligned} & \forall t, t' \in \mathbb{N}, t, t' \geq m, \forall \vec{x} \in S^{t+1} \\ & P(X_t = x_t | X_{t-1} = x_{t-1}, X_{t-2} = x_{t-2}, \dots, X_0 = x_0) \\ & = P(X_t = x_t | X_{t-1} = x_{t-1}, X_{t-2} = x_{t-2}, \dots, X_{t-m} = x_{t-m}) \\ & = P(X_{t'} = x_t | X_{t'-1} = x_{t-1}, X_{t'-2} = x_{t-2}, \dots, X_{t'-m} = x_{t-m}) \\ & = P(X_{t'} = x_t | X_{t'-1} = x_{t-1}, X_{t'-2} = x_{t-2}, \dots, X_0 = x_0) \end{aligned}$$

Далее, имея дело с Марковским процессом, вероятности вида

$$P(X_{t'} = x_t | X_{t'-1} = x_{t-1}, X_{t'-2} = x_{t-2}, \dots, X_{t'-m} = x_{t-m})$$

где $t' \geq m$, будем записывать как $P(x_t | x_{t-1} \dots x_{t-m})$ (запись корректна, в силу независимости такой вероятности от t').

Для удобства будем считать, что наш процесс растет справа налево

$$\dots x_t, x_{t-1}, x_{t-2} \dots$$

Так, если цепь $\dots x_t, x_{t-1}, x_{t-2} \dots$ была порождена процессом порядка 2, то

$$P(x_t | x_{t-1}, x_{t-2} \dots) = P(x_t | x_{t-1}, x_{t-2})$$

Определение 2. *Марковская модель порядка m* — это вероятностная модель, описывающая марковский процесс порядка m . Параметрами модели являются множество переходов $A = \{a(q; x^m)\}_{q \in S, x^m \in S^m}$, где $a(q; x^m) = P(q | x^m)$, и начальное распределение $\pi = \pi(x^m)_{x^m \in S^m}$, где $\pi(x^m) = P(X_{0:m} = x^m)$.

Контекстом состояния x_t называется любой префикс строки $x_{t-1}, x_{t-2} \dots$

Определение 3. *Контекстное дерево* — дерево, в котором каждая внутренняя вершина имеет $|S|$ ребер соответствующих состояниям из S и метку, которая является конкатенацией метки на ее родителе и метки ребра от него. Метка в корне — пустая строка.

Контексты, соответствующие листьям контекстного дерева будем называть *главными контекстами* (иногда, когда речь будет идти только о листьях, слово «главные» будем опускать).

Теперь множество переходов для Марковского процесса порядка m можно определить как контекстное дерево глубины $m + 1$, каждый лист которого содержит распределение $P(\cdot | w)$, где w — метка на листе.

Для того, чтобы по дереву определить распределение на следующем состоянии X_t , достаточно из корня спуститься по ветке, вершины которой соответствуют контекстам этого состояния, $(x_{t-1}), (x_{t-1}x_{t-2}), \dots$. Лист на конце ветки и будет задавать распределение X_t .

Замечание 1. Пометки на листьях контекстного дерева определяют все дерево.

На рисунке 2 изображен пример контекстного дерева для Марковского процесса порядка 2 (серым подкрашены листья, ниже прямоугольниками обозначены распределения переходов). Можно заметить, что в этом примере, имея для некоторого состояния x_t контекст «1», необходимость уточнять его (т.е. спускаться дальше к листу) отсутствует, т.к. распределение на контекстах «10» и «11» одно и то же. Таким образом подрезанное дерево с рисунка 3 задает такие же распределения переходов как и дерево с рисунка 2. Однако второе контекстное дерево меньше (число главных контекстов меньше). Но не один Марковский процесс фиксированного порядка напрямую его использовать не может.

Определим процесс, который может иметь распределение переходов в виде дерева с рисунка 3.

