Санкт-Петербургский государственный университет

ЛУНИНА Полина Сергеевна

Выпускная квалификационная работа

Комбинирование нейронных сетей и синтаксического анализа для предсказания вторичной структуры генетических цепочек

Уровень образования: магистратура

Направление *09.04.04 «Программная инженерия»*

Основная образовательная программа *ВМ.5666.2019 «Программная инженерия»*

Научный руководитель:
к.ф.-м.н., доцент Григорьев С.В.

Рецензент:
Специалист по анализу данных ООО «Интеллоджик» Малыгина Т.С.

Санкт-Петербург
2021

Saint Petersburg State University

Polina Lunina

Master's Thesis

Secondary structure prediction by combination of formal grammars and neural networks

Education level: master

Speciality *09.04.04 "Software Engineering"*

Programme *BM.5666.2019 "Software Engineering"*

Scientific supervisor:
C.Sc., docent S.V. Grigorev

Reviewer:
Data scientist at "Intellogic" LLC T.S. Malygina

Saint Petersburg
2021

Оглавление

1. Введение

Среди множества направлений научных исследований, которое охватывает вычислительная биология, особое место занимают различные прикладные задачи, связанные с анализом последовательностей, входящих в состав важнейших для всех живых организмов макромолекул — ДНК, РНК и белков. Процесс разработки и оптимизации алгоритмов для решения целого ряда задач, например, классификации организмов, расшифровки геномов, предсказания функций белков и других, не прекращается уже много лет, и за это время были выработаны некоторые общие принципы работы с биологическими данными.

Во-первых, линейная (первичная) и пространственная (вторичная) структуры этих макромолекул содержат важную информацию о клеточных функциях и эволюционном происхождении организмов и могут быть формально описаны с помощью различных математических моделей. В частности, первичная структура молекулы РНК представляет собой цепочку особых веществ — нуклеотидов, — и в том случае, когда два фрагмента этой цепи соединяются друг с другом, перегибаясь и образуя на конце неспаренный участок в форме петли, формируется элемент, называемый в биологии шпилькой (stem-loop). Совокупность вложенных шпилек разных размеров составляет сложную и стабильную вторичную структуру. Известно, что вторичная структура играет важную роль в регуляции клеточных процессов [42], поэтому во многих геномных исследованиях требуется учитывать или предсказывать ее конфигурацию. Существуют различные методы формального описания вторичной структуры, например, скрытые марковские модели [40], ковариационные модели [34] и формальные грамматики [8, 19, 31].

Во-вторых, при работе с биологическими данными важно учитывать их потенциальную зашумленность, т.е. присутствие различных неточностей, мутаций и случайных всплесков, и, кроме того, законы образования пространственных молекулярных структур сами по себе имеют стохастическую природу. Поэтому в данной области у точных алгоритмов зачастую выигрывают те, что предполагают некоторого рода

вероятностную оценку. Популярным способом обработки зашумленных данных являются методы машинного обучения, в частности, нейронные сети, которые в настоящее время успешно используются в том числе и в биоинформатике [13, 33].

В рамках моей бакалаврской дипломной работы был разработан подход для решения задач обработки последовательностей, обладающих некоторой синтаксической структурой [11]. Данный подход основан на комбинировании методов синтаксического анализа и машинного обучения и может быть применен в совершенно разных предметных областях. Предлагается использовать формальную грамматику для кодирования характерных элементов синтаксической структуры, алгоритм синтаксического анализа — для их поиска во входных данных, а обработку информации о наличии и расположении этих элементов в цепочке и вероятностную оценку провести с помощью нейронной сети, которая некоторым образом обрабатывает сгенерированные парсером данные. Анализ геномных последовательностей является одной из потенциальных областей применения этого подхода и, если говорить непосредственно об исследовании РНК, то входными данными являются нуклеотидные цепочки, под синтаксической структурой следует понимать вторичную структуру РНК, а под искомыми характерными элементами — составляющие ее шпильки.

Направлением исследования, представленного в данной работе, является предсказание вторичной структуры РНК с использованием описанного выше подхода. Правила контекстно-свободной грамматики описывают определенный по некоторым эмпирическим критериям общий вид шпилек вторичной структуры, а синтаксический анализатор выполняет задачу поиска подстроки в строке, что с теоретической точки зрения означает получение всех выводимых по правилам грамматики подстрок, а с практической — всех потенциально возможных в данной строке шпилек. Однако в контексте реальной вторичной структуры РНК живого организма эта информация является избыточной, так как из всех возможных комбинаций шпилек будет присутствовать только какая-то одна, а иногда и недостаточной, потому что грамма-

тика не может не содержать определенные ограничения, например, на максимальный размер петли внутри шпильки. Поэтому для генерации чистой вторичной структуры из результата работы парсера в рамках рассматриваемого подхода предлагается использовать нейронную сеть, задача которой в данном случае — отфильтровать лишние шпильки и достроить невыразимые в грамматике элементы.

2. Постановка задачи

Целью данной работы является исследование возможности применения подхода, основанного на комбинировании методов синтаксического анализа и машинного обучения, к задаче предсказания вторичной структуры молекулы РНК. Для реализации данной цели были поставлены следующие задачи.

- Разработка архитектуры решения, конкретизирующей форматы анализируемых данных, а также используемые формальные грамматики и нейронные сети.
- Проведение экспериментальных исследований предложенной архитектуры, сравнение полученных результатов с существующими решениями.

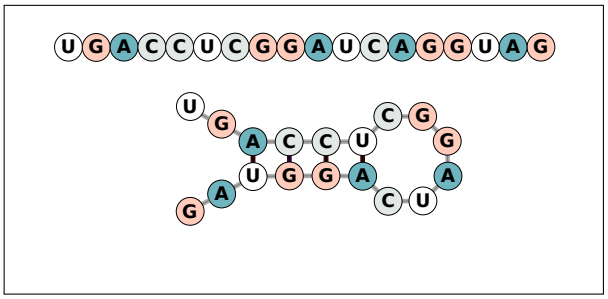
3. Обзор

3.1. Описание предметной области

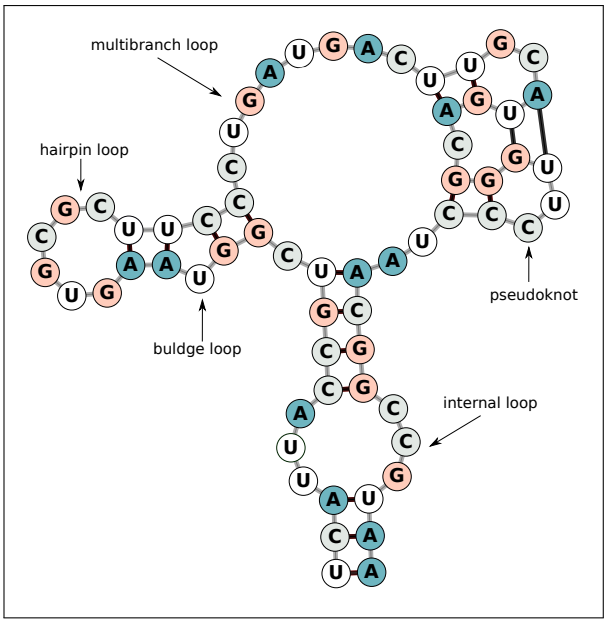
Входящие в состав клеток всех живых существ макромолекулы — нуклеиновые кислоты и белки — имеют очень разнообразные биологические функции и играют важные роли во множестве процессов, определяющих жизнедеятельность и эволюционное развитие организмов. Со структурной точки зрения их объединяет то, что они являются полимерами, т.е. состоят из некоторых элементарных единиц — мономеров, — соединенных в крупные молекулы по определенным законам. Как ДНК и РНК, так и белки имеют несколько уровней организации, т.е. характеризуются линейной (первичной) структурой, которая представляет собой последовательность из мономеров, и пространственной (вторичной), основанной на молекулярных взаимодействиях между ними.

