Исследование производительности алгоритмов множественного выравнивания нуклеотидных и белковых последовательностей на вычислительном кластере

К.В. Романенков

В статье содержится описание проведенного тестирования различных параллельных алгоритмов множественного выравнивания (модификация последовательного алгоритма Muscle с помощью системы Parus, ClustalW-MPI, Dialign P) на многопроцессорных кластерных системах. Для исследования было сформировано 3 файла в FASTA формате с семействами белков и семействами нуклеотидных повторов в геноме человека. В статье также даны краткие характеристики систем, на которых производился запуск программ, и анализ полученных результатов.

1. Введение

Алгоритмы множественного выравнивания представляют собой инструмент для установления функциональных, структурных или эволюционных взаимосвязей между биологическими последовательностями.

Несмотря на то, что задача множественного выравнивания сформулирована более 15 лет назад, она не утрачивает своей актуальности в свете развития новых технологий и методов. Появление новых, более высокоуровневых средств (к примеру, кроссплатформернного биоинформатического проекта UGENE[1]) не отменяет наличия качественной и быстродействующей базы, каковой является набор алгоритмов множественного выравнивания. Однако сложность задачи обусловливается экспоненциальным ростом времени счета не параллельной программы при увеличении либо числа биологических последовательностей, либо их длины. Поэтому существующие алгоритмы жертвуют качеством выдаваемого ответа, увеличивая эффективность собственного выполнения.

Активное внедрение многопроцессорных и многоядерных архитектур дает возможность значительно сократить время решения реальных задач, особенно тех из них, в которых исследуемая область плохо сужается в процессе итеративного подхода (то есть каждый шаг в независимости от предыдущего высчитывается по новой), позволяя реализовывать ту или иную вариацию перебора. В качестве примера реальной задачи, помимо целей эволюционной биологии, для которой исторически и создавались алгоритмы, можно привести проблему синтеза лекарственных препаратов (особенно антибиотиков), когда скорость мутаций в популяциях бактерий практически сравнялась, а местами и опередила скорость синтеза новых лекарственных препаратов традиционным способом [2 – 4]. Еще одним примером может служить раздел моделирования различных биологических комплексов (например, клеточная мембрана).

2. Виды множественного выравнивания

Практически все алгоритмы множественного выравнивания базируются на парном выравнивании последовательностей, которые, в свою очередь, зачастую основаны на алгоритмах динамического программирования (Алгоритм Нидельмана-Вунша, алгоритм Смита-Ватермана). Оба этих метода используют матрицу размера (N+1)*(M+1), где N и M длины соответствующих последовательностей, затем, исходя из значений, занесенных в эту таблицу согласно ходу алгоритма, строится оптимальное парное выравнивание.

Процедура множественного выравнивания аналогична парному, с той лишь разницей, что уже выровненные последовательности становятся так называемым профилем выравнивания, внутри которого запрещено производить перемещение последовательностей друг относительно друга, и которые можно считать одной последовательностью для дальнейшей работы

алгоритма. Изначально производится кластеризация входных данных, с той целью, чтобы выравнивания происходили, по возможности, в наиболее схожих последовательностях.

Существует множество реализаций алгоритмов множественного выравнивания, основанных на различных моделях:

1. Прогрессивные.
2. Итеративные.
3. Скрытые Марковские модели.
4. Генетические алгоритмы.

Наибольшее распространение получили алгоритмы, представляющие первые две группы.

Прогрессивное выравнивание характеризуется меньшим временем выполнения, так как процесс кластеризации идет только в самом начале метода, таким образом, гарантируется, что каждая последовательность будет выровнена не более одного раза. Этот подход позволяет ощутимо сократить время счета, однако получаемые результаты не всегда могут похвастаться отменной биологической корректностью, так как зависят от того, как будет произведена начальная кластеризация.

Алгоритм итеративного выравнивания подразумевает возможность проведения повторной кластеризации, если в ходе работы метода обнаружится, что точность выравнивания не удовлетворяет заданным критериям. Такой подход позволяет достичь наперед заданной точности, но при больших объемах входных данных это требует серьёзных временных затрат.