Пусть τ — конечное контекстное дерево. Для $s \in \tau$ будем обозначать через $C(s)$ множество всех потомков, являющихся листьями τ . Для $s \notin \tau$, $C(s)$ — лист τ , являющийся префиксом s (можно заметить, что он существует и единственен).

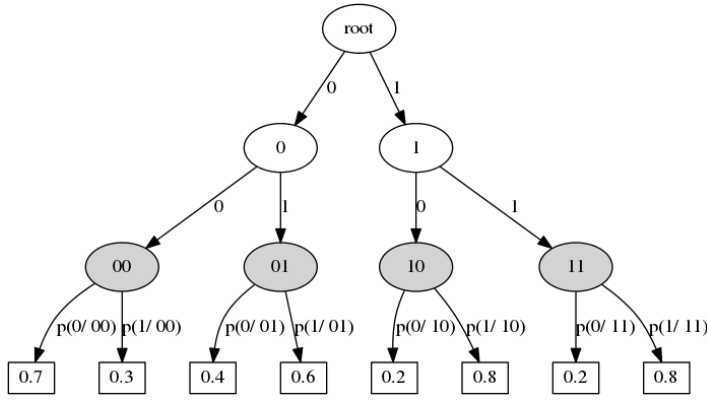


Рис. 2: Контекстное дерево переходов Марковского процесса порядка 2

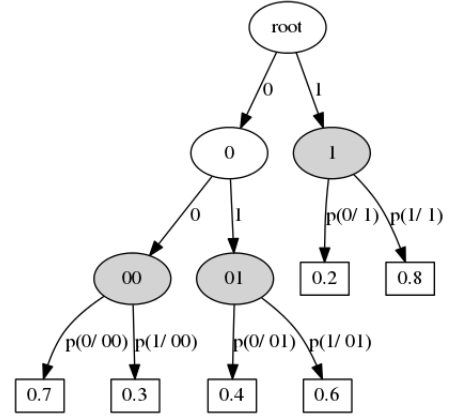


Рис. 3: Подрезанное контекстное дерево

Определение 4. Марковский процесс переменного порядка (МППП) с максимально-возможным порядком m — это вероятностный процесс, распределения на состояниях которого задаются распределениями на листьях некоторого контекстного дерева τ глубины не более чем $m + 1$, и

$$P(q|s) = \begin{cases} P(q|C(s)) & \text{для } s \notin \tau \\ \frac{\sum_{c \in C(s)} P(q|c)P(c)}{\sum_{q' \in S} \sum_{c \in C(s)} P(q'|c)P(c)} & \text{для } s \in \tau \end{cases}$$

вероятность того, что следующее состояние за цепью s является q

Определение 5. Марковская модель переменного порядка (ММПП) с максимально-возможным порядком m — вероятностная модель, описывающая соответствующий процесс. Параметрами модели являются множество переходов на листьях некоторого контекстного дерева τ глубины не более чем $m + 1$ и вероятностное распределение на них (листьях).

Замечание 2. МППП с максимально-возможным порядком m есть обобщение всех скрытых Марковских процессов порядка меньше либо равного, чем m .

2.2. Скрытые Марковские модели

Представим, что состояния — это какой-то скрытый признак/фактор (например, наличие или отсутствие связи белка и ДНК) цепи наблюдений $Y = \{y_t\}_{t \in Z_+}$. Для каждого наблюдения y_t он не известен, однако именно он определяет распределение на Y_t .

Тогда, имея Марковскую цепь $X = \{x_t\}_{t \in Z_+}$ и по координатно определяя для каждого состояния x_t новую случайную величину Y_t согласно распределению $P(\cdot | x_t)$, можно задать цепь $Y = \{y_t\}_{t \in Z_+}$.

Определение 6. Процесс, порождающий цепь по некоторому Марковскому процессу

$X = \{x_t\}_{t \in \mathbb{Z}_+}$ порядка m и распределению $P(\cdot | x_t)$, называется *скрытым Марковским процессом* порядка m . X называются *скрытыми состояниями*, Y — *наблюдениями*.

