Предсказание вторичной структуры РНК*

Самохина А. М.
alina.samokhina@phystech.edu
MIPT

Вторичная структура РНК используется для определения функций РНК, построения третичной структуры и других задач. Непосредственное наблюдение вторичной структуры затруднительно, в связи с чем существует задача построения вторичной структуры РНК по первичной. Данная задача биоинформатики является нерешённой. Методы машинного обучения начали применяться к ней только в последние несколько лет и в основном используют рекуррентные нейросети. В данной работе предлагается рассмотреть возможность использования сверточных нейросетей для предсказания вторичной структуры РНК.

1 Введение

Рибонуклеиновая кислота (РНК) — одна из основных макромолекул, содержащихся в клетках живых организмов. РНК участвует в кодировании, чтении и регуляции генов. РНК состоит из длинной цепи нуклеотидов, последовательность которых позволяет РНК кодировать генетическую информацию. В связи с тем, что РНК - одиночная цепочка нукеотидов, для неё характерны разнообразные пространственные структуры, в которых часть нуклеотидов одной и той же цепи спарены между собой. Именно пространственная структура определяет функциональные свойства РНК. Имеются примеры молекул, изменивших с течением времени нуклеотидную последовательность, но не изменивших вторичную структуру. Первичная структура РНК – её нуклеотидная последовательность. Для многих РНК она была получена экспериментально. Однако определение вторичной и третичной структур на практике связано с большими трудозатратами и дорогостоящими процедурами (рентгеноструктурный анализ, ЯМР, спектроскопия).

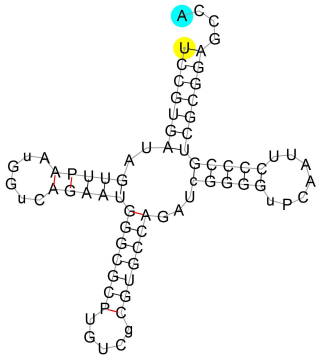


Рис. 1 tRNA of yeast

Тем временем, вторичная структура РНК важна для определения функции молекулы и для построения её третичной структуры. К примеру, транспортная РНК имеет форму "клеверного листа"(рис.1). Однако непосредственное наблюдение вторичной структуры РНК является достаточно затратным процессом. В связи с этим предсказание вторичной структуры РНК по первичной является актуальной проблемой. На протяжении долгого времени исследователи используют различные методы решения этой задачи: от статистических[1] до генетических[2][3] и нейросетевых[4]. Задача усложняется наличием во вторичной структуре различных классов псевдоузлов, которые являются топологически сложными структурами (рис.2.).

Большинство классических методов не могут предсказывать вторичную структуру РНК с учетом этих псевдоузлов. Классическими назовем алгоритмы, решающие данную задачу без применения методов машинного обучения, так как они развиваются уже на

* Преподаватель: Торшин И. Ю.

протяжении нескольких десятилетий. Из методов, предложенных для решения проблемы предсказания псевдоузлов, немногие достигли удовлетворительных результатов как в сложности, так и в точности [3]. Возможность предсказания структур с псевдоузлами — основное преимущество методов глубокого обучения перед классическими. Для стандартных алгоритмов было показано, что учёт данных "долгих"связей является NP-полной задачей[5]. Алгоритмы машинного обучения позволяют улавливать подобные зависимости. И в связи с тем, что данных для обучения алгоритмов машинного обучения стало достаточно (используемая в работе bpRNA BD имеет более чем 100 000 примеров РНК последовательностей), ученые обратились к исследованию возможностей предсказания вторичной структуры РНК с помощью машинного обучения без обращения к классическим подходам. Многие работы всё ещё опираются на результаты, полученные без использования нейросетей и объединяют классические подходы с машинным обучением [6] [5],[7]. Однако, к примеру, в работе [4] рассматривается исключительно нейросетевой подход с применением transfer learning для предсказания РНК из базы данных PDB db. Модель обучается на большом корпусе bpRNA и затем более точно настраивается на интересующий класс РНК.

2 Мотивация выбранного метода

Большинство работ по предсказанию вторичной структуры РНК с использованием машинного обучения применяют рекуррентные нейросети, так как задача легко представляется как задача машинного перевода. Базовый метод, использованный в данной работе, был вдохновлен статьёй [8]. В ней рассматриваются сверточные сети и их объединение с динамическим программированием для корректирования результатов работы нейросети. Новизна данной статьи - в представлении первичной структуры не как последовательности, а как матрицы вероятностей парности нуклеотидов. Предсказанные вероятности связей (вероятности символов dot-bracket нотации) корректируются алгоритмом, максимизирующим сумму вероятностей в ограничениях на правильную скобочную последовательность и базовые пары (G-C, A-U, G-U).