3. Рассмотренные алгоритмы

Для исследования были выбраны параллельные версии (основанные на библиотеке MPI) следующих алгоритмов:

1. ClustalW-MPI [5].
2. Dialign P [6].
3. Параллельная реализация Muscle с помощью Parus [7].

Первый алгоритм является прогрессивным методом, второй алгоритм представляет собой итеративный подход, а третий объединяет в себе итеративную и прогрессивную схемы. Существует довольно много последовательных алгоритмов множественного выравнивания, в том числе и opensourse проекты, но для параллельных версий оказались доступны исходные коды только этих программ.

4. Обзор использованных вычислительных кластеров

При исследовании были задействованы два многопроцессорных комплекса, к которым имеется доступ из сети МГУ:

1. СКИФ МГУ "ЧЕБЫШЕВ".
2. BlueGene P.

Ниже приведены краткие характеристики обеих систем:

Таблица 1. СКИФ МГУ "ЧЕБЫШЕВ"

Пиковая производительность	60 TFlop/s
Производительность на Linpack	47.04 TFlop/s (78.4% от пиковой)
Модель процессора	Intel Xeon E5472 3.0 GHz

Таблица 2. BlueGene P

Пиковая производительность	27.2 TFlop/s
Производительность на Linpack	23.2 TFlop/s (85% от пиковой)
Модель процессора	4-х ядерный процессор, каждое из которых представляет собой PowerPC450 850 MHz

Dialign P запускался на BlueGene P, а остальные алгоритмы на СКИФЕ.

5. Обзор входных данных

Для проведения исследования было сформировано 3 файла в FASTA [8] формате, два из них содержат белковые последовательности, а в остальных находятся нуклеотидные цепочки.

5.1 Файл Pfam.fasta

Файл содержит последовательности из 13 семейств белков, информация взята из протеиновой базы данных Pfam, характеристики:

1. Число последовательностей - 1011
2. Средняя длина — 526.

5.2 Файл Seq.fasta

Файл содержит нуклеотидные последовательности LTR (Long Terminal Repeat) класса 5 в геноме человека, характеристики:

1. Число последовательностей - 1500
2. Средняя длина – 1200.

5.3 Файл Short.fasta

Его содержимое не несет явно выраженной смысловой нагрузки, а файл является сокращенной версией Pfam.fasta, сформированным с целью исследования производительности алгоритма Dialign P, так как он весьма требователен к объему памяти на узле, характеристики:

1. Число последовательностей - 100
2. Средняя длина – 390.

6. Анализ результатов

На рисунках 1, 2, 3 представлены результаты проведенного тестирования, можно выделить следующие наиболее интересные особенности представленных диаграмм:

1. Алгоритм Dialign P не присутствует на первых двух графиках, так как он выполняется только для сравнительно небольших объемов данных, но и на них он проигрывает по времени выполнения параллельной программы остальным рассматриваемым алгоритмам. Одинаковое время для одного, двух и четырех процессоров означает, что Dialign P не уложился в отведенные ему 15 минут. Надо заметить, что программа не предоставляет отчет о времени, затраченном на работу, поэтому самостоятельно были выбраны точки для замера времени, и в измеряемом отрезке не были учтены временные затраты на считывание входных данных и распределение их по процессорам, в отличие от остальных алгоритмов.

2. Алгоритм Muscle показывает хорошую масштабируемость на больших объемах данных, но на меньшем числе последовательностей не демонстрирует достаточное сокращение времени выполнения, а то и его прирост.

3. Малое изменение времени работы ClustalW-MPI на одном и двух процессорах обусловливается тем, что один из процессов является управляющим. Программа показывает лучшую масштабируемость, но временные показатели у неё хуже чем у Muscle.

