

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:
Лебедев С. А.

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:
Sergei Lebedev

Saint-Petersburg
2015

Оглавление

Введение	4
1. Постановка задачи	7
2. Обзор существующих решений	8
2.1. Основные понятия и определения	8
2.2. Скрытые Марковские модели	10
2.3. Обучение модели СММП	11
2.3.1. Инициализация	12
2.3.2. ЕМ (Expectation–Maximization algorithm)	12
2.3.3. Подрезание дерева	14
2.4. Обучение на нескольких выборках	14
2.5. Сравнение	15
3. Реализация	16
4. Применение	17
4.1. Применение к симмулированным данным	17
4.1.1. Пуассоновская смесь	18
4.1.2. СММ	18
4.1.3. СММП, не являющаяся СММ фиксированного порядка	19
4.2. Применение к реальным данным	20
Заключение	23
Список литературы	24

Введение

Предметная область

Дезоксирибонуклеиновая кислота (ДНК) — молекула, обеспечивающая хранение генетического кода, который определяет развитие и функционирование живых организмов. ДНК хранит наследственную информацию, информацию о структуре РНК и белков. Белки выполняют структурные, сигнальные, механические и другие функции. Соединения ДНК с конкретным белком могут влиять на структуру ДНК, на внутренние механизмы управления клетки, поэтому изучение ДНК-белковых взаимодействий крайне важно и актуально.

Геном — это совокупность всех молекул ДНК ядра клетки. Каждая ДНК состоит из двух цепей нуклеотидов, поэтому позиции генома измеряется в парах нуклеотидов (пн).

В данной работе рассматривается задача нахождения позиций связывания генома и конкретного белка, то есть нахождения позиций ДНК-белковых взаимодействий при заранее выбранном белке.

ChIP-seq

ChIP-seq (иммунопреципитация хроматина с последующим секвенированием, англ. chromatin immunoprecipitation sequencing) — биологический эксперимент, позволяющий получить информацию о наличии или отсутствии взаимодействия ДНК с заданным белком.

Эксперимент проводится на множестве одинаковых клеток и включает в себя следующие стадии:

1. фиксация всех обнаруженных белков на ДНК;
2. расщепление ДНК на фрагменты;
3. вылавливание фрагментов, связанных с исследуемым белком (с помощью специфичного к исследуемому белку антитела);
4. считывание концов фрагментов (называемые риды или прочтениями) до тех пор, пока каждый фрагмент с высокой вероятностью не будет прочитан 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.

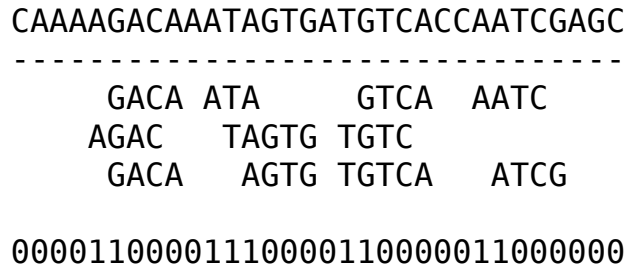


Рис. 1: Схематическое изображение выравнивания прочтений секвенатора (под чертой) на известную последовательность генома (над чертой).

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

Соединение белка с ДНК происходит не точно, а на некотором участке ДНК, поэтому, для дальнейшего анализа, полученный массив разбивается на отрезки заранее выбранной длины, называемые окнами (обычно 200 пн). 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 ([8], [5]) для данных ChIP-seq основано на аппарате скрытых Марковских моделей (СММ) первого порядка [4] с Пуассоновскими испусканиями. Использование распределения Пуассона для моделирования 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 при обучении сильно усложняет модель и способствует её переобучению. Переобучение — это одна из основных проблем машинного обучения, при которой модель слишком сильно подгоняется под обучающую выборку и находит в ней случайные закономерности, которые не характерны для

данных генеральной совокупности.

