УДК 577.71:528.854

ПРИМЕНЕНИЕ МЕТОДОВ РАСПОЗНАВАНИЯ ОБРАЗОВ В ЗАДАЧАХ МОЛЕКУЛЯРНОЙ БИОЛОГИИ[1]

А. Я. Червоненкис

Институт проблем управления им. В. А. Трапезникова, г. Москва

Отмечено, что новым полем применения методов распознавания образов в последнее время стали задачи молекулярной биологии — белки представляют собой последовательное соединение аминокислот, тогда как свойства молекулы ДНК определяются последовательностью нуклеотидных пар; поэтому задачи классификации белков, выделение фрагментов генома и предсказание их функций могут рассматриваться как задачи распознавания слов в заданном алфавите, которые в свою очередь оказываются подзадачами в задаче расшифровки структуры белка и генома и предсказания их функций. Предложен новый вид ядерной функции, используемой далее в машине опорных векторов для обучения распознаванию слов, и приведены результаты сравнения этого метода с другими на примере двух задач распознавания фрагментов генома. Работа выполнена в Лондонском университете (Royal Holloway University of London).

ВВЕДЕНИЕ

В последнее время методы распознавания образов все чаще применяются для решения задач молекулярной биологии. К числу таких задач относятся, например, определение функциональных свойств белков по их структуре (первичной, вторичной или третичной), выделение в геноме отдельных участков, ответственных за управление теми или иными молекулярно-биологическими процессами.

Состав белка однозначно определяется линейной последовательностью аминокислот. Всего имеется 20 аминокислот, и поэтому линейная (первичная) структура белка задается словом в алфавите из 20 символов. Длина слова варьируется от нескольких сот до нескольких тысяч букв. Функциональные свойства белка по преимуществу определяются его вторичной и третичной структурами. Однако, последние в естественных условиях однозначно определяются первичной структурой. Таким образом, вся информация о свойствах белка закодирована словом, задающим его первичную структуру. Для ряда белков путем длительных и дорогостоящих экспериментов удается установить их функциональные свойства. Свойства новых белков, а также таких, для которых эксперименты не проводились, приходится оценивать по сходству с теми, для которых свойства установлены экспериментально. Если сгруппировать белки в классы по их функциональным свойствам, то отнесение нового (или неисследованного) белка к тому или иному классу оказывается типичной задачей распознавания образов. При этом белки с установленными свойствами могут служить обучающей выборкой.

Аналогичные задачи возникают при исследовании генома. Свойства молекулы ДНК определяются последовательностью пар нуклеотидов. Всего (у всех организмов) встречаются четыре таких пары, которые кодируются символами *A*, *C*, *G* и *T*. Таким образом, каждая молекула ДНК может рассматриваться как слово в алфавите из четырех букв. В таком слове содержится до нескольких миллионов букв.

В настоящее время молекулы ДНК многих организмов интенсивно расшифровываются [1, 2]. После того, как экспериментально установлена последовательность нуклеотидных пар (т. е. проведено секвенирование) нужно понять, как эта молекула работает. Для этого на ней нужно, прежде всего, выделить последовательность генов — участков, содержащих коды соответствующих белков, разделенных межгенными промежутками.

При работе в организме гены сначала транскрибируются в молекулы мРНК, затем в молекулы РНК, после чего транслируются в белки. Но не весь код, транскрибируемый в РНК, служит для задания белка. Гены эукариотов содержат значительные участки, которые не транскрибируются в белок. Кодирующие зоны внутри

[1] Работа доложена на Научных чтениях памяти профессора А. М. Петровского, Москва, Ин-т проблем управления, 17 марта 2005 г.

гена называют *эксонами*, а некодирующие — *интронами*. Соответствующие белковые комплексы "распознают" начало и конец интрона и исключают его при очередном копировании. Этот процесс называется *сплайсингом*. После того, как из слова, кодирующего ген, исключены интроны (а также участок между началом транскрипции и началом трансляции), по нему можно прочитать первичную структуру белка, закодированного в гене. Известно, что каждой аминокислоте соответствует триплет нуклеотидных пар (слово из трех букв в алфавите *A*, *C*, *G* и *T*). Таким образом, задача выделения кодирующей части гена, т. е. разделение его на эксоны и интроны, оказывается весьма актуальной.

