Аннотация генома человека с использованием методов машинного обучения

Павел Северилов[1]
severilov.pa@phystech.edu
[1]Московский физико-технический институт

Большое количество белковых последовательностей становится доступным благодаря применению новых технологий секвенирования. Но функциональная аннотация этих белков занимает много времени и ресурсов, и часто проводится только для нескольких организмов. Функции белков классифицируются с помощью онтологии генов (GO). В данной работе применяются методы машинного обучения для решения задачи классификации последовательностей генома человека по GO-классам. Для использования данных методов из белковых последовательностей сэмплированием Гиббса были получены признаки в виде мотивов. Полученные результаты качества классификации сравнимы с качеством методов, дающих лучшие результаты в данной области.

Ключевые слова: *аннотация генома, последовательность белков, онтология генов, классификация, логистическая регрессия, SVM, случайный лес, сэплирование Гиббса, мотивы.*

1 Введение

Прогнозирование функций белков – важная задача вычислительной биологии. Поскольку сложность секвенирования продолжает снижаться, разрыв между количеством размеченных и не размеченных последовательностей продолжает расти [1]. Функции белков описываются классами Gene Ontology (GO) [2]. Прогнозирование функций белка – это задача мультиклассовой классификации, где входом является аминокислотная последовательность, а выходом – набор GO-классов. При анализе только данных аминокислотной последовательности для прогнозирования функций белка можно выделить два основных метода.

Первый метод основан на моделях сопоставления строк, таких как Basic Local Alignment Search Tool (BLAST) [3]. Ключевая идея, лежащая в основе методов BLAST, заключается в нахождении последовательности белков, которые напоминают неизвестную последовательность. Предположение метода заключается в том, что эти белки будут содержать схожие эволюционно консервативные области и мотивы, которые хорошо соответствуют неизвестной аминокислотной последовательности. Затем все метки GO, присвоенные этим похожим последовательностям белков, приписываются неизвестной последовательности [1].

Второй метод заключается в преобразовании аминокислотных последовательностей в признаки, а затем применении методов классификации к этим признакам. Например, DeepGO [4] преобразует аминокислотную последовательность в последовательность k-мер и используют сверточную нейронную сеть для классификации последовательностей по GO-классам. Хорошего качества классификации достигают и другие методы машинного обучения [**?**, 4–7]. Для получения признаков из последовательностей аминокислот применяют различные методы нахождения мотивов [9, 10]. Одним из эффективных методов нахождения мотивов в последовательности является вероятностный метод – сэмплирование Гиббса [11].

В данной работе для классификации последовательности аминокислот по GO-классам применяется метод с преобразованием последовательностей в признаки и применения к

ним методов классификации машинного обучения. Для преобразования последовательности аминокислот в признаки применяется сэмплирование Гиббса.

2 Постановка задачи

Дана выборка

$$\mathfrak{D} = \{\mathbf{s}_i, \{y_{i1}, \cdots y_{im}\}\}_{i=1}^{n},$$

где $\mathbf{s}_i$ – строки последовательностей белков. Последовательности белков представляют собой последовательные аминокислотные остатки. Каждая последовательность составлена из элементов алфавита A: $|A| = 24$. Значения $y_{ik} \in \{0, 1\} \quad \forall k \in \overline{1, m}$ показывают, соответствует k-ый GO-класс данной последовательности $\mathbf{s_i}$ или нет. 0 означает, что последовательности не соответствует GO-класс, 1 – соответствует.

Для использования методов машинного обучения все строки $\mathbf{s_i}$ переводятся в признаки $\mathbf{x_i} \in \{0, 1\}^l$, где l – количество мотивов, используемых в качестве признаков. Значения 0 или 1 в векторах $\mathbf{x_i}$ показывают, принадлежит ли соответствующий мотив последовательности $\mathbf{s}_i$.

Таким образом, имеем выборку

$$\tilde{\mathfrak{D}} = \{\mathbf{x}_i, \{y_{i1}, \cdots y_{im}\}\}_{i=1}^{n},$$

Рассмотривается множество моделей машинного обучения

$$\mathfrak{F} = \{\mathbf{f}_t \colon (\mathbf{w}, \mathbf{X}) \to \hat{\mathbf{y}} \mid t \in \mathfrak{T}\},$$

где $\mathbf{w} \in \mathbb{W}$ – параметры модели, $\hat{\mathbf{y}} = \{\hat{y}_{i1}, \cdots \hat{y}_{im}\} = \mathbf{f}(\mathbf{X}, \mathbf{w}) \in \mathbb{R}^m, \mathbf{X} = \bigcup_{i=1}^{n} \mathbf{x}_i$. Требуется построить алгоритм $f^* \in \mathfrak{F}$ способный классифицировать произвольный объект $\mathbf{x}_i \in \mathbf{X}$ по y_{ik} $\forall k \in \overline{1, m}$ (по каждому GO-классу).

