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

Предметная область

Наш организм есть огромное множество клеток. Клетки постоянно движутся, строят, разрушают. Вся жизнь наша заключается в их функционировании. Одна из интереснейших частей клетки — это ее память, ДНК (дезоксирибонуклеиновая кислота), которая хранит в себе просто неимоверное 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 клетки, во внутренних механизмах ее управления. Поэтому изучение ДНК-белковых взаимодействий крайне важно и актуально.

Однако перед самим 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, который по тысячам одинаковых клеток и выбранному белку, выдает вектор длины генома из 0 и 1, где 1 обозначает, что в окрестностях данной позиции ДНК был замечен белок, 0 - обратное.

Более подробно, но по-прежнему глубоко утрировано, все происходит 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.

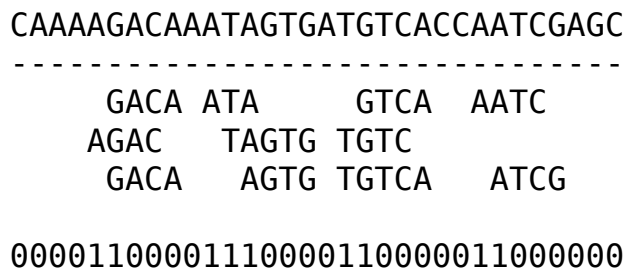

Рис. 1: Схематическое изображение выравнивания прочтений секвенатора (под чертой) на известную последовательность генома (над чертой).

Однако белок мог находиться не в самом начале фрагмента, и, кроме того, соединение белка с ДНК происходит не точечно, а на некотором участке ДНК. По этому, для дальнейшего анализа, полученный вектор разбивается на отрезки заранее выбранной длины, называемые окнами (обычно 200 пн (пар нуклеотидов)). Значение в окне определяется, как сумма единичек 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. По этому, для дальнейшего анализа результатов эксперимента, требуется построение некоторой 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 «+» — сигнал есть и «−» — сигнала нет. Первой 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 такой эффект, 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 (CMM 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 (СММПП). Такой подход обучения был предложен в [6].

Итоговым алгоритмом обучения СММПП был выбран слегка 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), $X_0, X_1, \ldots$ — последовательность случайных величин (дискретный случайный процесс), значения которых лежат в S , $x_0, x_1, \ldots$ - некоторая реализация случайных величин $X_0, X_1, \ldots$.

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

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

Далее, имея дело с Марковским процессом, вероятности вида

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

где $t' \geq m$, будем записывать как

$$P(x_t | x_{t-1} \ldots x_{t-m})$$

(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 $\pi(x^m) = P(X_{0:m} = x^m)$.

Другими словами, *Марковский процесс порядка* m — это случайный процесс, текущее состояние которого зависит лишь от m предшествующих состояний и не зависит от времени. Таким образом любая строка из m состояний задает распределение на следующее за ней состояние.

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.

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

Для того, чтобы по дереву определить распределение следующего состояния 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 ветке будет задавать распределение состояния X_t.

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

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

На рисунке 2 изображен пример контекстного дерева переходов 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 тоже. Таким образом подстриженное дерево с рисунка 3 задает такие же распределения переходов

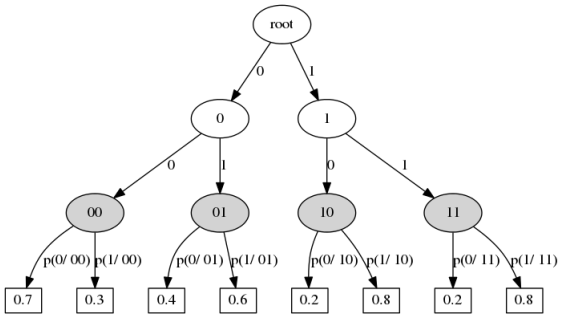

Рис. 2: Контекстное дерево переходов Марковского процесса порядка 2

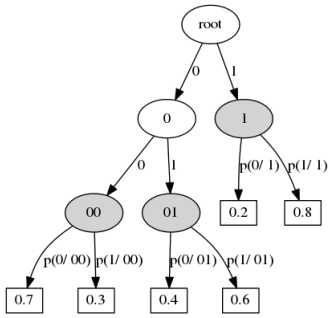

Рис. 3: Подрезанное контекстое дерево

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 такого дерева.

