1 Литература

1.1 Устройство генома

С появлением первых секвенированных последовательностей генов и геномов появились попытки описать эти последовательности с помощью математических моделей.

Одни из первых опирались на вирусные геномы – стационарная [1], нестационарная марковская модели [2], скрытые марковский модели [3].

Анализ устройства генома был начат с появлением первых секвенированных последовательностей генов. Анализ частот встречаемости моно-...гексануклеотидов был проведен в 1987 году по 80kbp кодирующих и некодирующих последовательностей генома E.coli [4].

В геноме найдено множество ассоциаций различных черт последовательностей [5].

Сейчас известно, что геномы про- и эукариот неравновесны по вхождениям ди-, три-, тетра-, нуклеотидных последовательностей.

1.2 Сетки

Искусственные нейронные сети показали себя как мощный инструмент машинного обучения.

Нейросети широко применяются во многих задачах. Нас интересуют такие, где идет непосредственная обработка нуклеотидных последовательностей.

Сверточные нейросети применяются для классификации последовательностей, распознавания в них каких либо мотивов.

Классификация последовательностей, в частности предсказание сайтов A $\rightarrow$ I редактирования РНК [6].

Сверточная нейронная сетка для предсказания сайтов сплайсинга кольцевых РНК [7].

Предсказание сигнала поли-А [8] обрабатывают последовательности CNN и lstm.

Распознавагие центромерных последовательностей эукариот [9] rnn bdrnn .

2 Методы

2.1 Методы кодировки последовательности

Все нуклеотидные последовательности были закодированны в one-hot-encoded векторы (единичные нуклеотиды) и матрицы (последовательности), где каждому нуклеотиду соответствует один из четырехмерных векторов (0, 0, 0, 1), (0, 0, 1, 0), (0, 1, 0, 0), (1, 0, 0, 0). Это позволяет добиться независимого влияния нуклеотидов на предсказание и используется в категориальных предсказаниях.

2.2 Использованные функции

В качестве функции активации использовавалась функция softmax и relu (уравнение 1).

$$softmax(x) = exp(x - \max_{axis}(x)) \tag{1}$$

$$relu(x) = max(x, 0) \tag{2}$$

Функция потерь во всех архитектурах – категориалная кроссэнтропия (уравнение 3).

$$categorical\_crossentropy(y_{pred}, y_{true}) = -\sum_{x} y_{pred}(x) \log(y_{true}(x)) \tag{3}$$

Для статистического сравнения выборок использовался критерий Манна-Уитни.

2.3 Простейшие нейросетевые модели

2.3.1 Построение выборки контекстов

Последовательности выбирались из генома Escherichia coli (Escherichia coli str. K-12 substr. MG1655, сборка GCF 000005845.2).

Выборка представляет собой набор контекстов (предикторных областей) определенной длины и соответственных нуклеотидов для предсказания.

Выборка состояла из нескольких частей – для обучения алгоритма (train), валидации в процессе обучения (validate), тестирования (test). При этом эти части были взяты из непересекающихся геномных областей, чтобы предотвратить выучивание алгоритмом последовательности нуклеотидов.

Для статистической проверки каждого метода было построено 30 выборок. Во всех 30 выборках области, соответствующие тренировочной и валидационной части, были разные.

Размер тренировочной выборки обычно составлял 100,000 нуклеотидов, валидационной и тренирочной – по одной десятой, соответственно 10,000 и 10,000.

Предикторные области (контексты) для каждого нуклеотида находились с 5' конца нуклеотида, их размер варьировал – 3, 6, 12, 24 нуклеотида. Также создавались выборки со предикторной областью, сдвинутой относительно предсказываемого нуклеотида в 5' сторону на 1, 2, 3, 6, 12, 50 нуклеотидов.

Все предикторные обасти и предсказываемые нуклеотиды были закодированы в виде one-hot-encoded векторов и матриц.

2.3.2 Архитектура нейронных сетей

Все нейронные сети были реализованы с помощью библиотек Keras [10] , TensorFlow. Для работы использовалось несколько архитектур и типов слоев.

