Algorytmy kombinatoryczne w bioinformatyce

wykład 5: asemblacja

prof. dr hab. inż. **Marta Kasprzak**
Instytut Informatyki, Politechnika Poznańska

Asemblacja

- Asemblacja *de novo* łańcuchów DNA jest problemem trudnym obliczeniowo. Nawet bardzo ograniczony wariant tego problemu – problem najkrótszego wspólnego superciągu (ang. *shortest common superstring*) – jest silnie NP-trudny
- Problem asemblacji jest dodatkowo skomplikowany przez liczbę i różnorodność błędów występujących w instancji, a także przez jej znaczny rozmiar
- Sekwencje mogą pochodzić z obu nici DNA i ich orientacja nie jest znana. Zawierają przekłamania przeniesione z etapu sekwencjonowania: insercje, delecje i zamiany nukleotydów.

Asemblacja

Cechy instancji problemu asemblacji DNA

- Ogólne sformułowanie problemu asemblacji zakłada sekwencje wejściowe różnej długości, gdzie jedne mogą zawierać się w innych
- Rozkład sekwencji w badanym fragmencie genomu jest nierównomierny. Brak pokrycia jest jedną z przyczyn rekonstrukcji genomu w postaci rozłącznych odcinków (tzw. *kontigów*)

Asemblacja

Fragmenty składowe:

TATGC, ATCAGCAC, GACTC, GTAGA, GCATCA

Jedno z możliwych rozwiązań:

```
TATGCAGCACTCTAC
---------------
TATGC G-ACTC
   GCATCA TCTAC
 AT-CAGCAC
```

Asemblacja

- Sformułowanie wersji optymalizacyjnej problemu asemblacji

Instancja: Multizbiór *S* sekwencji pochodzących z obu nici badanego łańcucha DNA.

Odpowiedź: Sekwencja wynikowa o maksymalnej wiarygodności zawierająca, z dopuszczonym pewnym odsetkiem niezgodności, wszystkie sekwencje z *S* czytane wprost lub z założeniem przeciwnej orientacji.

- Najczęściej rozwiązaniem jest nie jedna spójna sekwencja, lecz zbiór rozłącznych kontigów
- Istnieją alternatywne sformułowania dla tego problemu: inne kryterium optymalizacji (np. minimalizacja długości sekwencji wynikowej), inne ograniczenia (nieużycie części zbioru *S*)

Dopasowanie–uszeregowanie–konsensus

Dawniej algorytmy asemblacji opierały się często na trzyetapowym modelu obliczeń (ang. *overlap-layout-consensus*):

- Dopasowanie par sekwencji wejściowych – w celu znalezienia potencjalnych sąsiadów w rozwiązaniu, z dopuszczeniem pewnych niezgodności (dopasowanie semiglobalne)
- Uszeregowanie sekwencji – znajdowanie prawdopodobnego uporządkowania ich w sekwencji oryginalnej
- Generowanie sekwencji konsensusowej – wywiedzenie sekwencji nukleotydów ze znalezionego uszeregowania

Asemblacja – graf nałożeń

[J.D. Kececioglu i E.W. Myers, *Algorithmica* 13 (1995)]

- Na wejściu sekwencje z błędami (insercje, delecje, zamiany) pochodzące z obu nici łańcucha DNA. Zastosowano model obliczeń *overlap-layout-consensus*
- Porównanie wszystkich par sekwencji, z uwzględnieniem ich odwrotnie komplementarnych odpowiedników, w celu otrzymania nałożeń o dopuszczonym odsetku błędów
- Konstrukcja grafu skierowanego z wierzchołkami odpowiadającymi sekwencjom (×2) i łukami odpowiadającymi najlepszym nałożeniom danej pary wierzchołków
- Graf redukowany jest pod kątem maksymalizacji zysku do momentu pozostawienia po jednym wierzchołku z każdej pary i utworzenia struktury lasu skierowanego (zbioru drzew)

Dekompozycja odczytów

- Model obliczeń obecnie najczęściej wykorzystywany w asemblacji odczytów pochodzących z sekwencjonowania nowej generacji opiera się na różnych wariantach grafów z metody Pevznera (błędnie nazywanych *grafami de Bruijna*)
- Grafy powstają przez dekompozycję sekwencji wejściowych (odczytów z sekwenatora) na serie krótszych nakładających się *k*-merów, które reprezentowane są jako łuki. Dopuszcza się tylko bezbłędne nałożenia. Błędna informacja obecna w instancji jest korygowana lub ignorowana
- Rozwiązaniem jest zbiór ścieżek odpowiadających kontigom, włączających w miarę możliwości po jednej sekwencji z pary „czytana wprost – odwrotnie komplementarna”