В частности, молекула РНК состоит из цепи особых веществ — нуклеотидов четырех типов, которые называются аденин, гуанин, цитозин и урацил (A, G, C, U) и могут образовывать между собой попарные водородные связи. Законы формирования нуклеотидных пар обусловлены поддержанием термодинамического равновесия вторичной структуры, и самыми стабильными и распространенными в природе являются канонические, Уотсон-Криковские, пары $A - U$, $C - G$, однако с разной степенью вероятности могут встречаться и любые другие комбинации. Способность нуклеотидов формировать связи приводит к склеиванию определенных участков цепи РНК между собой и образованию вторичной структуры, базовым элементом которой является так называемая шпилька, которая образуется в том случае, когда две последовательности одной и той же цепи соединяются друг с другом, перегибаясь одна к другой и образуя на конце неспаренный участок — петлю. Пример элементарной шпильки высоты четыре, образованной только каноническими парами, показан на рис. 1а, и вторичная структура РНК в целом может быть представлена как рекурсивная композиция шпилек варьирующегося размера [22]. В различных работах по биоинформатике мож-

но встретить разделение элементов вторичной структуры по внешнему виду на несколько типов (hairpin, internal loop, bulge, helix, multi-loop и др.), однако формально все эти элементы могут быть определены через вложенные шпильки (рис. 1b). Особый интерес представляют псевдоузлы (pseudoknots), состоящие из двух шпилек, где половина стебля одной из них располагается между двумя половинами стебля другой, и имеющее большое функциональное значение для живых организмов.



(a) Простая шпилька



(b) Многообразие элементов

Рис. 1: Вторичная структура РНК

Различные исследования свойств и особенностей вторичной структуры РНК показали, что она активно участвует в таких процессах регуляции генома, как транскрипция и трансляция [37], а также играет важную роль для филогенетического и таксономического анализа последовательностей [27, 36], поэтому точное предсказание вторичной структуры РНК по ее нуклеотидной цепи относится к ключевым задачам современной геномики, которая усложняется тем, что вторичная структура обладает широкой вариативностью и в большинстве случаев не описывается исключительно через композицию простых шпилек, образованных Уотсон-Криковскими парами, а содержит более нетривиальные элементы.

3.2. Подходы к предсказанию вторичной структуры РНК

Существует большое количество методов для предсказания вторичной структуры РНК, основанных на совершенно разных концепциях и технологиях. Самыми точными являются результаты, полученные в лабораторных условиях, например с помощью рентгеноструктурного анализа [38] или ядерного магнитного резонанса [20], тем не менее, высокая цена и сложность постановки таких экспериментов приводят к активному развитию альтернативных методов — вычислительных.

Вычислительные методы предсказания вторичных структур можно поделить на две основные группы. К первой относятся техники сравнительного анализа гомологичных последовательностей, основанные на идее о том, что биологически важные вторичные структуры не сильно меняются в процессе эволюции, поэтому сохранение двух довольно удаленных друг от друга нуклеотидов указывает на наличие водородной связи между ними во вторичной структуре [12, 25]. Такого рода результаты представляются достаточно надежными и часто используются в качестве эталонных в различных экспериментах, однако данный подход требует значительного объема ручной работы по выравниванию последовательностей и наличия достаточного количества родственных РНК для каждого вида, что далеко не всегда реализуемо на практике. Вторая и наиболее интересная в рамках данной работы группа объединяет все методы, предсказывающие вторичную структуру для единичной последовательности РНК путем построения некоторой описывающей ее физической или математической модели и решения проблемы ее оптимизации. Одним из самых популярных здесь подходов является принцип минимизации свободной энергии, основанный на требовании термодинамической стабильности вторичной структуры. Для решения задачи минимизации энергии могут применяться разные техники, в частности, динамическое программирование [28, 30], эвристические алгоритмы [14, 32] или иные оптимизационные схемы [18, 29]. Кроме того, существует ряд методов, не основанных на физически из-

меряемых параметрах, в частности, часто максимизируется на всем пространстве вторичных структур некоторая функция оценки точности предсказания [3, 21] или используются стохастические контекстно-свободные грамматики для вероятностного моделирования вторичной структуры [8, 19].

Теоретическое понимание механизмов свертки последовательностей РНК и практическая реализация соответствующей модели являются достаточно нетривиальными задачами, что стало поводом к активному внедрению различных техник машинного обучения в процесс разработки алгоритмов предсказания вторичных структур как в качестве способа оценки оптимизируемых строгим алгоритмом параметров [1, 7], так и в качестве основного используемого метода [24, 26].

Несмотря на быстрое развитие идей и технологий в области вычислительной геномики, на данный момент проблема предсказания вторичной структуры молекулы РНК остается открытой, и основными сложностями при разработке алгоритмов являются предсказание псевдоузлов, неканонических пар оснований и обработка длинных последовательностей.

3.3. Используемые технологии

Предлагаемый в данной работе подход основан на комбинировании методов синтаксического анализа и машинного обучения, и в данном разделе будут введены основные используемые далее понятия из этих двух областей.

3.3.1. Синтаксический анализ

Алфавитом называется любое конечное непустое множество символов, и если V — алфавит, то через V^* обозначается множество всех строк, составленных из символов V, а $L \subseteq V^*$ называется языком над алфавитом V. Для описания структуры конкретного языка, т.е. для выделения определенного подмножества из множества всех строк заданного алфавита, используется абстракция, именуемая грамматикой.

В задании правил грамматики участвуют терминальные символы, т.е. элементарные единицы некоторого языка, и нетерминальные — синтаксические переменные, которые могут быть заменены группами терминальных символов. Формально, грамматикой называется четверка $G = (V_T, V_N, P, S)$, где V_T — алфавит терминалов, V_N — алфавит нетерминалов, через P обозначается конечное множество правил вида $\alpha \to \beta$, где $\alpha \in V^* V_N V^*$, $\beta \in V^*$, $V = V_N \cup V_T$, а S — стартовый нетерминал грамматики, т.е. тот нетерминал, из которого могут быть получены все предложения задаваемого ей языка. В теории формальных языков описаны различные классы грамматик, и в данной работе интерес для нас будут представлять контекстно-свободные грамматики, в которых для каждого правила $\alpha \to \beta$ выполняется условие $\alpha \in V_N$.

Строка w называется выводимой из нетерминального символа N грамматики G, если w может быть получена из N путем применения некоторой последовательности правил из множества правил P. Для автоматизации проверки выводимости строк используются различные алгоритмы синтаксического анализа, среди которых важное место занимают табличные алгоритмы, в процессе работы заполняющие для строки w и нетерминала грамматики N особую матрицу — матрицу разбора M_N, в которой $M_N[i][j] = 1$ тогда и только тогда, когда подстрока $w[i..j]$ выводима из N. Самым известным табличным алгоритмом, работающим с контекстно-свободными грамматиками, является CYK [5, 17, 41], на ключевых принципах работы которого основываются и многие современные алгоритмы.

В рамках исследовательского проекта YaccConstructor [39] лаборатории языковых инструментов JetBrains [16] проводятся исследования в области формальных языков. Среди прочего, в рамках данного проекта разрабатываются различные алгоритмы синтаксического анализа, и в данной работе использован алгоритм, основанный на матричных операциях [2], который демонстрирует высокую производительность на практике в связи с использованием параллельных вычислений.