В то же время по известным белкам, порождаемым данной группой генов, можно решить обратную задачу — указать эксоны, кодирующие эти белки. Случаи с известным разделением на эксоны и интроны можно использовать в качестве обучающей последовательности в задаче обучения распознаванию точек перехода от эксона к интрону и наоборот.

Гену предшествует (или частично захватывает его начало) так называемая промоуторная зона, содержащая промоуторы — участки ДНК, к которым присоединяются белковые комплексы, управляющие работой гена. Они включают или выключают, усиливают или ослабляют экспрессию гена. Поэтому задача распознавания промоуторов также оказывается весьма актуальной. Белковые комплексы своими средствами "распознают" соответствующие промоуторы, и, значит, эта задача должна быть разрешима и машинными средствами. В ряде случаев положение промоуторов установлено экспериментально. Эти примеры могут служить обучающей выборкой.

Существует много других участков (сайтов), регулирующих функционирование генома, по отношению к которым можно поставить аналогичные задачи распознавания. Разумеется, при расшифровке генома все эти задачи должны решаться в комплексе. Но распознавание отдельных участков может рассматриваться как система элементарных блоков (кирпичей) при решении общей задачи.

Различные методы обучения распознаванию образов связаны с соответствующей метрикой, определяющей степень близости объектов. Это может быть простая евклидова метрика, метрика, задаваемая положительно определенной квадратичной формой или ядерной функцией, а также иные метрики. В методах, основанных на построении разделяющей поверхности, метрика используется неявно — через соответствующую форму скалярного произведения, а в методе ближайшего соседа — явно. В последнее время появилось (и появляется) все больше работ па применению методов распознавания образов в молекулярной биологии с применением всех указанных средств [1—4].

Применительно к текстовым строкам (при кодировании строки вектором) простая Евклидова метрика соответствует числу несовпадающих букв в двух словах; метрика, задаваемая квадратичной формой, — степени различия с учетом различных штрафов, зависящих от конкретных букв и их положения в слове, а ядерные функции — более сложным схемам сравнения двух строк.

Близкие по функциям белки, как правило, произошли в ходе эволюции от одного прародителя: одни аминокислоты заменялись на другие, некоторые звенья выпадали, вставлялись новые. Возможна также перестановка, вставка или удаление больших фрагментов. Сказанное относится также к фрагментам генома, выполняющим сходные функции, поскольку они "распознаются" одинаковыми или сходными белковыми комплексами.

Если бы дело ограничивалось только заменами, то близость слов, кодирующих белки (или фрагменты генома), можно было бы мерить числом несовпадений или штрафной функцией с учетом различных замен. Но при наличии вставок и выпадений нужна более сложная процедура. Дело в том, что выпадение хотя бы одной буквы сдвигает остаток слова, и, даже если он полностью сохраняется, число совпадений будет мало. В связи с этим уже давно разработан так называемый метод выравнивания [3] (allignement). Назначается матрица штрафов и штраф (или штрафы) за пропуск буквы в одном из слов. Далее методом динамического программирования ищется такая последовательность пропусков и соответствий, которая обеспечивает минимум суммы штрафов. Метод, основанный на этих идеях, реализован в стандартной программе BLAST [4], используемой для классификации белков и других молекулярных цепей.

1. ЯДРО ВЫРАВНИВАНИЯ

Функция штрафа может интерпретироваться в вероятностных терминах. Представим себе, что некоторый генератор порождает одновременно два слова. Каждый такт генерируется или одновременно по одной букве в каждом слове с вероятностью $P(a, b)$, где a и b — переменные, пробегающие алфавит, $P(a, b) = P(b, a)$, или в одном слове генерируется буква, а в другом — пробел с вероятностью $\pi(a)$. После этого пробелы убираются и получаются два слова **x** и **y**. Зная только результат, можно по-разному расставлять пробелы в полученные слова. Для каждой расстановки пробелов можно подсчитать вероятность получения такой комбинации: она будет равна произведению вероятностей порождения соответствующих пар (буква — буква, буква — пробел, пробел — буква). Если за штраф, соответствующий паре, принять $S(a, b) = -\log P(a, b)$, $S(a,\_) = S(\_, a) = -\log\pi(a)$, то суммарный штраф будет равен взятому со знаком минус логарифму вероятности данной расстановки пробелов в словах **x** и **y**.

