wpł. 05.10.2023 M. Skow

UNIWERSYTET WARSZAWSKI | UW MIM

Wydział Matematyki, Informatyki i Mechaniki
Instytut Informatyki

dr hab. Paweł Górecki, prof. UW
Wydział Matematyki, Informatyki i Mechaniki
Uniwersytet Warszawski
Banacha 2, 02-097 Warszawa
gorecki@mimuw.edu.pl

Warszawa, 21 września 2023

Recenzja rozprawy doktorskiej mgr. inż. Macieja Długosza pt. *Korekcja danych z sekwencjonowania genomów.*

I. Problematyka naukowa rozprawy

Recenzowana rozprawa doktorska mgr. inż. Macieja Długosza poświęcona jest zagadnieniom dotyczącym korekcji odczytów otrzymanych z sekwencjonowania genomów urządzeniami marki Illumina. Rozprawa zawiera opis algorytmów do korekcji w tym opracowanego i zaimplementowanego przez Autora algorytmu RECKONER wraz z analizą złożoności pesymistycznej i studium porównawczego z istniejącymi alternatywnymi narzędziami. Podejmowana tematyka badawcza należy do pogranicza informatyki i bioinformatyki i dobrze wpisuje się w światowe trendy badawcze.

II. Zawartość rozprawy

Na bazie cech istniejących algorytmów korekcji w rozprawie przedstawiony jest nowy algorytm o nazwie RECKONER, który wprowadza nowe rozwiązania poprawiające skuteczność i efektywność procesu korekcji.

Rozprawa napisana w języku polskim składa się z 7 rozdziałów i 3 dodatków. Rozdział pierwszy to wprowadzenie do rozprawy przedstawiające tezy, cele oraz krótkie streszczenie kolejnych rozdziałów. Rozdział drugi zawiera podstawy ewolucyjne i biologiczne oraz wstęp do sekwencjonowania. Rozdział trzeci wprowadza pojęcia i zagadnienia informatyczne powiązane z tematyką rozprawy:

PG

struktury danych, heurystyki optymalizacyjnych i przegląd algorytmów sortowania k-merów. W Rozdziale czwartym Autor przedstawia istniejące rozwiązania dot. algorytmów korekcji oraz pokrewne zagadnienia w tym symulowanie sekwencjonawania. Główny wkład Autora jest zawarty w piątym (o algorytmach) i szóstym Rozdziale (studium porównawcze), które bardziej szczegółowo przedstawię poniżej. Rozdział siódmy zawiera podsumowanie. Ponadto w rozprawie znajdują się trzy dodatki z dodatkowymi pseudokodami, szczegółami wierszy poleceń z parametrami oraz dodatkowymi tablicami wyników eksperymentalnych.

III. Opinia

Główne wyniki rozprawy są zamieszczone w Rozdziałach 5 i 6. W Rozdziale 5 znajduje się szczegółowy opis algorytmów wraz z analizą złożoności, a w Rozdziale 6 przedstawione są wyniki eksperymentów obliczeniowych.

Autor w Rozdziale 5 przedstawia najpierw opis dwóch znanych algorytmów: KMC i BLESS. BLESS jest podstawą dla algorytmu RECKONER zaproponowanego przez Autora, natomiast algorytm KMC jest używany w RECKONER do preprocessingu w zliczaniu k-merów. Algorytm BLESS jest przedstawiony w postaci pseudokodów i opisu, ponadto przedstawiona też jest autorska analiza złożoności pesymistycznej BLESS. Następnie w analogiczny sposób zaprezentowany jest algorytm RECKONER.

Przedstawiony algorytm RECKONER jest dość skomplikowaną heurystyką o wielu etapach przetwarzania z dużym zestawem parametrów sterujących. Sam algorytm RECKONER i zawarte wyniki teoretyczne można uznać generalnie jako interesujące i nietrywialne. Jednakże zarówno styl jak i liczne błędy zarówno drobne jak i te poważniejsze, znacząco utrudniają czytelnikowi zrozumienie zawartości tego rozdziału. Szczegóły zamieszczam w dalszej części, a tutaj przedstawię podsumowanie.

