Федеральное государственное автономное образовательное учреждение
высшего образования
«Московский физико-технический институт
(национальный исследовательский университет)»
Физтех-школа физики и исследований им. Ландау
(уд.)Физтех-кластер академической и научной карьеры (Вычислительная физика конденсированного состояния и живых систем)

Направление подготовки / специальность: 03.03.01 Прикладные математика и физика
Направленность (профиль) подготовки: Общая и прикладная физика

БИОИНФОРМАТИЧЕСКИЙ ПОИСК НОВЫХ CRISPR-АССОЦИИРОВАННЫХ БЕЛКОВ С ПОМОЩЬЮ ГЛУБОКИХ НЕЙРОННЫХ СЕТЕЙ

(бакалаврская работа)

Студент:
Шаров Илья Игоревич

(подпись студента)

Научный руководитель:
Кириллов Богдан Александрович,

(подпись научного руководителя)

Консультант *(при наличии)*:

(подпись консультанта)

Москва 2023

Содержание

Аннотация

Предмет исследования: CRISPR-Cas - система адаптивного иммунитета бактерий, используемая для защиты от вредоносной ДНК. CRISPR-Cas система состоит из библиотеки участков чужеродной ДНК - CRISPR-кассеты, и CRISPR ассоциированных белков (Cas), которые: i) обслуживают кассету (добавляют в неё новые включения, подготавливают её к работе) ii) выполняют перехват и уничтожение участков ДНК, комплементарных участкам из кассеты. Благодаря способности высокоспецифично и в программируемой манере атаковать участки генома, эта система нашла распространённое применение в качестве инструмента для генного редактирования. **Задача:** Построить алгоритм для автоматизированной классификации и поиска Cas белков, который в отличие от существующих решений не опирался бы на генетическое окружение белка, или на прямое сравнение последовательностей с уже известными Cas белками. Это бы позволило машинным способом исследовать большие метагеномные данные на предмет новых вариаций Cas белков. **Результаты:** Обучен классификатор на основе нейросети трансформера, достигающий точности, превосходящей существующие алгоритмы классификации.

Ключевые слова: *CRISPR-Cas, трансформер, глубокое обучение, классификация белков*

Введение

Актуальность темы.

Принцип работы: CRISPR-Cas система находится в ДНК или на плазмидах почти всех архей и у половины бактерий. Она состоит из генов, кодирующих Cas белки, за которыми идёт CRISPR-кассета - участки ДНК (спейсеры), разделённые повторяющимися вставками. После экспрессии CRISPR-кассета разрезается в местах повторов, образуя отрезки РНК, содержащие спейсеры. Эти отрезки соединяется с эффекторными комплексами из Cas белков, образуя интерференционный функциональный модуль. При обнаружении последовательности ДНК, комплементарной спейсеру, Cas белки эффекторного комплекса разрезают найденную ДНК. Этот механизм используется клеткой для обнаружения и устранения чужеродного (например вирусного) генетического материала [1].

Генная инженерия: Редактируя содержимое спейсеров можно управлять местами разреза ДНК. За подобное применение CRISPR-Cas систем для генного редактирования в 2020 году была выдана нобелевская премия [2]. За последнее десятилетие CRISPR-Cas занял роль доминирующего инструмента генной инженерии в связи со своей простотой и многофункциональностью [3].

Задачи области: Однако исследования CRISPR продолжаются. Обнаруживаются новые варианты CRISPR систем, которые различаются в деталях их функционирования и областях возможного применения. В связи с потребностями генной инженерии и фундаментальных исследований бактериального иммунитета возникает задача поиска новых Cas-белков и определения их функций.

Функция белка: Белки являются биополимерами, которые определяются последовательностью аминокислот, кодируемой их генетической структурой. Функция белка задаётся его структурой, то есть формой, которую принимает цепочка аминокислот. Структура в свою очередь определяется его аминокислотной последовательностью. Поскольку предсказание структуры белка с помощью *ab initio* методов на данный момент является неразрешённой задачей, функция неизвестного белка определяется путем сравнения его с уже изученными белками.

Существующие алгоритмы: Существующие алгоритмы классификации и определения функции можно отнести к двум направлениям: поиск по гомологии и сравнение генетического контекста. i) Поиск по гомологии - сравнение целевой последовательности, кодирующей белок с уже известными представителями. Большое совпадение последовательностей предполагает похожие структуры белков, из чего подразумевается совпадение функций. [4] ii) Сравнение генетического контекста опирается на предположение, что похожие CRISPR системы будут требовать белки с похожими функциями. Поэтому найдя такую CRISPR систему можно предположить функции белков в изначальной. [5]

Ограничения алгоритмов: Поиск по гомологии крайне зависим от количества схожих белков с уже изученными функциями. В связи с ролью CRISPR в бактериальном иммунитете Cas белки участвовали в эволюционной гонке со средствами защиты бактериофагов, что привело к повышенному разнообразию вариаций Cas белков [6], затрудняющему поиск по гомологии. Подходы же, основывающиеся на генетическом контексте опираются на наличие CRISPR-Cas системы и не позволяют исследовать Cas белки, имеющие роли помимо адаптивного иммунитета [7].

Цель работы. Целью работы является разработка алгоритма машинной классификации Cas белков для предсказания их функций, который

- не опирается на прямое сравнение аминокислотных последовательностей;
- не учитывает контекст CRISPR системы;
- требует минимального курирования результатов со стороны специалиста;
- работает с достаточной скоростью для обработки больших метагеномных данных;

Методы исследования. В работе были использована предобученная на аминокислотных последовательностях нейронная сеть типа трансформер, поверх которой был обучен классификатор логистической регрессии. Были исследованы алгоритмы классификации с неограниченным количеством классов (open set classification), и реализована классификация с вариантом отмены (classification with reject option).

Научная новизна. Большие языковые модели (Large Language Models) ещё не применялись для аннотации Cas белков.

Практическая значимость. Точность и особенности требований полученного классификатора позволяют использовать его не только для исследования новых или не до конца аннотированных CRISPR-Cas систем, но и исследовать Cas белки, которые находятся вне систем адаптивного иммунитета и выполняют другие роли [8].

Анализ производительности модели. Модель даёт предсказания, превосходящие по точности существующие аналоги.

Дальнейшие направления исследований.

- Проверка модели на не аннотированных данных;
- Чистка кода и упрощение использования;
- Подача на включение в инструментарий для исследования CRISPR-Cas систем [9];
- Смена алгоритма с классификации на кластеризацию расширения применений модели;

Апробация работы Тезисы по содержанию работы поданы на постерную презентацию 11-ой Московская конференции по вычислительной молекулярной биологии (MCCMB)

Обозначения

- CRISPR – (*clustered regularly interspaced short palindromic repeats*) особые локусы бактерий и архей, состоящие из прямых повторяющихся последовательностей, которые разделены уникальными последовательностями (спейсерами);
- Cas – (*CRISPR assotiated*) семейство генов, работа которых ассоциирована с работой CRISPR;
- ДНК – (*дезоксирибонуклеиновая кислота*) двойная спиральная молекула, содержащая генетическую информацию, которая кодирует наследственные характеристики и функции организма;
- РНК – (*рибонуклеиновая кислота*) одноцепочечная молекула, выполняющая различные функции в передаче, трансляции и регуляции генетической информации;
- ReLU – (*Rectified Linear Unit*) активационная функция, которая преобразует входные значения в выходные, оставляя положительные значения без изменений, а отрицательные значения заменяет нулем;
- Softmax - функция активации, преобразующая вектор значений z в вероятностное распределение путем экспоненциального преобразования и нормализации. $\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad i = 1, 2, \ldots, K$;
- EAT – (*Embedding-based Annotation Transfer*) аннотирование белков при помощи ембеддингов;

1 Биологический контекст задачи

В этой секции будет описан принцип работы CRISPR, его представленность в природе, практическая важность и существующие в индустрии методы поиска.

