Tomasz Dzięcioł 132216 tomasz.dzieciol@student.put.poznan.pl
Ewa Fengler 132219 ewa.fengler@student.put.poznan.pl

Bioinformatyka, laboratoria środa godz. 11:45 – 13:15

Sprawozdanie 1
Teoretyczne opracowanie metody heurystycznej

1. Ogólne przedstawienie idei algorytmu.

W celu rozwiązania problemu 3. chcielibyśmy zastosować algorytm oparty na idei algorytmów mrówkowych. Generalnie rzecz ujmując, podejście do rozwiązywanego zadania można opisać w następujących kilku zdaniach. Program wykonuje mnóstwo iteracji, w których kolejno uzyskiwane rezultaty powinny prezentować coraz to lepsze wyniki. Taka pojedyncza iteracja będzie się składać z wielu prób utworzenia jak najdłuższej sekwencji nukleotydów. Tworzenie ciągu opierało by się tutaj na wyborze $i + 1$ wyrazu w sytuacji dokonania już wyboru i-tego wyrazu w wynikowej sekwencji. Wybór ten będzie oparty na macierzy prawdopodobieństw (określanej często w dziedzinie algorytmów mrówkowych jako macierz feromonów), która pozwoli dokonywać decyzje łączące najczęściej wyrazy o jak najdłuższym wspólnym ciągu znaków. Po zakończeniu wszystkich podejść w danej iteracji powinno zostać wykonane podsumowanie polegające na nagrodzeniu najlepszych rozwiązań. Podsumowanie to wpłynie bowiem na działanie programu w przyszłych iteracjach, otwierając tym samym furtkę do uzyskiwania coraz to lepszych rezultatów.

2. Inicjalizacja macierzy wag prawdopodobieństwa.

Działanie algorytmu trzeba zacząć od przygotowania macierzy feromonów, na której to głównie będzie opierało się tworzenie wynikowego ciągu. Nadanie początkowej wagi prawdopodobieństwa dla wyboru j-tego wyrazu, gdy wybrany został i-ty, będzie oparte na długości wspólnego sufiksu i prefiksu kolejnych słów (określanej dalej jako wspólne brzegi wyrazów). Strategia ta wydaje się być prawidłowa, gdyż im dłuższa wspólna część wyrazów, tym większa liczba wyrazów będzie mogła zostać umieszczona na ograniczonej długości wynikowego ciągu.

W związku z tą obserwacją, para wyrazów będzie premiowana k^p-krotnie większą wagą za posiadanie dłuższego o p znaków wspólnych brzegów, aniżeli inna para wyrazów o krótszym wspólnym brzegu o długość p znaków. Przykładowo dla $k = 2$ jeśli wyraz (1) posiada wspólną końcówkę z wyrazem (2) o długości 9, a z wyrazami (3), (4), (5) długości wspólnych końcówek wynoszą kolejno 8, 7, 6 to prawdopodobieństwo wyboru jako kolejnego słowa wyrazu (2) powinno być 2-krotnie, 4-krotnie, 8-krotnie większe niż kolejno wyrazów (3), (4), (5). Na ten moment pozostaje więc pytanie: ile powinien wynosić parametr k? Odpowiedź na nie będzie trzeba odnaleźć w trakcie przeprowadzania testów.

Zaprezentowana inicjalizacja macierzy wag prawdopodobieństwa może budzić jednak pewne zastrzeżenia. Być może zdarzy się możliwość wyboru kolejnego wyrazu o wspólnym brzegu o długości p dla tylko jednego wyrazu, podczas gdy dla długości $p - 1$ może znaleźć się mnóstwo możliwości dopasowania kolejnego wyrazu. W skutek tego okaże się, że sumarycznie większe prawdopodobieństwo będzie miało połączenie wyrazów mających mniejszą część wspólną brzegów, co może negatywnie wpływać już na samym początku algorytmu. Z tego powodu zbadane zostanie alternatywne podejście do inicjalizowania macierzy wag prawdopodobieństwa. Zastosowane

zostanie też podejście ogólnie k-krotnie mniejszego prawdopodobieństwa dla wyboru kolejnego słowa o długości wspólnego brzegu $p-1$, niż dla innego słowa o wspólnym brzegu długości p. Tym sposobem będzie ustalone już prawdopodobieństwo x, że kolejny wyraz tworzyć będzie z ostatnio wybranym już wyrazem wspólny brzeg o długości p. Wyrazów spełniających tą zależność może być wiele (n) i z tego powodu będzie trzeba równomiernie rozdzielić prawdopodobieństwo na wszystkie te słowa, co daje wartość $\frac{x}{n}$ dla każdego z tych poszczególnych słów.

