Санкт-Петербургский государственный университет

Программная инженерия
Кафедра системного программирования

Кутленков Дмитрий Александрович

Разработка системы предсказания вторичной структуры РНК с использованием синтаксического анализа и искусственных нейронных сетей

Курсовая работа

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

Санкт-Петербург
2020

Оглавление

Введение

Многие направления современной биоинформатики имеют дело с анализом биологических последовательностей. Наиболее интересным объектом для изучения являются кодирующие последовательности, присутствующие в клетках всех живых организмах — РНК и ДНК.

РНК — макромолекула, выполняющая множество различных функций в живых организмах. РНК состоит из цепи нуклеотидов — базовых органических соединений, с помощью которых в организме кодируется информация. Таким образом, РНК является носителем генетических функций и выполняет работу по переносу и реализации этой информации, например, играет основную роль в процессе синтеза белков — трансляции. Однако ее функции не ограничиваются трансляцией — она играет важную роль во множестве других процессов [1].

Как и другие макромолекулы, РНК практически никогда не находится в развернутом виде — она некоторым образом сворачивается в пространстве. При рассмотрении структур макромолекул выделяют несколько уровней (см. Рис. 2). Для понимания функций РНК зачастую необходимо знать ее структуру [2, 3]. Предсказывать вторичную структуру РНК проще, чем третичную, к тому же она предоставляет информацию, полезную для предсказания третичной структуры [4]. Экспериментальные методы предсказания вторичной структуры (такие, как ядерный магнитный резонанс и рентгеноструктурный анализ) сложны и требуют большого количества ресурсов. Из-за этого предпочитаемыми способами предсказания являются вычислительные методы.

При построении цепи РНК используются 4 нуклеотида. Они являются парными, то есть могут образовывать между собой связи. Вторичная структура РНК описывает водородные связи между нуклеотидами. Она может включать в себя несколько базовых структурных элементов, одним из самых частых является "шпилька" (см. Stem loop на Рис. 2) — элемент, состоящий из нескольких последовательно связанных нуклеотидов, за которыми идут несколько не связанных.

