

UNIVERZITA KOMENSKÉHO, BRATISLAVA
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

ZAROVNÁVANIE SEKVENCIÍ S POUŽITÍM METÓD
KLASIFIKÁCIE

Diplomová práca

2014

Bc. Michal Hozza

UNIVERZITA KOMENSKÉHO, BRATISLAVA
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

ZAROVNÁVANIE SEKVENCIÍ S POUŽITÍM METÓD KLASIFIKÁCIE

Diplomová práca

Študijný program: Informatika
Študijný odbor: 2508 Informatika
Školiace pracovisko: Katedra Informatiky
Školiteľ: Mgr. Tomáš Vinař, PhD.

Bratislava, 2014

Bc. Michal Hozza

Podakovanie...

Bc. Michal Hozza

Abstrakt

Slovenský abstrakt

Kľúčové slová: ...

Abstract

English abstract

Key words: ...

Obsah

Úvod	1
1 Zarovnávanie sekvencií	2
1.1 Podobnosť sekvencií a sekvenčná homológia	2
1.2 Skórovacie schémy	2
1.3 Párové zarovnávanie	3
1.3.1 Typy zarovnaní	3
1.3.2 Algoritmy na hľadanie zarovnaní	3
1.3.3 Algoritmus pre globálne zarovnanie: Needleman-Wunch	4
1.3.4 Algoritmus pre lokálne zarovnanie: Smith-Waterman	5
2 Prehľad problematiky	7
3 Doplnkové informácie k sekvenciám	8
4 Random forests	9
5 Návrh riešenia	10
6 Implementácia	11
6.1 Simulátor	11
7 Zhodnotenie úspešnosti	12
Záver	12
Literatúra	14

Zoznam obrázkov

1.1	Globálne zarovnanie	4
1.2	Lokálne zarovnanie	5

Zoznam tabuliek

Úvod

Najnovšie technológie sekvenovania DNA produkujú stále väčšie množstvo DNA sekvencií rôznych organizmov. Spolu s tým stúpa aj potreba rozumieť týmto dátam. Dôležitým krokom k ich porozumeniu je *zarovnávanie sekvencií*. Pomocou zarovnávanie sekvencií dokážeme zistiť ich štruktúru a následne funkciu jednotlivých častí.

Existujú rôzne algoritmy na zarovnávanie sekvencií. Väčšina z nich je založená na nejakom pravdepodobnostnom modeli, pričom sa snažia nájsť zarovnanie s čo najväčšou pravdepodobnosťou. Či už ide o algoritmy založené na dynamickom programovaní, ktoré s určitosťou nájdu najpravdepodobnejšie zarovnanie, ale za cenu kvadratického času v závislosti od dĺžok sekvencií, alebo heuristické algoritmy, ktoré nie vždy nájdu najpravdepodobnejšie zarovnanie, ale pracujú oveľa rýchlejšie.

My sa budeme zaoberať algoritmom založeným na dynamickom programovaní, kde kvalita výsledného zarovnania je ovplyvnená len pravdepodobnostným modelom.

Základný model berie do úvahy len jednotlivé *bázy* a pravdepodobnosti *substitúcie* (*mutácie*), *inzercie* a *delécie*. Naš model bude navyše uvažovať aj dodatočné informácie získané napríklad z rôznych anotátorov.

Keďže množstvo dodatočnej informácie môže byť veľmi veľké, nie je možné skonštruovať vhodnú skórovaciu maticu pre zarovnávací algoritmus. Namiesto nej teda použijeme klasifikátor, ktorý natrénujeme na sekvenciách so známim zarovnaním a potom použijeme na zarovnanie nových sekvencií. V našom prípade sme si vybrali klasifikátor *Náhodný les* (*Random forest*).

V práci sa budeme zaoberať rôznymi spôsobmi tréovania, pričom navrneme vlastné modely a porovnáme už s existujúcimi. Ďalej sa budeme zaoberať spôsobom zberu dodatočných informácií, využitiu natrénovaného klasifikátora na zarovnávanie nových sekvencií aj pri *prezarovnaní* už existujúcich zarovnaní a implementujeme univerzálny framework, ktorý umožní využívať náš klasifikátor pri rôznych zarovnaníach.

Kapitola 1

Zarovnávanie sekvencií

V tejto kapitole si stručne popíšeme čo je to globálne a lokálne zarovnanie a ukážeme základné algoritmy na hľadanie globálneho a lokálneho zarovnania. Tieto algoritmy budeme neskôr s miernou modifikáciou používať pri našom riešení.