Autor włożył wiele pracy by zaprezentować wszystkie algorytmy w postaci tzw. pseudokodów, ale wybór zbyt niskopoziomowego formalizmu w efekcie doprowadził do zamieszczenia w rozprawie ponad 60 pseudokodów dla BLESS i RECKONER. Nie jestem przekonany, że wszystkie z nich są potrzebne, ale część z nich można było zaprezentować zwięźle z użyciem bardziej abstrakcyjnego języka. Zamieszczone pseudokody są słabo udokumentowane i przypominają przepisanie z kodu źródłowego instrukcje programu z pominięciem istotnych komentarzy. Choć w większości przypadków jest to kod dość elementarny, to w wersji przedstawionej w rozprawie trzeba włożyć wiele wysiłku by go zrozumieć.

P.C

Ponadto, poziom opisów tych kodów w rozdziale jest ogólnikowy. Niestety błędy oraz występujące rozbieżności między opisami a realizacją w kodzie, nie pomagają w ich zrozumieniu.

Przedstawione wyniki teoretyczne dotyczą analizy złożoności czasowej pesymistycznej przedstawionych algorytmów BLESS oraz RECKONER (brakuje analizy złożoności pamięciowej). Do każdego z algorytmów Autor przedstawił szereg lematów i twierdzeń. Wyprowadzenia własności złożoności są zasadniczno poprawne, choć budzi niepokój powielany wielokrotnie błąd w rozumieniu symbolu O-duże. Dodatkowo, kluczowe wyprowadzenie w Twierdzeniu 2 i 3 wymagają przedstawienia dowodu.

Przestawiony model błędów nie spełnia wymogu matematycznej poprawności. Choć można domyślić się jak ten model działa, istotne konstrukcje są niejasne, a poczynione niektóre założenia, które Autor przedstawia jako fakty bez dowodu, wydają się nieprawdziwe.

Autor nie przedstawia wystarczającej motywacji do wyboru algorytmu BLESS jako bazy dla algorytmu RECKONER.

W Rozdziale 6 opisane są przeprowadzone eksperymenty obliczeniowe. Mgr inż. Maciej Długosz przedstawia wyniki dla różnych wariantów danych wejściowych tj. symulowanych i rzeczywistych, oraz porównuje implementacje algorytmów z alternatywnymi narzędziami do korekcji błędów. Przedstawione są wyniki jakościowe i wydajnościowe. Eksperymenty zostały starannie zaprojektowane, przeprowadzone i są dobrze zilustrowane tabelami i wykresami. Wyniki pokazują, że opracowane narzędzie stanowi dobrą alternatywę wobec istniejących rozwiązań osiągając w niektórych kryteriach najlepsze rezultaty testów.

IV. Podsumowanie

Recenzowana rozprawa zawiera oryginalne rozwiązanie z zakresu przetwarzania danych pochodzących z sekwencjonowania DNA w postaci algorytmu korekcji wraz z implementacją i opracowaniem teoretycznych wyników złożonościowych. Wyniki zaprezentowane w pracy potwierdzają postawione tezy. Autor wykazał się bardzo dobrą znajomością stanu wiedzy o tematach podejmowanych w rozprawie badawczej, dotyczy to zarówno wiedzy o charakterze teoretycznym jak i dotyczącym genomiki i metod sekwencjonowania. Ponadto, mgr inż. Maciej Długosz posiada umiejętności formułowania problemów, projektowania i implementacji wydajnych algorytmów dla ich rozwiązywania oraz potrafi je formalnie zbadać i opisać, choć jakość tych opisów nie jest zawsze jest na

najwyższym poziomie.

Uzyskane wyniki zostały częściowo opublikowane w czasopiśmie Bioinformatics (o wysokim współczynniku IF) w roku 2017. Artykuł posiada aktualnie 20 cytowań wg Google Scholar. Mgr inż. Maciej Długosz jest także współautorem innych prac o narzędziach bioinformatycznych: KMC 3 (2017, Bioinformatics) i Kmer-db (Bioinformatics, 2019), które uzyskały łączną liczbę cytowań ponad 400 wg Google Scholar (2023) co świadczy o bardzo dobrym wpływie dziedzinę.

Rozdział 6, który dotyczy opracowanych metod i ich analizy, jakościowo wzbudził moje największe wątpliwości, które potwierdzam uwagami dołączonymi w dalszej części recenzji. Uważam, że uwagi są do poprawienia, choć niektóre mogą być czasochłonne. Najlepszym rozwiązaniem byłoby przedstawienie poprawionej rozprawy, ale wymagania dotyczące konkluzji recenzji nie dopuszczają takiej możliwości.