Скрытые Марковские модели переменного порядка менее склонны к переобучению в силу того, что они не фиксируют длину строки, порождающей следующее состояние, и стараются ее уменьшить.

1. Постановка задачи

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

Для достижения цели были определены следующие задачи:

1. реализовать скрытую Марковскую модель переменного порядка;
2. проанализировать эффективность работы модели на синтетических данных;
3. применить к данным ChIP-seq, сравнение с более простыми моделями (СММ первого порядка).

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]. В работе [7] было предложено совмещение этих двух идей для обучения скрытых Марковских моделей переменного порядка (СММП).

В данной работе алгоритмом обучения СММП был выбран модифицированный под поставленную задачу алгоритм из [7]. Он был дополнен недостающей информацией об обучении контекстных деревьев из статей [2], [3]. Модификация заключается в следующем: наблюдения итоговой модели будут порождаться из соответствующих состояний, т.е. распределение значений для каждого окна задается скрытым состоянием, которое определяет, была ли там взаимосвязь с белком или нет. В работе [7] такие распределения определялись всем контекстом.

2.1. Основные понятия и определения

Путь $S = \{0, 1\}$ — множество состояний (в рамках рассматриваемой задачи, «1» означает наличие связи, «0» — ее отсутствие), 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

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

Далее, для Марковских процессов, 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 вероятностных распределений 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 $\pi(x^m) = P(X_{0:m} = x^m)$.

Определение 3. *Контекстное дерево* — это 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.

Контекстом состояния x_t будем называть любой префикс строки $x_{t-1}, x_{t-2} \dots$. Контексты, соответствующие листьям контекстного дерева, будем называть *главными контекстами* (иногда, когда речь будет идти только о листьях, слово «главные» будем опускать).

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. Метки на листьях контекстного дерева полностью определяют дерево.

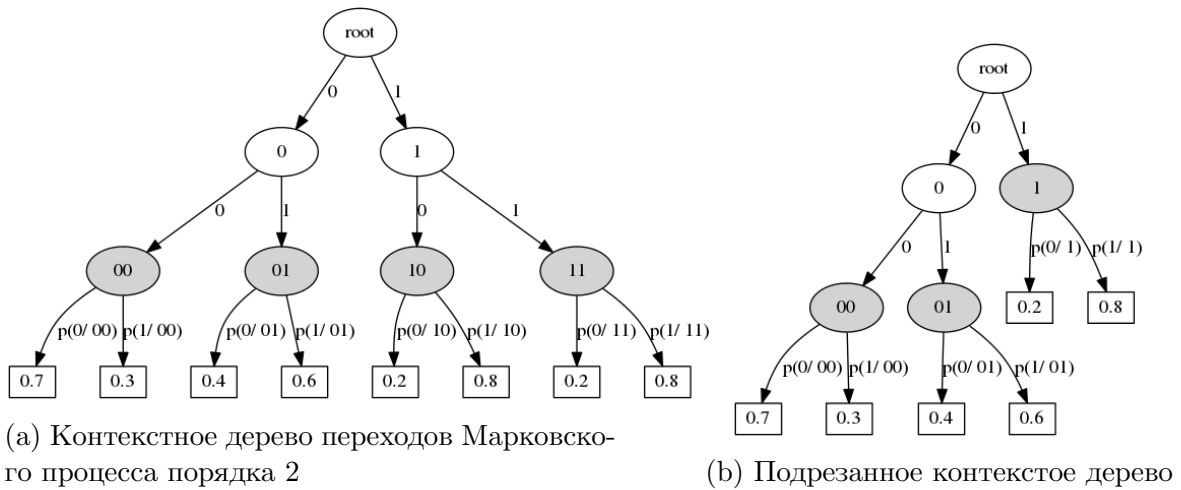


Рис. 2: Эквивалентные контекстные деревья.

Серым подкрашены листья, ниже прямоугольниками обозначены распределения переходов.