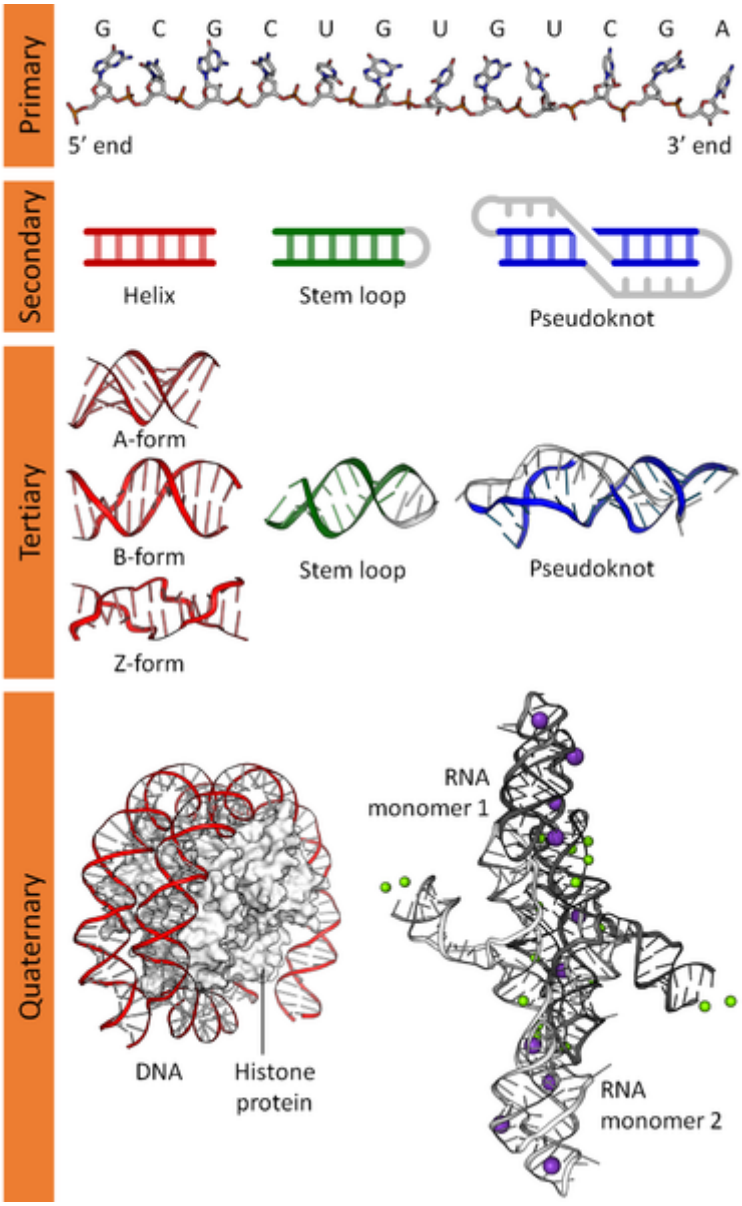


Рис. 1: Структура РНК [5]

Когда мы говорим о вторичной структуре РНК, нужно помнить о том, что в ней возможны псевдоузлы — структурные элементы, в которых новая ”шпилька” начинается в момент, когда предыдущая еще не закончилась (см. Pseudoknot на Рис. 2). Этот элемент накладывает ограничения на методы, которые мы можем использовать при предсказании вторичной структуры, и делает неприменимыми некоторые классические решения. В связи с этим на данный момент лишь относительно небольшое количество инструментов(см. Обзор существующих решений) предсказания вторичной структуры позволяют предсказы-

вать псевдоузлы. Поэтому создание инструмента, умеющего учитывать псевдоузлы, было одним из направлений нашего проекта.

1. Постановка задачи

Целью данной работы является разработка системы, способной с достаточной степенью точности предсказывать вторичную структуру РНК, учитывая при этом псевдоузлы.

Для достижения цели были поставлены следующие задачи:

- Изучить предметную область
- Проанализировать существующие решения
- Спроектировать систему на основе формальных грамматик и нейронных сетей
- Собрать и обработать данные для обучения нейронной сети
- Создать систему для подготовки данных
- Обработать результат нейронной сети для получения биологически возможного результата
- Собрать составные части в единую систему, с которой будет удобно работать целевой аудитории, то есть биологам и биоинформатикам

2. Обзор существующих решений

2.1. Обзор существующих методов

Существующие вычислительные методы можно разделить на две категории — проводящие сравнительный анализ и проводящие анализ одной последовательности. В своем обзоре я не буду касаться первого вида, так как он составляет вторичную структуру основываясь на нескольких гомологичных входных последовательностях, и сосредоточусь на обзоре решений, работающих непосредственно с одной цепочкой.

Одним из популярных подходов для предсказания вторичной структуры является нахождение структуры с минимальной свободной энергией (MFE метод). Такие инструменты используют динамический подход — считают энергию конечной структуры, основываясь на энергиях составных частей. В качестве примера такого подхода можно привести *RNAfold* [6]. Другой подход, увеличивающий точность MFE методов — метод максимальной ожидаемой точности (MEA метод), который выбирает наиболее точную структуру из возможных. Примером такого метода является *CentroidFold* [7].

Однако же, у перечисленных выше методов есть существенный недостаток — они не умеют предсказывать псевдоузлы, которые играют важную роль как в клетках, так и в вирусах [8, 9].

Для предсказания структур с псевдоузлами существует ряд других методов. *HotKnots* [10] использует схожий с MFE метод, добавляя части структуры в попытке минимизировать общую энергию структуры. Это, однако, занимает большое количество времени. Расширением MEA метода является *IPknot* [4] — метод, использующий целочисленное программирование для приближения распределения вероятностей связей между нуклеотидами в последовательности. Еще одним методом, основывающимся на гипотезе о том, что цепочки сначала складываются в структуры без псевдоузлов, а потом проводят дополнительные связи для минимизации энергии [11], является *HFold* [12]. Позднее была

выпущенная улучшенная версия — *Iterative HFold* [13], которая исправила проблемы, связанные с псевдоузлами. Тесты представленных программ, однако, не позволяют говорить об однозначном превосходстве какой-либо из них [14].