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

Определение 4. *Марковский процесс переменного порядка* с максимально-возможным порядком m— это вероятностный процесс, распределения на состояниях которого задаются распределниями на листьях некоторого контекстное дерево τ глубины не более, чем $m+1$

$$
\begin{aligned}
&[\textit{вероятность того, что следующее состояние цепи } s \textit{ является } q] \\
&= P(q|s) \\
&= P(q|C(s))
\end{aligned}
$$

для$s \notin \tau$

$$
= \frac{\sum_{c \in C(s)} P(q|c)p(c)}{\sum_{q' \in S} \sum_{c \in C(s)} P(q'|c)P(c)}
$$

для$s \in \tau$

Определение 5. Марковская модель переменного порядка 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. ММПП 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 он его определяет.

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

Определение 6. Процесс, порождающий цепь по некоторой Марковскому процессу $X = \{x_t\}_{t \in Z_+}$ порядка m и распределению $P(.|\ x_t)$,, называется *скрытым Марковским процессом порядка* m. X называеются *скрытыми состоянияниями*, Y — *наблюдениями*.

Определение 7. *скрытая Марковска 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, $B = \{b(y, x)\}_{y \in R^l, x \in S}$, где $b(y; x) = P(y|x)$ — множество распределений испусканий.

Определение 8. *скрытая Марковска 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

Задача:

По цепи наблюдений $Y = (y_1, ...y_T)$ найти параметры $\Lambda = (A, B, C, \pi)$ модели СММПП, которые бы максимизировали правдоподобие модели, минимизируя при этом длины контекстов. При этом, параметр алгоритма ϵ_{prune} определяет допустимое отклонение распределений, которым можно жертвовать в пользу уменьшения числа контекстов.

Алгоритм:

Параметры алгоритма: m - максимальная длина контекста, ϵ_{EM} - барьер для остановки ЕМ, ϵ_{prune} - барьер для обрезания дерева

1. Инициализация контекстов.

   $$C = S^m$$

   множество из всех строк длины m.

   Начальные распределения переходов произвольные. В определенных случаях

(Gauss, Poisson) частотное распределение, полученное из цепи алгоритмом k-means (k=m), ускоряет работу

2. EM (Expectation–Maximization algorithm).
   Пересчет производится подобно алгоритму Баума-Велша для СММ [4].

   (a) E-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.
   Зная параметры модели, нововведенные параметры считаются следующим образом:

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

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

$$\beta_T(c) = 1$$

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

$$p = P(Y|\Lambda) = \sum_{c \in C} \alpha_T(c)$$

   правдоподобие модели

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

(b) M-шаг (Maximization)

На этом шаге, алгоритм обновляет параметры модели максимизируя правдоподобие, при условии посчитанных α, β, γ;

Для пересчета множества распределений переходов вводится еще один дополнительный параметр ξ

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

s 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 $b(. \mid c) \sim Poisson(\lambda_c)$ пересчет параметров происходит следующим образом

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

EM-алгоритм запускает поочередно E-шаг и M-шаг, пока правдоподобие с предыдущей итерации отстает от правдоподобия с текущей итерации более, чем на ϵ_{EM} (т.е. пока итерация дает значимый прирост правдоподобия)

3. Обрезание дерева.

Если существует внутренний лист контекстного дерева s такой, что

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

(дети не уточняют родителя), то s становится листом, а все его потомки обреза-

ются, 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.

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

считается по замечанию [**??**]

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

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

Переход на второй шаг (EM-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. EM следует запускать несколько раз, т.к. он может застревать в локальных максимумах функции правдоподобия.

2.4. Обучение на нескольких выборках

В случае пропусков или разрывов в наблюдениях (связанных, например, с отсутствием данных), обучение модели может проходить на множестве из нескольких цельных кусков.
Более формально задачу можно описать так:
пусть дано N выборок $\{Y^1 \dots Y^N\}$ подчиненных единому скрытому Марковском процессу переменного порядка, требуется найти параметры модели Λ максимизирующие общее правдоподобие

$$P(Y^1 \dots Y^N|\Lambda) = \prod_i P(Y^I|\Lambda)$$

в классе рассматриваемой модели.
Приведем небольшие корректировки алгоритма выше для решения этой задачи.
EM-алгоритм

1. Expectation
   Дополнительные параметры $\alpha^d, \beta^d, \gamma^d, \xi^d$ пересчитываются отдельно по каждой

выборке $d \in 1, \ldots, N$

Общая γ - конкатенация гамм на выборках

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

$$p = \prod_d p^d$$

2. Maximization

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

нормировка $a(q; c) = \frac{a(q;c)}{\sum_q a(q;c)}$

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 параметров модели, L — максимальное правдоподобие модели на заданной выборке. Чем AIC меньше, тем модель лучше.