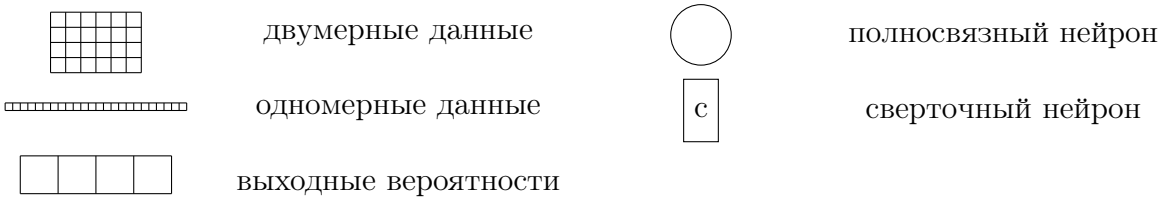
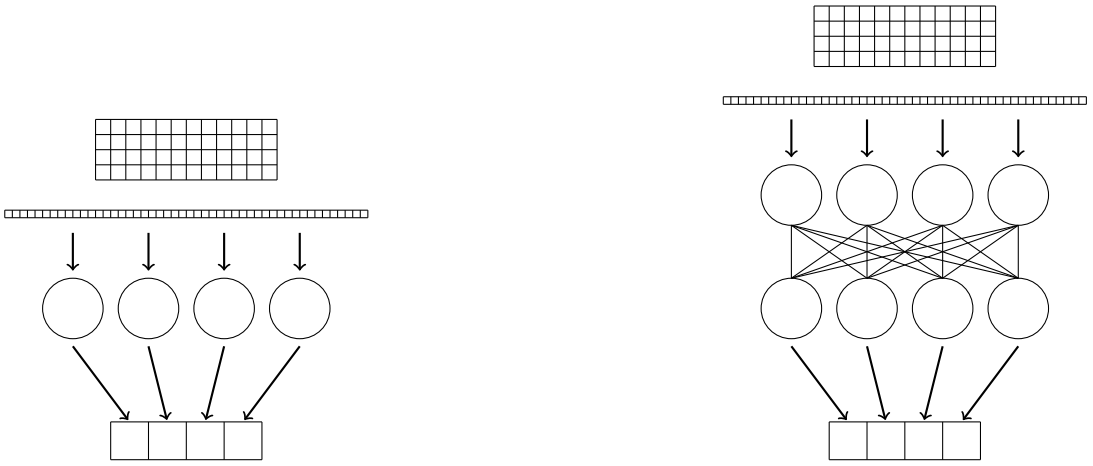


Рис. 1: Условные обозначения на схемах архитертур нейронныых сетей.



(a) **DNN model 1**
Простейшая полносвязная модель состоит из слоя, выпрямляющего начальные данные, четырех полносвязных нейронов с функцией активации softmax.

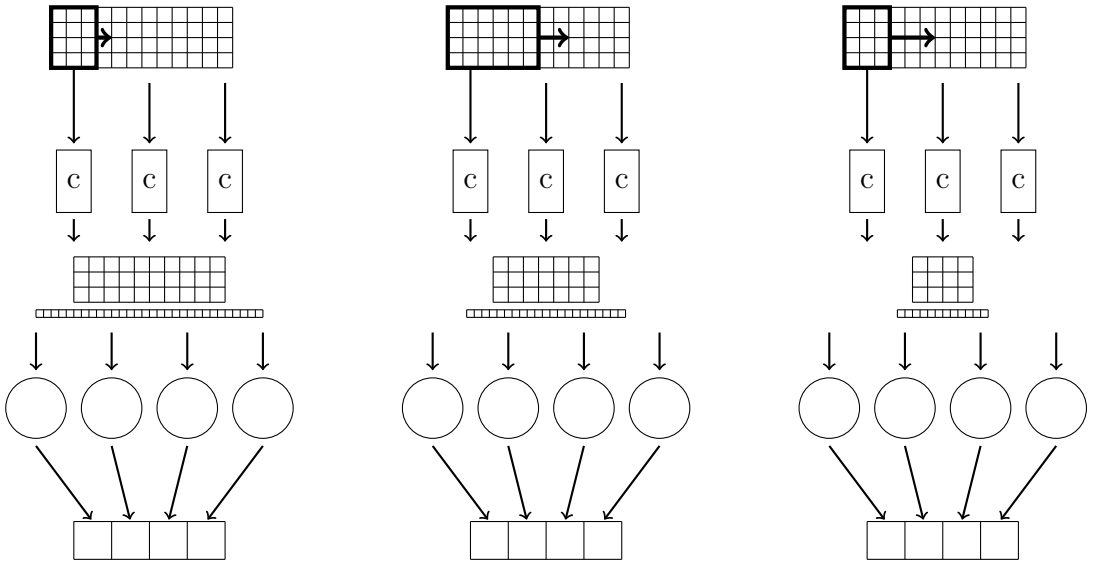
(b) **DNN model 2**
Более сложная полносвязная модель включается в себя два слоя нейронов.