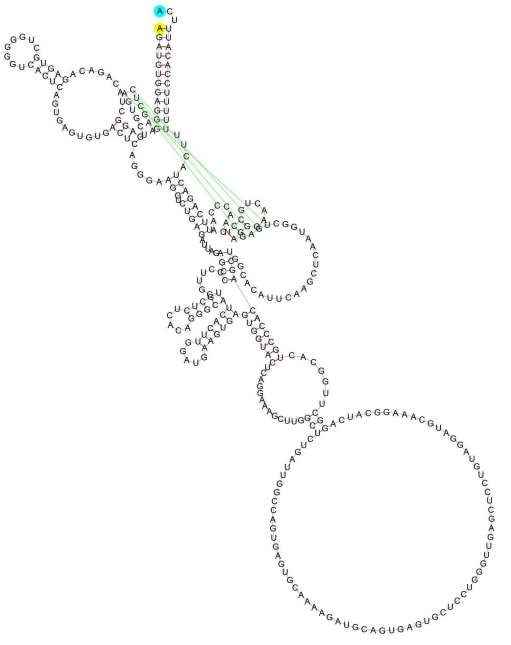


Рис. 2 Ribonuclease P RNA of Xenopus laevis

В данной работе рассматриваются результаты сверточной нейросети без корректирующего модуля на исходных последовательностях. Данный метод является релевантным, так как свёртки с различными ядрами позволяют учесть пространственную информацию для построения взаимосвязей как между близкими, так и между удаленными друг от друга нуклеотидами. Однако основной трудностью использования сверточных нейросетей является различная длина входных последовательностей. Данная проблема решена с помощью использования сверток, не изменяющих длину входной последовательности. Таким образом, длина входной последовательности не имеет значения, а длина выходной строки равна длине входной.

3 Формальная постановка задачи

В данной задаче выборка задается как набор пар последовательностей.

$$\mathbf{D} = \{(x_i, y_i)\}_{i=1,\dots,N}; \tag{1}$$

где N — количество имеющихся пар последовательностей, $\mathbf{x}_i \in \mathbf{X}$, $\mathbf{y}_i \in \mathbf{Y}$.

$\mathbf{X}$ — множество первичных структур молекул РНК, а $\mathbf{Y}$ — множество вторичных.

Исходные данные представлены в виде массивов последовательностей

$x_i = \{x_{i_j}\}_{j=1,\dots,m_i}$ $y_i = \{y_{i_j}\}_{j=1,\dots,m_i}$, где m_i - длина последовательностей под номером i.

Данные в виде таких последовательностей были взяты из файлов базы данных dbRNA в точечно-скобочной нотации (.dbn). В данном случае последовательности $\mathbf{x}_i \in \mathbf{X}$ задаются алфавитом $\{A, U, G, C\}$ (аденин, урацил, гуанин и цитозин). $\mathbf{y}_i \in \mathbf{Y}$, в свою очередь задаются трёх-буквенным алфавитом $\{\cdot\ (\ )\ \}$ и являются правильными скобочными последовательностями.

Для представления данных последовательностей в векторном формате в рамках подготовки данных создается словарь, сопоставляющий каждой букве алфавита свой индекс. Таким образом можем получить $\hat{\mathbf{x}}_i \in \hat{\mathbf{X}}$, а $\hat{}_i \in \hat{\mathbf{Y}}$, которые задются множествами из индексов словаря.

$$\hat{\mathbf{X}} \in \{ix_i\}_{i=1,\dots,V_x}; \tag{2}$$

$$\hat{\mathbf{Y}} \in \{iy_i\}_{i=1,\dots,V_y}; \tag{3}$$

Где V_y, V_x — размерности словарей, ix, iy — элементы словарей первичной и вторичной структур соответственно.

Общая постановка задачи: требуется построить алгоритм $a_{\mathbf{w}} : X \to Y$, где $\mathbf{w}$ - параметры модели.

$$\mathbf{w} = \arg\min_{\mathbf{w}} L(p, class) \tag{4}$$

Где $p = a(\mathbf{w}, x)$, $class = y$, а в качестве функции потерь выбрана кросс энтропия:

$$L(p, class) = -log\frac{exp(p_{class})}{\sum_j exp(p_j))} \tag{5}$$

4 Эксперимент

В качестве базовой модели для решения данной задачи были рассмотрены две свёрточные нейросети, отличающиеся порядком свёрток.

