«Сейфуллин оқулары – 12: Ғылым жолындағы жастар-болашақтың инновациялық әлеуеті» атты Республикалық ғылыми-теориялық конференция материалдары = Материалы Республиканской научно-теоретической конференции **«Сейфуллинские чтения-12:** Молодежь в науке - инновационный потенциал будущего" . – **2016**. – Т.1, ч.3 – С.283-287

РАЗРАБОТКА ИНФОМАЦИОННЫХ СИСТЕМ ДЛЯ АЛГОРИТМОВ ПО ОБРАБОТКЕ БИОПОСЛЕДОВАТЕЛЬНОСТЕЙ

Мазакова Б.М.

В современном мире существует множество классов задач, которые требуют большого количества, как вычислительных ресурсов, так и ресурсов памяти. Один из таких классов составляют задачи биоинформатики. В настоящее время под биоинформатикой понимают математические методы компьютерного анализа биологических последовательностей и разработку алгоритмов и программ предсказания пространственной структуры белков[2]. Использование современных, даже самых мощных однопроцессорных систем, не всегда позволяет не то что решить ряд из этих задач за приемлемое время, а даже достичь конечного решения. В качестве примера можно привести задачу создания молекулярных интерфейсов, заключающуюся в организации взаимодействия соединений, не взаимодействующих в природе, моделирование решения которой в 6 позициях (не очень много по меркам задачи) занимает более двух суток счета[2], или задачу генерирования белков с заданными свойствами, на сегодняшний день рекордная длина составляет 200-300 аминокислотных остатков, при том что, к примеру, в молекуле гемоглобина их содержится более 64000[3]. Единственный способ разрешения трудностей такого рода – это распараллеливание вычислений. В связи с этим для решения вычислительно-сложных задач биоинформатики применяются кластерные вычислительные системы Рассматриваемые задачи обработки биопоследовательностей: поиск гомологов нуклеотидных последовательностей, создание молекулярных интерфейсов, множественное выравнивание – имеют высокую временную сложность , что делает актуальным создание сервиса по их решению на вычислительных кластерах.

Необходимо отметить важность создания системы унифицированного запуска заданий посредством веб-интерфейса: поскольку конечными пользователями программного продукта предполагаются не специалисты в области организации распределенных вычислений, а биологи, которые могут быть не знакомы с деталями интерфейса системы управления заданиями в конкретной многопроцессорной системе, а также технологиями работы в окружении UNIX, на первый план выходит разработка простого и понятного для специалиста в молекулярной биологии веб-интерфейса для постановки задач на счет с отслеживанием её состояния и, при необходимости, прекращения её работы.

Биоинформатика сегодня подразумевает создание и совершенствование баз данных, алгоритмов, вычислительных и статистических методов и теории для решения практических и теоретических проблем, возникающих при управлении и анализе биологических данных.

Для удобства дальнейшего рассмотрения введем несколько определений.

Формализация задачи сравнения последовательностей: найти минимальное редакционное расстояние и набор преобразований, его реализующий.

Центральным понятием для данной задачи является выравнивание.

Выравнивание последовательностей – биоинформатический метод, основанный на размещении двух или более последовательностей мономеров ДНК, РНК или белков друг под другом таким образом, чтобы легко увидеть сходные участки в этих последовательностях. Сходство первичных структур двух молекул может отражать их функциональные, структурные или эволюционные взаимосвязи [2]. В большинстве представлений результата выравнивания, последовательности располагаются в строчках матрицы таким образом, что совпадающие элементы (нуклеотиды или аминокислоты) находятся один под другим (в одной колонке). «Разрывы» заменяются знаком «-», или ячейка остается пустой. Алгоритмы поиска применяются для поиска в больших базах данных последовательностей, схожих с некой заданной последовательностью по указанным критериям. Наиболее известные программы: BLAST и FASTA3x.

Глобальное выравнивание предполагает, что последовательности гомологичны по всей длине. В глобальное выравнивание включаются обе входные последовательности целиком.

Глобальное выравнивание Алгоритм Нидлмана-Вунша – это алгоритм для выполнения выравнивания двух последовательностей (будем называть их A и B), который используется в биоинформатике при построении выравниваний аминокислотных или нуклеотидных последовательностей. Алгоритм был предложен в 1970 году Солом Нидлманом и Кристианом Вуншем [3]. Алгоритм Нидлмана-Вунша является примером динамического программирования, и он оказался первым примером приложения динамического программирования к сравнению биологических последовательностей. Соответствие выровненных символов задается матрицей похожести. Здесь S (a, b) – похожесть символов a и b. Также используется линейный штраф за разрыв, называемый здесь d. Например, если матрица похожести задается таблицей:

	–	A	G	C	T
A		10	-1	-3	-4
G		-1	7	-5	-3
C		-3	-5	9	0
T		-4	-3	0	8,

то выравнивание

A G A C T A G T T A C (3.1)
C G A − − − G A C G T (3.2)

со штрафом за разрыв d = −5 будет иметь следующую оценку:

$$S(A,C) + S(G,G) + S(A,A) + 3\times d + S(G,G) + S(T,A) + S(T,C) + + S(A,G) + S(C,T) =$$
$$= -3 + 7 + 10 - (3\times 5) + 7 - 4 + 0 - 1 + 0 = 1.$$

