САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Математико-механический факультет

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
Mathematics & Mechanics Faculty

Chair of Software Engineering

Anna Atamanova

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

Graduation 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 состоит из огромного числа клеток. Клетки постоянно что-то строят, воспроизводят, разрушают. Любое наше действие основывается на их функционировании. Остается лишь удивляться, как такой сложный процесс еще не превратился в хаос. Но не превратился. 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 случайный 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

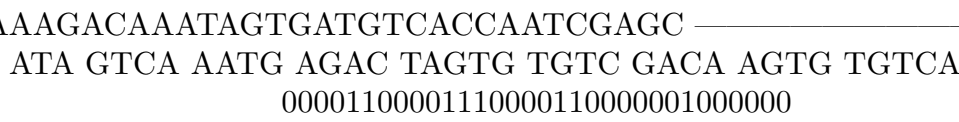

Рис. 1: Схематическое изображение выравнивания прочтений секвенатора (под чертой) на известную последовательность генома (над чертой).

стоит 1, если в соответствующей позиции генома начиналось хотя бы одно прочтение и 0 в обратном случае.

Для дальнейшего анализа, вектор разбивается на отрезки заранее выбранной длины, называемые окнами (обычно 200 пн[1]). Значение в окне определяется, как сумма единичек в нем. Такое разбиение обуславливается тем, что изначально белок мог находиться не в начале фрагмента и, кроме того, соединение белка с ДНК происходит не точечно а на некотором участке ДНК.

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] второго 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. Второй порядок модели означает, что состояние некоторого окна зависит только от состояния его прямого предшественника. Использование моделей второго 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 $m-1$ состояний содержит распределение на следующее состояние (2^{m-1} 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]пар нуклеотидов

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. разработка и реализация скрытой Марковской модели переменного порядка;
2. анализ эффективности работы модели на синтетических данных, сравнение с более простыми моделями (CMM второго порядка), 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 обнаружен в [6]. В [3] были выявлены похожие идеи.

Итоговым алгоритмом обучения СММПП был выбран слегка модифицированный под поставленную задачу алгоритм из [6], дополненный недостающей информацией из других статей.

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, а $X_0, X_1, \ldots$ – последовательность случайных величин, значения которых лежат в S (в нашем случае 1 будет обозначать связь, 0 - обратное), $x_0, x_1, \ldots$ - выборка (цепь с конкретными значениями $X_0, X_1, \ldots$).

Марковский процесс — это случайный процесс, будущее состояние которого зависит лишь от настоящего и не зависит от прошлого и времени.
Другими словами, Марковский процесс имеет 2 монетки (стороны которых не обязательно равновероятны), и, каждое следующее состояние определяется подкидыванием монетки, которая соответствует текущему состоянию. Более научно, монетки – это априорное вероятностное распределение $p(. \mid q)$, где q – это текущее состояние. Такие распределения задаются матрицей переходов $A = (a(q; x))$, где $a(q; x) = P(q|x)$ - вероятность из состояния x перейти в состояние q. Распределение на начальное состоянии процесса определяется еще одним параметром $\pi = (\pi_x)$, где $\pi_x = P(X_0 = x)$.
Итого, параметрами Марковской модели определяющей Марковский процесс являются матрица переходов A и вектор начального распределения π.
В другую сторону, цепь $\{x_t\}_{t \in Z_+}$ является Марковской, если она была порождена Марковским процессом.

Теперь предположим, что монетки ориентируются не только на одно предыдущее состояние, а на m предшествующих. Соответственно, их количество должно возрасти до 2^m (каждая монета соответствует своей строке из m состояний). Тогда параметрами модели, определяющей данный процесс, будут — 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 $\pi(x^m) = P(X_{0:m} = x^m)$. Такая модель называется *Марковской моделью порядка* $m + 1$ (процесс - Марковский процесс порядка $m + 1$). Соответственно, просто Мар-

ковский процесс – это Марковский процесс порядка 2. А Марковский процесс первого порядка – это процесс порождающий каждое следующее состояние всего по одной монетке. Т.е. распределение на состояниях такого процесса вообще не зависит от его истории.

Для удобства, будем считать, что наш процесс растет справа налево

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

.

