УДК 572.112

О ПЕРВЫХ АЛГОРИТМАХ СБОРКИ ГЕНОМА

Протько М.А.[1], магистрант гр.455801

Белорусский государственный университет информатики и радиоэлектроники[1]
г. Минск, Республика Беларусь

Борисенко О.Ф. – канд. физ.-мат. наук

Аннотация. В данной работе представлены краткие сведения, необходимые для изменений и оптимизаций существующих алгоритмов, используемых в сфере биоинформатики, а именно, сборке данных геномов на основе «коротких» ридов секвенирования Сэнгера.

Ключевые слова. Сборка генома, phiX174, shotgun sequencing, поиск регулярных выражений, jigsaw puzzle.

Данная статья является продолжением [1] и посвящена процессу сборки генома, поскольку, как показала практика [2], для построения желаемой модели для поиска болезней экспансии без применения данных алгоритмов не обойтись. Исходя из эвристического подхода, главенствующего в области (что является достаточно справедливым, учитывая огромное множество специализированных задач, с которыми призваны бороться данные инструменты биоинформатики), прежде чем использовать алгоритм/инструмент/идею, следует определить, где и как она применяется, и самое главное, к чему ее применение может привести.

Рассмотрение данных алгоритмов в подробностях имеет смысл, поскольку за их изначальной «очевидностью» подробное описания чего-то, что зовется de facto или алгоритмом state-of-the-art обнаружить достаточно проблематично. Потому, основная цель данной статьи – наиболее кратко и формально изложить данные алгоритмы, помня о том, что их полное описание (с учетом оптимизаций и эвристического подхода/допущений) необходимо объединять, отслеживая развитие по многочисленным источникам.

Начнем с самого первого базового алгоритма сборки, используемого при создании phix174. Самая первая полная версия генома данного бактериофага, несмотря на старания исследователей сразу содержала 33 ошибки, о которых было подробно описано в последующих работах, из чего вариант описанный в [3] отличается от того, что встречается в калибровке секвенаторов Illumina/Solexa [4]. Данные издержки стоит учитывать при рассмотрении алгоритма (а также то, что большая часть генома уже была собрана – исследователям необходимо было получить аналогичный результат, или же собрать прочие «малые» (до 80 bp) фрагменты).

Начальная программа (одна из первых программ сборки) [5] имела следующие возможности: хранение установленных последовательностей, отображение последовательностей, сравнения для гомологий в пределах или между последовательностями; поиск признаков, таких как определенные последовательности, повторяющиеся последовательности и петли для шпилек (hairpin loops). Последовательности для поиска могли включать остатки, указанные только как пурин или любое неизвестное основание, а также специфические нуклеотиды. Эти последовательности имели ограничение до 10 000 нуклеотидов, но фактический лимит по длине последовательности зависел от системных ресурсов компьютера. Время выполнения и количество выходных данных программы (являющихся основными ограничивающими факторами) могли быть скорректированы с помощью параметров, отвечающих за поиск гомологий. Ключевыми параметрами в данной программе являлись максимальная и минимальная дистанция гомологичного поиска, порог качества гомологии (минимальное количество совпадении, минимальная пропорция совпадения, максимальная и минимальная длина и т.д.)

Таким образом, из вышеописанных фактов о спецификации программы, которую использовал Сэнгер для полной сборки генома, наиболее вероятно являющуюся одной из вариаций SEQ, о чем можно только предполагать, потому как названия программы в статьях так и не приводится, но приводится название подпрограмм, что идут в комплекте. Можно предположить, что изначальное программное обеспечение не было специализировано под ДНК, но о разнице алгоритмических подходов к анализу намного большего массива данных (ДНК бактериофагов намного больше чем РНК простейших вирусов) можно только предполагать, поскольку статься [5] не описывает алгоритм, а представляет из себя инструкцию по использованию.

Каким же образом программа [5] определяла гомологию двух последовательностей, учитывая ограниченность ресурсов того времени? Для ответа на данный вопрос следует воспользоваться несколькими попытками, поскольку статья [5] упоминает [6]. Но [6] недалеко уходит от неё [5], являясь демонстрацией возможности программы, без упоминания алгоритма. Со второй попытки автору удалось выйти на упоминание статьи [7], которая позже (при анализе всех программ-сборщиков для

shotgun sequencing) упоминалась при разработке всех последующих программ (SEQ [8], MicroGenie [9], AssemblyLIGN [10], GeneWorks [11], AutoAssembler [12]). Сама статья [7] описывает подробно оптимизации, связанные с областью, но подробные описания самих алгоритмов, к счастью, формальные, описаны в [13].

