06

Построение последовательности нуклеотидов методами машинного обучения в секвенаторе „Нанофор СПС“

© В.В. Манойлов, А.Г. Бородинов, А.И. Петров, И.В. Заруцкий, Б.В. Бардин, А.Ю. Ямановская, А.С. Сараев, В.Е. Курочкин

Институт аналитического приборостроения РАН,
198095 Санкт-Петербург, Россия
e-mail: alex.niispb@yandex.ru

Поступило в Редакцию 12 февраля 2024 г.
В окончательной редакции 14 июня 2024 г.
Принято к публикации 8 июля 2024 г.

Рассмотрены способы обработки информации, включающие в себя методы фильтрации изображений, обнаружения кластеров флуоресценции, оценки параметров сигналов флуоресценции как для одиночных кластеров, так и для кластеров, „наложившихся“ друг на друга, а также методы построения последовательности буквенных кодов нуклеотидов ДНК по интенсивностям сигналов флуоресценции, полученных непосредственно по результатам обработки изображений. В этих методах использованы классификаторы, основанные на машинном обучении. Показано, что в результате выполненной апробации различных моделей машинного обучения к задаче построения последовательности нуклеотидов, полученные результаты показали достаточно высокие показатели качества генетического анализа. Показатели качества по Phred score оказались в диапазоне от 29 до 35 для референсного генома бактериофага Phix174.

Ключевые слова: секвенирование, нуклеиновые кислоты, обработка изображений, повышение качества генетического анализа, машинное обучение.

DOI: 10.61011/JTF.2024.09.58677.35-24

Введение

Важным элементом успешного развития геномного секвенирования является использование современных информационных технологий и математических методов обработки данных для установления различных особенностей в анализируемых нуклеиновых кислотах. В Институте аналитического приборостроения РАН (ИАП РАН) разработан аппаратно-программный комплекс (АПК) для расшифровки последовательности нуклеиновых кислот методом массового параллельного секвенирования („Нанофор СПС“) [1]. Решение задачи по расшифровке генома в АПК разделяется на ряд этапов обработки исходных данных. Одним из важных первоначальных этапов обработки данных является оценка значений интенсивностей сигналов флуоресценции для различных длин волн на кадрах изображения реакционной ячейки для нескольких циклов секвенирования методом синтеза. Такая оценка выполняется по программам обработки изображений, алгоритмы которых описаны в работах [2,3].

Методика массового параллельного секвенирования основана на принципе синтеза ДНК с использованием флуоресцентно меченых нуклеотидов. Процесс начинается с подготовки библиотек, где к фрагментам ДНК прикрепляются специальные адаптеры. Эти фрагменты закрепляются на поверхности реакционной ячейки, образуя плотный массив клонированных цепочек ДНК — кластеров. Таким образом, каждый кластер представляет собой множество копий одного и того же фрагмента ДНК.

Во время секвенирования флуоресцентно меченые нуклеотиды поочередно добавляются к растущим цепям ДНК. Каждый нуклеотид несет уникальный флуорофор, который испускает свет на определенной длине волны при лазерном возбуждении. Полученный флуоресцентный сигнал проходит через светофильтры, настроенные на различные длины волн, соответствующие излучению помеченных нуклеотидов. После прохождения через светофильтры сигнал флуоресценции фиксируется видеокамерами. В секвенаторе установлены четыре видеокамеры, каждая из которых регистрирует сигналы одного из типов нуклеотидов (каналов): аденин (A), цитозин (C), гуанин (G) и тимин (T).

Съемка изображения по каждому из четырех каналов происходит после добавления нуклеотидов к фрагментам ДНК. По завершении регистрации флуоресцентных сигналов по всей длине реакционной ячейки начинается следующий этап, на котором через микроканалы пропускаются реактивы, удаляющие краситель (флуорофор) и останавливающие процесс синтеза. Затем добавляются новые реактивы для начала следующего цикла синтеза.

Процесс повторяется циклически, добавляя нуклеотиды один за другим, фиксируя их сигналы видеокамерами до завершения синтеза всей последовательности.

Программы обработки изображений, разработанные в ИАП РАН, решают задачи по оценке значений интенсивностей сигналов флуоресценции и дальнейшей расшифровке последовательности нуклеотидов (base-

calling), но они имеют ряд недостатков, связанных с неполной коррекцией ошибок, вызванных рядом факторов, искажающих результаты генетического анализа. К таким факторам относятся: изменения в значениях регистрируемых интенсивностей из-за таких явлений как фазирование/префазирование (Phasing/Prephasing), затухание сигнала и перекрестные помехи (Cross-talk) [4,5].

Для нивелирования этих недостатков и проведения генетического анализа без коррекции описанных помех перспективными являются методы машинного обучения (Machine learning, ML), которые рассматриваются в настоящей работе.

Применение ML в задачах секвенирования ДНК включает в себя создание и оценку моделей, использующих алгоритмы, способные распознавать, классифицировать и прогнозировать определенные результаты на основе полученных данных [6]. Подходы ML подразделяются на обучение без учителя (Unsupervised), обучение с частичным привлечением учителя (Semi-supervised), обучение с учителем (Supervised) [7]. Например, часто целью Supervised ML, применяемого к данным секвенирования, является построение модели на основе обучающего набора собранных наблюдений с известной последовательностью нуклеотидов с целью прогнозирования нуклеотида для произвольного образца с неизвестным целевым значением типа определяемого нуклеотида, например с последовательностью нуклеотидов бактериофага Phix174 (референсный геном). Входные переменные часто при этом называют признаками (Features), а соответствующие выборки — наблюдениями (Observations).

ML может сыграть ключевую роль в улучшении точности и скорости проведения этого процесса. Применяется на следующих этапах анализа.

• Предобработка данных. Данные секвенирования включают в себя сырые сигналы от детекторов, такие, как электрические сигналы, флуоресценция или графики интенсивности. ML помогает фильтровать шум и калибровать данные для улучшения качества исходного сигнала.