На рисунке 2а изображен пример контекстного дерева для Марковского процесса порядка 2. 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 2b задает такие же распределения переходов как и дерево с рисунка 2а. 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 2b. 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).

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 $m + 1$.*

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 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 $Y = \{y_t\}_{t \in \mathbb{Z}_+}$. Для каждого наблюдения y_t он не известен, однако именно он определяет распределение на Y_t . Т.е. цепь Y порождается из Марковской цепи $X = \{x_t\}_{t \in \mathbb{Z}_+}$ путем покоординатного определения новой случайной величины Y_t для каждого состояния x_t согласно распределению $P(\cdot | x_t)$.

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, π, 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 не сильно ухудшая правдоподобие модели по сравнению с правдоподобием модели на полном дереве (допустимое отклонение распределений регулирует параметр ϵ_{prune}).

На рисунке 1 схематично представлен алгоритм обучения СММПП. Описание основных шагов: инициализация, ЕМ-алгоритм, подрезание дерева — приведено далее.

Data:

Y , // наблюдения

$m, \epsilon_{EM}, \epsilon_{prune}$

//параметры обучения: максимальная длина контекста,

//порог для остановки ЕМ, порог для обрезания дерева

Result: Λ // параметры СММПП

$\Lambda = \text{Инициализация}(Y, m)$;

// полное контекстное дерево

while контекстное дерево уменьшается **do**

$\Lambda = EM(Y, \Lambda, \epsilon_{EM})$;

 // максимизируем правдоподобие модели на наблюдениях

 //при фиксированной структуре дерева

$\Lambda = \text{Подрезание}(\Lambda, \epsilon_{prune})$;

 // подрезаем дерево, если обученные распределения при этом несильно

 // изменяются

end

Algorithm 1: Схема обучения СММПП

Конкатенацию строк a и b будем обозначать ab .

В качестве параметра модели π будем брать оценку вероятностного распределения на контекстах и использовать его в этом же смысле.

2.3.1. Инициализация

Начальное контекстное дерево является полным $|S|$ -нарным деревом глубины $m+1$ (соответствует СММ порядка m).

В качестве начальных распределений переходов и параметров испусканий берутся оценки этих величин на цепи состояний, полученной алгоритмом k -means ($k = |S|$) по цепи наблюдений Y .¹

2.3.2. EM (Expectation–Maximization algorithm)

Оценки параметров проводятся подобно алгоритму Баума-Велша для СММ [4].

1. Е-шаг (Expectation)

Введем дополнительный параметр α

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

$\alpha_t(c)$ — 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 .

$$\begin{aligned} \alpha_0(c) &= \pi(c)b(y_0; c) \\ \alpha_{t+1}(c) &= \sum_{q \in S, c' = C(cq)} \alpha_t(c')a(c[0]; c')b(y_{t+1}; c[0]) \end{aligned} \quad (2)$$

Введем дополнительный параметр β

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

$\beta_t(c)$ — 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 .

$$\begin{aligned} \beta_T(c) &= 1 \\ \beta_t(c) &= \sum_{q \in S, c' = C(cq)} a(q; c)b(y_{t+1}, c'[0])\beta_{t+1}(c') \end{aligned} \quad (3)$$

Введем дополнительный параметр γ

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

$\gamma_t(c)$ — вероятность того, что породив цепь Y моделью СММПП с параметрами

¹В общем случае, EM-алгоритм допускает случайную инициализацию.