Обычный метод выравнивания соответствует поиску наиболее вероятной расстановки пробелов. Чем выше эта вероятность, тем ближе слова. Однако более естественно в качестве меры близости взять не вероятность наиболее правдоподобной расстановки пробелов, а вероятность $P(\mathbf{x}, \mathbf{y})$ одновременного порождения слов **x** и **y**. Она будет равна сумме вероятностей всех вариантов расстановки пробелов. C. Watkins [5] предложил использовать вероятность $P(\mathbf{x}, \mathbf{y})$ в качестве ядра в методах распознавания. В той же работе показано, что такая ядерная функция будет положительно определенной.

Однако полный перебор всех вариантов расстановки пробелов требует комбинаторно большого числа шагов. Нами был предложен быстрый вариант вычисления вероятности $P(\mathbf{x}, \mathbf{y})$, который строится по схеме, близкой к динамическому программированию [1]. Кроме того, непосредственное использование вероятности $P(\mathbf{x}, \mathbf{y})$ в качестве ядра имеет ряд недостатков. Во-первых, эта величина очень резко зависит от длины слов, что сущест-

венно при сравнении слов разной длины. Во-вторых, она очень быстро убывает при уже небольших различиях в словах. Поэтому, чтобы привести ядро к виду, удобному для применения, предложено провести следующие операции:

- нормировка: $K^*(\mathbf{x}, \mathbf{y}) = (P(\mathbf{x}, \mathbf{y}))^{1/l}$; $K^{**}(\mathbf{x}, \mathbf{y}) = K^*(\mathbf{x}, \mathbf{y})/(K^*(\mathbf{x}, \mathbf{x})\ K^*(\mathbf{y}, \mathbf{y}))^{1/2}$;
- расширение: $K(\mathbf{x}, \mathbf{y}) = (K^*(\mathbf{x}, \mathbf{y}))^q$, где l — дина слова, q $(0 < q \leqslant 1)$ — параметр ядра.

Изменяя параметр q, можно "расширять" или "сужать" ядро и соответственно менять меру близости между словами.

Предлагаемое ядро (ядерная функция) получило название *ядра выравнивания* (allignement kernel). Это ядро предлагается использовать в рамках машины опорных векторов (SVM) [6], но его можно применить и как меру близости в алгоритме ближайшего соседа.

С применением этого ядра и машины опорных векторов решались две задачи: распознавание промоуторов σ^{70} у бактерии *E-coli* (*Eschericha coli*, кишечная палочка) и распознавание переходов "эксон—интрон" и "интрон—эксон" (сайтов сплайсинга) у растения *Arabidopsis thaliana*. Эти виды были выбраны потому, что их геномы полностью секвенированы (экспериментально установлена последовательность нуклеотидных пар) и достаточно аннотированы, чтобы составить обучающую и тестовую выборки. Предлагаемый метод сравнивался на том же материале с другими, как использующими SVM (но с другими ядрами), так и не опирающимися на SVM.

2. РАСПОЗНАВАНИЕ ПРОМОУТОРА σ^{70} У БАКТЕРИИ *E-COLI*

Бактерия *E-coli* (прокариот) имеет одну кольцевую хромосому, которая в настоящее время полностью секвенирована, т. е. у нее полностью установлена последовательность нуклеотидных пар. Иными словами, для нее найдено кольцевое слово, кодирующее эту последовательность. В ней экспериментально или по косвенным признакам выделено приблизительно 4000 генов. Работа примерно 80 % из них регулируется белковым комплексом (РНК-полимеразой), содержащим белок σ^{70}. Этот белок присоединяет РНК-полимеразу к соответствующему промоутору (участку ДНК), предшествующему гену, называемому σ^{70}-промоутором. После этого РНК-полимераза запускает транскрипцию гена. Гены, запускаемые σ^{70}-промоутором, ответственны за рост бактерии. Считается, что положение σ^{70}-промоутора тесно связано с точкой начала транскрипции и может служить для ее определения. Иногда один промоутор запускает транскрипцию сразу целой цепочки генов. Такая цепочка называется опероном. В то же время некоторые гены внутри оперона могут запускаться своим промоутором.

Мы собрали вместе список экспериментально установленных точек начала транскрипции (ТНТ), соединив данные из RegulonDB [7] и PromEC [8]. За промоуторную зону принимался участок, содержащий 60 нуклеотидных пар (букв слова), предшествующих ТНТ, и 19 пар, следующих за ней. Считается, что σ^{70}-промоутор заведомо находится внутри этой зоны.