3 Представление данных

3.1 GO-классы белков

Онтология гена (Gene Ontology, GO) – это словарь, состоящий из более 38000 тегов, называемых GO-классами, которые описывают молекулярные действия генов, биологические процессы, в которых происходят эти действия, и клеточные местоположения, в которых они присутствуют [2]. В файле human.swp.GO_terms.txt названия классов GO даны под идентификаторами вида GO:0005829, GO:0042470 и т.д.

3.2 Данные о последовательности белков

Формат данных human.swp.seq:
>1433B_HUMAN YWHAB 0005829 0042470 0048471 0017053 0003714 0000186 0006915 0007411 0051220 0007173 0008543 0010467 0035329 0008286 0000165 0016071 0035308 0045892 0048011 0006605 0007265 OMIM: 601289
MTMDKSELVQKAKL...,

где > – начало записи, затем идет идентификатор записи (1433B_HUMAN), название гена (YWHAB) и список классов "биологических функций относящихся к данному гену: семизначные числа (0005829 0042470 ...) – идентификаторы по Gene Ontology (GO), шестизначные – входные точки в базе данных OMIM (генетические дефекты и наследственные заболевания менделевского типа). Во второй строке записи идет последовательность белка, кодируемого соответствующим геном (MTMDKSELVQKAKL....). Именно эти символьные последовательности необходимо классифицировать по GO-классам. Количество

последовательностей в данных: 21024. Алфавит A, по которому составлены последовательности:

$$\text{A} = \{\text{'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'K', 'L', 'M', 'N', 'P', 'Q', 'R', 'S', 'T',} \\ \text{'U', 'V', 'W', 'X', 'Y', 'Z'}\} \quad (1)$$

3.3 Сэмплирование Гиббса

Преимущество данного сэмплирования заключается в том, что для него не требуется явно выраженное совместное распределение, а нужны лишь условные вероятности для каждой переменной, входящей в распределение. Алгоритм на каждом шаге берет одну случайную величину и выбирает её значение при условии фиксированных остальных. x_i выбираем по распределению $p(x_i|x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n)$ и повторяем.

Алгоритм получения мотивов длиной l:

1. Случайным образом выбирается стартовая позиция мотива в каждой последовательности
2. Случайно выбирается одна последовательность из всех n последовательностей
3. На основе $n-1$ последовательности (кроме выбранной на 2 шаге) создается матрица весов размера $|A| \times l$ с вероятностями каждой буквы в каждой позиции мотива.
4. Для последовательности, выбранной на шаге 2, создается матрица условных вероятностей на основе матрицы с шага 3. Условные вероятности вычисляются для каждой позиции в последовательности и показывают вероятность того, что в данной позиции начинается мотив. По полученным вероятностям находится наилучшая и устанавливается в качестве новой начальной позиции для данной последовательности аминокислот.
5. Процесс повторяется, пока список стартовых позиций мотивов в последовательностях не будет меняться

3.4 Получение признаков

Т.к. минимальная длина последовательности в данных – 4, то длина мотивов, которые получаем сэмплированием Гиббса, была задана 2, 3 и 4. Количество найденных различных мотивов составляет 403, 6280 и 18315 соответственно для мотивов длины 2, 3 и 4. Каждой последовательности поставлен в соответствие вектор признаков длины, равной количеству мотивов, использованных в качестве признаков. Вектор состоит из 0 и 1: 0 означает, что данный мотив не содержится в последовательности, 1 – содержится. При использовании мотивов длиной 2 в качестве признаков, матрица признаков получилась менее разреженной, чем при использовании мотивов длиной 3 или 4, поэтому итоговая матрица признаков содержит информацию только о мотивах длины 2.

На рисунке 1 представлена диаграмма того, сколько раз каждый полученный мотив входит в аминокислотные последовательности исследуемой выборки данных. Видно, что некоторые мотивы встречаются почти в каждой последовательности (например, LL: 19294/21024) и могут быть неинформативны, а некоторые почти нигде не встречаются (например, ZY: 3/21024) и тоже могут быть неинформативны. Поэтому важно провести отбор признаков.