CNN:

- Embedding(35, 100)
- Conv1d(100, 200, kernel_size=(3,), stride=(1,), padding=(1,))
- ReLU()
- Dropout(0.2)
- Conv1d(200, 100, kernel_size=(7,), stride=(1,), padding=(3,))
- ReLU()
- Dropout(0.2)
- Conv1d(100, 3, kernel_size=(11,), stride=(1,), padding=(5,)))

Были использованы три свёрточных слоя с размерами фильтров 3, 7 и 11 соответственно. Остальные параметры свёрток были выбраны так, чтобы сеть работала на последовательностях любой длины и получала на выходе последовательность той же длины,

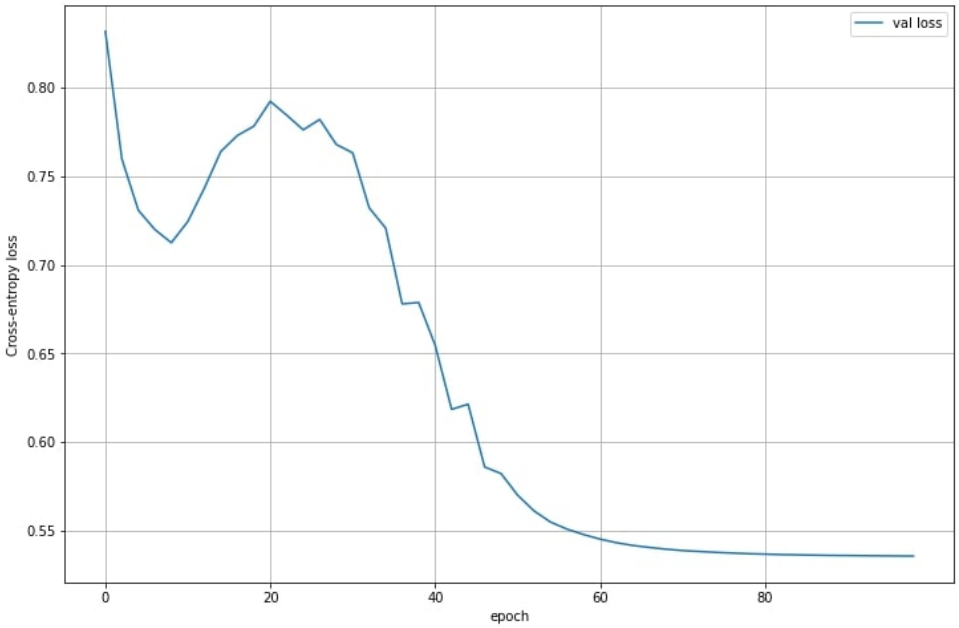


Рис. 3 Функция потерь при обучении нейросети CNN

что и на вход. Между свёрточными слоями добавлены нелинейные функции активации и регуляризация dropout.

CNN_inv:

- Embedding(35, 100)
- Conv1d(100, 200, kernel_size=(11,), stride=(1,), padding=(5,))
- ReLU()
- Dropout(0.2)
- Conv1d(200, 100, kernel_size=(7,), stride=(1,), padding=(3,))
- ReLU()
- Dropout(0.2)
- Conv1d(100, 9, kernel_size=(3,), stride=(1,), padding=(1,)))

Были использованы три свёрточных слоя с размерами фильтров 11, 7 и 3 соответственно. В остальном архитектура аналогична CNN.

model	train loss	val loss
CNN	0.571	0.536
CNN_inv	0.476	0.441

Таблица 1 Сравнение качества моделей

Модель CNN_inv показывает более плавное и более успешное обучение. Следовательно, можно сделать вывод о том, что для данной задачи, свёртки следует располагать в порядке уменьшения размера ядра фильтра, для того, чтобы с большей вероятностью уловить дальёкие пространственные зависимости.