Задача состояла в том, чтобы научиться отличать такие участки от случайно выбранного участка ДНК. В качестве положительных примеров (как для обучения, так и для теста) использовались упомянутые участки, содержащие экспериментально установленную точку начала транскрипции и σ^{70}-промоутор. В качестве отрицательных — случайно выбранные участки хромосомы той же длины, при условном предположении, что ТНТ в них, как и в положительных примерах, расположена в 61-й позиции слова. Поскольку частотный состав нуклеотидных пар различен в кодирующей и некодирующей частях ДНК, было сформировано два блока отрицательных примеров — один из участков, отобранных в кодирующей части, другой — из межгенных промежутков. Всего было использовано 699 слов по 80 букв в качестве положительных примеров, 709 слов из кодирующей и 709 слов из некодирующей частей как отрицательные примеры.

Для практического применения эта задача должна решаться, конечно, в комбинации с другими задачами (выделение кодирующей части гена, поиск точки начала трансляции, определение промоуторов другого типа и др.). Но для сравнения различных методов интересно было исследовать "чистую" задачу распознавания двух классов слов.

Для обучения и проверки использовались все положительные примеры и одна из двух групп отрицательных примеров. Затем случайно отбиралась половина примеров на обучение и столько же на экзамен. Такое случайное разбиение повторялось 100 раз, и результаты были усреднены по всем 100 экспериментам.

Сравниваемые методы обучения можно разделить на две группы: универсальные методы и методы, использующие специальные свойства расположения σ^{70}-промоутора относительно ТНТ, о которых будет сказано ниже.

В первую группу были, прежде всего, включены применение SVM с описанным выше ядром выравнивания (ЯВ + SVM) и алгоритм, использованный в стандартной программе BLAST. Последняя вычисляет меру близости двух слов, исходя из наиболее вероятной расстановки пробелов. Новое слово сравнивается со всеми словами из обучающей выборки, находится ближайшее и выясняется, к какому классу оно было отнесено по данным обучения. Новое слово относится к этому же классу. Оба этих метода учитывают, что при сравнении двух слов нужно принимать во внимание возможные вставки и пропуски, но ЯВ учитывает суммарную вероятность всех возможных вариантов расстановки пробелов, тогда как программа BLAST учитывает только наиболее вероятную из них.

Еще один метод основан на том, что частоты встречаемости отдельных букв и их комбинаций по две, три и более существенно отличаются в промоуторной зоне и в случайно выбранном участке ДНК [9]. Более того, эти частоты оказываются разными в зависимости от расстояния до точки начала транскрипции. Обозначим этот метод как Зоны + SVM. Слово разбивается на пять зон. В каждой из этих зон по обучающей выборке подсчитываются частоты встречаемости отдельных букв и их сочетаний. Логарифм частоты принимается за оценку буквы или сочетания. Для произвольного слова вычисляется его "правдоподобие" по зонам (т. е. для каждой зоны отдельно) путем сложения оценок правдоподобия действительно встретившихся там букв или сочетаний. По-

лученные пять оценок принимаются за координаты вектора, характеризующего слово. Далее ядро вычисляется как простое скалярное произведение этих векторов и используется в SVM. Этот метод основан не на прямом сравнении двух слов, а на сравнении частот встречаемости букв и их сочетаний.

Два метода из этой группы основаны на прямом применении машины опорных векторов (SVM) без учета вставок и удалений. Слово кодируется вектором путем замены каждой буквы на четыре бинарных признака:

A → (0, 0, 0, 1),
C → (0, 0, 1, 0),
G → (0, 1, 0, 0),
T → (1, 0, 0, 0).

Алгоритм "простое ядро + SVM" (ПЯ + SVM) использует далее полиномиальное ядро вида $(1 + N)^d$, где N — число совпадающих букв в двух сравниваемых словах, d — параметр алгоритма. Наилучшим оказалось значение $d = 3$.

Алгоритм "локально исправленное ядро" (ЛИЯ + + SVM) использует более сложное ядро [10]. Два "окна" бегут параллельно вдоль двух сравниваемых слов. При каждом положении окон попавшее в них содержимое сравнивается с помощью полиномиального ядра, указанного выше. Далее сумма полученных значений (по всем положениям окна) возводится в заданную степень, и результат считается значением ядра. Параметры алгоритма — ширина окна, степень внутреннего полинома, степень, в которую возводится сумма — подбирались.