• Обучение модели. Модели ML обучаются на большом объеме аннотированных данных, где известны правильные последовательности нуклеотидов. Это помогает моделям распознавать сложные паттерны в сигналах, соответствующие различным нуклеотидам.

• Классификация и предсказание. Современные методы ML, особенно класса Deep Learning, такие, как сверточные нейронные сети (CNN) и рекуррентные нейронные сети (RNN), позволяют моделям классифицировать сигналы и предсказывать последовательности нуклеотидов с высокой точностью. Эти модели могут учитывать контекстную информацию и последовательности соседних нуклеотидов для более точного base-calling.

• Коррекция ошибок. Алгоритмы ML могут также использоваться для обоснования и исправления ошибок, возникающих в результате секвенирования. Это включает в себя анализ контекстных частот появления определенных нуклеотидов и использование алгоритмов выравнивания.

• Интеграция с другими биоинформатическими инструментами. Результаты base-calling часто интегрируются с другими инструментами анализа геномных данных для дальнейшей аннотации и интерпретации, что также может включать дополнительные этапы ML, например, для технологий секвенирования Oxford Nanopore Technologies.

В статье [8] был сделан обзор методов машинного обучения для решения задач построения последовательности нуклеотидов и рассмотрены несколько примеров применения ML для обработки данных секвенатора „Нанофор СПС“. Кроме того, рассмотрены различные пути борьбы с проблемой переобучения (Overfitting): посредством регуляризации модели, выбора более простой модели с меньшим количеством параметров и уменьшения размерности пространства признаков для обучения.

Целью настоящей работы является поиск методов обработки изображений сигналов флуоресценции, которые позволяют повысить качество генетического анализа.

Для достижения данной цели нужно решить следующие основные задачи:

1. Проанализировать алгоритмы обработки изображений [2,3] выполняющие фильтрацию изображений, обнаружение кластеров флуоресценции, оценку параметров сигналов флуоресценции как для одиночных кластеров, так и для кластеров, „наложившихся“ друг на друга. Разработать новый алгоритм разделения слипшихся объектов, который позволит увеличить плотность кластеров в анализируемой пробе и тем самым количество оснований нуклеотидов в результатах генетического анализа.

2. На ряде данных секвенирования прибора „Нанофор СПС“ показать перспективность метода ML для решения задачи построения последовательности нуклеотидов. В результате применения различных моделей ML к задаче построения последовательности нуклеотидов попытаться повысить показатели качества генетического анализа.

3. На основе экспериментальных данных прибора „Нанофор СПС“ оценить возможность упрощения моделей ML за счет сокращения размерности признаков для выполнения процесса обучения.

Повышение точности и надежности проведения генетического анализа особенно важно для его применения в биомедицине. Высококачественный base-calling приводит к повышению надежности последующих этапов анализа геномных данных, таких, как выравнивание, аннотация и интерпретация. При ошибках процедуры base-calling могут возникнуть неверные генетические интерпретации, что особенно критично для диагностических исследований.

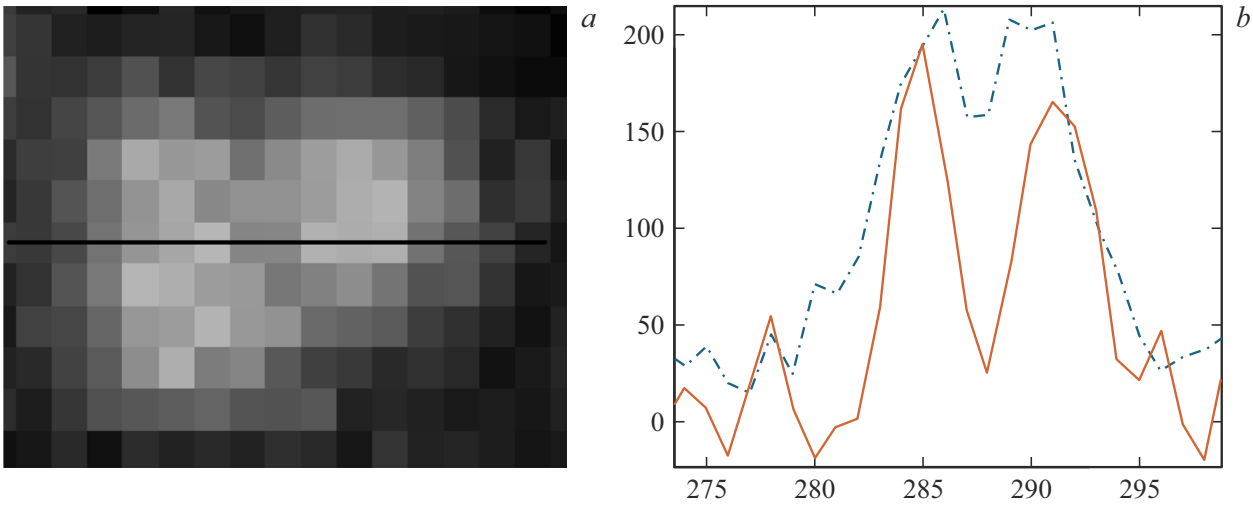

Рис. 1. Сравнение профилей сигналов до и после обостряющей фильтрации. До обработки — штрихпунктирная линия. После обработки — сплошная линия. По оси абсцисс — номера пикселей по горизонтальной оси.

1. Анализ алгоритмов обработки изображений в секвенаторе „Нанофор СПС“

В работе [2] перечислены основные этапы обработки изображений сигналов флуоресценции в приборе „Нанофор СПС“ и описаны алгоритмы их выполнения. Среди этих этапов наиболее важными являются: обостряющая фильтрация (ОФ), обнаружение объектов, коррекция фоновой составляющей и разделение „слипшихся“ объектов.

В разд. 1.1. дается пояснение работы алгоритма ОФ, приводится алгоритм определения порога для обнаружения объектов и описан новый алгоритм разделения „слипшихся“ объектов.