3.3.2. Машинное обучение

К машинному обучению относятся методы создания компьютерных систем, способных находить закономерности в больших объемах данных через некоторым образом организованный процесс самостоятельного обучения. Один из таких методов — нейронные сети — повсеместно применяется в науке и технике, и концептуальной основой для построения обучаемой математической модели в данном случае являются принципы работы нейронов человеческого мозга.

При проектировании нейросети в первую очередь следует определить ее архитектуру, и одной из классических архитектур являются сверточные нейронные сети (convolutional neural networks), обрабатывающие, как правило, различного рода изображения. Такие сети работают на основе фильтров, которые отвечают за распознавание определенных характеристик изображения. Фильтр — это коллекция ядер свертки, т.е. небольших матриц из чисел (весов), которые заранее неизвестны и устанавливаются в процессе обучения. Такие матрицы обрабатывают изображения по фрагментам с целью обнаружения искомых характеристик, осуществляя операцию свертки, которая является суммой произведений элементов фильтра и матрицы входных сигналов. Для решения сложных задач с многоуровневым выявлением признаков на практике требуются многослойные сверточные сети, которые часто оказывается затруднительно оптимизировать — с увеличением глубины начинает падать точность вследствие особенностей алгоритмов обновления весов. Для решения этой проблемы была предложена особая архитектура — остаточные нейронные сети (residual neural networks) [6], — основанная на добавлении дополнительных соединений быстрого доступа между блоками слоев, что упрощает процесс обучения на начальных этапах и позволяет улучшать точность результата с увеличением глубины.

Помимо архитектуры, важным аспектом разработки нейронной сети является определение минимизируемой в процессе обучения функции ошибки (loss function), а также подбор оптимальных гиперпарамет-

ров, к которым относятся число слоев и нейронов в них, коэффициент скорости обучения (learning rate), число итераций обучения, функция активации, размер батча, способ разделения данных на обучающую и тестовую выборки и т.д. Качество работы нейросети оценивается по результатам вычисления различных метрик на тестовой выборке.

Существует множество платформ для создания и обучения нейронных сетей, и в данной работе были использованы написанные на языке Python библиотека Keras [4] и фреймворк Tensorflow [35], так как данные технологии сочетают в себе удобство использования и высокую производительность.

4. Архитектура решения

Предложенный в предыдущей дипломной работе подход предназначен для решения задач анализа строковых данных, обладающих некоторого рода синтаксической структурой, которая, наряду с непосредственно символами рассматриваемых строк, является важным источником информации о каких-либо характеристиках входных данных, но при этом оказывается слишком сложной для формализации.

В рамках данного подхода характерные элементы синтаксической структуры необходимо описать средствами формальной грамматики, а для поиска во входных данных подстрок, подходящих под это описание, использовать алгоритм синтаксического анализа. Затем извлеченные парсером элементы синтаксической структуры предлагается использовать в процессе обучения нейронных сетей, спроектированных для решения поставленной задачи. Основная идея такого комбинирования формальных грамматик и нейронных сетей заключается в том, что достаточно простая грамматика призвана формализовать только базовые и, вероятно, неполные законы образования синтаксической структуры последовательностей, и предполагается, что нейронной сети будет достаточно этой информации для выявления уже более комплексных, стохастических закономерностей, необходимых для решения некоторой аналитической задачи. Стоит отметить, что в первоначальной формулировке данный подход не ограничивает ни спектр решаемых задач, ни используемые типы грамматик и технологии, а лишь описывает набор действий для обработки определенного класса данных. Соответствующая архитектура с точки зрения физической реализации представлена на рис. 2, и необходимыми компонентами для проведения любого рода экспериментальных исследований в рамках предлагаемого подхода являются модули для синтаксического анализа и нейронных сетей, хранилище всех необходимых данных (входных последовательностей, метаданных и т.д.), а также — формальная грамматика, представленная в принимаемом парсером формате.

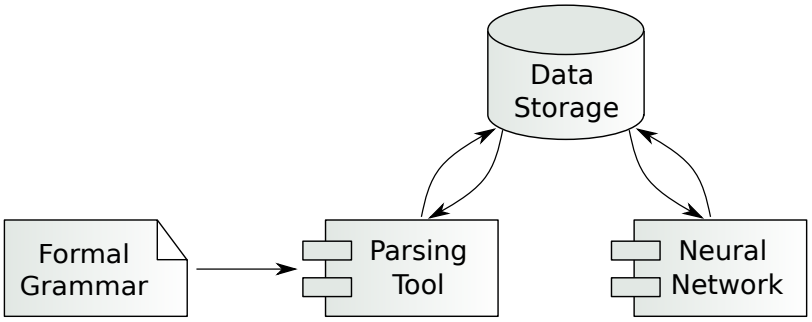


Рис. 2: Архитектурные компоненты предложенного подхода

Одной из потенциальных областей применения такого подхода является биоинформатика, в частности, различные задачи анализа РНК, где в качестве символьных последовательностей можно рассмотреть нуклеотидные цепочки РНК различных организмов, а в качестве синтаксической структуры — биологическую вторичную структуру молекулы РНК. В текущей работе исследуется возможность применения описанного выше подхода для решения задачи предсказания вторичной структуры РНК. Далее будет описана разработанная для этого архитектура решения, включающая в себя два основных шага: задание грамматики для поиска характерных элементов вторичной структуры, а затем – проектирование и обучение нейронных сетей, генерирующих для последовательности РНК максимально близкую к реальной вторичную структуру на основе полученных с помощью парсера данных.

4.1. Формальная грамматика

Первичная структура молекулы РНК представляет собой цепочку из нуклеотидов четырех типов (аденин, цитозин, гуанин и урацил), что в терминах синтаксического анализа есть последовательность символов алфавита {A, C, G, U}; вторичная же структура образовывается вследствие того, что некоторые участки первичной соединяются между собой, формируя рекурсивную композицию из шпилек разного размера и степени вложенности. Обобщенный вид таких шпилек может быть фор-

мализован средствами достаточно простой контекстно-свободной грамматики, каковой является используемая в данной работе грамматика G_0 (рис. 3). Грамматика учитывает только Уотсон-Криковские правила формирования нуклеотидных пар $A-U$, $C-G$ (строка **5**) и описывает рекурсивные композиции шпилек высоты от трех и более (строки **7-12**). Размер петли внутри шпильки лежит в пределах от одного до двадцати нуклеотидов, и такую же длину имеют последовательности, расположенные между любыми двумя шпильками (строка **2**). Эти числа были выбраны путем балансирования между следующими двумя соображениями: соответствие эмпирическим наблюдениям биологических данных и адекватность напрямую зависящих от длины и сложности грамматики временных затрат на работу парсера. По тем же причинам в G_0 не были включены неканонические нуклеотидные связи, которые могут встречаться в реальной вторичной структуре РНК — для того, чтобы учесть все возможные пары нуклеотидов, придется ввести большое количество правил, имеющих вероятностную природу. Кроме того, средствами контекстно-свободных грамматик невыразимы псевдоузлы, однако псевдоузел есть комбинация из двух шпилек, следовательно, G_0, не описывая псевдоузел как единое целое, позволяет, тем не менее, выделить из входной последовательности обе составляющие его подстроки. Здесь становится понятным основное отличие нашего подхода от классического использования формальных грамматик в данной области [8,19,31] — мы не пытаемся ни смоделировать вторичную структуру целиком, ни описать все возможные закономерности ее образования, но разбиваем ее на простые составные части, синтезировать из которых более сложные объекты предлагается уже с помощью нейронных сетей, что кратно уменьшает интеллектуальные и вычислительные затраты на синтаксический анализ.