Количество степеней свободы для СММ m-го порядка с n скрытыми состояниями и Пуассоновскими испусканиями можно посчитать как

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

При $n = 2$,

$$k = 2^{m+1} + 1$$

Количество степеней свободы для СММПП с n скрытыми состояниями, l контек-

стами, максимально-возможным порядком m и Пуассоновскими испусканиями

$$\begin{aligned} k &= l(n-1) \ + \ n \ + \ (l-1) + 1 \\ &= n(l+1) \end{aligned}$$

При$n = 2$,

$$k = 2(l+1)$$

Последняя единица — это параметр задающий вид дерева.

Видно, что если сравнивать СММ порядка m и эквивалентную ей СММПП (т.е. $l = n^m$), то количество параметров у второй окажется на один больше, в остальных же случаях СММПП имеет меньшее количество параметров, и, при схожем правдоподобии выигрывает по критерию Акаике.

3. Реализация

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. Генерация параметров Λ начальной модели СММПП.
2. Порождение выборки Y из заданной модели.
3. Обучение новой модели на Y, получение предсказанных параметров $\hat{\Lambda}$.
4. Сравнение параметров Λ и $\hat{\Lambda}$.

Ниже приведены три примера теста.
Первый проверяет работу модели для смеси (СММ 0-го порядка), второй — для СММ первого порядка, третий для модели СММПП, не являющейся СММ фиксированного порядка.

4.1.1. Смесь

1. Параметры начальной модели: $\Lambda = (C, A, B)$
   множество контекстов $C = \{\text{""}\}$, множество переходов $A = \{[0.4, 0.6]\}$ (контекстное дерево изображено на рисунке 4), множество распределений испусканий $B = \{Poisson(2), Poisson(10)\}$.
2. Выборка длиной $T = 5000$
3. Обучение проходило начиная с полного дерева глубиной $m = 4$, и распределением переходов инициализированным алгоритмом k-means.
   Остальные параметры алгоритма: барьер для обрезания $\epsilon_{prune} = 0.007$, барьер для остановки EM $\epsilon_{EM} = 0.01$
4. На рисунке 5 изображено предсказанное контекстное дерево.

   Параметры предсказанной модели $\hat{\Lambda} = (\hat{C}, \hat{A}, \hat{B})$
   $\hat{C} = \{\text{""}\}, \hat{A} = \{[0.41, .59]\}, \hat{B} = \{Poisson(2), Poisson(10.1)\}$.
   Параметры исходной и предсказанной модели идентичны.

4.1.2. СММ

1. Параметры начальной модели: контекстное дерево — рисунок **??**,
   $B = \{Poisson(1), Poisson(8)\}$.
2. Выборка длиной $T = 5000$

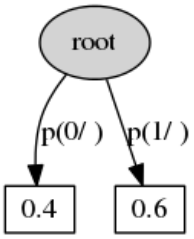

Рис. 4: Реальное дерево

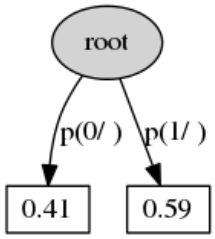

Рис. 5: Предсказанное дерево

3. Параметры обучение те же, что и примером выше.

4. Предсказанное контекстное дерево — рисунок 7, $\hat{B} = \{Poisson(1), Poisson(8)\}$
   Параметры исходной и предсказанной модели идентичны.
   Рисунок 8 — график логарифма правдоподобия по всем итерациям обучения. Грубо говоря, это карта обучения. На ней видно, как сначала алгоритм 6 итераций EM обучался на 16 контекстах, после чего дерево подстриглось до 2 контекстов. Следующему EM не удалось значимо увеличить правдоподобие модели, поэтому на третей 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 EM).