Λ , главный контекст скрытого состояния x_t является c .

$$\gamma_t(c) \propto \alpha_t(c)\beta_t(c) \quad (4)$$

2. М-шаг (Maximization)

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

Для пересчета параметра A введем параметр ξ

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

$\xi_t(q; c)$ — 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 .

$$\xi_t(q; c) \propto \alpha_t(c)a(q; c)b(y_{t+1}, q)\beta_{t+1}(qc) \quad (5)$$

Обновление A, π :

$$a(q; c) \propto \sum_t \xi_t(q, c) \quad (6)$$

$$\pi(c) \propto \sum_t \gamma(c) \quad (7)$$

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 $b(. | c) \sim Poisson(\lambda_c)$ пересчет параметров происходит следующим образом:

$$\lambda_c = \frac{\sum_t \gamma_t(c)y_t}{\sum_t \gamma_t(c)} \quad (8)$$

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

$$P(Y|\Lambda) = \sum_{c \in C} \alpha_T(c) \quad (9)$$

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

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

2.3.3. Подрезание дерева

Если существует внутренний лист контекстного дерева s такой, что

$$\forall q \in S \quad P(sq)KL(sq, s) < \epsilon_{prune} \quad (10)$$

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

$$KL(u, w) = \sum_{q' \in S} P(q'|u) \log \frac{P(q'|u)}{P(q'|w)} \quad (11)$$

расстояния Кульбака-Лейблера для апостериорных распределений.

Если таких листьев не существует, алгоритм заканчивает работу.

Пересчет параметров A , π на новых контекстах:

$$a(q; c_{new}) = \frac{\sum_{c \in C(s)} a(q; c) \pi(c)}{\sum_{q' \in S} \sum_{c \in C(s)} a(q'; c) \pi(c)} \quad (12)$$

$$\pi(c_{new}) = \sum_{c \in C(c_{new})} \pi(c) \quad (13)$$

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.

Для применения вышеописанного алгоритма для обучения СММПП на нескольких выборках были внесены изменения в Е-шаг и М-шаг.

1. Е-шаг

Дополнительные параметры $\alpha^d, \beta^d, \gamma^d, \xi^d$ пересчитываются отдельно на каждой выборке $d \in \{1, \dots, N\}$ по формулам 2, 3, 4, 5, соответственно.

Общая γ — конкатенация γ^d .

$$\gamma = [\gamma^1, \dots, \gamma^N] \quad (14)$$

2. М-шаг

$$a(q; c) \propto \sum_d \sum_t \xi_t^d(q; c) \quad (15)$$

$$P(\{Y^1 \dots Y^N\}|\Lambda) = \prod_d P(Y^d|\Lambda) \quad (16)$$

2.5. Сравнение

Чем больше параметров у модели, тем лучше она подстраивается под данные, и тем проще переобучается. Поэтому, 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)

$$AIC = 2k - 2 \log L \quad (17)$$

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

$$\begin{aligned} k &= [\text{количество степеней свободы } A] + [\text{количество степеней свободы } B] \\ &\quad + [\text{количество степеней свободы } \pi] \\ &= l(n-1) + n + (l-1) \\ &= nl + n - 1 \end{aligned} \quad (18)$$

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 версии 3.4.

Python является выразительным, но местами достаточно медленным языком программирования, по этому критические по производительности места (Е-шаг) были перенесены на Cython. Cython является расширением для языка программирования Python с дополнительными синтаксическими конструкциями, предназначенными для статического объявления типов, достигает эффективность С-языков.

С использованием библиотеки Joblib было распараллелено выполнение Е-шага на нескольких выборках по потокам. Для эффективной работы с матрицами были использованы библиотеки NumPy, SciPy. Для отрисовки деревьев использовалась библиотека Pygraphviz. Все графики строились с помощью библиотеки Matplotlib.

4. Применение

4.1. Применение к симмулированным данным

Проверка работы алгоритма обучения происходила следующим образом:

1. генерировались параметры Λ начальной модели СММПП;
2. порождались несколько выборок Y из заданной модели;
3. находились новые параметры $\hat{\Lambda}$ путем обучения модели на порожденных выборках;
4. сравнивались полученные параметры $\hat{\Lambda}$ с реальными Λ , считалось абсолютное среднее отклонение параметров реальной модели от предсказанных параметров.

Ниже проиллюстрирована работа алгоритма на простых моделях: Пуассоновская смесь (СММ нулевого порядка) и СММ (СММ первого порядка). Также проиллюстрирован более интересный случай: СММПП, не являющейся СММ фиксированного порядка.