V. Konkluzja

Mimo tych krytycznych uwag, biorąc pod uwagę całościowo opracowane wyniki, opracowane narzędzie bioinformatyczne oraz bardzo dobre powiązane publikacje, stwierdzam, że recenzowana przeze mnie rozprawa doktorska spełnia wymagania stawiane rozprawom doktorskim przez obowiązujące przepisy i wnoszę o dopuszczenie magistra inżyniera Macieja Długosza o dalszych etapów przewodu doktorskiego.

Dodatkową rekomendacją (poza trybem), jest opracowanie przez mgra inż. Macieja Długosza odpowiedzi na uwagi krytyczne najlepiej w formie zaktualizowanego dokumentu rozprawy wraz ze szczegółowymi uzasadnieniami. To istotnie ułatwi dalszą pracę Komisji Doktorskiej i usprawni ewentualną obronę w przypadku dopuszczenia.

VI. Poprawność rozprawy i uwagi redakcyjne

Uwagi zamieszczone poniżej dotyczą głównie Rozdziału 5, który zawiera wyniki teoretyczne: opis BLESS, algorytm RECKONER oraz analizy złożoności pesymistycznej dla BLESS i RECKONER.

Uwagi ogólne dotyczące pseudokodów.

Brak opisu funkcji w pseudokodach.

Specyfikacja wejścia i wyjścia funkcji często bywa niekomplenta i jeśli jest zazwyczaj ogranicza się do przedstwienia listy nazw parametrów, np. nie ma typów, nie ma określenia ich znaczenia, nie jest jasne co robi funkcja.

Ograniczone komentarze w przedstawionym kodzie.

Kody zawierają duplikacje fragmentów, co więcej powielających błędy.

W pseudokodach jest sporo drobnych błędów i literówek, ale zdarzają się też poważne błędy.

Opisy pseudokodów w Rozdziale 5 wymagają rozwinięcia i są zbyt skrótowe, a zdarza się, że ustalone założenia i fakty przedstawione w nich rozmijają się z kodem.

Przyjęta notacja, odwołania od pseudokodów i ich umieszczenie w rozprawie utrudniają sprawne czytanie. Sugerowałbym w opisach użyć odwołań postaci "nazwa funkcji", a obok w nawiasie jej adres czyli nr pseudokodu wraz z numerem strony, np. "... w funkcji RATE (Pseudokod X, str. Y) jest ...". Podobnie warto zrobić w pseudokodach w formie komentarza, by sprawie zlokalizować wołaną funkcję.

Brakuje bardziej abstrakcyjnego matematyczno-algorytmicznego zapisu operacji na zmiennych, strukturach danych itp., w którym instrukcje można wyrażać bardziej zwięźle i bardziej przyjaźnie dla czytelnika, zachowując precyzję, choć tego typu zmiana zapewne nie jest możliwa przy tak dużej zawartości kodu.

Występują oryginalne instrukcje gdzie wynik wołania funkcji jest l-wartością w przypisaniu. Semantyka takich operacji nie jest przedstawiona.

Widoczność i definicje zmiennych są traktowane luźno, dotyczy to też parametrów funkcji.

Uwagi szczegółowe dot. pseudokodów.

Pseudokod 43. Pętla powinna być do $l - k$. Zgodnie z definicją regionu (A,B), B wskazuje na pierwszy symbol ostatniego k-mera należącego do B. Tutaj w linii 12 powinno być END(psi):=a-1 (jest a), bo k-mer na pozycji a jest już "untrusted".

Pseudokod 43. Jeśli w r jest ciąg k-merów "trusted" od pewnego miejsca do końca r (tj. l-k), to odpowiadający im region nie zostanie dodany do Psi.

Pseudokod 43. W linii 19: powinna być nierówność < zgodnie z definicją $\theta_{MNSOLID}$. Jeśli suffix r jest krótkim regionem "untrusted" powinien być dołączony do ostatniego "trusted", ale w liniach 23-30 tego nie ma.

Pseudokod 44. W linii 35 powinno być ψ zamiast ψ_p.

Strona 106. Algorytm z filtrem Blooma powinien być opisany z wyjaśnieniem.