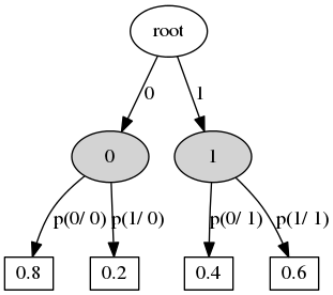

Рис. 6: Реальное дерево

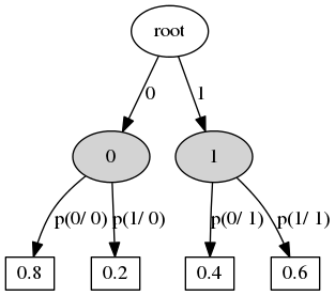

Рис. 7: Предсказанное дерево

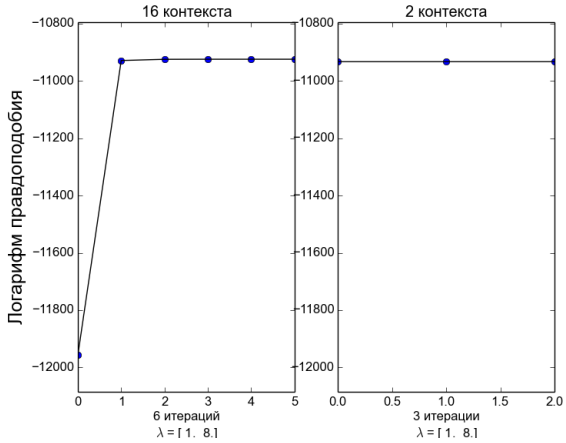

Рис. 8: Карта обучения

4.1.3. Более интересный случай, СММПП

1. Параметры начальной модели: контекстное дерево — рисунок 9,
   $B = \{Poisson(3), Poisson(15)\}$.

2. Выборка длиной $T = 5000$

3. Параметры обучение те же, что и примерами выше.

4. Предсказанное контекстное дерево — рисунок 10, $\hat{B} = \{Poisson(3.1), Poisson(15)\}$
   Параметры исходной и предсказанной модели идентичны.
   Рисунок 11 — график обучения.

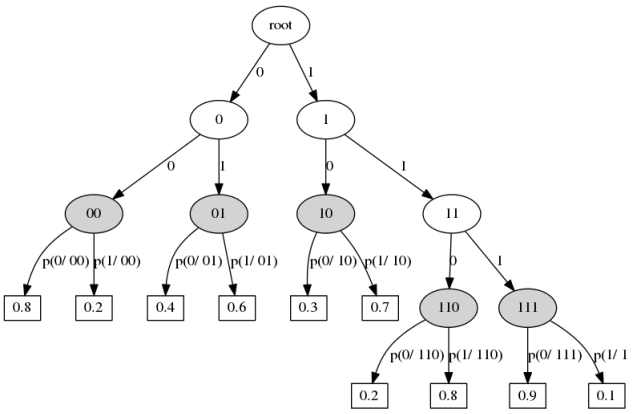

Рис. 9: Реальное дерево

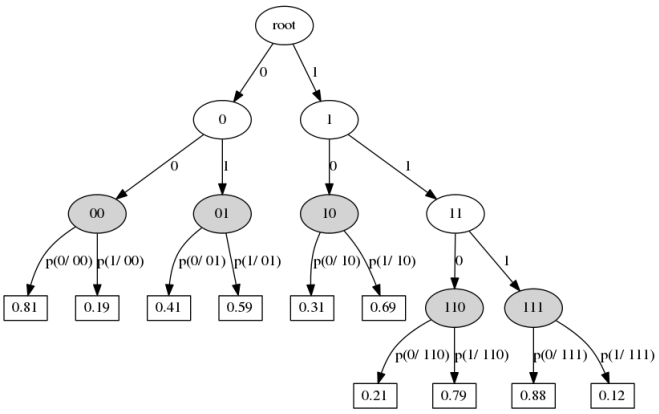

Рис. 10: Предсказанное дерево

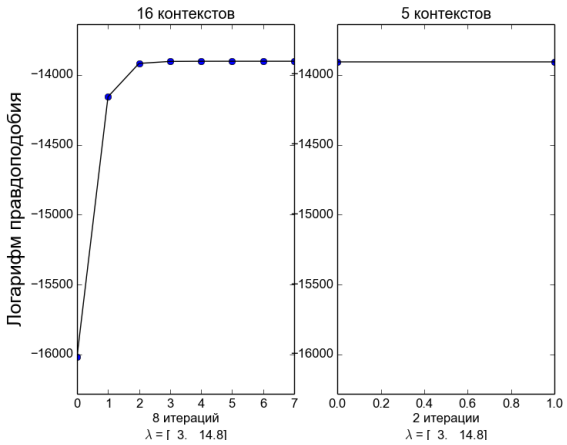

Рис. 11: График обучения

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 хвоста. 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:

Начальная глубина дерева = 6. $\epsilon_{prune} = 0.04$, $\epsilon_{em} = 0.05$

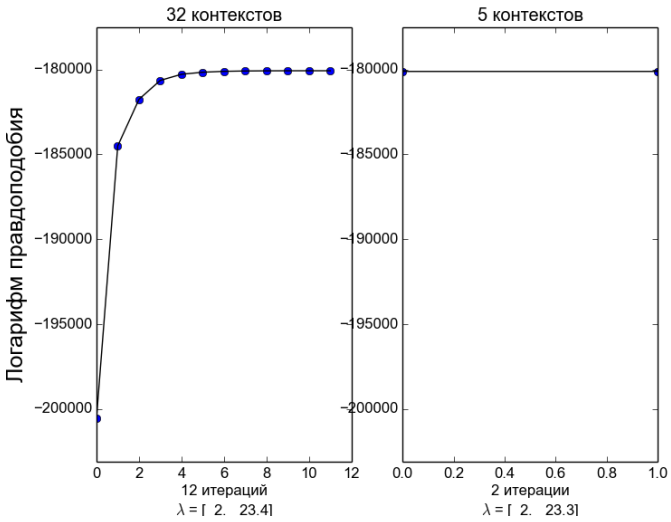

Рис. 12: График обучения

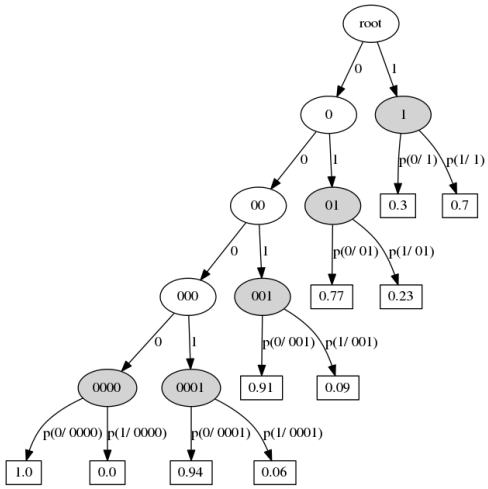

Рис. 13: Контекстное дерево

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

Рисунок 2 показывает получившееся контекстное дерево.

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

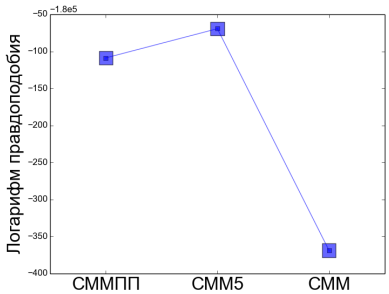

Рис. 14: Сравнение логарифма правдоподобия

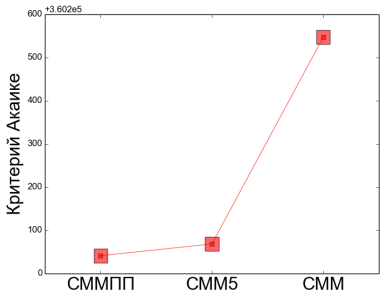

Рис. 15: Сравнение критерия Акаике

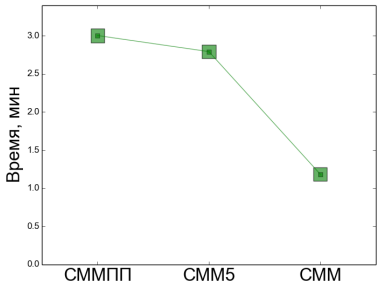

Рис. 16: Сравнение времени обучения

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 невилико, 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 (CMM 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 Eeckhoute 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.