Код реализации на PL/L, о котором идет речь в [7], нигде не был приведен из-за наличия корневого источника, описывающего изначальный алгоритм. Предполагалось, что изменения алгоритма, вносимые каждой новой программой будут достаточно подробны. Потому использование качественных прилагательных для описания случаев работы программ не являются корректными. Одним из примеров являются нынешние большие фрагменты (500-600 bp по технологии NanoPore), совсем не похожие на «большие» фрагменты 1970-х (до 80bp – сейчас это «короткие» риды Illumina/Solexa). Возникает закономерный вопрос, как вышло так, что первая сборка генома (PhiX175), которая будет положена в алгоритмы калибровки секвенаторов нынешнего времени, имеет такую обрывочную документацию? Одной из причин может быть то, что все исследователи знали друг друга лично, потому в официальной статье [14] (первое полное описание сборки бактериофага) приведены такие объяснения выбора заполнения пропуска как «personal communication» - со временем данное исследование подтвердили более точными методами.

Возвращаясь к одной из первых реализаций алгоритма сборки de novo [7] (то есть, без использования референсного генома – в то время его попросту не было), использовались стандартные алгоритмы сортировки, поиска и быстрой организации данных, которые, согласно [13], скорее всего будут являться: быстрой сортировкой (QuickSort, в параграфе 3.5), в качестве поиска, сравнивая с последующими реализациями, можно предположить об использовании бинарного поиска (поскольку учитывается ограниченный размер памяти и «дизайн для гибкости и простоты» [7], [13]), а в качестве организации данных, возможно, были использованы как деревья (и прочих графовых структурах), так и хэшмапы, так и обыкновенные сеты и кортежи (об их использовании можно предположить, ознакомившись с описанием параграфа 9 [13] о финитных автоматах (или о задаче поиска наибольшей подстроки). Самая первая реализация [7] использовала связный список и быстрый хэш для хранения подстрок (linked list и fast hashing [15])

Работа этого алгоритма заключается в следующем: на первом шаге (procedure) выбирается 10 нумерованных нуклеотидов (размер окна), далее рассчитывается процент каждого из четырех оснований (на данный момент этим занимается стороннее программное обеспечение на шаге до сборки, к примеру, FastQC), на третьем шаге рассчитывается частота встречи всевозможных 16 пар нуклеотидов, а также рассчитывается ожидаемый процент (на основании того, что комплиментарные основания пропорциональны друг другу) – на данном этапе процесс может завершиться при обнаружении аномалий (говорят о неточных данных). Четвертый шаг состоит из расчета всех 64 тринуклиотидов, как для всех рамок считывания в сумме, так и для одной (рамка считывания подробно описана в [4]). Стоит заметить, что программа [7] поддерживает запрет всех процедур на выбранную пару последовательности (что в современных программах сборки не встречается – количество последовательностей слишком велико, чтобы останавливаться и выделять конкретные области). Пятый шаг заключается в сравнивании: начало поиска происходит с фрагментов, имеющих «богатые регионы» – в них хотя бы 6 из 8 последовательных частей относятся либо к пуринам, либо к пиримидинам. Пересечение (overline) происходит с «богатыми» пуринами на «богатые» пиримидины (аналогично с GC/AT). После происходит переход к шестому шагу – пересечению по олигонуклеотидам (последовательностям нуклеотидов до 50 оснований). Для этого последовательность из n нуклеотидов выражается через n пересекающихся «слов», каждое из которых начинается с разного нуклеотида и продолжается до конца последовательности. Далее используется бинарная сортировка на ключах [13] для представления полученных слов в лексикографическом порядке. Цель данного шага является получение словаря с минимальными частями слов для отличия слов друг от друге (современные подходы используют схожую идею – создаются хэшмапы и словари на основании деревьев поиска с различным шагом между тринуклеотидов [1]) – данным минимумом представления строк являлась позиция начала последовательности и сам олигонуклеотид (так был представлен словарь поиска, олигонуклеотиды, скорее всего, не превышали размера 10 оснований [7]). Второй оптимизацией поиска по словарю является представление олигонуклеотидов в созданном словаре согласно частоте их встреч – таким образом можно обнаруживать гомологичные регионы (или же, повторы текста – полностью совпадающие олигонуклеотиды).

Стоит отметить, что частью реализации является поиск позиций с «неидеальным» совпадением. Данные позиции разделяют на две группы: диадные совпадения (dyad matches – совпадения комплиментарных оснований A-T, C-G) и почти гомологичные совпадения последовательностей с заданной строгостью (stringent conditions – в будущем данный термин не прижился). Изначальной идеей для так называемого порога совпадений, было предположение о том, что для нахождения совпадающих последовательностей необходимо, чтобы две последовательности начинались с двух совпадений, для последующей части последовательности выполнялись условия 1-4 (описаны ниже), и также, что после каждого несовпадения следует совпадение. Таким образом, получаем первое

упрощение работы программы (которое можно назвать эвристическим подходом), исходящее из определения гомологии [7].