Strona 104 i podobne. Funkcje boolowskie powinny być zdefiniowane bezpośrednio z return CONDITION.

Strona 107. Ostatni region untrusted w przykładzie powinien być $(b_2 + 1, l - k)$; jest $(b_2 + 1, l - k + 1)$.

Pseudokod 45. Wg pseudokodu skracanie dla regionu następuje tylko raz i dot. ostatniego k-mera, wtedy gdy wsród testowanych pozycji tego k-mera występuje jeden złej jakości nawet jeśli nie wystepuje na ostatniej pozycji. Ale jeśli symbol złej jakości nie jest ostatni, nadal ta pozycja będzie obecna w k-merach regionu. Opis na 106 stronie nie wyjaśnia dlaczego tak to działa (choć bardziej naturalne wydaje się usuwanie ostatniego k-mera jeśli jego ostatnia pozycja jest słabej jakości, bo wtedy ta ostatnia słaba pozycja nie będzie obecna).

Pseudokod 5. Linia 15. Nawiasowanie z BEG(..) niepoprawne. Występują powtórzenia kodu 12-16, 33-37, 43-47. Niejasny jest typ zmiennych i zwracanych wartości z funkcji. Brak opisów wyjścia i wejścia dla użytych funkcji. CORRECT3' jest wołane z Pseudokod z dwoma parametrami, a definicja funkcji ma parametr region Psi, czyli parę. Ale przy wołaniu koniec Psi jest przesunięty o k-1, czyli ostatnie k-mery regionu Psi są pominięte. Dlaczego?

Pseudokody 7 i 8. W opisie na stronach 110 i 111 podane są informacje o rozszerzeniu regionów, jednak w Pseudokodach 7 i 8 nie ma takiej operacji. Jedynie następuje sprawdzenie i usunięcie ścieżek z $\mathbf{\Pi}$, których nie da się rozszerzyć.

Pseudokod 8. Funkcja nie ma return $\mathbf{\Pi}$.

Pseudokod 12. Strona 117. Algorytm wyboru dwóch ścieżek o najmniejszym koszcie nie jest poprawny. Aktualizacja zmiennej q_{min2} jest tylko wtedy gdy q_{min} jest ustawiany, zatem q_{min2} nie będzie w ogólności reprezentować ścieżki o drugiej najmniejszej wartości. Np. jeśli ścieżki w $\mathbf{\Pi}$ są uporządkowane wg rosnących kosztów, to wtedy q_{min2} będzie $+\infty$. Błąd występuje zawsze jeśli element o drugiej wartości jest za elementem minimalnym (czyli dość często). Ponieważ ten fragment kodu opisuje program BLESS (autorem nie jest mgr inż. Długosz), pytanie czy rzeczywiście tak to jest zaimplementowane w BLESS, czy jest to błąd przy przepisywaniu kodu? To jest dość zaskakujące, bo problem jest prosty, a po drugie funkcja RATE jest kluczowa w ocenianiu ścieżek korekcji, zatem błąd tutaj wpływa istotnie na działanie tego programu.

Pseudokod 12. Strona 117. Komentarz pomiędzy 15/16. Przypuszczalna postać warunku zaproponowana mgr Długosza jest zawsze spełniona bo zachodzi jest $q_{min} <= q_{min2}$ i $\theta_{min\Delta} >= 0$. Podany warunek w linii 16 nie jest błędem implementacji i działa zgodnie z opisem na stronie 109 (z dokładnością do przypadków, który opisuję powyżej). Konflikt oznaczeń w linii 12: q pod sumą (zakładam wektor jakości) i zmienna q.

Pseudokod 9. Linia 7, PI_Q_LOW nie jest zdefiniowane. Zakładam, że to A_Q_LOW. Brak opisu parametrów. Wołanie CORRECTFIRST z Pseudokod 4 bez parametrów. Np. co to jest q i skąd jest q przy wołaniu? Czy |q|=|r|? W szczególności zakładam, że pozycje w A_Q_LOW muszą być z zakresu pierwszego k-mera wpp FIRSTBRUTEFORCE nie zadziała poprawnie. Brak informacji o takim założeniu.

