Резюме

В роботі описується пакет програм GenBee, призначений для аналізу біологічних послідовностей. Пакет орієнтований головним чином на задачі теоретичної молекулярної біології і поєднує зручний для користування інтерфейс з розвинутими сучасними алгоритмами (в тому числі оригінальні). Він написаний на мові Сі і пригодний для роботи на комп'ютерах типу IBM PC.

Summary

The package of programs GenBee intended to analyze nucleotide and amino acid sequences is described. This package combines high-level algorithms (including original ones) suitable for advanced theoretical studies with flexibility and user-friendly service typical of commercially available packages. The package is designed for IBM-compatible personal computers. Accordingly it will be useful both for researches and students with practically no background in computer methods, and for theoreticians.

СПИСОК ЛИТЕРАТУРЫ

1. *Кунин Е. В., Чумаков К. М., Горбаленя А. Е.* Метод поиска структурных мотивов в аминокислотных последовательностях. Программа Site пакета GenBee // Биополимеры и клетка.— 1990.— **6**, № 6.— С. 42—48.
2. *Trifonov E. N.* Translation framing code and frame-monitoring mechanism as suggested by the analysis of mRNA and 16S-rRNA nucleotide sequences // J. Mol. Biol.— 1987.— **194**, N 2.— P. 643—652.
3. *Леонтович А. М., Бродский Л. И., Горбаленя А. Е.* Построение полной карты локального сходства двух полимеров. Программа DotHelix пакета GenBee // Биополимеры и клетка.— 1990.— **6**, № 6.— С. 14—21.
4. *Бродский Л. И., Драчев А. Л., Леонтович А. М.* Новый метод множественного выравнивания последовательностей биополимеров. Программа H-Align пакета GenBee // Там же.— 1991.— **7**, № 1.— С. 14—22.
5. *Lipman D. J., Pearson W. R.* Rapid and sensitive protein similarity searches // Science.— 1985.— **227**, N 1.— P. 1435—1441.
6. *Hartigan J. A.* Clastering algorithms.— New York : John Wiley and Sons, 1975.— 256 p.
7. *Чумаков К. М., Юшманов С. В.* Принцип максимального топологического подобия в молекулярной систематике // Молекуляр. генетика, микробиология, вирусология.— 1988.— **3**, № 1.— С. 3—9.
8. *Эволюция* РНК-зависимых РНК-полимераз позитивных рибовирусов : сравнение филогенетических деревьев, построенных различными способами / Е. В. Кунин, К. М. Чумаков, С. В. Юшманов, А. Е. Горбаленя // Там же.— С. 16—19.
9. *Gibrat J.-F., Garnier J., Robson B.* Further developments of protein secondary structure prediction using information theory // J. Mol. Biol.— 1987.— **198**, N 4.— P. 425—443.
10. *Гультяев А. П., Монаков Ю. Н.* Метод построения вторичной структуры РНК на основе принципов самоорганизации // Биополимеры и клетка.— 1991.— **7**, № 1.— С. 31—36.

Науч.-произв. кооператив «Комби», Москва
Межфакультет. н.-и. лаб. им. А. Н. Белозерского, МГУ

Получено 28.06.90

УДК 577.112

Л. И. Бродский, А. Л. Драчев, А. М. Леонтович

НОВЫЙ МЕТОД МНОЖЕСТВЕННОГО ВЫРАВНИВАНИЯ ПОСЛЕДОВАТЕЛЬНОСТЕЙ БИОПОЛИМЕРОВ (ПРОГРАММА H-Align ПАКЕТА GenBee)

В работе предложен новый алгоритм множественного выравнивания биологических последовательностей. В этом алгоритме вначале на основе метода DotHelix строятся консенсусные участки в данном наборе последовательностей разной толщины и раз-

ной степени сходства, а затем из этих консенсусов составляются цепочки, согласованные с порядком букв в последовательностях, и такие цепочки являются каркасами выравниваний. На основе алгоритма на языке Си написана программа H-Align из пакета GenBee. Рассмотрен модельный пример, иллюстрирующий эффективность предложенного алгоритма.

Введение. Как известно, процедура выравнивания широко используется при анализе первичных структур биополимеров. Такая процедура применяется, например, при поиске функционально важных участков белков или молекул ДНК/РНК, при построении филогенетических деревьев и т. д.