1.1 Краткий обзор CRISPR

CRISPR - локус в ДНК бактерий и архей, состоящий из прямых повторяющихся последовательностей, которые разделены уникальными последовательностями (спейсерами).

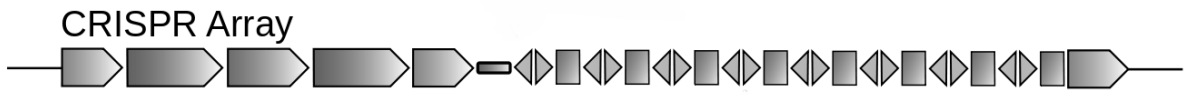


Рис. 1: Схематичное изображение Cas-генов и CRISPR-кассеты.
Здесь: треугольники - палиндромные повторы, прямоугольники - спейсеры, а пятиугольники - гены. Cas-гены в основном находятся выше самой CRISPR-кассеты (слева). Маленький чёрный прямоугольник - лидерная последовательность, увеличивающая количество копий кассеты. Автор: James Atmos [10]

Гены Cas кодируют белки, функции которых можно разделить на три группы: встраивание новых спейсеров, и уничтожение агентов с идентичными спейсепам последовательностями (протоспейсерами) и помощь в процессинге CRISPR-транскрипта. Функцию уничтожения протоспейсеров выполняют Cas-белки, называемые эффекторными. В зависимости от типа эффекторов все CRISPR-системы разделяют на два класса: у I класса мишень уничтожается мультибелковым комплексом, а у II — одним крупным белком. Далее эти классы подразделяются на шесть типов. Большинство эффекторов атакует ДНК, лишь один — исключительно РНК [11], редкие — обе молекулы.

Один организм может содержать несколько разных систем CRISPR, набор Cas-генов и вид палиндромных повторов (размером 24-48 пар нуклеотидов)в свою очередь зависит от типа этой системы. Вставки спейсеры представляют собой библиотеку опасных генетических элементов и могут разниться от особи к особи, в зависимости от её истории. Число спейсеров может разниться от единиц до сотен.

Для целей нашей задачи не имеет смысла вдаваться в детальный принцип работы CRISPR-Cas системы, отметим однако что количество Cas-генов варьируется от нескольких штук до пары десятков, также иногда для функционирования системы рекрутируются сторонние белки такие например как RNase III [12].

Отметим только важные для повествования пункты: белки Cas1 и Cas2 (нет совпадения нумерации с Рис. 2), вариации которых присутствуют почти во всех CRISPR-Cas системах, отвечают за адаптацию (встраивание) спейсеров новых в начало CRISPR-кассеты. После чего кассета экспрессируется, нарезается на куски РНК - crRNA, каждый содержит один спейсер. crRNA связывается с эффекторным комплексом и, найдя полное совпадение спейсера с генетическим материалом, слипается с ним, активируя эффекторный белок, который разрезает спейсер и комплементарный ему захваченный генетический элемент.

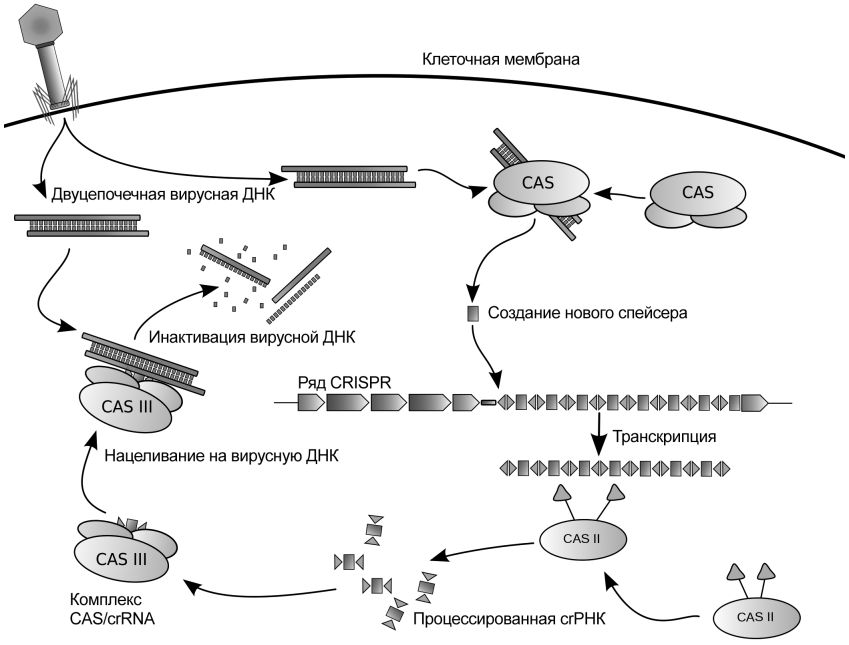


Рис. 2: Цикл работы CRISPR-Cas системы. Автор: James Atmos [10]

В качестве дальнейшего чтения, освещающего принцип работы CRISPR-Cas машинерии предлагается обзор [13].

1.2 Разнообразие, роли, классификация CRISPR-Cas систем

CRISPR системы встречаются в отсеквенированных геномах 50% бактерий и 90% архей [14]. Аналогично другим биологическим механизмам защиты, архейные и бактериальные системы CRISPR-Cas проявляют выделяющееся разнообразие последовательностей Cas-белков, генетических составов и архитектуры геномных локусов [15, 16].

Чтобы упростить исследование CRISPR-Cas систем была предложена универсальная система классификации, последняя ревизия которой описана в [17]. CRISPR-Cas системы делятся на два класса по составу эффекторного комплекса. У класса I эффекторный комплекс состоит из нескольких Cas (или рекрутированных) белков, а у II класса это один мультидоменный Cas белок. Всего система CRISPR-Cas классификации насчитывает 2 класса, 6 типов и 33 подтипа. Группирование осуществляется на основе сходства последовательностей, генетического родства консервативных Cas белков и состава системы, однако высокое генетическое разнообразие Cas белков иногда затрудняет однозначную классификацию системы. В таких случаях требуется ручное курирование биологом. С расширением базы геномов и обнаружением новых CRISPR-Cas систем количество подтипов постоянно увеличивается. Примеры такой ручной классификации, когда вычислительные методы не дают однозначного результата, можно найти в [18].

Помимо традиционной функции адаптивного иммунитета в связке CRISPR, Cas белки были найдены вдалеке от CRISPR-кассет, что предполагает вариативность их функции [8]. Также было обнаружено, что Cas-белки III типа играют роль в транс-

портных путях [19,20]. Однако все функции Cas-белков ещё не полностью исследованы. Дополнительная литература по данной тематике может быть найдена здесь [7].

В ходе данной работы мы будем проводить классификацию отдельных генов. Названия генов были присвоены в порядке их обнаружения, а их группировка происходит на основе функционального сходства. Некоторые гены, такие как Cas1 и Cas2, обладают достаточно консервативным нуклеотидным составом, в то время как другие проявляют значительное разнообразие. Мы будем классифицировать как основные гены, обычно расположенные перед CRISPR-кассетой, так и вспомогательные гены (расположенные рядом с CRISPR-Cas локусом, но имеющие подтвержденную роль в CRISPR-машинерии).