Рассмотрим теперь формальный вид и практический смысл результата работы синтаксического анализатора для вышеописанной грамматики и последовательности РНК некоторого организма. Синтаксический анализ в данном случае используется для поиска всех подстрок входной строки, выводимых из стартового нетерминала $s1$ грамматики

```
s1: stem<s0>
any_str : any_smb*[1..20]
s0: any_str | any_str stem<s0> s0
any_smb: A | U | C | G
stem1<s>: A s U | G s C | U s A | C s G
stem2<s>: stem1< stem1<s> >
stem<s>:
      A stem<s> U
    | U stem<s> A
    | C stem<s> G
    | G stem<s> C
    | stem1< stem2<s> >
```

Рис. 3: Контекстно-свободная грамматика G_0 для описания шпилек вторичной структуры РНК

ки G_0, иными словами, для поиска тех участков этой строки, которые, в терминах G_0, могут свернуться в шпильки при формировании вторичной структуры. Формально, для входной строки w парсер заполнит верхнетреугольную булеву матрицу — матрицу разбора M_P, где $M_P[i,j] = 1 \iff$ подстрока $w[i..j]$ выводится в G_0. Так как интересующие нас шпильки должны иметь высоту от трех, каждой шпильке высоты n в матрице разбора будет соответствовать цепочка из $n-2$ единиц. На рис. 4 представлен результат работы парсера для изображенного на рис. 1а случая последовательности, сворачивающейся в шпильку высоты четыре. Каждой нуклеотидной связи, образующей шпильку высоты от трех (сплошные линии голубого цвета), соответствует единица в ячейке матрицы разбора, при этом очевидно, что шпилька высоты три инкапсулирует в себе шпильки высоты два и один, обозначенные на рисунке пунктирными линиями. Помимо уже знакомой нам шпильки с рис. 1а, в данной строке парсер обнаружил еще одну выводимую в грамматике подстроку (единица в позиции $[0, 11]$): таким образом, для рассматриваемой цепочки существует два теоретически возможных варианта свертки, из которых реализованным на практике оказался только один.

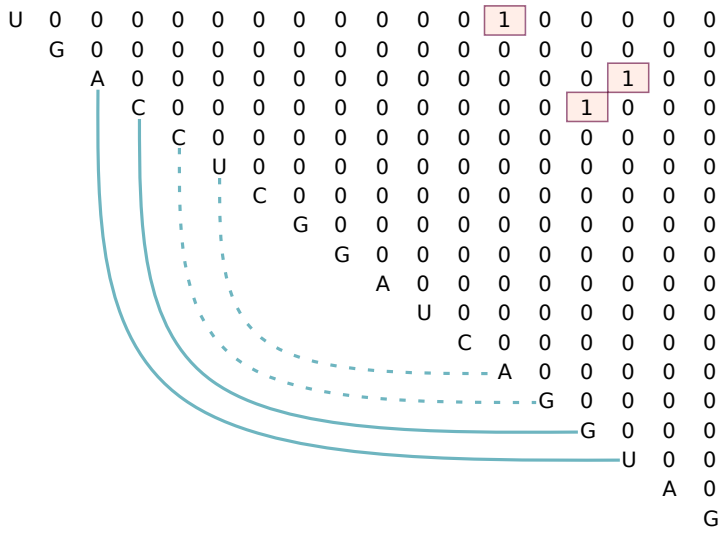
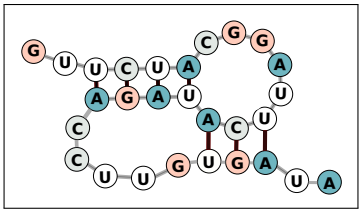


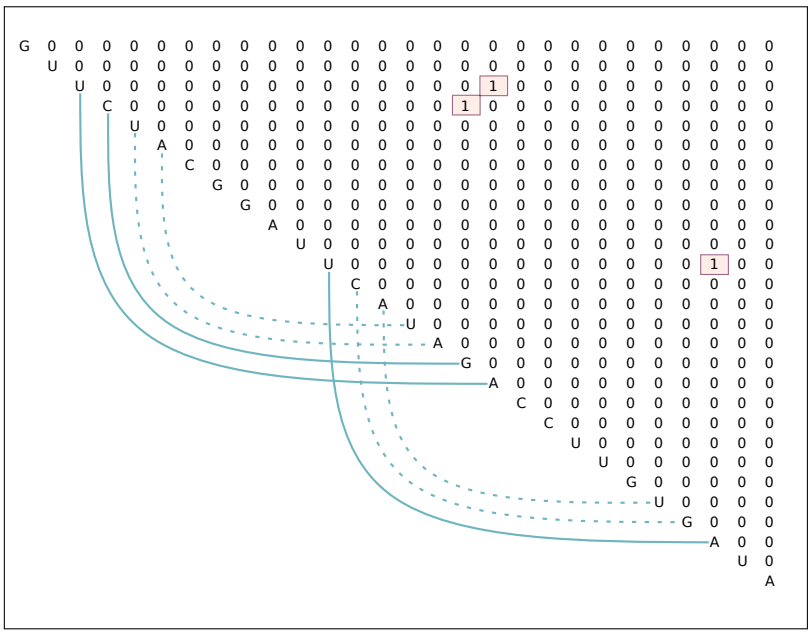
Рис. 4: Матрица разбора для последовательности РНК

Остановимся в контексте предложенного подхода на проблеме обработки псевдоузлов, которые, как уже упоминалось ранее, не выводимы в используемой грамматике G_0. На рис. 5a представлен пример последовательности, сворачивающейся в псевдоузел, а на рис. 5b — соответствующая данной последовательности матрица разбора. Рассматриваемый псевдоузел состоит из двух взаимопересекающихся шпилек высоты три и четыре, каждая из которых по отдельности выводима в G_0 и, следовательно, будет отражена в матрице разбора одной и двумя единицами соответственно. Несмотря на то, что на этапе синтаксического анализа еще не известно, образуют ли эти две найденные шпильки псевдоузел или же являются просто двумя теоретически возможными вариантами свертки цепочки, для нашего подхода предсказание псевдоузлов не становится ни сложностью, ни ограничением, так как матрицы разбора содержат всю необходимую о них информацию, которая должна быть более четко интерпретирована уже на этапе анализа данных нейронными сетями.

Таким образом, матрицы разбора хранят информацию о всех возможных расположениях шпилек вторичной структуры во входных последовательностях, однако на данный момент это только теоретические, искусственные объекты, для соотнесения которых с реальными



(a) Псевдоузел



(b) Матрица разбора для последовательности с псевдоузлом

Рис. 5: Обработка псевдоузлов в рамках предложенного подхода

биологическими данными требуется последующая обработка, и об этом будет подробно рассказано в следующем разделе.

4.2. Нейронная сеть

На данном этапе поставленная задача конкретизируется до следующей — разработать нейронную сеть, которая принимает на вход матрицы, полученные синтаксическим анализатором по грамматике G_0 для некоторого набора последовательностей РНК, и, обучаясь на вторичных структурах, предоставленных в качестве эталонных для рассматриваемых последовательностей, оптимизирует параметры для преобразования матриц разбора в корректные вторичные структуры. Данный раздел посвящен описанию всех тонкостей этого процесса.