Стоит отметить, что структуру РНК без псевдоузлов можно задать с помощью контекстно-свободной грамматики [15]. Также доказано, что можно задать вторичную структуру, используя стохастические контекстно-свободные грамматики [16]. Однако, в общем случае создание и применение такой грамматики слишком сложная задача, так как в природе при определенных условиях существует вероятность создания нетипичных соединений. Поэтому на практике предполагается использовать формальные грамматики для получения базовой структуры, а сложные взаимодействия обрабатывать с применением другого метода.

При работе с реальными данными нужно иметь в виду, что они подвержены большому числу искажений, возникающих как из-за технических причин (ошибки при получении данных с макромолекулы), так и из-за биологических (мутации). Это делает плохо применимыми точные методы. Одним из способов борьбы с этой проблемой является использование методов машинного обучения, например, искусственных нейронных сетей. Недавние успехи в области предсказания структуры белков с помощью нейронных сетей позволяют рассматривать этот метод как потенциально успешный в области предсказания структуры РНК [17]. Кроме того, на данный момент существует несколько проектов, ведущих исследования по предсказанию структуры РНК с помощью методов машинного обучения, например, LSTM сетей [18] и ансамблей сетей [19].

Комбинируя методы синтаксического анализа и машинного обучения, то есть извлекая основные особенности структуры с помощью алгоритмов синтаксического анализа, а затем обрабатывая полученные данные с помощью нейронной сети, мы хотим получить систему, способную достаточно точно предсказывать вторичную структуру РНК с учетом псевдоузлов. Так как работа была выполнена в рамках иссле-

довательского проекта с несколькими участниками, был использован уже готовый парсер, а нейронная сеть разрабатывалась параллельно с данной работой магистранткой кафедры системного программирования Полиной Сергеевной Луниной. Опишем эти составные части подробнее.

2.2. Парсер

Парсер принимает на вход первичную структуру (строку, в которой записана последовательность нуклеотидов) и распознает возможные места для соединений между нуклеотидами. На выходе парсер выдает картинку, в которой белые пиксели стоят в позиции $[i, j]$, если между i-м и j-м нуклеотидом возможна связь. На диагонали в картинке стоят пиксели различного цвета в зависимости от нуклеотида — это необходимо для того, чтобы была возможность восстановить изначальную структуру. Парсер создан с помощью платформы *YaccConstructor* [20].