Pseudokodów jest sporo w pracy i są umieszczone w miejscach, które nie jest łatwo odszukać (np. oddalonych o kilkanaście, czasem kilkaset stron). To istotnie utrudnia czytanie rozprawy. Sugeruję wpisanie nr strony, na której dany pseudokod jest umieszczony w miejscu gdzie jest on omawiany. Dodatkowo także w komentarzach gdzie występuje wołanie funkcji. Również w opisach jest odwoływanie do pseudokodów, a brakuje bezpośrednio użycia nazw funkcji, które realizują dane zadanie co również utrudnia odszukiwanie. Wskazane stosowanie hiperłącz w PDF.

Niekonsekwente pętle. Np. for z downto, albo z iteracją po zbiorze kolejnych

liczb, itd.

W opisanych pseudokodach jest sporo instrukcji redundatnych albo niepotrzebnych, np. zmienne zdefiniowane jednokrotnie, by je użyć tylko raz w kolejnej linii. Np. Pseudokod 9 linia 10 i 11, to samo w linii 12 i 13. Parameter I jest zdefiniowany na wejściu, ale nie jest użyty w Pseudokod 9. Linia 65 niepotrzebna w Pseudokod 10. Takich przykładów jest sporo.

Pseudokod 10. Linia 35. κ jest k-merem z poz. nr 1, (zatem drugim k-merem), ale pozycje w Π są wygenerowane dla oryginalnej sekwencji dla k-mera z pozycji 0. Skąd APPLY(k, Pi) wie jakie są poprawne indeksy do korekty? Zakładam, że kappa to k-mer z poz. 0.

Pseudokod 49 (RECKONER). Uwagi dot. Pseudokodu 43 też mają zastosowanie tutaj. Ten kod jest identyczny z Pseudokodem 43.

Pseudokod 50. Kod posiada błąd. Po zakończeniu pętli 20-26. Ψ_{trust} posiada tylko jeden element tj. MINREG(Ψ_{trust}), bo w linii 21 z Ψ_{trust} jest element iterowany Ψ. Z drugiej strony, po co jest ustawiany $short_{dist}$?

Pseudokod 51. Krok 0.2. Kopia kroku z 0.2 z BLESS. Ten sam problem - powinna być nierówność. Krok 0.3. Kopia kroku z 0.3 z BLESS. Analogiczny problem.

Pseudokod 52. i 53. Krok. 0.5. W opisie na str. 149 jest uwaga o dodatkowym skracaniu względem Pseudokod 45 z BLESS, ale kod kroku 0.5 Pseudokod 52 i Pseudokod 45 sa identyczny. Krok 0.7. To prawie kopia Pseudokod 45, ale problem opisany dla Pseudokodu 45 pozostaje. Ponadto występuje przypisanie na zmienne (max_{adj}), które nie są użyte, a powinny wpp pętle mogą wyjść przez BEG(Psi). W szczególnym przypadku może dać BEG(Psi)>END(Psi), albo ujemne pozycje w END(Psi). W linii 147 jest inny warunek, ale komentarz został skopiowany bez zmiany.

Pseudokod 17. Funkcja rate jest niezdefiniowana. Zakładam, że to jest RATE.

Pseudokod 25. CORRECTINTERNAL - funkcja ma parametr Psi czyli region, ale jest wołana z parą liczb. Nawet jeśli uznamy, że ta para przy wołaniu jest regionem, to podobnie jak przy CORRECT3' koniec jest przesunięty o k-1. Dlaczego? Czy te k-1 k-merów nie jest uzwględnianych w korekcji? Przypadek regionu z BEG()>END(), jest możliwy w takim przypadku, ale komentarz sugeruje jedną delecję w takim przypadku. To też wymaga wyjaśnienia.

Pseudokod 28. Zmienna Psi pojawia się w linii 144. Nie jest zadeklarowana jako zmienna lokalna, ani nie jest parametrem. O ile dobrze zauważyłem nie występuje także jako zmienna globalna.

Pseudokod 14. Pętla powinna być do |r*|-k". Obecnie jest |r*|-k"-1, zatem ostatni k"-mer w r* nie jest sprawdzany. Ten rodzaj błędu pojawia się też w kilku innych miejscach.

