Politechnika Śląska
Wydział Automatyki, Elektroniki i Informatyki
Informatyka Techniczna i Telekomunikacja

Rozprawa doktorska
Autoreferat

Korekcja danych z sekwencjonowania genomów

Maciej Długosz

Promotor: prof. dr hab. inż. Sebastian Deorowicz

Gliwice, 2023

Spis treści

1 Wprowadzenie

Proces sekwencjonowania DNA ma na celu uzyskanie ciągów symboli ze zbioru $\{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{T}\}$, reprezentujących sekwencję nukleotydów budujących analizowaną próbkę kwasu nukleinowego. Ciągi te są nazywane *odczytami* i we współczesnych technikach sekwencjonowania są pozyskiwane w praktyce z losowych miejsc w DNA. Długość odczytów jest zwykle zdecydowanie mniejsza od liczby nukleotydów budujących próbkę, a w szczególności mniejsza od długości pełnego genomu. Własność ta jest rekompensowana poprzez znaczną nadmiarowość danych sekwencjonowania — łączna długość wszystkich odczytów uzyskanych w eksperymencie może wielokrotnie (typowo dziesiątki do setek razy) przekraczać długość sekwencji DNA. Dokładna wartość ilorazu długości wszystkich odczytów do długości genomu jest nazywana *głębokością sekwencjonowania*. Dominującą na rynku metodą sekwencjonowania jest technika przedsiębiorstwa Illumina, dostępna w formie automatycznych urządzeń sekwencjonujących, nazywanych *sekwenatorami*.

Opracowanie algorytmów przetwarzania odczytów wymaga uwzględnienia ich własności. Należą do nich, poza niewielką długością i brakiem informacji o lokalizacji odczytu w próbce, także bardzo duży rozmiar zestawów odczytów, implikujący konieczność dostosowania kwestii technicznych algorytmu i jego implementacji, w szczególności potencjalnie duże zapotrzebowanie na czas obliczeń oraz pamięć. W wyniku sekwencjonowania genomów znacznej długości, np. genomu ludzkiego, uzyskuje się zbiory składające się z setek milionów odczytów długości kilkuset symboli. Przetworzenie danych o takiej skali wymaga staranności w doborze struktur danych, pozwalających na osiągnięcie rozsądnego zapotrzebowania na pamięć, a także położenia nacisku na szybkość działania, redukując go do akceptowalnego poziomu. Do osobnej grupy wyzwań należy zaliczyć te związane z aspektami jakościowymi odczytów. W praktyce rozkład pozycji sekwencji odczytów w próbce nie jest jednostajny, tzn. niektóre części sekwencji genomu są reprezentowane przez wiele odczytów, a inne przez pojedyncze lub wcale. Z kolei same odczyty charakteryzują się obecnością błędów, tzn. różnic między ciągami symboli uzyskanymi w wyniku sekwencjonowania, a oryginalną sekwencją nukleotydów. Powszechną praktyką jest dołączenie do symboli odczytów wartości nazywanych *współczynnikami jakości*, kodujących szacunkowe prawdopodobieństwo błędu danego symbolu.

Obecność błędów jest źródłem komplikacji zadań przetwarzania odczytów oraz niesie negatywne konsekwencje dla jakości uzyskiwanych wyników. Z drugiej strony efektem nadmiarowości danych jest fakt, że określony nukleotyd danego genomu zwykle jest reprezentowany w wielu odczytach. Błąd występujący w jednym z nich skutkuje powstawaniem niespójności danych, która musi zostać uwzględniona poprzez konsensus co do wartości rzeczywistego nukleotydu. Zadanie to

nie jest łatwe i jest przeprowadzane z ograniczoną skutecznością. Przykładowo, w zadaniu detekcji wariantów genomów, w którym określany jest zestaw różnic między genomem poddawanym sekwencjonowaniu a znanym wcześniej genomem odniesienia (*referencyjnym*), obecność błędu może być wykryta, a sam błąd pomięty, lub w przypadku niepowodzenia być uznany za wariant genomu.

Charakterystyka błędów wynika z wykorzystanej techniki sekwencjonowania. W przypadku odczytów z urządzeń marki Illumina, dominującym rodzajem błędów są *substytucje*, będące prostą zamianą symboli odpowiadających rzeczywistym nukleotydom na inne. Osobną grupą błędów, występujących rząd wielkości rzadziej, są błędy typu *insercja* oraz *delecja* (łącznie nazywane błędami typu *indel*), objawiające się odpowiednio wstawieniem dodatkowych symboli do odczytu lub całkowitym brakiem pewnych symboli. W przypadku techniki Illumina błędy typu indel w zdecydowanej większości przypadków dotyczą pojedynczych symboli.

W literaturze opisano liczne algorytmy wyspecjalizowane wyłącznie do detekcji i eliminacji błędów w odczytach, bez dalszego przetwarzania odczytów, wykorzystując szereg informacji, w szczególności nadmiarowość danych w odczytach. Do zalet wykorzystania takich algorytmów należy możliwość dostosowania odczytów do różnych algorytmów lub potoków dalszego przetwarzania, usprawniając ich działanie poprzez dostarczenie danych o mniejszej liczbie błędów.

Do opracowanych dotychczas algorytmów korekcji odczytów z sekwenatorów Illumina lub algorytmów częściowo uniwersalnych ze względu na charakterystykę odczytów należą m.in. Musket [19], RACER [15], BLESS [12, 13], Fiona [21], Blue [10], Lighter [22], BFC [18], Karect [2] oraz SAMDUDE [9]. Zostały one oparte na kilku różnych modelach komputerowej reprezentacji odczytów wejściowych oraz odmiennych strategiach detekcji i eliminacji błędów. Najpopularniejszą metodą modelowanie danych zawartych w odczytach jest forma *spektrum k-merów*, czyli zbioru podsłów (oligomerów) odczytów długości k, opierając się na obserwacji, że dobierając wartość k odpowiednio mniejszą od długości odczytu, większa część k-merów w odczytach jest sekwencjami pozbawionymi błędów. Ponadto k-mer o pewnej danej sekwencji z dużym prawdopodobieństwem będzie obecny w wielu odczytach, z kolei k-mer będący podsekwencją odczytu z błędem w zdecydowanej większości przypadków pojawi się tylko w jednym odczycie (przyjmując, że w wyniku generacji wielu odczytów z tego samego fragmentu próbki nie wystąpi błąd w tym samym miejscu i o takiej samej charakterystyce).

Algorytmy korekcji różnią się stopniem dogłębności w podejściu do zadania: tylko w niektórych przypadkach w jawny sposób uwzględniono korekcję błędów typu indel lub wykorzystanie współczynników jakości symboli. W publikacjach przeglądowych [20, 23] wykazano, że algorytmy cechują się rozmaitą skutecznością działania, różnym zapotrzebowaniem na zasoby oraz praktycznymi właściwościami technicznymi. Ponadto niektóre modele odczytów zawierają niewielką

ilość informacji, w wielu przypadkach ograniczając się do przechowania binarnej informacji o obecności k-meru w modelu, nieraz w sposób uproszczony przez zastosowanie jako struktury danych filtrów Blooma.

W literaturze ocena skuteczności korekcji jest wykonywana przy pomocy kilku strategii: komputerowej symulacji odczytów, korekcji odczytów uzyskanych z sekwenatorów i obserwacji ich wpływu na zadania asemblacji *de novo* lub mapowania, a także biorąc pod uwagę kryteria związane z szybkością pracy algorytmu i zapotrzebowaniem na pamięć. W niniejszej pracy oparto się na wybranych istniejących protokołach oceny i wprowadzono nowe, w szczególności polegające na obserwacji wpływu korekcji na detekcję wariantów genomów. Tego rodzaju zadanie było rozpatrywane w literaturze tylko w bardzo ograniczonym zakresie i w innych formach. Jednocześnie, ze względu na zastosowanie potoku przetwarzania o wysokim potencjale praktycznym, pozwala na ocenę korekcji w sposób nastawiony na maksymalną użyteczność.

