PYTHON КАК УДОБНЫЙ МУЛЬТИТУЛ ПРИ ОЦЕНКЕ ПЕРВИЧНЫХ ДАННЫХ ГЕНОМНОГО СЕКВЕНИРОВАНИЯ

И. С. Трусов

Белорусский государственный университет, г. Минск;
ivan.trusau@yandex.by;
науч. рук. – В. В. Гринев, канд. биол. наук, доц.

Появление секвенирования следующего поколения (NGS) привело к генерации огромных объёмов данных. Высокопроизводительные платформы, такие как Illumina HiSeq, производят терабайты необработанных данных, которые требуют быстрой обработки. Контроль качества данных является важным компонентом перед последующим анализом. В рамках нашего проекта по поиску и идентификации полиморфизмов (SNP) в геноме человека мы создали программу на Python, которая позволила нам решить эту проблему. Наша программа выполнена полностью с использованием языка программирования Python. Она может использоваться как автономное приложение и может обрабатывать как сжатые, так и несжатые файлы с данными секвенирования формата *FASTQ*. Наше приложение поддерживает интеграцию Python с другими инструментами и языками программирования с открытым кодом, например, язык программирования R. При оценке данных используется распараллеливание задач с использованием библиотеки Python *multiprocessing*, а общение с пользователем происходит через консоль. Кроме того, наше приложение превосходит по скорости другие аналогичные программы на больших объёмах данных. Приложение доступно по URL-адресу https://github.com/IvanTrusov/Quality_control для быстрой оценки качества данных геномного секвенирования (Illumina).

Ключевые слова: NGS; Python; оценка качества; визуализация данных; SNP.

МАТЕРИАЛЫ И МЕТОДЫ

В качестве эталона в данной работе мы использовали самую популярную программу для оценки качества NGS-чтений – *FastQC* [1]. Библиотеки для тестирования были взяты из European Nucleotide Archive (коды доступа библиотек SRR11479148 и SRR1608628). При работе в Python использовались пакеты *pyfastx, matplotlib, numpy, pandas, scipy, seaborn, statsmodels, PyPDF2, multiprocessing*. Для оценки быстродействия программы на Python использовалась библиотека *time*. Для оценки скорости *FastQC* был использован скрипт на bash с командой *time*. Расчёты проводились на компьютере с 8-ядерным процессором и тактовой частотой 3,2 ГГц и 16 Гб оперативной памяти, а также SSD-диском.

ЗАГРУЗКА ПЕРВИЧНЫХ ДАННЫХ ГЕНОМНОГО СЕКВЕНИРОВАНИЯ В СРЕДУ РАЗРАБОТКИ PYTHON

Для загрузки данных в среду Python был использован пакет *pyfastx* [2] с функцией *pyfastx.Fastq*. Данный пакет был нами выбран в ходе сравнения скорости загрузки и потребления оперативной памяти между аналогами на Python и со стороны пакетов на языке программирования R [3]. Основными преимуществами данного пакета являются скорость и низкое потребление оперативной памяти. Также *pyfastq* отличается высокой гибкостью, так как при прочтении файла данные могут быть сохранены в любом формате.

В нашей работе для оценки качества библиотеки чтений мы использовали частичный анализ всего файла с данными. Для нахождения репрезентативной выборки мы провели сравнение распределения и плотности качества чтений в разных выборках. В первом эксперименте из библиотеки полноразмерных 150-нуклеотидных чтений были взяты 6 выборок с разным числом чтений (рис. 1 А). По результатам, разницы между 100000 и 1 млн чтений не было выявлено. Во втором эксперименте сравнивались 3 выборки с разным методом генерации и размером 100000 чтений, а также контрольная, представляющая весь файл целиком (рис. 1 Б). По результатам был сделан вывод о том, что 100000 первых чтений достаточно для оценки качества всей библиотеки.

Данные результаты позволили нам существенно увеличить скорость обработки больших объёмов данных, в сравнении с FastQC. Сравнение скорости проводилось на двух библиотеках размером 1 млн чтений (рис. 1 В) и 65 млн (рис. 1 Г) чтений с длиной чтения в 150 нуклеотидов. Из результатов видно, что на 1 млн чтений скорость различается не существенно, однако на 65 млн чтений разница составила в 38,5 раз. Таким образом наш подход позволил существенно выиграть в скорости, при этом не потеряв точности в оценке качества данных геномного секвенирования.

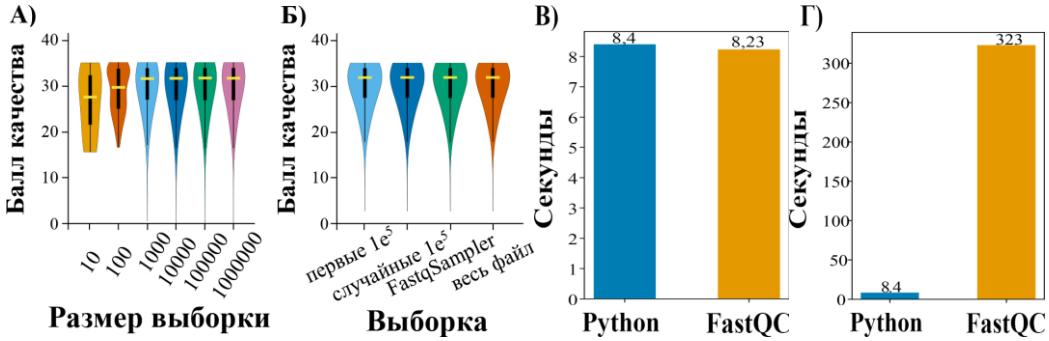

Рис. 1. Распределение качества чтения в зависимости от размера выборки (А) и метода генерации выборки (Б), сравнение скорости между Python и FastQC на 1 млн чтений (В) и на 65 млн (Г)