1.3 Методы поиска Cas генов

Так как Cas белки нашли распространённое применение в генной инженерии [3], а роль белков в работе бактериального иммунитета представляет фундаментальный интерес, появилась потребность в инструментах поиска, идентификации и классификации Cas-генов.

Большая часть алгоритмов поиска Cas-генов опирается на поиск CRISPR-кассеты, так как можно выделить её периодическую структуру в больших метагеномных данных.

Так например CRISPRDetect [21] ищет в геноме повторяющиеся палиндромные вставки путём сравнения строк, CRISPR-SE [22] ищет спейсеры, по генетический материал бактериофагов, а CRISPRIdentify [23] использует машинное обучение для поиска повторов.

После обнаружения CRISPR-кассеты гены, находящихся в её окружении, рассматриваются как потенциальные кандидаты на Cas-белки. К этим кандидатам могут быть применены дополнительные методы отсева, такие как подсчёт CRISPRicity [24] - отношение частоты встречаемости гена в связке с CRISPR к общей частоте встречаемости гена.

Однако окончательно функцию гена можно только сопоставив его с уже изученным. Для этого стандартным методом считается поиск по гомологии.

Из-за сильного генетического разнообразия Cas-генов процент совпадений бывает мал, что часто затрудняет поиск по гомологии других Cas белков.

Геном Совокупность всего генетического материала, заключённого в клетке.

Метагеном Совокупность генетического материала нескольких (обычно большого числа) организмов.

Поиск по гомологии Поиск по гомологии для определения функции белка является методом анализа белков, основанным на их сходстве с уже известными белками и последовательностями аминокислот. Этот подход основан на предположении, что белки, имеющие схожие последовательности аминокислот, вероятно выполняют схожие функции. В процессе поиска по гомологии, используя компьютерные алгоритмы и базы данных с известными белками, анализируются степень сходства между неизвестным белком и белками с известной функцией. Если найдены близкие гомологи с хорошим совпадением последовательностей,

то можно предположить, что неизвестный белок выполняет аналогичную функцию. Дополнительные эксперименты и функциональные анализы могут быть проведены для подтверждения предполагаемой функции белка, полученной из поиска по гомологии. Этот метод является ценным инструментом в биоинформатике и позволяет установить функцию неизвестных белков на основе сходства с уже изученными и хорошо описанными белками.

Биологическая задача

Необходим алгоритм для поиска Cas-генов и определения их функции, который бы не зависел от наличия рядом CRISPR-кассеты, и мог бы успешно классифицировать белки, подверженные сильному эволюционному разнообразию.

2 Глубокое обучение

2.1 Генетический код, как текст

Генетическая информация кодируется с использованием двух основных молекул - ДНК и РНК. Гены, которые являются основными носителями наследственной информации, содержатся в ДНК и состоят из последовательности нуклеотидов - аденина (A), цитозина (C), гуанина (G) и тимина (T). Эта последовательность нуклеотидов в ДНК служит шаблоном для синтеза молекул РНК, которая затем транслируется в белки. Каждая последовательность трех нуклеотидов, называемая кодоном, кодирует определенную аминокислоту. Аминокислоты затем соединяются в цепочку, образуя белок. Таким образом, генетическая информация переходит из последовательности нуклеотидов в последовательность аминокислот. Не всегда правда есть взаимооднозначное соответствие между последовательностью нуклеотидов в гене и последовательностью аминокислот в белке. Иногда в процессе переноса информации происходят посттрансляционные модификации [25]. Однако их влиянием в нашей задаче можно пренебречь.

Аналогично тому, как набор символов в тексте образует слова и фразы, последовательность нуклеотидов в гене формирует кодоны, которые в свою очередь формируют последовательность аминокислот в белке. Такая аналогия позволяет применять методы машинного обучения, обработки естественного языка и анализа текста для изучения генетической информации.

С этого момента мы будем рассматривать белки как последовательности, представляющие собой цепочку аминокислот. Каждая аминокислота в цепочке кодируется одним символом. Общее количество аминокислот, которые кодируются генетическим кодом, составляет 20. Таким образом, белки можно рассматривать как слова, составленные из алфавита из 20 символов.

2.2 Языковые модели для анализа текста

Языковые модели - это статистические модели, которые позволяют предсказывать вероятность следующего слова или последовательности слов в тексте. Они основаны на анализе больших объемов текстовых данных и выявлении зависимостей между словами.

Один из примеров архитектуры языковой модели - архитектура encoder-decoder [26]. Архитектура encoder-decoder является моделью, которая используется в области машинного обучения и обработки естественного языка. Она состоит из двух основных компонентов: кодировщика (encoder) и декодера (decoder):

Encoder Принимает входные данные (например текстовую последовательность) и преобразует их во внутреннее представление - вектор фиксированной длинны, называемый контекстом или ембеддингом. Контекст содержит информацию о входных данных и служит в качестве своеобразного "сжатого"представления.

Decoder принимает контекст, а также начальное состояние и генерирует выходную последовательность.

Если поставить целью декодеру восстановить изначальные данные, то система будет работать, как архиватор и деархиватор информации с вектором контекста, как промежуточным состоянием. Далее мы будем называть его эмбеддингом.

Типичным процесс обучения подобных моделей:

- Входные данные токенизируются: Каждый символ или группы символов получают численное представление, что превращает строку текста в числовой вектор.

- Часть входных данных маскируется: некоторые токены заменяются на токены-маски, которые ничего не значат.

- МаскированнаяПоследовательность пропускается через encoder-decoder: причём целью декодера является восстановление последовательности без масок.

Так как модель требуют сжимать последовательность, а потом восстанавливать изначальную информацию encoder обучается сжимать её в эмбеддинг с минимальной потерей информации (понятно, что при размере эмбеддинга, сравнимом с размером входных данных никакой потери информации не происходит, однако обычно эмбинг обладает размерностью в несколько раз меньше, чем входная последовательность). Так как информация попадает в encoder-decoder в маскированном состоянии, с пробелами, encoder обучается взаимосвязям между токенами. Какие токены можно ставить вместе, а какие нет, какие паттерны среди них можно пронаблюдать.

В применении к белкам эта модель учится сжимать информацию о цепочке аминокислот фиксированной длинны, а также обучается паттернам расположения аминокислот. В итоге она может восстанавливать маскированные участки биологически и физически обоснованно.

Вектор эмбеддинга, как сжатое численное представление исходных данных, широко применяется для классификации или анализа данных, поскольку он содержит всю существенную информацию, заключенную в исходных данных, но обадает меньшей размерностью.

Здесь представлены только основные представления о языковых моделях, однако эти техники нашли повсеместное применение в индустрии и информация о них может быть найдена в открытых источниках.

2.3 Трансформеры

Трансформер - это архитектура нейронной сети, которая была представлена в статье "Attention is All You Need" [27]. Он применяется в задачах обработки естественного языка и достигает высоких результатов в машинном переводе, а также в других задачах, где важна работа с последовательностями данных.

В отличие от полностью связанной нейронной цепи, где результатом следующего слоя является линейная комбинация предыдущего с применением нелинейной функции f(x) (часто используются ReLU или SoftMax) трансформеры используют механизм внимания (attention).