Другая группа методов основана на специальных свойствах промоуторной зоны, содержащей σ^{70}-промоутор. В работах генетиков [10—13] установлено, что на расстояниях −10 и −35 нуклеотидных пар от точки начала транскрипции находятся два "консервативных" блока, называемых −10 и −35 боксами. "Консервативный" означает, что в них обычно повторяется одна и та же последовательность букв — TATAAT для −10 бокса и TTGAC для −35 бокса. Однако оказалось, что очень часто те или иные буквы заменяются на другие и положение боксов относительно ТНТ может "плавать". Можно составить матрицы, характеризующие вероятность замены буквы на другую в зависимости от ее положения в боксе и кандидата на замену [14, 15]. Если бы эти матрицы и распределение вероятностей возможных сдвигов боксов от их стандартного положения были известны, то в предположении независимости признаков можно было бы вычислить функцию правдоподобия и сравнить ее с такой же функцией для случайно выбранного участка. Но имеющиеся в литературе сведения (для *E-coli*) основаны на меньшем объеме данных, нежели тот, которым располагали мы. Поэтому оценку матрицы вероятностей замены и распределения вероятностей сдвигов мы включили в процесс обучения. Для этого был применен прием максимизации правдоподобия, предложенный в работе [16]. Он состоит в следующем. Задаются некоторой матрицей штрафов (например, известной из литературы) и распределением вероятностей сдвигов боксов (например, равномерным в заданном диапазоне). По этим оценкам находится наиболее правдоподобное положение боксов на всех положительных объектах обучающей выборки. При этом положении подсчитываются частоты встречаемости букв во всех положениях внутри боксов и фактическое распределение вероятности сдвигов. Затем операция повторяется с новой матрицей и новым распределением, пока при очередной итерации боксы во всех примерах не сохранят свое положение. Полученные в результате распределения принимаются за оценку истинных. По ним для произвольного слова вычисляется значение правдоподобия того, что оно действительно содержит промоутор: ищется наиболее правдоподобное положение боксов и перемножаются вероятности букв в заданных позициях и вероятности выбранных сдвигов (или складываются их логарифмы). Порог распознавания находится путем сравнения значений правдоподобия для положительных и отрицательных примеров обучающей выборки. Для распознавания нового объекта вычисляется его оценка правдоподобия и сравнивается с порогом. Этот метод мы обозначаем как "боксы + максимум правдоподобия" (Боксы + МП).

Сложение логарифмов вероятности при вычислении правдоподобия основано на гипотезе о независимости факторов, что, быть может, не соответствует действительности. Поэтому в качестве альтернативы был испытан более сложный прием (Боксы + МП + SVM). Матрицы и распределение вероятностей сдвигов находятся так же, как и в предыдущем методе. Так же находится наиболее правдоподобное положение сдвигов. Но далее логарифмы вероятностей не складываются, а служат координатами вектора, кодирующего слово. Полученные векторы обрабатываются затем машиной опорных векторов с простым ядром.

Более прямой метод применения SVM после того, как найдено положение боксов (Боксы + SVM) состоит в следующем. Как и раньше, оцениваются матрицы замены и вероятности сдвигов боксов. Далее для произвольного слова находится наиболее правдоподобное положение боксов, и их содержимое объединяется в новое слово. Это слово кодируется бинарным вектором, как указано выше, и используется в SVM с простым полиномиальным ядром.

В литературе имеются указания, что на сцепление белка σ^{70} с геномом помимо −10 и −35 боксов влияют также и другие регуляторные сайты (участки), встречающиеся в промоуторной зоне. В качестве таковых нами были использованы типовые последовательности:

IHF: 'WATCAANNNNTTR' (Friedman 1988),

LexA: 'TACTGTATATATATACAGTA'
(Hertz et al. 1990),

CRP: 'AAATGTGATCTAGATCACATTT'
(Wang и Church 1992).

Характерное положение этих сайтов относительно ТНТ не установлено, многие буквы могут заменяться на другие, а случаев, где такие сайты наблюдались экспериментально, мало. Тем не менее, поскольку учет этих участков дает дополнительную информацию, мы испытали схему, близкую к методу Боксы + МП + SVM, обозначив ее "Боксы + Регуляторные сайты + Максимум правдоподобия + SVM" (Боксы + РС + МП + SVM).