3.5 Отбор признаков

Были посчитаны статистики χ^2 между каждым признаком X и y. Маленькое значение статистики означает, что признак независим от y. Большое значение будет означать, что

функция не случайно связана с y и, следовательно, может предоставить важную информацию. В качестве наиболее важных признаков выбирается k признаков с наибольшим значением статистики χ^2. Среднее значение качества F1-метрики при использовании логистической регрессии для $k = 20$ составляет 0.69, 0.71 для $k = 100$, 0.725 для $k = 200$, 0.727 для всех признаков. Т.к. значение метрики практически не отличаются и использование меньшего количества признаков дает более устойчивую модель значение k выбрано равным 20.

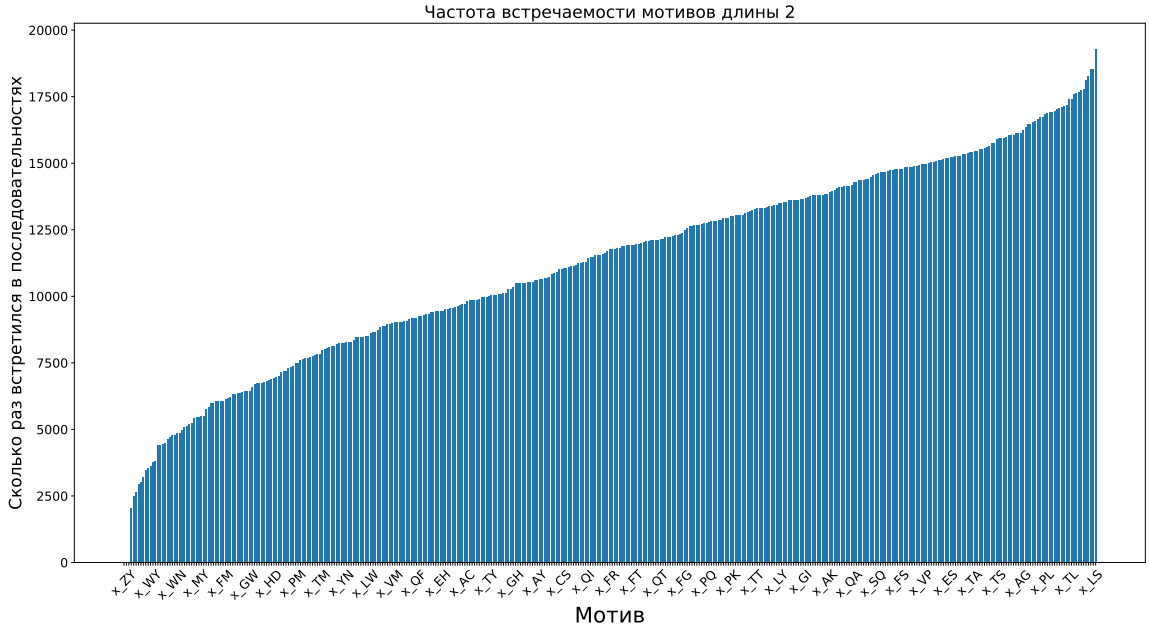

Рис. 1 Число вхождений мотивов длины 2 в аминокислотные последовательности исследуемой выборки данных

4 Вычислительный эксперимент

4.1 Встречаемость GO-классов

Встречаемость GO-классов в последовательностях сильно различается. 80% всех GO-классов, которые встретились в последовательностях аминокислот, встретились меньше, чем в 10 последовательностях, т.е. почти нигде. Из-за такой сильной несбалансированности наиболее вероятным решением задачи классификации одним из методов машинного обучения будет константа 0, т.е. GO-класс не относится к последовательности. Точность классификации на тестовой выборке (20% данных – 4204 последовательности) в таком случае достигала бы почти 100%. Поэтому для решения задачи были взяты только те GO-классы, для которых решение в виде константы давало бы точность меньше 90% на тестовой выборке.

Таким образом, были взяты GO-классы, которые встретились не менее 420 раз в последовательностях. Всего вышло 46 GO-классов из 12879:
’0004984’, ’0005783’, ’0016032’, ’0043066’, ’0020002’, ’0005525’, ’0000122’, ’0005622’, ’0048471’, ’0045087’, ’0043565’, ’0007165’, ’0000139’, ’0000166’, ’0030430’, ’0044281’, ’0005794’, ’0005730’, ’0045944’, ’0006508’, ’0005789’, ’0004930’, ’0005198’, ’0016020’, ’0005509’, ’0019048’, ’0042025’, ’0005615’, ’0006915’, ’0005739’, ’0005654’, ’0005887’, ’0003700’, ’0003723’, ’0006355’, ’0005576’,

’0005524’, ’0003677’, ’0008270’, ’0006351’, ’0005829’, ’0005886’, ’0005737’, ’0046872’, ’0005634’, ’0016021’.