4.2.1. Подготовка данных

Входные данные для нейросети (матрицы разбора) были описаны в прошлом разделе, и теперь необходимо определить источник и формат эталонных данных. Существуют специализированные биологические базы

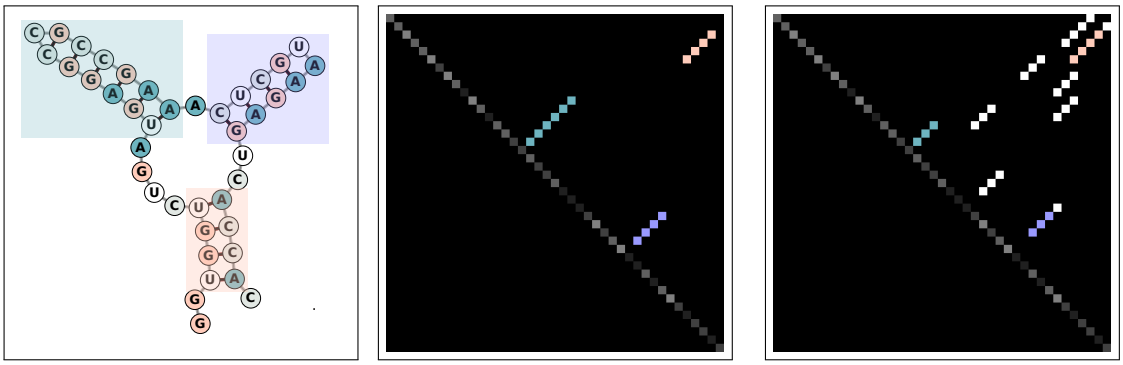
данных, в которых размещены цепочки РНК различных организмов вместе с их извлеченными из природного материала или же полученными надежными методами вторичными структурами. Такие данные оптимальны для валидации, а, следовательно, и для обучения предсказывающих вторичную структуру алгоритмов.

Как правило, в базах данных вторичные структуры РНК хранятся в скобочной (dot-bracket) нотации, из которой легко получить еще один классический формат представления вторичной структуры — так называемую матрицу контактов (contact map). Матрица контактов описывает наличие или отсутствие связи между каждыми двумя нуклеотидами последовательности: формально, для строки w это верхнетреугольная булева матрица M_C, где $M_C[i,j] = 1 \iff w[i]$ и $w[j]$ образуют пару во вторичной структуре. Как было описано в прошлом разделе, результат работы парсера на входной строке w — верхнетреугольная булева матрица M_P, где $M_P[i,j] = 1 \iff w[i..j]$ свернется в шпильку по правилам грамматики. Нетрудно проверить, что наличие контакта между нуклеотидами $w[i]$ и $w[j]$ эквивалентно тому факту, что последовательность $w[i..j]$ является шпилькой, поэтому, несмотря на то, что наше определение вторичной структуры как композиции вложенных шпилек не относится к общеупотребимым, матрицу разбора можно также описать как матрицу контактов, формируемых только выразимыми в грамматике элементами. Таким образом, использование матричного представления вторичной структуры при подготовке эталонных данных для нейронной сети представляется самым удобным в свете специфики используемых в качестве входных данных матриц разбора.

Для наглядности и удобства применения нейронных сетей мы предлагаем смотреть на матрицу контактов и матрицу разбора как на изображения: пикселями белого цвета обозначим позиции в матрицах с единицами, черного — с нулями. Матрицы разбора содержат $n-2$ единицы для каждой шпильки высоты $n > 3$, следовательно, в качестве предобработки перед обучением нейросети для каждой единицы в матрице разбора следует добавить еще две единицы в направлении главной диагонали. Кроме того, сама нуклеотидная последовательность РНК мо-

жет содержать некоторую важную информацию, поэтому предлагается закодировать ее на главной диагонали изображений равноотстающими друг от друга серыми пикселями.

На рисунке 6 продемонстрировано, как для одной и той же последовательности РНК будут выглядеть входной и эталонный образцы для нейронной сети, а также показана визуализация соответствующей вторичной структуры. Контакты, относящиеся к трем присутствующим в рассматриваемой цепочке шпилькам, на всех изображениях выделены голубым, сиреневым и розовым цветами. Данный рисунок является наглядным примером того, что далеко не все найденные парсером шпильки будут представлены в реальной вторичной структуре (белые пиксели изображения 6c). Кроме того, видно, что в сгенерированной синтаксическим анализатором матрице отсутствует несколько эталонных контактов: в данном случае это произошло из-за того, что они были образованы непредусмотренными грамматикой неканоническими нуклеотидными парами.



(a) Визуализация вторичной структуры

(b) Эталонный образец для нейросети

(c) Входной образец для нейросети

Рис. 6: Примеры представления вторичной структуры

Таким образом, в рамках данного исследования перед нейронной сетью стоит задача отфильтровать и дополнить матрицу разбора, сгенерировав корректную матрицу контактов, соответствующую максимально близкой к эталонной вторичной структуре.

4.2.2. Параллельная остаточная архитектура

Рассмотрим общую модель нейронной сети, разработанной в рамках данной работы. Входными и выходными данными являются изображения, и для решения поставленной задачи необходимо найти достаточно сложные закономерности между элементами данных, находящимися на большом расстоянии друг от друга, поэтому была использована глубокая сверточная сеть. Для оптимизации процесса обучения и повышения скорости сходимости была применена технология остаточных нейронных сетей. В процессе экспериментальных исследований нами было выявлено, что точность результатов значительно повышает использование n остаточных сетей с одинаковой архитектурой, которые обучаются параллельно на одних и тех же данных, находя в них, по всей видимости, немного разные паттерны, а затем соединяются слоем, подсчитывающим линейную комбинацию их n выходов и передающим ее уже общему остаточному блоку, завершающему обработку данных. Такая новая параллельная архитектура представлена на рис. 7; там же показано, как выглядит типичный остаточный блок (residual unit) нейронной сети, состоящий из пяти сверточных слоев с постепенно убывающими количеством фильтров и размером ядра свертки. В данной работе была использована модель, состоящая из четырех остаточных сетей с пятью одинаковыми блоками в каждой.

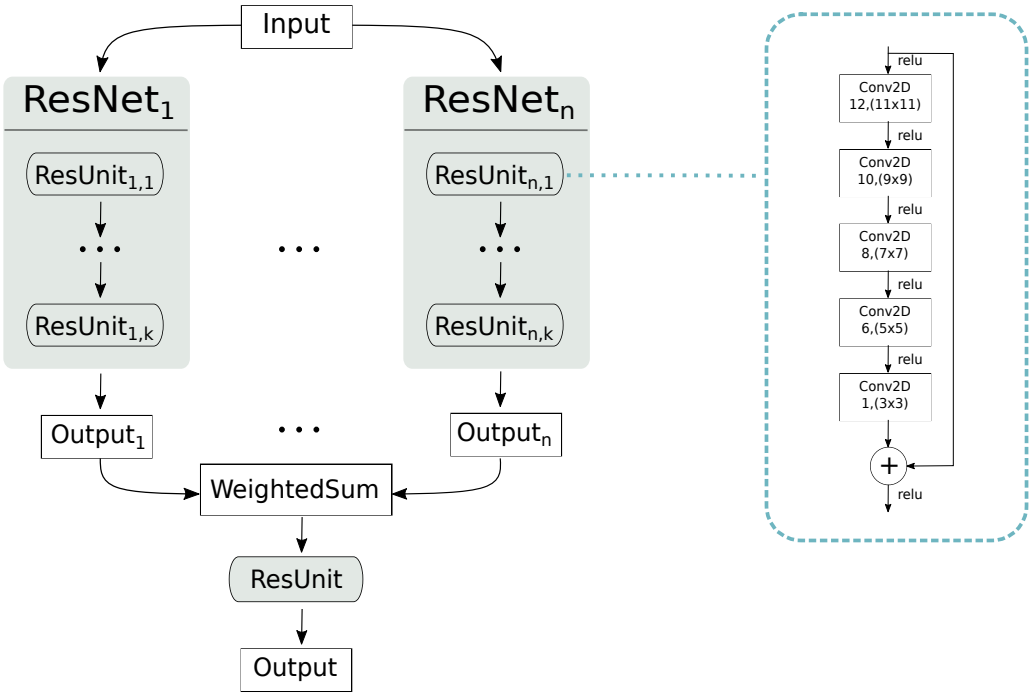


Рис. 7: Параллельная остаточная нейронная сеть

5. Эксперименты

Для экспериментальных исследований были необходимы данные двух типов: последовательности РНК для подачи на вход синтаксическому анализатору и эталонные вторичные структуры для этих последовательностей — и то, и другое было получено из популярной в исследовательских работах базы данных RNAstrand [23]. Эта база представляет собой сборку тщательно отобранных и приведенных к единому формату данных сразу из нескольких надежных баз, содержащих цепочки РНК вместе с полученными методами лабораторного эксперимента или эволюционного анализа вторичными структурами. Из выгруженных данных были удалены дубликаты и образцы с неточностями в нуклеотидной цепи или же вторичной структуре, а также было выставлено ограничение на максимальную длину последовательности — таким образом была получена выборка из 801 последовательности длин от 8 до 100, для которой были сгенерированы матрицы разбора и матрицы контактов, переведенные в черно-белые изображения. Для цепочки длины

