Отчет о постановке эксперимента

Система: Ubuntu 18.04
Intel Core i7-7500U 2.70GHz
RAM: 8Gb DDR4

Автор: Ахметьянов Азат, СПбГУ
a.r.akhmetyanov@gmail.com

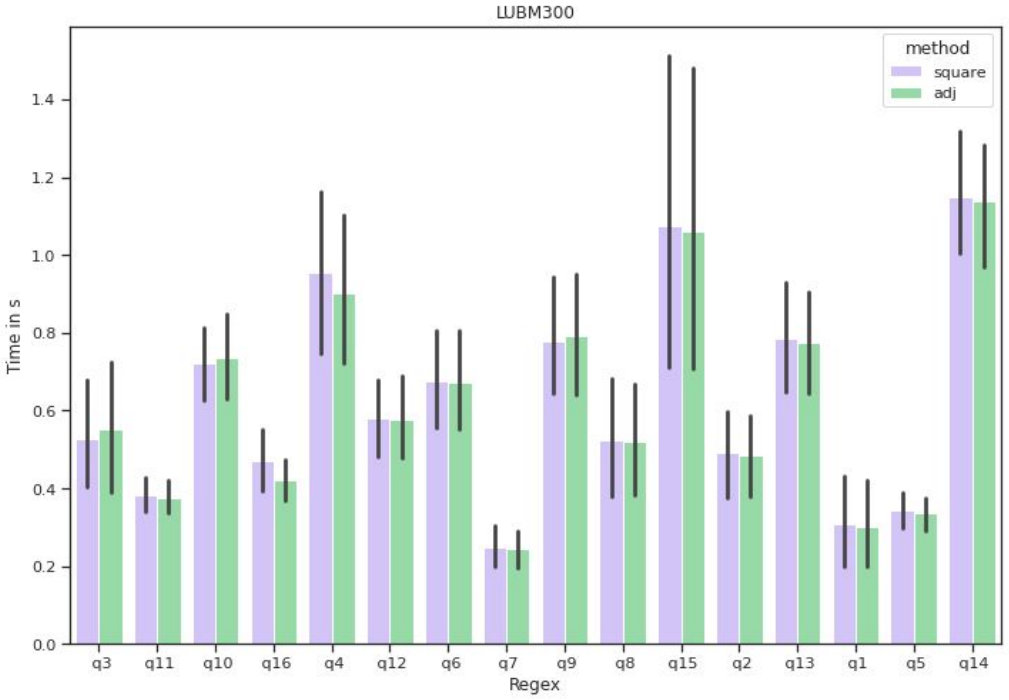
Экспериментальный анализ

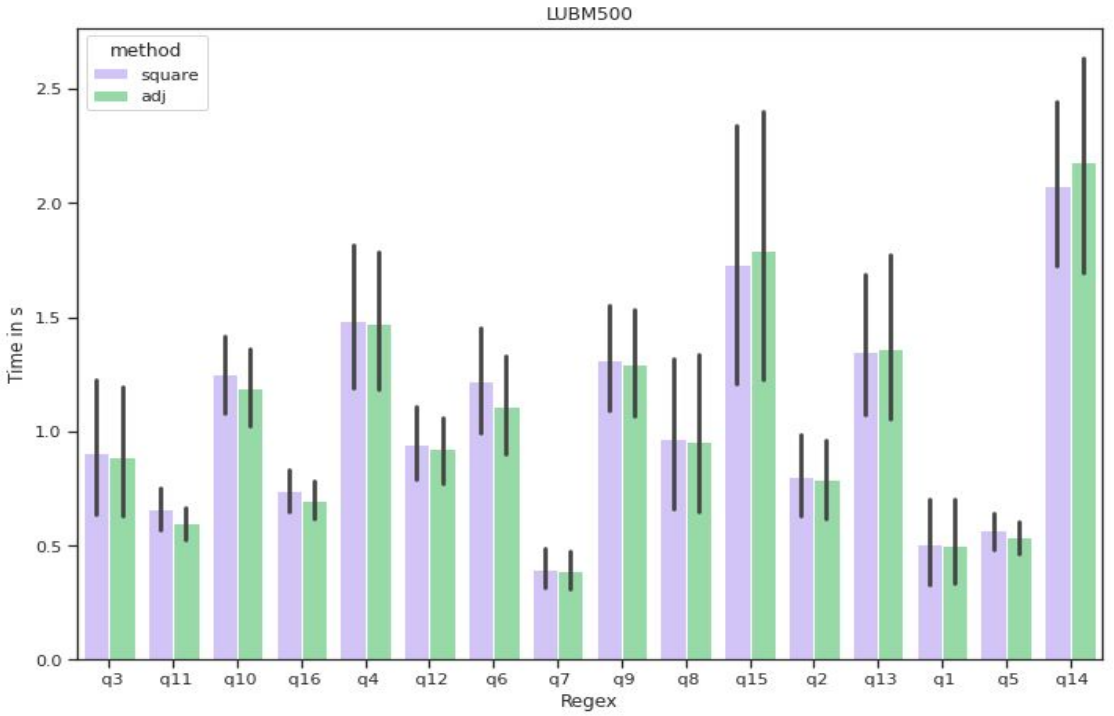
При проведении экспериментов работа алгоритма вычисления регулярных запросов с двумя методами транзитивного замыкания (через возведение в квадрат и через умножение на матрицу смежности) на каждой паре граф-запрос повторялась 5 итераций, контрольные числа и среднее значение времени исполнения для алгоритма записывались в CSV файл с результатами *{GRAPH_NAME}_bench.csv*.