W pracy podjęto analizę zagadnienia korekcji odczytów, z postawieniem szczególnego nacisku na odczyty uzyskane z sekwenatorów marki Illumina. Dokonano przeglądu literatury pod kątem istniejących algorytmów oraz metod oceny ich skuteczności. Stwierdzono, że istnieje potencjał do opracowania nowego algorytmu, spełniającego szereg wymagań, wynikających z analizy i obserwacji istniejących algorytmów. W tym celu dokładnej analizie poddano algorytm BLESS, którego ogólna zasada działania została zaadaptowana w celu opracowania nowego algorytmu RECKONER.

Tezy pracy były następujące:

1. zastosowanie niektórych nowoczesnych metod korekcji odczytów sekwencjonowania genomów pozwala na redukcję liczby błędów powstałych w trakcie sekwencjonowania oraz poprawę jakości różnych wyników algorytmów przetwarzania odczytów, wliczając w to detekcję wariantów genomów przy niskiej głębokości sekwencjonowania,

2. możliwe jest opracowanie nowego algorytmu korekcji, zapewniającego lepszy od istniejących algorytmów bilans skuteczności do zapotrzebowania na zasoby, a przy tym niskie ryzyko degradacji jakości danych.

Szczegółowymi celami pracy były:

- opracowanie nowego algorytmu, opierając się na postawionych wymaganiach dotyczących jakości, szybkości działania i zapotrzebowania na pamięć oraz doświadczeniach płynących z analizy innych algorytmów, a także wyznaczenie złożoności obliczeniowej tego algorytmu,

- przeprowadzenie badań eksperymentalnych, polegających na korekcji odczytów uzyskanych z rzeczywistych urządzeń sekwencjonowania (*odczyty*

rzeczywiste) oraz obserwacji wpływu korekcji na ich późniejsze wykorzystanie; eksperymenty miały pozwolić na wielokryterialną ocenę nowego algorytmu oraz wzajemne porównanie grupy innych algorytmów,

- w ramach powyższego celu podjęto nowatorską próbę oceny skuteczności zadania korekcji, poprzez obserwację wpływu korekcji odczytów na proces detekcji wariantów genomów,
- przeprowadzenie eksperymentalnej oceny algorytmów korekcji, opierając się na odczytach symulowanych komputerowo, oceniając jednocześnie adekwatność uzyskanych wyników w stosunku do pozostałych wyników.

Zaprezentowany algorytm oraz wyniki eksperymentów zostały do tej pory opublikowane m.in. w formie artykułów (w tym jednego w trakcie procesu recenzji) [7, 8] oraz rozdziałów w monografiach [5, 6].

2 Opracowany algorytm

W ramach prac przygotowawczych do opracowania nowego algorytmu analizie poddano istniejący algorytm BLESS (opis w podrozdziale 5.3 pracy), wraz z jego implementacją. Stwierdzono, że może on stanowić bazę do opracowania lepszego rozwiązania, poprzez wykorzystanie ogólnej zasady działania oraz niektórych organizacyjnych etapów.

Algorytm BLESS jest oparty na spektrum k-merów. Jego działanie polega na analizie k-merów w odczytach wejściowych i wyznaczeniu pewnego *progu obcięcia* częstości ich wystąpień. k-mery obecne w odczytach mniejszą liczbę razy niż wartość progu są traktowane jako błędne (*niezaufane*). Informacja o pozostałych k-merach, uznanych za prawidłowe (*zaufane*), jest zachowywana w formie filtru Blooma. Następnie odczyty są poddawane niezależnej korekcji, która polega na wyodrębnianiu z nich kolejnych k-merów i weryfikacji ich obecności w spektrum. Stwierdzenie, że pewna grupa k-merów nie jest zaufana stanowi podstawę do wskazania w odczycie błędnego *regionu*, czyli ciągu kolejnych symboli odczytu, w których mogą być obecne błędy. Każdy region jest poddawany korekcji przy pomocy algorytmu z powrotami.

Analiza algorytmu BLESS pozwoliła na wyznaczenie kwestii, które powinny zostać uwzględnione w przygotowaniu nowego algorytmu RECKONER (opis algorytmu w podrozdziale 5.4). Pierwszym, kluczowym aspektem jest modyfikacja głównej struktury danych. Filtr Blooma pozwala na zachowanie w pamięci tylko binarnej informacji o obecności k-meru w spektrum, obarczonej na dodatek pewnym prawdopodobieństwem zgłoszenia fałszywej odpowiedzi twierdzącej. W związku z tym filtr został zamieniony na strukturę danych generowaną przez narzędzie zliczania częstości wystąpień k-merów KMC oraz narzędzia zarządzania

zbiorami odczytów KMC tools [17]. Wraz z nimi dostępny jest interfejs programistyczny, pozwalający na łatwe przeszukiwanie struktury danych, w której zawarte są zarówno pełne sekwencje k-merów, jak też częstości ich wystąpień (liczebności). Ulepszeniu została poddana także metoda określania progu obcięcia.

W samej koncepcji algorytmu z powrotami uwzględniono kilka rozwiązań, których obecność w innych algorytmach jest autorowi nieznana. Proces korekcji błędów typu substytucja odbywa się poprzez przechodzenie do kolejnych pozycji błędnego regionu odczytu oraz ewentualnym wprowadzaniu zmian ich symboli, co odpowiada przechodzeniu w drzewie reprezentującym przebieg algorytmu w kierunku liści (i ewentualnym cofaniu się). Postęp korekcji stanowi podstawę do wykrycia potencjalnych błędów typu indel, sugerowanych przez wystąpienie korekcji błędów typu substytucja kilku sąsiednich pozycji regionu. Wykrycie tego rodzaju sytuacji powoduje dodanie kolejnych możliwości korekcji regionu (pojawienie się nowych gałęzi do drzewa), obejmujących próbę wstawienia bądź usunięcia symbolu z regionu.

W ramach korekcji dopuszczalne są sytuacje, gdy przeprowadzenie korekcji spowoduje wygenerowanie i zamieszczenie w odczycie wyjściowym niezaufanych k-merów, zakładając spełnienie kilku warunków. Ponadto dokładniejszej korekcji od pozostałych pozycji poddawane są te, dla których wykryto obecność współczynnika jakości o bardzo niskiej wartości. Te dwie strategie, w określonych okolicznościach, pozwalają na dalszą poprawę skuteczności korekcji.

Ze względu na wykładniczą maksymalną liczbę sprawdzanych możliwości w algorytmie z powrotami, wprowadzono kilka limitów, dotyczących m.in. liczby: ścieżek, błędów typu indel w regionie i podejmowanych prób korekcji w regionie. Dobrane górne wartości ograniczają pracę algorytmu tylko w niewielkiej części odczytów, których korekcja stanowi szczególnie trudne zadanie. Celem limitowania jest redukcja czasu pracy algorytmu, zmniejszenie zapotrzebowania na pamięć (poprzez eliminację przypadków, gdy dla danego regionu istnieje możliwość zaproponowania bardzo dużej liczby *ścieżek korekcji*, czyli potencjalnych ciągów zmian w regionie) oraz rezygnacja z korekcji odczytów, gdy jej powodzenie jest wątpliwe. Duża szybkość algorytmu jest zapewniana także przez równoległe przetwarzanie odczytów.