Asemblacja – graf dekompozycji

[R. Idury i M. Waterman, *J. Comp. Biol.* 2 (1995)]

- Sekwencje wejściowe i ich odwrotnie komplementarne odpowiedniki dekomponowane są na serie $n–k+1$ k-merów, gdzie n jest długością sekwencji
- Konstrukcja grafu skierowanego z łukami odpowiadającymi k-merom i wierzchołkami odpowiadającymi ich prefiksom/ sufiksom o długości $k–1$. Rozwiązaniem jest zbiór ścieżek pokrywających graf
- W celu zachowania informacji o sekwencjach wejściowych preferowana jest duża wartość k, dodatkowo pamiętana jest przynależność łuków do sekwencji
- Metoda zakłada brak błędów w sekwencjach, ale autorzy dopuścili pominięcie pewnych łuków w rozwiązaniu lub użycie niektórych więcej razy niż przewidziano

Asemblacja – graf dekompozycji

[P.A. Pevzner, H. Tang i M.S. Waterman, *PNAS 98* (2001)]

- Algorytm *EULER*, pierwszy z serii wielu podobnych bazujących na dekompozycji odczytów (EULER-DB, EULER-SF, EULER-CN, EULER+, EULER-SR). Na wejściu sekwencje z dopuszczeniem błędów pochodzące z obu nici DNA
- Etap korekcji błędów: dążenie do eliminacji maksymalnej liczby potencjalnych błędów w sekwencjach (insercji, delecji, zamian) poprzez relatywnie małą liczbę mutacji. Efektywność mutacji jest mierzona całkowitą liczbą k-merów w instancji (włączając w to odwrotnie komplementarne odpowiedniki sekwencji)
- Graf skierowany konstruowany jest jak w metodzie Pevznera. Dodatkowo pamiętana jest kolekcja ścieżek odpowiadających sekwencjom wejściowym. Rozwiązaniem jest ścieżka Eulera (lub zbiór ścieżek) zawierająca wszystkie zapamiętane ścieżki

Asemblacja – graf dekompozycji

[D.R. Zerbino i E. Birney, *Genome Research 18* (2008)]

- Algorytm *Velvet*: zastosowanie dekompozycji odczytów i grafu Pevznera. Ścieżki proste (bez rozgałęzień) kumulowane są do jednego wierzchołka, a odcinki odwrotnie komplementarne łączone są w pary
- Etap korekcji błędów: usuwanie krótkich „ślepych" ścieżek, usuwanie ścieżek równoległych do innych o zbliżonej sekwencji, usuwanie ścieżek o zbyt niskim pokryciu
- Rozwiązanie budowane jest z uwzględnieniem oryginalnych odczytów jako wskazówek do przechodzenia przez rozgałęzienia
- W wariancie programu dostosowanym do odczytów sparowanych wierzchołki łączone są mostkami wskazującymi na ich sparowanie. Mostki służą jako wskazówki do budowy rozwiązania

Porównanie podejść

- Mocną stroną modelu opartego na dekompozycji odczytów jest mniejszy wpływ błędów sekwencjonowania na postać generowanego rozwiązania. Tutaj *k*-mer z błędem często nie bierze udziału w tworzeniu rozwiązania, gdyż w danych przeważają *k*-mery bezbłędne powstałe z dekompozycji sąsiednich odczytów — korekcja błędów jest prostsza
- Zastosowana reprezentacja grafu dekompozycji skutkuje mniejszą zajętością pamięci (mniej wierzchołków, mniej łuków), brak błędów nałożeń skraca też czas tworzenia grafu oraz eliminuje etap generowania sekwencji konsensusowej

Porównanie podejść

- Słabą stroną modelu dekompozycji odczytów jest utrata informacji spowodowana rozbiciem ich na krótsze fragmenty. Niektóre algorytmy próbują sobie z tym radzić, odwołując się do odczytów źródłowych w trakcie poszukiwania sekwencji wynikowej. Rośnie jednak wtedy złożoność algorytmu, a braku informacji zwykle nie da się nadrobić w całości (procedura jest heurystyczna)
- Przyjęcie modelu ścieżki Eulera zamiast Hamiltona nie daje w tym problemie zysku na złożoności, obecne dodatkowe ograniczenia czynią problem trudnym obliczeniowo

