

# Zarovňávanie sekvencií s použitím metód klasifikácie

Diplomová práca

Bc. Michal Hozza

**Vedúci práce:** Mgr. Tomáš Vinař, PhD.

**Konzultant:** Mgr. Michal Nánási

Fakulta matematiky, fyziky a informatiky, Univerzita Komenského, Bratislava

17.6.2014

## 1 Úvod

- Zarovnávanie sekvencií
- Ciele práce

## 2 Naše riešenie

- Klasifikácia na základe lokálnej informácie
- Zakomponovanie výsledkov klasifikácie do pHMM

## 3 Výsledky

## 4 Záver

# Zarovňavanie sekvencií

- jedným zo základných bioinformatických problémov
- identifikuje časti sekvencie, ktoré vznikli z toho istého predka (zarovnané bázy), inzercie a delécie v priebehu evolúcie (medzery v zarovnaní)

Príklad:

```
GTGGACCGTT-----CCTTCGGCAATCACGAGAAAAGCCACGT
GTTCGACCGTTTCAGTGACTTGAAGCAATCAGG---AACACCACCT
```

# Zarovňavanie sekvencií ako informatický problém

- Vstupom sú dve sekvencie  $X = x_1x_2 \dots x_n$  a  $Y = y_1y_2 \dots y_m$
- Výstupom je zarovnanie sekvencií, ktoré má najvyššie možné skóre podľa danej skórovacej schémy
- Skórovacia schéma môže byť napr.  $+1$  za zhodu a  $-1$  za medzeru alebo nezhodu
- V tomto prípade sa dá najlepšie zarovnanie nájsť v čase  $O(nm)$  dynamickým programovaním
- V praxi sa používajú zložitejšie schémy, v našej práci používame párové HMM

# Párové Skryté Markovovské modely (pHMM)

- pravdepodobnostný model inšpirovaný konečnými automatmi

## Definícia (pHMM)

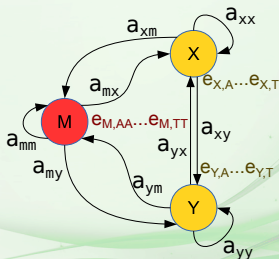
pHMM je 6-ica  $(K, \Sigma_X, \Sigma_Y, \pi, a, e)$ , kde  $K$  je množina stavov,  $\Sigma_X$  je množina symbolov  $X$  – ovej sekvencie,  $\Sigma_Y$  je množina symbolov  $Y$  – ovej sekvencie,  $\pi = \{\pi_i\}_{i \in K}$  je distribúcia začiatočných stavov,  $a = \{a_{i,j}\}_{i,j \in K}$  je distribúcia prechodov a  $e = \{e_{i,x,y}\}_{i \in K, x \in \Sigma_X \cup \{\varepsilon\}, y \in \Sigma_Y \cup \{\varepsilon\}}$  je distribúcia emisií.

- narozdiel od jednoduchého HMM generuje dve sekvencie, pričom môže mať stavy, ktoré generujú symboly v oboch sekvenciách, alebo len v jednej zo sekvencií

# Párové Skryté Markovovské modely (pHMM)

Príklad: pHMM pre zarovnávanie sekvencií

- $K = \{M, X, Y\}$ ,  $\Sigma_X = \Sigma_Y = \{A, C, G, T\}$
- M (*Match*) – emituje dvojice báz:  $AA, AC, AG, \dots, TT$
- X (*Insert X*) a Y (*Insert Y*) – emitujú jednotlivé bázy:  $A, C, G, T$



# Párové Skryté Markovovské modely (pHMM)

- Postupnosť stavov definuje konkrétne zarovnanie vstupných sekvencií
- Pri zarovnávaní hľadáme najpravdepodobnejšiu postupnosť stavov (pomocou Viterbiho algoritmu)

Príklad:

- Vstupné sekvencie

```
GTGGACCGTTCCTTCCGGCAATCACGAGAAAAGCCACGT
GTCGACCGTTTCAGTGACTTGAAGCAATCAGGAACACCACCT
```

- Jedno z možných zarovnaní spolu so stavmi

```
GTGGACCGTT-----CCTTCCGGCAATCACGAGAAAAGCCACGT
MMMMMMMMMMYYYYYYMMMMMMMMMMMMMMMMMMXXXMMMMMMMMMM
GTCGACCGTTTCAGTGACTTGAAGCAATCAGG---AACACCACCT
```