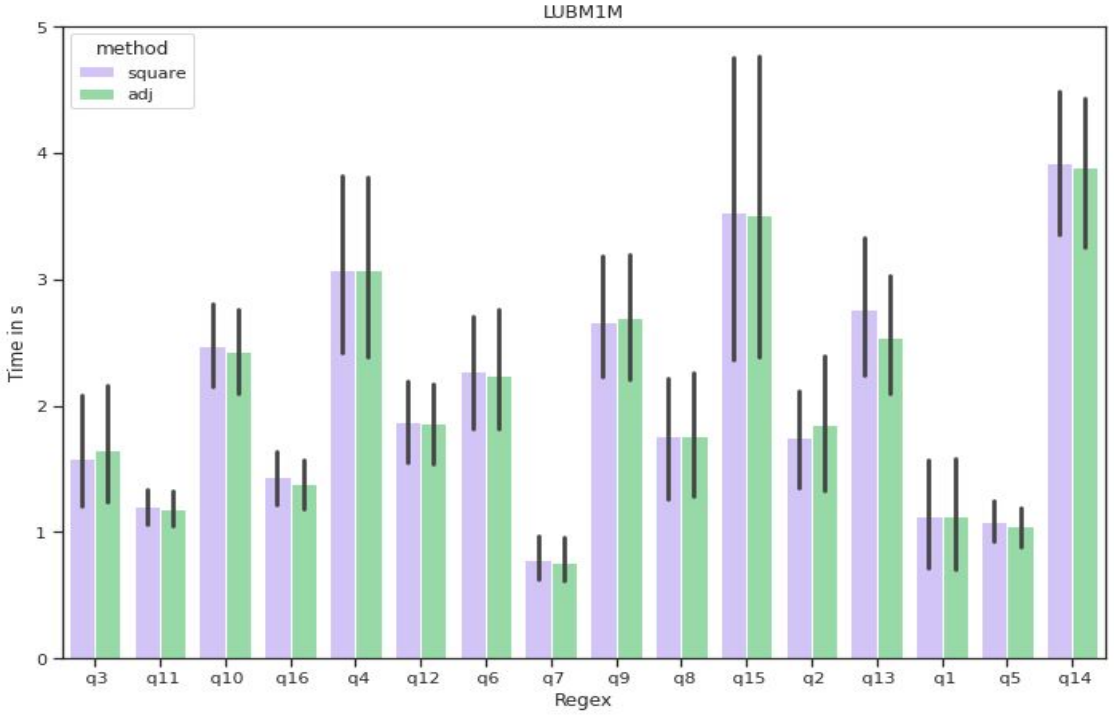
Результаты визуализированы с помощью графика *barplot* библиотеки *seaborn* (этот вид графика показывает на гистограмме средние значения с дополнительно обозначенным доверительным интервалом). Данное представление было выбрано с учетом того, информация о характере распределения, которую показывают *boxplot* и *violinplot*, не так полезна в нашем случае для небольших выборок (< 10 значений).

На графиках сгруппированы запросы с похожей структурой, которые в использованном наборе данных были объединены в папки.

Результаты анализа для набора данных LUBM (300, 500, 1M)