n и входное, и эталонное изображения имеют размер $n \times n$, поэтому для корректной обработки изображений разного размера перед каждой эпохой обучения нейросети данные группировались по батчам, в каждом из которых присутствовали изображения только одного размера. Если количество образцов какого-либо размера оказывалось меньше величины batch size или же не делилось на него нацело, данные циклическим образом дублировались до достижения необходимого объема. Распределение длин последовательностей в итоговой выборке продемонстрировано на рис. 8, при этом медианным значением является 44, а средним — 47, что говорит о практически одинаковой представленности коротких и длинных цепочек среди исследуемых данных.

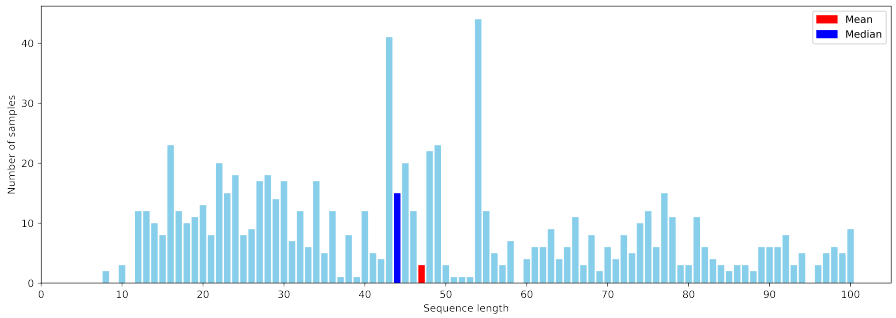


Рис. 8: Распределение длин последовательностей РНК в выборке

Для оценки качества работы обученных на данных изображениях нейронных сетей были выбраны следующие метрики, посчитанные относительно попиксельной разницы между предсказанным и эталонным изображениями. Далее TP (true positive), FP (false positive) и FN (false negative), где под positive и negative понимаются белые и черные пиксели изображений соответственно, — информация о том, сколько раз нейронная сеть приняла верное и сколько раз неверное решение по каждому пикселю (кроме диагональных) каждого изображения выборки.

- $Precision = \frac{TP}{TP+FP}$ (доля предсказанных контактов, которые действительно являются контактами в эталонном изображении).
- $Recall = \frac{TP}{TP+FN}$ (доля найденных нейронной сетью контактов среди всех искомых).

- $F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$ (гармоническое среднее $Precision$ и $Recall$, используется как удобная объединяющая метрика).

При обучении нейросети была использована функция потерь, в основе построения которой лежит идея о максимизации метрики $F1$ с несколькими уточнениями. Во-первых, $F1$ дискретна, а функция ошибки должна быть дифференцируема вследствие вычисления на ней градиента. Во-вторых, передача среднего по выборке значения $1 - F1$ в качестве функции ошибки не гарантирует отсутствие большого разброса $Precision$ и $Recall$ как в пределах отдельно взятого изображения, так и в масштабах всей выборки, следствием чего будет нестабильность качества работы модели и высокая вероятность появления очень низкой точности результата для случайно взятого тестового образца. На основании данных соображений была реализована функция $F1\_loss$, представленная на рис. 9. Здесь дифференцируемость обеспечивается заменой сумм дискретных целочисленных значений на непрерывную сумму значений вероятности, а поддержка баланса между $Precision$ и $Recall$ для каждого изображения и для выборки в целом — двумя пропорциональными величине разброса штрафными коэффициентами $k1$ и $k2$, накладываемыми на метрику $F1$.

Вследствие того, что количество обучаемых параметром используемой модели является достаточно большим относительно размера выборки, после каждого остаточного блока был добавлен слой Dropout, исключающий заданный процент случайных нейронов во время обучения. Кроме того, во всех сверточных слоях была применена регуляризация L2, которая, помимо уменьшения переобучения нейросети, оказывает положительное влияние на процесс поиска сложных закономерностей в данных. В качестве оптимизатора был использован адаптивный градиентный спуск (Adagrad) [9], удобный для работы с разреженными данными, а также автоматически настраивающий скорость обучения.

Для сравнения результатов работы обученной модели с существующими в области аналогами был проведен анализ различных инструментов, предсказывающих вторичную структуру РНК, по следующим критериям: заявленная высокая точность результатов, возможность пред-

```
from keras import backend as K

def f1_loss(y_true, y_pred):
    #normalize pixels values to [0, 1]
    y_true, y_pred = K.minimum(y_true / 255, 1), K.minimum(y_pred / 255, 1)
    #calculate differentiable versions of TW, FW and FB
    tw = K.sum(K.cast(y_true * y_pred, 'float32'), axis=[1, 2, 3])
    fw = K.sum(K.cast((1 - y_true) * y_pred, 'float32'), axis=[1, 2, 3])
    fb = K.sum(K.cast(y_true * (1 - y_pred), 'float32'), axis=[1, 2, 3])
    #calculate precision and recall secure from zero division error
    precision = tw / (tw + fw + K.epsilon())
    recall = tw / (tw + fb + K.epsilon())
    #penalty coefficients for huge difference between precision and recall
    #calculated for each image and whole dataset respectively
    k1 = 1 -  K.abs(precision - recall)
    k2 = 1 -  K.abs(K.mean(precision) - K.mean(recall))
    #calculate upgraded f1 score
    f1 = k1 * k2 * 2 * precision * recall / (precision + recall + K.epsilon())
    return 1 - K.mean(f1)
```

Рис. 9: Функция потерь нейронной сети

сказания псевдоузлов, удобство использования и адекватное время работы. На основании данных соображений были отобраны шесть инструментов, основанных на различных подходах.

- HotKnots — минимизации свободной энергии через эвристический алгоритм [14].
- SPOT-RNA — глубокое обучение, основанное на технике transfer learning [26].
- PknotsRG — минимизация свободной энергии с использованием Turner energy rules [29].
- RNAstructure — минимизация свободной энергии с помощью динамического программирования [28].
- Ipknot — поиск оптимальной вторичной структуры методом целочисленного программирования [15].
- Knotty — алгоритм для минимизация свободной энергии, основанный на разреженном динамическом программировании [18].