Вышеописанный порог используется для того, чтобы определить достаточное качество поиска гомологии. Сам расчет производится исходя из следующих условий: если обе пары совпадают, алгоритм продолжает построение гомологии согласно следующему рекурсивному процессу. Предположим, что алгоритм нашел гомологию определенной длины. Затем совершается проверка (согласно [7] loop-out – это либо одно основание, либо пара, либо триада (тринуклеотид), один из которых становится искомой подстрокой в сравниваемой иной последовательности):

1. Следующая пара нуклеотидов совпадает.

2. Совершается "looping.out" 1, 2 или 3 нуклеотидов в первой последовательности, что состоит в том, что путем сравнения второй последовательности с нуклеотидами в первой последовательности могут состояться три совпадения (по одному, двум или трем нуклеотидам).

3. Путем совершения “looping.out” 1, 2 или 3 нуклеотидов во второй последовательности (аналогично 2).

4. Следующая пара нуклеотидов не совпадает, но 2 из последующих 3 пар совпадают.

Пока хотя бы одно из условий 1-4 совпадает, процесс поиска совпадений (пересечений) продолжается. На выходе из цикла первой части алгоритма получается строка, условие гомологичности которой затем проверяется согласно параметрам, описанным в предыдущем абзаце. Стоит отметить, что сами эти параметры определяются «на глаз» («In practice, the algorithm finds all substantial homologies found by eye» [7]), что не является достаточно весомым выводом. Но данная оптимизация все-таки была допущена, поскольку с ее помощью возможно было избежать «длинных бессмысленных гомологий» (“long meaningless homologies” – увы, примера «бессмысленной» гомологии в статье [7] не приводится). Еще одним порогом-параметром является соотношение правильных совпадений (которые прошли все вышеописанные проверки) с длиной потенциального гомологичного региона. Можно настроить дистанцию между двумя гомологичными регионами в двух рассматриваемых последовательностях (настройка шага окна).

Стоит заметить, что в данной реализации пропуски вызваны не работой алгоритма, а рамкой считывания и прочими химическими процессами (то есть, предполагается работа с неполными последовательностями).

Из вышеописанного следует что процесс сборки перового полного генома имел несколько причин возникновения ошибок и неточностей. Одна из них – использование эвристического подхода для некоторых упрощений (пункты 1-4 появились из-за этого упрощения, чтобы сокращать найденные цепочки). Также неточности могли возникнуть из-за того, что в алгоритме [7] не использовались матрицы цен и выравнивания, оставляя выбор порогов пользователю (что давало некую гибкость, результатом которой стали последующие более «специализированные» реализации алгоритма как [3],[6] и [8-12]). Вышеперечисленное говорит о том, что не только человеческий фактор и наличие неотработанной технологии секвенирования того времени влияло на получение результата.

Еще одним важным выводом является то, что точность данной сборки можно рассчитать с помощью конечных цепей Маркова и симуляции Монте-Карло. Рассчитанные данным методом вероятности представлены в [7], вместе с заявлениями авторов, что (речь идет о процедурах описанного алгоритма) «Они также не дают никаких указаний на то, насколько найденные пересечения вероятны (inherently improbable), а также насколько найденная гомология «значимая» (“significant”). Это говорит о том, что сборка, полученная в [4] с произвольной рассчитываемой вероятностью, могла получиться совсем иного вида. Из чего можно заключить, что референсный геном phiX174 и его производная, используемая для калибровки секвенатора Illumina/Solexa [5] также была получена по воле некого случая.

Таким образом, проведенный анализ процесса сборки первого полного генома приводит к возникновению закономерного вопроса: насколько сильно «случай», накопленный посредством нескольких лет исследований влияет на результаты нынешних исследований. Или же, какие предубеждения (bias) мы можем получить, повторив все исследования получения генома калибровки, но изменив полученный в [4] результат согласно случаю, вызванному допущениям в [7]. Ответы на данные вопросы позволят улучшить точность нынешних алгоритмов и данных, а также рассчитать пороговые значения для математических моделей и новых алгоритмов выравнивания и сборки, а также влияние данных допусков на конечный результат (численное влияние, в отличие от предположительного, что можно найти во многих статьях, подобных [16]).