Graf dla odczytów sparowanych

[P. Medvedev i in., *Lect. Notes Comput. Sci.* 6577 (2011)]

- Zaproponowany został nowy rodzaj grafów (*paired de Bruijn graphs*), w których informacja o sparowaniu odczytów wpływa na ich konstrukcję
- W dotychczas istniejących algorytmach asemblacji informacja o parach jest początkowo ignorowana, tworzone są grafy asemblacji jak dla pojedynczych odczytów i dopiero na etapie budowy ścieżki informacja o parach służy do wskazywania właściwej drogi
- Sposób konstrukcji nowych grafów sprawia, że zwykle są mniej zawikłane, gdyż dopuszcza się połączenia tylko wtedy, gdy nakładają się na siebie oba elementy z dwóch rozważanych par, w zadanej kolejności i w dopuszczalnym przesunięciu

Graf dla odczytów sparowanych

[P. Medvedev i in., *Lect. Notes Comput. Sci.* 6577 (2011)] — cd.

- Zostały wyróżnione dwa typy sparowanych grafów: *idealny* z dokładną odległością pomiędzy odczytami (taką samą dla wszystkich odczytów w instancji) oraz *przybliżony* z odległością z dopuszczonego zakresu $\pm\Delta$
- Podobnie jak w klasycznej metodzie opartej na dekompozycji odczytów, pary odczytów rozbijane są na pary krótszych k-merów związanych z łukami grafu
- Grafy są konstruowane z bezbłędnym nałożeniem k-merów, dla odczytów przetłumaczonych na tę samą nić DNA i o znanej względem siebie orientacji (lewy odczyt w parze jest poprzednikiem prawego). Dla odczytów o nieznanej orientacji proponowane jest dublowanie danych z instancji i budowanie podwójnego grafu

Graf dla odczytów sparowanych

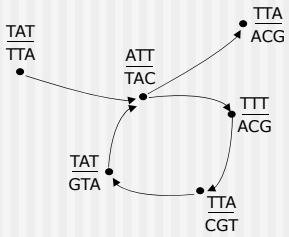
[P. Medvedev i in., *Lect. Notes Comput. Sci.* 6577 (2011)] — cd.

- *W grafie idealnym* dla dokładnych odległości każdą parę odczytów dekomponuje się na serię par k-merów. Łuki rozpięte są pomiędzy wierzchołkami reprezentującymi pary ich sufiksów i prefiksów o długości k–1
- W *grafie przybliżonym* dodatkowo skleja się wierzchołki, których lewe etykiety pokrywają się, a prawe są względem siebie przesunięte w ramach dopuszczonego zakresu $\pm\Delta$. Jeśli $\Delta \geq \frac{1}{2} k$, to zamiast porównywać prawe etykiety dwóch wierzchołków należy obliczyć długość najkrótszej ścieżki w grafie pomiędzy tymi wierzchołkami

Graf dla odczytów sparowanych

- Przykład grafu idealnego z metody Medvedeva i in.

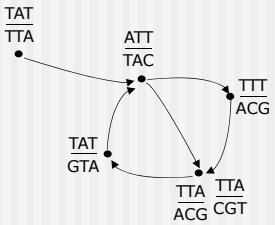
Sparowane odczyty z odległością pomiędzy nimi równą 1:
TATTT+TTACG, ATTTA+TACGT, TTTAT+ACGTA, TTATT+CGTAC, TATTA+GTACG
Sparowane k-mery o długości 4:
TATT+TTAC, ATTT+TACG, TTTA+ACGT, TTAT+CGTA, TATT+GTAC, ATTA+TACG



Sekwencja wynikowa:
TATTTATTACGTACG

Graf dla odczytów sparowanych

- Przykład grafu przybliżonego dla progu błędu $\Delta = 1$



Sekwencje wynikowe:
TATTTATTACGTACG
TATTATTTACGTACG

- Dopuszczenie dużej wartości Δ spotykanej w rzeczywistych eksperymentach (nawet do 25% odległości pomiędzy odczytami, odległość nawet do kilku tysięcy nukleotydów) mocno komplikuje postać grafu. Mimo to graf taki będzie źródłem nie większej liczby możliwych ścieżek niż graf tradycyjny