Алгоритм для быстрого анализа перепредставленности генов[1]

Сергушичев А. А., аспирант кафедры компьютерных технологий Университета ИТМО, alserg@rain.ifmo.ru

Аннотация

Анализ перепредставленности генов и, в частности, его вариант с преранжированными генами, является очень часто используемым инструментом при анализе биологических данных по экспрессии генов. Проблемой преранжированного анализа является то, что для вычисления статистической значимости результатов требуется проводить сэмплирование, что требует много вычислительных ресурсов. В данной работе предлагается алгоритм, который позволяет переиспользовать результаты сэмплирования, за счет чего достигается возможность быстрого проведения анализа.

Формальная постановка задачи

Преранжированный анализ перепредставленности принимает на вход два объекта:

1. Отсортированный по убыванию массив вещественных чисел S длиной N, в котором для каждого гена i, $1 \leq i \leq N$, записано число S_i, характеризующее его поведение в рассматриваемом биологическом процессе. Например, если $S_i > 0$, то ген активируется, причем тем сильнее, чем больше S_i. В реальных задачах N принимает значения около 20000.

2. Список наборов генов P, $|P| = M$. Каждый набор генов P_i, например, может содержать в себе гены, соответствующие какому-то биологическому процессу. Размер P_i ограничен сверху числом K. Для практических задач можно считать, что K примерно равно 500.

Статистикой перепредставленности набора генов называется функция s, которая принимает на вход набор генов p и возвращает вещественное число такое, что чем больше $s(p)$, тем больше генов в наборе имеют положительное значение S и меньше с – отрицательное, и наоборот для отрицательных $s(p)$.

[1] Работа поддержана грантом Правительства Российской Федерации №074-U01.

В данной работе рассматривается конкретный вид функции $s = s_r$, описанный в работе [1]. Рассмотрим набор генов p размера k. Введем нормализовачный коэффициент $\mathrm{NS} = \Sigma_{i \in p}|S_i|$. Введем вспомогательный массив ES с кумулятивной суммой специального вида:

$$\mathrm{ES}_i = \begin{cases} 0 & \text{если } i = 0, \\ ES_{i-1} + \frac{1}{\mathrm{NS}}|S_i| & \text{если } 1 \le i \le N \text{ и } i \in p, \\ ES_{i-1} - \frac{1}{N-k} & \text{если } 1 \le i \le N \text{ и } i \notin p. \end{cases}$$

Итоговым значением статистики будет максимальное по модулю значение кумулятивной суммы:

$$s_r(p) = \mathrm{ES}_{i^*}, \text{ где } i^* = \arg\max_i |\mathrm{ES}_i|.$$

Задачей анализа перепредставленности является подсчет значения $s_r(p)$ для всех $p \in P$ и p-значений, характеризующих вероятность получить настолько большие или малые значения случайно. Подсчет значения $s_r(p)$ может быть легко произведен по указанной формуле. Самым простым способом для вычисления p-значений является сэмплирование для каждого набора генов p, $|p| = k$, n случайных наборов из k генов и подсчет для каждого такого набора значения статистики s_r. Таким образом можно восстановить нуль-распределение и вычислить p-значение с точностью до $\Theta(1/n)$ по формуле (для правого хвоста распределения):

$$p^+_{value}(P_j) = \Sigma^n_{i=1}[s_r(\pi_{i,j}) \ge s_r(P_j)],$$

где $\pi_{i,j}$ – i-й случайный набор из $|P_j|$ генов.

Такая реализация анализа перепредставленности требует $O(MnK \log K)$ вычислительного времени. Для каждого из M наборов генов требуется n раз вычислить значение статистики для случайного набора генов, что при правильной реализации может быть сделано за $O(K \log K)$.