Pseudokody 15-17 + APPLYFINAL. Widoczność i definicje zmiennych są traktowane luźno, np. w APPLYFINAL powstaje nowa sekwencja z wektorem jakości, obecnie wynik jest przypisywany na parę (r,q) w Pseudokod 17, przy czym r jest na wejściu Pseudokod 15. Natomiast zmienna q nie jest zdefiniowana, co więcej q nie jest nawet zdefinionane jako wynik działania tego pseudokodu, czy zatem q jest ignorowane? Czy nie jest lepiej napisać wprost słowami, że APPLYFINAL(...) nadpisuje sekwencję r z jego kodem jakości jako wynik korekcji? Błędny nawias w linii 17.

Pseudokod 17. W VERIFYLONGKMERS w linii 83, jest problem z ι_{i-1} gdy i=0. Po ewentualnej korekcie, czy zatem region 0 nigdy nie zostanie wybrany? Poważniejszy błąd, to odwołanie do ι_{i-1} w regionie i. Przecież, ι_{i-1} daje indeksy

ścieżek z regionu i-1, a nie z regionu i, z którego są ścieżki w pi. Zakładam, zatem że powinno być $rate_\Delta := \pi_{\iota_i} - \pi_{\iota_i+1}$ z wymaganym dodatkowym warunkiem poprawności zakresu $\iota_i + 1$. Wtedy $rate_\Delta >= 0$, bo ścieżki w Π_i są posortowane względem wartości RATE i też instrukcja w linii 92 ma założony sens.

Pseudokod 21. Powtórzenia kodu: linie 62-75 i linie 83-95. Także 36-45 (bez korekty indeli). W linii 66 brak argumentu s.

Pseudokod 22-23. Duplikaty fragmentów kodu. EXPINDEL jest funkcją, która zwraca element krotki (tutaj wartość true lub false). W linii 21 na wynik wołania tej funkcji jest przypisana wartość true. Podobna notacja używana jest wielokrotnie (np. PATHPROB). Znów domyślam się co Autor zamierza wyrazić takim przypisaniem, ale stosowanie takiej notacji jest niepoprawne i mylące. Należy zastosować odpowiednie zmienne, albo opisać słowami operację.

Pseudokod 18. Funkcja RATE. Definicja funkcji oceniania (str. 135) uwzględnia czynnik $\theta_{INDPROB}^{n_{ind}}$ (wzory 5.19 i 5.20), gdzie n_{ind} to liczba indeli. Natomiast częściowo opisowa definicja funkcji RATE w Pseudokod 18 tego nie uwzględnia. Dodatkowo jest kolejna rozbieżność między RATE a tymi wzorami ze strony 5.19 i 5.20, chodzi o przypadek gdy $n_{ins} = |pi|$.

W niektórych pseudokodach jest obliczanie czynników występujących w podanym wzorze (np. CreatePathExtend3') i użyta jest stała $\theta_{INDPROB}$, stosując w/w notację przypisującą na wartość wołania funkcji wartości wyrażenia arytmetycznego np. PATHPROB(pi) <- .... Ale nawet interpretując te wyliczenia w postaci przypisań na odpowiednią zmienną (może element słownika?), jak te obliczone wartości są przekazywane do funkcji RATE, by później obliczyć ocenę ścieżki? W rozprawie nie znalazłem wyjaśnienia. Dodam, że funkcja CreatePathExtend3' i pokrewne zwracają ścieżkę korekcji, czyli listę krotek (choć to nie jest wprost napisane w tych pseudokodach, trzeba zagłębić się w sam kod by to wywnioskować).

Analiza złożoności.

Lemat 1. Strona 118. Wyprowadzenie równania rekurencyjnego i formuły zamkniętej jest poprawne, ale nie jest prawdą, że 3^n należy do $O(2^n)$. Ponadto, w dowodzie jest nieprawdziwa równość klas $O(2^n)$ i $O(4^n)$. Podobne i niepoprawne należenie do klasy $O(2^n)$ funkcji postaci 3^n i 4^n jest w sformułowaniach Lematu 2, Twierdzeniu 1 i w wielu kolejnych łącznie z podsumowującym Twierdzeniem 2. Dodatkowo, warto zwrócić uwagę, że w (5.4) są dwa parametry, ale tutaj też nie ma należenia do podanej klasy funkcji.

Lemat 6. Tutaj widać, że autor stosuje założenia o ograniczeniu pozycji z A_Q_LOW (tego założenia nie ma w pseudokodach).