Механизм внимания в основе использует скалярное произведение между различными частями входных данных. Эта операция позволяет нейронной сети выделять

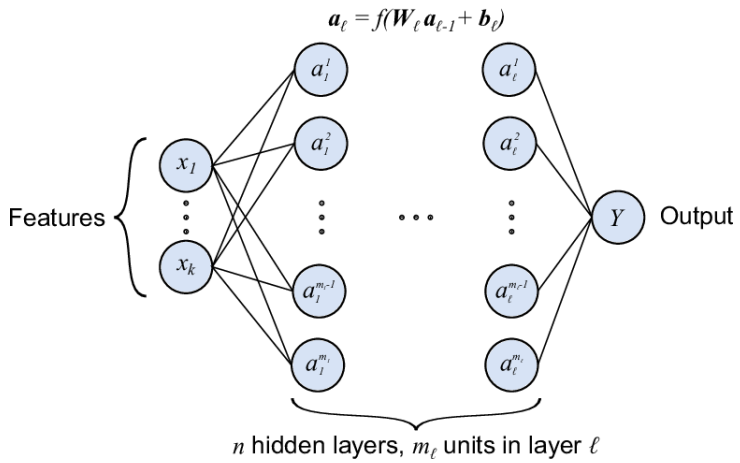


Рис. 3: Схематичное изображение полносвязной нейронной сети Автор: Gregory Teichert [28]

наиболее схожие части входных данных и придавать им большее важность. Результат этого скалярного произведения, умноженный на исходные данные, представляет собой "нормированное внимание".

Позже, добавляя обучаемые параметры (матрицы Query, Key и Value на Рис. 4) мы можем обучать сеть не только видеть близкие по значению вектора, но и более сложные паттерны связей. Это крайне упрощённое описание механизма внимания в нейросетях архитектуры трансформер. За дальнейшим чтением можно обратиться к [27].

2.4 Трансформеры в применении к белкам

Для достижения желаемых результатов в обучении трансформеров, которые должны правильно сжимать информацию в эмбеддинги и учитывать физические и биологические законы расположения аминокислот в белках, требуются значительные вычислительные ресурсы и объем данных. Поэтому в индустрии стандартной практикой является обучение больших трансформеров, а затем использование полученных эмбеддингов для обучения различных задач анализа.

На данный момент практически каждый отдел исследований и разработки у ведущих IT-компаний, таких как DeepMind, Nvidia и Amazon Web Services, обладает обученным белковым трансформером. Кроме того, независимые научные группы также активно работают над улучшением качества белковых трансформеров. Они могут различаться по деталям архитектуры и процесса обучения, использованным данным, вычислительной мощности, требованиям к видеопамяти, качеству эмбеддингов и доступности для использования. В рамках этой работы, я сравнил две наиболее популярные и широко используемые модели в биоинформатике:

ESM-2 - трансформер, разработанный группой Facebook Research [30]. ESM-2 (далее ESM) - пример модели, следующей тренду на увеличение количества обучаемых параметров моделей, что связывается с улучшением качества репрезентации ембеддинга [31]. Количество параметров её ревизии с конкурентоспособной точностью - 650 миллионов параметров, а для ревизии с наибольшей точностью - 15 миллиардов.

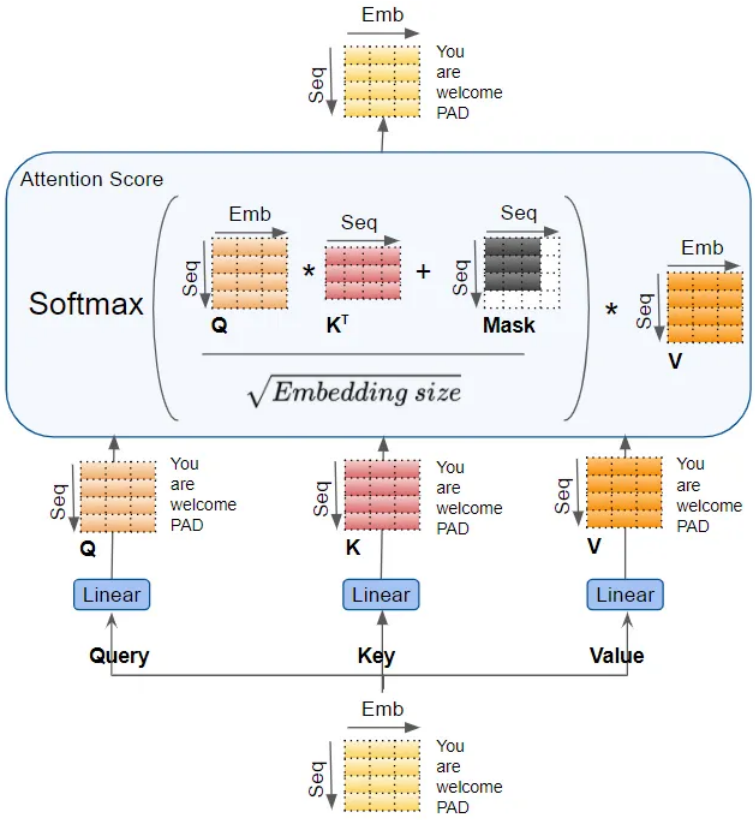


Рис. 4: Схематичное изображение одного блока внимания Автор: Ketan Doshi [29]

Ankh - это трансформер, разработанный научной группой совместно с Google Cloud [32]. Авторы модели достигли конкурентоспособных результатов, уделяя внимание детальной настройке процесса обучения и обучающих данных, вместо простого увеличения размера модели. Модель Ankh_base имеет размер около 450 миллионов параметров.

Однако основным критерием выбора были тестовые задачи классификации, обученные на ембеддингах, полученных разными моделями [32]. Крайне релевантным для задачи классификации Cas белков является тест по EAT (*Embedding-based Annotation Transfer*). В процессе теста на ембеддингах (как входных данных, представляющих изначальные входные данные - аминокислотные последовательности белков) обучается классификатор, определяющий белок в группу по классу, архитектуре, топологии, и гомологии. Модель обучается на 69000 последовательностях, аннотированных вручную и проверяется на 219 тестовых последовательностях. Подробнее о процедуре эксперимента читать в [33, 34]. Эксперимент EAT практически повторяет задачу Cas классификации на ембеддингах, за исключением разве что логики составления групп - в классификации Cas группы составляются по функции Cas белка. Как видно из Таблицы 1 Ankh выигрывает в модельной задаче, а меньший размер позволит запустить эту модель на имеющихся в доступе видео-ускорителях.

Task	Ankh (1.15B)	Ankh_base (450M)	ESM-2 (650M)	ESM-2 (3B)	ESM-2 (15B)
EAT	71.7±6%	74.8±6%	55.5±7%	65.6±6%	65.4±7%

Таблица 1: Сравнение средней для разных групп точности эксперимента EAT, построенного на основе ембеддинга разных моделей. В скобках указано количество параметров в миллионах (M), и миллиардах (B)

Для дальнейшей работы была выбрана модель Ankh_base, которая доступна в открытом доступе по условиям лицензии [35] (свободное некоммерческое распространение и модификация).

3 Классификация Cas белков

Все исходники кода и скрипты для обработки могут быть найдены в репозитории проекта по ссылке [36].

3.1 Препроцессинг данных и получение ембеддингов

Прежде чем обучать классификатор был подготовлен датасет, связывающий аминокислотные последовательности и функции Cas белков. Для этого был взят датасет всех аннотированных Cas белков из статьи [17] (далее KiraDataset). На данный момент существуют более новые ревизии датасетов аннотированных вручную, или достаточно достоверными машинными способами (сравнение по гомологии). Включение их в обучающий датасет является одной из первоочерёдных задач дальнейшего развития данного проекта.