1.1 Podobnosť sekvencií a sekvenčná homológia

V prírode vznikajú nové sekvencie modifikáciou už existujúcich. Preto môžeme často spozorovať podobnosť medzi neznámou sekvenciou a sekvenciou o ktorej už niečo vieme. Ak zistíme podobnosti medzi sekvenciami, môžeme preniesť informácie o štruktúre a/alebo funkcii na novú sekvenciu.

Hovoríme, že dve súvisiace sekvencie sú *homologické*, a že prenášame informácie *podľa homológie*.

Na prvý pohľad rozhodnutie podobnosti dvoch biologických sekvencií nie je nič iné, ako rozhodnutie podobnosti dvoch textových reťazcov. Mnoho metód analýzy biologických sekvencií je preto zakorenená v informatike, kde je už mnoho literatúry týkajúcej sa tejto problematiky.

Vývoj sekvencií hromadí *inzercie*, *delécie* a *substitúcie*, takže predtým ako môže byť vyhodnotená podobnosť, treba urobiť zarovnanie sekvencií. Preto je zarovnanie sekvencií veľmi dôležité.

1.2 Skórovacie schémy

Takmer všetky metódy zarovnania hľadajú zarovnanie dvoch reťazcov na základe nejakej *skórovacej schémy*. Skórovacie schémy môžu byť veľmi jednoduché, napr. +1 za *zhodu* a -1 za *nezhodu*. Hoci ak chceme mať schému, kde biologicky najkorektnjšie zarovnanie má najvyššie skóre, musíme vziať do úvahy, že biologické sekvencie majú

evolučnú históriu, 3D štruktúru a mnohé ďalšie vlastnosti obmedzujúce ich primárnu evolúciu. Preto skórovací systém vyžaduje starostlivé premyslenie a môže byť veľmi zložitý.

1.3 Párové zarovnávanie

Párové zarovnávanie je základná úloha zarovnávanie sekvencií, kde sa k sebe zarovnávajú dve sekvencie. V tejto práci sa budeme zaoberať len párovým zarovnávaním.

Kľúčové problémy sú:

1. Aké typy zarovnávaní by sme mali uvažovať
2. Skórovací systém, ktorý použijeme na ohodnotenie zarovnaní
3. Algoritmus, ktorý použijeme na hľadanie optimálneho alebo dobrého zarovnaní podľa skórovacieho systému
4. Štatistické metódy na vyhodnotenie významnosti skóre zarovnaní.

1.3.1 Typy zarovnaní

Základné typy zarovnaní sú *Globálne zarovnanie* a *Lokálne zarovnanie*.

Pri globálnom zarovnaní je výstupom zarovnanie celých sekvencií s najvyšším skóre.

Pri lokálnom zarovnaní je výstupom zarovnanie nejakých poradiacov sekvencií s najvyšším skóre.

1.3.2 Algoritmy na hľadanie zarovnaní

Máme daný skórovací systém, potrebujeme algoritmus, ktorý nájde optimálne zarovnanie dvoch sekvencií. Budeme uvažovať zarovnávanie s medzermi. To znamená, že môžeme do sekvencie pridať ľubovoľne veľa medzier, aby sme dosiahli lepšie skóre. Pre 2 sekvencie dĺžky n existuje

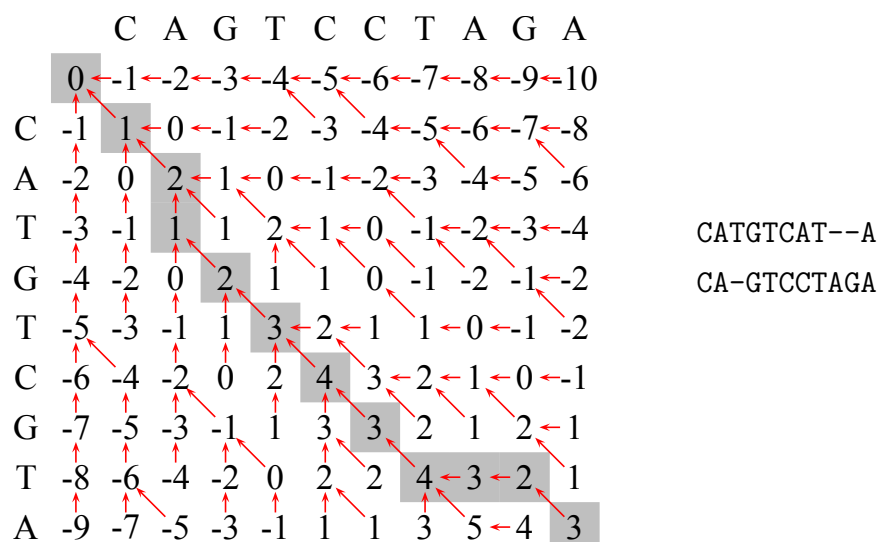
$$\binom{2n}{n} = \frac{(2n)!}{(n!)^2} \simeq \frac{2^{2n}}{\sqrt{\pi n}}$$

možných globálnych zarovnaní. Čiže nie je možné v rozumnom čase nimi prejsť.