Контекстом состояния x_t называется любой префикс строки $x_{t-1}, x_{t-2} \dots$. Тогда недавние состояния контекста окажутся в его голове, а давние в хвосте. Так, если цепь $\dots x_t, x_{t-1}, x_{t-2} \dots$ была порождена процессом порядка 3, то

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

Контекстное дерево — дерево (бор), в котором каждая внутренняя вершина имеет 2 ребра соответствующих состояниям $\{0, 1\}$ и метку, которая является конкатенацией метки на ее родителе и метки ребра от него. Метка на корне - пустая строка.

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

Для того, чтобы по дереву определить распределение следующего состояния x_t, достаточно взять его за корень и спуститься по вершинам, соответствующим контекстам этого состояния, (x_{t-1}), $(x_{t-1}x_{t-2}), \dots$. Конечный лист будет задавать распределение состояния X_t.

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

На рисунке 2 изображен пример контекстного дерева переходов для Марковского процесса порядка 3 (серым подкрашены листья, ниже прямоугольниками обозначены распределения переходов).

Заметим, что в примере 2, имея для некоторого состояние 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 как и дерево с рисунка 2. Однако второе контекстное дерево имеет разные длины главных контекстов, по этому не один Марковским процесс фиксированного порядка напрямую его использовать не может. Определим процесс, который может иметь распределение переходов в виде такого дерева.

Будем называть контекстное дерево *правильным*, если все его внутренние узлы

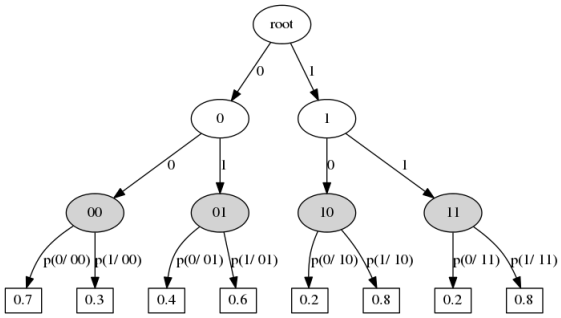

Рис. 2: Контекстное дерево переходов Марковского процесса порядка 3

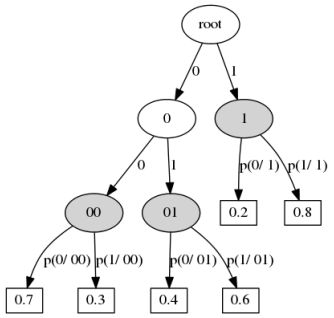

Рис. 3: Подрезанное контекстое дерево

имеют ровно по 2 ребенка.

Процесс, переходы которого задаются правильным контекстным деревом, а листья задают распределения на контекстах, и π - распределение на листьях, называется *Марковским процессом переменного порядка* (ММПП). При этом максимально-возможный порядок ММПП (т.е. максимальная длина контекстов) будем фиксировать (и обозначать тоже за m).

Можно заметить, что ММПП – это обобщение всех скрытых Марковских процессов порядка меньше либо равного, чем m.

2.2. Скрытые Марковские модели

Представим, что каждое состояние Марковского процесса тоже имеет некоторое вероятностное распределение (например, еще одну монетку, или подкошенный ∞-гранник, или распределение Пуассона)

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 породить цепь $Y = \{y_t\}_{t \in Z_+}$.

Процесс, порождающий такую цепь по некоторой Марковской цепи порядка m, называется *скрытым Марковским процессом порядка* m.
X — *скрытые состояния*, Y — *наблюдения*.

Итого, *скрытая Марковска модель* (СММ) порядка m имеет следующие параметры: множество переходов $A = \{a(q, x^m)\}_{q \in S, x^m \in S^m}$, начальное распределение π и множество распределений испусканий $B = \{b(y, x)\}_{y \in R^l, x \in S}$, где $b(y; x) = P(y|x)$.

Аналогично, *скрытая Марковска модель переменного порядка* (СММПП) задается Марковской моделью переменного порядка (конечное множество контекстов $C = \{c_i\}_i$, где c_i - листья некоторого правильного контекстного дерева, распределения на листьях $A = \{a(q; c)\}_{q \in S, c \in C}$, начальное распределение π) и множеством распределений испусканий $B = \{b(y, x)\}_{y \in R^l, x \in S}$, где $b(y; x) = P(y|x)$.

Заметим, что все определяется аналогично, будь у нас множество состояний S не

из двух элементов, а из n.

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. [2]

Другими словами, найти

$$\Lambda = \underset{\Lambda}{\operatorname{argmin}} \{|\Lambda(C)| \; | \Lambda \in \underset{\Lambda}{\operatorname{argmax}} \{P(Y|\Lambda)\}\}$$

Алгоритм:

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

1. Инициализация контекстов.

   $$C = \{c | c \in S^m\}$$

   множество из всех строк длины m.

