Projekti za 100 bodova na predmetu Bioinformatika 1, 2023./2024.

- broj članova tima: 2
- implementacija: C/C++
- opis algoritma, implementacije i testiranje
- dozvoljeno je korištenje pomoćnih knjižnica u zadatcima gdje je tako navedeno, a za ostale situacije možete se dogovoriti s nastavnikom koji je zadao temu
- za svaki dan zakašnjenja umanjuje se konačan broj bodova za 3 boda

Bodovanje zadataka (1) – (3)

	Broj bodova
Program - testiranje • ako program ne radi ispravno na testnim podatcima umanjuje se konačan broj bodova za 10 bodova • prepravke napraviti u roku 2 dana Performanse programa (vrijeme izvođenja i utrošak memorije) • ako se program uspoređuje s objavljenim rješenjem, odstupanje implementacije treba biti do najviše 100% vremena izvođenja i utroška memorije u odnosu na referentni rezultat (npr. ako referentni program koristi 1 GB memorije za neki skup podataka, onda Vaša implementacija treba koristiti najviše 2 GB memorije) o oduzima se 10 bodova, ako je odstupanje do 200% o oduzima se 15 bodova, ako je odstupanje veće od 200%	65
Testiranje na umjetno generiranim podatcima 10^3-10^7 znakova • svi rezultati moraju biti u dokumentaciji – prikazani u tablici i/ili grafu	10
Testiranje na stvarnim podatcima (*Escherichia coli* ili po dogovoru ovisno o zadatku) • svi rezultati moraju biti u dokumentaciji – prikazani u tablici i/ili grafu	10
Dokumentacija • opis algoritma i vizualizacija na jednostavnom primjeru (4 boda) • obvezno navesti popis literature i navesti izvore unutar teksta (3 boda) • za svaki algoritam napraviti analizu točnosti, vremena izvođenja i utroška memorije za različite testne slučaje (3 boda)	10
Prezentacija • oduzimaju se bodovi, ako je prezentacija dulja od predviđenoga vremena (1 bod za svaku minutu prekoračenja)	5

(1) The Logarithmic Dynamic Cuckoo Filter (Zhang et al. 2021) (MDL)

- Zhang et al. The Logarithmic Dynamic Cuckoo Filter
  doi: 10.1109/ICDE51399.2021.00087
- Chen et al. 2017. The dynamic cuckoo filter; https://ieeexplore.ieee.org/abstract/document/8117563
- Fan et al. 2013. Cuckoo Filter: Better Than Bloom;
  https://www.cs.cmu.edu/~binfan/papers/login_cuckoofilter.pdf
- Fan et al. 2014. Cuckoo Filter: Practically Better Than Bloom;
  http://www.cs.cmu.edu/%7Ebinfan/papers/conext14_cuckoofilter.pdf
- tražiti slučajne podnizove (k-mere uz različite k, npr. k = 10, 20, 50, 100, 200) u *E. coli* genomu
- napraviti vlastiti LDCF te usporediti s originalnom implementacijom

(2) HRCM algorithm (Yao et al. 2019) (MDL)

- Yao et al. 201.9 HRCM: An Efficient Hybrid Referential Compression Method for Genomic Big Data
  doi: 10.1155/2019/3108950
- napraviti vlastitu implementaciju algoritma za sažimanje i dekompresiju
- usporediti s originalnom implementacijom
- testirati na skupovima podataka koji su priloženi uz originalnu implementaciju

(3) Bamboo Filter (Wang et al. 2022) (MDL)

- Wang et al. Bamboo Filters: Make Resizing Smooth
  doi: 10.1109/ICDE53745.2022.00078
- Fan et al. 2013. Cuckoo Filter: Better Than Bloom;
  https://www.cs.cmu.edu/~binfan/papers/login_cuckoofilter.pdf
- Fan et al. 2014. Cuckoo Filter: Practically Better Than Bloom;
  http://www.cs.cmu.edu/%7Ebinfan/papers/conext14_cuckoofilter.pdf
- tražiti slučajne podnizove (k-mere uz različite k, npr. k = 10, 20, 50, 100, 200) u *E. coli* genomu
- napraviti vlastiti BF te usporediti s originalnom implementacijom

(4) Pronalaženje varijanti gena iz podataka dobivenih sekvenciranjem (kresimir.krizanovic@fer.hr)

Cilj: Sekvenciran je uzorak koji sadrži nekoliko varijanti istog gena. Potrebno je primijeniti tehnike grupiranja (engl. *clustering*) na očitanja da bi se otkrile sve varijante danog gena koje su prisutne u uzorku. Očitanja je potrebno grupirati na temelju međusobne udaljenosti. Za računanje centroida pojedine grupe (engl. *cluster*) dopušteno je koristiti postojeću biblioteku SPOA (https://github.com/rvaser/spoa)

Ulazni podaci:

- Skup očitanja

Izlazni podaci:

- Skup otkrivenih varijanti gena u FASTA formatu
- Popis očitanja koja pripadaju kojoj varijanti/grupi/clusteru

Skupovi očitanja bit će pripremljeni kao ulazni podaci, kao i nekoliko uzoraka sa poznatim varijantama.

Za preuzimanje testnih podataka te za detaljnije upute o projektu potrebno se javiti na kresimir.krizanovic@fer.hr.

Evaluacija:

- Testiranje na osnovnim podacima za koje su rezultati poznati.
- Testiranje na podacima za koje stvarni podaci nisu poznati te usporedba s drugim rješenjima.