Рис. 2: **Архитектура полносвязных моделей.**

Полносвязные модели (DNN model 1 - рисунок 2a). Первая простейшая полносвязная модель состоит из входного слоя, принимающего нуклеотидный контекст, слоя, уплощающего данные в один вектор, четырех полносвязных нейронов с функцией активации softmax, выходом которых являются вероятности для четырех выходных букв. Число параметров модели $16n + 4$, где n – размер контекста.

(DNN model 2 – рисунок 2b). В полносвязную модель добавлен второй слой нейронов. Число параметров модели $16n + 24$, где n – размер контекста.

Сверточные модели (CNN model 1 - рисунок 3). Простейшая сверточная модель состоит из слоя сверточных нейронов (с функцией активации relu), который работает непосредственно с матрицей контекста, далее выход свертки уплощается в вектор и подается в слой из 4 решающих выходных полносвязных нейронов (с функцией активации softmax).



(a) **CNN model 1 (kernel=3)*3**
Минимальный размер фильтра 3×4 был выбран исходя из кодонной структуры бóльшей части генома – кодирующей.

(b) **CNN model 1 (kernel=6)*3**
Размер фильтра в сверточном слое может быть увеличен до 6×4, вследствие чего уменьшается размерность выхода и число параметров модели.

(c) **CNN model 1 (kernel=3)*3 stride=3**
Шаг сверточного фильтра может быть увеличен до 3, что может имитировать распознавание кодонной тринуклеотидной структуры кодирующих частей генома.

Рис. 3: **Архитектура некоторых сверточных моделей.**
Простейшая сверточная модель состоит из трех сверточных нейронов, выход которых представляет из себя матрицу высотой 3, выпрямляющего слоя, четырех полносвязных нейронов с функцией активации softmax.

Конфигурация сверточного слоя может быть различной. Мы исследовали комбинации из разного числа нейронов, с разным размером фильтра (kernel), которые обрабатывают контекст с различным шагом (stride, по умолчанию шаг сверточного фильтра равен 0).

Были использованы следующие конфигурации сверточного слоя:

1. (kernel = 3)*3, три сверточных фильтра размером 3
2. (kernel = 6)*3, три сверточных фильтра размером 6
3. (kernel = 3)*3 stride = 3, три сверточных фильтра размером 3, которые обрабатывают матрицу с шагом 3.

Рекуррентыне модели (RNN model 1 - рисунок 4) Рекуррентная модель состояла из одного LSTM (long short-term memory) слоя, который содержал разное число скрытых состояний, и выходного слоя полносвязных нейронов с функцией активации softmax.

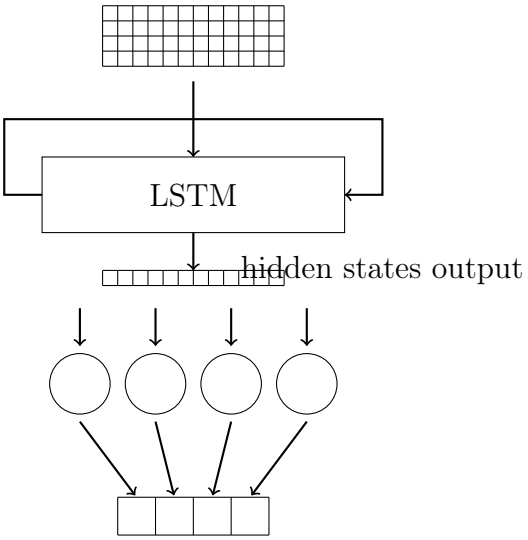


Рис. 4: **Архитектура рекуррентной модели.**

2.3.3 Процесс обучения

Все архитектуры компилировались с использованием оптимизатора Adam [11] с параметрами по умолчанию (learning rate = 0.01). Во время обучения контролировалась точность предсказания на валидационной выборке, с прекращением роста точности обучение останавливалось.