Для нахождения выравнивания с наивысшей оценкой назначается матрица F, содержащая столько же строк, сколько символов в последовательности A, и столько же столбцов, сколько символов в последовательности B. Запись в строке i и столбце j обозначается далее как Fij. Таким образом, если мы выравниваем последовательности размеров n и m, то количество требуемой памяти будет O(nm). В процессе работы алгоритма величина Fij будет принимать значения оптимальной оценки для выравнивания первых i = 0, ..., n символов в A и первых j = 0, ..., m символов в B. Тогда принцип оптимальности Беллмана может быть сформулирован следующим образом:

Базис: $F0j = d\cdot j$ (3.4), $Fi0 = d\cdot i$ (3.5)

Рекурсия, основанная на принципе оптимальности:

$$F_{ij} = \max(F_{i-1, j-1} + S(A_i,B_j), F_{i, j-1} + d, F_{i-1, j} + d).$$

Когда матрица F рассчитана, её элемент Fij дает максимальную оценку среди всех возможных выравниваний. Для вычисления самого выравнивания, которое получило такую оценку, нужно начать с правой нижней клетки и сравнивать значения в ней с тремя возможными источниками (соответствие, вставка или делеция), чтобы увидеть, откуда оно появилось. В случае соответствия Ai и Bj выровнены, в случае делеции Ai выровнено с разрывом, а в случае вставки с разрывом выровнено уже Bj. (В общем случае может быть более одного варианта с одинаковым значением, которые приведут к альтернативным оптимальным выравниваниям.)

Алгоритм Смита-Ватермана предназначен для получения локального выравнивания последовательностей, то есть для выявления сходных участков двух нуклеотидных или белковых последовательностей. В отличие от алгоритма Нидлмана-Вунша, который осуществляет выравнивание последовательностей по всей длине, алгоритм Смита-Ватермана сравнивает отрезки всех возможных длин и оптимизирует меру сходства по всем отрезкам и всем выравниваниям этих отрезков. Алгоритм был предложен Т.Ф. Смитом и М. Ватерманом в 1981 [3]. Подобно алгоритму Нидлмана-Вунша, алгоритм Смита-Ватермана использует принцип динамического программирования. Он гарантирует нахождение оптимального, относительно используемой им меры оценки качества, локального выравнивания. Эта мера оценки – так называемый вес или счёт (Score) выравнивания, предусматривающий использование матрицы замен и штрафов за «гэпы» (то есть вставки и делеции). Так, в примере (3.1) оптимальным будет выравнивание:

GA
GA,

а соответствующая ему оценка

$$S(G,G) + S(A,A) = 7 + 10 = 17.$$

Как и в алгоритме Нидлмана-Вунша, здесь строится матрица F по правилу (3.3), где рекурсия имеет вид:

$$F_{ij} = \max(F_{i-1, j-1} + S(A_i,B_j), F_{i, j-1} + d, F_{i-1, j} + d, 0).$$

Точка конца пути, с которой начинается построение выравнивания, определяется так:

$$(i_{max},j_{max}) = \operatorname{argmax} F_{ij}.$$

Процесс построения выравнивания заканчивается, когда у текущего элемента Fij не останется положительных предшественников.

Псевдо-код для данного алгоритма будет выглядеть следующим образом:

```
AlignmentA = ""
```

```
AlignmentB = ""
i = length(A)
j = length(B)
while (i > 0 and j > 0){
  Score = F(i,j)
  ScoreDiag = F(i - 1, j - 1)
          ScoreUp = F(i, j — 1)
ScoreLeft = F(i - 1, j)
  if (Score == ScoreDiag + S(Ai, Bj))  {
      AlignmentA = A(i) + AlignmentA
      AlignmentB = B(j) + AlignmentB
      i = i - 1
      j = j - 1
  }
  else if (Score == ScoreLeft + d)  {
      AlignmentA = A(i) + AlignmentA
      AlignmentB = "-" + AlignmentB
      i = i - 1
  }
  else if (Score == ScoreUp + d)  {
      AlignmentA = "-" + AlignmentA
      AlignmentB = B(j) + AlignmentB
      j = j - 1
}
}
while (i > 0){
  AlignmentA = A(i) + AlignmentA
  AlignmentB = "-" + AlignmentB
  i = i - 1
}
while (j > 0){
  AlignmentA = "-" + AlignmentA
  AlignmentB = B(j) + AlignmentB
  j = j - 1
}
```

Список литературы

1. Дурбин Р., Эдди Ш., Крог А., Митчисон Г. \emph{Анализ биологических последовательностей.}~--- М.~-- Ижевск: НИЦ <<Регулярная и хаотичная динамика>>, 2006.~--- 480~с.
2. Бородовский М., Екишева С. \emph{Задачи и решения по анализу биологических последовательностей.}~--- М.~-- Ижевск: НИЦ <<Регулярная и хаотичная динамика>>, 2008.~--- 420~с.

3. BLAST: Basic Local Alignment Search Tool [электронный ресурс] http://blast.ncbi.nlm.nih.gov/Blast.cgi
4. The NCBI C++ Toolkit Book [электронный ресурс], дата обращения 20.03.2013

http://www.ncbi.nlm.nih.gov/books/NBK7160/

T.K. Attwood, A. Gisel, N-E. Eriksson and E. Bongcam-Rudloff (2011). Concepts, Historical Milestones and the Central Place of Bioinformatics in Modern Biology: A European Perspective, Bioinformatics - Trends and Methodologies, Dr. Mahmood