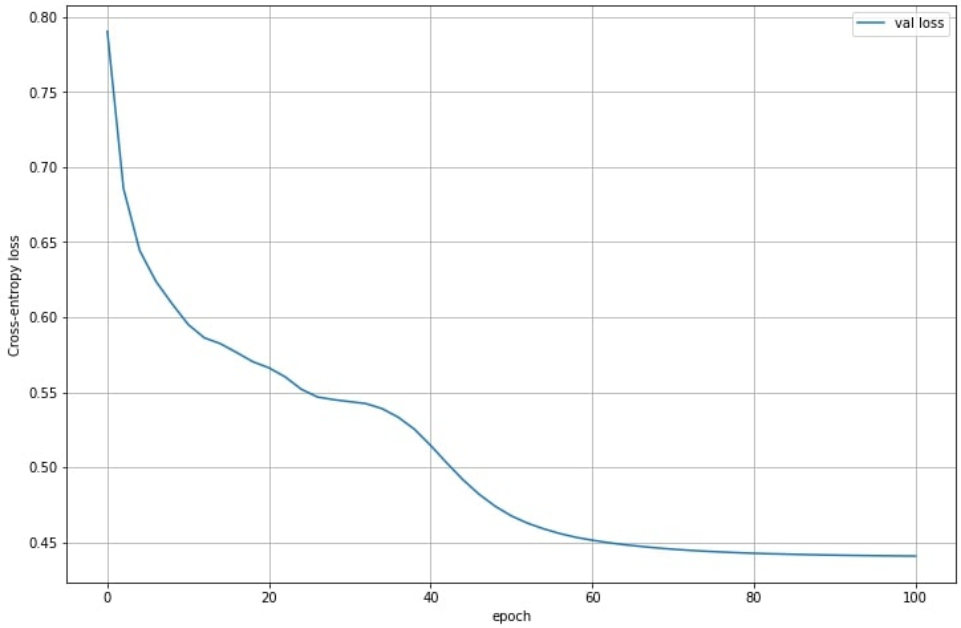
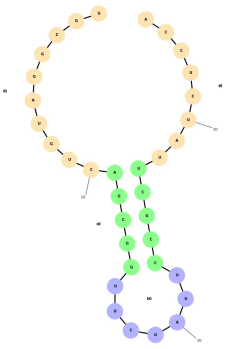
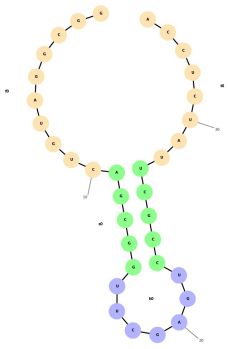


Рис. 4 Функция потерь при обучении нейросети CNN_inv

5 Результаты

Для простых небольших последовательностей модель показывает высокие результаты. Чем длиннее последовательность РНК, тем больше отклонений между предсказанием и реальной последовательностью можно увидеть. Ниже приведены визуализации результатов работы модели на некоторых последовательностях.





