Сборка генома *de novo* на персональном компьютере

Казаков С.В., аспирант кафедры «Компьютерные технологии»
Университета ИТМО, svkazakov@rain.ifmo.ru
Шалыто А.А., заведующий кафедрой «Технологии программирования»
Университета ИТМО, shalyto@mail.ifmo.ru

Аннотация

Задача *de novo* сборки генома является одной из основных задач в биоинформатике. В настоящее время технологии высокопроизводительного секвенирования (NGS) позволяют получать большие объемы исходных данных – чтений молекулы ДНК. Обработка таких данных (в том числе *de novo* сборка) требует большого объема оперативной памяти компьютера. В работе предлагается метод сборки генома *de novo* на персональном компьютере. Также приводится экспериментальное сравнение с известными сборщиками.

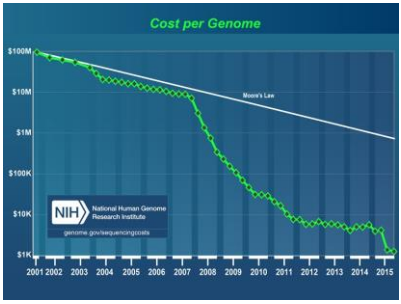
Введение

Задача сборки геномных последовательностей *de novo* является, в определенном смысле, центральной среди всех задач биоинформатики [1]. Это объясняется тем, что без ее решения нельзя приступить к детальному изучению генома живого существа и его анализу с применением других алгоритмов биоинформатики.

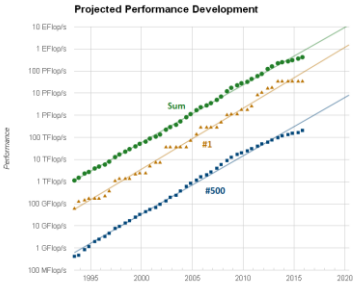
С развитием технологий высокопроизводительного секвенирования (Next-generation sequencing, NGS) был накоплен огромный объем исходных данных. При этом современные технологий позволяют получать сотни гигабайт исходных данных секвенирования за несколько дней. Обработка данных таких объемов требует специальных алгоритмов и, регулярно, специальных вычислительных мощностей для обработки.

Для решения задач по сборке генома к настоящему времени были разработаны десятки алгоритмов по сборке [2-5]. Реализовав предложенные алгоритмы в программы (сборщики), многие из них сейчас широко распространены и популярны (например, ABySS [2], Velvet [3], Spades [4] и др.). Однако подавляющее большинство из них требуют для работы большого объема оперативной памяти. Пытаться собрать геном на персональном компьютере – сложная задача, для которой распространенные сборщики просто не подходят [6-8].

Задача осложняется тем, что развитие технологий высокопроизводительного секвенирования продолжается и по сей день, при этом стоимость секвенирования (а, следовательно, и скорость производства данных) давно обогнала закон Мура [9] (см. рис. 1. а). Развитие компьютерных мощностей же в последние годы имеет тенденцию отставать от закона Мура (см. рис. 1. б). Все это означает, что, вероятно, в ближайшем будущем будет очень много исходных данных, а вычислительных мощностей на их обработку будет не хватать.



а



б

Рисунок 1: а) График изменения цены секвенирования в 2001-2015 годах; б) график изменения производительности компьютеров в 1994-2015 годах.

Подобная проблема относительно недавно была отмечена рядом экспертов, как основная трудность [6]. С 2012 года несколько групп ученых изучали вопросы возможности проведения сборки геномов при ограниченных системных ресурсах, а также предлагали алгоритмы и структуры данных для уменьшения необходимой памяти при сборке [7-8]. В таких исследованиях регулярно ставятся вопросы теоритических оценок для минимально возможных объемов для сохранения основной структуры данных при сборке – графа де Брёйна. Например, в работе [8] 2014 года были доказаны несколько нижних оценок на размер графа де Брёйна, а также показана значительная разница минимальных оценок с оценками, получаемых для используемых на сегодняшний день алгоритмов и структур данных для сборки генома.