2.4 Deep Image Prior

Deep Image Prior – нейронная сеть с архитектурой автоэнкодера с пробросочными соединениями, подробно описана в [12]

Данная архитектура была адаптирована для геномной последовательности, закодированной в виде one-hot-encoded матрицы соответственного размера $n \times 4$, где n – длина обрабатываемой геномной области. В данной архитектуре двумерные функции свертки, пулинга, нормализации, апсемплинга были заменены на соответствующие одномерные аналоги. Суть подхода заклюючается в следующем:

1. Выбиралась геномная область длиной 500,000 нуклеотидов. Такой размер области позволял наиболее эффективно проводить расчеты.

2. На области случайно равномерно выбиралось 10% нуклеотидов, которые далее предсказывались, которые обозначаются как тест или маска.

3. Нейронная сеть обучалась получать из случайно сгенерированного массива чисел целевую геномную последовательность. Функция потерь при этом не учитывала тестовые (маскированные) нуклеотиды.

4. Когда функция потерь достигала низких значений, обучение останавливалось. Проверялось, что же предсказывает модель на месте замаскированных нуклеотидов.

3 Результаты

Зависимость от размера предикторной области. Была исследована зависимость качества предсказания нуклеотида от размера использованной предикторной области. Простейшие архитектуры нейронных сетей (две полносвязных модели, сверточная с разным размером сверточного фильтра) были обучены и протестированы на 30 датасетах. Полученные распределения точности приведены на рисунке 5. Для всех моделей наблюдается увеличение точности при увеличении размера предикторной области, что подтверждается статистическим критерием Манна-Уитни.

Предикторная область большего размера содержит больше информации, от которой может зависеть следующий нуклеотид. Тем не менее при геометрическом увеличении размера области качество растет практически линейно. Из этого можно сделать вывод о том, что более далекие нуклеотиды слабее влияют на предсказываемую позицию.