Для каждого теста было сгенерировано по 100 выборок длиной 5000. Значения параметров алгоритма обучения были выбраны следующими: максимально-возможный порядок $m = 4$, порог для обрезания $\epsilon_{prune} = 0.007$, порог для остановки ЕМ $\epsilon_{EM} = 0.01$

Были посчитаны абсолютное среднее отклонение (MAE — Mean Absolute Error) полученных распределений переходов от реальных распределений, отклонение параметров Пуассоновского испускания λ от реальных, оценки \hat{fdr} , \hat{fndr} (формулы 20, 21, соответственно) [6].

Средние значения этих величин, полученных обучением модели на разных выборках при фиксированной начальной модели, показаны в соответствующих таблицах (4.1.1, 4.1.2, 4.1.3).

fdr — математическое ожидания доли ложных единиц среди всех предсказанных единиц (false discovery rate), $fndr$ — математическое ожидания доли ложных нулей среди всех предсказанных нулей (false nondiscovery rate).

$$fdr = E \frac{FP}{P}, \quad fndr = E \frac{FN}{N} \quad (19)$$

где

FP (False Positive) — количество неправильно идентифицированных единиц (количество состояний 0, предсказанных как 1).

P (Positive) — количество предсказанных единиц.

FN (False Negative) — количество неправильно идентифицированных нулей,

N (Negative) — количество предсказанных нулей

Оценки \hat{fdr} , \hat{fndr} :

$$\hat{fdr} = \frac{\sum_t I[x_t]P(x_t == 0)}{\sum_t I[x_t]} \quad (20)$$

$$\hat{fndr} = \frac{\sum_t I[x_t == 0]P(x_t == 1)}{\sum_t I[x_t == 0]} \quad (21)$$

4.1.1. Пуассоновская смесь

Модель смеси была выбрана с параметрами переходов, изображенных контекстным деревом на рисунке 3 и параметрами испусканий

$$B = \{Poisson(\lambda = 2), Poisson(\lambda = 10)\}$$

Сравнение полученных в ходе обучения параметров с реальными, \hat{fdr} , \hat{fndr} показаны в таблице 4.1.1. Пример предсказанного дерева изображен на рисунке 4.

	$MAE(A, \hat{A})$	$MAE(\lambda, \hat{\lambda})$	\hat{fdr}	\hat{fndr}
<i>Среднее по тестам</i>	0.07	0.21	0.01	0.02

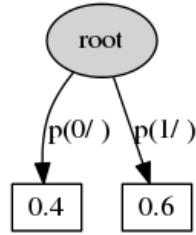


Рис. 3: Реальное дерево

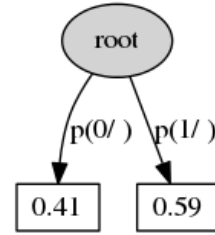


Рис. 4: Предсказанное дерево

4.1.2. СММ

Модель СММ была выбрана с параметрами переходов, изображенных контекстным деревом на рисунке 5 и параметрами испусканий

$$B = \{Poisson(\lambda = 1), Poisson(\lambda = 8)\}$$

Сравнение полученных в ходе обучения параметров с реальными, \hat{fdr} , \hat{fndr} показаны в таблице 4.1.2. Пример предсказанного дерева изображен на рисунке 6.

	$MAE(A, \hat{A})$	$MAE(\lambda, \hat{\lambda})$	\hat{fdr}	\hat{fndr}
<i>Среднее по тестам</i>	0.09	0.21	0.02	0.02

На рисунке 7 изображен график обучения. Каждая ЕМ-часть выделена контуром, сверху которого написано число контекстов на момент обучения, снизу количество итераций в этой части и параметры распределения испусканий, полученных на последней итерации, внутри — график логарифма правдоподобия по итерациям ЕМ. На такой схеме видно, как сначала алгоритм 6 итераций ЕМ обучался на 16 контекстах, после чего дерево подрезалось до 2 контекстов. Следующему ЕМ не удалось значительно увеличить правдоподобие модели, поэтому на третьей итерации он закончил работу. Далее дерево не удалось подрезать, поэтому весь алгоритм закончил свою работу (это видно из отсутствия следующего контура под график ЕМ).