Strona 123. Zamiast odsyłać czytelnika do tabelki, lepiej wprost podać te wartości parametrów θ.

Twierdzenie 2. Dowód nie przedstawia wyprowadzenia tej formuły, dlatego budzi ona pewne wątpliwości. Tu jest również problem z asymptotyczną klasą wyprowadzonych funkcji. W przypadku $k >= 21$, Autor pisze o złożoności $O(k + 2^l)$. Oczywiście przy uwzględnieniu poprawnej asymptotycznie klasy, tutaj zapewne pojawi się $O(k + k3^l)$. Ale w formule (5.14) mamy wyrażenie z $k3^{l-k}$, które daje główne szacowanie asymptotyczne i które w mojej opinii powinno zostać w takiej formie jeśli stosowane są parametry k i l. Co zatem stało się z wykładnikiem -k, którego uwzględnienie daje lepsze szacowanie niż podane przez Autora, tj. przy założeniu, że $k >= 21$, mamy $k3^{l-k} = (k/3^k)3^l < 3^l$ (zamiast $k3^l$)? Dlaczego podawać gorsze szacowanie? Sugeruję też podanie pełnego dowodu. Być może nie ma potrzeby rozróżniania przypadków z Lematów 12 i 13. Pozostawiam jeszcze pytanie jak założenie $k < l$ wpływa na wyprowadzoną

złożoność.

Lemat 15. Dowód. Strona 188. W dowodzie jest stwierdzenie, że zapytanie w linii 59 jest wykonywane trzykrotnie (właściwie <=), ale jesli symbol jest niskiej jakości albo k-mer jest "untrusted" to to sprawdzenie będzie wykonane <= 4 razy. Choć można to zoptymalizować do <=1+3 przy pewnych założeniach.

Uzasadnienie nierówności na funkcjach rekurencyjnych T''_{R3} i T'_{R3} nie jest wystarczające i opiera się o argumenty o szacowaniu głębokości tych równań rekurencyjnych, a nie ma porównania wartości tych funkcji. To za mało. Wystarczy tutaj zastosować prosty dowód indukcyjny z tezą $T''_{R3}(a,b) <= T'_{R3}(a+b)$ dla każdego a, b. Jedyna trudność to weryfikacja bazy indukcji.

Równanie na stronie 189. Wydaje się, że powinny być uwzględnione <=4+4 wołania CONTINUE'3 (4 z CORRECT3' oraz 4 z CORRECT3'INDEL). Brakuje 4 wołań. Sugerowałbym, też dokładniejsze uzasadnienie wyprowadzenia. Z drugiej strony, te wyprowadzenia nie są potrzebne. Teza z O-duże, wynika z faktu, że liczba zapytań do bazy jest szacowana przez $const * \theta_{MXCHECK}$ (każda dekrementacja licznika odpowiada <= stałej liczbie zapytań w tych kodach, a zerowanie licznika kończy rekursję w rozważanych funkcjach).

Lemat 16 i kolejne lematy i twierdzenia. Ponowne problemy z symbolem O: 4^n nie należy do $O(2^n)$.

Lemat 29. Strona 196. W szacowaniu powinno być $k * \theta_{MXPATH}$, jest +.

Algorytm RECKONER. Rozdział 5.4.1. Model błędów.

Przyjęty model błędów jest formalnie niezrozumiały. W szczególności nie jest jasne jak konkretnie sekwencja z genomu r' długości l' jest przekształcona w sekwencję o r długości l zgodnie z podanym opisem.

Symbole: const, true, false powinny być zdefiniowane. Również symbol a (domyślam się, że brakuje kwantyfikatora: dla każdego a ...). Po co jest użyty symbol const? Obecnie z niego wynika, że p_del = p_ins. Również symbol p (w $p(q[a])$) uzyty w f_{subst} (albo przynajmniej przypomnienie co oznacza), również q^* nie jest formalnie zdefiniowane, np. jakie wartości przyjmuje. Co to za zmienna losowa, która daje te wartości. Tutaj domyślam się, że chodzi o wektor jakości. Nie rozumiem zastosowania P_r w def. f_{del} (i podobnie f_{ins}). Zgodnie z tą definicją f_{del} jest funkcją o określonej dziedzinie i o określonym zbiorze wartości. Jaki jest związek z określaniem p-stwa dla wartości tej funkcji? Dodam, że f_{del} nie jest zmienną losową, co ewentualnie mogłoby sugerować zastosowanie tej notacji.