Определение 7. *Скрытая Марковская модель* (СММ) порядка m — вероятностная модель, описывающая соответствующий процесс. Параметрами модели является $\Lambda = (A, \pi, B)$, где A, π — параметры скрытого процесса X порядка m , $B = \{b(y; x)\}_{y \in R^l, x \in S}$ — множество распределений испусканий (где $b(y; x) = P(y|x)$).

Определение 8. *Скрытая Марковская модель переменного порядка* (СММПП) — вероятностная модель, описывающая соответствующий процесс. Параметрами модели является $\Lambda = (A, \pi, B)$, где A, π — параметры скрытого процесса переменного порядка X , $B = \{b(y; x)\}_{y \in R^l, x \in S}$, где $b(y; x) = P(y|x)$ — множество распределений испусканий.

2.3. Обучение модели СММПП

Задачу обучения скрытой Марковской модели переменного порядка можно сформулировать следующим образом: по цепи наблюдений $Y = (y_1, \dots, y_T)$ найти параметры $\Lambda = (A, B, C, \pi)$ модели СММПП, которые бы минимизировали количество контекстов несильно ухудшая правдоподобие модели по сравнению с правдоподобием модели с максимальным набором контекстов (допустимое отклонение регулирует параметр ϵ_{prune}).

Алгоритм:

Параметры алгоритма: m — максимальная длина контекста, ϵ_{EM} — порог для остановки EM, ϵ_{prune} — порог для обрезания дерева

1. Инициализация контекстов.

$$C = S^m$$

множество из всех строк длины m .

Начальные распределения переходов произвольные. Для нормального или Пуассоновского распределения испусканий в качестве начального распределения переходов имеет смысл взять оценку распределения переходов на цепи состояний, полученной алгоритмом k -means ($k = |S|$) по цепи наблюдений Y (это может уменьшить количество итераций на шаге EM).

2. EM (Expectation–Maximization algorithm).

Пересчет производится подобно алгоритму Баума-Велша для СММ [4].

(a) Е-шаг (Expectation)

Вводятся дополнительные параметры:

$$\alpha_t(c) = P(y_0^t, c(x_t) = c | \Lambda)$$

вероятность породить первые $t + 1$ наблюдений равными y_0^t , имея главным контекстом скрытого состояния x_t контекст c , из модели СММПП с параметрами Λ ;

$$\beta_t(c) = P(y_{t+1}^T | c(x_t) = c, \Lambda)$$

вероятность того, что последние $T - t$ наблюдений цепи длины T , порожденной из модели СММПП с параметрами Λ , в которой главный контекст скрытого состояния x_t является c , совпадают с y_{t+1}^T ;

$$\gamma_t(c) = P(x_t = c | Y, \Lambda)$$

вероятность того, что породив цепь Y моделью СММПП с параметрами Λ , главный контекст скрытого состояния x_t является c ;

$$p = P(Y | \Lambda)$$

правдоподобие модели.

Зная параметры модели, нововведенные параметры считаются следующим образом:

$$\alpha_0(c) = \pi(c)b(y_0; c)$$

$$\alpha_{t+1}(c) = \sum_{q \in S, c' = C(cq)} \alpha_t(c')a(c[0]; c')b(y_{t+1}; c[0])$$

$$\beta_T(c) = 1$$

$$\beta_t(c) = \sum_{q \in S, c' = C(qc)} a(q; c)b(y_{t+1}, c'[0])\beta_{t+1}(c')$$

$$p = \sum_{c \in C} \alpha_T(c)$$

$$\gamma_t(c) = \frac{\alpha_t(c)\beta_t(c)}{p}$$

(b) М-шаг (Maximization)

На этом шаге, алгоритм обновляет параметры модели максимизируя прав-

доподобие, при условии посчитанных α, β, γ .

