Lossless seeds for approximate string matching

Přibližné vyhledávání pomocí bezeztrátového filtrování

Karel Břinda
karel.brinda@fjfi.cvut.cz

Theoretical Informatics Group
Fakulta jaderná a fyzikálně inženýrská
České vysoké učení technické v Praze

Prezentace diplomové práce
Praha, 6. června 2013

VEDOUCÍ PRÁCE
Ing. Karel Klouda, Ph.D.
FIT ČVUT v Praze

KONZULTANT
prof. Gregory Kucherov, Ph.D.
Université Paris-Est Marne-la-Vallée

Cíle práce

i) Seznamte se s obecnou teorií související s přibližným vyhledáváním.
ii) Nastudujte existující výsledky o bezeztrátových seedech včetně nejnovějších příspěvků.
iii) Nastudujte kombinatorické metody používané pro konstrukci seedů.
iv) Udělejte systematický souhrn existujících výsledků.
v) Implementujte některé z algoritmů.
vi) Pokud to bude možné, zlepšete existující algoritmy používané pro konstrukci seedů.

Úvod do problému

DNA – Next Generation Sequencing

- Nové metody pro zpracování a analýzu DNA, produkují ohromné množství dat

Úvod do problému

DNA – Next Generation Sequencing

- Nové metody pro zpracování a analýzu DNA, produkují ohromné množství dat
- Nutnost rychlého přibližného vyhledávání

Úvod do problému

DNA – Next Generation Sequencing

- Nové metody pro zpracování a analýzu DNA, produkují ohromné množství dat
- Nutnost rychlého přibližného vyhledávání
    - Dán krátký řetězec T a dlouhý referenční genom G, hledáme výskyty podřetězců G, které jsou podobné T.

Úvod do problému

DNA – Next Generation Sequencing

- Nové metody pro zpracování a analýzu DNA, produkují ohromné množství dat
- Nutnost rychlého přibližného vyhledávání
    - Dán krátký řetězec T a dlouhý referenční genom G, hledáme výskyty podřetězců G, které jsou podobné T.
    - K měření podobnosti používána Hammingova vzdálenost.

Úvod do problému

- Klasické algoritmy – pomalé $\Longrightarrow$ zrychlení pomocí filtrace

vylepšené vyhledávání $\left\{ \begin{array}{l} \textbf{1. filtrační fáze} \\ \textbf{2. ověřovací fáze} \end{array} \right.$

Úvod do problému

- Klasické algoritmy – pomalé $\Longrightarrow$ zrychlení pomocí filtrace

  vylepšené vyhledávání $\left\{\begin{array}{l}\textbf{1. filtrační fáze}\\ \textbf{2. ověřovací fáze}\end{array}\right.$

- Myšlenka bezeztrátových seedů:

  „Podobné řetězce musí sdílet některé vzory.“ – tzv. **seedy**

Úvod do problému

- Klasické algoritmy – pomalé $\Longrightarrow$ zrychlení pomocí filtrace

vylepšené vyhledávání $\left\{ \begin{array}{l} \textbf{1. filtrační fáze} \\ \textbf{2. ověřovací fáze} \end{array} \right.$

- Myšlenka bezeztrátových seedů:

„Podobné řetězce musí sdílet některé vzory.“ – tzv. **seedy**

- Příklad: ###–#–#

. . . pozice korespondence
– . . . „žolíková pozice“

Úvod do problému

- Klasické algoritmy – pomalé $\Longrightarrow$ zrychlení pomocí filtrace

  vylepšené vyhledávání { **1. filtrační fáze**, **2. ověřovací fáze** }

- Myšlenka bezeztrátových seedů:

  „Podobné řetězce musí sdílet některé vzory." – *tzv.* **seedy**

  - Příklad: ###-#-#

    \# ... pozice korespondence
    \- ... „žolíková pozice"

- Na myšlence seedů postavena většina programů pro podobnostní hledání v genomech
  - BLAST, FASTA, PatternHunter, BLASTZ, PatternHunter 2, YASS, ...