Алгоритмы выравнивания, в частности множественного, можно разбить на две основные группы: одноэтапные, оптимизирующие сразу все выравнивание, и двухэтапные. В двухэтапной процедуре на первом шаге находится достаточно большое количество наборов сходных между собой фрагментов из разных последовательностей, а на втором — из этих наборов составляются непротиворечивые цепочки, лучшие из которых и являются альтернативными выравниваниями.

Алгоритмы первой группы, как правило, базируются на процедурах динамического программирования (типа Нидельмана — Вунша [1, 2]) и требуют назначения величины штрафа за введение пробелов в выравненные последовательности (соответствующих делециям или вставкам остатков). Результат выравнивания существенно зависит от величины штрафа, но в то же время нет никаких сколько-нибудь серьезных оснований для выбора этой величины. Другой недостаток одноэтапных алгоритмов — это то, что обычно в них множественное выравнивание делается последовательно: сначала выравниваются две самые близкие последовательности, затем эта пара как целое выравнивается с третьей, затем тройка как целое выравнивается с четвертой и т. д. Если на одном из шагов выравнивание в каком-то месте было сделано неудачно, то в дальнейшем в этом месте оно будет все более искажаться.

Основной недостаток существующих методов второго типа — это грубость процедуры поиска «консенсусов» наборов сходных между собой фрагментов последовательностей. Некоторые из реально существующих консенсусов могут быть пропущены, и вместе с тем границы найденных консенсусов определяются недостаточно точно.

В представляемом ниже алгоритме, относящемся ко второй группе методов, консенсусные участки, по нашему мнению, ищутся более полно и точно, чем обычно. Наш метод исходит из построения попарных карт локального сходства и основан на алгоритме DotHelix [5].

Методы. Пусть имеется несколько последовательностей биополимеров (например, белков). Результат их выравнивания представляет собой набор последовательностей одинаковой длины. Каждая последовательность из набора получается за счет внесения пробелов в соответствующую ей исходную последовательность; эти пробелы отвечают либо делециям букв в этой последовательности, либо вставкам букв, произошедшим в каких-то других последовательностях исходного набора. При этом на некоторых участках набора выравненных последовательностей между всеми или некоторыми его слоями существует сходство. Такие участки будем называть «консенсусами». Степень сходства между слоями консенсуса характеризуется его мощностью, которая вычисляется как нормированная сумма весов за попарные замены остатков (букв), стоящих в одном столбце консенсуса:

$$A = \left[\sum_{l=1}^{L} \sum_{1<r_1<r_2<k} (\rho_{b_{l,r_1},b_{l,r_2}} - m)\right] / \sqrt{D \cdot L}, \qquad (1)$$

где A — мощность рассматриваемого консенсуса; L — длина консенсуса, т. е. общая длина входящих в него фрагментов; k — толщина консенсуса, т. е. число входящих в него слоев; $b_{1,r}$, $b_{2,r}$, ..., $b_{L,r}$ — буквы (остатки), составляющие r-й слой консенсуса; $\rho_{b,c}$ — вес за замену буквы b на

букву c (для белков эти веса обычно берутся из матрицы Дэйхофф [6]);

$$m = M(\rho) = \sum_{b,c} \rho_{b,c} \cdot p_b \cdot p_c — \quad (2)$$

среднее значение веса за замену букв (суммирование в (2) производится по всевозможным буквам алфавита; для белков их 20);

$$D = k \cdot (k-1)/2 \cdot D(\rho) + k \cdot (k-1) \cdot (k-2) \cdot \mathrm{cov}(\rho_{12}, \rho_{13}) =$$

$$= k \cdot (k-1)/2 \cdot \sum_{b,c} (\rho_{b,c} - m)^2 \cdot p_b \cdot p_c + k \cdot (k-1) \cdot (k-2) \times$$

$$\times \sum_{b,c_1,c_2} (\rho_{b,c_1} - m) \cdot (\rho_{b,c_2} - m) \cdot p_b \cdot p_{c_1} \cdot p_{c_2};$$

p_b — частота буквы b во всем наборе последовательностей.

Для случая консенсуса толщины два обоснование этого определения приведено в [5]. Следует сказать, что формула (1) несколько огрублена по сравнению с аналогичной формулой в [5]: в формуле (1) частоты вхождения букв p_b предполагаются одинаковыми для всех выравниваемых последовательностей. Это сделано для ускорения времени счета; такое огрубление слабо влияет на точность оценок мощности консенсусов.