Для каждого из этих классов решалась задача бинарной классификации: соответствует ли этот GO-класс белковой последовательности или нет.

Для сбалансированности поровну классов в решении задачи классификации брались только последовательности, в которых встречались GO-классы, и такое же количество последовательностей, в которых не встречался класс.

4.2 Результаты

В качестве тестируемых моделей машинного обучения были взяты: логистическая регрессия, SVM и случайный лес. Данные были разделены в соотношении 80 к 20 на тренировочную и тестовую выборку. Оценка качества проводилась на тестовом наборе данных. Рассматривались метрики ROC AUC (площадь под кривой ROC) и F1 мера:

$$F_1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}, \text{ где} \quad \text{precision} = \frac{TP}{TP+FP} \quad \text{recall} = \frac{TP}{TP+FN}$$

На рисунках 2, 3, 4 представлены ROC кривые для соответствующих моделей. На каждом рисунке построено 46 ROC кривых для каждого рассматриваемого GO-класса. Зеленым выделена ROC кривая для GO-класса с наибольшим значением F1 и ROC-AUC: GO-класс 0004984. Синим выделена ROC кривая для GO-класса с наименьшим значением F1 и ROC-AUC: GO-класс 0044281. Желтым показаны все остальные ROC кривые.

В таблице 1 представлены результаты классификации логистической регрессией, SVM и случайным лесом. Результаты показывают, что качество работы алгоритмов практически не отличается, но SVM в среднем показывает более точную классификацию. Минимальные значения метрик показывают, насколько хорошо алгоритм классифицирует самый сложный GO-класс. Логистическая регрессия показывает максимальные значения минимальных метрик. Таким образом, логистическая регрессия является более устойчивым алгоритмом для решения данной задачи.

Таблица 1 Сравнение результатов классификации логистической регрессией, SVM и случайным лесом

Метод	mean F1	max F1	min F1	mean AUC	max AUC	min AUC
Логистическая регрессия	0.69 ± 0.08	**0.98**	**0.56**	0.69 ± 0.08	**0.98**	**0.57**
SVM	**0.70** ± 0.08	**0.98**	0.52	**0.70** ± 0.08	**0.98**	0.53
Случайный лес	0.69 ± 0.08	**0.98**	0.52	0.69 ± 0.08	**0.98**	0.54

5 Заключение

В работе убедились, что мотивы, найденные сэмплированием Гиббса, могут являться признаками для использования в решении задачи классификации генома по GO-классам. Качество, полученное такими методами, как логистическая регрессия, SVM, случайный лес сравнимы с результатами, полученными нейронными сетями, которые дают лучшее качество в данной задаче. Главной проблемой задачи является несбалансированность встречаемости GO-классов в последовательностях, что затрудняет решение задачи для редковстречающихся классов. Для таких GO-классов лучше подходят классические методы аннотации генома, чем методы машинного обучения.