1.1. Алгоритм ОФ

ОФ при обработке изображений позволяет осуществлять сужение объектов для разделения при слипании. Кроме того, ОФ производит восстановление исходного сигнала, искаженного аппаратной функцией [9,10]. Алгоритм ОФ основан на обратном двумерном преобразовании фурье-произведения фурье-образа исходного изображения и фурье-образа производной второго порядка гауссовой функции с шириной, равной примерно половине средней ширины изображения объекта флуоресценции [2,3,10]. Для устранения фоновой составляющей используется алгоритм, описанный в работе [2].

1.1.1. Сигналы флуоресценции до и после ОФ

Приведем пример работы программы ОФ. Для этого построим профиль строки, показанной черной линией, проходящей через пиксель с максимальной яркостью фрагмента изображения, представленного на рис. 1, *a*. На рис. 1, *b* исходный сигнал показан линией с точками, а сигнал после обработки по алгоритму ОФ — сплошной. Из рисунка видно, что в результате ОФ практически слипшиеся два кластера флуоресценции разделились.

1.2. Построение гистограмм для определения порога обнаружения объектов

Для обнаружения объектов флуоресценции на фоне шумов и нахождения координат их центров важным является определение порога, который бы позволил надежно отделить „сигнал“ (объект) от шумов и помех.

Распределения интенсивностей сигналов разных нуклеотидов (A, C, G, T) отличаются друг от друга, и поэтому значения порога для изображений сигналов флуоресценции каждого из нуклеотидов будут разные. Назовем изображения, полученные для сигналов флуоресценции нуклеотидов A, C, G, T, соответственно каналами A, C, G, T. Для определения значений порога для каждого из каналов строятся гистограммы распределения нормированных на максимальное значение интенсивностей сигналов в каждом пикселе. По горизонтальной оси гистограммы отложены интервалы интенсивностей от 0 до 1 с шагом 0.001. На рис. 2 показана гистограмма распределения интенсивностей канала A.

Как видно из рис. 2, распределение интенсивностей сигналов флуоресценции представляет собой одномодальную ассиметричную функцию. Величины интенсивностей, отличные от шума, „вносят вклад“ в ассиметричную часть функции. Определение порога с помощью гистограмм, имеющих ассиметричную функцию распределения, немного сложнее, чем для гистограмм, имеющих двухмодальную функцию распределения, которая описывается в работе Отсу [11]. Для двухмодальных функций

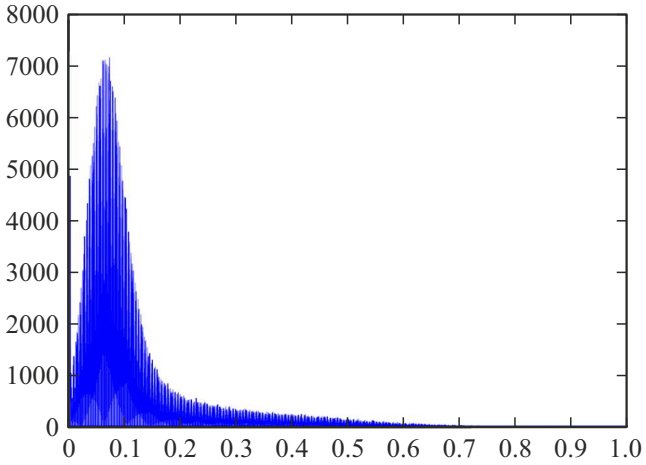

Рис. 2. Гистограмма интенсивностей сигналов в канале A. По оси абсцисс отложены интервалы нормированных интенсивностей. По оси ординат — количество интенсивностей, попавших в определенный интервал.

распределения порог определяется как среднее значение между двумя максимумами в функции распределения. В нашем случае определение порога обнаружения с помощью гистограмм происходило следующим образом. Производилась оценка среднеквадратичного значение (СКО) шума. Оценка СКО шума представляет собой значение полуширины на полувысоте пика гистограммы. Величина порога равна произведению коэффициента k на оценку СКО. Коэффициент k подбирался экспериментально путем обработки большого количества изображений сигналов флуоресценции. Для большинства задач для определения порога пригодным оказался коэффициент $k = 9$.

На основе алгоритмов обработки обостряющей фильтрации, обнаружения и оценки параметров кластеров сигналов флуоресценции в ИАП РАН были разработаны программы, которые были внедрены в опытные образцы приборов „Нанофор СПС“. Анализ результатов работы этих программ показал, что программы надежно обнаруживают кластеры сигналов флуоресценции при минимальной амплитуде полезного сигнала 80 условных единиц и среднеквадратичного значения шума примерно 15 условных единиц. Фоновая составляющая в приборе „Нанофор СПС“, как правило, представляет собой нелинейную функцию со значениями интенсивностей от 50 до 150 единиц. Исходный размер изображения в приборе „Нанофор СПС“ составляет 2000×2400 пикселей. В связи с тем, что ширина кластера на его полувысоте составляет от 6 до 10 пикселей, фоновую составляющую под кластером можно считать линейной. Как отмечалось в работе [2], обостряющая фильтрация практически полностью корректирует влияние фоновой составляющей, что можно также видеть из рис. 1. Отрицательные значения сигнала, возникающие после обостряющей фильтрации, заменяются нулями и не оказывают влияние на качество построения последовательности нуклеотидов.

1.3. Итерационный алгоритм разделения слипшихся объектов

На рис. 3, *a* представлено изображение двух слипшихся объектов в градациях серого.