В состав KiraDataset входили идентификаторы сборки прочтений бактериального генома, информация о позиции аннотированных генов в геноме и их аннотации. Датасет содержал не только Cas белки, но и другие белки, которые могли находиться рядом с CRISPR-кассетой, а также возможные ошибки аннотации. Для последующего анализа было подсчитано количество аннотированных белков каждого типа. Затем список был сокращен, оставляя только белки, которые встречались в датасете более 100 раз. Это позволило удалить мусор и ошибки аннотации, а также сосредоточиться только на Cas белках, которые имели достаточно информации для построения надежного классификатора с использованием методов машинного обучения. В будущих обновлениях классификатора будет проведена работа по классификации и распознаванию более редко встречающихся Cas белков. Кроме того, из сокращенного KiraDataset'a были исключены белки, которые явно не имели отношения к функционированию CRISPR-Cas системы, такие как PTPS (фермент, играющий роль в регуляции сигнальных путей и фосфорилировании тирозина в клетках).

Финальный список белков для обучающего датасета представлен на Таблице 2. Именно эти Cas белки будет уметь различать классификатор. Финальный датасет будем дальше называть CasVoc (*Cas Vocabulary*). В него входит ∼36000 различных белков.

Для белков из CasVoc при помощи инструментария датабазы NCBI [37] были загружены геномы. По указанным в KiraDataset координатам из геномов были вырезаны гены, кодирующие Cas белки.

Кодирующая нить Двойная спираль ДНК состоит из двух комплементарных нитей нуклеотидов: положительной и отрицательной. Положительная кодирующая нить ориентирована в 5' - 3' направлении, отрицательная кодирующая нить ориентирована в 3' - 5' направлении.

Все последовательности ДНК геномов, извлеченных из базы данных NCBI, записаны в 5'-3' направлении. Однако, в генах из CasVoc присутствуют гены, которые кодируются отрицательной нитью ДНК. Поскольку процесс синтеза РНК происходит только в 5'-3' направлении, ген, который кодируется отрицательной нитью ДНК, будет записан в обратном порядке на комплементарной положительной цепи. Для участков положительной цепи, соответствующих этим генам, были найдены комплементарные цепи, и последовательность этих цепей была обращена. Полученные

нуклеиновые последовательности были транслированы в аминокислотные последовательности. Для выполнения всех операций с генетическими данными использовался пакет Biopython [38].

Id	Название гена	Количество включений
0	cas1	4967
1	cas2	4717
2	cas5	3209
3	cas7	2764
4	cas3	2363
5	cas6e	1860
6	cas8e	1830
7	cse2gr11	1825
8	cas4	1395
9	cas6	1374
10	cas9	1162
11	cas10	759
12	csm3gr7	745
13	cas8c	664
14	csn2	617
15	cas6f	589
16	cas7f	587
17	cas8f	530
18	cas5f	528
19	cas3f	502
20	cas7b	466
21	csx1	420
22	csm2gr11	402
23	csm4gr5	364
24	csm5gr7	322
25	cas8b1	268
26	cmr3gr5	250
27	csm6	242
28	cmr5gr11	240
29	cmr4gr7	234
30	cas8b2	229
31	cmr6gr7	226
32	cmr1gr7	189
33	csb2gr5	130
34	csb1gr7	125
35	csx19	107
36	csx10gr5	106

Таблица 2: Cas белки, которые вошли в финальный датасет для классификатора

Для датасета CasVoc был применен encoder Ankh, который создал 756-мерные эмбеддинги для каждой аминокислотной последовательности. Для сравнения, длина

этих последовательностей составляла порядка 3500 аминокислотных остатков. Вычисления с использованием фреймворка PyTorch и ускорителя cuda на видеокарте GTX 1080ti заняли около 72 часов.

3.2 Построение классификатора

Для выбора архитектуры классификатора был использован пакет перебора гиперпараметров TPOT [39], который, руководствуясь метрикой balanced_accuracy, нашёл оптимальные параметры для классификации при помощи RobustScaler и логистической регрессии

Balanced accuracy - (Сбалансированная точность) метрика оценки производительности классификационных моделей, учитывающая несбалансированность классов. Она вычисляется как среднее значение чувствительности для каждого класса, где чувствительность определяется как отношение правильно классифицированных положительных примеров к общему числу примеров этого класса. Таким образом, сбалансированная точность позволяет учесть различия в размере и распределении классов, обеспечивая объективную оценку производительности модели при работе с несбалансированными данными.

Robust Scaler - метод масштабирования данных, используемый в машинном обучении для нормализации признаков. Он относится к методам, которые устойчивы к выбросам и аномалиям в данных. В отличие от стандартного масштабирования, где данные приводятся к среднему значению и стандартному отклонению, Robust Scaler использует медиану и интерквартильный размах для центрирования и масштабирования данных. Это позволяет уменьшить влияние выбросов и сохранить относительные различия между значениями признаков.

3.3 Логистическая регрессия

Сначала рассмотрим метод классификации логистической регрессии для бинарного случая, а потом расширим его для мультиклассовой задачи.

Бинарная задача

В бинарной задаче есть всего два класса, которые мы обозначим, как 0 и 1. X_i - вектор входных данных, y_i - настоящий класс X_i, $\hat{p}(X_i)$ - предсказываемая вероятность принадлежности X_i к положительно размеченному классу $P(y_i = 1|X_i)$.

$$\hat{p}(X_i) = \operatorname{expit}(X_i w + w_0) = \frac{1}{1 + \exp(-X_i w - w_0)}$$

Здесь ω и ω_0 - параметры, характеризующие модель. Во время обучения решается задача оптимизации логарифмической функции потерь:

$$\min_w C \sum_{i=1}^{n} (-y_i \log(\hat{p}(X_i)) - (1 - y_i) \log(1 - \hat{p}(X_i))) + r(w)$$

Где в качестве регуляризационного члена, не допускающего взрыва весов ω и ω_0, ведущего к переобучению была выбрана l_2 регуляризация: $r(\omega) = \frac{1}{2}\|w\|_2^2 = \frac{1}{2} w^T w$ На

практике для точного причисления X_i к какому-то классу выбирается параметр обрезания. Если предсказанная вероятность выше этого параметра, модель причисляет включение X_i к классу 1.

Мультиклассовая задача

Для мульти классовой задачи логистическая регрессия расширяется, поддерживая несколько классов: $y_i \in 1, \ldots, K$. Вместо вектора коэфициентов ω мы используем матрицу W, в которой W_k соответствует k-ому классу. Тогда уже описанные функции будут иметь вид:

$$\hat{p}_k(X_i) = \frac{\exp(X_i W_k + W_{0,k})}{\sum_{l=0}^{K-1} \exp(X_i W_l + W_{0,l})}$$

$$\min_W -C \sum_{i=1}^{n} \sum_{k=0}^{K-1} [y_i = k] \log(\hat{p}_k(X_i)) + r(W)$$

$$\frac{1}{2}\|W\|_F^2 = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{K} W_{i,j}^2$$

Мультиклассовое обучение идёт по модели "one-vs-rest один против остальных. Для каждого класса обучается бинарная логистическая регрессия, отделяющая этот класс от остальных.

3.4 Производительность классификатора

Для тестовой выборки из CasVoc (на этой части датасета классификатор не обучался) мною были посчитаны Precision, recall и f1-score. Результаты можно видеть на Таблице 3.