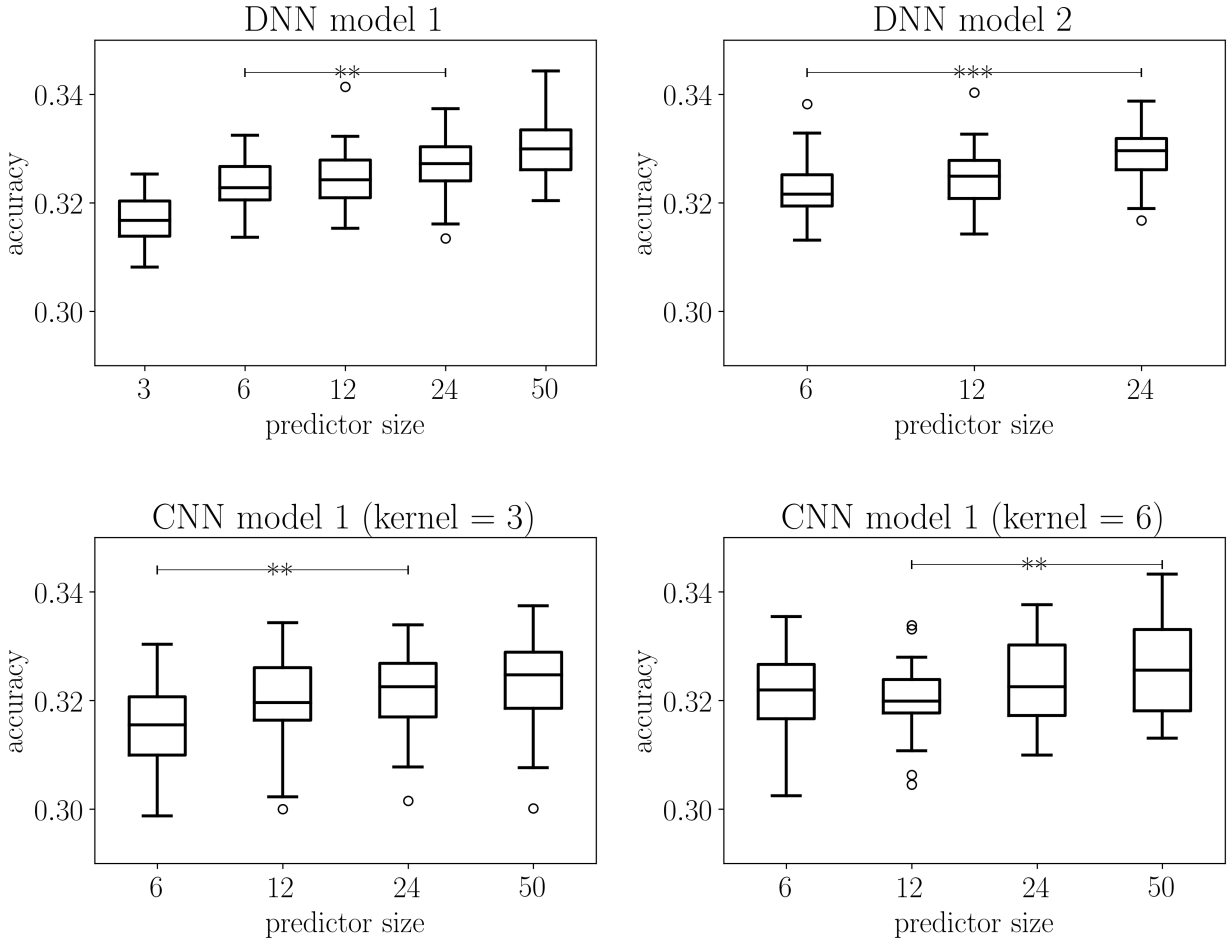
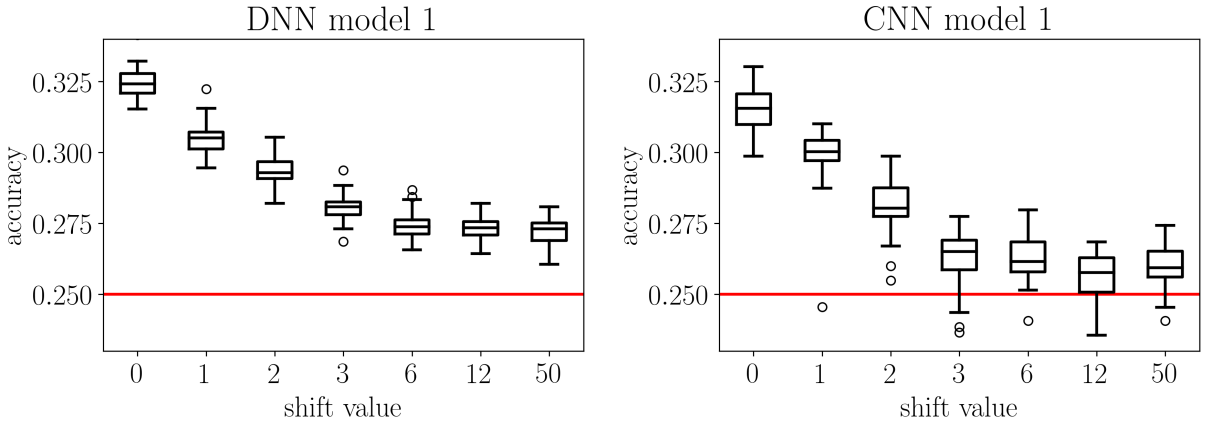


Рис. 5: **Зависимость точности предсказания от размера предикторной области для различных архитектур.**
По горизонтальной оси обозначен размер области. По вертикальной оси показано распределение точностей обученной модели в сете из 30 запусков с различными наборами данных.

Зависимость от отступа. Была исследована зависимость качества предсказания от расстояния между предикторной областью и предсказываемым нуклеотидом для двух моделей – полносвязной и сверточной.

С увеличением отступа качество предсказания падает, причем резко. Это подтверждает то, что в простых моделях (полносвязных и сверточных с небольшим числом параметров) предсказание основывается на ближайших в предсказываемому нуклеотидах.



(a) **DNN model 1**
Полносвязная однослойная модель. Размер предикторной области 12.

(b) **CNN model 1 (kernel = 3)**
Сверточная однослойная модель. Размер предикторной области 6.