Заметим, что для поиска функционально важных участков знание консенсусов, по-видимому, даже более важно для биолога, чем само выравнивание. Поэтому процедура поиска консенсусов имеет самостоятельную ценность. В сущности, таким образом мы получаем многомерный аналог карты локальных сходств.

Из вышеизложенного вытекает, что при поиске правильного выравнивания естественно сначала находить все консенсусные участки без пробелов, а затем цепочку из таких участков достроить до полного выравнивания последовательностей, вставляя необходимое число пробелов в зонах между этими участками. При этом из биологических соображений ясно, что в правильном выравнивании родственных последовательностей, с одной стороны, консенсусы должны быть по возможности более мощными и, с другой — пробелов должно быть немного.

В соответствии с этим в нашем алгоритме выделяются два этапа:

во-первых, построение всех консенсусов (заметим, что в консенсус необязательно входят все слои);

во-вторых, перебор всех правильных с точки зрения порядка букв в последовательностях расположений таких консенсусов относительно друг друга и выбор среди них наилучших по двум критериям: максимизации суммарной мощности входящих консенсусов и минимизации общего числа пробелов.

Начнем с первого этапа. Для нахождения всех возможных в данном наборе последовательностей консенсусов следовало бы просмотреть все «регистры сравнения» (возможные сдвиги исходных последовательностей относительно друг друга) и для каждого из них искать совокупность непересекающихся достаточно мощных консенсусов, что можно сделать, например, с помощью программы DotHelix [5]. Однако такая процедура требует даже при сравнительно небольшом числе последовательностей ($t \geqslant 3$) весьма большого времени счета, поскольку велико число разных регистров сравнения — оно порядка n^{t-1}, где n — средняя длина последовательности. Чтобы избежать полного перебора всех регистров сравнения, предлагается использовать последовательную процедуру нахождения наиболее мощных консенсусов. Эта процедура основана на том обстоятельстве, что в мощном консенсусе каждая или почти каждая пара слоев значимо похожи.

Соответствующий алгоритм заключается в следующем. Для каждой пары последовательностей из набора методом DotHelix находим все сходные фрагменты этих последовательностей на сравнительно низком уровне значимости. Тем самым будут найдены все достаточно мощные

консенсусы толщины два. Далее происходит постепенное утолщение консенсусов: консенсусы толщины три получаются из консенсусов толщины два, консенсусы толщины четыре получаются из консенсусов толщины три и т. д. за счет поочередного подцепления к каждому из них подходящих фрагментов очередной последовательности. Таким образом, каждый консенсус как жесткое целое сдвигается вдоль всех незадействованных в этом консенсусе последовательностей и присоединяет к себе в качестве дополнительного слоя подходящие фрагменты.

Опишем более подробно процедуру присоединения нового слоя. Ее проводили методом DotHelix [5], но по несколько другим правилам вычисления мощности (которую мы будем называть «мощностью подцепления»). Консенсус, продолженный в обе стороны на несколько позиций, сдвигается относительно сравниваемой последовательности, и для каждого сдвига формируется диагональ карты локального сходства консенсуса и этой последовательности (для сравнения см. [5]); числа a_{ij}, заполняющие карту локального сходства, являются суммами весов за замены букв из i-го столбца консенсуса на букву из j-й позиции сравниваемой последовательности:

$$a_{ij} = \sum_{r=1}^{k} \rho_{b_{ir}, c_j},$$

где k — число слоев консенсуса, b_{ir} — буква, стоящая в r-м слое i-м столбце консенсуса, c_j — буква, стоящая в j-й позиции сравниваемой с консенсусом последовательности, $\rho_{b,c}$ — вес за замену буквы b на букву c. Как было указано выше, формула для мощности подцепления несколько отличается от приведенной выше формулы (1) для мощности консенсуса. Это связано с тем, что слои консенсуса нельзя рассматривать как независимые бернуллиевские последовательности (например, вполне реальная ситуация, когда все слои консенсуса в точности совпадают между собой). Поэтому консенсус следует рассматривать не как набор случайных, а как набор фиксированных последовательностей, состоящих из букв $b_{i,r}$. В то же время вполне естественно представлять сравниваемую последовательность бернуллиевской и независимой от слоев консенсуса. В результате вместо формулы (1) мощность подцепления следовало бы вычислять так:

$$A_{\text{con}} = \left[\sum_{l=0}^{L-1} \sum_{r=1}^{k} (\rho_{b_{i+l,r}, c_{j+l}} - M_{l,r})\right] / \sqrt{D},$$