В приведенных далее таблицах качество распознавания оценивалось по следующим показателям:

- общее число ошибок (ОШ) — процент ошибочно распознанных примеров среди всех объектов, предъявленных на экзамене;
- пропуск цели (ПЦ) — процент ошибок среди положительных примеров на экзамене;

Таблица 1

Результаты экзамена, усредненные по 100 вариантам разбиения на обучение и тест. Отрицательные примеры из кодирующей части

Метод	ОШ	ПЦ	ЛТ
ЯВ + SVM	**16,5**	**18,5**	14,6
BLAST	34,6	40,9	28,7
Зоны + SVM	21,0	32,2	**10,4**
ПЯ + SVM	29,1	29,8	28,4
ЛИЯ + SVM	19,3	24,9	14,1
Боксы + МП	19,5	24,4	14,8
Боксы + МП + SVM	19,1	23,6	14,8
Боксы + SVM	21,6	23,2	20,0
Боксы + РС + МП + SVM	16,8	22,7	11,3

Таблица 2

Результаты экзамена, усредненные по 100 вариантам разбиения на обучение и тест. Отрицательные примеры из некодирующей части

Метод	ОШ	ПЦ	ЛТ
ЯВ + SVM	18,6	19,0	18,2
BLAST	35,4	40,9	30,2
Зоны + SVM	22,5	33,1	12,5
ПЯ + SVM	33,5	35,4	31,7
ЛИЯ + SVM	23,5	38,8	9,1
Боксы + МП	21,0	28,4	14,0
Боксы + МП + SVM	20,5	25,6	15,7
Боксы + SVM	23,0	23,8	22,2
Боксы + РС + МП + SVM	30,3	25,7	34,6

- ложные тревоги (ЛТ) — процент ошибок среди отрицательных примеров на экзамене.

При сравнении результатов следует помнить, что путем изменения порога ошибки типа "пропуск цели" легко переходят в ошибки типа "ложная тревога". По суммарному числу ошибок, как видно из табл. 1 и 2, предложенный нами метод ЯВ + SVM имеет преимущество перед всеми другими сравниваемыми методами.

3. РАСПОЗНАВАНИЕ ТОЧЕК ПЕРЕХОДА "ЭКСОН—ИНТРОН" И "ИНТРОН—ЭКСОН" У *ARABIDOPSIS THALIANA*

Arabidopsis thaliana — это растение (эукариот), обладающее очень коротким циклом размножения. По этой причине оно интенсивно используется в генетических экспериментах. Геном *Arabidopsis thaliana*, в настоящее время полностью секвенированный, состоит из пяти хромосом и имеет общую длину более 117 миллионов нуклеотидных пар. На текущий момент в этом геноме выделено 4714 генов, расшифрованных "Ceres. Inc.". Только у 20 % генов кодирующая часть непрерывна. У 3703 генов кодирующая часть (эксоны) прерывается некодирующими участками (интронами). Важнейший момент в расшифровке генома состоит в том, чтобы найти точки, где заканчивается эксон и начинается интрон (доноры), и, наоборот, точки, где заканчивается интрон и начинается очередной эксон (акцепторы). Схема распознавания этих точек, установленная для *Arabidopsis thaliana*, видимо, может быть применена и для других растений.

У всех выделенных генов экспериментально (или по косвенным признакам) установлено деление на интроны и эксоны и, соответственно, найдены точки перехода. Все эти данные использовались как положительные примеры в обучающей и тестовой выборках. Более 99 % процентов интронов начинается с пары букв GT и в более чем 99 % случаев перехода от интрона к эксону интрон заканчивается парой букв AG. Конечно, эти признаки не достаточны для распознавания, так как такие двухбуквенные сочетания встречаются достаточно часто в произвольных частях генома. Но поиск точек перехода может быть ограничен этим условием. Поэтому в качестве отрицательных примеров брались случайно выбранные участки ДНК, центрированные вокруг пары букв GT (для доноров) или пары AG (для акцепторов).

Поскольку эксон служит для кодирования белка, то информация о точке перехода должна быть сосредоточена в основном на участке интрона, прилегающем к этой точке. Поэтому для распознавания доноров в качестве положительных примеров брались участки ДНК, включающие в себя 30 нуклеотидных пар (букв) из интрона и 10 пар из эксона, прилегающих к точке перехода. Для акцепторов — 10 пар из эксона и 50 пар из интрона. В качестве отрицательных примеров брались слова аналогичного формата, центрированные соответственно относительно случайно выбранных пар букв GT (для доноров) или AG (для акцепторов).