Рис. 6: **Зависимость точности предсказания от расстояния между предикторной областью и предсказываемым нуклеотидом (от отступа) для различных архитектур.**
По горизонтальной оси обозначен размер отступа. По вертикальной оси показано распределение точностей обученной модели в 30 запусках с различными наборами данных. Горизонтальная линия отмечает точность случайного предсказания 25%.

Сравнение полносвязных моделей. Мы сравнили между собой полносвязные модели – с одним (DNN model 1) и двумя слоями (DNN model 2) на двух типах данных – предсказание по предикторной области 12 и 24 (рисунок 7). Статистически значимой разницы между моделями не наблюдалось.

Полносвязные модели по построению не могут должным образом использовать информацию о близком расположении и последовательности нуклеотидов. В таких моделях предсказание, большей степени, основывается на нуклеотидном составе предикторной области.

Сравнение сверточных моделей. Мы исследовали несколько вариантов конфигурации сверточного слоя. Результаты тестов приведены на рисунке 8.

Рекурретные модели.

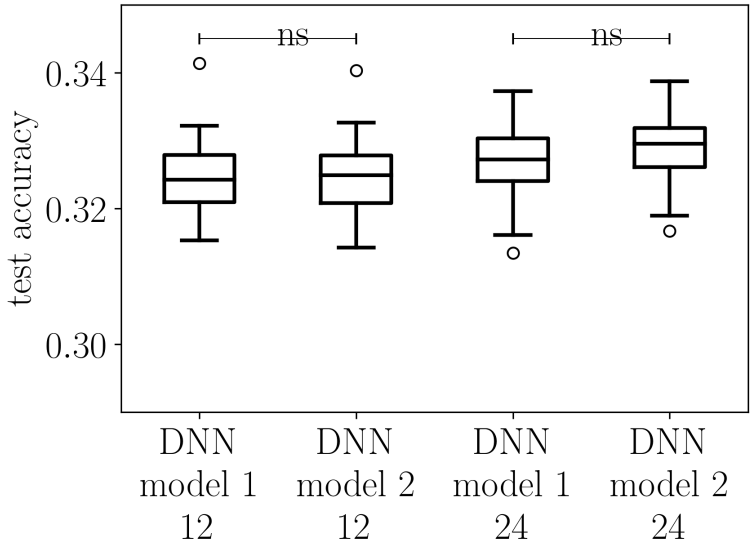


Рис. 7: **Сравнение полносвязных моделей.**
На рисунке показано распределение качестве предсказания полносвязных моделей на двух вариантах данных – с предикторной областью 12 и 24. В обоих случаях разница в качестве предсказания статистически не значима.
Значимость отличия выборок по критерию Манна-Уитни: ns $P > 0.05$, * $P \leq 0.05$, ** $P \leq 0.01$, *** $P \leq 0.001$

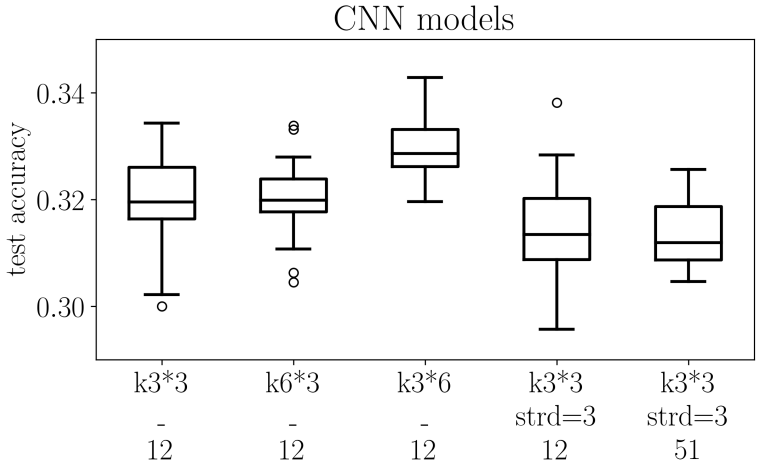


Рис. 8: **Сравнение сверточных моделей.**
По горизонтальной оси отмечены разные конфигурации сверточного слоя в нейронной сети.
Значимость отличия выборок по критерию Манна-Уитни: ns $P > 0.05$, * $P \leq 0.05$, ** $P \leq 0.01$, *** $P \leq 0.001$

