doi 10.18699/bgrs2024-12.3-07

Децентрализованный алгоритм выравнивания нуклеотидных последовательностей

Дедок В.
Институт математики СО РАН, Новосибирск, Россия
dedok@math.nsc.ru

Ключевые слова: сходство строк; выравнивание последовательностей; параллельные вычисления

Мотивация и цель: Среди задач фундаментальной биоинформатики можно выделить задачу выравнивания нуклеотидных последовательностей. Решение этой задачи требуется в филогенетическом анализе, нахождении эволюционно консервативных участков ДНК, но она крайне сложная и входит в класс NP-сложных.

Возможно ли выполнить полный перебор двух последовательностей? Количество способов выравнивания двух последовательностей длины n и m имеет порядок 2^{n+m}, поэтому вычислительно задача полного перебора превращается в решаемую за бесконечное время.

Задача выравнивания последовательностей в биоинформатике является, в каком-то смысле частным случаем задач поиска совпадений в строках для алфавита, состоящего из четырех символов: A, T, G, C. Для решения такой задачи предложено некоторое количество алгоритмов [1, 2]: алгоритм Ратклифа–Мезнера, Смита–Ватермана, Нидлмана–Вунша. Можно выделить группу алгоритмов по выделению длиннейшей общей подпоследовательности LCS, в которых сопоставляются максимально возможное количество элементов заданных строк с сохранением порядка следования.

Очевидным образом напрашивается идея использования суперкомпьютеров. Известны параллельные реализации алгоритмов Нидлмана-Вунш и Matt (multiple alignment with translation and twists, множественное выравнивание с поворотами и сдвигами) – алгоритм, основанный на методе выравненных пар участков структур (AFP) [3–5]. В работе [3] реализация параллельного алгоритма проводилось на суперкомпьютере «Ломоносов», входящем в состав суперкомпьютерного комплекса МГУ имени М.В. Ломоносова.

Использование суперкомпьютеров такого класса – достаточно дорогое решение поэтому авторами была предложена идея использовать децентрализованное решение, основанное на подходе таких проектов, как MilkyWay@Home, Einstein@Home, SETI@home и др. Данный подход имеет название добровольных вычислений – распределенных вычислений с использованием предоставленных добровольно вычислительных ресурсов. Общая схема участия в таком проекте распределенных вычислений состоит в том, что потенциальный участник загружает клиентскую часть программного кода на свой персональный компьютер и запускает ее. Клиентское приложение по сети интернет обращается к серверу, получает вычислительное задание, данные для обработки, выполняет вычисления и их результаты отправляет обратно. При этом вычисления проводятся в фоновом режиме и с минимальным приоритетом, так что работа такого приложения не сказывается на основной работе пользователя. Такой подход был предложен в

середине 1990-х гг. в связи с бурным развитием сети интернет и лавинообразным увеличением количества подключенных к ней компьютеров пользователей.
Недавним примером успехов подобных методов решения тяжелых вычислительных задач является открытие 5000 околосолнечной кометы космической лабораторией SOHO [6] добровольцами гражданской науки Sungrazer [7]. Поэтому авторы пошли по пути разработки алгоритма, позволяющего выполнять биологические вычисления на персональных компьютерах научных волонтеров.
Методы и алгоритмы: В качестве алгоритмов выравнивания последовательностей были выбраны 2 алгоритма: Matt [5] и разработанный авторами итерационный алгоритм последовательного поиска расширяющейся последовательности. Суть предложенного алгоритма можно описать следующей схемой. Пусть у нас имеется две последовательности U и D. В данных последовательностях имеются уже выровненные подпоследовательности U_0 и D_0. Последовательности U и D можно представить в виде $U = U_L U_0 U_R$, $D = D_L D_0 D_R$. На каждом шаге алгоритма будем пытаться расширить имеющиеся последовательности U_0 и D_0 символами из U_L и D_L слева и U_R и D_R справа. В случае, если это возможно, выровненные подпоследовательности будут соответственно расширяться, а расширенные последовательности будут сохраняться в памяти для дальнейшей попытки расширения.
Для распараллеливания вычислений на основе алгоритма Matt используется метод коллективного решения, предложенный в работе [3] и адаптированный для вычислений на отдельных вычислительных узлах.
Мастер-процесс (серверный процесс) на первом этапе алгоритма получает всевозможные пары структур и равномерно распределяет их между всеми доступными вычислительными узлами. Далее происходит выравнивание всеми процессами и сбор данных в мастер-процессе.
Результаты: Результатом данной работы является компьютерная реализация алгоритмов Matt и последовательного поиска расширяющейся последовательности. Клиентская часть обоих алгоритмов реализована в виде хранителя экрана, выполняющегося в периоды неактивности рабочих месте. Серверная часть реализована в виде приложения, работающего в среде ОС Linux. Численные эксперименты показали большую эффективность подхода, основанного на алгоритме Matt в случае, когда одновременно работает достаточно много клиентских приложений и они готовы оперативно принимать вычислительные задания. При этом, если вычислительные узлы с клиентскими приложениями появляются нерегулярно, более эффективным является подход расширяющейся последовательности.
Выводы: Подводя итог исследованию, можно с уверенностью сказать, что подход, основанный на добровольных вычислениях, вполне имеет право на жизнь и позволяет обрабатывать последовательности длиной в десятки тысяч символов на обычных персональных компьютерах научных волонтеров, не требуя при этом мощностей суперкомпьютеров. При этом результирующий алгоритм должен предусматривать этап верификации результатов, полученных на внешних вычислительных узлах, но эта задача на порядки более простая и может быть легко выполнена и не очень производительных ресурсах.