Cały czas jest tutaj mowa o szansie wyboru kolejnego wyrazu w momencie wybrania już poprzednio jakiegoś innego wyrazu. Brakuje tylko wytłumaczenia, na jakiej podstawie będzie odbywał się wybór słowa rozpoczynającego tworzony ciąg sekwencji. Otóż, będzie to oparte również na prawdopodobieństwie, które będzie się zmieniać w ramach kolejnych iteracji. Należy je tylko odpowiednio zainicjować. Dla poruszanego problemu ważna wydaje się być informacja, że dla danego słowa ciężko idzie znaleźć długi prefiks, który by pokrywał się z jakimkolwiek sufiksem innego słowa. W związku z tym szansę rozpoczęcia ciągu danym wyrazem można oprzeć na maksymalnej długości wspólnego prefiksu z sufiksem jakiegokolwiek innego wyrazu. Im krótsza ta maksymalna długość, tym bardziej prawdopodobny powinien być fakt rozpoczynania wynikowego ciągu przez dany wyraz. Zasadę wyznaczania wag prawdopodobieństwa będzie trzeba zbadać analogicznie na dwa sposoby zaprezentowane dla poszerzania wynikowego ciągu o kolejne wyrazy.

3. Przeprowadzenie kolejnych iteracji algorytmu.

Posiadając już gotową macierz wag prawdopodobieństwa można przystąpić do wykonywania właściwego algorytmu. Będzie to wykonywanie serii prób utworzenia wynikowego ciągu, przeprowadzenie podsumowania i nagrodzenia najlepszych rozwiązań, a następnie przejście do kolejnej iteracji. Liczba iteracji może zostać uzależniona od postępu programu. Można by uznać, że jeśli od kilku iteracji program nie uzyskuje nowych lepszych rozwiązań, to została już odnaleziona ta najlepsza możliwość złożenia wyrazów w jeden ciąg. Parametryzacji trzeba natomiast poddać liczbę prób utworzenia wyniku w danej iteracji. Należy zbadać, czy wystarczająca będzie pewna stała liczba tworzonych rozwiązań, czy może jednak będzie trzeba uzależnić ją od pewnego czynnika, np. liczby słów w danej instancji.

Tak więc każda iteracja rozpoczyna się od wielu prób utworzenia rozwiązania, jakim jest ciąg znaków otrzymany przez złożenie jak największej liczby wyrazów, nie przekraczający określonej długości. Generowanie to odbywa się na podstawie omawianej już macierzy wag prawdopodobieństwa, gdzie określona jest szansa wyboru dla każdego dostępnego wyrazu. Zaznaczyć tu trzeba, że przez dostępny wyraz rozumie się taki wyraz, który nie został już użyty do zbudowania dotychczasowego ciągu. Dzięki opieraniu się na macierzy wag prawdopodobieństw zapewnione jest, że jako kolejny wyraz najczęściej zostanie wybrany ten, który ma całkiem długi prefiks, który pokrywa się z sufiksem poprzednio wybranego wyrazu. Jednocześnie nie jest wykluczone, że jako kolejny wyraz zostanie wybrany być może ten, który w ogóle nie posiada wspólnego ciągu znaków. Takie nieoptymalne zachowanie raczej nie wystąpi w świetle możliwości dokonania znacznie lepszego wyboru. Z drugiej strony, dla wybranego już słowa o nietypowym sufiksie algorytm nie będzie miał kompleksów z wybraniem jakiegokolwiek z dostępnych jedynie słabych połączeń. Nie nastąpi żadne nieoczekiwane przerwanie dobierania następnych wyrazów, proces ten będzie mógł zostać wstrzymany jedynie przez ograniczenie na ilość znaków zawartych w wynikowej sekwencji.