5.1. Результаты

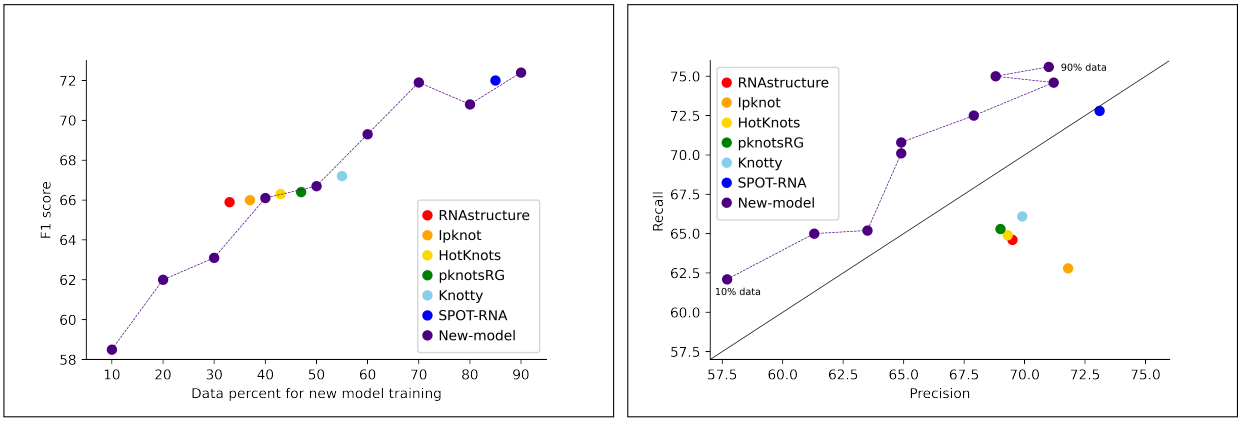
Все тестовые запуски проводились на рабочей станции со следующими характеристиками.

- Операционная система: Ubuntu 20.04.2 LTS.
- Центральный процессор: Intel Core i5-10210U CPU 1.60GHz.
- Графический процессор: NVIDIA GeForce MX250.
- Объем оперативной памяти: 7.5 GB.

На рис. 10a представлены значения метрики $F1$, показанные шестью вышеописанными инструментами на всей выборке из 801 образца, а разработанной моделью (New-model) — для различных разделений данных на обучающую и тестовую выборки (10%:90%, ..., 90%:10%). На графике видно, что при малых размерах обучающей выборки новая модель демонстрирует достаточно низкую точность, однако при увеличении выборки до 40% результаты становятся сравнимыми с остальными подходами, а при максимальном объеме выборки (90%) — лучшими в приведенном сравнении.

На рис. 10b показаны результаты аналогичного тестирования всех моделей по метрикам *Precision* и *Recall*; здесь черная прямая $y = x$ символизирует оптимальное для рассматриваемой задачи положение этих метрик — их равенство, — а фиолетовая пунктирная линия указывает направление увеличения размера обучающей выборки для нашей модели от 10% до 90% с шагом в 10%. Значения метрик для New-model расположены достаточно близко к желаемой прямой, что говорит о сбалансированности предсказаний разработанной нейросети. Кроме того, реализованный в данной работе алгоритм — единственный на данном графике, имеющий *Recall*, больший, чем *Precision*: это произошло из-за того, что парсер находит значительную часть требуемых контактов, поэтому нейронная сеть, владея этой информацией еще до начала обучения, основной своей задачей имеет улучшение точности, а не полноты

системы. Это делает наш подход несколько нетрадиционным относительно аналогов, которые, по всей видимости, сталкиваются с рядом проблем в процессе поиска контактов во вторичной структуре.



(a) Значения метрики $F1$ (b) Значения метрик *Precision* и *Recall*

Рис. 10: Сравнение разработанного подхода с аналогами

Помимо точности, важной характеристикой алгоритма в области биоинформатики является время его работы, так как исследователям часто приходится работать с достаточно большими биологическими базами данных. В таблице 1 приведены замеры времени, потраченного всеми инструментами на обработку 100 цепочек РНК различных длин из рассматриваемого промежутка от 8 до 100. Несмотря на то, что разные подходы могут предполагать разные сценарии использования (обработка одной или нескольких последовательностей, вывод ответа через интерфейс командной строки или в специальный файл, а также сохранение результатов в различных форматах), одним из традиционных вариантов является обработка файла в формате fasta, содержащего набор последовательностей с метаданными, и последующее сохранение результата в одном из общепринятых форматов (например, dot-bracket или bpseq). Для данного сценария и был произведен сравнительный анализ производительности подходов: файл с последовательности был преобразован в необходимые для всех инструментов входные форматы, выходные же форматы были оставлены без изменений. В таблице 1 представлены средние значения для десяти прогонов в секундах, упорядоченные по возрастанию времени. Инструменты Ipknot, Hotknots,

PknotsRG, RNAstructure и Knotty работают только на CPU, SPOT-RNA имеет и CPU, и GPU-реализации, а для нашего подхода как алгоритм синтаксического анализа (PA), так и нейронная сеть (NN) используют GPU. Можно увидеть, что New-model значительно проигрывает по времени большинству аналогов и наиболее времязатратной операцией здесь является синтаксический анализ, занимающий почти 80% от общего времени работы.

Таблица 1: Время работы инструментов для 100 последовательностей

Tool	**Time, s**
Ipknot	0.8
RNAstructure	10.3
PknotsRG	14.9
Hotknots	37.0
SPOT-RNA (GPU)	67.8
New-model (PA + NN)	103.1 (80.7 + 22.4)
SPOT-RNA (CPU)	109.7
Knotty	282.8

Подводя итоги, экспериментальные исследования показали работоспособность разработанного подхода применительно к задаче предсказания вторичной структуры РНК даже в сравнении с лучшими инструментами в области. Высокая точность уже полученных результатов вместе с общей гибкостью подхода и обширными возможностями для дальнейших экспериментов позволяют полагать, что предложенные в данной работе идеи имеют значительный потенциал. Однако на данный момент наш проект по большей части исследовательский — для создания полноценного инструмента требуется тщательный анализ качества всех обученных на различного размера выборках моделей с целью выбора оптимальной, а также, несомненно, повышение производительности подхода, в частности, ускорение синтаксического анализатора.

6. Заключение

В данной работе было проведено исследование возможности применения подхода, основанного на комбинировании формальных грамматик и нейронных сетей, к задаче предсказания вторичных структур РНК. Были получены следующие результаты.

- Разработана необходимая архитектура решения.
- Проведены экспериментальные исследования предложенной архитектуры применительно к задачам предсказания вторичных структур последовательностей РНК.
- Представлен постер "Secondary structure prediction by combination of formal grammars and neural networks" на конференции Biata 2020 и опубликована одноименная статья (BMC Bioinformatics, Scopus) [10].
- Исходный код и докуметация доступны по данной ссылке.

Мы предлагаем следующие направления будущего развития предложенного решения.

- Изучение возможности дальнейшего повышения точности нейронной сети путем более тонкой настройки гиперпараметров модели.
- Улучшение производительности подхода через снижение временных затрат на синтаксический анализ.
- Разработка предсказывающего вторичную структуру инструмента на основе обученной нейронной сети.

Список литературы

[1] Akiyama Manato, Sato Kengo, Sakakibara Yasubumi. A max-margin training of RNA secondary structure prediction integrated with the thermodynamic model // Journal of bioinformatics and computational biology. — 2018. — Vol. 16, no. 06. — P. 1840025.

[2] Azimov Rustam, Grigorev Semyon. Context-free path querying by matrix multiplication // Proceedings of the 1st ACM SIGMOD Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA). — 2018. — P. 1–10.

[3] CENTROIDFOLD: a web server for RNA secondary structure prediction / Kengo Sato, Michiaki Hamada, Kiyoshi Asai, Toutai Mituyama // Nucleic acids research. — 2009. — Vol. 37, no. suppl_2. — P. W277–W280.

[4] Chollet François et al. Keras. — https://keras.io. — 2015.

[5] Cocke John. Programming languages and their compilers: Preliminary notes. — New York University, 1969.

[6] Deep residual learning for image recognition / Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2016. — P. 770–778.

[7] Do Chuong B, Woods Daniel A, Batzoglou Serafim. CONTRAfold: RNA secondary structure prediction without physics-based models // Bioinformatics. — 2006. — Vol. 22, no. 14. — P. e90–e98.