Algoritmy na hľadanie zarovnaní využívajú *dynamické programovanie*. Algoritmy s dynamickým programovaním garantujú nájdenie optimálneho zarovnaní. Existujú aj heuristické algoritmy, ktoré môžu byť veľmi rýchle, avšak majú určité predpoklady a môže sa stať, že nenájdu najlepšie zarovnanie pre niektoré páry sekvencií. My sa budeme zaoberať len algoritmi využívajúcimi dynamické programovanie. Pre rôzne typy zarovnaní máme rôzne algoritmy zarovnávaní.

1.3.3 Algoritmus pre globálne zarovnanie: Needleman-Wunch

Máme dané 2 sekvencie $X = x_1x_2 \dots x_n$ a $Y = y_1y_2 \dots y_n$, budeme zarovnávať všetky znaky sekvencie X a všetky znaky sekvencie Y . Budeme používať jednoduché skórovanie: $+1$ za zhodu, -1 za nezhodu alebo pomlčku.



Obr. 1.1: Globálne zarovnanie

Algoritmus postupne vyplňa 2-rozmernú tabuľku A . Riadky zodpovedajú bázam sekvencie X a stĺpce bázam Y . Na políčku $A[i, j]$ bude skóre najlepšieho zarovnania prvých i báz sekvencie X a prvých j báz Y .

Ak zarovnáваме sekvenciu s prázdnu sekvenciou, tak skóre bude $-n$, kde n je dĺžka sekvencie. Bude tam n pomlčiek, každá nám dá skóre -1 . Takto vyplníme riadky a stĺpce $A[i, 0]$ a $A[0, j]$.

Ak chceme vyplniť políčku $A[i, j]$, musíme si uvedomiť ako môže vyzeráť posledný stĺpec zarovnania $x_1x_2 \dots x_i$ a $y_1y_2 \dots y_j$. Máme iba 3 možnosti ako môže vyzeráť posledný stĺpec najlepšieho zarovnania. Buď obsahuje x_i alebo y_j alebo oboje. Ak posledný stĺpec obsahuje oboje, cena tohto stĺpca je buď $+1$ ak $x_i = y_j$ alebo -1 ináč. AK by sme posledný stĺpec zmazali, dostali by sme zarovnanie $x_1x_2 \dots x_{i-1}$ a $y_1y_2 \dots y_{j-1}$, pričom musí ísť o najlepšie zarovnanie. To už máme vypočítané v políčku $A[i - 1, j - 1]$, čiže výsledné skóre bude $A[i - 1, j - 1] + s(x_i, y_j)$.

Ak posledný stĺpec obsahuje len x_i zarovnané s pomlčkou, skóre stĺpca bude -1 a po zmazaní dostávame zarovnanie $x_1x_2 \dots x_{i-1}$ a $y_1y_2 \dots y_j$, výsledné skóre bude teda $A[i - 1, j] - 1$. V prípade, že posledný stĺpec obsahuje len y_i , tak skóre vypočítame analogicky.

Najlepšie skóre bude maximum skóre pre všetky 3 prípady. Dostávame teda nasle-

dujúci vzťah pre výpočet $A[i, j]$:

$$A[i, j] = \max \begin{cases} A[i-1, j-1] + s(x_i, y_j) \\ A[i-1, j] - 1 \\ A[i, j-1] - 1 \end{cases}$$