Суть итерационного алгоритма состоит в следующем. В результате пороговой обработки получается бинарное изображение, в котором области исходного изображения, превышающие порог, заменяются нулями и изображаются черным цветом, а оставшиеся области заменяются единицами и изображаются белым цветом. В бинарном изображении, полученном в результате пороговой обработки, представленном в виде примера на рис. 3, *a*, ищутся области, состоящие из нулей, которые не имеют связей с другими областями, состоящими из нулей. Пример такой области представлен на рис. 3, *b*. Далее используется изображение этой же области, но в градациях серого. В этой области находятся координаты самого яркого пикселя. Затем в бинарном изображении область (обычно 7×7 или 9×9 пикселей) самого яркого объекта заменяется единицами, и эта область становится белой. Таким образом, из изображения двух слипшихся объектов удаляется самый яркий из них и процедура поиска координат, соответствующих объектам флуоресценции, продолжается. Если в исходном изображении были фрагменты, состоящие из трех слипшихся объектов, то сначала определяются и запоминаются координаты самого яркого объекта. Затем этот объект удаляется и анализируется фрагмент, состоящий уже из двух слипшихся объектов и так далее. В рабочей программе секвенатора „Нанофор СПС“ процедура разделения слипшихся объектов продолжается до 5 итераций, т. е. разделяются фрагменты изображений, которые могут содержать от двух до пяти слипшихся объектов, что является достаточным для тех изображений, которые получаются в приборе при разных плотностях объектов флуоресценции.

Применение данного итерационного алгоритма позволило различать кластеры нуклеиновых кислот при более чем двукратном увеличении плотности загрузки обнаруживаемых объектов, что составляет примерно 106 кластеров на квадратный миллиметр реакционной ячейки и зависит от концентрации исследуемой пробы.

При разработке программы разделения слипшихся объектов флуоресценции исследовался также алгоритм „водораздела“ (Watershed), который описан в работе [12]. Реализация этого алгоритма требует большого количества операций и не может быть реализована в режиме реального времени. Предполагается его использование в постобработке для повышения достоверности генетического анализа.

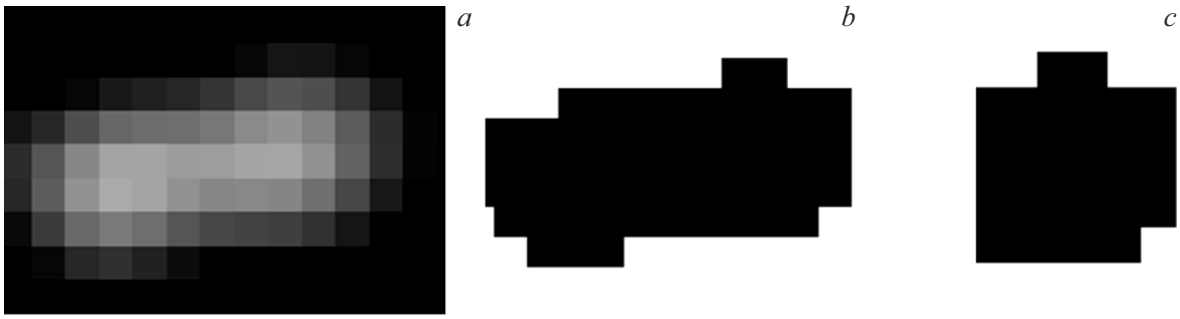

Рис. 3. Пример двух слипшихся объектов и результаты обработки с помощью итерационного алгоритма: *a* — изображение двух слипшихся объектов в градациях серого, *b* — бинарное изображение двух слипшихся объектов после пороговой обработки до начала итерационной процедуры, *c* — бинарное изображение одного из двух слипшихся объектов после первой итерации, в результате которой удаляется более яркий объект.

2. ML в задачах построения последовательности нуклеотидов (base-calling)

2.1. Постановка задачи

Задача base-calling попадает в класс задач классификации, типичных для приложения алгоритмов ML. На изображениях для различных каналов флуоресценции детектируются кластеры амплифицированных цепочек ДНК в виде паттернов различной величины и локализации. С помощью специального программного обеспечения, описанного в работах [2,3], определяется положение пятен и их интенсивностные характеристики вместе с параметрами окружающего фона.

Положения и радиусы пятна используются для извлечения ряда характеристик из каждого пятна и его непосредственного фона. Эти функции затем используются в качестве входных данных для алгоритмов ML. Извлечение признаков выполняется путем изучения интенсивности света каждого пикселя определенной прямоугольной области пятна и некоторого окружающего фона. Каждый набор изображений состоит из четырех микроскопических изображений, по одному для каждого основания. Данные показатели используются моделью ML для задачи классификации оснований. Выходные данные — последовательность нуклеотидов с различными основаниями [13].

Для алгоритма ML из каждого объекта (кластера) флуоресценции и его непосредственного фона извлекаются следующие признаки из изображений сигналов флуоресценции: для фона (BG) — max, mean, median и mode; для центральной зоны изображения кластера (FG) — max, mean, pct90 и pct99, где max — максимальное значение интенсивности, mean — среднее арифметическое значение, mode — наиболее часто встречающееся значение, pct90 и pct99 — 90-й и 99-й процентили соответственно. Указанные признаки образуют строки матрицы M. Таким образом, информация о каждом нуклеотиде содержит 8 признаков (4 признака для FG и 4 признака для BG). В последнем столбце такой матрицы содержится label — буквенный код нуклеотида, полученный из данных заранее секвенированной последовательности, предварительно отображенной на известной (референсной) последовательности, например, бактериофага Phix174. Матрица M имеет 34 столбца: номер кластера, 32 признака для каждого нуклеотида и буквенный код нуклеотида. Количество строк в матрице M определяется количеством кластеров объектов флуоресценции, информация о которых используется для построения обучающей выборки.

Использованы следующие методы классификации для решения задачи base-calling:

- модель персептрона (Perceptron) [14];
- модель логистической регрессии (Logistic regression) [15];
- модель на основе метода опорных векторов (SVM) [16];
- модель деревьев решений (Decision tree) [17];
- модель случайного леса (Random forest) [18];
- модель k-ближайших соседей (k-nearest neighbors) [19].