Всего для обучения распознаванию (как доноров, так и акцепторов) и для экзамена использовались 2000 положительных примеров и 8000 отрицательных примеров на обучение, по 5000 положительных и отрицательных примеров на экзамен.

Буквы (нуклеотидные пары) кодировались, как и в предыдущей задаче, четырьмя бинарными признаками:

$A \to (1, 0, 0, 0)$,
$C \to (0, 1, 0, 0)$,
$G \to (0, 0, 1, 0)$,
$T \to (0, 0, 0, 1)$.

При практическом применении распознавания необходимо учесть, что пары букв GT и AG встречаются в ДНК гораздо чаще, чем фактические точки перехода. С другой стороны, имеется возможность дополнительной проверки правильности распознавания: выделенная в виде эксонов часть гена должна быть сплошным кодом белка, т. е. не должна содержать некодирующих триплетов. Но это выходит за рамки настоящего исследования. Для сравнения различных методов собственно распознавания предлагаемая постановка достаточно удобна.

В качестве фонового метода рассматривалось сравнение правдоподобия гипотезы 1 о том, что данный пример относится к числу положительных, против гипотезы 2, что он относится к классу отрицательных. В этой задаче не известно характерных участков (сайтов). Поэтому просто оценивалась (по частоте встречаемости в обучающей выборке) вероятность $P(a, n)$ встретить букву a в n-й позиции слова для положительных примеров и такая же вероятность $P^*(a, n)$ — для отрицательных. Для нового слова в предположении независимости правдоподобие гипотезы 1 вычислялась как сумма логарифмов $P(a, n)$ по всем позициям слова, где a — буква, фактически встретившаяся в n-й позиции, и правдоподобие гипотезы 2 как сумма логарифмов $P^*(a, n)$. За истинную принималась та гипотеза, у которой прав-

доподобие больше. Этот метод известен как ПЗВМ (позиционно-зависимая весовая матрица).

В табл. 3 приведены результаты этого метода на материале теста.

Этот метод сравнивался с результатами распознавания с помощью машины опорных векторов (SVM) при различных ядрах, а также методом ближайшего соседа в евклидовой метрике (БСЕ) и на основе стандартной программы BLAST.

В качестве ядерных функций использовались описанное ранее ядро выравнивания при наилучшем значении параметров (ЯВ + SVM), локально исправленное ядро при наилучшем значении параметров (ЛИЯ + SVM), простое скалярное произведение $(\mathbf{x}, \mathbf{y})$ (СП + SVM), полиномиальное ядро $((\mathbf{x}, \mathbf{y}) + 1)^d$ при различных степенях d (ПЯ + SVM) и Гауссово ядро (ГЯ + SVM) — $\exp(-q\,|\mathbf{x} - \mathbf{y}|^2)$ при различных значениях параметра q.

Результаты распознавания приведены в табл. 4.

Наилучший результат получен для локально исправленного ядра. Заметим, что все методы, использующие SVM (при нормальном выборе параметров), дали лучший результат, чем другие. Ядро выравнивания на этот раз не дало преимущества. Отметим также, что применение простого скалярного произведения в качестве ядра (т. е. обычный обобщенный портрет) дало результат, близкий к оптимальному. Решающее правило в этом случае может быть записано в форме весовой матрицы. Однако, в отличие от весовых матриц, которые строятся по всему материалу обучения, эта матрица построена только по крайним (опорным) векторам со специально подобранными весами, т. е. по наиболее трудным для распознавания примерам. Это существенно в тех случаях, когда распознавание "трудных" объектов требует иного подхода, чем в среднем.

