Tomasz Fabisiak 109784

Filip Waligórski 109765

Sprawozdanie z projektu „Sekwencjonowanie DNA”

Problem 7: model [0, 1, {2,3}, wiele], ISBH, błędy pozytywne.

# Opis problemu

Sekwencjonowanie DNA polega na odczytaniu sekwencji, czyli kolejności par nukleotydowych   
w cząsteczce DNA. Najpierw należy metodami laboratoryjnymi pozyskać spektrum (zbiór oligonukleotydów – fragmentów kwasów nukleinowych), a następnie, wykorzystując odpowiedni algorytm, połączyć otrzymane elementy w pojedynczą nić DNA. Kolejne fragmenty ze spektrum nakładają się na siebie z przesunięciem o 1 nukleotyd tworząc rozwiązanie. W wyniku sekwencjonowania izotermicznego otrzymuje się oligonukleotydy o różnej długości, zatem dodanie do budowanej nici DNA krótkiego fragmentu może nie wydłużyć rozwiązania. Jeśli w sekwencji, którą badamy występują powtórzenia (ten sam oligonukleotyd występuje w więcej niż jednym miejscu), informację o tym uzyskuje się poprzez zbadanie intensywności sygnału – im silniejszy, tym więcej powtórzeń. Jednakże wraz ze wzrostem intensywności maleje precyzja w określeniu liczby powtórzeń. Dodatkowo podczas pozyskiwania spektrum mogą wystąpić błędy – pozytywne (zawyżenie) lub negatywne (zaniżenie odczytanej liczby powtórzeń w stosunku do rzeczywistej ich liczby   
w sekwencji).

Instancją problemu jest spektrum zawierające oligonukleotydy wraz z informacją   
o powtórzeniach. Znane są także: długość badanej sekwencji (liczba nukleotydów, z których jest zbudowana), informacja, który z oligonukleotydów występuje w sekwencji jako pierwszy oraz informacja, który rodzaj błędów wystąpił (pozytywne, negatywne czy oba typy jednocześnie).

Rozwiązaniem jest sekwencja o zadanej długości (podanej w instancji), wykorzystująca wszystkie elementy spektrum zgodnie z informacją o powtórzeniach w granicach błędów pozytywnych.

Rozwiązanie uzyskane algorytmem dokładnym respektuje wszystkie powyższe ograniczenia, natomiast algorytm przybliżony może podać takie rozwiązanie, które nie spełnia ściśle wszystkich ograniczeń.

# Algorytmy

Na wstępie należy wspomnieć, że oba algorytmy szukają pierwszego prawidłowego rozwiązania, a nie wszystkich tudzież najlepszego. Z tego powodu czas poszukiwań zależy głównie od instancji – im mniej powtórzeń, tym szybciej owo rozwiązanie zostaje znalezione.

2.1 Algorytm dokładny

Cechą algorytmu dokładnego jest dążenie do przejrzenia pełnej przestrzeni potencjalnych rozwiązań – jeśli prawidłowe rozwiązanie istnieje, algorytm ten na pewno je znajdzie. Nasz algorytm dokładny jest algorytmem rekurencyjnym przechodzącym po drzewie rozwiązań w głąb (DFS) – dla każdego pasującego w danym miejscu oligonukleotydu dodaje go do rozwiązania, a następnie poszukuje kolejnych pasujących oligonukleotydów dla zmienionego rozwiązania. Na każdym etapie proponowane są tylko te pasujące elementy, które nie przekroczyły maksymalnej liczby powtórzeń.   
W momencie osiągnięcia zadanej długości rozwiązania algorytm sprawdza, czy wszystkie elementy ze spektrum zostały użyte minimalną liczbę razy (z uwzględnieniem błędów pozytywnych).