Precision (точность) и recall (полнота) являются метриками оценки производительности модели в задачах классификации. Precision измеряет долю верно предсказанных положительных классов от общего числа положительных предсказаний. Она позволяет оценить, насколько точно модель предсказывает положительные классы. Recall, с другой стороны, измеряет долю верно предсказанных положительных классов от общего числа истинных положительных классов. Recall показывает, какую часть положительных классов модель смогла обнаружить. Выбор между precision и recall зависит от конкретной задачи. Если важно минимизировать ложноположительные предсказания, то стоит ориентироваться на высокую precision. Если же важно минимизировать ложно-отрицательные предсказания, то следует стремиться к высокому recall.

F1-мера (F1-score) является средним гармоническим значением между precision и recall и является метрикой, объединяющей оценки точности и полноты в задачах классификации. Она используется для оценки сбалансированности между precision и recall. F1-мера достигает своего максимального значения в случае, когда precision и recall равны между собой, что означает, что модель имеет хорошее сочетание точности и полноты. Важно отметить, что F1-мера учитывает только истинно положительные, ложно положительные и ложно отрицательные предсказания, но не учитывает истинно отрицательные предсказания. Высокое значение F1-меры свидетельствует

	precision	recall	f1-score	support
cas1	1.00	0.98	0.99	1232
cas2	1.00	1.00	1.00	1169
cas5	0.99	0.99	0.99	795
cas7	0.98	0.98	0.98	685
cas3	1.00	1.00	1.00	584
cas6e	0.99	0.98	0.99	462
cas8e	0.99	0.99	0.99	454
cse2gr11	0.96	0.97	0.96	453
cas4	1.00	0.99	1.00	343
cas6	0.95	0.97	0.96	339
cas9	0.99	0.98	0.99	290
cas10	1.00	0.99	0.99	187
csm3gr7	0.97	0.96	0.97	185
cas8c	0.99	0.98	0.98	165
csn2	0.98	0.99	0.98	154
cas6f	0.95	0.99	0.97	146
cas7f	0.91	0.99	0.95	146
cas8f	0.95	0.95	0.95	132
cas5f	0.99	1.00	1.00	132
cas3f	1.00	0.99	1.00	125
cas7b	0.91	0.95	0.93	115
csx1	0.97	0.94	0.96	104
csm2gr11	0.88	0.97	0.92	100
csm4gr5	0.98	0.97	0.97	90
csm5gr7	1.00	0.99	0.99	80
cas8b1	0.89	0.94	0.91	67
cmr3gr5	0.96	0.90	0.93	61
csm6	1.00	0.92	0.96	60
cmr5gr11	0.88	0.85	0.86	59
cmr4gr7	0.98	0.95	0.96	57
cas8b2	0.96	0.91	0.94	56
cmr6gr7	0.98	0.96	0.97	55
cmr1gr7	0.98	0.96	0.97	46
csb2gr5	1.00	0.88	0.93	32
csb1gr7	0.96	0.77	0.86	31
csx19	0.74	0.77	0.75	26
csx10gr5	0.81	0.96	0.88	26
accuracy			0.98	9243
macro avg	0.96	0.95	0.95	9243
weighted avg	0.98	0.98	0.98	9243

Таблица 3: Здесь представленны precision, recall и f1-score для каждого класса Cas белков. Также посчитаны точность, macro average метрик (простое средне арифметическое метрик всех классов) и weighted average (взвешенное представленностью класса в тестовой выборке среднее арифметическое). Support здесь -представленность в выборке

о хорошей производительности модели, особенно в случаях, когда важно достичь баланса между точностью и полнотой.

Также мною была посчитана матрица неточностей для многоклассовой классификации (multi class confusion matrix). Эта матрица отображает количество правильно и неправильно классифицированных примеров для каждого класса. В каждой ячейке матрицы указывается количество примеров, отнесенных к определенному классу, по вертикальной оси, и классифицированных моделью в определенный класс, по горизонтальной оси. Таким образом, матрица неточностей позволяет легко определить, в каких классах модель делает ошибки и какие классы она успешно распознает.

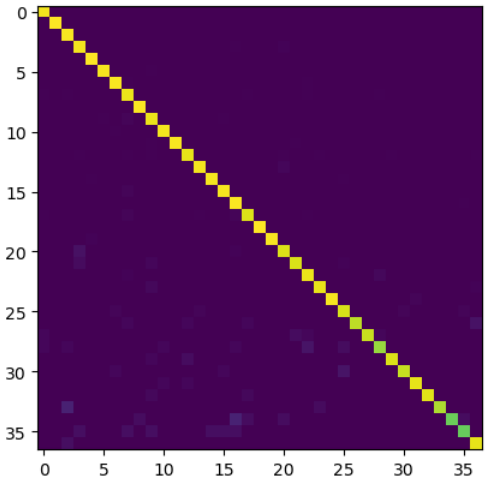


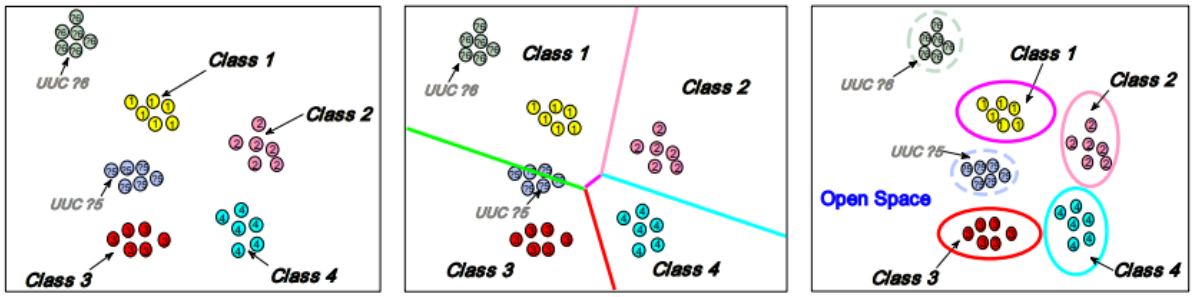
Рис. 5: Жёлтые (светлые в чб) пиксели - классы с большим количеством правильных соотнесений, голубые (серые в чб) - классы с меньшим количеством правильных соотнесений. Классы токенизированны числами от 0 до 36 для распознавания моделью. Для соотнесения числа и названия класса обращаться к Таблице 2

Низкая точность классов 34 и 35, возможно, обусловлена их малым количеством обучающих примеров, что ограничивает возможности модели для обучения. Однако класс 36, который также имеет очень малое представление, демонстрирует хорошую точность. Этот класс относится к вспомогательным белкам, и для выяснения причин такого поведения классификатора необходимы дополнительные биологические анализы, которые планируются в дальнейшем. Средняя точность классификатора составляет 98%, что является высоким значением, но требует дальнейшего анализа для подтверждения результатов.

3.5 Идентификатор

Обученный классификатор страдает от существенного недостатка: ложных срабатываний на "неизвестные"классы. В контексте разработки нашей утилиты, которая направлена на поиск новых Cas-белков, требуется способность не только определять тип белка, но и различать Cas-белки от не Cas-белков. В настоящее время модель может демонстрировать ложные срабатывания, идентифицируя Cas-белки в различных регионах генома. Эта проблема описывается форматом задачи Open Set Classification:

модель может столкнуться с данными из классов, которые ей никогда не встречались в процессе обучения. Схематическое изображение проблемы представлено на Рисунке 6.



(a) Distribution of the original data set. (b) Traditional recognition/classification problem. (c) Open set recognition/classification problem.

