

АЛГОРИТМ ПОИСКА КОМБИНАЦИЙ МУТАЦИЙ, СВЯЗАННЫХ С ЛЕКАРСТВЕННО УСТОЙЧИВЫМ ТУБЕРКУЛЕЗОМ

Цурикова Е. А., Сергеев Р. С.

*Белорусский государственный университет, Минск, Беларусь,
e-mail: kattytsurikova@gmail.com*

Математический анализ полногеномных данных микобактерии туберкулеза позволяет сегодня с высокой вероятностью прогнозировать устойчивость к лекарствам первой линии. В то же время устойчивость к лекарствам второй линии плохо объясняется однонуклеотидными полиморфизмами. Устойчивость к препаратам второй линии гипотетически может быть связана с взаимодействием набора мутаций, в то время как влияние каждой отдельной мутации незначительно. При реализации многомаркерных тестов на практике, которые учитывают аддитивные эффекты между множеством отдельных мутаций, возникает проблема вычислительной трудоемкости. В качестве решения проблемы предлагается алгоритм поиска комбинаций мутаций в сочетании с ансамблевым методом машинного обучения бустингом.

Пусть есть набор последовательностей $N = \{N_1, \dots, N_k\}$. Каждая последовательность N_i состоит из набора нуклеотидов и пропущенного символа $\{A, T, G, C, -\}$. Любое отличие j -ого символа i -ой последовательности N_{ij} от соответствующей позиции в референсном геноме N_{0j} будем считать мутацией. Вектор фенотипа $Y = \{Y_1, \dots, Y_n\}$, где n – количество элементов в выборке, $Y_i = 1$, если i -ый образец устойчив к препарату, и $Y_i = -1$, если восприимчив.

Рассмотрим матрицу генотипов размера $n \times m$, строки которой соответствуют последовательностям, а столбцы – позициям, в которых содержится хотя бы одна мутация, $x_{ik} = 1$, если в i -ой последовательности произошла мутация, соответствующая k -ому столбцу, иначе $x_{ik} = 0$. Чтобы не терять значимость мутаций, которые произошли в одной и той же позиции, и в тоже время учитывать их по отдельности, были сформированы две матрицы: XG , которая будет рассматривать мутации, произошедшие в одной позиции как одну, и XD , которая будет учитывать их по отдельности.

Если в образце x_k произошла i -ая мутация, то будем говорить, что x_k обладает признаком s_i . Если в образце x_k произошел набор мутаций с номерами $(i_1, \dots, i_l)$, то x_k обладает составным признаком $(s_{i_1}, \dots, s_{i_l})$. Обозначим через S_i множество тех элементов, в которых произошла i -ая мутация. Тогда $S_{i_1, \dots, i_l}$ – множество элементов, в которых одновременно присутствует набор мутаций $(i_1, \dots, i_l)$. Особенностью задачи поиска комбинаций мутаций является то, что общее количество рассматриваемых мутаций значительно превосходит количество наблюдений $n \ll m$. Задача перебора всевозможных комбинаций мутаций требует больших вычислительных ресурсов. Чтобы избежать этой проблемы можно перейти с помощью приведенного ниже алгоритма от задачи перебора мутаций к задаче перебора образцов, поскольку их количество значительно меньше.

Основные шаги алгоритма:

Вход: выравнивание геномных последовательностей

1. Построить матрицу генотипов X
2. На основании X построить супермножество
3. Из супермножества отобрать элементы по критерию
4. Выполнить переход от множеств к мутациям

Выход: комбинации мутаций и оценки их значимости

Для описания алгоритма удобно оперировать абстрактной сущностью *Element*, которая включает *positions* – набор индексов мутаций, которые произошли у набора образцов *samples*. Псевдокод построения супермножества S_m может быть описан следующим образом:

buildSuperSet(X, m)

$S_m = \{\}$

for j from 0 to m

$S_j = \mathbf{buildElementSj}(X, j)$

joinWithSuperSet(S_m, S_j)

addToSuperSet(S_m, S_j)

Изначально супермножество представляет собой пустое множество. Затем при рассмотрении каждой мутации формируется *Element* S_j , включающий те образцы, для которых в матрице генотипов для j -ой колонки стоят единицы. Псевдокод функции **joinWithSuperSet** можно записать как:

joinWithSuperSet(S_m, S_j)

for element in super set S_m

intersection = **intersectElements**(element, S_j)

addToSuperSet($S_m, \text{intersection}$)

Для каждого элемента уже входящего в супермножество строится пересечение с новым элементом S_j , списки с номерами мутаций объединяются, а для списков образцов находится пересечение. Таким образом пересечение представляет собой построение множества образцов, в которых присутствуют мутации с обоих элементов, а затем S_j добавляется в супермножество.