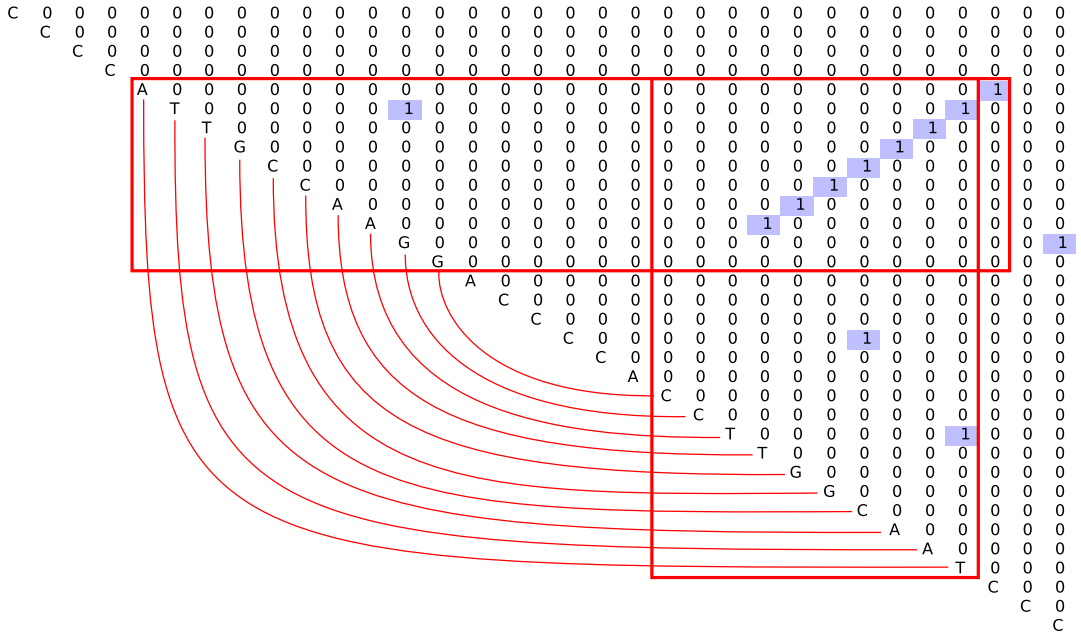


Рис. 2: Представление вторичной структуры, используемое в работе (для последовательности ДНК) [21]

2.3. Нейронная сеть

Нейронная сеть должна уметь очищать результат работы парсера от ненужных связей. Для обучения нейросети мы подаем ей на вход результат работы парсера и эталонную структуру в виде картинки. Нейросеть должна быть способна предсказывать сложные связи внутри структуры и взаимодействия между базовыми структурами. На данный момент используется остаточная нейронная сеть.

Так как итоги работы над нейронными сетями важны для понимания работы в целом, кратко приведем полученные результаты.

Предпринимались попытки использования двух подходов к созданию нейросети с использованием данных, подготовленных в рамках этой курсовой работы. Первый заключался в том, чтобы использовать данные примерно одинаковой длины и дополнять более короткие последовательности до самой длинной, заполняя недостающую часть изображения черными точками. Другой позволял использовать данные разной длины.

2.3.1. Сеть для данных фиксированной длины

При обучении данной сети были использованы 56689 последовательностей длиной 90 нуклеотидов, разделенных на обучающую, валидационную и тестовую выборки в отношении 70%:10%:20%. Наилучший полученный результат имеет F-меру (среднее гармоническое precision и recall) равную 87%.

2.3.2. Сеть для данных переменной длины

Реальные последовательности имеют разную длину, поэтому была использована другая архитектура, позволяющая обучаться на данных переменной длины. Кроме того, такая архитектура позволяет нам использовать больше данных. Были использованы 145054 цепочек длиной от 50 до 90 нуклеотидов, разделенных в той же пропорции. Для этих данных удалось достичь значения F-меры равного 62%.

Полученную сеть дообучили на данных с псевдоузлами. F-мера нейросети поднялась до 75%. Провести тестирование по качеству предсказания исключительно псевдоузлов пока не удалось, но полученные результаты показывают потенциал дальнейших исследований в данном направлении.

Полученные результаты имеют один существенный недостаток — используемые методы обучения и измерения качества нейронной сети плохо отслеживают, между какими из нуклеотидов устанавливаются связи, в то время как на самом деле в подавляющем большинстве случаев связи образуются между парными нуклеотидами. Этот факт породил необходимость использовать алгоритм выравнивания, о котором будет рассказано далее.

3. Архитектура процесса обучения нейронной сети

Для выполнения поставленной задачи была разработана следующая архитектура (Рис. 3). Рассмотрим общий процесс работы системы.