Список использованных источников:

1. Протько, М. А. Об обработке данных высокопроизводительного секвенирования = Processing of high-through sequencing data / М. А. Протько // Компьютерные системы и сети : сборник статей 60-й научной конференции аспирантов, магистрантов и студентов, Минск, 22–26 апреля 2024 г. / Белорусский государственный университет информатики и радиоэлектроники. – Минск, 2024. – С. 18–22.

2. Протько, М. А. Создание математической модели данных короткого секвенирования / М. А. Протько // Информационные технологии и системы 2024 (ИТС 2024) = Information Technologies and Systems 2024 (ITS 2024) : материалы

международной научной конференции, Минск, 20 ноября 2024 г. / Белорусский государственный университет информатики и радиоэлектроники ; редкол.: Л. Ю. Шилин [и др.]. – Минск, 2024. – С. 175–176.

3. F. Sanger, A.R. Coulson [и др.], The Nucleotide Sequence of Bacteriophage phiX174 // Medical research Council Laboratory of Molecular Biology, Hills Road. Cambrige C B2 2QH. England, Received 2 June 1978. J. Mol. Biol. (1978) 125, p 225-246.

4. How much PhiX spike in is recommended when sequencing low diversity libraries on Illumina platforms? [Электронный ресурс]. – Режим доступа: https://knowledge.illumina.com/instrumentation/general/instrumentation-general-reference_material-list/000001527. – Дата доступа: 1.05.2024.

5. Delaney A.D., A DNA sequence handling program. Department of Biochemistry, University of British Columbia, Vancouver, B.C., Canada V6T 1W5. Received 10 November 1981, Nucleic Acids Research, Volume 10 Number 1 1982, IRL Press Limited, 1 Falconberg Court, London W1V 5FG, U.K., p. 61-67.

6. R.Staden Sequence data handling by computer // Nucleic Acids Research, Volume 4 Number 11 November 1977, p 4037-4051. MRC Laboratory of Molecular Biology, Hills Road, Cambridge CB2 2QH, UK, DOI: 10.1093/nar/4.11.403.

7. Korn, L. J., Queen, C. L. and Wegman, M. N., "Computer analysis of nucleic acid regulatory sequences," Proc. Nat. Acad. Sci. USA, Vol. 74,1977, pp. 4401-4405. Staden, R., "Sequence data handling by computer," Nuc. Acids. Res., Vol. 4

8. Douglas L.Brutlag, J.Clayton [и др.] SEQ: A nucleotide sequence analysis and recombination system // Nucleic Acids Research, Volume 10 Number 1 1982, IRL Press Limited, 1 Falconber Court, Londin W1V 5FG, U.K. p 279 -294.

9. Merrifield, R.K. (1994). MicroGenie: Protein Analysis. In: Griffin, A.M., Griffin, H.G. (eds) Computer Analysis of Sequence Data. Methods in Molecular Biology, vol 24. Humana Press.

10. Oxford Molecular Group PLC. AssemblyLIGN 1.0. 9. Oxford, United Kingdom: Oxford Molecular Group PLC; 1998.

11. Broveak T. Geneworks. Biotechnol Software Internet J 1996;13:1114.

12. Parker S. Autoassembler sequence assembly software. Methods Mol Biol 1997;70:107–18.

13.Aho, A. V., Hopcroft, J. E. & Ullman, J. D. (1974) The Designand Analysis of Computer Algorithms (Addison & WesleyPublishing Co., Reading, MA).

14. F. Sanger The Croonian Lecture, 1975^ Nucleotide Sequences in DNA // Proc. R. Soc. Lond. B. 191, 317-333 (1975), Great Britain, Medical Research Council Laboratory of Molecular Biology, Cambrige, The Royal Society, Vol. 191, B. (2 December 1975)

15. Carter,J. L. & Wegman,M. N. (1977)Proc.Ninth AnnualACM Symp.on Theoryof Computing (Boulder,CO), pp. 106-112

16. Alkan, C., Sajjadian, S. & Eichler, E. Limitations of next-generation genome sequence assembly. Nat Methods ***8****, 61–65 (2011). https://doi.org/10.1038/nmeth.1527*

UDC 572.112

ABOUT FIRST DE NOVO GENOME ASSEMBLY PROGRAMS

Protsko M.A.[1]

Belarusian State University of Informatics and Radioelectronics[1]*, Minsk, Republic of Belarus*

Borisenko O.F. – PhD in Physics and Mathematics

Annotation. This paper presents brief information necessary for changes and optimizations of existing algorithms used in the field of bioinformatics, namely, the assembly of genome data based on "short" reads of sequencing by Sagner.

Keywords. Genome assembly, phiX174, shotgun sequencing, regular expression search, jigsaw puzzle.