Идея алгоритма на примере статистики среднего

Для демонстрации идеи алгоритма рассмотрим более простую статистику среднего s_m, задающуюся формулой:

$$s_m(p) = \frac{1}{|p|}\Sigma_{i \in p} S_i.$$

Идея алгоритма с переиспользованием информации сэмплирования состоит в том, что подсчет p-значений для разных наборов генов не требует независимости сэмплов между наборами. С практической точки зрения это означает, что можно на одну итерацию сэмплирования i брать случайный набор из K генов π_i и определять случайные наборы генов $\pi_{i,j}$ для разных наборов P_j как первые $|P_j|$ генов сгенерированного набора:

$$\pi_{i,j} = \pi_i[1..|P_j|].$$

Можно заметить, что при таком определении наборов $\pi_{i,j}$ становится возможно вычислить значение статистики $s_m(\pi_{i,j})$ одновременно для всех j при фиксированном i за время $O(K)$. Это возможно, так как для s_m можно вычислять кумулятивное значение для всех префиксов:

```
vector<double> stat_mean_cumulative(vector<double> S, vector
    <int> p) {
    vector<int> res;
    double sum = 0;
    for (int i = 0; i < p.size(); ++i) {
        sum += S[p[i]];
        res[i] = sum / (i + 1);
    }
    return res;
}
```

Таким образом, с помощью предлагаемого подхода можно вычислить p-значения для статистики s_m за время $O\left(n(K+M)\right)$, а не $O\left(MnK\right)$) при простой реализации (по сравнению со статистикой s_r отсутствует множитель $\log K$, так как значение s_m можно вычислить за линейное время от K, а не за $O(K \log K)$).

Подсчет кумулятивной преранжированной статистики

Для того чтобы ускорить подсчет p-значений для преранжированной статистики s_r, мы буем использовать ту же самую идею об одновременном подсчете значений статистики для всех наборов генов из P для одного сэмпла. В отличие от s_m вычисление $s_r(p)$ сразу для всех префиксов p является нетривиальной задачей.

Далее в этом разделе мы будем рассматривать не исходную статистику s_r, а s_r^+ соответствующую максимальному значению ES, а не

максимальному по модулю:

$$s_r^+(p) = \mathrm{ES}_{i^*}, \text{ где } i^* = \arg\max_i \mathrm{ES}_i.$$

Алгоритм кумулятивного вычисления s_r^+ легко обобщается на s_r.

Геометрическая формулировка задачи

Значение статистики s_r удобно представлять в геометрическом виде. Для этого рассмотрим $N+1$ точку (рис. 1) с координатами нулевой точки (x_0, y_0) равными нулю $(0, 0)$. Для остальных N точек $1 \leq i \leq N$ координаты задаются по формуле:

$$x_i = x_{i-1} + [i \notin p],$$
$$y_i = y_{i-1} + [i \in P] \cdot |S_i|.$$

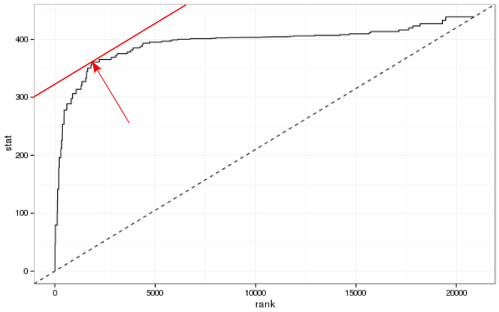

Рис. 1: График, соответствующий преранжированной статистике. Красной стрелкой показана точка, наиболее удаленная от диагонали (выделена пунктиром), соответствующая максимальному значению промежуточной статистики.

Несложно заметить, что $x_N = N - |p| = N - k$ и $y_N = \Sigma_{j \in p}|S_j| = \mathrm{NS}$. Также промежуточные значения статистики ES_i могут быть вычислены как $\mathrm{ES_i} = \frac{1}{NS} y_i - \frac{1}{N-k} x_i$, что пропорционально расстоянию от точки (x_i, y_i) до прямой, проходящей через (x_0, y_0) и (x_N, y_N).

Таким образом, определение итогового значения статистики s_r соответствует определению точки, максимально удаленной от диагонали. Возможность быстро обновлять такую точку при добавлении нового гена в p позволила бы последовательно добавлять гены из π_i и находить значение статистики для всех промежуточных $\pi_{i,j}$.

Применение корневой оптимизации

Рассмотрим, что происходит при добавлении нового гена с индексом g в набор p (рис. 2). В этом случае координаты точек (x_i, y_i) для $i < g$ не изменяются, а координаты всех точек (x_i, y_i) для $i \geq g$ изменяются на одинаковое $(\delta_x, \delta_y) = (-1, |S_g|)$.