W algorytmie zastosowano metodę oceny ścieżek korekcji dla danego regionu odczytu, pozwalającą na wybranie jak najlepszej ścieżki spośród wielu zaproponowanych w wyniku pracy algorytmu z powrotami. Jej idea opiera się na preferowaniu ścieżek o jak najmniejszej liczbie zmian oraz tych, w których zmiany w większej mierze dotyczą pozycji, którym przyporządkowane są mniejsze wartości współczynników jakości. Ponadto wyżej oceniane są ścieżki, które skutkują uzyskaniem k-merów w wyjściowym odczycie o wyższych liczebnościach (tj. częściej pojawiających się w odczytach wejściowych). Ocena jest wyznaczana zgodnie

z równaniem:

$$r_{\pi,3'}(a,b,\mathcal{K}_\pi) = \frac{\sum\limits_{\kappa\in\mathcal{K}_\pi} \text{weight}(\kappa)\eta(\kappa)}{\sum\limits_{\kappa\in\mathcal{K}_\pi} \text{weight}(\kappa)} \left(\prod_{i=a+k-1}^{b+k-1} \text{prob}(i) \right) (\theta_{\text{IND\_PROB}})^{n_{\text{ind}}},$$

gdzie a, b — odpowiednio początkowy i końcowy indeks błędnego regionu w odczycie, $\mathcal{K}_\pi$ — zbiór k-merów w regionie po wprowadzeniu zmian ze ścieżki, $\eta(\kappa)$ — liczebność k-meru κ, weight(κ) — współczynnik wagowy, zależny od lokalizacji k-meru, prob(i) — prawdopodobieństwo obecności błędu typu substytucja na pozycji i (uzyskane m.in. w oparciu o współczynnik jakości), n_{ind} — liczba skorygowanych błędów typu indel w regionie, $\theta_{\text{IND\_PROB}}$ — przybliżone prawdopodobieństwo pojawienia się błędu typu indel na zadanej pozycji.

Dodatkową, fakultatywnie wykonywaną metodą poprawy jakości korekcji jest weryfikacja odczytów wyjściowych w oparciu k''-mery, tzn. oligomery długości $k'' > k$. Jest ona proponowana głównie dla korekcji odczytów mających zostać poddanym asemblacji *de novo*. Metoda polega na obserwacji, czy odczyt wyjściowy składa się wyłącznie z k''-merów obecnych przynajmniej raz w odczytach wejściowych. Jeżeli nie, następuje dobór innego zestawu ścieżek korekcji regionów odczytu.

Praca algorytmu RECKONER wymaga określenia długości oligomeru, tj. wartości k. Może być ona wyznaczona w sposób automatyczny, poprzez wyznaczenie wartości k_{pred}:

$$k_{\text{pred}} = \max(20; 0{,}9 \log_2 \widehat{\ell_{\mathcal{G}}} + 3),$$

gdzie $\widehat{\ell_{\mathcal{G}}}$ jest przybliżoną długością genomu, zadaną przez użytkownika lub oszacowaną w oparciu o zestaw odczytów wejściowych. Parametry równania zostały wyznaczone empirycznie, w wyniku korekcji kilku zestawów odczytów dla ciągu różnych wartości k i doboru wartości, dla których uzyskano najlepsze wyniki. Metoda parametryzacji równania została wyjaśniona w podrozdziale 6.3 pracy.

3 Analiza złożoności obliczeniowej

W ramach prac wyznaczono złożoności obliczeniowe algorytmów BLESS (jako punkt odniesienia) oraz algorytmu RECKONER, w obu przypadkach zakładając sytuację korekcji odczytu, w którym wyznaczono jeden błędny region obejmujący cały odczyt. W obu przypadkach zadanie polega na korekcji pierwszego k-meru w odczycie (pierwszych k symboli odczytu), a następnie pozostałej jego części algorytmem z powrotami. Jako operację dominującą przyjęto uwzględnienie jednego symbolu k-meru przy wyznaczaniu wartości wartości jednej funkcji mieszającej filtru Blooma (BLESS) oraz porównanie jednego symbolu k-meru z symbolem w bazie k-merów (RECKONER). Wyprowadzania złożoności oraz

postawione założenia zostały przedstawione odpowiednio w podrozdziałach 5.2.2 oraz 5.3.

Złożoność obliczeniowa algorytmu BLESS wynosi:

$$T_{\mathrm{B}}(k,\ell,\theta_{\mathrm{MX\_E}},\theta_{\mathrm{MX\_LQ}}) = \begin{cases} dkT_{\mathrm{Bb}}(k,\ell,\theta_{\mathrm{MX\_E}},\theta_{\mathrm{MX\_LQ}}) = O\left(k2^{\ell}\right), \text{ gdy } k \leqslant 21, \\ dkT_{\mathrm{Bc}}(k,\ell,\theta_{\mathrm{MX\_E}},\theta_{\mathrm{MX\_LQ}}) = O\left(k^2 + k^2 2^{\ell}\right), \text{ gdy } k > 21. \end{cases} \tag{3.1}$$

Funkcje $T_{\mathrm{Bc}}(\cdot,\cdot,\cdot,\cdot)$ oraz $T_{\mathrm{Bc}}(\cdot,\cdot,\cdot,\cdot)$ są zdefiniowane następująco:

$$T_{\mathrm{Bb}}(k,\ell,\theta_{\mathrm{MX\_E}},\theta_{\mathrm{MX\_LQ}}) = \\ = \frac{1}{3}4^{\theta_{\mathrm{MX\_LQ}}}\left[4^{\theta_{\mathrm{MX\_E}}+1} + 3^{\ell-k-1}\left(4^{\theta_{\mathrm{MX\_E}}+2}+8\right) - 1\right] - 4,$$

$$T_{\mathrm{Bc}}(k,\ell,\theta_{\mathrm{MX\_E}},\theta_{\mathrm{MX\_LQ}}) = \\ = 4^{\theta_{\mathrm{MX\_LQ}}} + 4^{\theta_{\mathrm{MX\_E}}+1}\left(\frac{1}{3}+k+\theta_{\mathrm{MX\_E}}\right) + k\cdot 3^{\ell-k-1}\left(4^{\theta_{\mathrm{MX\_E}}+2}+8\right) - 13k - \frac{4}{3},$$

gdzie: ℓ — długość odczytu, $\theta_{\mathrm{MX\_E}}$ — maksymalna liczba symboli dołączanych do regionu w celu weryfikacji korekcji jego końcowych symboli, $\theta_{\mathrm{MX\_LQ}}$ — maksymalna liczba uwzględnionych symboli o niskich wartościach współczynnika jakości (w korekcji pierwszego k-meru), $d = O(1)$ — liczba funkcji mieszających filtru Blooma. Powyższe równania zostały udowodnione w formie twierdzenia 3 oraz lematów 12 i 13. W celu uzyskania postaci równania 3.1 zostały podstawione dokładne wartości stałych $\theta_{\mathrm{MX\_E}}$ oraz $\theta_{\mathrm{MX\_LQ}}$, stąd też rząd złożoności obliczeniowej nie jest od nich zależny.

Złożoność obliczeniowa algorytmu RECKONER wynosi:

$$\begin{aligned} &T_{\mathrm{R}}(k,\ell,\theta_{\mathrm{MX\_CHCK}},\theta_{\mathrm{MX\_E}},\theta_{\mathrm{MX\_FIRST\_PTHS}},\theta_{\mathrm{MX\_IND}},\theta_{\mathrm{MX\_LQ}}) = \\ &= kT_{\mathrm{Ri}}(k,\ell,\theta_{\mathrm{MX\_CHCK}},\theta_{\mathrm{MX\_E}},\theta_{\mathrm{MX\_FIRST\_PTHS}},\theta_{\mathrm{MX\_IND}},\theta_{\mathrm{MX\_LQ}}) = \\ &= O(k^2). \end{aligned} \tag{3.2}$$