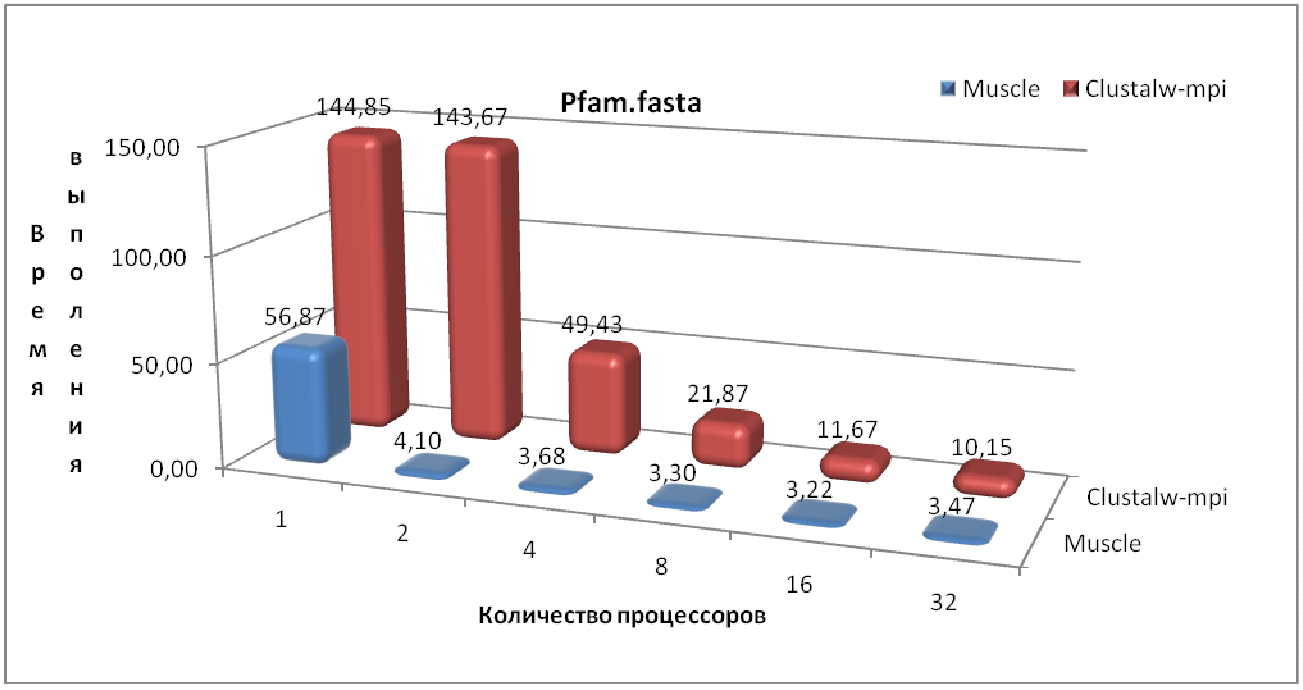

Рис. 1. Время выравнивания (в минутах) последовательностей из файла Pfam.fasta

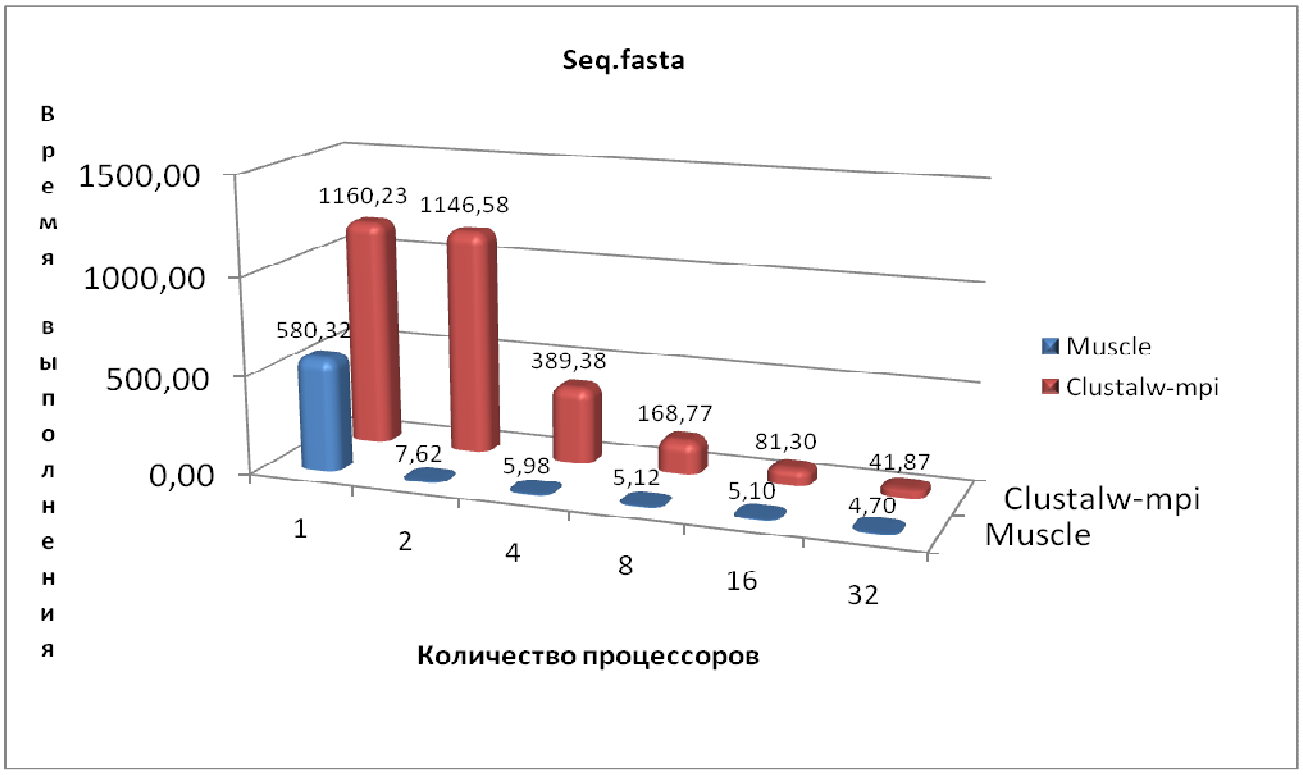

Рис. 2. Время выравнивания (в минутах) последовательностей из файла Seq.fasta

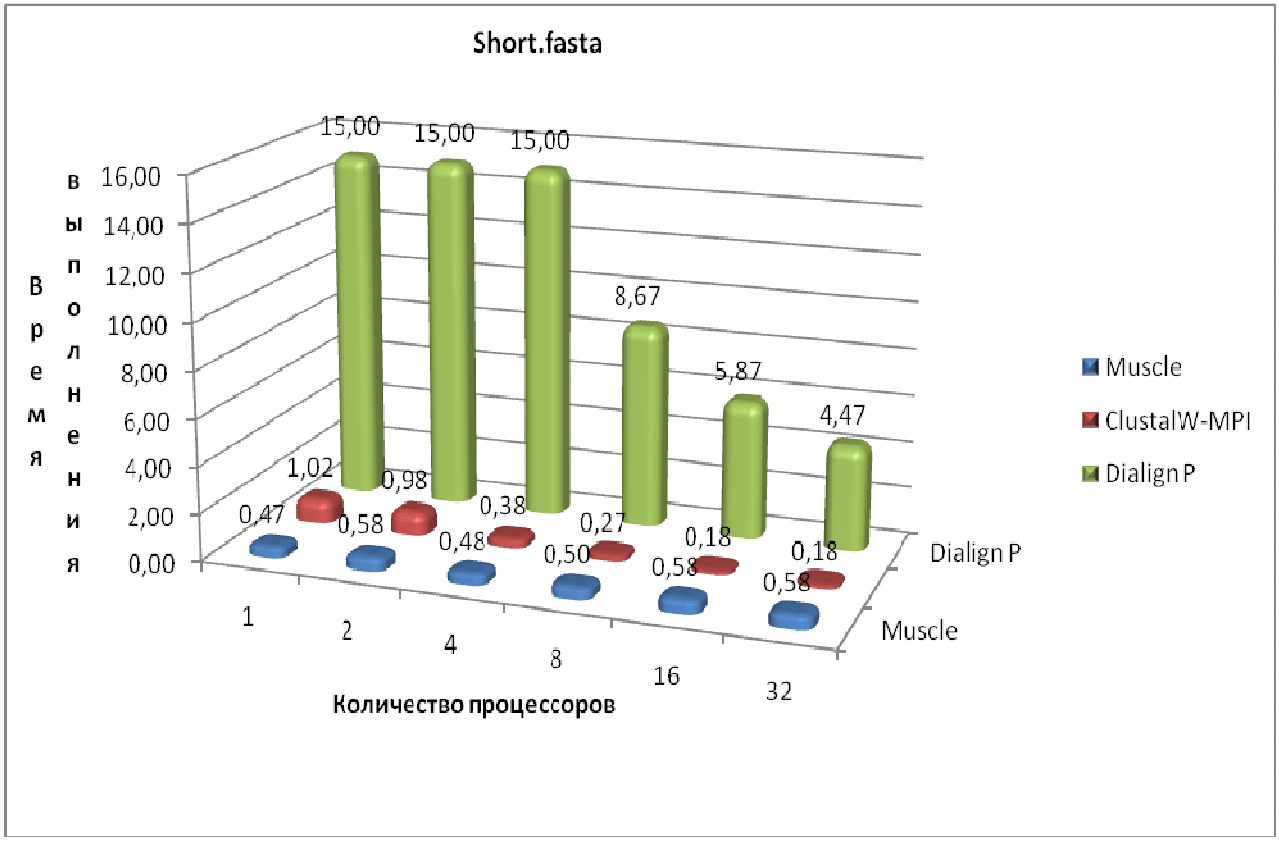

Рис. 3. Время выравнивания (в минутах) последовательностей из файла Short.fasta

7. Заключение

Полученные результаты позволяют говорить о том, что на сегодняшний день не существует доступных универсальных алгоритмов множественного выравнивания, одинаково быстро работающих и для малого числа коротких последовательностей, и для большого числа длинных. Почти все программы при построении парного выравнивания используют матрицу, размерности которой равняются длине выравниваемых участков, поэтому для последовательностей достаточно больших размеров требуется колоссальное количество памяти. Решение этой проблемы в рамках использования