Złożoności:

* optymistyczna: O(1)
* pesymistyczna: O(mn)

gdzie: n – długość sekwencji,

m – średnia liczba pasujących w danym miejscu nukleotydów zależna od temperatur   
i powtórzeń (generalnie od instancji)

2.2 Algorytm przybliżony

Cechą algorytmu przybliżonego jest próba podania prawidłowego rozwiązania bez konieczności przeglądania pełnej przestrzeni. Odbywa się to jednakże kosztem jakości rozwiązania – zdarza się,   
że końcowe rozwiązanie nie spełnia wszystkich ograniczeń, co oznacza, że nie udało się znaleźć prawidłowego rozwiązania. Nasz algorytm przybliżony podzielony jest na 2 etapy powtarzane na przemian – budowę oraz ocenę. Na etapie budowy tworzona jest od podstaw sekwencja podobnie jak w przypadku algorytmu dokładnego, ale bez nawrotów – dopóki to możliwe, do rozwiązania dodawany jest oligonukleotyd wylosowany lub wytypowany spośród pasujących. Wytypowanie najlepszego oligonukleotydu odbywa się na podstawie informacji pochodzącej z etapu oceny.   
W momencie osiągnięcia zadanej długości lub braku możliwości dodania elementu do rozwiązania algorytm przechodzi do etapu oceny. Ocena polega na przypisaniu otrzymanej sekwencji punków karnych za niespełnione ograniczenia – złą długość oraz liczbę oligonukleotydów ze spektrum, które nie zostały wykorzystane odpowiednią liczbę razy. Każdemu takiemu oligonukleotydowi zwiększony zostaje wskaźnik informujący, że na etapie budowy ma być preferowany podczas typowania. Jeśli odpowiednio długo (*n razy*) nie znaleziono rozwiązania lepszego od najlepszego dotychczas, wskaźniki są kasowane, a sekwencja budowana jeden raz losowo. Po kilku (*m*) cyklach rozpoczynających się od losowej sekwencji algorytm jest się kończy.

Złożoności:

* optymistyczna: O(1)
* pesymistyczna: O(n\*m)

gdzie: n, m – wyjaśnione w tekście

# Wyniki eksperymentów

Przeprowadziliśmy 4 rodzaje testów – wpływ poniższych czynników na czas potrzebny   
do odczytania sekwencji:

1. długość sekwencji,
2. temperatura oligonukleotydów (2 sąsiadujące, parzyste temperatury),
3. liczba błędów pozytywnych polegających na dodaniu do spektrum nowych elementów   
   z liczbą powtórzeń równą 1 (I rodzaj błędów),
4. liczba błędów pozytywnych polegających na zwiększeniu liczby powtórzeń istniejących   
   w spektrum oligonukleotydów (II rodzaj błędów).

Dla każdego czynnika przygotowaliśmy 10 sekwencji, z których połowa pochodzi z 10 chromosomu ryżu, natomiast druga połowa została wygenerowana ręcznie w celu uzyskania większej liczby powtórzeń (w naturalnych sekwencjach powtórzenia w obrębie 100 nukleotydów występują bardzo rzadko).

W celu porównania otrzymanego rozwiązania z sekwencją wejściową, z której powstało spektrum, użyliśmy odległości Levenshteina (z tej metody nie korzystają oczywiście same algorytmy, gdyż badana sekwencja nie jest im znana). Może się jednak zdarzyć, że algorytm znalazł poprawną sekwencję inną niż badana (odległość Levenshteina różna od 0) – o prawidłowej długości oraz wykorzystująca wszystkie elementy zgodnie z powtórzeniami.

Na osi pionowej (y) znajduje się czas obliczeń wyrażony w sekundach, a na osi poziomej (x) – badany czynnik.