2.2. Платформа Scikit-learn как база моделей ML

Для применения различных алгоритмов ML используется платформа Scikit-learn, поддерживающая простой в использовании интерфейс, тесно интегрированный с языком Python [20]. API-интерфейс Scikit-learn оптимально спроектирован для работы с методами ML. Ниже перечислены главные проектные принципы согласно работе [21]:

- Согласованность вызова объектов. Все объекты разделяют согласованный и простой интерфейс вызова функций.
- Оценщики (Estimator). Любой объект, который способен проводить оценку параметров на основе набора данных, называется оценщиком (например, imputer,

предназначенный для восстановления данных, является оценщиком). Сама оценка производится с помощью метода fit, принимающего в качестве параметра единственный набор данных (или два для алгоритмов обучения с учителем; второй набор данных при этом содержит метки). Любой другой параметр, необходимый для управления процессом оценки, считается гиперпараметром (например, strategy в imputer) и должен быть указан как переменная экземпляра.

• Трансформеры (Transformer). Некоторые оценщики (такие, как imputer) могут также трансформировать набор данных; они называются трансформерами. API-интерфейс при этом достаточно прост: трансформация выполняется методом transform, которому в параметре передается набор данных, подлежащий трансформации. Он возвращает трансформированный набор данных. Все трансформеры имеют удобный метод fit_transform, который представляет собой эквивалент последовательного вызова fit и затем transform.

• Предикторы (Predictor). Наконец, некоторые оценщики способны вырабатывать прогнозы, имея набор данных; они называются предикторами. Предиктор располагает методом predict, который принимает набор данных с новыми образцами и возвращает набор данных с соответствующими прогнозами. Предиктор также имеет метод score, оценивающий качество прогнозов с помощью указанного испытательного набора и соответствующих меток в случае алгоритмов обучения с учителем.

• Инспектирование. Все гиперпараметры предикторов доступны напрямую через переменные экземпляра (например, imputer.strategy), и все изученные параметры предикторов также доступны через открытые переменные экземпляра с суффиксом в виде подчеркивания (например, imputer.statistics_).

• Нераспространение классов. Наборы данных представляются как массивы NumPy или разреженные матрицы SciPy, вместо самодельных классов. Гиперпараметры — это просто обычные строки или числа Python.

• Легкость композиции. Существующие строительные блоки максимально возможно используются повторно. Например, из произвольной последовательности трансформеров легко создать предиктор Pipeline, за которым следует вызов финального предиктора.

• Стандартные значения параметров по умолчанию. Scikit-learn предоставляет обоснованные стандартные значения для большинства параметров, облегчая быстрое создание базовой рабочей системы.

Для оценки качества работы обученной модели с ранее не встреченными данными мы разделим набор данных на обучающий (train) и тестовый (test) наборы данных. Для этого мы воспользуемся функцией train_test_split из модуля model_selection библиотеки Scikit-learn, производя случайное разделение массивов данных на train и test выборки. В случае задачи классификации в задаче base-calling случайное разбиение данных проводим, выделяя 30% данных для тестирования и 70% для обучения. Специальная функция Scikit-learn train_test_split уже выполняет внутреннее перемешивание обучающих данных перед разделением. В противном случае порядок примеров из различных классов в обучающих наборах данных мог бы иметь критическое значение, что является нежелательным. Путем использования параметра RandomState с фиксированным случайным начальным числом мы гарантируем воспроизводимость результатов при последующих процессах обучения. Кроме того, как правило, во всех методах машинного обучения используем встроенную поддержку стратификации. Стратификация подразумевает, что метод train_test_split возвращает обучающие и тестовые подмножества с такими же пропорциями меток классов, как в исходном наборе данных.

Для оптимальной производительности многие алгоритмы ML и оптимизации требуют масштабирования входных данных. Для этого функции используется класс StandardScaler из модуля предварительной обработки Scikit-learn. Важно использовать одни и те же параметры масштабирования для стандартизации обучающего и тестового наборов данных, чтобы значения в обоих наборах данных были сопоставимы.

В ML существует несколько мер точности, которые используются для оценки производительности и адекватности моделей обучения. В настоящей работе точность классификации в процессе подбора моделей оценивалась по следующим метрикам [22]:

1. Точность ACC (Accuracy). Точность измеряет долю правильных предсказаний модели относительно общего числа предсказаний. $ACC = (TP + TN)/(TP + TN + FP + FN)$, где TP — истинно положительные, TN — истинно отрицательные, FP — ложно положительные и FN — ложно отрицательные предсказания.

2. Полнота (Recall). Показывает, какую долю положительных случаев модель способна правильно обнаружить из всех реальных положительных. $Recall = TP/(TP + FN)$.

3. Точность (Precision). Определяет долю истинно положительных предсказаний относительно всех положительных предсказаний модели. $Precision = TP/(TP + FP)$.

4. F-мера (F1-score). Является средним гармоническим между полнотой и точностью. $F1\text{-}score = 2 \cdot (Precision \cdot Recall)/(Precision + Recall)$.

5. Площадь под ROC-кривой (ROC AUC). Оценивает качество двоичной классификации, измеряя площадь под кривой ROC (Receiver Operating Characteristic). Она показывает, как хорошо модель различает положительные и отрицательные классы.

Эти меры точности использовались во всех исследуемых моделях обучения. В окончательных результатах мы остановились на оценке точности обучения в терминах ошибки классификации (misclassification $error = 1 - ACC$), так как она наиболее точно соответствует Phred quality score (пояснен ниже), являющимся

общепринятой мерой качества идентификации азотистых оснований, полученных автоматическим секвенированием ДНК.

Также во всех методах обучения применялась перекрестная проверка (Cross-validation) как метод оценки модели ML, который позволяет оценить, насколько хорошо модель обобщает данные, — необходимый процесс при работе с ограниченными объемами данных [23]. Для проведения перекрестной проверки данные разбивались на несколько частей, называемых „складками" (Folds) (на практике от 3 до 5 частей). Затем модель обучалась на нескольких комбинациях этих складок и оценивалась на оставшихся частях данных. Путем повторения этого процесса методом перекрестной проверки, для каждой части данных получалась интегральная оценка модели. Перекрестная проверка помогает более эффективно использовать доступные данные для оценки модели и принятия решений относительно ее точности, что особенно актуально в рассматриваемой задаче, в которой данных для обучения критически не хватает.