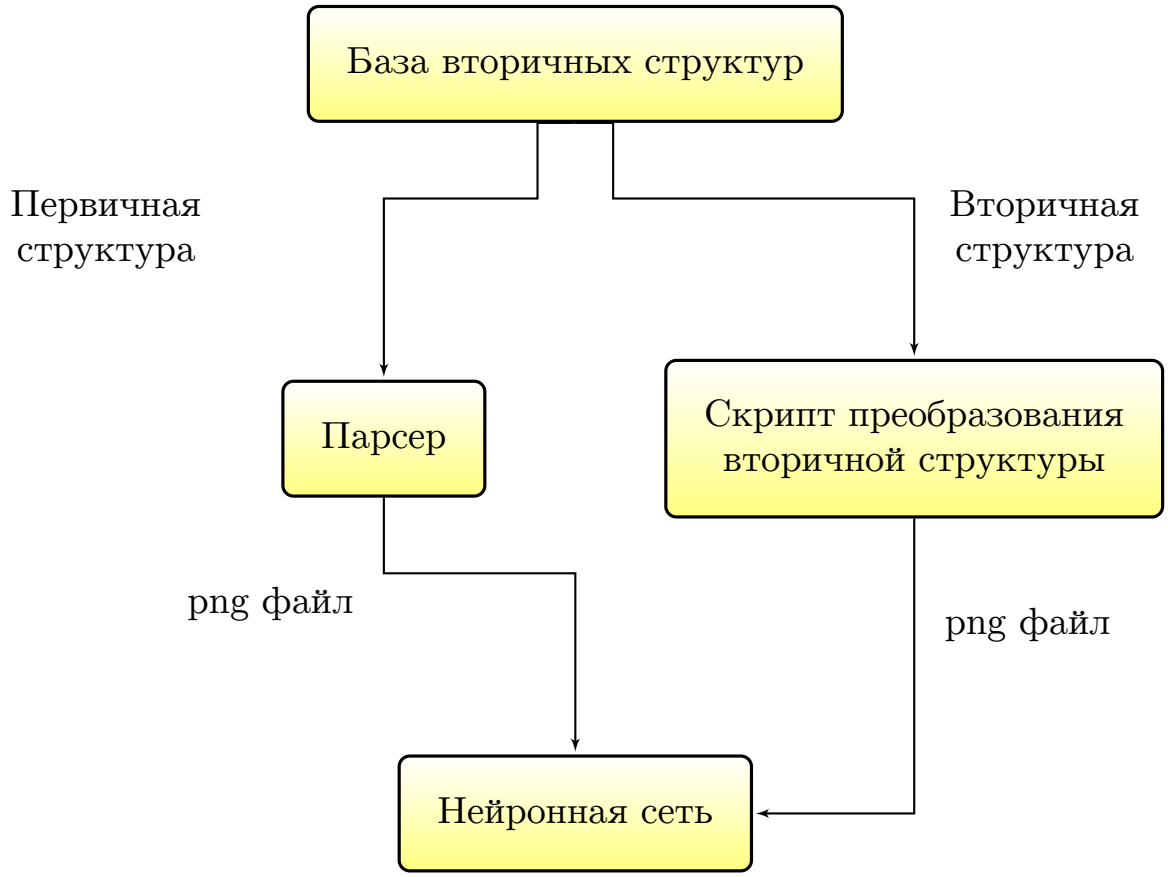


Рис. 3: Архитектура процесса обучения нейронной сети

В процессе обучения нейросети из базы берутся пары из первичной (строка, в которой записана последовательность нуклеотидов) и вторичной (строка в формате dot-bracket — в ней непарные нуклеотиды отмечены точками, а парные отмечаются парными скобками) структур последовательности. Первичная структура отправляется на вход парсера, а вторичная на вход скрипта, который превращает ее в изображение, где белые пиксели стоят в позиции $[i, j]$, если между i-м и j-м нуклеотидом есть связь. Выход парсера и скрипта мы передаем на вход нейронной сети. Таким образом, сеть учится убирать ненужные связи и добавлять недостающие в результат работы парсера.