Таблица 3

Результаты распознавания точек перехода с помощью ПЗВМ

Вид перехода	ОШ	ПЦ	ЛТ
Доноры	7,92	12,84	3,00
Акцепторы	12,73	20,36	5,10

Таблица 4

Результаты экзамена

Метод	Параметр	ОШ	ПЦ	ЛТ
ЯВ + SVM	—	7,41	12,84	1,96
ЛИЯ + SVM	—	5,99	10,04	1,94
СП + SVM	—	6,88	11,62	2,14
ПЯ + SVM	$d = 2$	6,44	10,68	2,20
	$d = 3$	6,56	11,00	2,02
	$d = 4$	7,05	12,38	1,72
	$d = 5$	8,19	15,24	1,14
ГЯ + SVM	$d = 0{,}001$	6,81	11,52	2,10
	$d = 0{,}01$	6,62	11,26	1,98
	$d = 0{,}025$	6,51	11,00	2,02
	$d = 0{,}04$	6,77	11,70	1,84
БСЕ	—	22,24	29,52	14,96
BLAST	—	37,8	63,3	12,3

ЗАКЛЮЧЕНИЕ

Полученные результаты свидетельствуют о высокой эффективности методов распознавания, основанных на применении машины опорных векторов. Дальнейшая работа будет направлена на включение элементов распознавания в более сложные системы, предназначенные для анализа структуры белка и генома.

ЛИТЕРАТУРА

1. *Sequence* Alignment Kernel for recognition of promoter regions / Л. Гордон, А. Червоненкис, А. Гаммерман, И. Шахмурадов, В. Соловьев // Bioinformatics. — 2003. Vol. 20. — P. 1964—1971.
2. *Genom-wide* prokariotic promoter recognition based on Sequence Alignment Kernel / Л. Гордон, А. Червоненкис, А. Гаммерман, И. Шахмурадов, В. Соловьев // Труды конференции IDA2003. Берлин, 2003. В кн. Advances in Intelligent Data Analysis. — Springer-Verlag. Vol. LNCS 2810. — P. 386—396.
3. *Goth O.* An improved algorithm for matching biological sequences // J. Mol. Bio. — 1982. — 162 (3). — P. 705—708.
4. *Gapped* BLAST and PSI-BLAST: a new generation of protein search programs / *S. Altschul, T. Madden, A. Schaffer, et al.* // Nucleic Acids Res. — 1997. — Vol. 25, — P. 3389—3402.
5. *Watkins C.* Dynamic alignment kernels // Advances in Large Margin Classifiers. MA: MIT Press, 2000. Cambridge, P. 39—50.
6. *Vapnik V.* Statistical learning theory. — New-York: Wiley, 1998.
7. *Regulondb* (version 3.0): transcriptional regulation and operon organization in *Escherichia coli* K-12 / H. Salgado, A. Santos-Zavaleta, S. Gama-Gastro, et al. // Nucleic Acids Res., — 2000. — Vol. 28 (1). P. 65, 66.
8. *Promec*: An updated database of *Escherichia coli* mRNA promoters with experimentally identified transcriptional start sites / R. Hershberg, G. Bejerano, A. Santos-Zavaleta, H. Margalit // Nucleic Acids Res. — 2001. 29 (1). — P. 277.
9. *Oppon J., Hide W.* A statistical model for procariotic promoter prediction // In the Nineteenth Workshop on genomic Informatics. — P. 271—273. Poster.
10. *Sholkopf D., Gedergen J., Lapalme G.* Frequency of insertion/deletion, transversion and transition in evolution of 5S ribosomal RNA // Journal of Molecular Evolution. —1996. — N 7. — P. 133—149.
11. *Prinbow D.* Nucleotide sequence of an rna polymerase binding site at an early t7 promoter // Proc. of Natl Acad Sci USA. 1975. — Vol. 72. P. 784—788.
12. *Shaller H., Gray C., Herrmann K.* Nucleotide sequence of an rna polymerase binding site from dna of bacteriophage fd // Proceedings of Natl Acad Sci USA. Vol. 72. P. 737—741.
13. *S*equence of promoter for coat protein gene of bacteriophage fd / *K. Takanami., K. Sigimoto, H. Sugisaki., T. Okamoto* // Nature. — 1976. Vol. 260 (5549). — P. 297—302.
14. *Staden R.* Computer methods to locate signals in nucleic acid sequences // Nucleic Acids Res. — 1984. Vol. 12 (1, pt 2). — P. 502—519.
15. *Harley C., Reynolds R.* Analysis of *E. coli* promoter sequences // Nucleic Acids Res. — 1987. Vol. 15 (5). — P. 2343—2361.
16. *Bailey T., Elkan C.* Unsupervised learning of multiple motives in biopolymers using expectation maximization // Machine learning 21 (1—2). P. 51—80.

☎ *(095) 334-88-20*

E-mail: chervnks@ipu.rssi.ru ❑