Funkcja $T_{\mathrm{Ri}}(\cdot,\cdot,\cdot,\cdot,\cdot,\cdot,\cdot)$ jest zdefiniowana następująco:

$$T_{\mathrm{Ri}}(k,\ell,\theta_{\mathrm{MX\_CHCK}},\theta_{\mathrm{MX\_E}},\theta_{\mathrm{MX\_FIRST\_PTHS}},\theta_{\mathrm{MX\_IND}},\theta_{\mathrm{MX\_LQ}}) = \\ = 4^{\theta_{\mathrm{MX\_LQ}}} + \frac{20}{3}\left[(k-\theta_{\mathrm{MX\_E}}-1)\left(4^{\theta_{\mathrm{MX\_E}}}-1\right) - \theta_{\mathrm{MX\_E}} - \frac{1}{3}4^{\theta_{\mathrm{MX\_E}}}\right] + \\ +(5k-5)\cdot \\ \cdot\left[\min\left(\theta_{\mathrm{MX\_CHCK}};\frac{20}{3}\left(19^{\ell-k+\theta_{\mathrm{MX\_IND}}-1}-1\right)+\frac{5}{6}\left(19^{\ell-k+\theta_{\mathrm{MX\_IND}}-2}-1\right)+8\right)+\right. \\ \left.\frac{1}{3}\min\left(\theta_{\mathrm{MX\_PTH}};9\cdot 8^{\ell-k+\theta_{\mathrm{MX\_IND}}-1}\right)\left(4^{\theta_{\mathrm{MX\_E}}}-4\right)\right] + 8k - \frac{65}{9},$$

gdzie: $\theta_{\mathrm{MX\_CHCK}}$ — limit liczby podejmowanych prób korekcji w jednym regionie, $\theta_{\mathrm{MX\_FIRST\_PTHS}}$ — limit liczby ścieżek korekcji pierwszego k-meru, $\theta_{\mathrm{MX\_IND}}$ —

limit liczby korekcji błędów typu indel w regionie. Powyższe równania zostały udowodnione w formie twierdzenia 6 oraz lematu 29. W celu uzyskania postaci równania zostały podstawione dokładne wartości stałych $\theta_{\text{MX\_E}}$, $\theta_{\text{MX\_LQ}}$, $\theta_{\text{MX\_CHCK}}$ oraz $\theta_{\text{MX\_FIRST\_PTHS}}$, $\theta_{\text{MX\_IND}}$, stąd też rząd złożoności obliczeniowej nie zależy od nich.

Wyprowadzenie złożoności wymagało postawienia licznych założeń. Ponadto fakt, że rząd złożoności obliczeniowej algorytmu RECKONER jest zależny (w korzystnej proporcji) wyłączenie od zmiennej k, wynika z faktu ograniczenia liczby wykonywanych w algorytmie operacji przez kilku stałych limitów, w efekcie unikając wykładniczej złożoności obliczeniowej w funkcji długości odczytu. Obie powyższe przyczyny każą podkreślić, że rząd złożoności stanowi bardzo uproszczone podsumowanie charakterystyki algorytmu.

4 Wyniki eksperymentów

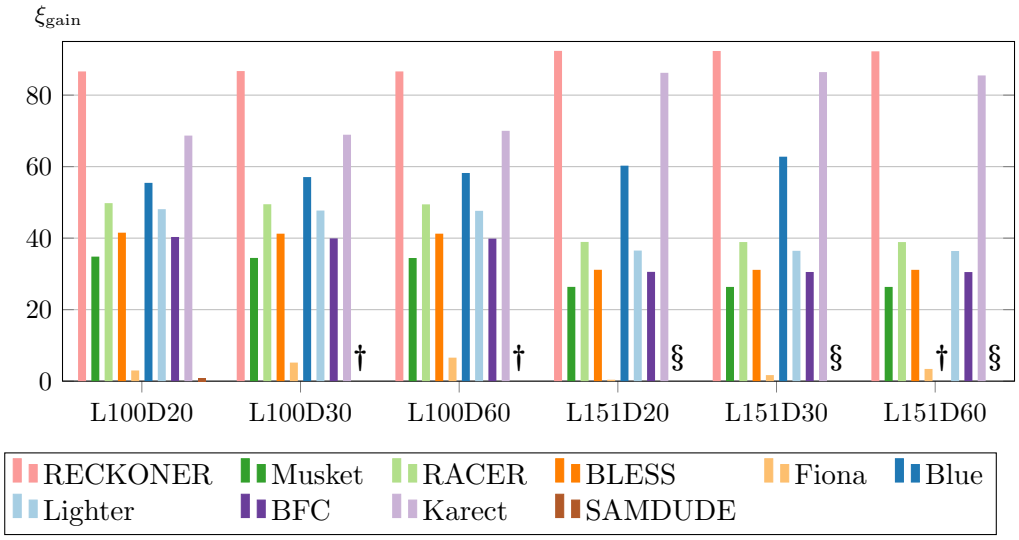
Kryteria oceny algorytmów zostały podzielone na trzy grupy: jakościowe (opis w podrozdziale 6.2), wydajnościowe (6.4) oraz techniczne (6.5). Analiza jakościowa polegała na przeprowadzeniu korekcji odczytów przy pomocy różnych algorytmów oraz obserwacji jakości uzyskanych rezultatów. W tym przypadku wyróżniono dwa podejścia: korekcja odczytów symulowanych oraz rzeczywistych.

Poniżej zawarto wybór przedstawionych w pracy wyników. Jeśli nie zaznaczono inaczej, dotyczą one odczytów genomu *Musa acuminata* długości ok. 460 Mpz (mega par zasad). Na przedstawionych dalej wykresach niektóre słupki zostały zastąpione symbolami, oznaczającymi różne przyczyny niepowodzenia przeprowadzenia korekcji.

Odczyty symulowane zostały uzyskane w wyniku wykorzystania dwóch strategii: *metody Quake*, polegającej na losowym wyborze krótkich podsłów genomu referencyjnego i wprowadzeniu do nich błędów w oparciu o współczynniki jakości rzeczywistych odczytów, oraz wykorzystania specjalizowanego narzędzia ART [14]. Eksperymenty przeprowadzono m.in. dla dwóch średnich wartości prawdopodobieństwa błędów symboli, kilku wartości głębokości sekwencjonowania oraz dwóch dobranych długości odczytów. Skuteczność korekcji jest wyrażona przy pomocy miary $\xi_{\text{gain}} = \frac{\text{TP} - \text{FP}}{\text{TP} + \text{FN}}$, gdzie TP — liczba odczytów zawierających błędy, które zostały prawidłowo skorygowane, FP — liczba odczytów niezawierających błędów, które zostały niesłusznie zmodyfikowane przez algorytm korekcji, FN — liczba odczytów zawierających błędy, które nie zostały skorygowane lub zostały skorygowane błędnie. Wyznaczenie tej miary, z racji znajomości prawidłowej sekwencji symulowanych odczytów, jest możliwe w bezpośredni sposób.

Na rys. 4.1 przedstawiono wyniki odczytów symulowanych przy pomocy metody Quake, z kolei na rys. 4.2 — narzędziem ART (wyniki przemnożono przez

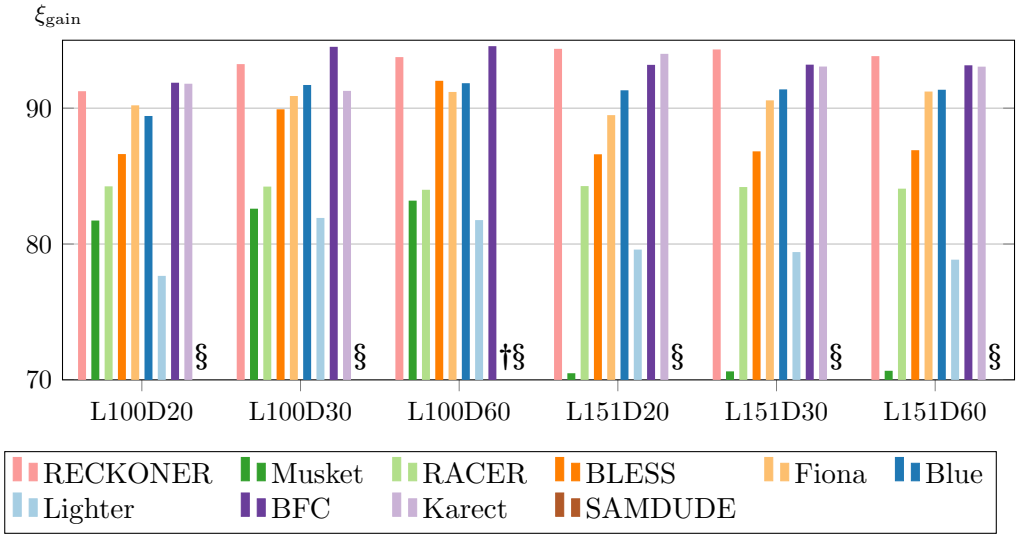
100). Przedstawione wyniki dotyczą przypadku, gdy średnie prawdopodobieństwo błędu wynosi $p_{\text{mean}} = 4$–5% (tj. 4% do 5%), z kolei notacja LxDy oznacza długość odczytu równą x oraz głębokość sekwencjonowania y. Wyniki uzyskane poprzez wykorzystanie odczytów symulowanych dwiema metodami różnią się w umiarkowanym stopniu. W prawie wszystkich przypadkach najlepsze wyniki uzyskano dla algorytmu RECKONER albo Karect, a dla odczytów symulowanych narzędziem ART często także BFC. Najlepsze algorytmy w niektórych przypadkach pozwalają na skuteczną korekcję więcej niż 90% odczytów zawierających błędy (w wybranych, nieprzedstawionych w niniejszym autoreferecie przypadków — prawie 100%).