Для пересчета множества распределений переходов вводится параметр ξ

$$\xi_t(q; c) = P(c(x_t) = c, x_{t+1} = q | Y, \Lambda)$$

вероятность того, что породив цепь Y моделью СММПП с параметрами Λ , главный контекст скрытого состояния x_t является c и состояние x_{t+1} совпадает с q

$$\xi_t(q; c) = \frac{\alpha_t(c)a(q; c)b(y_{t+1}, q)\beta_{t+1}(qc)}{p}$$

Обновление A по ξ

$$a(q; c) = \frac{\sum_t \xi_t(q, c)}{p(c)}$$

Обновление π

$$\pi(c) = \sum_t \gamma_t(c)$$

Пересчет B зависит от принятого семейства моделей испусканий и производится с помощью γ в точности также как и в алгоритме Баума-Велша. В случае распределения Пуассона $b(\cdot | c) \sim Poisson(\lambda_c)$ пересчет параметров происходит следующим образом

$$\lambda_c = \frac{\sum_t \gamma_t(c)y_t}{\sum_t \gamma_t(c)}$$

ЕМ-алгоритм запускает поочередно Е-шаг и М-шаг, пока правдоподобие с предыдущей итерации отличается от правдоподобия с текущей итерации более, чем на ϵ_{EM} , т.е. пока итерация дает значимый прирост правдоподобия

3. Подрезание дерева.

Если существует внутренний лист контекстного дерева s такой, что

$$\forall q \in S \quad P(sq)KL(sq, s) < \epsilon_{prune}$$

(дети не уточняют родителя), то s становится листом, а все его потомки обрезаются, где

$$KL(u, w) = \sum_{q' \in S} P(q'|u) \log \frac{P(q'|u)}{P(q'|w)}$$

расстояния Кульбака-Лейблера для апостериорных распределений.

Если таких листьев не существует, алгоритм заканчивает работу.

4. Пересчет параметров A , π на новых контекстах.

$$a_{new}(q; c_{new}) = P(q|c_{new})$$

пересчитывается по определению [4] контекстного дерева

$$\pi_{new}(c_{new}) = \sum_{c \in C(c_{new})} \pi(c)$$

$$a = a_{new}, \quad \pi = \pi_{new}$$

Переход на второй шаг (ЕМ-алгоритм).

Замечание 3. При пересчете вероятности могут очень близко подходить к нулю, что отрицательно влияет на точность расчета. Для избежания этой проблемы все расчеты следует проводить не с вероятностями, а с их логарифмами.

Замечание 4. ЕМ-алгоритм следует запускать несколько раз, т.к. он может «застрять» в локальных максимумах функции правдоподобия.

2.4. Обучение на нескольких выборках

В случае пропусков в наблюдениях (связанных, например, с отсутствием данных), обучение модели может проходить на множестве из нескольких цельных кусков наблюдений. Т.е. на вход алгоритма будет подаваться не одна выборка Y , а N выборок $\{Y^1 \dots Y^N\}$ подчиненных единому скрытому Марковскому процессу переменного порядка.

Алгоритм описанный выше подходит для обучения модели СММПП на таких данных с небольшим изменением в Е-шаге и М-шаге.

1. Е-шаг

Дополнительные параметры $\alpha^d, \beta^d, \gamma^d, \xi^d$ пересчитываются отдельно по каждой выборке $d \in 1, \dots, N$

Общая γ - конкатенация гамм на выборках

$$\gamma = [\gamma^1, \dots, \gamma^N]$$

$$p = \prod_d p^d$$

2. М-шаг

$$a(q; c) \propto \frac{\sum_d \sum_t \xi_t^d(q; c)}{\sum_t \gamma_t(c)}$$

2.5. Сравнение

Чем больше параметров у модели, тем лучше она подстраивается под данные, и тем проще переобучается. По этому, при сравнении моделей обученных на одних и тех же данных со схожим правдоподобием, предпочтительней будет та, которая проще. Конкретную величину, которую следует сравнивать для моделей обученных на одинаковых данных, предлагает критерий Акаике (AIC)

$$AIC = 2k - 2 \log L$$

где k — число степеней свободы или число параметров модели, L — максимальное правдоподобие модели на заданной выборке. Чем AIC меньше, тем модель лучше.