Многие из предложенных алгоритмов для минимизации памяти действительно имеют хорошие оценки по хранению графа де Брёйна и работе с ним, приближаясь к теоритическому миниму. Тем не менее, остаются проблемы и с такими подходами – например, не все из них имеют готовую реализацию в виде программ для сборки.

Однако сравнивать сборщики только по расходу памяти тоже

неправильно – качество итоговой сборки не должно сильно ухудшаться по сравнению с результатами сборки в «комфортных условиях».

Таким образом, у ряда ученых возникает вопрос: возможно ли производить сборку генома *de novo* на персональном компьютере? Каким при этом будет качество получаемой сборки?

Предлагаемый подход

В лаборатории “Компьютерные технологии” Университета ИТМО был разработан набор алгоритмов для выполнения сборки *de novo* при сильно ограниченных вычислительных ресурсах. В том числе предложенные методы позволяют производить сборку геномов на персональных компьютерах под управлением любой операционной системы.

Сборщик генома *ITMO Genome Assembler* [10-12] основан на совместном применении графов де Брёйна и графов перекрытий, позволяя использовать преимущества обеих структур данных. При хранении графа де Брёйна используется хеш-таблица с открытой адресацией, при этом сохраняется только сам граф (вершины графа), без сохранения дополнительной информации в вершинах и на ребрах. Данная стратегия позволяет избежать чрезмерного использования памяти при работе с графом. Вся дополнительная информация не теряется, а используется далее на следующих этапах сборки (вместе с другой структурой данных – графом перекрытий).

Подробное описание используемых подходов уже было приведено в работах [10-12]. В настоящей работе приведено более полное сравнение известных сборщиков с *ITMO Genome Assembler* при их работе на персональных компьютерах.

Экспериментальное исследование

Для сравнения были использованы несколько разных сборщиков, в том числе традиционные (Velvet [3], Spades [4], Masurca [13]) и ориентированные на экономию памяти (Minia [7], SparseAssembler [14], ITMO Genome Assembler [10-12]).

Сравнение производилось на следующих исходных данных: бактерия Escherichia coli str. K-12 substr. MG1655, размер генома – 4,6 Мбазы; данные секвенирования были получены на секвенаторе Illumina Genome Analyzer, 20,8 миллионов парных чтений, размер фрагмента 200 нуклеотидов, размер чтения 36 нуклеотидов, покрытие исходного генома

161.5x. Размер исходных данных – 4,0 Гб.

Тестирование сборщиков проводилась на компьютере с 16 ГБ оперативной памяти и 6-ядерным процессором AMD PhenomTM II X6 1090T под управлением OS Linux 3.13.0 x86_64. В зависимости от того, умеет ли сборщик контролировать используемую память, сборщику либо давалась вся память на использование, либо предлагалось использовать в разных экспериментах 4 Гб, 2 Гб, 1 Гб, 0.5 Гб памяти. При этом реально используемый объем оперативной памяти контролировался средствами операционной системы Linux, и пик памяти записывался.

Результаты экспериментов приведены в таблице 1.

Сборщик	Используемая память, Гб	Время работы, min:s	Число контигов	N50, тыс.нукл.	Процент собранного генома	Ошибки сборки/ локальные ошибки
Spades	2.90	19:35	127	82.4	98.0	0/3
Velvet	2.13	7:32	110	95.4	97.6	0/6
Masurca	3.75	11:08	166	54.3	97.9	0/4
Minia	1.21	0:59	461	16.3	97.2	0/1
Sparse-Assembler	0.15	3:51	561	12.7	96.6	0/0
ITMO Genome Assembler	2.30	24:11	246	37.1	98.2	0/7
ITMO Genome Assembler	1.17	12:11	247	35.8	98.1	2/6
ITMO Genome Assembler	0.60	6:40	270	32.3	98.1	4/4

Таблица 1: Численные характеристики полученных сборок

Из представленной таблицы можно сделать следующие выводы:

- SparseAssembler требует всех меньше памяти для сборки,

однако средняя длина контигов (и N50) достаточно маленькая.