где L — длина отрезка диагонали, мощность которого мы вычисляем;

$$M_{l,r} = \sum_{c} \rho_{b_{i+l,r}, c} \cdot p_c;$$

$$D = \sum_{l=0}^{L-1} \left\{ \sum_{r=1}^{k} D(\rho_{b_{i+l,r}}) + 2 \sum_{1 \le r_1 < r_2 \le k} \operatorname{cov}(\rho_{b_{i+l,r_1}} \cdot \rho_{b_{i+l,r_2}}) \right\} =$$

$$= \sum_{l=0}^{L-1} \left\{ \sum_{r=1}^{k} \left[\sum_{c} (\rho_{b_{i+l,r}, c} - M_{l,r})^2 \cdot p_c \right] + 2 \sum_{1 \le r_1 < r_2 \le k} \left[\sum_{c} (\rho_{b_{i+l,r_1}, c} - \right.\right.$$

$$\left.\left. - M_{l,r_1}) \cdot (\rho_{b_{i+l,r_2}, c} - M_{l,r_2}) \cdot p_c \right] \right\};$$

p_c — частота буквы c.

К сожалению, вычислять по этой громоздкой формуле приходится очень часто, а это самым пагубным образом сказывается на скорости счета. Поэтому в программе используется оценка мощности подцепления по приближенной формуле. Эта оценка исходит из следующих предположений (которые, безусловно, являются лишь приблизительными): во-первых, что частоты букв в консенсусе такие же, как во всех последовательностях в совокупности; во-вторых, зависимость между слоями консенсуса выражается только в увеличении количества совпадений

букв в его столбцах. Первое предположение приводит к тому, что M_{rl} тождественно равно m, в силу же второго предположения имеет место формула

$$D = L \cdot \{(k + n_{eq}) \cdot \sum_{b,c} (\rho_{b,c} - m)^2 \cdot p_b \cdot p_c + [k \cdot (k-1) - n_{eq}] \times$$

$$\times \sum_{b_1, b_2, c} (\rho_{b_1,c} - m) \cdot (\rho_{b_2,c} - m) \cdot p_{b_1} \cdot p_{b_2} \cdot p_c.$$

Здесь n_{eq} характеризует степень согласованности в столбцах консенсуса (по сравнению с той, которая была бы для случая бернуллиевских и независимых между собой слоев), переведенную в ожидаемое число совпадений букв в столбце (точнее говоря, увеличение этого числа совпадений по сравнению со случаем независимых слоев).

Мы вычисляем эту величину n_{eq} исходя из мощности консенсуса, предполагая, что такое значение мощности и соответствующее значение нормированной суммы весов (стоящей в числителе формулы (1)) происходит только за счет точных совпадений букв (конечно, и это предположение является лишь приближенным); из этого предположения легко следует, что

$$n_{eq} = S/(m_{eq} - m) \cdot L,$$

где $m_{eq} = \sum_b \rho_{b,b} \cdot p_b$ — среднее значение весов $\rho_{b,b}$, m вычисляется по формуле (2), S — нормированная сумма весов для консенсуса, получаемая из значения его мощности (см. формулу (1)).

При процедуре присоединения нового слоя мы отбираем только те новые консенсусы, мощность подцепления для которых выше некоторого порога.

В результате всей этой работы мы получим некоторую совокупность консенсусов разной толщины; каждому консенсусу соответствует его мощность, рассчитанная уже по формуле (1). Неполная толщина консенсуса в окончательном наборе означает, что входящие в него последовательности заведомо имеют хорошее соответствие между собой в этом месте; что же касается пропущенных в этом консенсусе последовательностей, то хорошего соответствия со слоями консенсуса там нет.

Промежуточное хранение консенсусов (для экономии памяти и времени счета) осуществляется по принципу «стека» так, что сохраняется определенное число наилучших по мощности. При построении очередного консенсуса набор переупорядочивается и худший выталкивается из стека.