Rysunek 4.1: Wpływ korekcji na wartość ξ_{gain} dla odczytów symulowanych uproszczoną metodą Quake, $p_{\text{mean}} = 4$–5%

Eksperymenty, w których wykorzystano odczyty rzeczywiste, zostały przeprowadzone zgodnie z następującą metodą. Wejściowy zestaw odczytów był poddawany korekcji różnymi algorytmami, a uzyskany rezultat przekazywany do algorytmu albo potoku wykonującego typowe zadanie przetwarzania odczytów. Następnie wynik był poddawany ocenie. Dodatkowo, jako przypadek kontrolny, na wejście algorytmu albo potoku niezależnie przekazywano zestaw odczytów niepoddanych korekcji (na wykresach oznaczony jako odczyty surowe).

Pierwszym algorytmem wykorzystującym odczyty był asembler *de novo*, przeprowadzający proces wzajemnego dopasowania odczytów i łączenia ich, mając na celu uzyskanie możliwie pełnej postaci genomu analizowanego organizmu. W praktyce wynikiem jest zestaw *kontigów*, czyli sekwencji połączonych odczytów, w idealnym przypadku reprezentujących sekwencje chromosomów, choć typowo tylko ich fragmenty. Asemblacja była przeprowadzana przy pomocy algorytmu Minia [4], a ocena uzyskanych wyników — przy pomocy narzędzia Qu-

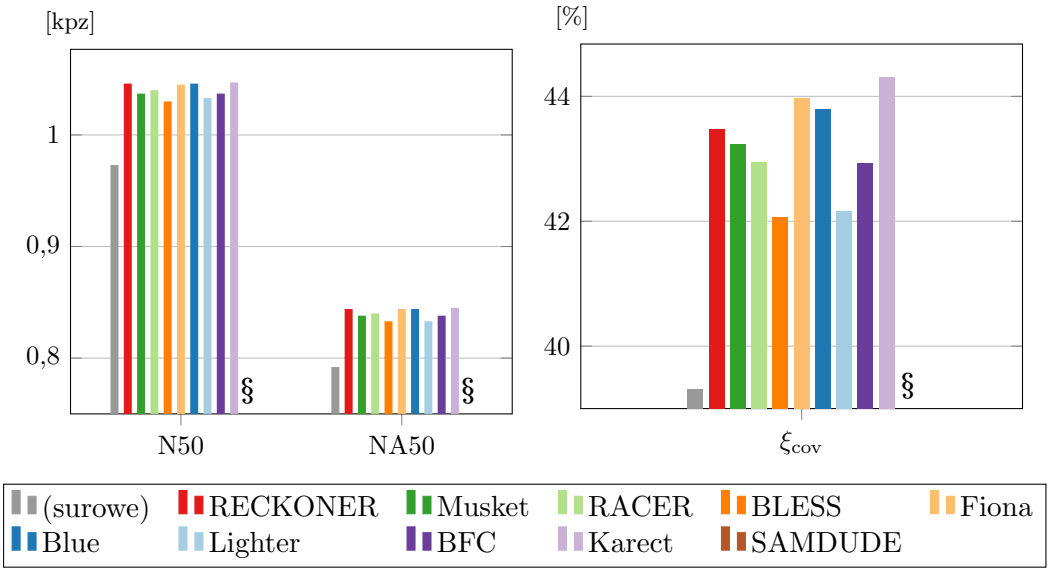


Rysunek 4.2: Wpływ korekcji na wartość ξ_{gain} dla odczytów symulowanych narzędziem ART, $p_{\text{mean}} = 4$–5%

ast [11]. Na rys. 4.3 przedstawiono wybrane spośród uzyskanych wyników, wyrażone w formie trzech miar (dobór innego koloru słupków odpowiadających wynikom algorytmu RECKONER jest efektem uruchomienia go w adekwatnym do tego zastosowania trybie z weryfikacją w oparciu k''-mery). Wartość N50 rozumiana jest jako taka długość kontigu, że suma długości kontigów posiadających długość N50 lub większą stanowi 50% sumy długości wszystkich kontigów. Miara NA50 jest definiowana analogicznie, jednak przed wyznaczeniem jej wartości kontigi zawierające błędy asemblacji są dzielone w miejscach wystąpienia błędów. Z kolei pokrycie kontigami ξ_{cov} jest rozumiane jako stosunek liczby symboli dopasowanych do genomu do długości genomu.

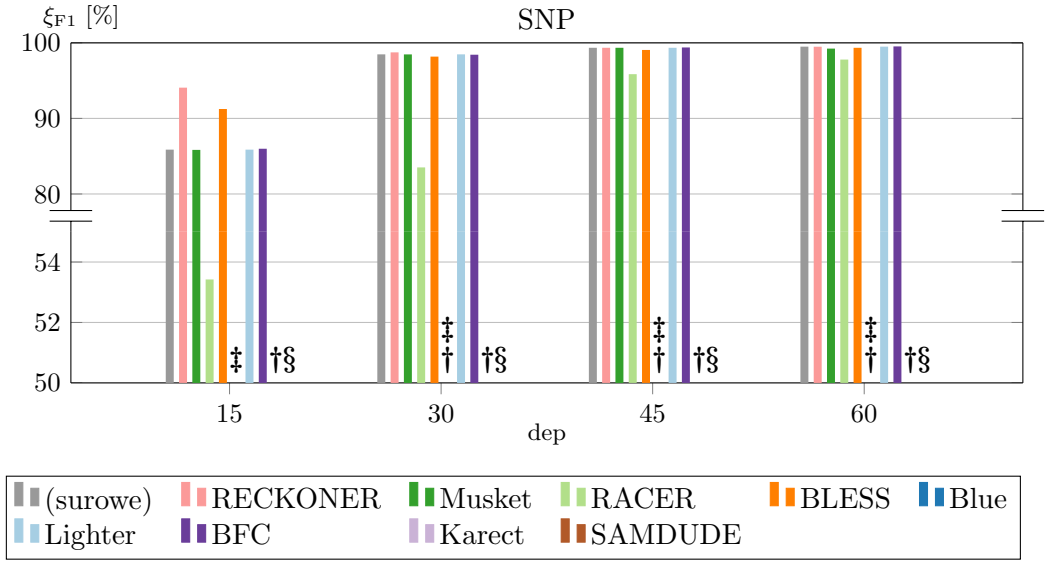
Alternatywnym podejściem była korekcja odczytów w celu wykorzystania ich w procesie detekcji wariantów, przeprowadzonym za pomocą potoku Strelka [16]. Uzyskane zbiory wariantów zostały poddane ocenie przy pomocy m.in. narzędzia Haplotype VCF comparison tools (hap.py) [1], poprzez porównanie ze znanym zestawem prawdy podstawowej.