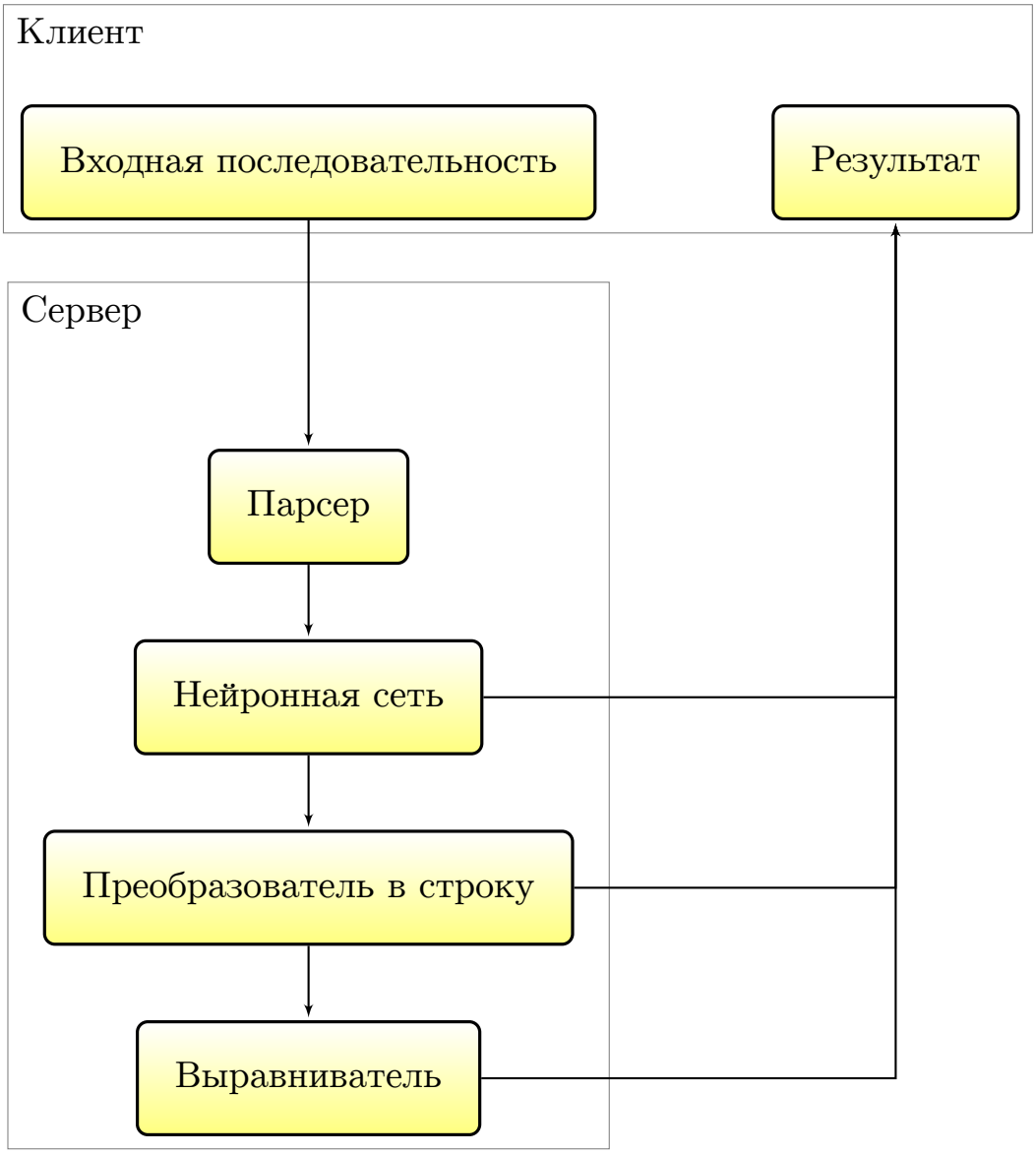
Используемые парсер и нейросеть были описаны ранее. Рассмотрим оставшийся компонент — данные.

3.1. Данные

В качестве данных для обучения нейросети мы используем пары из последовательности нуклеотидов и ее вторичной структуры. Наилучшим вариантом является использование данных, полученных биологическими методами. Баз с такими последовательностями существует две — *RNA STRAND* [22] и *Pseudobase++* [23]. К сожалению, в этих базах довольно мало данных. Для обучения нейросети требуется много данных примерно одинаковой длины. В результате было решено использовать последовательности из базы РНК *RNAcentral* [24], а затем получать их вторичные структуры с помощью сторонних инструментов. После получения хорошего результата для последовательностей без псевдоузлов можно использовать дообучение нейронной сети на данных с псевдоузлами.

4. Архитектура конечной системы

Итоговая система была воплощена в виде веб-сервиса. Серверная часть написана на языке *Python 3*, с использованием библиотеки *flask*[1]. Общение между клиентом и сервером происходит с помощью *REST API*. Клиентская часть написана с помощью фреймворков *Vue.js*[2] и *Bulma.io*[3]. Вышеперечисленные технологии были выбраны, так как они позволяют в сжатые сроки создать прототип, который затем будет удобно расширять.