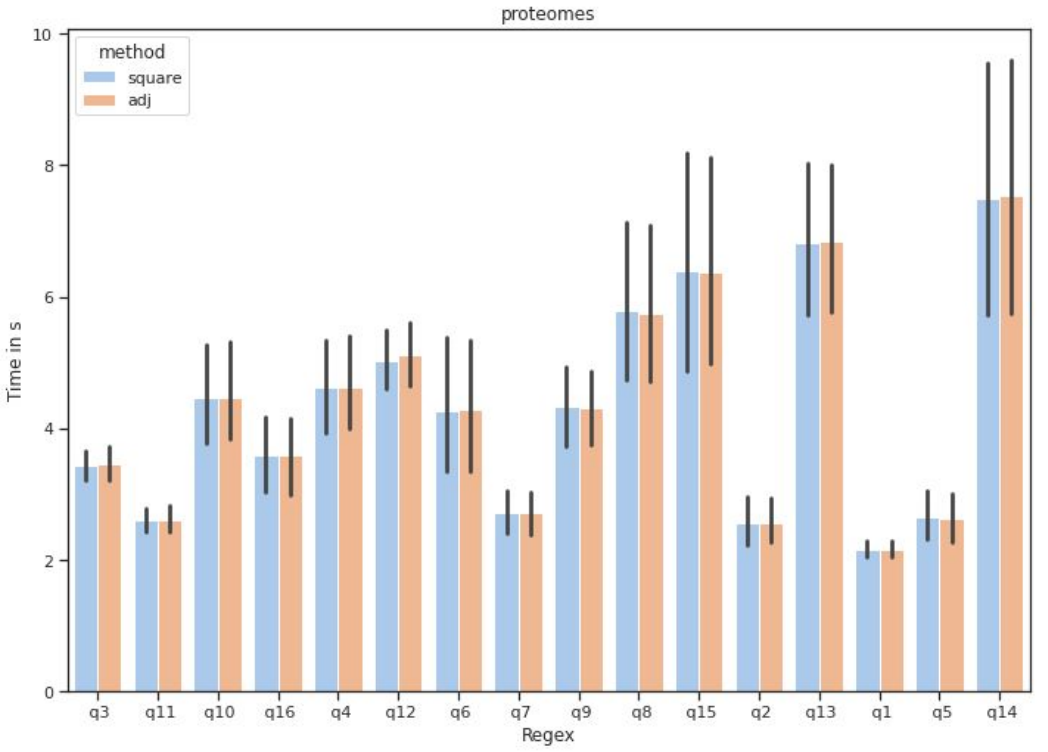


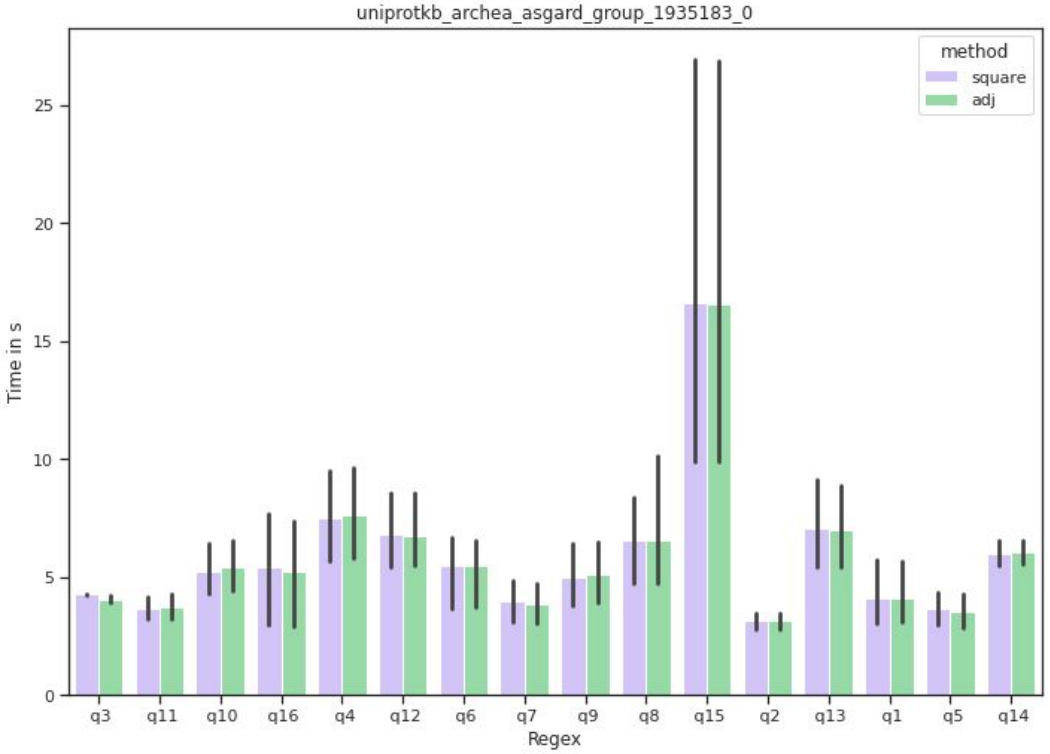




По собранным значениям времени существенного различия между производительностью методов установить не удалось (на графиках средние значения почти совпадают), поэтому было решено перейти к анализу на другом наборе данных.

Результаты анализа для набора данных Uniprot (база последовательностей протеинов)





Анализ на этих графах и наборе запросов также не показал значительных различий между методами как на запросах, вычисление которых в среднем заняло 0-5 секунд, так и на запросах, вычислявшихся около 10 секунд при проведении эксперимента.

Выводы

Использованные наборы данных (LUBM и Uniprot) и проведенный анализ не позволили сделать вывод о разнице в производительности двух методов вычисления транзитивного замыкания. Это может быть связано с тем, что реализация умножения для разреженных матриц в библиотеке *pygraphblas* достаточно эффективна и позволяет компенсировать большое число умножений в одном методе, тем, что на каждое умножение на разреженную матрицу уходит меньше времени, чем на возведение более плотной матрицы в квадрат в другом методе транзитивного замыкания.

Возможно, в качестве продолжения анализа, следует использовать графы и запросы еще большего объема, чтобы лучше исследовать как ведут себя методы при увеличении числа вершин в графах.