Vyhľadávanie DNA sekvencií v reťazcovom grafe

Angelika Fedáková

Problém

Máme DNA vzorku D a prístroj, ktorý z daného reťazca vytvorí čítania, ktoré sú oproti celému reťazcu krátke a pochádzajú z náhodných miest v reťazci. Na vstupe dostaneme graf G týchto čítaní, ktorý nám podľa prekryvov jednotlivých čítaní hovorí, ktoré časti na seba pravdepodobne nadväzovali a tie dá do jednotlivých vrcholov. Následne pospája hranami tie vrcholy, ktoré by mohli ísť za sebou, ale nie je to isté, z dôvodu napr. opakovania častí v pôvodnej DNA, teda z jedného vrcholu môže ísť hrana do viacerých vrcholov. Ďalej dostaneme nejaký druhý DNA reťazec X, a našou úlohou je zistiť, či v našom grafe pre daný reťazec existuje cesta, alebo ak neexistuje, nájsť ideálne cesty pre nejaké časti reťazca alebo nájsť cesty, ktoré sa na reťazec alebo jeho časti čo najviac podobajú (napr. na základe editačnej vzdialenosti).

Zimný semester

Spracovanie dát

Máme hlavný súbor v psl formáte, v ktorom máme zarovnania jednotlivých reťazcov oproti X. Sú to informácie o podobnosti, mená vrcholov, chybovosť daných vrcholov a ďalšie. Z toho si vyberieme nám potrebné info a vytvoríme si zoznam zarovnaní, ktoré za sebou nasledujú, pričom sú zoradené podľa začiatočnej pozície zarovnania v reťazci X, medzi ktorými budeme hľadať cesty. Tie neskôr budeme aj filtrovať. K týmto zarovnaniam si zároveň vypíšeme aj štatistiky, teda napríklad, o koľko sa reťazce daných zarovnaní prekrývajú, medzi ktorými je medzera alebo ich chybovosť. Ďalej máme súbor LastGraph, v ktorom máme zoznam vrcholov, teda obsahov ich reťazcov a

nakoniec zoznam hrán - dvojice susediacich vrcholov, pričom každý vrchol má označenie + alebo -, ktoré znamenajú začiatok alebo koniec reťazca, ktorý sa spája.

Filtrácia

Prvým krokom pri hľadaní ciest v grafe bolo vybrať, ktoré zarovnania sú pre nás relevantné. Zvažovali sme viacero metód filtrácie. Hlavným kritériom bola chybovosť zarovnania. Spomedzi absolútneho a relatívneho pomeru chybovosti sme nakoniec zvážili za vhodnejšie odstraňovať zarovnania, ktoré mali pomer chýb v reťazci menší ako konštanta, v našom konkrétnom testovacom prípade genómu Escherichia coli, sme zvolili 10%. Taktiež sme zanedbávali zarovnania, ktoré boli príliš krátke. V letnom semestri chceme ďalej pracovať na zlepšení filtrácie, na pridaní viacerých kritérií pre filtráciu dát.

Nový graf

Ďalším krokom bolo vytvoriť nový graf Gv tak, aby sme v ňom dokázali hľadať cesty, keďže hrany máme zatiaľ iba ako susedné dvojice vrcholov. Po zvážení sme modifikovali graf tak, že sme ku každému vrcholu priradili dva nové vrcholy, reprezentujúce kladný a záporný koniec vrcholu. Ak sme potom našli v súbore hranu medzi dvoma vrcholmi, napr. kladným vrcholom A a záporným vrcholom B, rovnako sme vytvorili symetrickú hranu, teda u nás medzi záporným vrcholom A a kladným vrcholom B, kvôli vlastnostiam spájania DNA reťazcov.

Hľadanie cesty

V novovytvorenom grafe Gv sme už mohli hľadať cesty, pomocou algoritmu BFS. Výsledkom zimnej časti ročníkového projektu, bolo nájdenie ciest medzi jednotlivými zarovnaniami, alebo vypísať, že medzi danými zarovnaniami cesta neexistuje.

Letný semester

Reprezentácia

Ako prvé sme zmenili reprezentáciu grafu. Miesto toho, aby sme vytvorili pre každý vrchol dva nové konce, vytvorili sme pôvodnú verziu vrcholu a následne prevrátenú verziu vrcholu, ktorá mala opačné znamienko(pri vrchole 4 sme vytvorili opačný vrchol -4 a naopak). Vytvorili sme taktiež reprezentáciu grafu pomocou mapy, kde sme k číslu vrcholu priradili vrchol ako inštanciu štruktúry. Nazvime tento graf Gz. Následne sme mohli jednotlivé funkcie a procedúry zjednodušiť.

Najkratšia cesta

Hrany v Gz sme si ohodnotili. Váha hrany sa rovnala dĺžke vrcholu, do ktorého hrana vchádzala. Následne sme pri hľadaní ciest implementovali Dijkstrov algoritmus, ktorý sme mierne upravili, keďže v našom grafe Gz hrany z vrcholu do samého seba nie sú nutne nulové, resp. prítomné.

Výsledná cesta

Keď sme už vedeli medzi vrcholmi v Gz hľadať najkratšie cesty, cieľom bolo nájsť celkovú najlepšiu cestu. Ešte pred začiatkom hľadania sme každému vrcholu(zarovnaniu) v Gz priradili hodnotu podľa toho, koľko mal zhôd, nezhôd a podobne. Následne sme nehľadali cesty len medzi priamo susediacimi zarovnaniami (susedia v zozname usporiadania podľa začiatku výskytu v sekvencii), ale pre každé zarovnanie sme našli cesty do všetkých zarovnaní viac napravo, ktoré sa prekrývali o maximálne 50 a zároveň ich začiatok bol od konca nášho pôvodného zarovnania vzdialený maximálne o 1000. Z tohto sme si vytvorili nový acyklický graf s ohodnotenými hranami a vrcholmi, pričom hodnoty mohli byť aj záporné. Všetky hrany išli zľava doprava, t.j. od vrcholov s menším číslom k tým s väčším. Dynamickým programovaním, kde sme si pre každý vrchol pamätali najlepšiu cestu, ktorá v ňom končí, sme vyrátali najlepšiu cestu pre každý vrchol. Mohlo ísť buď o cestu pozostávajúcu len z aktuálneho vrcholu v alebo o cestu, ktorá ide do niektorého predchodcu w vrcholu v a potom použije hranu w->v. Jej cena bude najlepšia cena uložená pre w plus cena hrany w->v. Keď sme to spravili pre všetky vrcholy, našli sme plne maximálnu cenu v celom grafe.

Výsledok

Celý proces sme opakovali (nie nutne) pre viacero vstupných sekvencií, ktoré boli uložené v jednom psl súbore (rozlíšili sme ich podľa mena zarovnania qName). Vo výslednom súbore pre každú sekvenciu vypisujeme nájdené najlepšie cesty pre každú vstupnú sekvenciu na štyroch riadkoch v nasledovnom formáte:

- prvý riadok obsahuje ">" a meno sekvencie (qName)

- druhý riadok obsahuje zoznam node-ov na ceste, t.j. postupnosť čísel so znamienkom oddelené medzerami

- tretí riadok obsahuje odkiaľ pokiaľ je nájdená cesta v našej sekvencii a celkovú dĺžku sekvencie z psl súboru(qSize)

- štvrtý riadok obsahuje celkovú cenu cesty .