3.1 Wpływ długości sekwencji na czas obliczeń:

Badane dla temperatur 30 i 32, bez dodawania sztucznych błędów do spektrum.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Przybliżony | | |  | Dokładny | | |
| długość sekwencji | czas [s] | liczba poprawnych rozwiązań |  | długość sekwencji | czas [s] | liczba poprawnych rozwiązań |
| 20 | 0,0048 | 9 |  | 20 | 0,009 | 10 |
| 30 | 0,0052 | 9 |  | 30 | 0,002 | 10 |
| 40 | 0,0074 | 9 |  | 40 | 0,002 | 10 |
| 50 | 0,0096 | 9 |  | 50 | 0,133 | 10 |
| 60 | 0,0277 | 8 |  | 60 | 0,433 | 10 |
| 70 | 0,0246 | 9 |  | 70 | 0,449 | 10 |
| 80 | 0,0749 | 7 |  | 80 | 0,467 | 10 |
| 90 | 0,0999 | 7 |  | 90 | 0,481 | 10 |
| 100 | 0,1024 | 8 |  | 100 | 0,597 | 10 |
| 125 | 0,2477 | 7 |  | 125 | 0,956 | 9 |
| 150 | 0,2993 | 7 |  | 150 | 0,016 | 9 |
| 175 | 0,3520 | 6 |  | 175 | 2,192 | 9 |
| 200 | 0,5143 | 6 |  | 200 | 8,035 | 7 |

Test pokazuje, że dla dłuższych sekwencji znalezienie rozwiązania trwa dłużej. Niższa liczba poprawnych rozwiązań w algorytmie dokładnym wynika z konieczności przerwania poszukiwań po upływie ustalonego czasu granicznego (czas jest średnią z pozostałych instancji). W przypadku algorytmu przybliżonego wraz ze wzrostem długości sekwencji również spada częstotliwość znajdowania prawidłowych rozwiązań.

3.2 Wpływ temperatury oligonukleotydów na czas obliczeń:

Badane dla zestawu sekwencji o długości 50, bez dodawania sztucznych błędów do spektrum.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Przybliżony | | |  | Dokładny | | |
| Przybliżony | czas [s] | liczba poprawnych rozwiązań |  | temperatury | czas [s] | liczba poprawnych rozwiązań |
| 40-42 | 0,005 | 10 |  | 40-42 | 0,003 | 10 |
| 36-38 | 0,011 | 9 |  | 36-38 | 1,183 | 10 |
| 32-34 | 0,010 | 9 |  | 32-34 | 0,004 | 10 |
| 28-30 | 0,009 | 9 |  | 28-30 | 0,325 | 10 |
| 24-26 | 0,011 | 9 |  | 24-26 | 0,578 | 10 |
| 20-22 | 0,008 | 9 |  | 20-22 | 0,468 | 10 |
| 16-18 | 0,028 | 7 |  | 16-18 | 0,142 | 10 |
| 12-14 | 0,064 | 2 |  | 12-14 | 10,897 | 7 |
| 8-10 | 0,076 | 0 |  | 8-10 | 0,661 | 7 |
| 4-6 | 0,045 | 0 |  | 4-6 | 1,304 | 2 |

Zwiększenie temperatur powoduje wydłużenie oligonukleotydów, zwiększając ich unikalność   
w spektrum (mniejsza szansa na powtórzenia), dzięki czemu algorytmy wykonują się szybciej (istnieje mniej możliwości dopasowania oligonukleotydów). Natomiast przy niskich temperaturach liczba powtórzeń wzrasta, algorytmy stają się nieefektywne i nie podają prawidłowych rozwiązań.

3.3 Wpływ I rodzaju błędów na czas obliczeń:

Badane dla zestawu sekwencji o długości 80 i temperatur 30 i 32.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Przybliżony | | |  | Dokładny | | |
| % błędów | czas [s] | liczba poprawnych rozwiązań |  | % błędów | czas [s] | liczba poprawnych rozwiązań |
| 0 | 0,051 | 9 |  | 0 | 0,3859 | 10 |
| 5 | 0,088 | 7 |  | 5 | 0,3897 | 10 |
| 10 | 0,051 | 9 |  | 10 | 0,4091 | 10 |
| 15 | 0,078 | 7 |  | 15 | 0,4472 | 10 |
| 20 | 0,080 | 6 |  | 20 | 0,4377 | 10 |
| 25 | 0,075 | 7 |  | 25 | 0,4869 | 10 |
| 30 | 0,060 | 8 |  | 30 | 0,4859 | 10 |
| 35 | 0,081 | 6 |  | 35 | 0,5002 | 10 |
| 40 | 0,081 | 5 |  | 40 | 0,508 | 10 |
| 45 | 0,072 | 5 |  | 45 | 0,4614 | 10 |

Test polega na umieszczeniu w spektrum dodatkowych oligonukleotydów, które mają zmieniony 1 nukleotyd względem losowo wybranych oligonukleotydów będących już w spektrum. Liczba dodanych oligo wynosi procent z liczności spektrum.