Число параметров для СММПП с n скрытыми состояниями, l контекстами, и Пуассоновскими испусканиями

$$\begin{aligned} k &= [\text{количество степеней свободы } A] + [\text{количество степеней свободы } B] \\ &\quad + [\text{количество степеней свободы } \pi] \\ &= l(n - 1) + n + (l - 1) \\ &= nl + n - 1 \end{aligned}$$

При $n = 2, k = 2l + 1$

Для СММ порядка $m, l = 2^m$, поэтому $k = 2^{m+1} + 1$

3. Реализация

Общий алгоритм обучения скрытой Марковской модели переменного порядка был реализован на языке программирования Python.

Критическим по производительности является E-шаг, он был перенесен на Cython. Основные использованные библиотеки.

- NumPy, SciPy для операций над матрицами.
- Joblib для распараллеливания по потокам.

В случае обучения на нескольких выборках, E-шаг для каждой выборки считается независимо, по этому эту часть можно параллелить.

- Pygraphviz для отрисовки деревьев.
- Matplotlib для отрисовки графиков.

4. Применение

4.1. Применение к симмулированным данным

План проверки работы СММПП.

1. Генерируем параметры начальной модели СММПП.
2. K раз порождаем выборки Y из заданной модели.
3. Обучаемся на каждой из выборок.
4. Считаем абсолютное среднее отклонение параметров реальной модели от предсказанной.

Ниже приведены три примера теста. Первый проверяет работу модели на примере смеси (СММ 0-го порядка), второй — для СММ первого порядка, третий для модели СММПП, не являющейся СММ фиксированного порядка. Для каждого теста было сгенерировано по 100 выборок длиной 5000. Значения параметров алгоритма обучения: максимально-возможный порядок $m = 4$, порог для обрезания $\epsilon_{prune} = 0.007$, порог для останова ЕМ $\epsilon_{EM} = 0.01$

Было посчитано среднее отклонение (MAE – Mean Absolute Error) следующих величин: A — распределение переходов, λ — параметр распределения Пуассона (распределение испусканий), \hat{fdr} (false discovery rate) — математическое ожидание доли ложных единиц среди всех предсказанных единиц, \hat{fndr} (false nondiscovery rate) — математическое ожидание доли ложных нулей среди всех предсказанных нулей.

4.1.1. Смесь

Рисунок 4 — реальное контекстное дерево для скрытого слоя, $B = \{Poisson(\lambda = 2), Poisson(\lambda = 10)\}$ — множество распределений испусканий.

Рисунок 5 — пример предсказанного дерева

	A	λ	\hat{fdr}	\hat{fndr}
MAE	0.07	0.21	0.01	0.02

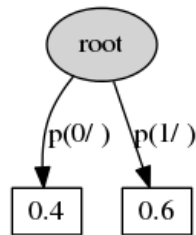


Рис. 4: Реальное дерево

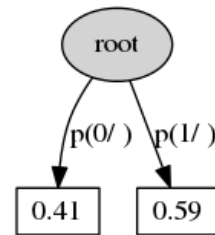


Рис. 5: Предсказанное дерево

4.1.2. СММ

Рисунок 6 — реальное контекстное дерево для скрытого слоя, $B = \{Poisson(\lambda = 1), Poisson(\lambda = 8)\}$ — множество распределений испусканий.

Рисунок 7 — пример предсказанного дерева

	A	λ	\hat{fdr}	\hat{fndr}
MAE	0.09	0.21	0.02	0.02

Рисунок 8 — график обучения. На нем изображен логарифм правдоподобия на всех итерациях обучения. Каждая ЕМ-часть выделена боксом, сверху которого написано число контекстов на момент обучения, снизу количество итераций в этой части и параметры распределения испусканий, полученных на последней итерации. На графике видно, как сначала алгоритм 6 итераций ЕМ обучался на 16 контекстах, после чего дерево подрезалось до 2 контекстов. Следующему ЕМ не удалось значимо увеличить правдоподобие модели, поэтому на третьей итерации он закончил работу. Далее дерево не удалось еще раз подрезать, поэтому весь алгоритм закончил свою работу (это видно из отсутствия следующего бокса под график ЕМ).

