Bioinformatika – porovnávání sekvencí DNA

Anna Smejkalová[1], Martin Volek[2], Michal Chobola[3],
Gymnazium Friedricha Schiller[1], Gymnázium Plasy[2], Střední průmyslová škola Karviná[3]
anicka.sm@gmail.com, volek000@seznam.cz, majkee@chobola.net

Abstrakt:
V našem projektu jsme se věnovali porovnávání různých sekvencí DNA za pomoci dynamického programování s implementací v jazyku Python. Cílem naší práce bylo porovnat dvě reálné sekvence DNA.

1. Úvod

DNA (deoxyribonukleová kyselina) se nachází v jádře buněk a je tvořena čtyřmi základními bázemi, ty se nazývají Adenin(A), Guanin(G), Cytosin(C) a Thymin(T). Pořadí, v jakém se tyto báze vyskytují, určuje aminokyseliny, které bude organismus vytvářet. Aminokyseliny tvoří základ proteinů.

Poprvé byl základ DNA popsán v roce 1869 švýcarským lékařem Friedrichem Miescherem. Funkce DNA byla objevena, ale až v roce 1943 Oswaldem Averym, Colinem MacLeodem a Maclynem McCartym, kteří za pomoci transformací pneumokoků zjístili, že DNA je genetickým materiálem buněk.

Nejvýznamější objev nastal v roce 1953, kdy vědci James D.Watson a Francis Crick ukázali, že DNA má tvar dvoušroubovice. Za tento významný objev získali Nobelovu cenu.

Adenin a Thynin vždy tvoří dvojici, stejně tak jako Cytosin a Guanin. Díky tomu můžeme zapisovat pouze jednu linii dané DNA a tím utvoříme textový soubor. Ten se pak sekvenčně srovnává pomocí dynamického programování.

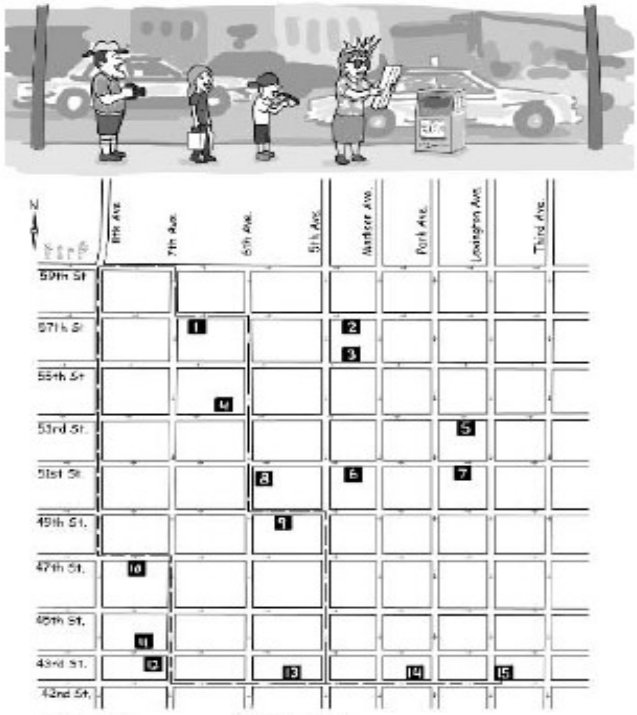
V současné době je ještě mnoho neprozkoumaných úseků DNA. Touto problematikou se zabývá právě bioinformatika. Pomocí dynamického programování porovnáváme nám neznámé sekvence DNA s již známými sekvencemi DNA nižších živočichů.

2. Srovnávání DNA pomocí dynamického programování

Dynamické programování je optimalizační metoda, která nejčastěji hledá nejkratší nebo nejdelší cestu v grafu. Uplatnění tohoto způsobu denně používáme na internetových mapách, které za nás najdou právě nejkratší nebo nejrychlejší cestu.

Stěžejní myšlenkou je rozdělení složité úlohy na úlohy menší, které jsme schopni vyřešit. Pomocí jejich řešení pak konstruujeme i řešení původního problému.

Pro lepší pochopení zde uvedeme příklad úlohy **Turista na Mannhattanu**

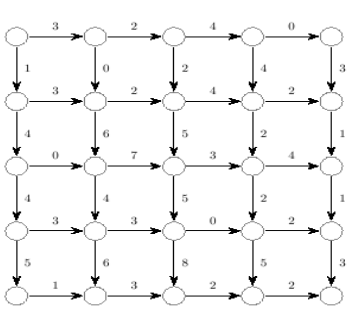


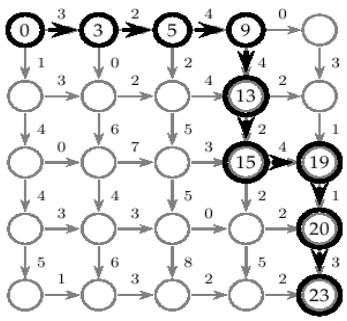
Turista se nachází na rohu 59té ulice a 8mé Avenue(tzn. V levém horním rohu) a chce se dostat k Chrysler budově na 42 ulici a Lexington Avenue(tzn. V pravém dolním rohu) při své cestě může chodit pouze doprava a dolů a nesmí se vracet. Vaším úkolem je naplánovat turistovi takovou trasu, aby se dostal přesně tam kam chce a přitom navštívil co nejvíce památek. Ty jsou na mapě vyobrazeny pomocí černých čtverečků s čísly.