2.3. Итоги выбора моделей ML

В результате применения различных моделей ML к задаче base-calling полученные результаты по точности предсказания могут быть представлены в сводной таблице. В качестве входных данных использовались значения интенсивностей сигналов флуоресценции кластеров, полученные от системы параллельного секвенирования „Нанофор СПС". Затем на основе этих интенсивностей были извлечены статистические характеристики, такие, как среднее значение, медиана, pct99, pct90 и т.д. При этом коррекции изменений интенсивностей из-за таких явлений как фазирование/префазирование (Phasing/Prephasing), затухание сигнала и перекрестные помехи (Cross-talk) не выполнялись.

В правом столбце таблицы приведены показатели качества по Phred quality score, принятые в биоинформатике. Оценки качества Phred логарифмически связаны с вероятностью ошибок построения последовательности букв нуклеотидов и определяются как

$$Q = -10 \cdot \log_{10} P.$$

Результаты применения различных моделей ML

Модель ML	Ошибка классификации	Phred quality score
Perceptron	0.0008	30.9
Logistic regression	0.0003	35.2
SVM	0.0006	32.2
Decision tree	0.0012	29.2
Random forest	0.0005	33.0
KNN	0.0003	35.2

Это соотношение можно записать как

$$P = 10^{\frac{-Q}{10}}.$$

Например, Phred присваивает букве оценку качества, равную 30. Вероятность того, что эта буква в последовательности была названа неправильно, равна 1 к 1000. Другими словами вероятность правильности буквы равна 99.9%.

Все модели ML продемонстрировали высокую точность предсказания нуклеотидной последовательности в процессе base-calling. Интересно отметить, что достигнутая точность предсказания на данных эксперимента прибора „Нанофор СПС" в значительной степени соответствует Phred quality score, рассчитанному в соответствии с общепринятыми протоколами Illumina [24,25]. Модель Random forest предсказуемо превзошла модель Decision tree. Отличные результаты простой модели KNN, сравнимые с регуляризованной моделью логистической регрессии, указывают на необходимость оптимизации выбора признаков с применением метода понижения размерности.

3. Оптимальный выбор признаков для уменьшения размерности

Как упоминалось ранее, в процессе апробации различных моделей отмечается, что во многих случаях модель демонстрирует значительно более высокую точность на обучающем наборе данных, чем на тестовом наборе, что свидетельствует о переобучении. При использовании библиотеки Scikit-learn переобучение означает, что модель слишком точно подстраивает параметры под конкретные наблюдения в обучающем наборе данных, но плохо обобщает новые данные, что проявляется в высокой дисперсии модели. Основной причиной переобучения является излишняя сложность модели для имеющихся данных обучения. Общие подходы к сокращению ошибки обобщения включают в себя следующее [26]:

- использование большего объема данных в обучающем наборе;
- введение штрафа за сложность с помощью регуляризации модели;
- выбор более простой модели с меньшим количеством параметров;
- уменьшение размерности данных.

Сбор большего количества обучающих данных эффективен, но часто неприменим. В работе рассмотрены распространенные способы уменьшения переобучения путем регуляризации и уменьшения размерности за счет выбора признаков. Это приводит к упрощению моделей путем уменьшения количества параметров. Для модели логистической регрессии параметры регуляризации L1 и L2 использованы как штраф за сложность модели. И если регуляризация L2 использует подход к уменьшению сложности модели за счет штрафов на большие

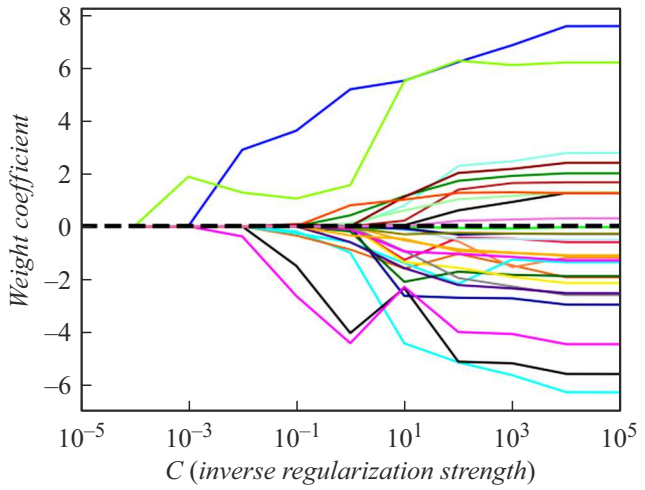

Рис. 4. Демонстрация эффекта регуляризации в виде уменьшения весов логистической регрессии при изменении параметров регуляризации.

отдельные веса для всех параметров, то регуляризация L1 обычно дает разреженные векторы признаков, и веса большинства признаков будут равны нулю. В этом смысле регуляризацию L1 можно понимать, как метод выбора признаков и уменьшения размерности модели. Приведенный график дает дополнительную информацию о поведении регуляризации L1 (рис. 4). На рисунке показано изменение весовых коэффициентов (Weight coefficient) 32 признаков (8 на каждый канал) от обратного значения параметра регуляризации C. Зависимости некоторых признаков от параметра регуляризации C не видны, так как они „слились“ с зависимостью от других признаков. Из информации, представленной на рисунке, видно, что для параметра регуляризации C, меньшего 0.1, весовые коэффициенты не равны нулю только для 4 признаков. Такое положение дало повод для исследований по сокращению пространства признаков.

Эффективным методом оптимального выбора признаков для машинного алгоритма является метод PCA (Principal component analysis) [27]. Преобразование данных, содержащих информацию о сигналах флуоресценции методом главных компонент, и последующая классификация методом k-средних, позволило в этой работе создать оптимальную выборку для идентификации буквенного кода нуклеотида. Метод линейного дискриминантного анализа (LDA) также применим для решения задачи уменьшения размерности. Общая концепция LDA очень похожа на PCA, но, хотя PCA пытается найти ортогональные оси компонентов максимальной дисперсии в наборе данных, цель LDA — найти подпространство признаков, которое оптимизирует разделимость классов. На рис. 5, a показано представление разделение классов по принадлежности к определенному нуклеотиду (4 класса в задаче классификации) при сведении многомерного пространства признаков к двум дискриминантам.