Na rys. 4.4 przedstawiono wyniki dla odczytów ludzkiego genomu, wyrażone w formie miary F1 (tj. średniej harmonicznej czułości i precyzji) ξ_{F1} detekcji wariantów pojedynczych nukleotydów (SNP) dla różnych głębokości sekwencjonowania. W wielu przypadkach korekcja nie przynosi wyraźnej zmiany jakości detekcji wariantów. Wśród wariantów typu SNP korzystnie wyróżniają się algorytmy RECKONER oraz BLESS, które pozwalają na znaczącą poprawę detekcji przy głębokości sekwencjonowania równej 15×, w przypadku algorytmu RECKONER wynoszącą ok. 10%. Poprawa jest nieznacznie widoczna także dla algorytmu RECKONER i głębokości 30×. Korzystnym wnioskiem jest potencjał



Rysunek 4.3: Wpływ korekcji na wartości miar jakości asemblacji

tych algorytmów do poprawy wyników, gdy nie ma możliwości przeprowadzenia sekwencjonowania o dostatecznie wysokiej głębokości.



Rysunek 4.4: Wpływ korekcji na wartość ξ_{F1} detekcji wariantów genomu *H. sapiens*

Jako alternatywną możliwość w eksperymentach związanych z detekcją wariantów wybrano analizę opartą na odczytach genomu gatunku *Arabidopsis thaliana* (genom długości ok. 120 Mpz), motywując to dostępnością w ramach projektu 1001 Genomów [3] odpowiednich zestawów odczytów. Protokół eksperymen-

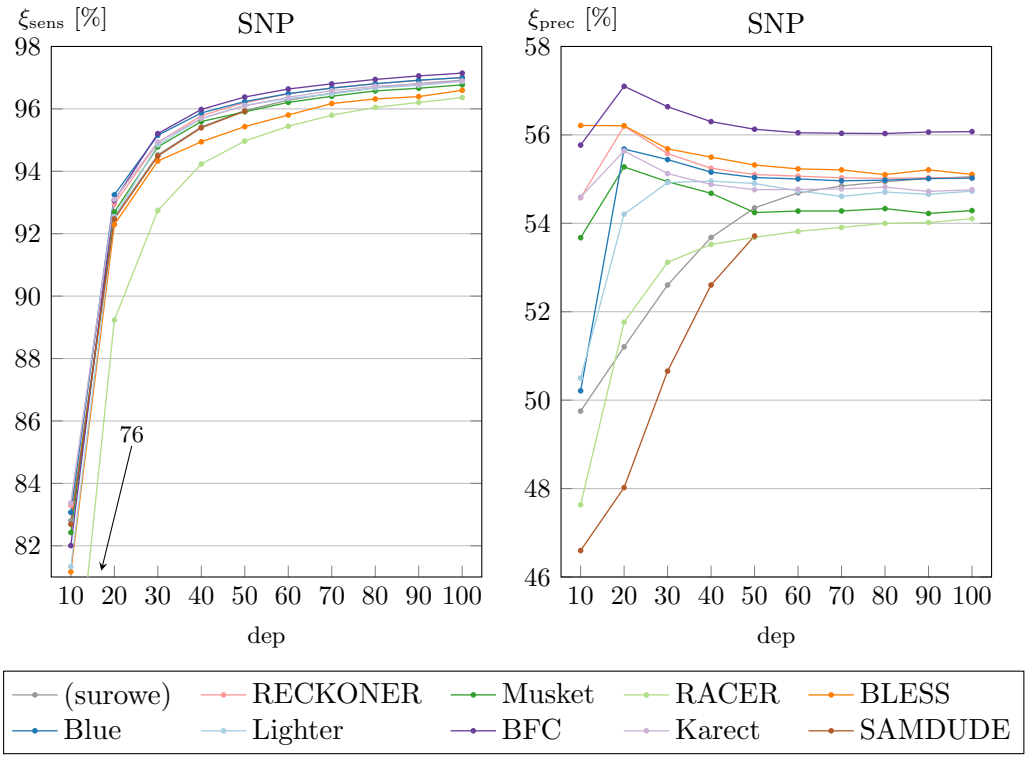
tów był podobny jak dla ludzkich wariantów, jednak ocena wyników przy pomocy narzędzia hap.py została przeprowadzona nie w oparciu o zestaw prawdy podstawowej, ale opracowany w ramach tego projektu zestaw wariantów. Opracowanie to polegało na wykonaniu przecięcia wyników dwóch różnych potoków detekcji wariantów oraz filtrację wyników. Tego rodzaju punkt odniesienia jest słabszej jakości niż zestaw prawdy podstawowej, jednak daje możliwość wykonania analizy w kontekście innych danych niż ludzkie warianty.

Na rys. 4.5 przedstawiono rozłącznie wartości czułości ξ_{sens} i precyzji ξ_{prec} detekcji wariantów typu SNP. Rozdzielenie tych dwóch miar w miejsce syntetycznej wartości F1 wynika z dużej różnicy uzyskanych wartości. Powolny spadek precyzji, który od pewnego momentu występuje wraz ze wzrostem głębokości sekwencjonowania, oraz ogólnie niskie wartości precyzji mogą być wytłumaczone m.in. zastosowaniem wspomnianej metody oceny — wzorcowe zestawy wariantów nie są przygotowane ze starannością analogiczną jak zestaw prawdy podstawowej ludzkiego genomu. W rezultacie prawdopodobnie są one pozbawione wielu wariantów, które zostały wykryte w wyniku przeprowadzonych eksperymentów, a następnie uznane za fałszywie pozytywne.

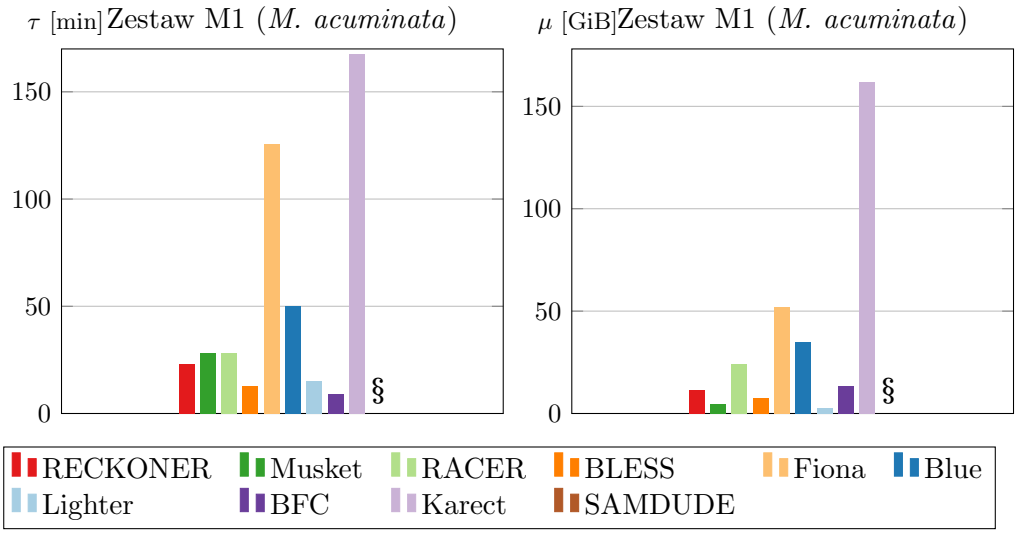
Pomimo tego uzyskane wyniki są źródłem pewnych informacji użytecznych w kontekście oceny korekcji odczytów. W większości przypadków korekcja skutkuje zwiększeniem liczby prawidłowych wariantów w stosunku do odczytów nieskorygowanych, co jest obserwowalne w postaci wzrostu czułości. Wśród wyników precyzji korzystnie wygląda wzrost w stosunku do surowych odczytów, obserwowany dla prawie algorytmów dla głębokości sekwencjonowania mniejszej od 60×, co świadczy o znacznym zmniejszeniu liczby fałszywie pozytywnych wariantów. Dodatkowo, podjęto liczne próby zastosowania innych prób porównania w oparciu o warianty odczytów *A. thaliana*, nie przynosząc jednak bardziej obrazowych zależności.