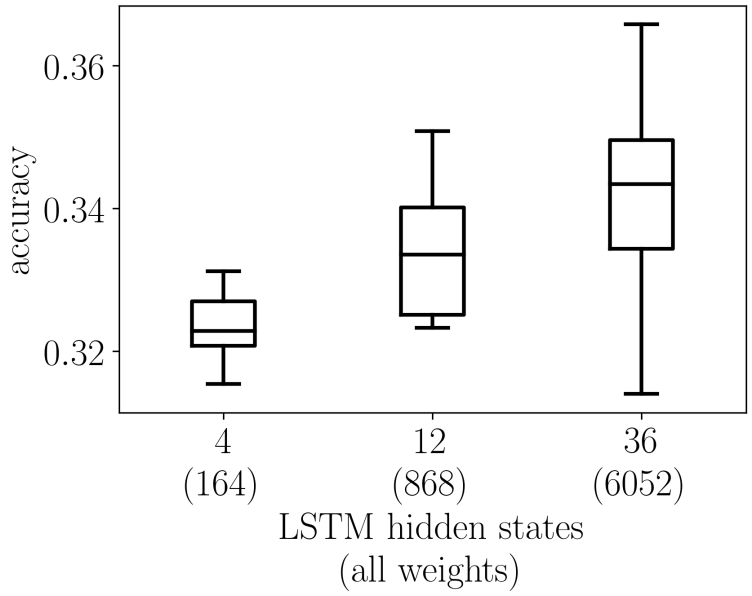


Рис. 9: **Сравнение рекуррентных моделей.**
Значимость отличия выборок по критерию Манна-Уитни: ns $P > 0.05$, * $P \leq 0.05$, ** $P \leq 0.01$, *** $P \leq 0.001$

Список литературы

[1] Garden PW (**1980**) Markov analysis of viral DNA/RNA sequences. ***Journal of Theoretical Biology*** 82(4):679–684. ISSN 00225193. URL http://dx.doi.org/10.1016/0022-5193(80)90186-1.

[2] Tavaré S, Song B (**1989**) Codon preference and primary sequence structure in protein-coding regions. ***Bulletin of Mathematical Biology*** 51(1):95–115. ISSN 0092-8240.

[3] Churchill GA (**1989**) Stochastic models for heterogeneous DNA sequences. ***Bulletin of Mathematical Biology*** 51(1):79–94. ISSN 0092-8240.

[4] Phillips GJ, Arnold J, Ivarie R (**1987**) Mono- through hexanucleotide composition of the Escherichia coli genome: a Markov chain analysis. ***Nucleic Acids Research*** 15(6):2611–2626. ISSN 0305-1048. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC340672/.

[5] Pevzner PA (**1992**) Nucleotide sequences versus Markov models. ***Computers & Chemistry*** 16(2):103–106. ISSN 00978485. URL http://dx.doi.org/10.1016/0097-8485(92)80036-Y.

[6] Budach S, Marsico A (**2018**) pysster: classification of biological sequences by learning sequence and structure motifs with convolutional neural networks. ***Bioinformatics*** 34(17):3035–3037. ISSN 1367-4803. URL http://dx.doi.org/10.1093/bioinformatics/bty222.

[7] Wang J, Wang L (**2019**) Deep Learning of the Back-splicing Code for Circular RNA Formation. ***Bioinformatics*** ISSN 1367-4803, 1460-2059. URL http://dx.doi.org/10.1093/bioinformatics/btz382.

[8] Arefeen A, Xiao X, Jiang T (**2019**) DeepPASTA: deep neural network based polyadenylation site analysis. ***Bioinformatics*** ISSN 1367-4803, 1460-2059. URL http://dx.doi.org/10.1093/bioinformatics/btz283.

[9] Li H (**2019**) Identifying centromeric satellites with dna-brnn page 3.

[10] Chollet F, others (**2015**) *Keras*. URL https://keras.io.

[11] Kingma DP, Ba J (**2014**) Adam: A Method for Stochastic Optimization. ***arXiv:14126980 [cs]*** ArXiv: 1412.6980, URL http://arxiv.org/abs/1412.6980.

[12] Ulyanov D, Vedaldi A, Lempitsky V (**2018**) Deep Image Prior.