Использование алгоритма PCA и классификация методом k-средних при обработке данных прибора „Нанофор СПС“ показала, что количество ошибочной классификации нуклеотидов не превышало 0.7%.

Наконец, для визуализации многомерных признаков в двумерном пространстве был использован метод t-SNE [28]. Он строит модель данных на основе их попарных расстояний в многомерном пространстве признаков. Затем этот метод находит распределение вероятностей попарных расстояний в новом пространстве более низкой размерности, близкое к распределению вероятностей этих же попарных расстояний в изначальном пространстве (рис. 5, b). Другими словами, t-SNE обучается отображать точки данных в пространство меньшей размерности таким образом, чтобы попарные расстояния в изначальном пространстве сохранялись.

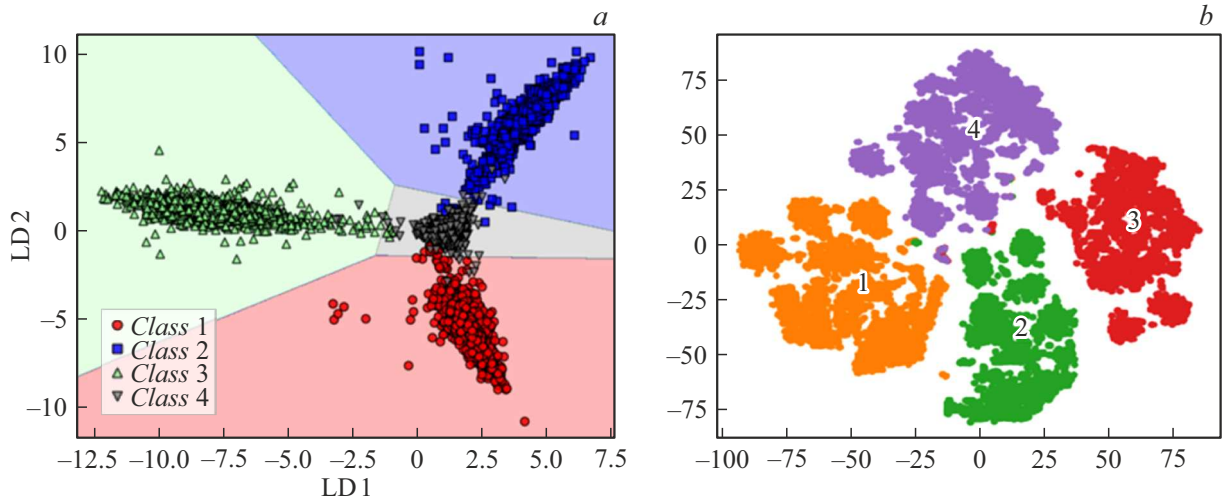

Рис. 5. a — распределение признаков обучающей выборки в пространстве двух дискриминантов, b — t-SNE редукция пространства признаков обучающей выборки до двух измерений.

Рис. 5, *b* дает основание быть уверенным, что пространство признаков обучения может быть эффективно сжато до минимального количества размерностей, хотя и не дает конкретной реализации такой редукции.

Заключение

Рассмотренные алгоритмы и программы предварительной обработки изображений являются развитием и доработкой алгоритмов и программ, описанных в работах [2,3,8]. К программам разделения „наложившихся" кластеров флуоресценции добавлен итерационный алгоритм, обрабатывающий изображения в градациях серого и бинарные (черно-белые). Данный алгоритм позволяет повысить плотность обнаруживаемых кластеров и тем самым увеличить количество нуклеотидных оснований, выявляемых в обрабатываемой пробе.

В результате апробации различных моделей машинного обучения к задаче base-calling полученные результаты показали достаточно высокие показатели качества генетического анализа. Показатели качества по Phred score оказались в диапазоне от 29 до 35 единиц, тогда как значение этого показателя для выполнения base-calling в приборе „Нанофор СПС" без применения методов ML обычно равно 30.

В качестве входных данных использовались значения интенсивностей сигналов флуоресценции кластеров, полученные от системы параллельного секвенирования прибора „Нанофор СПС". Затем на основе этих интенсивностей были извлечены статистические характеристики каждого кластера, такие, как среднее значение, медиана, pct99, pct90, так что информация о каждом нуклеотиде содержит 8 признаков (4 признака для изображения кластера, 4 признака для фона). Всего для четырех каналов получается 32 признака. При этом коррекции изменений интенсивностей из-за таких явлений как фазирование/префазирование (Phasing/Prephasing), затухание сигнала и перекрестные помехи (Cross-talk) не выполнялись.

По результатам апробации следует отметить, что деревья решений особенно привлекательны, если мы заботимся об интерпретируемости, т. е. заинтересованы в явном выделении наиболее информативных признаков в задаче классификации при определении типа нуклеотида. Логистическая регрессия — не только полезная модель для обучения в режиме реального времени поступления новых данных секвенирования (при использовании SGD, стохастического градиентного спуска при решении оптимизационной задачи), но и позволяет нам прогнозировать вероятность истинности классификации.

Хотя SVM являются мощными линейными моделями, которые можно расширить для нелинейных задач с помощью трюка с ядром, они требуют оптимальной настройки множества параметров для достижения хороших прогнозов. Ансамблевые методы, такие, как случайные леса, не требуют трудоемкой настройки параметров и позволяют избегать эффекта переобучения (в отличие от деревьев решений), что делает их привлекательными моделями для многих практических проблемных областей. Классификатор KNN предлагает альтернативный подход к классификации посредством „ленивого обучения", который позволяет делать прогнозы без какого-либо обучения модели, но с более затратным в вычислительном отношении шагом прогнозирования.