Говоря метафорически, в предложенном нами способе поиска консенсусов из рассмотрения всевозможных пар последовательностей намечаются потенциальные зоны сходства, а потом некоторые из них все более проявляются за счет сходств этого участка с фрагментами из других последовательностей, а другие стираются ввиду отсутствия дополнительных сходств.

При вышеописанном способе получения консенсусов довольно большое количество их будет шумом, мешающим последующему построению выравнивания из этих консенсусов (увеличивается перебор). Это обстоятельство в основном связано с нашей нулевой гипотезой о независимости выравниваемого набора последовательностей, хотя само желание их выравнить предполагает обратное. Очевидно, что для зависимых (в сторону возможности выравнивания) последовательностей среднее значение $M(\rho)$ весов за замену букв увеличивается по сравнению с формулой (2). В нашей программе имеется возможность учесть это обстоятельство, увеличивая величину m. Эксперименты показывают, что такая процедура отфильтровывает значительное количество шумовых консенсусов. Однако вопрос об оптимальном наборе величины m нуждается в дальнейшем изучении.

Перейдем теперь к описанию второго этапа — построению наилучших выравниваний из консенсусов.

После выделения полного набора достаточно мощных консенсусов нужно искать все возможные комбинации их состыковки, причем для составления полного выравнивания, кроме пробелов, придется использовать и «неконсенсусные» участки. Поскольку трудно ожидать, что в биологических последовательностях существует значимое «отрицательное» сходство с отрицательной мощностью, т. е. маловероятное с позиций случайности несходство, можно думать, что порядок расположения букв и пробелов в неконсенсусном участке не очень существен. В нашем алгоритме в неконсенсусных участках по каждому слою сначала идут буквы, а затем добавляется необходимое число пробелов.

В случае наложения двух консенсусов из разных регистров (т. е. в случае, когда эти консенсусы содержат хотя бы один общий фрагмент какой-либо последовательности, входящей в оба эти консенсуса) с помощью вставки необходимого числа пробелов можно перейти от одного консенсуса к другому. Этот переход осуществляется у нас в том месте, которое обеспечивает наибольшее значение суммарной мощности.

Как было уже сказано, выбор выравнивания состоит в построении согласованной по порядку букв в последовательностях цепочки консенсусов. При этом для уменьшения перебора таких цепочек все найденные консенсусы разделяются на несколько групп по мощности (всего возможно не более десяти групп). После этого вначале составляются все возможные цепочки из сравнительно небольшой группы самых мощных консенсусов, затем в эти цепочки всеми возможными способами вставляются консенсусы из следующей группы по мощности и т. д. до исчерпания всех групп.

В итоге мы получим набор выравниваний с максимально возможным включением в каждый из построенных на первом этапе консенсусов (максимальность понимается в том смысле, что цепочки насыщены — больше добавить туда консенсусов нельзя). Мерой качества выравнивания служит нормированная сумма попарных весов за замены букв из всех столбцов консенсусов соответствующей цепочки (сравни с формулой (1): в качестве длины L берется общая длина выравненных последовательностей). Видно, что при таком определении хорошее качество имеют такие выравнивания, в которые по возможности входят мощные консенсусы при небольшом числе вставленных пробелов.

После получения выравнивания пользователь имеет возможность вручную поправить его, выкинув из исходного набора тот консенсус, включение которого в цепочку заставляет сильно увеличить число вставленных пробелов и затем с новым набором консенсусов вновь проделать процедуру составления цепочек.

Окончательный выбор наилучшего выравнивания можно осуществить разными способами. Например, такое наилучшее из набора найденных с разным числом пробелов может выбирать пользователь. Можно также использовать то обстоятельство, что для того значения числа пробелов, при котором имеет место «истинное» выравнивание (восстанавливающее предковую последовательность, если таковая существует), должно резко подскочить нормированное значение суммарной мощности консенсусов.

Данный алгоритм реализован на языке Си для компьютеров типа IBM PC в виде программы H-Align из пакета программ для анализа последовательностей биополимеров GenBee, разработанного в НПК «Комби».