Но при этом стоит отметить и минус такого алгоритма – очень сложно гарантировать время его работы. Но при достаточном количестве пользователей системы этот недостаток может быть практически сведен на нет.
Финансирование: Работа выполнена при финансовой поддержке программы фундаментальных научных исследований СО РАН № I.1.5 (проект FWNF-2022-0009).

Decentralized algorithm for nucleotide sequence alignment

Dedok V.
S.L. Sobolev Institute of Mathematics, SB RAS, Novosibirsk, Russia
** dedok@math.nsc.ru*

Key words: string similarity; sequence alignment; parallel computing

Motivation and Aim: Among the problems of fundamental bioinformatics, one can highlight the task of aligning nucleotide sequences. The solution to this problem is required in phylogenetic analysis, finding evolutionarily conserved DNA regions, but it is extremely complex and belongs to the class of NP-difficult.
Is it possible to perform an exhaustive search between two sequences? The number of ways to align two sequences of length n and m is of the order of 2^{n+m}, so computationally the exhaustive search problem becomes solvable in infinite time.
The problem of sequence alignment in bioinformatics is, in a sense, a special case of the problem of finding matches in strings for an alphabet consisting of 4 characters A, T, G, C. To solve this problem, a few algorithms have been proposed [1, 2]: Ratcliffe–Mezner, Smith–Waterman, Needleman–Wunsch algorithm. We can distinguish a group of algorithms for identifying the longest common LCS subsequence, which compare the maximum possible number of elements of given strings while maintaining the order.
The obvious idea is to use supercomputers. There are known parallel implementations of the Needleman-Wunsch and Matt algorithms (multiple alignment with translation and twists), an algorithm based on the method of aligned pairs of structure sections (AFP) [3–5]. In work [3], the implementation of the parallel algorithm was carried out on the Lomonosov supercomputer, which is part of the supercomputer complex of Moscow State University named after M.V. Lomonosov.
The use of supercomputers of this class is a rather expensive solution, so the authors proposed the idea of using a decentralized solution based on the approach of projects such as MilkyWay@Home, Einstein@Home, SETI@home, etc. This approach is called voluntary computing - distributed computing using provided volunteer computing resources. The general scheme of participation in such a distributed computing project is that the potential participant downloads the client part of the program code onto his personal computer and runs it. The client application accesses the server over the Internet, receives a computational task, data for processing, performs calculations and sends their results back. In this case, calculations are carried out in the background and with minimal priority, so that the operation of such an application does not affect the user's main work. This approach was proposed in the mid-1990s. due to the rapid development of the Internet and an avalanche-like increase in the number of user computers connected to it.

A recent example of the success of such methods for solving heavy computational problems is the discovery of the 5000th circumsolar comet by the SOHO space laboratory [6] by Sungrazer citizen science volunteers [7]. Therefore, the authors took the path of developing an algorithm that allows performing biological calculations on the personal computers of scientific volunteers.