[1]Фреймворк для создания веб-приложений на языке Python: `https://palletsprojects.com/p/flask/`[Accessed: 15th April, 2020].

[2]JavaScript-фреймворк для создания пользовательских интерфейсов: `https://vuejs.org/`[Accessed: 15th April, 2020].

[3]CSS-фреймворк: `https://bulma.io/`[Accessed: 15th April, 2020].

Парсер и нейронная сеть были рассмотрены ранее. Рассмотрим подробнее алгоритм выравнивания.

4.1. Алгоритм выравнивания

Нейронная сеть плохо способна отслеживать, между какими из нуклеотидов она предсказывает связи. Так как в основном в природе связи образуются между парными нуклеотидами, необходимо обработать результат. Для этого был разработан алгоритм, в основе которого лежит следующая идея — если в предсказанной нейросетью последовательности существует какая-то петля, то где-то рядом с этим местом энергетически выгодно образовать петлю схожего размера. Алгоритм состоит из следующих шагов:

1. Найти следующую петлю
2. Расширить границы, в которых мы будем искать выравнивание. Важно не пересекать петли между собой.
3. На полученных интервалах провести локальное выравнивание
4. Запомнить границы найденного результата
5. Вернуться в п. 1

Заключение

В рамках курсовой работы были выполнены следующие задачи:

- Изучена предметная область
- Проведен анализ уже существующих решений
- Разработана архитектура системы
- Собраны, проанализированы и обработаны данные из нескольких источников - *RNA STRAND*, *Pseudobase++*, *RNACentral*
- Создана система подготовки данных
- Разработан алгоритм перевода полученных последовательностей в биологически возможные
- Разработана система предсказания вторичной структуры РНК последовательностей
- Создано клиент-серверное приложение, предоставляющее доступ к системе

Исходный код доступен по ссылкам `https://github.com/SacredArrow/Secondary_structure_public` и `https://github.com/SacredArrow/Course_work_web`.

Разработанная система задумывалась максимально простой для пользователя и поэтому может быть использована биологами и биоинформатиками при проведении исследований.

Данная система может быть улучшена путем улучшения составных систем, например, нейронной сети. Улучшение может состоять как в получении дополнительных данных, так и в дальнейших экспериментах с архитектурой. Также для системы удалось лишь частично поддержать псевдоузлы, так как они требуют более сложной постобработки.

Список литературы

[1] Lodish H, Berk A, Zipursky SL, *et al.*, *The Three Roles of RNA in Protein Synthesis.*, ch. 4.4. New York: W. H. Freeman, 2000.

[2] A. A. Saraiya, T. N. Lamichhane, C. S. Chow, J. SantaLucia, and P. R. Cunningham, "Identification and role of functionally important motifs in the 970 loop of escherichia coli 16s ribosomal rna," *Journal of Molecular Biology*, vol. 376, no. 3, pp. 645 – 657, 2008.

[3] K. Lee, S. Varma, J. SantaLucia, and P. R. Cunningham, "In vivo determination of rna structure-function relationships: analysis of the 790 loop in ribosomal rna11edited by d. e. draper," *Journal of Molecular Biology*, vol. 269, no. 5, pp. 732 – 743, 1997.

[4] K. Sato, Y. Kato, M. Hamada, T. Akutsu, and K. Asai, "IPknot: fast and accurate prediction of RNA secondary structures with pseudoknots using integer programming," *Bioinformatics*, vol. 27, pp. i85–i93, 06 2011.

[5] Thomas Shafee, "Summary of nucleic acid structure (primary, secondary, tertiary, and quaternary) using DNA helices and examples from the VS ribozyme and telomerase and nucleosome. (PDB: ADNA, 1BNA, 4OCB, 4R4V, 1YMO, 1EQZ)." `https://en.wikipedia.org/wiki/File:DNA_RNA_structure_(full).png`, 2017. [Online; accessed November 25, 2019].