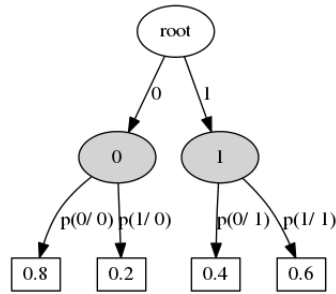


Рис. 6: Реальное дерево

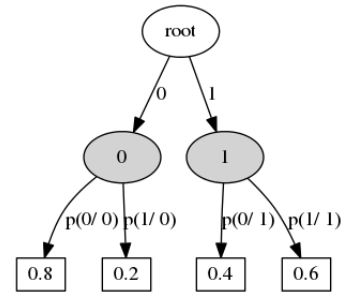


Рис. 7: Предсказанное дерево

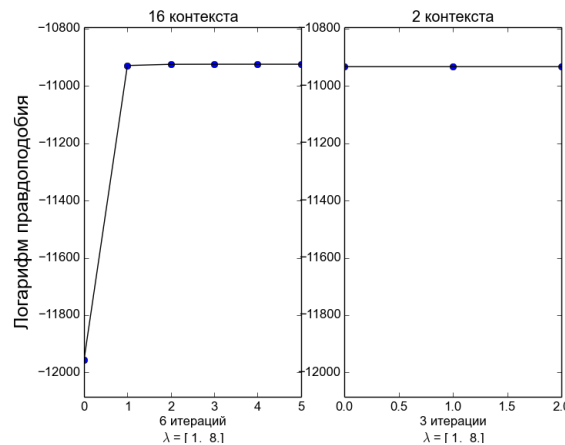


Рис. 8: График обучения

4.1.3. Более интересный случай, СММПП

Рисунок ?? — реальное контекстное дерево для скрытого слоя, $B = \{Poisson(\lambda = 3), Poisson(\lambda = 15)\}$ — множество распределений испусканий.

Рисунок 10 — пример предсказанного дерева

	A	λ	fdr	$\hat{f}dr$
MAE	0.11	0.20	0.02	0.02

Рисунок 11 — график обучения.