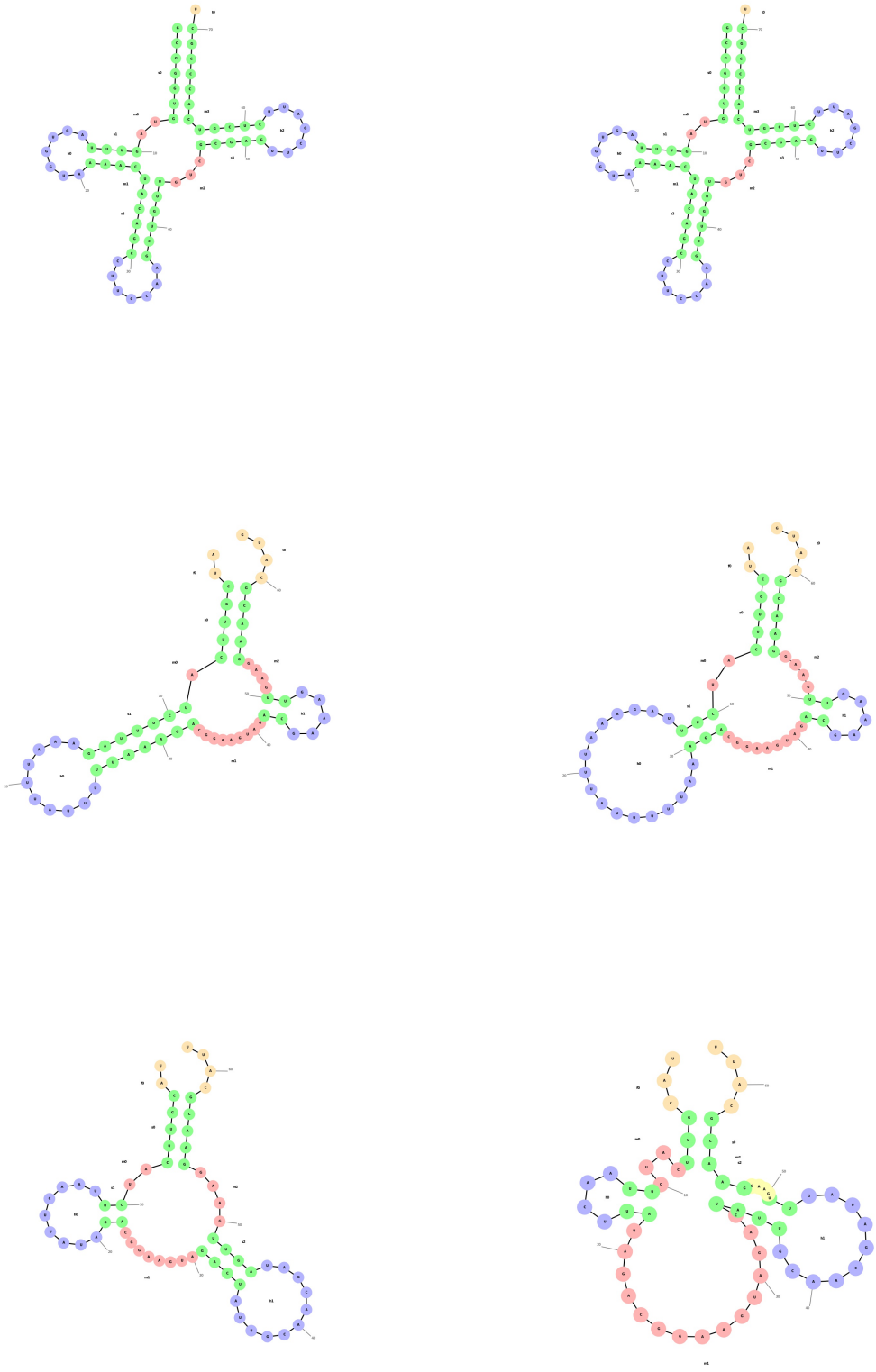


Рис. 5 Истинная и предсказанная вторичные структуры РНК

6 Выводы и дальнейшая работа

В рамках данной работы:

1. удалось подтвердить предположение о возможной эффективности сверточных сетей для предсказания вторичной структуры РНК
2. было установлено, что нисходящий порядок сверток по размеру их ядра является предпочтительным

Для дальнейшей работы предлагается:

1. добавить стандартные для данной задачи метрики качества (positive predictive value(precision), sensitivity(recall), F_1 и specificity)
2. исследовать влияние размера сверток на качество модели
3. оценить зависимость качества предсказаний вторичной структуры от длины последовательности и размеров фильтров нейросети
4. дополнить модель корректирующим модулем для исправления неправильных скобочных последовательностей

7 Список литературы

[1] Ye Ding and Charles E Lawrence. A bayesian statistical algorithm for RNA secondary structure prediction. *Computers & Chemistry*, 23(3-4):387–400, June 1999.

[2] Sha Shi, Xin-Li Zhang, Xian-Li Zhao, Le Yang, Wei Du, and Yun-Jiang Wang. Prediction of the RNA secondary structure using a multi-population assisted quantum genetic algorithm. *Human Heredity*, 84(1):1–8, 2019.

[3] Abdelhakim El Fatmi, M. Ali Bekri, and Said Benhlima. RNAknot: A new algorithm for RNA secondary structure prediction based on genetic algorithm and GRASP method. *Journal of Bioinformatics and Computational Biology*, 17(05):1950031, October 2019.

[4] Jaswinder Singh, Jack Hanson, Kuldip Paliwal, and Yaoqi Zhou. RNA secondary structure prediction using an ensemble of two-dimensional deep neural networks and transfer learning. *Nature Communications*, 10(1), November 2019.

[5] Linyu Wang, Yuanning Liu, Xiaodan Zhong, Haiming Liu, Chao Lu, Cong Li, and Hao Zhang. DMfold: A novel method to predict RNA secondary structure with pseudoknots based on deep learning and improved base pair maximization principle. *Frontiers in Genetics*, 10, March 2019.

[6] Devin Willmott, David Murrugarra, and Qiang Ye. Improving RNA secondary structure prediction via state inference with deep recurrent neural networks. *Computational and Mathematical Biophysics*, 8(1):36–50, March 2020.

[7] Weizhong Lu, Ye Tang, Hongjie Wu, Hongmei Huang, Qiming Fu, Jing Qiu, and Haiou Li. Predicting RNA secondary structure via adaptive deep recurrent neural networks with energy-based filter. *BMC Bioinformatics*, 20(S25), December 2019.

[8] Hao Zhang, Chunhe Zhang, Zhi Li, Cong Li, Xu Wei, Borui Zhang, and Yuanning Liu. A new method of RNA secondary structure prediction based on convolutional neural network and dynamic programming. *Frontiers in Genetics*, 10, May 2019.