Результаты и обсуждение. В качестве проверки работоспособности алгоритма мы исследовали следующий модельный пример. Были взяты четыре последовательности цитохромов С из разных животных — шимпанзе, собаки, индейки и утки (последовательности — «прародители»). Они очень похожи и состоят из 104 аминокислотных остатков. Далее их

«мутировали»: около 45 % букв было заменено дефисами — символами «—». На следующем шаге все дефисы были убраны. В результате получились более короткие последовательности с длинами 43, 53, 66 и 75 букв. Эти последовательности и взяты как исходные для выравнивания. Схема модели приведена на рис. 1.

Результаты работы по программе H-Align представлены на рис. 2, где изображены два набора выравненных последовательностей, полученные с использованием разных матриц весов за замены остатков: матрицы Дэйхофф [6] и единичной матрицы (в ней вес за совпадение букв

Рис. 1. Конструирование исходных для выравнивания последовательностей
Fig. 1. Constructing of the source sequences for an alignment

равен единице, а за несовпадение — нулю). В обоих изображенных наборах выравненных последовательностей большими буквами показаны места, входящие в консенсусы, а маленькими — в межконсенсусные участки.

На рис. 2, кроме результатов выравнивания по программе H-Align, приведено также «истинное» выравнивание. Оно получается из образованных нами последовательностей с дефисами (второй набор последовательностей — на рис. 1) выбрасыванием столбцов, состоящих из одних дефисов.

Сравнение трех выравниваний показывает, что все они достаточно хорошо соответствуют друг другу, однако выравнивание с использованием единичной матрицы более похоже на «истинное». Это обстоятельство объясняется способом построения исходных последовательностей, в результате чего практически все соответствующие друг другу буквы «истинного» выравнивания совпадают между собой. Поэтому единичная матрица весов, стремящаяся выделить консенсусы с совпадением букв, больше подходит для выравнивания таких последовательностей, чем матрица Дэйхофф.

Кроме того, отметим, что в межконсенсусных участках места расположения дефисов указываются не вполне точно. Это объясняется тем,

Выравненные по матрице Дэйхофф последовательности

```
  1: kg---KKIF----------GKHKTGPNLH-------------KNKGIIWGDTLYLENpky
  1: gdvkgKKIFVQKtv----------EKNLHglfGRKtpgfsy-KNKGITWGE---------
  1: gdkg-----VQKCSQTVekGKHKTGPNIH---GRKtgegfs-KNKGITWGDTLYLEN---
  1: -----KKIFVQKCSQTV----EKGGPNIHg--GRKtgqaegtdaNGITGEDTEYLEN---

 61: IPGTKMIfra--------DLIAYLKka--
 61: -------------------LIA-------
 61: IPGTKMI--GIKKKservDLIAYLKdats
 61: IPGTKMIfaGIKKKsatak----------
```

Выравненные по единичной матрице последовательности

```
  1: kg---KKIF----------GKHKTGPNLH-------------KNKGIIWGDTLYLENpky
  1: gdvkgKKIFVQK---TVEk-------NLHGlfGRKTpgfsy-KNKGITWGe---------
  1: gdkg-----VQKCSQTVEkGKHKTGPNLH---GRKTGegfs-KNKGITWGDTLYLEN---
  1: -----KKIFVQKCSQTVE----KGGPNLHG--GRKTGqaegtdanGITgeDTEYLEN---

 61: IPGTKMIFra--------DLIAYLKka--
 61: -------------------LIA-------
 61: IPGTKMI--GIKKKServDLIAYLKdats
 61: IPGTKMIFaGIKKKSatak----------
```

"Истинное" выравнивание

```
  1: ---KGKKIF----------GKHKTGPNLH-------------------KNKGIIWG-DT
  1: GDVKGKKIFVQK---TVEK--------NLHGLFGRKT---PGFSY----KNKGITWGE--
  1: GD-KG----VQKCSQTVEK-GKHKTGPNLH---GRKTG--EGFS-----KNKGITWG-DT
  1: -----KKIFVQKCSQTVEKG-----GPNLHG--GRKTGQAEG---TDAN---GIT-GEDT

 61: L--YLENPK-YIPGTKMIF--------RADLIAYLKKA---
 61: ----------------------------LIA--------
 61: L--YLEN----IPGTKMI--GIKKKSERVDLIAYLKDATS-
 61: --EYLEN----IPGTKMIFAGIKKKS-----------ATAK
```

Рис. 2. Результаты выравнивания по программе H-Align исходных последовательностей с использованием разных весовых матриц (Дэйхофф и единичной), а также «истинное» выравнивание

Fig. 2. H-Align program output alignmet of the source sequences with two different weight matrices (Dayhoff and identity). The «true» alignment is represented also

что наш алгоритм действует по правилу сдвига всех таких мест в конец соответствующего участка. Однако на таких участках искать какое-либо соответствие невозможно (по крайней мере, без использования биологических соображений), поскольку наблюдаемое на них сходство между буквами находится на уровне случайного.