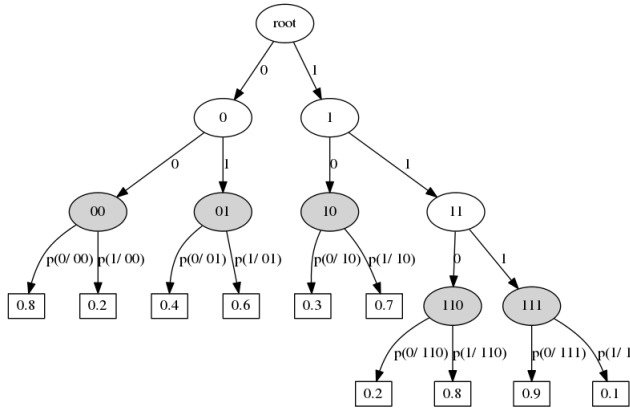


Рис. 9: Реальное дерево

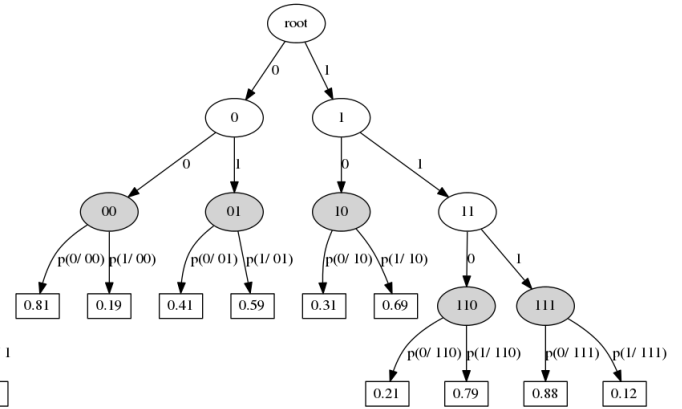


Рис. 10: Предсказанное дерево

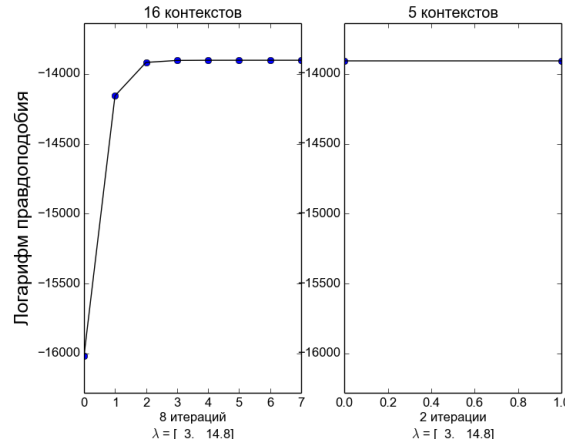


Рис. 11: График обучения

4.2. Применение к реальным данным

Данные были взяты из проекта ENCODE (ENCyclopedia of DNA Elements). В качестве исследуемого белка был выбран гистон H3 с ацетилированным лизином в 27-й позиции. Рассматриваемые клетки — эмбриональные стволовые клетки человека [1]. Размер окна был выбран равный 200 п.н.

В качестве выборок были рассмотрены ненулевые участки вектора, полученного после деления результата эксперимента ChIP-seq на окна.

Параметры обучения:
 $m = 5$, $\epsilon_{prune} = 0.04$, $\epsilon_{em} = 0.05$

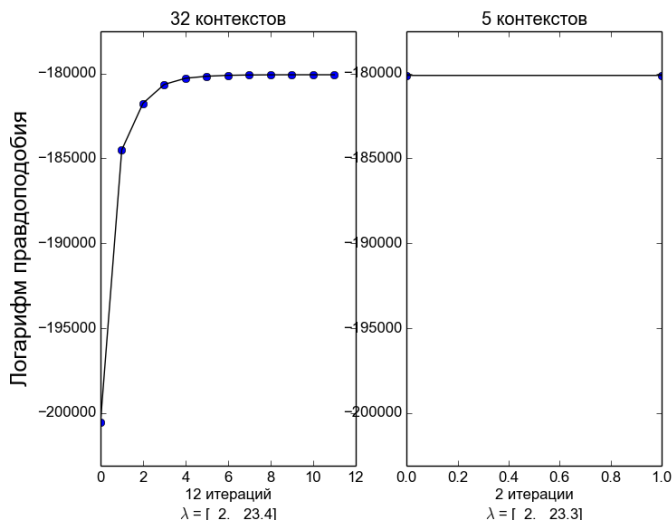


Рис. 12: График обучения

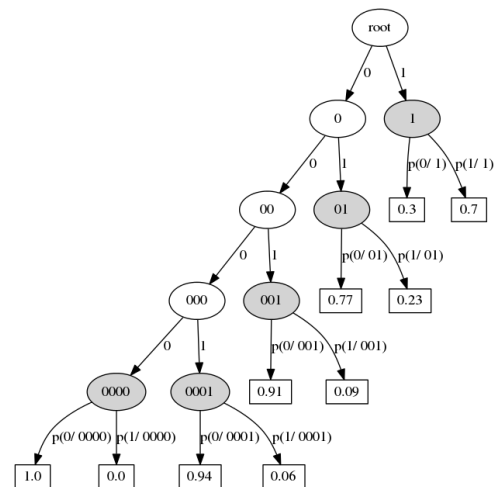


Рис. 13: Контекстное дерево

Рисунок 12 показывает, что сначала алгоритм 12 итераций ЕМ обучался на 32 контекстах, потом подрезал дерево до 5 контекстов. После чего ни обучение, ни подрезание не дало результатов, поэтому, алгоритм закончил работу.