Bodovanje:

	Broj bodova
Program • ako program ne radi ispravno na osnovnim podacima prilikom demonstracije umanjuje se konačan broj bodova za 20 bodova (prepravke napraviti u roku od 2 dana) • program mora ispravno raditi na dva najveća clustera na skupovima podataka s poznatim rješenjem	80
Dokumentacija • opis algoritma i vizualizacija na jednostavnom primjeru • obavezno navesti popis literature te navesti izvore unutar teksta • napraviti ocjenu točnosti, vremena izvođenja i utroška memorije	15
Prezentacija • oduzimaju se bodovi, ako je prezentacija dulja od predviđenoga vremena	5

Preporučena literatura:

1. Skripta iz bioinformatike
2. Biblioteka SPOA (https://github.com/rvaser/spoa)
3. Završni rad Sanje Kosier (mailom nakon prvih konzultacija)

(5) Navarrov algoritam za približno uspoređivanje teksta (kresimir.krizanovic@fer.hr)

Zadatak: Implementirati Navarrov algoritam opisan u radu (Improved approximate pattern matching on hypertext) https://www.sciencedirect.com/science/article/pii/S0304397599003333.

Evaluacija:

Usporediti s bit parallel sequence-to-graph alignment algoritmom (opisanom u radu https://academic.oup.com/bioinformatics/article/35/19/3599/5372677 . Algoritam usporediti na 3 vrste graf topologija koje su opisane u poglavlju 6.2 Graph topology experiment. Nije potrebno testirati na topologijama koje imaju cikluse. Skripte za generiranje testnih podataka dostupne su na https://github.com/maickrau/GraphAligner/tree/PaperExperiments/WabiExperimentSnake.

Bodovanje:

	Broj bodova
Program • ako program ne radi ispravno na linearnoj topologiji prilikom demonstracije umanjuje se konačan broj bodova za 20 bodova (prepravke napraviti u roku od 2 dana)	80
Dokumentacija • opis algoritma i vizualizacija na jednostavnom primjeru • obavezno navesti popis literature te navesti izvore unutar teksta • napraviti usporedbu točnosti, vremena izvođenja i utroška memorije vaše implementacije i izvorne	15
Prezentacija • oduzimaju se bodovi, ako je prezentacija dulja od predviđenoga vremena	5

Preporučena literatura:

1. Rad Improved approximate pattern matching on hypertext (https://www.sciencedirect.com/science/article/pii/S0304397599003333)
2. Rad Bit-parallel sequence-to-graph alignment (https://academic.oup.com/bioinformatics/article/35/19/3599/5372677)

(6) Pronalazak mutacija pomoću treće generacije sekvenciranja (kresimir.krizanovic@fer.hr)

Ulaz: referentni genom i skup očitanja dobiven sekvenciranjem mutiranog genoma. Obje datoteke su u FASTA formatu.

Cilj: Za dani ulaz, pronaći razlike između referentnog genoma i sekvenciranog mutiranog genoma. Mutacije uključuju jednostruke substitucije, umetanja i brisanja. Očitanja je potrebno mapirati na danu referencu pomoću alata minimap2 (https://github.com/lh3/minimap2), poravnati ih te iz gomile poravnanja razlučiti mutacije.

Izlaz: Lista mutacija u odnosu na referencu (gdje je prvi nukleotid na poziciji 0), u CSV formatu kao što je prikazano u tablici ispod.

Mutacija	**Linija u CSV datoteci**		
Substitucija	X	Pozicija u referenci na kojoj se dogodila substitucija	Zamjenska nukleotidna baza
Umetanje	I	Pozicija u referenci prije koje se dogodilo umetanje	Umetnuta nukleotidna baza
Brisanje	D	Pozicija u referenci na kojoj se dogodilo brisanje	-

Evaluacija: usporediti rezultate s referentnom implementacijom pomoću alata FreeBayes (https://github.com/freebayes/freebayes). Alat FreeBayes vraća varijante u VCF formatu.

Bodovanje:

	Broj bodova
Program • ako program ne radi ispravno na testnim podatcima prilikom demonstracije umanjuje se konačan broj bodova za 20 bodova (prepravke napraviti u roku od 2 dana)	80
Dokumentacija • opis algoritma i vizualizacija na jednostavnom primjeru • obavezno navesti popis literature te navesti izvore unutar teksta • napraviti usporedbu točnosti, vremena izvođenja i utroška memorije vaše implementacije i izvorne	15
Prezentacija • oduzimaju se bodovi, ako je prezentacija dulja od predviđenoga vremena	5

Preporučena literatura:

1. Skripta iz bioinformatike

(7) Određivanje sastava metagenomskog uzorka (kresimir.krizanovic@fer.hr)

Ulaz: skup referentnih genoma i skup očitanja dobiven sekvenciranjem uzorka koji sadrži više organizama. Obje datoteke su u FASTA formatu.

Cilj: Za dani ulaz, za svako očitanje odrediti kojem od referentnih genoma pripada (kojem je najsličnije). Za određivanje sličnosti između očitanja i genoma koristiti distribuciju kmer-a.

Izlaz: Za svaki od genoma izlaz treba sadržavati koliko mu očitanja pripada.

Evaluacija: Očitanja mapirati na skup genoma koristeći alat minimap2 (https://github.com/lh3/minimap2). Na temelju kvalitete mapiranja svako očitanje pridjeliti jednom od genoma. Usporediti rezultate dobivene pomoću distribucije kmer-a sa rezultatima dobivenim mapiranjem.

Bodovanje:

	Broj bodova
Program • ako program ne radi ispravno na testnim podatcima prilikom demonstracije umanjuje se konačan broj bodova za 20 bodova (prepravke napraviti u roku od 2 dana)	80
Dokumentacija • opis algoritma i vizualizacija na jednostavnom primjeru • obavezno navesti popis literature te navesti izvore unutar teksta	15
Prezentacija • oduzimaju se bodovi, ako je prezentacija dulja od predviđenoga vremena	5

Preporučena literatura:

2. Skripta iz bioinformatike