   Начальные распределения переходов произвольные. [3]

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

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

[2] Параметр алгоритма ϵ определяет допустимое отклонение распределений

[3] В определенных случаях (Gauss, Poisson) частотное распределение, полученное из цепи алгоритмом k-means (k=m), ускоряет работу

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

$\xi_t(q; c) = P(c(x_t) = c, x_{t+1} = q|Y, \Lambda)$ — 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 $b(.\mid c)Poisson(\lambda_c)$ 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 обрезаются.

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

расстояния Кульбака-Лейблера для апостериорных распределений.

4. Если на третьем шаге ничего не произошло, то алгоритм заканчивает работу, иначе происходит обновление параметра A для новых контекстов

$$a(q;c) = P(q|c)$$

и алгоритм переходит на второй шаг (EM-алгоритм).

Обозначения:
$c[0]$ - состояние, являющееся началом контекста c [4]
$c(x_t)$ - контекст состояния x_t
$C(s)$ - листья, являющиеся потомками s, если s принадлежит дереву
$C(s)$ - контекст максимальной длины, являющийся префиксом s, если s не принадлежит дереву

[4]Контекст c представляем как последовательность состояний $c[0]c[1]\dots c[l-1]$, где l - длина контекста.

Замечание. Вероятностные переходы на главных контекстах задают вероятностные переходы на всем дереве

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

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

Замечание. 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, 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

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

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

1. Expectation Считаем для каждой выборки $\alpha^d, \beta^d, \gamma^d, \xi^d$
   Общая γ - конкатенация гамм на выборках

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

   $p = \prod_d p^d$
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 имеет меньшее число степеней свобод или параметров. Конкретную величину, которую следует сравнивать для моделей обученных на одинаковых данных, предлагает критерий Акаике (AIC).

$$AIC = 2k - 2log(L)$$

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
k = [количество степеней свободы A] + [количество степеней свободы B] + [количество степеней свободы π] =
$= n^{m-1}(n-1) \ + \ n \ + \ (n^{m-1}-1) =$
$= n^m + n - 1$
При $n = 2$, $k = 2^m + 1$

Количество степеней свободы для СММПП с n скрытыми состояниями, l контекстами, максимально-возможным порядком m и Пуассоновскими испусканиями =
$= l(n-1) \ + \ n \ + \ (l-1) + 1 =$
$= n(l+1)$
При $n = 2$, $k = 2(l+1)$
Последняя единичка – это параметр задающий вид дерева.

Видно, что если сравнивать СММ порядка m и эквивалентную ей СММПП (т.е. $l = n^{m-1}$), 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. Проверка на смоделированных данных

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 первого порядка), второй — для СММ второго порядка, третий для модели СММПП, не являющейся СММ фиксированного порядка.

- Смесь.

  1. Параметры начальной модели: $\Lambda = (C, A, B)$
     множество контекстов $C = \{\texttt{""}\}$, 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 $\hat{\Lambda} = (\hat{C}, \hat{A}, \hat{B})$
     $\hat{C} = \{\texttt{""}\}, \hat{A} = \{[0.41, .59]\}, \hat{B} = \{Poisson(2), Poisson(10)\}$.
     Параметры исходной и предсказанной модели идентичны.

- СММ порядка 2.

  1. Параметры начальной модели: контекстное дерево – рисунок 6,
     $B = \{Poisson(1), Poisson(8)\}$.
  2. Выборка длиной $T = 5000$
  3. Параметры обучение те же, что и примером выше.

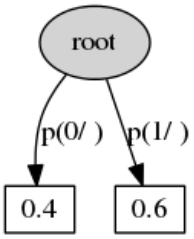

Рис. 4: Реальное дерево

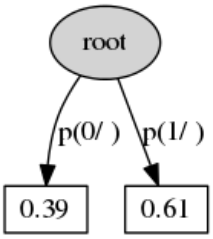

Рис. 5: Предсказанное дерево

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 7 итераций EM обучался на 16 контекстах, после чего дерево подстриглось до 2 контекстов. Следующему EM не удалось значимо увеличить прадоподобие модели, поэтому на второй 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).