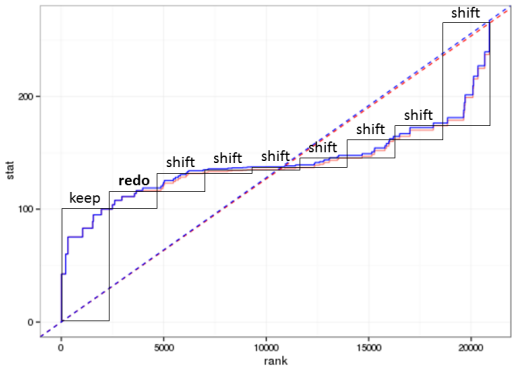

Рис. 2: На рисунке представлены два графика до (красным) и после (синим) добавления одного гена. При этом левая часть графика не изменяется, а правая одинаково сдвигается вверх и влево.

Разобьем все $N + 1$ точек на $b = O(\sqrt{N})$ одинаковых блоков размера $k_b = O(b)$ (для упрощения будем считать, что $N + 1$ делится на b без остатка). Для каждого такого блока будем поддерживать в нем индекс наиболее удаленной точки от диагонали. Соответственно, наиболее удаленную точку среди всех $N + 1$ точек можно найти за время $O(b)$, рассмотрев последовательно наиболее удаленные точки блоков.

Также будем в каждом блоке поддерживать выпуклую оболочку его точек. Заметим, что выпуклая оболочка для всех блоков, кроме содержащего точку g, остается неизменной. Для этого же блока перестроим оболочку за время $O(k_b) = O(b)$ Это можно сделать за линейное время алгоритмом Грэхема ([2]), так как точки уже упорядочены по координате x. Таким образом, за время $O(b)$ мы можем поддерживать выпуклые оболочки для всех блоков.

Чтобы поддерживать индекс самой удаленной от диагонали точки внутри блока, воспользуемся знанием выпуклой оболочки. Самая удаленная от диагонали точка будет всегда лежать на выпуклой оболочке. При этом, если выпуклая оболочка не изменяется, то, так как диаго-

наль при добавлении гена вращается против часовой стрелки, индекс наиболее удаленной точки может только уменьшаться. Соответственно, для $b-1$ блоков, в которых не изменяется выпуклая оболочка, новый индекс самой удаленной точки можно найти последовательно сравнивая текущую точку и точку на выпуклой оболочке, идущую сразу слева, до тех пор пока что-то изменяется. Для одного блока, в котором выпуклая оболочка перестраивается, наиболее удаленную точку найдем простым проходом по всем точкам выпуклой оболочки за не более чем линейное время $O(k_b) = O(b)$. Можно доказать, что такая процедура обновления самой удаленной точки амортизированно требует времени $O(b)$.

Таким образом, предлагаемый алгоритм позволяет найти все промежуточные значения статистики s_r за время $O(Kb) = O(K\sqrt{N})$. Достаточно простым образом можно перейти ко времени работы $O(K\sqrt{K})$, если рассматривать только K точек из набора π_i, а не все N. Суммарное же время работы сэмплирования тогда составляет $O(nK\sqrt{K})$, что в $O(M\frac{\log K}{\sqrt{K}})$ раз меньше, чем при простой реализации.

Заключение

Предварительное тестирование на данных из статьи [3] и наборов генов из базы Reactome [4] показало, что с помощью предложенного алгоритма при работе в четыре потока за одну минуту возможно провести 100000 сэмплов. Этого достаточно, чтобы применяя стандартные методы корректировки на множественное сравнение все еще получать p-значения на уровне 0,01 при тысяче проверяемых наборов. Это, в свою очередь, позволяет использовать стандартные методы корректировки на множественное сравнение, вместо неточного метода, предоложенного в работе [1].

Литература

[1] Subramanian A. et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles / Proceedings of the National Academy of Sciences of the United States of America. — 2005. — Т. 102, № 43. — С. 15545-15550.

[2] Кормен Т. и др. Алгоритмы: построение и анализ — 2-е. — М.: Вильямс, 2005. — 1296 с. — ISBN 5-8459-0857-4.

[3] Wei G. et al. Global mapping of H3K4me3 and H3K27me3 reveals specificity and plasticity in lineage fate determination of differentiating CD4+ T cells. / Immunity. — 2009 — Т. 30, № 1. – С. 155-167.

[4] Joshi-Tope G. et al. Reactome: a knowledgebase of biological pathways / Nucleic acids research — 2005 — Т. 33, Database issue — С. D428-D432.