- ITMO Genome Assembler может работать при достаточно низких объемах доступной памяти. Качество сборки достаточно высокое, однако может снижаться при уменьшении объема памяти.
- Velvet получает наиболее длинные контиги.

Заключение

В работе представлен обзор текущих возможностей сборщиков по сборке геномов на персональных компьютерах. Было выполнено экспериментальное исследование существующих программ по сборке генома, при этом оценивалось как качество полученной сборки, так и запрашиваемые вычислительные ресурсы.

Экспериментальная проверка показала возможность сборки бактериальных геномов на персональных компьютерах.

Литература

1. Miller J.R., Koren S., Sutton G. Assembly algorithms for next-generation sequencing data // Genomics. — 2010. — 95, pp. 315–327.
2. Simpson J.T., Wong K., Jackman S.D., Schein J.E., Jones S.J. et al. ABySS: a parallel assembler for short read sequence data // Genome Res, — 2009. — 19, pp. 1117–1123.
3. Zerbino D.R., Birney E. Velvet: algorithms for de novo short read assembly using de Bruijn graphs // Genome Res, — 2008. — 18, pp. 821–829.
4. Bankevich A., et al. SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing // J. Comput. Biol., — 2012. — 19, pp. 455–477.
5. Butler J., MacCallum I., Kleber M., Shlyakhter I.A., Belmonte M.K. et al. ALLPATHS: de novo assembly of whole-genome shotgun microreads // Genome Res, — 2008. — 18, pp. 810–820.
6. Kleftogiannis D., Kalnis P., Bajic V.B. Comparing Memory-Efficient Genome Assemblers on Stand-Alone and Cloud Infrastructures // PLoS ONE, — 2013. — 8(9): e75505.
7. Chikhi R., Rizk G. Space-efficient and exact de Bruijn graph representation based on a Bloom filter // Algorithms for Molecular Biology, — 2013. — 8:22.
8. Chikhi R., Limasset A., Jackman S., Simpson J. T., Medvedev P. On the

representation of de Bruijn graphs / In Research in Computational Molecular Biology, — 2014. — PP. 35–55.
9. Jackman S.D., Birol I. Assembling genomes using short-read sequencing technology // Genome Biol, — 2010. — Vol. 11, No. 1, p. 202.
10. Сергушичев А., Александров А., Казаков С., Царев Ф., Шалыто А. Совместное применение графа де Брёйна, графа перекрытий и микросборки для de novo сборки генома // Известия Саратовского университета. Новая серия. Серия Математика. Механика. Информатика. — 2013. — Т. 13, вып. 2, ч. 2, с. 51–57.
11. Александров А., Казаков С., Мельников С., Сергушичев А., Царев Ф. Метод сборки контигов геномных последовательностей на основе совместного применения графов де Брюина и графов перекрытий // Научно-технический вестник информационных технологий, механики и оптики. — 2012. — № 6 (82), с. 93-98.
12. Alexandrov A., Kazakov S., Melnikov S., Sergushichev A., Shalyto A., Tsarev F. Combining de Bruijn graph, overlap graph and microassembly for de novo genome assembly / In Proceedings of "Bioinformatics 2012", — 2012. — p. 72.
13. Zimin A., Marçais G., Puiu D., Roberts M., Salzberg S.L., Yorke J.A. The MaSuRCA genome Assembler // Bioinformatics, — 2013. — 29 (21), pp. 2669–2677.
14. Ye C., Ma Z.S., Cannon C.H., Pop M., Yu D.W. Exploiting sparseness in de novo genome assembly // BMC Bioinformatics, — 2012. — 13 Suppl 6: S1.