Рис. 6: Здесь схематично изображено двумерное пространство емебеддинга, классы Class i - классы, определённые в процессе в процессе обучения, UUC ?i - "неизвесные"классы, которые не встречались в датасете. Линии - репрезентаци границ решений классификатора.
На картинке (b) показан принцип ложных срабатываний. На картинке (c) показано желаемое поведение классификатора. Картинка заимствована из [40]

Решение Open Set проблемы - открытый вопрос в машинном обучении, однако существует несколько наиболее распространённых методов [40]. Один из методов при ограниченном доступе к данным из "неизвестных"классов - обучение дополнительного классификатора (далее идентификатор), который будет разделять входные данные на представленные в датасете или на сторонние.

Для построения такого идентификатора были собраны все гены бактерий и архей (так как нужно обучить отличать идентификатор Cas-гены от не-Cas-генов, а Cas встречается только в бактериях и археях), из них сделана случайная выборка в 1000 генов. Для этих генов были проведены все процедуры препроцессинга, что и для Cas-генов, включая трансляцию и генерирование емебеддингов.

3.6 Производительность идентификатора

Для идентификатора посчитаны метрики, схожие с метриками классификатора.

Использование простой логистической регрессии на емебеддингах позволяет достичь очень хороших результатов. Эти результаты, как подтверждено научным руководителем, являются впечатляющими и, вероятно, не могли быть достигнуты без предварительного обучения encoder'a.

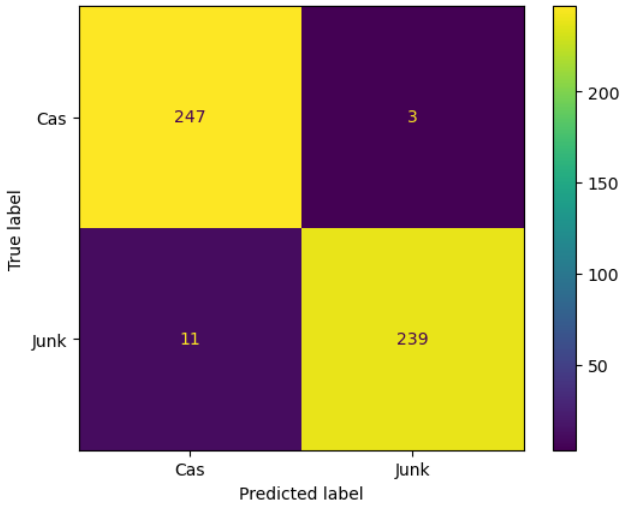


Рис. 7: Матрица неточностей идентификатора

	precision	recall	f1-score	support
Cas	0.96	0.99	0.97	250
Junk	0.99	0.96	0.97	250
accuracy			0.97	500
macro avg	0.97	0.97	0.97	500
weighted avg	0.97	0.97	0.97	500

Таблица 4: Метрики точности идентификатора

4 Результаты

Полученный пайплайн состоит из предобученного энкодера, идентификатора и классификатора, что позволяет решать задачу поиска и классификации Cas белков без использования множественных выравниваний генетических последовательностей или контекста CRISPR-Cas системы. Эта особенность пайплайна позволяет исследовать Cas белки, имеющие различные функции помимо адаптивного иммунитета, независимо от их присутствия в CRISPR-кассете. Единственным известным аналогом является CRISPRCasIdentifier [5], однако он обладает совершенно другой архитектурой и применим только в контексте CRISPR-Cas системы. CRISPRCasIdentifier выполняет аннотацию Cas белков CRISPR-Cas путем сопоставления набора присутствующих белков с наиболее распространенным набором Cas белков CRISPR системы определенного типа. Точность и f1-мера CRISPRCasIdentifier составляют 0.91 и 0.89 соответственно, в то время как точность и f1-мера полученного пайплайна составляют 0.97 и 0.99 соответственно. Точность полученного классификатора превосходит ожидания, однако требуется дальнейшее исследование для изучения поведения модели. Учитывая, что Ankh был выпущен в январе 2023 года, на данный момент сложно определить, какой именно вклад он вносит в высокую точность модели. Возможно, как сама постановка задачи, так и использование Ankh, способствуют достижению превосходных результатов простой классификацей.

5 Дальнейшая работа

Проверка модели на не аннотированных данных: Для оценки реальной применимости модели и ее обобщающей способности, необходимо провести проверку на данных, которые не использовались при обучении. Это позволит оценить ее способность обнаруживать и классифицировать Cas белки в новых и неизвестных образцах. Классификацию найденных Cas белков нужно будет проверить либо другими вычислительными методами, либо эксперементально.

Обновление датасета: Поскольку датасет является основой для обучения модели, обновление его данных поможет улучшить точность и актуальность модели. Использование более нового датасета позволит включить новые открытия и исследования в области Cas белков и CRISPR-Cas системы.

Чистка кода и упрощение использования: Для улучшения эффективности и удобства работы с моделью необходимо провести чистку кода, оптимизацию и упрощение процесса использования модели. Это может включать улучшение документации, создание более интуитивного пользовательского интерфейса или разработку удобных инструментов для работы с моделью.

Подача на включение в инструментарий для исследования CRISPR-Cas систем: Разработанная модель может представлять интерес для исследователей и специалистов в области CRISPR-Cas систем. Планируется подать заявку на включение модели в инструментарий, используемый в исследованиях и анализе CRISPR-Cas системы. Это может способствовать распространению и применению модели в научном сообществе.

Коллаборация со специалистами из области генной инженерии и изучения природы CRISPR систем: Такая коллаборация поможет более полно закрыть потребности исследователей в индустрии.

Смена алгоритма с классификации на кластеризацию: При достижении определенного уровня точности и стабильности модели, можно рассмотреть возможность перехода от алгоритма классификации к алгоритму кластеризации. Это может расширить применение модели и помочь в идентификации новых и неизвестных типов Cas белков и их группировке на основе сходства.

Список литературы

[1] *Marraffini, Luciano A.* CRISPR-Cas immunity in prokaryotes / Luciano A Marraffini // *Nature.* — 2015. — Vol. 526, no. 7571. — Pp. 55–61.

[2] Nobel prise in chemistry for CRISPR in gene editing. https://www.nobelprize.org/prizes/chemistry/2020/popular-information/.

[3] *Pickar-Oliver, Adrian.* The next generation of CRISPR-Cas technologies and applications / Adrian Pickar-Oliver, Charles Gersbach // *Nat Rev Mol Cell Biol.* — 2019. — . — Vol. 20, no. 8. — Pp. 490–507.

[4] *Pearson, William R.* An introduction to sequence similarity ("homology") searching / William R Pearson // *Current protocols in bioinformatics.* — 2013. — Vol. 42, no. 1. — Pp. 3–1.

[5] CRISPRcasIdentifier: Machine learning for accurate identification and classification of CRISPR-Cas systems / Victor A Padilha, Omer S Alkhnbashi, Shiraz A Shah et al. // *GigaScience.* — 2020. — Vol. 9, no. 6. — P. giaa062.

[6] *Koonin, Eugene V.* Diversity, classification and evolution of CRISPR-Cas systems / Eugene V Koonin, Kira S Makarova, Feng Zhang // *Current opinion in microbiology.* — 2017. — Vol. 37. — Pp. 67–78.

[7] CRISPR–Cas: complex functional networks and multiple roles beyond adaptive immunity / Faure, Guilhem, Makarova et al. // *Journal of molecular biology.* — 2019. — Vol. 431, no. 1. — Pp. 3–20.

[8] CRISPR Arrays Away from cas Genes / Sergey A. Shmakov, Irina Utkina, Yuri I. Wolf et al. // *The CRISPR Journal.* — 2020. — Vol. 3, no. 6. — Pp. 535–549. — PMID: 33346707. https://doi.org/10.1089/crispr.2020.0062.