Na rys. 4.6 przedstawiono czas obliczeń τ oraz zapotrzebowanie na pamięć μ różnych algorytmów. W większości przypadków czas korekcji nie przekracza 0,5 godziny (obliczenia wykonano na komputerze wyposażonym w dwa 12-rdzeniowe procesory oraz 256 GiB RAM), choć w dwóch przypadkach przekracza 2 godziny (w przypadku algorytmu SAMDUDE, niewyposażonego w mechanizm przetwarzania równoległego, czas ten przypuszczalnie byłby jeszcze dłuższy). Różnice między zapotrzebowaniem na pamięć cechują się podobną skalą. W oparciu o tego rodzaju wyniki można wnioskować, że wykorzystanie algorytmów wiąże się z postawieniem bardzo odmiennych wymagań co do dostępnych zasobów, w praktyce dla odczytów genomów większych długości uniemożliwiających korekcję, nawet korzystając z rozbudowanego serwera obliczeniowego.

Niezależnej obserwacji poddano techniczne kwestie wykorzystania algorytmów, które nie mają bezpośredniego przełożenia na uzyskane rezultaty, ale mogą się wiązać z trudnościami z wykonaniem algorytmów oraz wykorzystaniem ich



Rysunek 4.5: Wpływ korekcji na wartości ξ_{sens} oraz ξ_{prec} detekcji wariantów genomu *A. thaliana*



Rysunek 4.6: Zapotrzebowanie na zasoby korekcji odczytów genomu *M. acuminata*

wyników. W szczególności zaobserwowano, że kompilacja implementacji niektórych algorytmów może nastręczyć problemów. Ponadto wybrane algorytmy nie są dostosowane do przetwarzania odczytów w formie skompresowanej lub tzw. *odczytów sparowanych*. Zaobserwowano przypadki zmian pewnych elementów struktury plików wyjściowych, a w jednym przypadku stwierdzono, że algorytm wymaga przekazania odczytów wejściowych zmapowanych uprzednio do genomu referencyjnego. Wiele spośród implementacji charakteryzuje się niską stabilnością, powodując w trakcie eksperymentów zgłoszenie błędów wykonania.

Mając na uwadze uzyskanie pełnego podsumowania, wykonano ścisły ranking algorytmów. W tym celu niezależnie dokonano oceny siedmiu grup eksperymentów, w ramach których wyznaczono osobne rankingi cząstkowe (mniejsza liczba punktów odpowiada lepszemu algorytmowi) dla następujących grup eksperymentów (pełne wyniki wszystkich grup zostały przedstawione wyłącznie w pracy): (*i*) odczyty symulowane metodą Quake, (*ii*) odczyty symulowane narzędziem ART, (*iii*) asemblacja, (*iv*) detekcja wariantów, (*v*) czas korekcji, (*vi*) zapotrzebowanie na pamięć korekcji, (*vii*) skalowalność. Następnie wyniki cząstkowe zostały poddane sumowaniu

Wyniki przedstawiono w pierwszej części tabeli 4.1. Pozwalają one na stworzenie syntetycznego podsumowania wszystkich algorytmów. Ich potwierdzeniem jest drugi zaproponowany ranking, w którym eksperymenty z wymienionych wyżej grup zostały ocenione niezależnie od siebie, a następnie poddane sumowaniu. W obu przypadkach wykazano, że najlepszym algorytmem jest RECKONER, nieznacznie wyprzedzając BFC.

5 Podsumowanie

Przedstawione wyżej cele pracy zostały osiągnięte w następujący sposób:

- opracowano nowy algorytm RECKONER, wyposażony w liczne autorskie rozwiązania, w dużej mierze zapewniając możliwość pracy przy umiarkowanym zapotrzebowaniu na zasoby i pozwalając na uzyskanie wyników dobrej jakości, co zostało wykazane eksperymentalnie,

- przeprowadzono znaczną grupę eksperymentów, wskazujących jakość poszczególnych algorytmów oraz poziom celowości poddawania odczytów korekcji; w eksperymentach położono nacisk na uwzględnienie różnorodności charakterystyki odczytów oraz podejść do metod oceny algorytmów,

- opracowano metodę oceny korekcji w oparciu o analizę wpływu korekcji na skuteczność zadania detekcji wariantów genomu,

- dokonano porównania wyników analizy algorytmów w korekcji przy zastosowaniu zarówno odczytów symulowanych, jak też rzeczywistych.

Tabela 4.1: Ranking algorytmów w oparciu o grupy eksperymentów (pierwsza część tabeli): (*i*) odczyty symulowane metodą Quake, (*ii*) odczyty symulowane narzędziem ART, (*iii*) asemblacja, (*iv*) detekcja wariantów, (*v*) czas korekcji, (*vi*) zapotrzebowanie na pamięć, (*vii*) skalowalność; ranking algorytmów w oparciu o niezależne eksperymenty (druga część tabeli)

Grupa	RECKONER	Musket	RACER	BLESS	Fiona	Blue	Lighter	BFC	Karect	SAMDUDE
i	1	9	4	6,5	6,5	3	5	8	2	10
ii	3	8	7	4	6	5	9	1	2	10
iii	1	6	8	9	7	2,5	4,5	2,5	4,5	10
iv	3	4,5	7	2	9,5	6	4,5	1	8	9,5
v	4	6	5	2	7	8	3	1	9	10
vi	5	2	6	3	8	7	1	4	10	9
vii	1	4	9,5	6	3	7	8	2	5	9,5
Suma	**18**	**39,5**	**46,5**	**32,5**	**47**	**38,5**	**35**	**19,5**	**40,5**	**68**

	RECKONER	Musket	RACER	BLESS	Fiona	Blue	Lighter	BFC	Karect	SAMDUDE
Suma	**116**	**200,5**	**213,5**	**178,5**	**241,5**	**197**	**163,5**	**123,5**	**210**	**336**

W kontekście uzyskanych wyników stwierdzono, że tezy pracy zostały udowodniona.

W oparciu o wyniki eksperymentów zaleca się przeprowadzanie korekcji odczytów Illumina przy pomocy jednego z algorytmów z grupy: RECKONER, Blue, Lighter, BFC albo Karect. Algorytmy te wprawdzie różnią się uzyskiwanymi rezultatami, a szczególnie szybkością działania i zapotrzebowaniem na pamięć, jednak w wyniku prac wykazano, że każdy z nich pozwala na uzyskanie dobrych wyników. Nie zaobserwowano też ryzyka znacznego uszkodzenia odczytów w rezultacie niepoprawnej korekcji.

Podsumowując zakres pracy można stwierdzić, że dalsze wysiłki powinny objąć poddanie analizie również inne opracowane do tej pory algorytmy korekcji. Zaproponowana metoda oceny korekcji w oparciu o detekcję wariantów powinna zostać rozszerzona, w szczególności o zestawy wariantów innych organizmów (przy założeniu dostępności zestawów prawdy podstawowej). W pracy wykorzystano odczyty rzeczywiste uzyskane z kilku modeli urządzeń marki Illumina. Na rynku dostępnych jest jednak więcej rodzajów urządzeń, w tym także innych producentów, których odczyty również warto wykorzystać w celu weryfikacji skuteczności korekcji. Istotnym zaobserwowanym brakiem w wiedzy jest niedostępność skutecznej strategii doboru długości oligomeru, zarówno algorytmu RECKONER, jak też innych algorytmów, nie tylko korekcji odczytów. Zadanie to jest skomplikowane, jednak szeroki zakres zastosowań ewentualnego rozwiązania powinien stanowić skuteczną motywację do podjęcia prac.