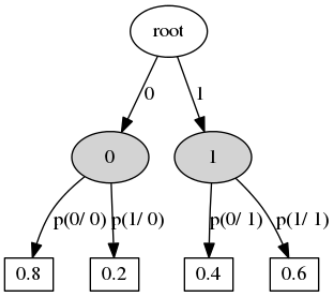

Рис. 6: Реальное дерево

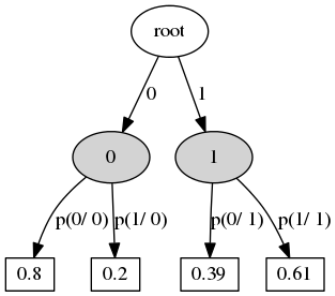

Рис. 7: Предсказанное дерево

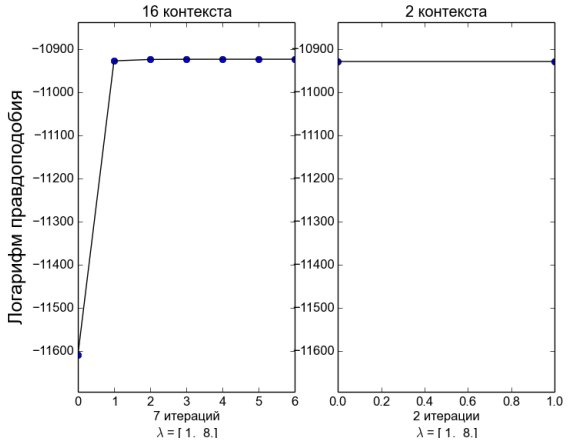

Рис. 8: Карта обучения

- Более интересны 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 – карта обучения.

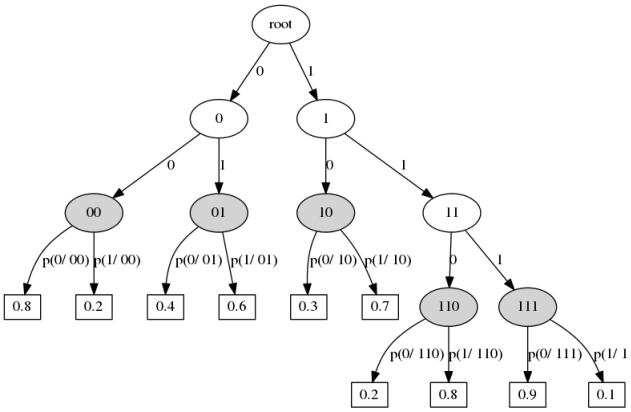

Рис. 9: Реальное дерево

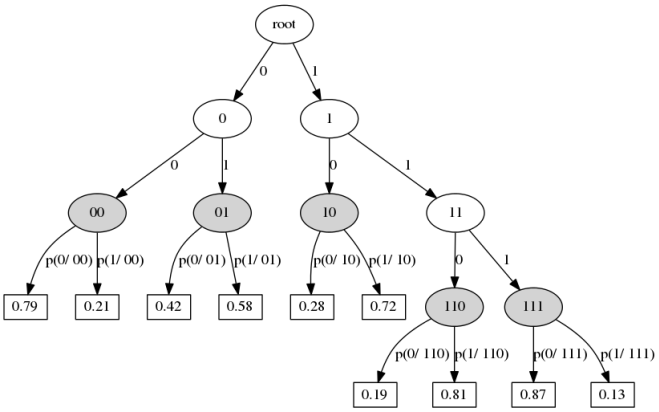

Рис. 10: Предсказанное дерево

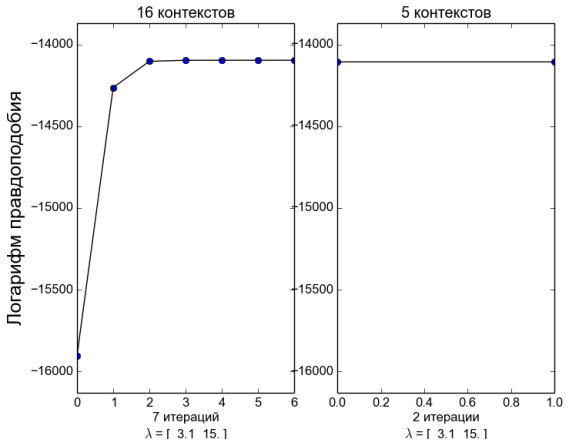

Рис. 11: Карта обучения

4.2. ChIP-seq, реальные данные

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. $\epsilon_{prune} = 0.04$, $\epsilon_{em} = 0.05$