Резюме

У роботі запропоновано новий алгоритм множинного вирівнювання біологічних послідовностей. В ньому спочатку на основі методу DotHelix будуються консенсусні ділянки в даному наборі послідовностей різної товщини і ступеня схожості, а потім із цих консенсусів складаються ланцюжки, погоджені з порядком букв в послідовностях, і такі ланцюжки є каркасами вирівнювання. На основі алгоритму на мові Сі написана програма H-Align з пакету GenBee. Розглянутий модельний приклад ілюструє ефективність запропонованого алгоритму.

Summary

Generalization of the multiple alignment is central to the entire field of biological sequence analysis. The algorithm of alignment by program H-align incorporated in GenBee package is a result of development of the local similarity search principle. It has two stages:

1) generalization of all the conservative regions (they cannot be present in all the aligning sequences).

2) optimal arrangement of these regions using two criteria — maximization of the total power of the conservative regions and minimization of the total number of spaces.

This algorithm has at least two advantages over traditional algorithms (such as Needleman-Wunsch's one) : no penalty for insertion / deletion; not subsequent pair aligning procedure. The efficiency of the algorithm is shown at model example.

СПИСОК ЛИТЕРАТУРЫ

1. *Needleman S. B., Wunch C. D.* A general method applicable to the search for similarities in the amino acid sequence of two proteins // J. Mol. Biol.— 1970.— **48**, N 2.— P. 443—453.
2. *Gotoh O.* Alignment of three biological sequences with an efficient traseback procedure // J. Theor. Biol.— 1986.— **121**, N 1.— P. 327—337.
3. *Sobel E., Marttinez H. M.* A multiple sequence alignment program // Nucl. Acids Res.— 1986.— **14**, N 2.— P. 363—374.
4. *Bacon D. J., Anderson W. J.* Multiple sequence alignment // J. Mol. Biol.— 1986.— **191**, N 1.— P. 153—161.
5. *Леонтович А. М., Бродский Л. И., Горбаленя А. Е.* Построение полной карты локального сходства двух полимеров (программа DotHelix пакета GenBee // Биополимеры и клетка.— 1990.— **6**, № 6.— С. 14—21.
6. *Dayhoff M. O., Barker W. C., Hunt L. T.* Establishing homologies in protein sequences // Meth. Enzymol.— 1983.— **91**.— P. 524—545.

Науч.-произв. кооператив «Комби», Москва
Межфакультет. н.-и. лаб. им. А. Н. Белозерского, МГУ

Получено 28.06.90

УДК 576.315.42

В. А. Шепелев

АЛГОРИТМ УСКОРЕННОГО ПОСТРОЕНИЯ ТОЧЕЧНЫХ МАТРИЦ ГОМОЛОГИИ

Метод анализа гомологичных участков с помощью точечных матриц гомологии заключается в нахождении и отображении на прямоугольной матрице общих для двух последовательностей слов, в которых совпадает определенное количество букв. Предложен алгоритм ускоренного построения таких матриц с различными параметрами фильтрации.

Введение. Метод исследования гомологичных участков с использованием точечных матриц гомологии [1] состоит в том, чтобы на прямоугольной матрице найти общие для двух последовательностей слова, то есть подпоследовательности длиной W, в которых совпадает не менее M букв. Параметры W и M называются параметрами фильтрации, а величина W — размером окна. Такое построение обладает большой наглядностью, так как гомологичные участки выявляются в виде диагональных линий. При $W=M=1$ матрица для двух случайных последовательностей оказывается забитой случайными (с вероятностью около 1/4) точками, отмечающими совпадение отдельных букв. Вообще число точек на матрице для двух случайных последовательностей можно оценить по формуле

$$Z = P \cdot L_1 \cdot L_2,$$

где

$$P = \sum_{k=M}^{W} C_w^k p^k (1-p)^{w-k}.$$