[6] I. L. Hofacker, W. Fontana, P. F. Stadler, L. S. Bonhoeffer, M. Tacker, and P. Schuster, "Fast folding and comparison of rna secondary structures," *Monatshefte für Chemie / Chemical Monthly*, vol. 125, pp. 167–188, Feb 1994.

[7] M. Hamada, H. Kiryu, K. Sato, T. Mituyama, and K. Asai, "Prediction of RNA secondary structure using generalized centroid estimators," *Bioinformatics*, vol. 25, pp. 465–473, 12 2008.

[8] D. W. Staple and S. E. Butcher, "Pseudoknots: Rna structures with diverse functions," *PLOS Biology*, vol. 3, 06 2005.

[9] B. A. Deiman and C. W. Pleij, "Pseudoknots: A vital feature in viral rna," *Seminars in Virology*, vol. 8, no. 3, pp. 166 – 175, 1997.

[10] J. Ren, B. Rastegari, A. Condon, and H. Hoos, "Hotknots: heuristic prediction of rna secondary structures including pseudoknots. rna 11:1494-1504, <http://dx.doi.org/10.1261/rna.7284905," *RNA (New York, N.Y.)*, vol. 11, pp. 1494–504, 11 2005.

[11] I. Tinoco and C. Bustamante, "How rna folds," *Journal of Molecular Biology*, vol. 293, no. 2, pp. 271 – 281, 1999.

[12] H. Jabbari, A. Condon, and S. Zhao, "Novel and efficient rna secondary structure prediction using hierarchical folding," *Journal of computational biology : a journal of computational molecular cell biology*, vol. 15, pp. 139–63, 04 2008.

[13] H. Jabbari and A. Condon, "A fast and robust iterative algorithm for prediction of rna pseudoknotted secondary structures," *BMC bioinformatics*, vol. 15, p. 147, 05 2014.

[14] H. Jabbari, I. Wark, and C. Montemagno, "Rna secondary structure prediction with pseudoknots: Contribution of algorithm versus energy model," *PLOS ONE*, vol. 13, pp. 1–21, 04 2018.

[15] D. B. Searls, "The linguistics of dna," *American Scientist*, vol. 80, no. 6, pp. 579–591, 1992.

[16] B. Knudsen and J. Hein, "Rna secondary structure prediction using stochastic context-free grammars and evolutionary history," *Bioinformatics*, vol. 15 6, pp. 446–54, 1999.

[17] S. Wang, S. Sun, Z. Li, R. Zhang, and J. Xu, "Accurate de novo prediction of protein contact map by ultra-deep learning model," *bioRxiv*, 2016.

[18] W. Lu, Y. Tang, H. Wu, H. Huang, Q. Fu, J. Qiu, and H. Li, "Predicting rna secondary structure via adaptive deep recurrent neural networks with energy-based filter," *BMC Bioinformatics*, vol. 20, 12 2019.

[19] J. Singh, J. Hanson, K. Paliwal, and Y. Zhou, "Rna secondary structure prediction using an ensemble of two-dimensional deep neural networks and transfer learning," *Nature Communications*, vol. 10, 12 2019.

[20] JetBrains Research, "YaccConstructor." `https://github.com/YaccConstructor/YaccConstructor`. [Online; accessed 22-November-2019].

[21] P. Lunina and S. Grigorev, "On secondary structure analysis by using formal grammars and artificial neural networks," in *Proceedings of CIBB 2019*, 2019.

[22] M. Andronescu, V. Bereg, H. Hoos, and A. Condon, "Rna strand: the rna secondary structure and statistical analysis database," *BMC bioinformatics*, vol. 9, p. 340, 09 2008.

[23] M. Taufer, A. Licon, R. Araiza, D. Mireles, F. Batenburg, A. Gultyaev, and M.-Y. Leung, "Pseudobase++: An extension of pseudobase for easy searching, formatting and visualization of pseudoknots," *Nucleic acids research*, vol. 37, pp. D127–35, 12 2008.

[24] The RNAcentral Consortium, "RNAcentral: a hub of information for non-coding RNA sequences," *Nucleic Acids Research*, vol. 47, pp. D221–D229, 11 2018.