- cieľom práce je vytvoriť nové metódy na korekciu zarovnaní biologických sekvencií na základe prídavnej informácie
- integrácia tejto informácie bude zabezpečená pomocou techník využívaných na klasifikáciu v strojovom učení
- vzhľadom ku komplexnosti pridanej informácie, je veľmi ťažké systematickým spôsobom zakomponovať anotáciu do skórovacích schém založených na pHMM
- modelovanie takejto informácie v rámci pHMM by bolo neefektívne

# Zhrnutie nášho riešenia

- ① informáciu zosumarizujeme do jedného čísla
  - pre pár pozícií  $(i, j)$  chceme číslo  $S(i, j)$ , ktoré na základe lokálnej informácie vyjadruje, ako dobre k sebe príslušné pozície pasujú
  - to je vhodná úloha pre klasifikátory
- ② navrhujeme systematický spôsob, ako toto číslo zakomponovať do skórovacej schémy pHMM

# Klasifikácia na základe lokálnej informácie

- anotácie sme zakomponovali pomocou dvoch typov klasifikátorov
  - ① Match – rozhoduje či dané pozície majú byť zarovnané k sebe
  - ② Indel – rozhoduje či daná pozícia má byť zarovnaná k medzere
- výstup  $\in \langle 0, 1 \rangle$  – istota klasifikátora, že dané dve pozície majú byť zarovnané k sebe (v Indel klasifikátore, že daná pozícia má byť zarovnaná k medzere)
- atribúty sú okná veľkosti  $w$

# Okno pre klasifikátor

i:012345678 9  
Ax:00011111 0  
X:ACCATTCCTA--C  
Y:ACG----TGTTC  
Ay:000 11111  
j:012 34567

(a) Match klasifikátor

i:012345678 9  
Ax:00011111 0  
X:ACCATTCCTA--C  
Y:ACG----TGTTC  
Ay:000 11111  
j:012 34567

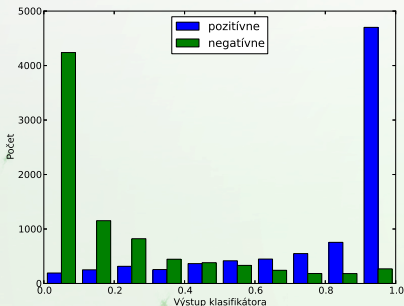
(b) Indel klasifikátor

Obr. : Okno klasifikátora pre pozície  $i = 6$  a  $j = 3$

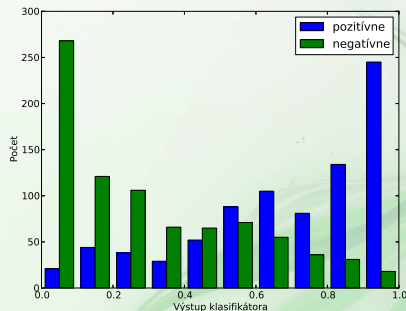
# Klasifikácia na základe lokálnej informácie

- k dátam sme pridali informácie o zhodách na zodpovedajúcich pozíciách, čím sa nám podarilo vylepšiť úspešnosť klasifikátora
- úspešnosť Match klasifikátora: **89,87%**
- úspešnosť Indel klasifikátora: **81,78%**
- klasifikátor sa dokáže naučiť, ktoré okná majú byť zarovnané k sebe a ktoré nie

# Úspešnosť klasifikátora



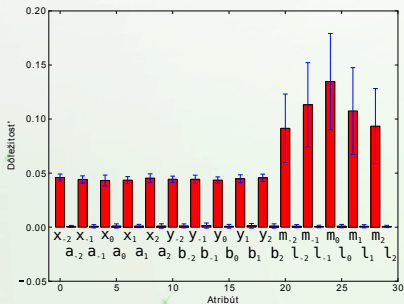
(a) Match klasifikátor



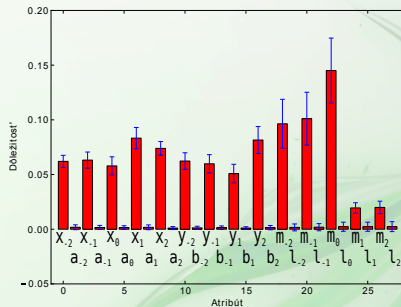
(b) Indel klasifikátor

Obr. : Distribúcia výstupu klasifikátora pre pozitívne a negatívne príklady.