Po uzyskaniu żądanej ilości różnych rozwiązań przychodzi czas na podsumowanie. Aby móc w kolejnych iteracjach spodziewać się lepszych rezultatów, należy odpowiednio zmodyfikować macierz prawdopodobieństwa. Istotnym źródłem zmian będzie nagrodzenie najlepszych uzyskanych rezultatów. Głównym celem algorytmu jest uzyskanie sekwencji złożonej z jak największej ilości wyrazów, dlatego też nagradzane będą rozwiązania korzystające z największej ilości słów. W przypadku gdy jakieś rozwiązania wykorzystały jednakową liczbę wyrazów, to nadal można rozróżnić, które z nich było lepsze, jeśli weźmie się pod uwagę długość stworzonej sekwencji. Mniejsza liczba zużytych znaków świadczy o lepszym skompresowaniu wyrazów, dzięki czemu można uważać to rozwiązanie za lepsze. Nagrodzenie danego rozwiązania wiązało by się z powiększeniem szans wybrania danego połączenia dwóch kolejno wybranych wyrazów, które oczywiście zostały użyte w otrzymanym wyniku. Jednakowe wynagrodzenie dla każdych sąsiadujących par w sekwencji może być raczej mało sprawiedliwe, gdyż dla tych poszczególnych par wyrazów długości wspólnych brzegów nie koniecznie będą jednakowe. Tak więc słowa nachodzące na siebie na większej liczbie znaków powinny być lepiej nagrodzone od słabiej zazębiających się kolejno wybieranych par słów.

Do przeprowadzenia nagrodzenia należy wybrać odpowiednią liczbę uzyskanych najlepszych rozwiązań. Będzie trzeba to przebadać w procesie strojenia heurystyki, spodziewana liczba będzie wynosić około 10% wszystkich stworzonych rozwiązań. Metoda naliczenia nagrodzenia, która wydaje się być słuszną, jest zwielokrotnienie obecnej wagi prawdopodobieństwa zestawienia dwóch słów. Spodziewane zwielokrotnienie nie powinno być duże, aby nie sprowokować algorytmu do wybierania w późniejszych iteracjach jednakowych utartych szlaków. Wartości mnożnika nieprzekraczające wartości $1{,}2$ powinny być wystarczające, co i tak będzie trzeba odpowiednio wybadać w trakcie testów. Co więcej, bardziej poprawniej brzmi liniowy spadek nagrody dla kolejnego słabszego rozwiązania. Pozwoli to na wyróżnienie rozwiązań najlepszych z najlepszych. Pozostaje też przedstawić kwestię właściwego nagrodzenia wykorzystanych połączeń między kolejno użytymi wyrazami. Tutaj też zastosowana zostanie liniowa metoda nagradzania. Oznacza to zwielokrotnienie danego połączenia wyrazów, mających maksymalnie długi wspólny brzeg na $l-1$ pozycjach, o zadeklarowaną wartość dla danego rozwiązania. Posiadanie wspólnego prefiksu i sufiksu o długości coraz to mniejszej powodować będzie coraz to mniejszy liniowy spadek zwielokrotnienia. Dla skrajnego przypadku wyrazów, które nie posiadały części wspólnej, oznaczać to będzie współczynnik przemnożenia wagi prawdopodobieństwa o wartości 1, co będzie oznaczał brak jakiejkolwiek zmiany. Te spostrzeżenia wyczerpują temat nagradzania najlepszych rozwiązań uzyskiwanych w danej iteracji.

W kwestii modyfikowania macierzy wag prawdopodobieństw są jeszcze do zaimplementowania elementy charakterystyczne dla algorytmów mrówkowych. Wykorzystane zostanie tak zwane odparowywanie oraz wygładzanie feromonów. Pod pojęciem odparowywania rozumie się jednakowe osłabienie wszystkich elementów macierzy wag prawdopodobieństw. Dzięki temu możliwe jest wyeliminowanie nieużywanych połączeń między wyrazami, których w związku z tym już i tak nie warto będzie dalej brać pod uwagę. Drugą wspomnianą rzeczą było wygładzanie, którego to celem jest zapobieganie występowania nadmiernego dominowania wyboru kolejnego wyrazu nad innymi. Uzyskanie tego celu będzie możliwe dzięki zastosowaniu odpowiedniego wzoru, na przykład: $m_{i,j} = m_0 * (1 + \log_{podstawa} \frac{m_{i,j}}{m_0})$, gdzie m_0 oznacza minimalną wartość i-tego wiersza w macierzy feromonów. Parametry odparowywania i wygładzania stają się kolejnymi wartościami, które muszą zostać rozpoznane w trakcie pracy z gotowym algorytmem.