Methods and Algorithms: Two algorithms were chosen as sequence alignment algorithms: Matt [5] and an iterative algorithm developed by the authors for sequential search for an expanding sequence. The essence of the proposed algorithm can be described by the following diagram. Let us have two sequences U and D. These sequences contain already aligned subsequences U_0 and D_0. Sequences U and D can be represented as $U = U_L U_0 U_R$, $D = D_L D_0 D_R$. At each step of the algorithm, we will try to expand the existing sequences U_0 and D_0 with symbols from U_L and D_L on the left and U_R and D_R on the right. If possible, the aligned subsequences will be expanded accordingly, and the expanded sequences will be stored in memory for further expansion attempts. To parallelize calculations based on the Matt algorithm, the collective solution method proposed in [3] and adapted for calculations on individual computing nodes is used. At the first stage of the algorithm, the master process (server process) receives all possible pairs of structures and distributes them evenly among all available computing nodes. Next, alignment occurs by all processes and data is collected in the master process.

Results: The result of this work is a computer implementation of the Matt algorithm and the sequential search for an expanding sequence. The client part of both algorithms is implemented in the form of a screen saver that runs during periods of inactivity on the desktop. The server part is implemented as an application running in the Linux OS environment.

Numerical experiments have shown the great efficiency of the approach based on the Matt algorithm in the case when quite a lot of client applications are running simultaneously, and they are ready to quickly accept computational tasks. However, if computing nodes with client applications appear irregularly, the expanding sequence approach is more effective.

Conclusion: Summing up the study, we can say with confidence that the approach based on voluntary computing has the right to life and makes it possible to process sequences of tens of thousands of characters in length on ordinary personal computers of scientific volunteers, without requiring the power of supercomputers. In this case, the resulting algorithm must include the stage of verifying the results obtained on external computing nodes, but this task is orders of magnitude simpler and can be easily completed even with not very productive resources. But it is worth noting the disadvantage of such an algorithm – it is very difficult to guarantee its operating time. But with a sufficient number of clients, this disadvantage can be practically eliminated.

Funding: The study is supported by the SB RAS program of fundamental scientific research No. I.1.5 (project FWNF-2022-0009).

Список литературы/References

1. Durbin R., Eddy S., Krogh A., Mitchison G. Biological sequence analysis: probabilistic models of proteins and nucleic acidsю Cambridge University Press, 1998
2. Знаменский С.В. Модель и алгоритм выравнивания последовательностей. *Программные системы: теория и приложения*. 2015;1(24):189-197
   [Znamenskij S.V. A model and algorithm for sequence alignment. *Program Systems: Theory and Applications.* 2015;1(24):189-197 (in Russian)]

3. Воеводин В.В., Шегай М.В., Попова Н.Н., Суплатов Д.А., Швядас В.К. Использование параллельных вычислений для эффективного построения множественных выравниваний структур белков. В: Параллельные вычислительные технологии (ПаВТ'2016). Архангельск: Издательский центр ЮУрГУ, 2016;81-92
[Voevodin V.V., Shegay M.V., Popova N.N., Suplatov D.A., Svedas V.K. Use of parallel computing in effective protein multiple structure alignment. In: Parallel computational technologies (PCT'2016). Arkhangelsk, 2016;81-92 (in Russian)]
4. Тетуев Р.К., Пятков М.И., Панкратов А.Н. Параллельный алгоритм глобального выравнивания протяжённых аминокислотных и нуклеотидных последовательностей. *Математическая биология и биоинформатика*. 2017;12(1):137-150. doi 10.17537/2017.12.137
[Tetuev R.K., Pyatkov M.I., Pankratov A.N. Parallel algorithm for global alignment of long aminoacid and nucleotide sequences. *Math Biol Bioinform*. 2017;12(1):137-150. doi 10.17537/2017.12.137 (in Russian)]
5. Menke M., Berger B., Cowen L. (2008) Matt: Local Flexibility Aids Protein Multiple Structure Alignment. *PLoS Comput Biol*. 2008;4(1):e10. doi 10.1371/journal.pcbi.0040010
6. ESA, NASA Solar Observatory Discovers Its 5,000th Comet. https://science.nasa.gov/science-research/heliophysics/esa-nasa-solar-observatory-discovers-its-5000th-comet/
7. The Sungrazer Project. https://science.nasa.gov/citizen-science/the-sungrazer-project/