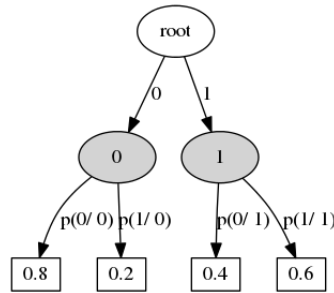


Рис. 5: Реальное дерево

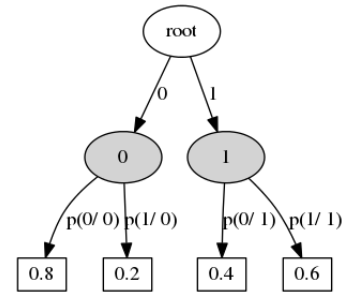


Рис. 6: Предсказанное дерево

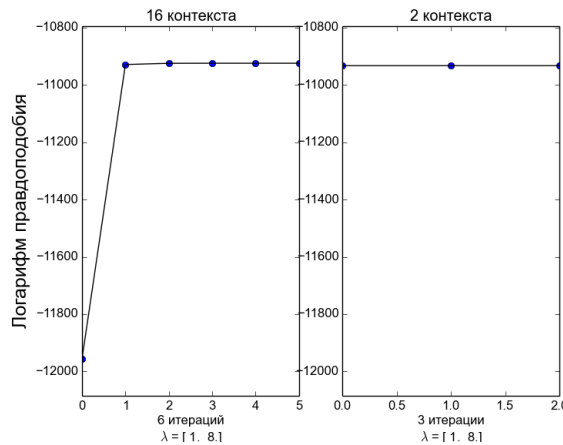


Рис. 7: График обучения

4.1.3. СММПП, не являющаяся СММ фиксированного порядка

Модель СММПП была выбрана с параметрами переходов, изображенных контекстным деревом на рисунке 8 и параметрами испусканий

$$B = \{Poisson(\lambda = 3), Poisson(\lambda = 15)\}$$

Сравнение полученных в ходе обучения параметров с реальными, $\hat{f}dr$, $\hat{f}n\hat{d}r$ показаны в таблице 4.1.3. Пример предсказанного дерева изображен на рисунке 9. На рисунке

10 представлен график обучения модели на одном из тестов.

	$MAE(A, \hat{A})$	$MAE(\lambda, \hat{\lambda})$	$\hat{f}dr$	$\hat{f}ndr$
Среднее по тестам	0.11	0.20	0.02	0.02