[9] CRISPRloci: comprehensive and accurate annotation of CRISPR–Cas systems / Omer S Alkhnbashi, Alexander Mitrofanov, Robson Bonidia et al. // *Nucleic Acids Research.* — 2021. — Vol. 49, no. W1. — Pp. W125–W130.

[10] https://commons.wikimedia.org/wiki/File:Crispr$_{-r}u.png$.

[11] C2c2 is a single-component programmable RNA-guided RNA-targeting CRISPR effector / Omar O Abudayyeh, Jonathan S Gootenberg, Silvana Konermann et al. // *Science.* — 2016. — Vol. 353, no. 6299. — P. aaf5573.

[12] CRISPR RNA maturation by trans-encoded small RNA and host factor RNase III / Deltcheva E, Chylinski K, Sharma CM et al. // *Nature.* — 2011 Mar 31;471(7340):602-7.

[13] CRISPR-Cas: Adapting to change / Simon A. Jackson, Rebecca E. McKenzie, Robert D. Fagerlund et al. // *Science.* — 2017. — Vol. 356, no. 6333. — P. eaal5056.

[14] The biology of CRISPR-Cas: backward and forward / Frank Hille, Hagen Richter, Shi Pey Wong et al. // *Cell.* — 2018. — Vol. 172, no. 6. — Pp. 1239–1259.

[15] *Klompe.* Harnessing "A Billion Years of Experimentation": the ongoing exploration and exploitation of CRISPR–Cas immune systems / Klompe, Sternberg // *The CRISPR journal.* — 2018. — Vol. 1, no. 2. — Pp. 141–158.

[16] *Koonin, Eugene V.* Evolutionary genomics of defense systems in archaea and bacteria / Eugene V Koonin, Kira S Makarova, Yuri I Wolf // *Annual review of microbiology.* — 2017. — Vol. 71. — Pp. 233–261.

[17] Evolutionary classification of CRISPR–Cas systems: a burst of class 2 and derived variants / Kira S Makarova, Yuri I Wolf, Jaime Iranzo at al. // *Nat. Rev. Microbiol.* — 2020. — . — Vol. 18, no. 2. — Pp. 67–83.

[18] Cas13b is a type VI-B CRISPR-associated RNA-guided RNase differentially regulated by accessory proteins Csx27 and Csx28 / Aaron A Smargon, David B T Cox, Neena K Pyzocha at al. // *Mol. Cell.* — 2017. — . — Vol. 65, no. 4. — Pp. 618–630.e7.

[19] A cyclic oligonucleotide signaling pathway in type III CRISPR-Cas systems / Migle Kazlauskiene, Georgij Kostiuk, Česlovas Venclovas at al. // *Science.* — 2017. — . — Vol. 357, no. 6351. — Pp. 605–609.

[20] Type III CRISPR-Cas systems produce cyclic oligoadenylate second messengers / Ole Niewoehner, Carmela Garcia-Doval, Jakob T Rostøl at al. // *Nature.* — 2017. — . — Vol. 548, no. 7669. — Pp. 543–548.

[21] CRISPRDetect: A flexible algorithm to define CRISPR arrays / Ambarish Biswas, Raymond H J Staals, Sergio E Morales et al. // *BMC Genomics.* — 2016. — . — Vol. 17, no. 1. — P. 356.

[22] *Li, Bin.* CRISPR-SE: a brute force search engine for CRISPR design / Bin Li, Poshen B Chen, Yarui Diao // *NAR Genomics and Bioinformatics.* — 2021. — 02. — Vol. 3, no. 1. — lqab013. https://doi.org/10.1093/nargab/lqab013.

[23] CRISPRidentify: identification of CRISPR arrays using machine learning approach / Alexander Mitrofanov, Omer S Alkhnbashi, Sergey A Shmakov et al. // *Nucleic Acids Research.* — 2020. — 12. — Vol. 49, no. 4. — Pp. e20–e20. https://doi.org/10.1093/nar/gkaa1158.

[24] Systematic prediction of genes functionally linked to CRISPR-Cas systems by gene neighborhood analysis / Sergey A. Shmakov, Kira S. Makarova, Yuri I. Wolf et al. // *Proceedings of the National Academy of Sciences.* — 2018. — Vol. 115, no. 23. — Pp. E5307–E5316. https://www.pnas.org/doi/abs/10.1073/pnas.1803440115.

[25] *Mann, Matthias.* Proteomic analysis of post-translational modifications / Matthias Mann, Ole N Jensen // *Nat. Biotechnol.* — 2003. — . — Vol. 21, no. 3. — Pp. 255–261.

[26] Generative deep neural networks for dialogue: A short review / Iulian Vlad Serban, Ryan Lowe, Laurent Charlin, Joelle Pineau. — 2016. — .

[27] Attention is all you need / Ashish Vaswani, Noam Shazeer, Niki Parmar et al. // *Advances in neural information processing systems.* — 2017. — Vol. 30.

[28] https://www.semanticscholar.org/paper/Machine-learning-materials-physics%3A-Integrable-deep-Teichert-Natarajan/0bdc7ce61069969c9ef598dfcbf9f5a6aa798aa3.

[29] https://towardsdatascience.com/transformers-explained-visually-not-just-how-but-why-they-work-so-well-d840bd61a9d3.

[30] Language models of protein sequences at the scale of evolution enable accurate structure prediction / Zeming Lin, Halil Akin, Roshan Rao et al. // *bioRxiv*. — 2022.

[31] Rita: a study on scaling up generative protein sequence models / Daniel Hesslow, Niccoló Zanichelli, Pascal Notin et al. // *arXiv preprint arXiv:2205.05789*. — 2022.

[32] Ankh: Optimized Protein Language Model Unlocks General-Purpose Modelling / Ahmed Elnaggar, Hazem Essam, Wafaa Salah-Eldin et al. // *arXiv preprint arXiv:2301.06568*. — 2023.

[33] Contrastive learning on protein embeddings enlightens midnight zone / Michael Heinzinger, Maria Littmann, Ian Sillitoe et al. // *NAR genomics and bioinformatics*. — 2022. — Vol. 4, no. 2. — P. lqac043.

[34] CATH: increased structural coverage of functional space / Ian Sillitoe, Nicola Bordin, Natalie Dawson et al. // *Nucleic acids research*. — 2021. — Vol. 49, no. D1. — Pp. D266–D273.

[35] Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International. https://creativecommons.org/licenses/by/4.0/.

[36] Github repository with all the code written for this work. https://github.com/Dartilder/Cas$_{d}eep_{s}earch/tree/main$.

[37] NCBI Genomes databse. https://www.ncbi.nlm.nih.gov/genome/.

[38] Biopython. https://biopython.org/docs/1.75/api/index.html.

[39] Evaluation of a Tree-based Pipeline Optimization Tool for Automating Data Science / Randal S. Olson, Nathan Bartley, Ryan J. Urbanowicz, Jason H. Moore // Proceedings of the Genetic and Evolutionary Computation Conference 2016. — GECCO '16. — New York, NY, USA: ACM, 2016. — Pp. 485–492. http://doi.acm.org/10.1145/2908812.2908918.

[40] *Geng, Chuanxing*. Recent advances in open set recognition: A survey / Chuanxing Geng, Sheng-jun Huang, Songcan Chen // *IEEE transactions on pattern analysis and machine intelligence*. — 2020. — Vol. 43, no. 10. — Pp. 3614–3631.