# Dôležitosť atribútov



(a) Match klasifikátor



(b) InDel klasifikátor

Obr. : Dôležitosť atribútov v klasifikátore.

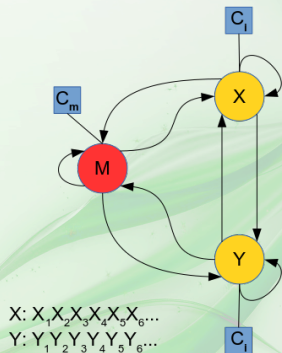
# Zakomponovanie výsledkov klasifikácie do pHMM

- skonštruovali sme dva modely pre zarovnanie sekvencií s anotáciami za pomoci klasifikátora
  - 1 Model s klasifikátorom ako emisiou
  - 2 Model s klasifikátorovou páskou
- založené na párových skrytých Markovovských modeloch.



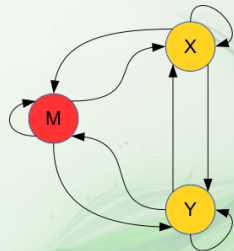
# Model s klasifikátorom ako emisiou (Model A)

- emisné tabuľky stavov nahradíme výstupom z klasifikátora
- model nie je korektný pravdepodobnostný model, pretože výstupy klasifikátora nesčítajú do 1
- prechodové pravdepodobnosti sme naštudovali zo zarovnaní z trénovacej vzorky



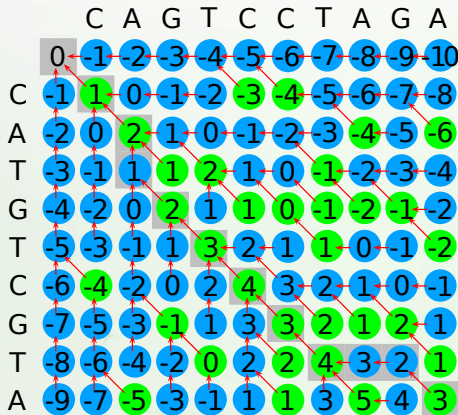
# Model s klasifikátorovou páskou (Model B)

- modelujeme navyše sekvenciu výstupov klasifikátora vo forme pásky
- trénujeme všetky parametre na trénovacej vzorke zarovnaní obohatenej o pásku s výstupmi z klasifikátora
- páska je cesta v 2D tabuľke výstupov klasifikátorov
- zhoduje sa s cestou zarovnania
- ak sa pohneme horizontálne, alebo vertikálne, používame Indel klasifikátor a ak sa pohneme diagonálne, tak použijeme Match klasifikátor



X:  $x_1 x_2 x_3 x_4 x_5 x_6 \dots$   
Y:  $y_1 y_2 y_3 y_4 y_5 y_6 \dots$   
C:  $c_1 c_2 c_3 c_4 c_5 c_6 \dots$

# Klasifikátorová páska



CATGTCAT--A

CA-GTCCTAGA

MMIMMMMIIM

Obr. : Použité klasifikátory v klasifikátorovej páske

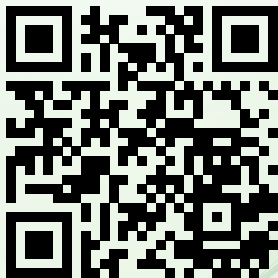
Dáta	Model A		Model B		Ref. Model		Muscle	
	Zhoda	Tranz.	Zhoda	Tranz.	Zhoda	Tranz.	Zhoda	Tranz.
sim1	79,75%	44,97%	84,35%	56,5%	<b>85,78%</b>	<b>61,03%</b>	82,72%	58,76%
sim2	70,14%	—	<b>71,47%</b>	—	60,38%	—	61,47%	—
bio	<b>91,40%</b>	96,63%	91,24%	<b>96,89%</b>	91,34%	96,45%	91,28%	95,98%

**Tabuľka :** Porovnanie našich modelov s referenčným modelom a zarovnávačom muscle.

- Tranzitivitu počítame z troch zarovnaní  $AB$ ,  $BC$  a  $AC$ , ako percentuálnu zhodu medzi zložením prvých dvoch zarovnaní  $(AB \circ BC)$  a tretieho zarovnaní  $AC$ .