ОЦЕНКА КАЧЕСТВА ПЕРВИЧНЫХ ДАННЫХ И ИХ ВИЗУАЛИЗАЦИЯ

Существует множество критериев для оценки качества первичных данных геномного секвенирования, одними из важнейших являются показатели качества нуклеотидных чтений, а также степень контаминации чтений адаптерами. Для визуализации данных мы использовали библиотеки Python *matplotlib, seaborn, statsmodels*. Основным изменением в визуальной составляющей, в сравнении с FastQC, является использование столбчатых диаграмм, а также специально подобранная цветовая палитра, способная обеспечить полноценную передачу информации для людей с различными видами дальтонизма (рис. 2). Все алгоритмы, использованные при построении графиков, были написаны без использования сторонних пакетов и библиотек, а также являются аналогичными программе FastQC. Однако для проведения сложных вычислений и удобной работы с данными применялись библиотеки Python *pandas, numpy, scipy*. Функционал готового приложения на Python является схожим с FastQC. Однако в ходе нашей работы были ряд графиков подвергся доработке, например, график демонстрирующий распределение GC-нуклеотидов вдоль чтения. Мы добавили оценку нормальности этого распределения, представленного в виде Q-Q графика вместе с тестом Шапиро-Уилка, реализованного с помощью пакета *scipy* и *statsmodels*. Данное изменение поможет лучше оценивать загрязнённость и однородность полученной библиотеки коротких чтений. Другим изменением стало добавление графика, показывающего количественное содержание адаптерных подпоследовательностей по каждой позиции в чтении. Данный график может стать очень полезным в тех случаях, когда требуется сохранить определённую длину чтения, при этом избавившись от адаптеров.

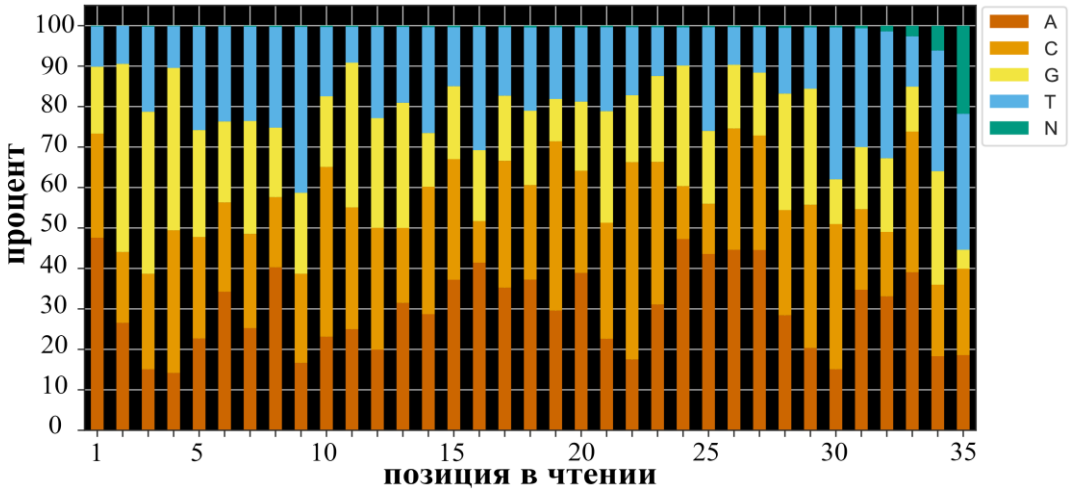

Рис. 2. График доли четырёх основных нуклеотидов для каждой позиции внутри чтения

ЗАКЛЮЧЕНИЕ

По результатам нашей работы видно, что язык программирования Python обладает богатым функционалом и удобными инструментами, позволяющими создавать на их основе полноценные организованные проекты с пайплайнами, которые ничем не уступают уже существующим решениям или даже превосходят их. Так как код нашего приложения является открытым, то его можно без проблем редактировать, модифицировать и интегрировать в другие проекты. Также преимуществами являются кроссплатформенность, простота в установке и использовании. Выходной формат данных представлен в виде файла с векторными графиками формата *PDF*, что обеспечивает удобный доступ как для просмотра, так и для редактирования.

Таким образом по итогам работы мы получили очень удобную и гибкую программу, которая позволяет нам очень быстро проводить оценку данных геномного секвенирования. Кроме того, благодаря интеграции с другими программами обеспечивается прямая передача полученных результатов последующим этапом анализа, что обеспечивает быстроту не только оценки качества данных, но и всего процесса анализа в целом.

Библиографические ссылки

1. *FastQC.* [Электронный ресурс]. – Режим доступа: *https://www.bioinformatics.babraham.ac.uk/projects/fastqc*. – Дата доступа: 10.05.2022.
2. *Python module for fast access to sequences from FASTA/Q file*. [Электронный ресурс]. – Режим доступа: https://github.com/lmdu/pyfastx. – Дата доступа: 31.03.2022.
3. *Гринев, В. В* Оценка качества данных массового параллельного секвенирования в средах программирования R и Python / *В. В. Гринев* [и др.] // Компьютерные технологии и анализ данных (CTDA’2022)*:* материалы III междунар. науч.-практ. конф., Минск, 21–22 апреля 2022 г / БГУ. – Минск, 2022. – С. 306–309.