Следует также отметить, что для выбранных методов машинного обучения были исследованы и получены результаты по сокращению пространства признаков из характеристик используемых данных, что дало возможность уменьшить количество ошибок классификации и упростить процессы вычислений, так как вместо 32 признаков характеристик каждого кластера использовались только 4.

Рассмотренные методы были реализованы с помощью средств системы Scikit-learn, что дало возможность обеспечить простоту и наглядность в составлении алгоритмов и программ.

Финансирование работы

Работа выполнена при финансовой поддержке Министерства науки и высшего образования Российской Федерации в рамках проекта Федеральной научно-технической программы развития генетических технологий на 2019–2027 годы (Соглашение № 075-15-2021-1057).

Конфликт интересов

Авторы заявляют, что у них нет конфликта интересов.

Список литературы

[1] В.Е. Курочкин, Я.И. Алексеев, Д.Г. Петров, А.А. Евстрапов. Известия Российской ВМА, **40** (3), 69 (2021). DOI: 10.33917/es-3.189.2023.36-41

[2] В.В. Манойлов, А.Г. Бородинов, А.С. Сараев, А.И. Петров, И.В. Заруцкий, В.Е. Курочкин. ЖТФ, **92** (7), 985 (2022). DOI: 10.21883/JTF.2022.07.52655.318-21

[3] В.В. Манойлов, А.Г. Бородинов, И.В. Заруцкий, А.И. Петров, В.Е. Курочкин. Труды СПИИРАН, **18** (4), 1010 (2019). DOI: 10.15622/sp.2019.18.4.1010-1036

[4] Kao, Wei-Chun. *Algorithms for Next-Generation High-Throughput Sequencing Technologies* (Thesis, University of California, 2011), https://escholarship.org/uc/item/86b9c87d

[5] RTA Theory of Operations v1.13 ILLUMINA PROPRIETARY Pub. No. 770-2009-020, current as of 09 Nov. 2011

[6] S. Paliwal, A. Sharma, S. Jain, S. Sharma. *Machine Learning and Deep Learning in Bioinformatics. In Bioinformatics and Computational Biology* (Chapman and Hall/CRC, 2024), p. 63–74

[7] H. Izadkhah. *Deep Learning in Bioinformatics: Techniques and Applications in Practice* (Academic Press, 2022)

[8] А.Г. Бородинов, В.В. Манойлов, И.В. Заруцкий, А.И. Петров, В.Е. Курочкин, А.С. Сараев. Информатика и автоматизация, **21** (3), 572 (2022). DOI: 10.15622/ia.2022.3.21
[9] Р. Гонсалес, Р. Вудс. *Цифровая обработка изображений* (Техносфера, М., 2005)
[10] Б.В. Бардин, И.В. Чубинский-Надеждин. Научное приборостроение, **19** (4), 96 (2009.)
[11] N. Otsu. IEEE Transactions on Systems, Man, and Cybernetics, **9** (1), 62 (1979).
[12] L. Najman, M. Schmitt. Signal Processing, **38** (1), 99 (1994).
[13] E. Tegfalk. *Application of Machine Learning Techniques to Perform Base-Calling in Next-Generation DNA Sequencing* (2020). https://www.diva-portal.org/smash/get/diva2:1465444/FULLTEXT01.pdf
[14] S.I. Gallant. IEEE Transactions on Neural Networks, **1** (2), 179 (1990). DOI: 10.1109/72.80230
[15] S. Dreiseitl, L. Ohno-Machado. J. Biomed. Inform., **35** (5–6), 352 (2002). DOI: 10.1016/S1532-0464(03)00034-0
[16] J. Abello, G. Carmode. (eds.). *Discrete Methods in Epidemiology.* DIMACS Series in Discrete Mathematics and Theoretical Computer Science, **70**, 13 (2004).
[17] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone. *Classification and Regression Trees* (Wadsworth & Brooks/Cole Advanced Books & Software, Monterey, CA, 1984) DOI: 10.1201/9781315139470
[18] G. Biau, E. Scornet. Test, **25**, 197 (2016). DOI: 10.1007/s11749-016-0481-7
[19] T. Hastie, R. Tibshirani, J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer, N Y., 2009), DOI: 10.1007/978-0-387-84858-7
[20] J. Hao, T.K. Ho. J. Educational and Behavioral Statistics, **44** (3), 348 (2019). DOI: 10.3102/1076998619832248
[21] L. Buitincketal. *API Design for Machine Learning Software: Experiences from the Scikit-Learn Project.* arXiv preprint arXiv:1309.0238. 2013.
[22] J. Zhou, A.H. Gandomi, F. Chen, A. Holzinger. Electronics, **10** (5), 593 (2021). DOI: 10.3390/electronics10050593
[23] F. Masoodi, M. Quasim, S. Bukhari, S. Dixit, S. Alam. (eds.). *Applications of Machine Learning and Deep Learning on Biological Data* (CRC Press, 2023), DOI: 10.1201/9781003328780
[24] *Quality Scores for Next-Generation Sequencing* (Illumina Inc., San Diego, CA, 2011)
[25] A.G. Borodinov, V.V. Manjilov, I.V. Zarutskiy, A.I. Petrov, V.I. Kurochkin. *Quality Control Metrics at Different Stages of Genomic Assembly in the Parallel Sequencing Using the Nanofor SPS.* XV International scientific-technical conference on actual problems of electronic instrument engineering (APEIE), IEEE, 516 (2021). DOI: 10.1109/APEIE52976.2021.9647574
[26] X. Li, L. Zhang, J. Yang, F. Teng. J. Med. Biolog. Engineering, **44**, 231 (2024). DOI: 10.1007/s40846-024-00863-x
[27] В.В. Манойлов, А.Г. Бородинов, А.И. Петров, И.В. Заруцкий, В.Е. Курочкин. Научное приборостроение, **33** (2), 35 (2023).
[28] G.C. Linderman, S. Steinerberger. SIAM J. Mathematics of data Science, **1** (2), 313 (2019). DOI: 10.1137/18M1216134