- navrhli sme spôsob zakomponovania dodatočnej informácie do zarovnávaní sekvencií
- vytvorili sme vhodnú sadu atribútov pre klasifikátory
- vytvorili sme dva modely, ktoré zahŕňajú výstup klasifikátora do zarovnania
- úspešne sme implementovali zarovnávač s dodatočnou informáciou
  - naše riešenie je ľahko rozšíriteľné (umožňuje pridanie nových modelov, sady atribútov, výmenu klasifikátora ...)
- podarilo sa nám prekonať úspešnosť referenčných zarovnávačov bez anotácie na biologických dátach

# Ďakujem za pozornosť!



<https://github.com/mhozza/realigner>

# Čo považujeme za heuristiku?

- Algoritmy, ktoré hľadajú optimálne zarovnanie vzhľadom na nejakú skórovaciu schému a vždy ho nájdú nepovažujeme za heuristické. Takýmto algoritmom je aj Viterbiho algoritmus, ktorý používame v našej práci. Skórovacia schéma je v našom prípade pHMM s klasifikátorom.
- Okrem toho existujú rôzne heuristické algoritmy, ktoré pracujú rýchlejšie za cenu nenájdenia optimálneho zarovnania.

# Dôvody pre výber klasifikátora Náhodný les

- V práci sme sa nezaoberali porovnaním rôznych klasifikačných algoritmov
- Namiesto toho sme implementovali zarovnávač, v ktorom je možné jednoducho klasifikátor vymeniť za iný
- Za najlepší tento klasifikátor považujú jeho autori v článku [Breiman, 2001]
- Nás zaujímalo hlavne to, že funguje dostatočne dobre, a že je rýchly
- V rámci pokusov sme vyskúšali aj algoritmus SVM (dosahoval značne horšiu úspešnosť a bol pomalý) a rôzne stromové algoritmy (dosahovali podobnú úspešnosť aj rýchlosť tréningu)
- Z dôvodu časovej náročnosti detailnejších experimentov sme takéto porovnanie nezahrnuli do našej práce



# Trénovanie náhodných lesov

- V práci sme si zaviedli dva typy listov – uzavretý a otvorený
- Uzavretý má priradenú triedu, otvorený nie. Jediný list na začiatku je otvorený (pokiaľ neobsahuje len dáta jednej triedy)
- List uzavrieme a priradíme mu triedu, ak všetky dáta v tomto liste patria do tej triedy. Keďže ide učenie s učiteľom, príslušnosť dát do danej triedy poznáme.
- Dátovú sadu, pre každý strom v lese, získame náhodným výberom  $N$  vektorov s opakovaním z danej trénovacej sady veľkosti  $N$ .
- Druhý prvok náhodnosti spočíva v tom, že v každom vrchole sa vyberie len malá podmnožina atribútov, z ktorých sa vyberá najinformatívnejší atribút
- Časť dát z globálnej sady sa nám v danej množine nevyskytne. Tieto dáta sú tzv. *out of bag* a OOB chyba je chyba počítaná na týchto dátach.

# Klasifikátory vs. modely

- v našich modeloch máme tri stavy dvoch typov – jeden Match stav a dva Insert stavy
- Match stav používa Match klasifikátor a Insert stavy používajú Indel klasifikátor

# DNA vs. proteíny

- z hľadiska modelov a algoritmov, medzi DNA a proteínmi nie je rozdiel
- rozdiel je vo vlastnostiach sekvencie a v dostupných anotáciách
- časť DNA (gény) kóduje proteíny, preto sa na zarovnávanie proteínov môžeme pozeráť ako na podproblém zarovnávanía DNA.



Breiman, L. (2001).

Random forests.

*Machine learning*, 45(1):5–32.