Bibliografia

[1] hap.py. https://github.com/Illumina/hap.py. [dostęp: 29.08.2020]. [cytowanie na str. 11]

[2] A. Allam, P. Kalnis oraz V. Solovyev. Karect: accurate correction of substitution, insertion and deletion errors for next-generation sequencing data. *Bioinformatics*, 31(21):3421–3428, 2015. [cytowanie na str. 3]

[3] C. Alonso-Blanco, J. Andrade, C. Becker, F. Bemm, J. Bergelson, K. M. Borgwardt, J. Cao, E. Chae, T. M. Dezwaan, W. Ding, et al. 1,135 genomes reveal the global pattern of polymorphism in Arabidopsis thaliana. *Cell*, 166(2):481–491, 2016. [cytowanie na str. 12]

[4] R. Chikhi and G. Rizk. Space-efficient and exact de Bruijn graph representation based on a Bloom filter. *Algorithms for Molecular Biology*, 8(1):1–9, 2013. [cytowanie na str. 10]

[5] M. Długosz. Genome variant calling in context of sequencing reads correction. In *Recent Advances in computational oncology and personalized medicine*, pages 89–98. Springer, 2021. doi: https://doi.org/10.34918/83567. [cytowanie na str. 5]

[6] M. Długosz and S. Deorowicz. Reckoner: read error corrector based on KMC. *Bioinformatics*, 33(7):1086–1089, 2017. [cytowanie na str. 5]

[7] M. Długosz and S. Deorowicz. Illumina reads correction—evaluation and improvements. 2023. doi: https://doi.org/10.21203/rs.3.rs-2715541/v1. [cytowanie na str. 5]

[8] M. Długosz, S. Deorowicz oraz M. Kokot. Improvements in DNA Reads Correction. In *Man-Machine Interactions 5: 5th International Conference on Man-Machine Interactions, ICMMI 2017 Held at Kraków, Poland, October 3-6, 2017*, pages 115–124. Springer, 2018. [cytowanie na str. 5]

[9] I. Fischer-Hwang, I. Ochoa, T. Weissman oraz M. Hernaez. Denoising of Aligned Genomic Data. *Scientific reports*, 9(1):1–11, 2019. [cytowanie na str. 3]

[10] P. Greenfield, K. Duesing, A. Papanicolaou oraz D. C. Bauer. Blue: correcting sequencing errors using consensus and context. *Bioinformatics*, 30(19):2723–2732, 2014. [cytowanie na str. 3]

[11] A. Gurevich, V. Saveliev, N. Vyahhi oraz G. Tesler. QUAST: quality assessment tool for genome assemblies. *Bioinformatics*, 29(8):1072–1075, 2013. [cytowanie na str. 11]

[12] Y. Heo, X.-L. Wu, D. Chen, J. Ma oraz W.-M. Hwu. BLESS: bloom filter-based error correction solution for high-throughput sequencing reads. *Bioinformatics*, 30 (10):1354–1362, 2014. [cytowanie na str. 3]

[13] Y. Heo, A. Ramachandran, W.-M. Hwu, J. Ma oraz D. Chen. BLESS 2: accurate, memory-efficient and fast error correction method. *Bioinformatics*, 32(15):2369–2371, 2016. [cytowanie na str. 3]

[14] W. Huang, L. Li, J. R. Myers oraz G. T. Marth. ART: a next-generation sequencing read simulator. *Bioinformatics*, 28(4):593–594, 2012. [cytowanie na str. 9]

[15] L. Ilie and M. Molnar. RACER: Rapid and accurate correction of errors in reads. *Bioinformatics*, 29(19):2490–2493, 2013. [cytowanie na str. 3]

[16] S. Kim, K. Scheffler, A. L. Halpern, M. A. Bekritsky, E. Noh, M. Källberg, X. Chen, Y. Kim, D. Beyter, P. Krusche, et al. Strelka2: fast and accurate calling of germline and somatic variants. *Nature methods*, 15(8):591–594, 2018. [cytowanie na str. 11]

[17] M. Kokot, M. Długosz oraz S. Deorowicz. KMC 3: counting and manipulating k-mer statistics. *Bioinformatics*, 33(17):2759–2761, 2017. [cytowanie na str. 6]

[18] H. Li. BFC: correcting Illumina sequencing errors. *Bioinformatics*, 31(17):2885–2887, 2015. [cytowanie na str. 3]

[19] Y. Liu, J. Schröder oraz B. Schmidt. Musket: a multistage k-mer spectrum-based error corrector for Illumina sequence data. *Bioinformatics*, 29(3):308–315, 2012. [cytowanie na str. 3]

[20] M. Molnar and L. Ilie. Correcting Illumina data. *Briefings in bioinformatics*, 16(4): 588–599, 2014. [cytowanie na str. 3]

[21] M. H. Schulz, D. Weese, M. Holtgrewe, V. Dimitrova, S. Niu, K. Reinert oraz H. Richard. Fiona: a parallel and automatic strategy for read error correction. *Bioinformatics*, 30(17):i356–i363, 2014. [cytowanie na str. 3]

[22] L. Song, L. Florea oraz B. Langmead. Lighter: fast and memory-efficient sequencing error correction without counting. *Genome biology*, 15(11):509, 2014. [cytowanie na str. 3]

[23] X. Yang, S. P. Chockalingam oraz S. Aluru. A survey of error-correction methods for next-generation sequencing. *Briefings in bioinformatics*, 14(1):56–66, 2012. [cytowanie na str. 3]

Wykaz dorobku autora

Artykuły w czasopismach

1. Maciej Długosz, Sebastian Deorowicz. RECKONER: read error corrector based on KMC. *Bioinformatics*, 33(7):1086–1089, 2017. doi: https://doi.org/10.1093/bioinformatics/btw746.

2. Marek Kokot, Maciej Długosz, Sebastian Deorowicz. KMC 3: counting and manipulating k-mer statistics. *Bioinformatics*, 33(17):2759–2761, 2017. doi: https://10.1093/bioinformatics/btx304.

3. Sebastian Deorowicz, Adam Gudyś, Maciej Długosz, Marek Kokot, Agnieszka Danek. Kmer-db: instant evolutionary distance estimation. *Bioinformatics*, 35(1):133–136, 2019. doi: https://doi.org/10.1093/bioinformatics/bty610.

4. Maciej Długosz, Sebastian Deorowicz. Illumina reads correction — evaluation and improvements. Scientific Reports (w recenzji). doi: https://doi.org/10.21203/rs.3.rs-2715541/v1.

Publikacje konferencyjne

1. Maciej Długosz, Sebastian Deorowicz. RECKONER — a new tool for DNA read error correction. *VII Konwersatorium Chemii Medycznej & VIII Sympozjum PTBI, 17–19 września 2015, Lublin* (plakat), 2015.

2. Marek Kokot, Maciej Długosz, Sebastian Deorowicz. Operations on k-mers' sets. *VII Konwersatorium Chemii Medycznej & VIII Sympozjum PTBI, 17–19 września 2015, Lublin* (plakat), 2015.

3. Maciej Długosz, Sebastian Deorowicz. A new algorithm for Illumina sequencing reads correction. *XXth Gliwice Scientific Meetings, Gliwice, November 18-19, 2016* (plakat), 2016.

4. Maciej Długosz, Sebastian Deorowicz, Marek Kokot. Improvements in DNA Reads Correction. *Man-Machine Interactions 5: 5th International Conference on Man-Machine Interactions, ICMMI 2017 Held at Kraków, Poland, October 3-6, 2017*, 115–124. Springer, 2018. doi: https://doi.org/10.1007/978-3-319-67792-7_12.

5. Marek Kokot, Sebastian Deorowicz, Maciej Długosz. Even faster sorting of (not only) integers. *Man-Machine Interactions 5: 5th International Conference on Man-Machine Interactions, ICMMI 2017 Held at Kraków, Poland, October 3-6, 2017*, 481–491. Springer, 2018. doi: https://doi.org/10.1007/978-3-319-67792-7_47.

6. Sebastian Deorowicz, Adam Gudyś, Maciej Długosz, Marek Kokot, Agnieszka Danek. Kmer-db: instant evolutionary distance estimation. *ISMB/ECCB 2019, Basel, Switzerland July 21–July 25* (plakat), 2019.

7. Maciej Długosz. Genome variant calling in context of sequencing reads correction. *Recent Advances in computational oncology and personalized medicine*, 89–98. Springer, 2021. doi: https://doi.org/10.34918/83567.