Z powyższego testu wynika, że dodanie nowych oligonukleotydów do spektrum ma znikomy wpływ zarówno na czas znalezienia rozwiązania, jak i na jego jakość w przypadku algorytmu dokładnego. W przypadku algorytmu przybliżonego wraz ze wzrostem liczby błędów spada prawdopodobieństwo znalezienia rozwiązania spełniającego wszystkie ograniczenia.

3.4 Wpływ II rodzaju błędów na czas obliczeń:

Badane dla zestawu sekwencji o długości 50 i temperatur 30 i 32.

|  |  |  |  |
| --- | --- | --- | --- |
| Przybliżony | | | |
| % błędów | czas [s] | liczba poprawnych rozwiązań | średnia odległość Levenshteina |
| 0 | 0,0102 | 9 | 0,8 |
| 5 | 0,0248 | 8 | 0,6 |
| 10 | 0,0309 | 7 | 3,8 |
| 15 | 0,0666 | 3 | 0,7 |
| 20 | 0,0756 | 2 | 1,6 |
| 25 | 0,0874 | 1 | 0,7 |
| 30 | 0,0969 | 0 | 1,4 |
| 35 | 0,0961 | 0 | 0,7 |
| 40 | 0,0967 | 0 | 0,8 |
| 45 | 0,0965 | 0 | 3,6 |

|  |  |  |  |
| --- | --- | --- | --- |
| Dokładny | | | |
| % błędów | czas [s] | liczba poprawnych rozwiązań | średnia odległość Levenshteina |
| 0 | 0,1474 | 9 | 0,2 |
| 5 | 3,3417 | 8 | 0,222222 |
| 10 | 1,5100 | 7 | 0,285714 |
| 15 | 7,6682 | 3 | 0,666667 |
| 20 | 8,1110 | 2 | 1 |
| 25 | 8,5254 | 1 | 0 |
| 30 | 8,6633 | 0 | ---- |
| 35 | 8,2236 | 0 | ---- |
| 40 | 8,2000 | 0 | ---- |
| 45 | 8,6717 | 0 | ---- |

Dla algorytmu dokładnego nie wzięliśmy pod uwagę jednej z sekwencji, gdyż 8/10 testów dla niej zostało przerwanych.

Powyższy test polegający na sztucznym zwiększeniu powtórzeń wybranych oligonukleotydów, wyraźnie pokazuje, że im większy procent błędów tego rodzaju, tym czas wykonania staje się dłuższy. W przypadku algorytmu dokładnego wynika to jednak z faktu, że algorytm przeglądając całą przestrzeń rozwiązań nie jest w stanie znaleźć prawidłowego – przy takiej liczbie błędów jest to niemożliwe. Algorytm przybliżony natomiast podaje często prawidłową sekwencję (stwierdzone na podstawie odległości Levenshteina podczas analizy wyników), jednakże rozwiązanie nie jest prawidłowe ze względu na nieużycie wszystkich elementów spektrum zgodnie z ich powtórzeniami (nawet przy uwzględnieniu błędu pozytywnego).

# Podsumowanie

Niewątpliwie wpływ na czas potrzebny do znalezienia prawidłowego rozwiązania ma długość badanej sekwencji. Istotnym czynnikiem jest również liczba powtórzeń w spektrum – w sekwencjach naturalnych występują one rzadziej niż w utworzonych ręcznie, dlatego algorytmy znacznie szybciej badały te pierwsze.