концепции динамического программирования не представляется очевидным. Таким образом, задача создания эффективного и универсального алгоритма множественного выравнивания является достаточно важной и нужной, особенно с учетом развития многопроцессорных систем (в частности, ввода в эксплуатацию многопроцессорного комплекса «Ломоносов» в МГУ).

Литература

1. Fursov, M. Y.; Oshchepkov, D. Y; Novikova, O. S. (2009). "UGENE: interactive computational schemes for genome analysis". *Proceedings of the Fifth Moscow International Congress on Biotechnology* 3: 14–15. ISBN 5-7237-0372-2.
2. Howard Hughes Medical Institute (2005, September 22). Gaining Ground In The Race Against Antibiotic Resistance. *ScienceDaily*. Retrieved January 13, 2011, URL: http://www.sciencedaily.com/releases/2005/09/050922021043.htm (дата обращения: 10.01.2011).
3. Robert F. Resistance is futile. URL: http://news.sciencemag.org/sciencenow/2007/03/28-03.html (дата обращения: 10.01.2011).
4. Mark Joffe, The Canadian Journal of Infectious Diseases & Medical Microbiology, 2006 Sep–Oct; 17(5): 285.
5. Kuo-Bin Li ClustalW-MPI: ClustalW analysis using distributed and parallel computing //Bioinformatics Vol. 19, No. 12, 2003, pp. 1585-1586. ISSN: 1460-2059 (Electronic),ISSN: 1367-4803 (Print).
6. Martin Schmollinger, Kay Nieselt, Michael Kaufmann and Burkhard Morgenstern DIALIGN P: Fast pair-wise and multiple sequence alignment using parallel processors // BMC Bioinformatics, 2004, 5:128, ISSN: 1471-2105 (Электронный).
7. Alexey N. Salnikov The modification of MUSCLE multiple sequence alignment algorithm for multiprocessors Proceedings of the 3-rd Moscow conference on computational molecular biology, Moscow, Russia, July 27-31 2007, pp. 270-271.
8. Pearson and Lipman. "Improved tools for biological sequence comparison", 1988. PNAS 85(8): 24444 – 2448.