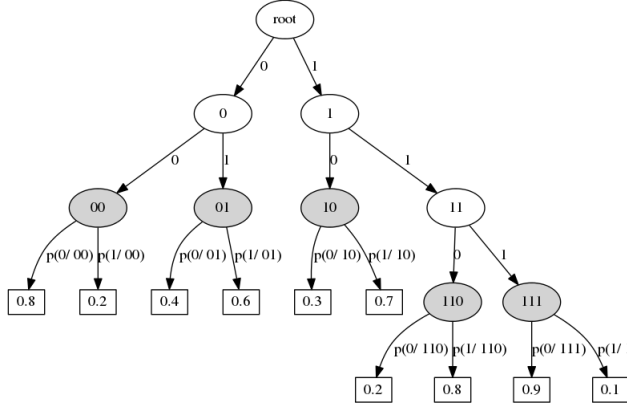


Рис. 8: Реальное дерево

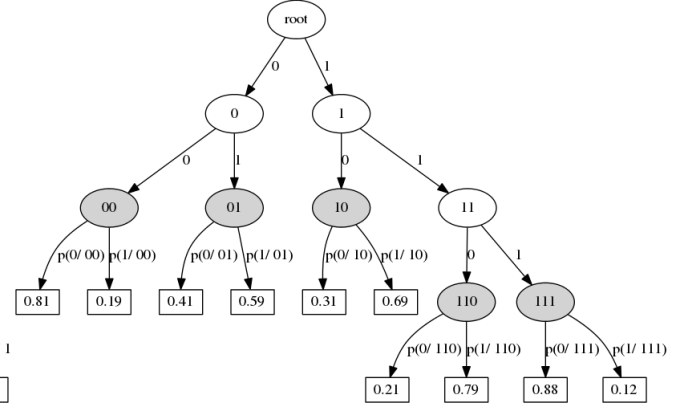


Рис. 9: Предсказанное дерево

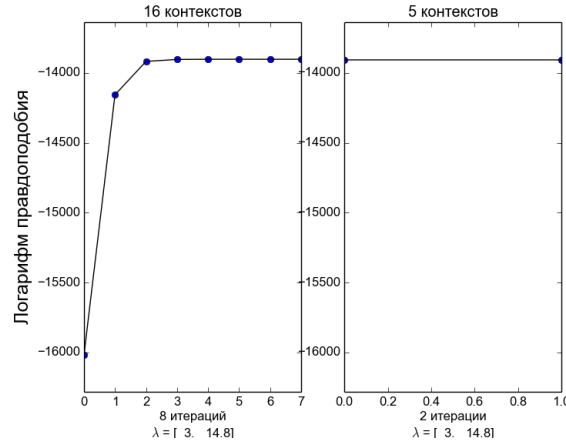


Рис. 10: График обучения

4.2. Применение к реальным данным

Для оценки модели на реальных данных использовались данные из проекта ENCODE (ENCyclopedia of DNA Elements). В качестве изучаемого белка рассматривался гистон H3 с ацетилированным лизином в 27-й позиции. Исследуемые клетки — эмбриональные стволовые клетки человека [1]. Размер окна был выбран равным 200 п.н.

В качестве выборок брались ненулевые участки массива, полученного после деления результата эксперимента ChIP-seq на окна.

Ниже приведены результаты обучения на данных четвертой хромосомы (сумма длин обучающих выборок $\sim 10^5$).

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

Из графика обучения (рис. 11) видно, как 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.

Полученное контекстное дерево переходов для скрытого слоя состояний, отвечающих за ДНК-белковую связь проиллюстрировано на рисунке 12. Из него видно каким образом наличие взаимодействия с белком в фиксированном окне генома определяется взаимодействиями в предшествующих окнах.