4. Rozważania na temat dodatkowej wiedzy o typie błędów zawartych w instancji

Przedstawiona metoda jest całkiem elastyczna. Pozwala ona nie przejmować się problemem brakujących wyrazów, czy też nadmiarowych wyrazów. Spośród podanej puli wyrazów będą generowane wielokrotnie rozwiązania, które to będą dopasowywały się do ograniczeń i korzystały z odkrytych wcześniej i nagrodzonych pożądanych rozwiązań. Możliwe jest to głównie dzięki braku obowiązku tworzenia połączeń opierających się na konkretnie określonej liczbie znaków wspólnego sufiksu-prefiksu wyrazów. Mimo tego, można wyróżnić dodatkowe funkcjonalności, które przydałyby się dla bardziej konkretnych typów instancji.

Dla problemu sekwencjonowania łańcuchów DNA z błędami negatywnymi nie widać w zastosowanym algorytmie wad dla braku jakiegoś unikalnego wyrazu spośród idealnego spektrum. Algorytm powinien teoretycznie poradzić sobie ze złożeniem oryginalnej sekwencji, nie przeszkodzi w tym brak dopasowani na maksymalnych $l-1$ pozycjach. Dopiero błędy wynikające z powtórzeń okazują się być kłopotliwe. W zaprezentowanym algorytmie mowa była o składaniu ciągu poprzez złożenie w pewnej kolejności dostępnych wyrazów. Nie była tutaj brana pod uwagę jakakolwiek możliwość ponownego wykorzystania wyrazów.

Rozwiązanie tego problemu być może jest całkiem nie trudne. Najprostszym ruchem było by umożliwienie sklejania ciągów bez wykluczania ponownego użycia wykorzystanych wyrazów. Już tylko taka zmiana pozwalałaby na powtórzenia, lecz powstaje pytanie, czy takie udogodnienie niebyło by nadużywane. Być może prowadziło by to do zapętlania się decyzji algorytmu i banalnego powtarzania pewnej grupki wyrazów. Krokiem skierowanym ku zapobiegnięciu takiej sytuacji było by ograniczenie wielokrotnego użycia danych wyrazów. Ograniczenie wykorzystania danego wyrazu opierało by się na liczbie innych wyrazów, z którymi to uzyskiwana długość wspólnego brzegu wyrazów była by całkiem spora. Im większa liczba takich wyrazów, tym więcej razy może być potencjalnie wykorzystany wyszczególniony wyraz. Takie ograniczenie powtórzeń zdaje się być najlepszym kompromisem dla tego problemu.

Drugi z problemów sekwencjonowania, który skupia się na błędach pozytywnych, również na pierwszy rzut oka nie wydaje się być kłopotliwy. Wyrazy błędnie uznane za należące do sekwencji nie powinny przeszkodzić, jeśli posiadane są wszystkie wyrazy składające się na oryginalną sekwencje. Dzięki dokonywaniu wielu prób utworzenia rozwiązań, wysoce prawdopodobne jest odbudowanie sekwencji, dzięki użyciu tylko tych właściwych wyrazów. Co więcej, aby pomóc programowi osiągnąć optymalne rozwiązanie, nie trzeba rozbudowywać algorytmu, a wręcz go ograniczyć.

Przedstawiany algorytm brał pod uwagę możliwość utworzenia ciągu poprzez dobieranie kolejnych słów, które nie koniecznie musiały pokrywać się na $l-1$ pozycjach, Było to odzwierciedlane niejako poprzez różną wagę zaistnienia danego połączenia. Dla problemu sekwencjonowania z błędami pozytywnymi do dyspozycji są wszystkie wyrazy bazowej sekwencji wraz z pewnymi dodatkowymi fałszywymi wyrazami. Wystarczy więc uwzględniać tylko pary wyrazów posiadające maksymalnie możliwą długość wspólnego sufiksu-prefiksu i tylko im nadać wagę w macierzy prawdopodobieństw. Dzięki temu generowane rozwiązania algorytmu mrówkowego będą opierały się tylko na silnie zazębiających się wyrazach, co znacznie ulepszy działanie algorytmu zaprezentowanego dla ogólnego problemu sekwencjonowania DNA. Poprzez system generowania wielu rozwiązań i nagradzania najlepszych rozwiązań nawet przekłamania na końcach oligonukleotydów nie wydają się być straszne. Błędy te pojawiające się w instancjach będą mogły sprowokować powstawanie bardzo nieoptymalnych rozwiązań, lecz nie powinny one wpływać na rozwój heurystyki. Takie rozwiązania dające słaby wynik nie będą brane pod uwagę. Uwzględnione zostaną najlepsze rozwiązania, a więc te, które nie będą zawierały niewygodnych słów wynikających ze zjawiska przekłamań na końcach oligonukleotydów.