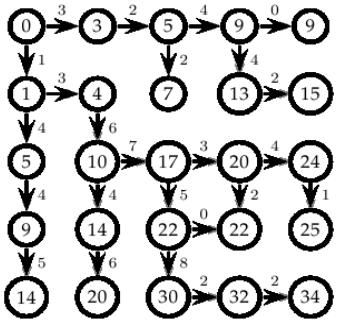
Jak vyřešit tento úkol pomocí dynamického programování

Pomocí dynamického programování jsme se rozhodli nastavit tenhle postup:

- Z realného obrázku jsme přešli ke grafu, kde hrany reprezentují ulice a vrcholy zastupují křižovatky. Každé hraně přiřadíme hodnotu podle toho, jak je daná ulice pro turistu zajímavá.
- Hodnoty ve vrcholech jsou součty hodnot hran podél nejzajímavější cesty do daného vrcholu.
- Nejprve vyplníme vrcholy podél levého a horního okraje. Následně vyplňujeme vždy takové vrcholy jejichž levý a horní soused má již vyplněnou hodnotu. K hodnotě horního souseda přičteme hodnotu hrany shora, k hodnotě levého souseda přičteme hodnotu hrany zleva a zapíšeme maximum z těchto hodnot.







- Tímto způsobem zaplníme celou tabulku. Ale nikdy nesmíme zapomenout směr odkud jsme přišli. Zpětnou vazbou pak zjistíme nejlepší variantu pro turistu.

Porovnávání sekvencí DNA

Předpokládejme, že máme dvě sekvence DNA označené **v** a **w**. Editační vzdálenost udává počet operací, které je potřeba provést k přepsání sekvence **v** na sekvenci **w**. Mezi povolené operace patří – vymazání znaku, vložení znaku a přepsání znaku.

Na této sekvenci se pokusíme spočítat editační vzdálenost pomocí vymazání a vložení znaků. Na daném úseku se tyto operace provedli čtyřikrát tudíž editační vzdálenost je rovna čtyřem. Pomlčka v proměnné **v** označuje že se daný prvek z proměnné **w** vloží na místo pomlčky. V opačném případě, kdy je pomlčka v proměnné **w** se prvek odstraní.

3. Realné porovnávání DNA

Na internetové stránce ENTREZ (www.ncbi.nlm.nih.gov/entrez) jsme našli 2 realná sekvence DNA. První sekvence patří bakterii E.coli a ta druhá mouše octomilce. Které jsme porovnali za pomoci dynamického programování s implementací v jazyku Python.

```
S1 (E.coli):                    AGATTTCGACGCCACCGACC
S2 (octomilka):                 CATGCTAAGCGAGCGCTCTA
          srovnání: σ = 0 a μ = 1
Výsledek:
S1:                       -A-G---A-TTTCGA-CGC-C-A-CCGACC
S2:                       CATGCTAAG---CGAGCGCTCTA-------
```

Na obrázku jsou zvýrayněny delší shody CGA a CGC, které algortimus nalezl.

4. Shrnutí

Pochopili jsme principy dynamického programování. Porovnali jsme dvě realné sekvence DNA za pomoci implementace v jazyku Python. A našli jsme v nich shodné úseky. Dozvěděli jsme se o obtížnosti rozluštění DNA kódu, které ani poté nemusí být správné.

Poděkování

Děkujeme našim supervizorům Ing. Tomáši Oberhubrovi, Ph.D. a Ing.Radkovi Mácovi za vysvětlení a uvedení do problému bioinformatiky. A dále také Fakultě jaderné a fyzikálně inženýrské a Českému vysokému učení technickému v Praze za celý projekt Týden vědy.

Reference:

[1] C. JONES,NEIL – A.PEVZNER, PAVEL.: *AN INTRODUCTION TO BIOINFORMATICS ANGORITHMS* A Bradford Book. 2004.

[2] KINSER, JASON.: *Python for Bioinformatics* Jones and Bartlett Publishers. 2009.

[3] M. LESK, ARTHUR.: *Introduction to Bioinformatics* OXFORD. 2008.

[4] MICHEL CLAVERIE, JEAN, PHD – NOTREDAME, CEDRIC, PHD.: *Bioinformatics for Dummies* Wiley Publishing, Inc. 2007.