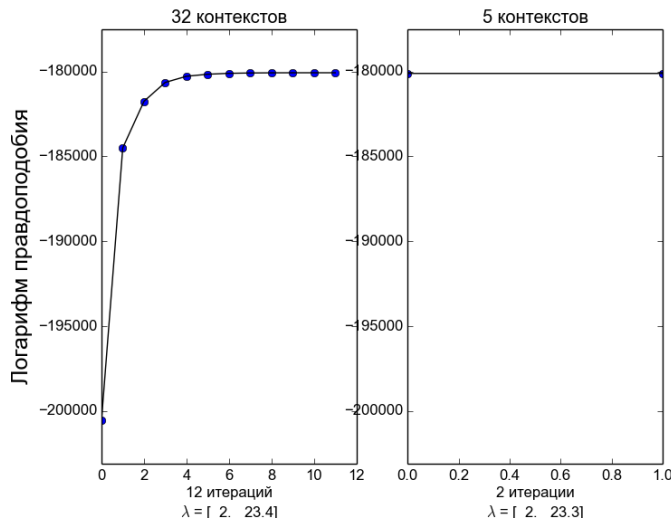


Рис. 11: График обучения

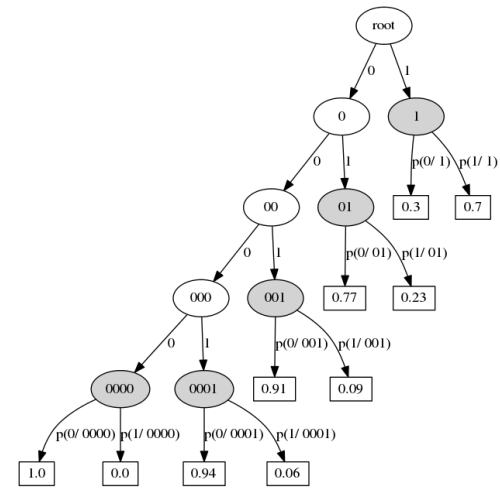


Рис. 12: Контекстное дерево

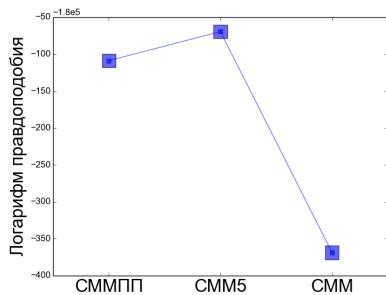


Рис. 13: Сравнение логарифма правдоподобия

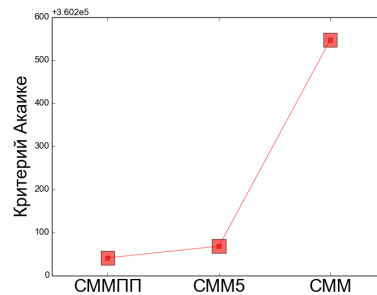


Рис. 14: Сравнение критерия Акаике

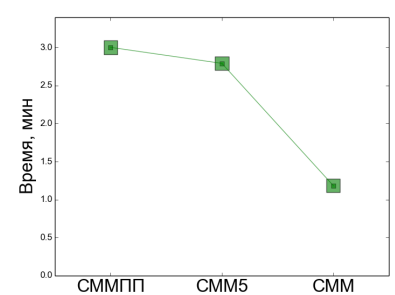


Рис. 15: Сравнение времени обучения

Ниже представлено сравнение логарифма правдоподобия (рис. 13), критерия Акаике (рис. 14) и времени обучения (рис. 15) для моделей СММПП, СММ5 (СММ 5-го порядка, соответствует дереву, с которого начиналось обучение СММПП) и СММ (СММ 1-го порядка, именно его чаще всего используют для анализа данных ChIP-seq).

По критерию Акаике выигрывает СММПП (напомним, что данный критерий, чем меньше, тем лучше).

СММ5 имеет лучшее среди этих трех моделей правдоподобие, однако ее губит большое количество параметров. СММ имеет меньшее среди данных моделей количество параметров, однако ее правдоподобие совсем невелико.

Рисунок 12 показывает сравнение времени обучения. Тут СММПП дает похожий результат с СММ5, немного ей уступая. СММ, в силу простоты структуры, обучается быстрее всех.

Заключение

В ходе работы были решены поставленные задачи:

1. реализована СММПП, подходящая под данные ChIP-seq;
2. проведен анализ эффективности работы СММПП на синтетических данных;
3. осуществлено применение СММПП к данным ChIP-seq, проведено сравнение СММПП на реальных данных с более простыми моделями (СММ первого порядка, пятого). Реализованная модель дала лучший результат по критерию Акаике.

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 Tavaré S Spyrou C, Stark R. BayesPeak: Bayesian analysis of ChIP-seq data. 2009.
- [6] Wenguang Sun and T. Tony Cai. Large-scale multiple testing under dependence. *J. R. Statist. Soc. B*, pages 393–424, 2009.
- [7] Y Wang, Lizhu Zhou, and Jianhua Feng. Mining complex time-series data by learning Markovian models. In *International Conference on Data Mining*, 2006.
- [8] Jérôme Eeckhoutte 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.