Strona 128. Zdanie, "f_{del} określa p-stwo usunięcia danego symbolu" jest skrótem myślowym, f_{del} przyjmuje wartości true i false, a nie wartości z przedziału $[0,1]$. Dodatkowo, dziedziną f_{del} są pozycje nie symbole. Podobnie f_{ins}.

Jak należy rozumieć: "Przekształcenie sekwencji r' w r jest ciągiem wartości funkcji f_{subst}, f_{del}, f_{ins} oraz ..."? Tj. jak dokładnie wygladają przekształcenia, które dają r? Z opisu w tym podrozdziale nie wynika jak uwzględnić stosowanie insercji i delecji, np. co pierwsze jest stosowane, jak interpretować reguły dodatkowe na zmieniających się zbiorach pozycji. Sugestia: opisać dokładnie algorytm generowania.

Strona 128/9. Uwaga dotycząca "założenia o prawdziwości uproszczeń". Niektóre z tych własności stanowią istotnie dodatkowe założenia (np. to o pokryciu odczytami), ale cześć z nich jest konsekwencją przyjętego modelu, zatem powinny być udowodnione, a nie podawane jako własności które zachodzą. Nie jestem też przekonany, że wszystkie są prawdziwe. Np. czy wobec reguł ograniczających występowanie insercji i delecji, prawdą jest, że zdarzenia tego

typu w ostatecznym modelu są niezależne od pozycji? Np. jeśli na pozycji b jest błąd to na pozycji a=b-1 nie może być błedu. Ponadto, na początku/końcu sekwencji być może jest większa szansa na zjawisko insercji np. jeśli p-stwo insercji, czyli p_{ins} jest 1, a sekwencja ma długość 3, to poz. 0 i 2 będą częściej z insercją niż poz. 1). Wymaga uzasadnienia pkt. 5 i 6 z tej listy. Ostatnia pozycja wymaga sprecyzowania działania modelu jak insercje i delecje są stosowane.

Część eksperymentalna.

Autor pokazuje i skrupulatnie uzasadnia jak wybrać parametr k, który istotnie jest kluczowym dla wszystkich przedstawionych narzędzi. Jednocześnie przedstawione narzędzia, w tym RECKONER posiadają także inne parametry, które mają wpływ na przebieg i jakość korekcji. Czy przeprowadzono testy nad innymi parametrami?

Drobna uwaga do diagramów. Symbole pokazujące niemożność przeprowadzenia analiz powinny być pokolorowane kolorem narzędzia z legendy, albo oś X powinna być wzbogacona o odpowiedni klucz.

Inne uwagi.

Strona 56. W pkt. 1 należy określić multizbiór K_N. W obecnej wersji może być to dowolny multizbiór k-merów spełniających podany warunek. Np. zbiór wszystkich k-merów spełnia ten warunek. Konflikt w definicjach. Zbiór $\mathcal{K}$ najpierw jest definiowany jako zbiór par (κ_i, η_i), gdzie κ_i to k-mer, a następnie jest nazwany "zbiorem k-merów" (po pkt. 2).

Pojęcie spektrum powinno być zdefiniowane.

W opisie algorytmu BLESS brakuje wyjaśnienia z motywacją konstrukcji poszczególnych kroków.

Warto zrobić diagram zależności funkcji algorytmu BLESS i RECKONER z podaniem gdzie dana funkcja jest zdefiniowana.

Strona 133. Jest "nowy symbol T", na rysunku jest G.

Strona 133. Opis zbioru K_π: dla a=0 otrzymujemy drugi przypadek (nie ma potrzeby wyróżniania).

Funkcja oceniania str. 135. We wzorze 5.21, co to jest p(q[a])? Tzn. jaka jest formalna definicja tego p-stwa.

"PATHPROB(π) to "Iloczyn prawdopodobieństw zmodyfikowanych symboli przyporządkowany ścieżce π" zdefinionany w Pseudokod 18. Jak rozumiem chodzi tutaj o iloczyn wartości prob(i) ze wzorów ze str. 135. Natomiast w tych w/w wyliczeniach pojawiają się $\theta_{INDPROB}$, który modeluje prawdopodobieństwo indeli. To dodatkowa niejasność.