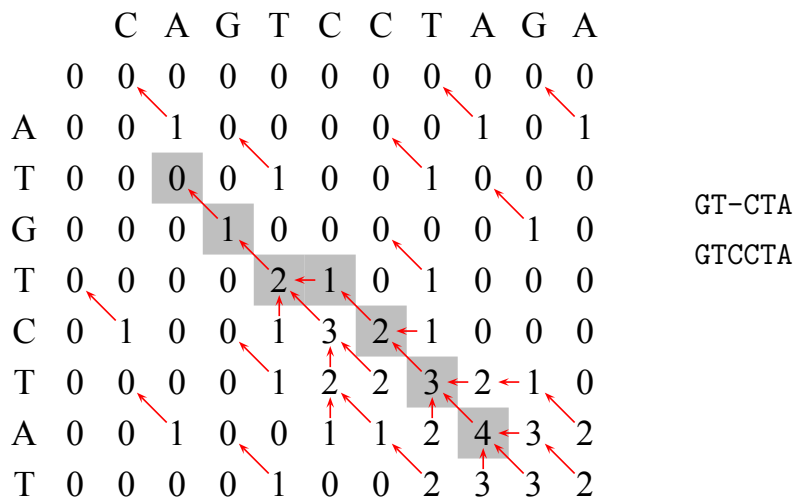
Tabuľku vieme vyplňať po riadkoch, pričom každé políčko vieme vypočítať z troch políčok, ktoré už máme vypočítané.

Ak nás zaujíma aj zarovnanie – nie len jeho skóre – vieme si pre každé políčko zapamätať ktorá z 3 možností dosiahla maximálnu hodnotu. Na základe tejto informácie potom vieme zrekonštruovať zarovnanie tak, že postupne z posledného políčka ($A[n, m]$) budeme prechádzať na políčko, z ktorého sme vypočítali aktuálnu hodnotu.

Časová zložitosť je $O(nm)$, pretože vyplníme nm políčok, každé v konštantnom čase. Zjavne aj pamäťová zložitosť je $O(nm)$.

Pamäťová zložitosť sa dá zredukovať na $O(n + m)$ za cenu zhruba dvojnásobného času výpočtu [Hir75].

1.3.4 Algoritmus pre lokálne zarovnanie: Smith-Waterman



Obr. 1.2: Lokálne zarovnanie

Algoritmus pre lokálne zarovnania sa líši len v niekoľkých malých detailoch. Opäť vyplníme tabuľku A , s tým, že v $A[i, j]$ bude najvyššie skóre lokálneho zarovnania medzi sekvenciami $x_1x_2 \dots x_i$ a $y_1y_2 \dots y_j$, ktoré buď obsahuje bázy x_i aj y_j , alebo je prázdne. Teda na ľubovoľnom mieste uvažujeme aj prázdne zarovnanie so skóre 0 (v

matici nebudú záporné čísla). Vzťah pre výpočet $A[i, j]$ vyzerá takto:

$$A[i, j] = \max \begin{cases} 0 \\ A[i - 1, j - 1] + s(x_i, y_j) \\ A[i - 1, j] - 1 \\ A[i, j - 1] - 1 \end{cases}$$

Časová aj pamäťová zložitosť sú, rovnako ako pri globálnom zarovnaní $O(nm)$.

Kapitola 2

Prehľad problematiky

Kapitola 3

Doplnkové informácie k sekvenciám

Kapitola 4

Random forests

Kapitola 5

Návrh riešenia

Kapitola 6

Implementácia

6.1 Simulátor

Kapitola 7

Zhodnotenie úspešnosti

Záver

Literatúra

- [BPSS11] Alvis Brazma, Helen Parkinson, Thomas Schlitt, and Mohammadreza Shojatalab. EBI Research - Functional Genomics - Introduction To Biology. 2011. [Online; accessed 26-Oct-2012].
- [BV11] Broňa Brejová and Tomáš Vinař. *Metódy v bioinformatike [Methods in Bioinformatics]*. Knižničné a edičné centrum FMFI UK, 2011. Lecture notes.
- [DEKM98] R. Durbin, S.R. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.
- [DGB06] Chuong Do, Samuel Gross, and Serafim Batzoglou. Contralign: Discriminative training for protein sequence alignment. In Alberto Apostolico, Concettina Guerra, Sorin Istrail, Pavel Pevzner, and Michael Waterman, editors, *Research in Computational Molecular Biology*, volume 3909 of *Lecture Notes in Computer Science*, pages 160–174. Springer Berlin / Heidelberg, 2006. 10.1007/11732990_15.
- [Hir75] D. S. Hirschberg. A linear space algorithm for computing maximal common subsequences. *Commun. ACM*, 18(6):341–343, June 1975.
- [NW70] Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443 – 453, 1970.
- [SW81] T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195 – 197, 1981.
- [YJEP07] Chun-Nam Yu, Thorsten Joachims, Ron Elber, and Jaroslaw Pillardy. Support vector training of protein alignment models. In Terry Speed and Haiyan Huang, editors, *Research in Computational Molecular Biology*, volume

4453 of *Lecture Notes in Computer Science*, pages 253–267. Springer Berlin / Heidelberg, 2007. 10.1007/978-3-540-71681-5_18.