Добавление нового элемента в супермножество выделено отдельно, поскольку возможны три ситуации:

- Добавляемый элемент является пустым по *samples*. Построенный на каждом шаге элемент S_j не может быть пустым по построению матрицы генотипов, а вот пересечение двух элементов во многих случаях является пустым. В таком случае оно не добавляется в супермножество.
- В супермножестве уже существует элемент с такими же *samples*. Тогда из двух элементов формируется новый с объединенными мутациями и добавляется в супермножество.
- В супермножестве нет элемента с такими же *samples*, тогда элемент добавляется в супермножество.

Для построения набора элементов супермножества используется ансамблевый метод машинного обучения – бустинг. Основная идея бустинга в последовательном построении сильного классификатора из набора слабых. В качестве слабых классификаторов выступают сами элементы супермножества, а именно принадлежит ли набору *samples* проверяемый образец. Для элемента el супермножества и проверяемого образца x_i классификатор будет иметь следующий вид:

$$h_{el}(x_i) = I[x_i \in el] - I[x_i \notin el] \quad (1)$$

При классификации используется оценка информативности. Пусть есть некоторый набор элементов супермножества S . Обозначим через $tp(S)$ количество образцов, которые входят во множество S и являются устойчивыми, а $fn(S)$ – количество образцов, которые также входят во множество S , но являются чувствительными. Тогда задача алгоритма заключается в том, чтобы одновременно максимизировать $tp(S)$ и при этом минимизировать $fn(S)$. Эти два условия можно записать одной функцией, значение которой будет является оценкой информативности классификатора. Ее значение и надо будет максимизировать алгоритму:

$$J(S) = \sqrt{tp(S)} - \sqrt{fn(S)} \quad (2)$$

В ходе вычислительного эксперимента для всех лекарств первой линии, а также их комбинаций, алгоритм нашел одну доминирующую мутацию C2155175G, связанную с устойчивостью к лекарству изониазиду. При бустинге на этих выборках алгоритм добавлял к доминирующей мутации новые, относящиеся к маркерам филогенетических линий, при этом оценка информативности классификаторов не увеличивалась.

Одними из немногих препаратов, для которых играет роль матрица генотипов являются фторхинолоны: офлоксацин, левофлоксацин и их комбинация. По информации из TBDRandomDB, за лекарственную устойчивость к этой группе отвечают мутации по позициям 7570, 7572, 7581, 7582, и учет всех мутаций в одной позиции как одну смог выявить этот набор. Результат для фторхинолонов, за исключением левофлоксацина, можно считать удовлетворительным, поскольку с устойчивостью связан набор мутаций, распределенный по всей выборке, и как результат отдельная мутация наблюдается у маленького количества образцов.

Для аминогликозидов, за исключением канамицина, получились достаточно низкие оценки. В каждом наборе полученных мутаций присутствует мутация A1473252G, которая связана с устойчивостью к этой группе лекарств. Однако этой мутации, даже в сочетании с другими, недостаточно, чтобы построить хороший классификатор. Для лекарства канамицина алгоритм построил набор мутаций, в который вошла мутация G2715356A, связанная с устойчивостью именно к этому лекарству.

Литература

1. Koser C.U. Whole-genome sequencing for rapid susceptibility testing of *M. tuberculosis* / C.U. Koser, J.M. Bryant, J. Becq et al // *N Engl J Med.* – 2013. – Vol. 369. – P. 290–292.
2. Сергеев, Р.С. Алгоритмы анализа и поиска ассоциаций в генетических данных: дис. кандидата физ.-мат. наук: 12.00.01 / Р.С. Сергеев. – Минск, 2019. – 140 л.
3. Вапник В. Н. Восстановление зависимостей по эмпирическим данным / В. Н. Вапник. – М.: Наука, 1979. – 448 с.