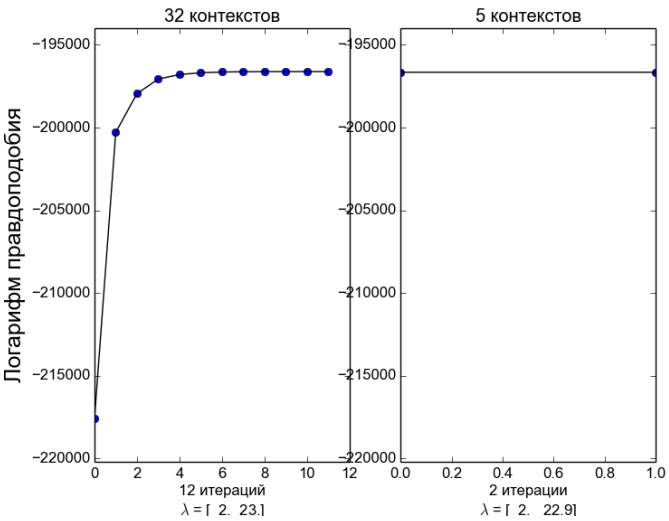

Рис. 12: Карта обучения

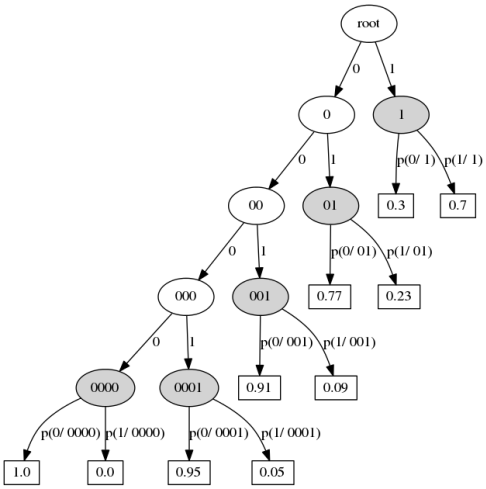

Рис. 13: Контекстное дерево

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, СММ6 (СММ 6-го порядка, соответствует дереву, с которого мы начали обучения)и СММ2 (СММ 2-го порядка, именно его чаще всего используют для анализа данных ChIP-seq)

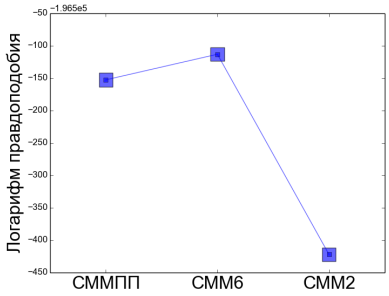

Рис. 14: Сравнение логарифма правдоподобия

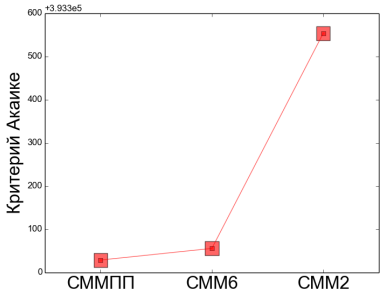

Рис. 15: Сравнение критерия Акаике

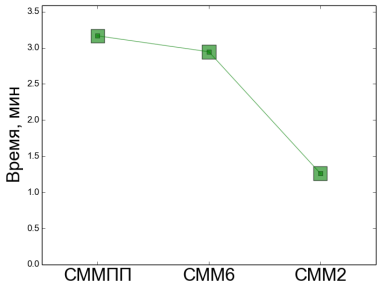

Рис. 16: Сравнение времени обучения

На рисунке 14 приведено сравнение правдоподобия моделей (в логарифмической шкале). Видно, что СММ6 лидирует. Однако СММПП ей не сильно уступает по сравнению с СММ2.

Интересный результат показал критерий Акаике. Напомним, что данный критерий, чем меньше тем лучше. Сравнение его изображено на рисунке 15. Хотя СММ2

имеет гораздо меньшее количество параметров, чем СММПП, правдоподобие ее невилико, по этому по критерию Акаике она проигрывает. И наоборот, хотя СММ6 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 СММ6, немного ей уступая. СММ2, 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, разработана и реализована подходящая под данные ChIP-seq модель.

2. Проведен анализ эффективности работы модели на синтетических данных, сравнение с более простыми моделями (CMM второго порядка, пятого), 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.