Рисунок 13 иллюстрирует получившееся контекстное дерево переходов для скрытого слоя состояний, отвечающих за ДНК-белковую связь.

Приведем таблицу сравнения для СММПП, СММ5 (СММ 5-го порядка, соответствует дереву, с которого начиналось обучения) и СММ (СММ 1-го порядка, именно его чаще всего используют для анализа данных ChIP-seq)

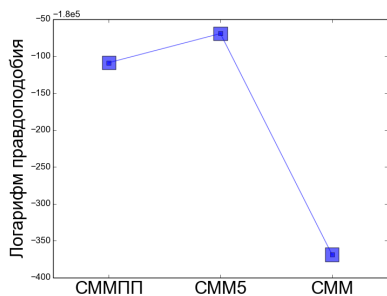


Рис. 14: Сравнение логарифма правдоподобия

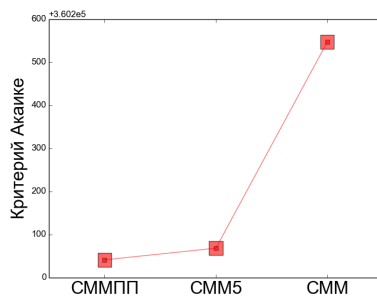


Рис. 15: Сравнение критерия Акаике

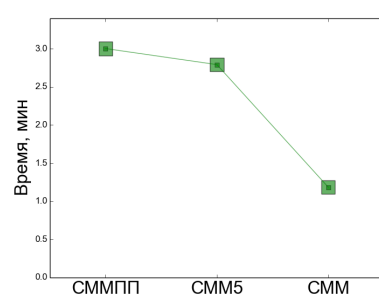


Рис. 16: Сравнение времени обучения

На рисунке 14 приведено сравнение правдоподобия моделей (в логарифмической шкале). Видно, что СММ5 лидирует. Однако СММПП ей не сильно уступает по сравнению с СММ.

Интересный результат показал критерий Акаике. Напомним, что данный критерий, чем меньше, тем лучше. Сравнение его изображено на рисунке 15. Хотя СММ

имеет гораздо меньшее количество параметров, чем СММПП, правдоподобие ее невелико, по этому по критерию Акаике она проигрывает. И наоборот, хотя СММ5 имеет лучшее среди этих трех моделей правдоподобие, оно имеет слишком много параметров, поэтому СММПП по критерию Акаике выигрывает и ее.

Рисунок 13 показывает сравнение времени обучения. Тут СММПП дает похожий результат с СММ5, немного ей уступая. СММ, в силу того, что она имеет более простую структуру, обучается быстрее всех.

Заключение

В ходе работы были решены поставленные задачи.

1. Проанализированы существующие скрытые Марковские модели переменного порядка, реализована подходящая под данные ChIP-seq модель.
2. Проведен анализ эффективности работы модели на синтетических данных, сравнение с более простыми моделями (СММ первого порядка, пятого),
3. Осуществлено применение к данным ChIP-seq

Список литературы

- [1] Broad Bradley Bernstein. Experiment summary for encsr000anp, 2011.
- [2] P Bühlmann and AJ Wyner. Variable length Markov chains. *The Annals of Statistics*, 27(2):480–513, April 1999.
- [3] Thierry Dumont. Context tree estimation in variable length hidden Markov models. *IEEE Transactions on Information Theory*, 60:3196–3208, 2014.
- [4] Lawrence Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
- [5] Lynch AG Tavare S Spyrou C, Stark R. BayesPeak: Bayesian analysis of ChIP-seq data. 2009.
- [6] Y Wang, Lizhu Zhou, and Jianhua Feng. Mining complex time-series data by learning Markovian models. In *International Conference on Data Mining*, 2006.
- [7] Jérôme Eeckhout David S Johnson Bradley E Bernstein Chad Nusbaum Richard M Myers Myles Brown Wei Li Yong Zhang, Tao Liu Clifford A Meyer and X Shirley Liu. Model-based Analysis of ChIP-Seq (MACS). 2008.