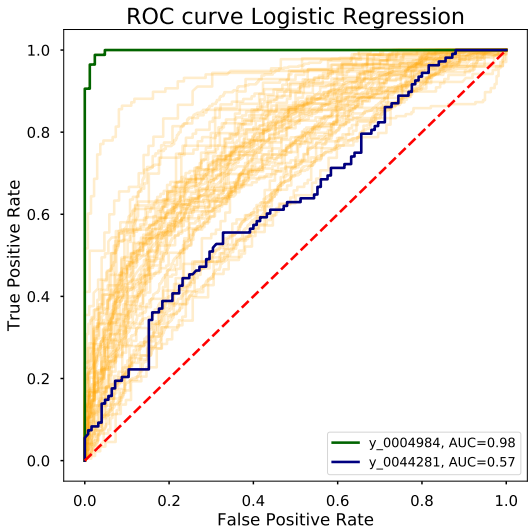

Рис. 2 ROC-AUC для логистической регрессии

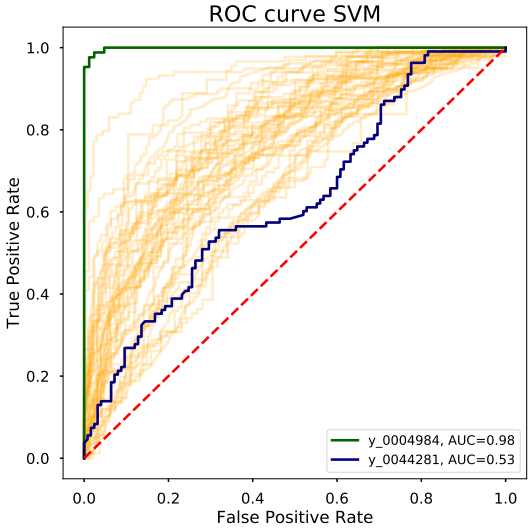

Рис. 3 ROC-AUC для SVM

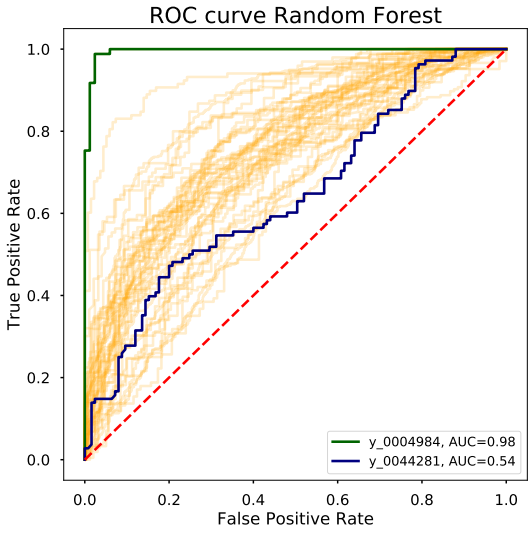

Рис. 4 ROC-AUC для случайного леса

Литература

[1] Zhang C Zheng W Freddolino PL Zhang Y. MetaGO: Predicting Gene Ontology of Non-homologous Proteins Through Low-Resolution Protein Structure Prediction and Protein-Protein Network Mapping // Mol Biol. 2018 Jul 20;430(15). — 2018. — 07. — Vol. 2013. — P. 2256–2265.

[2] A guide to best practices for Gene Ontology (GO) manual annotation / Rama Balakrishnan, Midori Harris, Rachael Huntley et al. // Database : the journal of biological databases and curation. — 2013. — 01. — Vol. 2013. — P. bat054.

[3] Profiti Giuseppe, Martelli Pier Luigi. The Bologna Annotation Resource (BAR 3.0): Improving protein functional annotation // Nucleic acids research. — 2017. — 04. — Vol. 45.

[4] Kulmanov Maxat, Khan Mohammed Asif, Hoehndorf Robert. DeepGO: predicting protein functions from sequence and interactions using a deep ontology-aware classifier // Bioinformatics. —

2017. — 10. — Vol. 34, no. 4. — P. 660–668. — https://academic.oup.com/bioinformatics/article-pdf/34/4/660/25117339/btx624.pdf.

[5] Using Machine Learning and Gene Nonhomology Features to Predict Gene Ontology / Xiuru Dai, Zheng Xu, Zhikai Liang et al. // bioRxiv. — 2019. — https://www.biorxiv.org/content/early/2019/08/09/730473.full.pdf.

[6] Using Deep Learning to Annotate the Protein Universe / Maxwell L. Bileschi, David Belanger, Drew Bryant et al. // bioRxiv. — 2019. — https://www.biorxiv.org/content/early/2019/05/04/626507.full.pdf.

[7] Mahood Elizabeth, Kruse Lars H., Moghe Gaurav. Machine learning: A powerful tool for gene function prediction in plants // Applications in Plant Sciences. — 2020. — 07. — Vol. 8.

[8] Cai Yideng, Wang Jiacheng, Deng Lei. SDN2GO: An Integrated Deep Learning Model for Protein Function Prediction // Frontiers in Bioengineering and Biotechnology. — 2020. — Vol. 8. — P. 391. — URL: https://www.frontiersin.org/article/10.3389/fbioe.2020.00391.

[9] Yang Wen-Yun, Lu Bao-Liang, Yang Yang. A Comparative Study on Feature Extraction from Protein Sequences for Subcellular Localization Prediction. — 2006. — 09. — P. 1–8.

[10] Saidi Rabie, Maddouri Mondher, Mephu Nguifo Engelbert. Protein sequences classification by means of feature extraction with substitution matrices // BMC bioinformatics. — 2010. — 04. — Vol. 11. — P. 175.

[11] Neuwald Andrew, Liu Jun, Lawrence Charles. Gibbs motif sampling: Detection of bacterial outer membrane protein repeats // Protein science : a publication of the Protein Society. — 1995. — 08. — Vol. 4. — P. 1618–32.

[12] Protein Structure Prediction Center. — http://predictioncenter.org/.