[8] Dowell Robin D, Eddy Sean R. Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction // BMC bioinformatics. — 2004. — Vol. 5, no. 1. — P. 1–14.

[9] Duchi John, Hazan Elad, Singer Yoram. Adaptive subgradient methods for online learning and stochastic optimization. // Journal of machine learning research. — 2011. — Vol. 12, no. 7.

[10] Grigorev Semyon, Kutlenkov Dmitry, Lunina Polina. Secondary structure prediction by combination of formal grammars and neural networks // BMC Bioinformatics. — 2020. — Vol. 21, no. SUPPL 20.

[11] Grigorev Semyon, Lunina Polina. The Composition of Dense Neural Networks and Formal Grammars for Secondary Structure Analysis. // BIOINFORMATICS. — 2019. — P. 234–241.

[12] Gutell Robin R, Lee Jung C, Cannone Jamie J. The accuracy of ribosomal RNA comparative structure models // Current opinion in structural biology. — 2002. — Vol. 12, no. 3. — P. 301–310.

[13] Higashi SUSAN, Hungria Mariangela, Brunetto MADC. Bacteria classification based on 16S ribosomal gene using artificial neural networks // Proceedings of the 8th WSEAS International Conference on Computational intelligence, man-machine systems and cybernetics. — 2009. — P. 86–91.

[14] HotKnots: heuristic prediction of RNA secondary structures including pseudoknots / Jihong Ren, Baharak Rastegari, Anne Condon, Holger H Hoos // Rna. — 2005. — Vol. 11, no. 10. — P. 1494–1504.

[15] IPknot: fast and accurate prediction of RNA secondary structures with pseudoknots using integer programming / Kengo Sato, Yuki Kato, Michiaki Hamada et al. // Bioinformatics. — 2011. — Vol. 27, no. 13. — P. i85–i93.

[16] JetBrains Programming Languages and Tools Lab [Электронный ресурс]. — URL: https://research.jetbrains.org/groups/plt_lab/ (online; accessed: 11.05.2021).

[17] Kasami Tadao. An efficient recognition and syntax-analysis algorithm for context-free languages // Coordinated Science Laboratory Report no. R-257. — 1966.

[18] Knotty: efficient and accurate prediction of complex RNA pseudoknot structures / Hosna Jabbari, Ian Wark, Carlo Montemagno, Sebastian Will // Bioinformatics. — 2018. — Vol. 34, no. 22. — P. 3849–3856.

[19] Knudsen Bjarne, Hein Jotun. RNA secondary structure prediction using stochastic context-free grammars and evolutionary history. // Bioinformatics (Oxford, England). — 1999. — Vol. 15, no. 6. — P. 446–454.

[20] NMR spectroscopy of RNA / Boris Fürtig, Christian Richter, Jens Wöhnert, Harald Schwalbe // ChemBioChem. — 2003. — Vol. 4, no. 10. — P. 936–962.

[21] Prediction of RNA secondary structure using generalized centroid estimators / Michiaki Hamada, Hisanori Kiryu, Kengo Sato et al. // Bioinformatics. — 2009. — Vol. 25, no. 4. — P. 465–473.

[22] Quadrini Michela, Merelli Emanuela, Piergallini Riccardo. Loop Grammars to Identify RNA Structural Patterns. // BIOINFORMATICS. — 2019. — P. 302–309.

[23] RNA STRAND: the RNA secondary structure and statistical analysis database / Mirela Andronescu, Vera Bereg, Holger H Hoos, Anne Condon // BMC bioinformatics. — 2008. — Vol. 9, no. 1. — P. 1–10.

[24] RNA Secondary Structure Prediction by MFT Neural Networks / B Apolloni, L Lotorto, A Morpurgo, A Zanaboni. — 2003.

[25] RNA secondary structure and compensatory evolution / Ying Chen, David B Carlini, John F Baines et al. // Genes & genetic systems. — 1999. — Vol. 74, no. 6. — P. 271–286.

[26] RNA secondary structure prediction using an ensemble of two-dimensional deep neural networks and transfer learning / Jaswinder Singh, Jack Hanson, Kuldip Paliwal, Yaoqi Zhou // Nature communications. — 2019. — Vol. 10, no. 1. — P. 1–13.

[27] RNAscClust: clustering RNA sequences using structure conservation and graph based motifs / Milad Miladi, Alexander Junge, Fabrizio Costa et al. // Bioinformatics. — 2017. — Vol. 33, no. 14. — P. 2089–2096.

[28] RNAstructure: web servers for RNA secondary structure prediction and analysis / Stanislav Bellaousov, Jessica S Reuter, Matthew G Seetin, David H Mathews // Nucleic acids research. — 2013. — Vol. 41, no. W1. — P. W471–W474.

[29] Reeder Jens, Steffen Peter, Giegerich Robert. pknotsRG: RNA pseudoknot folding including near-optimal structures and sliding windows // Nucleic acids research. — 2007. — Vol. 35, no. suppl_2. — P. W320–W324.

[30] Rivas Elena, Eddy Sean R. A dynamic programming algorithm for RNA structure prediction including pseudoknots // Journal of molecular biology. — 1999. — Vol. 285, no. 5. — P. 2053–2068.

[31] Rivas Elena, Eddy Sean R. The language of RNA: a formal grammar that includes pseudoknots // Bioinformatics. — 2000. — Vol. 16, no. 4. — P. 334–340.

[32] Ruan Jianhua, Stormo Gary D, Zhang Weixiong. ILM: a web server for predicting RNA secondary structures with pseudoknots // Nucleic acids research. — 2004. — Vol. 32, no. suppl_2. — P. W146–W149.

[33] Sherman Douglas. Humidor: Microbial community classification of the 16s gene by training cigar strings with convolutional neural networks. — 2017.

[34] Sippl Manfred J. Biological sequence analysis. Probabilistic models of proteins and nucleic acids, edited by R. Durbin, S. Eddy, A. Krogh, and G. Mitchinson. 1998. Cambridge: Cambridge University Press. 356 pp.£ 55.00 (80.00)(*hardcover*); $19.95(34.95) // Protein Science. — 1999. — Vol. 8, no. 3. — P. 695–695.

[35] Abadi Martín, Agarwal Ashish, Barham Paul et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. — 2015. — Software available from tensorflow.org. URL: https://www.tensorflow.org/.

[36] Variation in secondary structure of the 16S rRNA molecule in cyanobacteria with implications for phylogenetic analysis / Klára Řeháková, Jeffrey R Johansen, Mary B Bowen et al. // Fottea. — 2014. — Vol. 14, no. 2. — P. 161–178.

[37] Wada Akiyoshi, Suyama Akira. Local stability of DNA and RNA secondary structure and its relation to biological functions // Progress in biophysics and molecular biology. — 1986. — Vol. 47, no. 2. — P. 113–157.

[38] Westhof Eric. Twenty years of RNA crystallography // Rna. — 2015. — Vol. 21, no. 4. — P. 486–487.

[39] YaccConstructor [Электронный ресурс]. — URL: https://github.com/YaccConstructor (online; accessed: 11.05.2021).

[40] Yoon Byung-Jun, Vaidynathan PP. HMM with auxiliary memory: a new tool for modeling RNA structures // Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, 2004. / IEEE. — Vol. 2. — 2004. — P. 1651–1655.

[41] Younger Daniel H. Recognition and parsing of context-free languages in time n3 // Information and control. — 1967. — Vol. 10, no. 2. — P. 189–208.

[42] The conservation and function of RNA secondary structure in plants / Lee E Vandivier, Stephen J Anderson, Shawn W Foley, Brian D Gregory // Annual review of plant biology. — 2016. — Vol. 67. — P. 463–488.