Seedy

Definice

seed Q	libovolný řetězec nad abecedou $\mathcal{A} = \{$#, -$\}$
váha seedu Q	$w(Q) = \|Q\|_{\#}$, chceme maximalizovat
(m, k)**-problém**	m – délka porovn. řetězců, k – počet chyb
Q **řeší** (m, k)**-problém**	$\forall$ polohu k chyb $\quad \exists$ umístění seedu Q:
	# $\Longrightarrow$ „nechyba“ (korespondence znaků)

Seedy

Definice

seed Q — libovolný řetězec nad abecedou $\mathcal{A} = \{$#, -$\}$

váha seedu Q — $w(Q) = |Q|_{\#}$, chceme maximalizovat

(m,k)-problém — m – délka porovn. řetězců, k – počet chyb

Q řeší (m,k)-problém — $\forall$ polohu k chyb $\exists$ umístění seedu Q:

$\Longrightarrow$ „nechyba“ (korespondence znaků)

Příklad

$Q =$ ###--#-### řeší $(16,2)$-problém. Ukázka pro konkrétní situaci:

c	t	**t**	**a**	**g**	**g**	c	**t**	**c**	**c**	**g**	**g**	g	a	t	a
c	t	**t**	**a**	**g**	**a**	c	**t**	**t**	**c**	**g**	**g**	g	a	t	a
		#	#	#	-	-	#	-	#	#	#				

Seedy

Definice

seed Q	libovolný řetězec nad abecedou $\mathcal{A} = \{\texttt{\#}, \texttt{-}\}$
váha seedu Q	$w(Q) = \lvert Q \rvert_{\texttt{\#}}$, chceme maximalizovat
(m, k)**-problém**	m – délka porovn. řetězců, k – počet chyb
Q **řeší** (m, k)**-problém**	$\forall$ polohu k chyb $\exists$ umístění seedu Q:
	# $\Longrightarrow$ „nechyba“ (korespondence znaků)

Příklad

$Q =$ ###--#-### řeší $(16, 2)$-problém. Ukázka pro konkrétní situaci:

c	t	**t**	**a**	**g**	**g**	c	**t**	**c**	**c**	**g**	**g**	g	a	t	a
c	t	**t**	**a**	**g**	**a**	c	**t**	**t**	**c**	**g**	**g**	g	a	t	a
		#	#	#	-	-	#	-	#	#	#				

Problematice seedů se věnuje 112 článků.

Výsledky práce

i) Shrnuty dosud publikované výsledky.

Výsledky práce

i) Shrnuty dosud publikované výsledky.

ii) $(m, 1)$-problém:

- Podrobně analyzován.
- Navržen rychlý algoritmus pro hledání optimálních seedů.

Výsledky práce

i) Shrnuty dosud publikované výsledky.

ii) $(m, 1)$-problém:
- Podrobně analyzován.
- Navržen rychlý algoritmus pro hledání optimálních seedů.

iii) $(m, 2)$-problém
- Dokázáno nové kritérium pro řešitelnost.
- Navržen algoritmus, který vrací pro nízká m dobré seedy.
- Vyslovena hypotéza o struktuře optimálních seedů.

Výsledky práce

i) Shrnuty dosud publikované výsledky.

ii) $(m, 1)$-problém:
- Podrobně analyzován.
- Navržen rychlý algoritmus pro hledání optimálních seedů.

iii) $(m, 2)$-problém
- Dokázáno nové kritérium pro řešitelnost.
- Navržen algoritmus, který vrací pro nízká m dobré seedy.
- Vyslovena hypotéza o struktuře optimálních seedů.

iv) Vytvořeno několik programů.

$(m, 1)$-problém

Věta

Nechť je dán $(m, 1)$-problém a seed Q délky s. Označme $\ell = m - s$. Pak

$$Q \text{ řeší } (m,1)\text{-problém} \quad \Leftrightarrow \quad Q \text{ neobsahuje } \#^{\ell+1}.$$

$(m, 1)$-problém

Věta

Nechť je dán $(m, 1)$-problém a seed Q délky s. Označme $\ell = m - s$. Pak

$$Q \text{ řeší } (m,1)\text{-problém} \quad \Leftrightarrow \quad Q \text{ neobsahuje } \#^{\ell+1}.$$

Hledání optimálního seedu:

1. Pro každou délku $s \in \{2, 3, \ldots, m-1\}$ sestrojíme hladovým algoritmem kandidáta $Q^{(s)}$, tedy:

$$Q^{(s)} = \text{prefix } (\#^{\ell}-)^{\omega} \text{ délky } s.$$

2. Z kandidátů vyberem toho s nejvyšší váhou.

$(m, 2)$-problém: laserová metoda

Snaha o podobné kritérium $\implies$ laserová metoda

$(m, 2)$-problém: laserová metoda

Snaha o podobné kritérium $\Longrightarrow$ laserová metoda

Příklad (Incidenční matice seedu $Q =$ #--#---# v $(10, 2)$-problému)

$$M_Q = \left(\begin{array}{cc|cccccccc|cc}
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1
\end{array}\right).$$

$(m,2)$-problém: laserová metoda

Snaha o podobné kritérium $\Longrightarrow$ laserová metoda

Příklad (Incidenční matice seedu $Q =$ #--#---# v $(10,2)$-problému)

$$M_Q = \left(\begin{array}{cc|cccccccc|cc}
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1
\end{array}\right).$$

Věta

Seed Q řeší $(m,2)$-problém $\Leftrightarrow$ pokud incidenční matice M_Q neobsahuje „diagonální řetězec nul“ délky $\ell + 1$.

$(m, 2)$-problém: laserová metoda

Příklad ($(14, 2)$-problém, $s = 8$, $\ell = 6$)

$Q =$ ###-#--#

#	0	0	0	0	0	0	0							
#	0	0	0	0	0	0	0	0						
#	0	0	0	0	0	0	0	0	0					
-	1	1	1	1	1	1	0	0	0	1				
#	0	0	0	0	0	0	0	0	0	0	0			
-	1	1	1	1	1	1	0	0	0	1	0	1		
-	1	1	1	1	1	1	0	0	0	1	0	1	1	
#	0	0	0	0	0	0	0	0	0	0	0	0	0	0
							0	0	0	1	0	1	1	0
							0	0	0	1	0	1	1	0
							0	0	0	1	0	1	1	0
							0	0	0	1	0	1	1	0
							0	0	0	1	0	1	1	0
							0	0	0	1	0	1	1	0

$(m, 2)$-problém: laserová metoda

Příklad ($(14, 2)$-problém, $s = 8$, $\ell = 6$)

$Q =$ ###-#-##

#	0	0	0	**0**	0	**0**	0							
#	0	0	0	0	**0**	0	**0**	0						
#	0	0	0	0	0	**0**	0	**0**	0					
-	1	1	1	1	1	1	**0**	0	**0**	1				
#	0	0	0	0	0	0	0	**0**	0	**0**	0			
-	1	1	1	1	1	1	0	0	**0**	1	**0**	1		
#	0	0	0	0	0	0	0	0	0	**0**	0	**0**	0	
#	0	0	0	0	0	0	0	0	0	0	**0**	0	**0**	0
							0	0	0	1	0	1	0	**0**
							0	0	0	1	0	1	0	0
							0	0	0	1	0	1	0	0
							0	0	0	1	0	1	0	0
							0	0	0	1	0	1	0	0
							0	0	0	1	0	1	0	0

$(m, 2)$-problém: hladový algoritmus

1. Pro fixované m a všechna $s \in \{2, 3, \ldots, m-1\}$: hladově přidáváme lasery do –^{s}.
2. Z kandidátů vybereme seed s nejvyšší vahou.

$(m, 2)$-problém: hladový algoritmus

1. Pro fixované m a všechna $s \in \{2, 3, \ldots, m-1\}$: hladově přidáváme lasery do –^{s}.
2. Z kandidátů vybereme seed s nejvyšší vahou.

Získané seedy jsou tvaru

$$Q^{(s)} = \text{ prefix } (\#^{a}\text{–}\#^{c}\text{–}^{d})^{\omega} \text{ délky } s,$$

kde $a = \left\lfloor \frac{\ell}{2} \right\rfloor$, $c = \left\lfloor \frac{\ell-1}{4} \right\rfloor$, $d = \left\lfloor \frac{\ell}{4} \right\rfloor + 1$

$(m, 2)$-problém: hladový algoritmus

1. Pro fixované m a všechna $s \in \{2, 3, \dots, m-1\}$: hladově přidáváme lasery do $-^s$.
2. Z kandidátů vybereme seed s nejvyšší vahou.

Získané seedy jsou tvaru

$$Q^{(s)} = \text{ prefix } (\#^a - \#^c -^d)^\omega \text{ délky } s,$$

kde $a = \left\lfloor \frac{\ell}{2} \right\rfloor$, $c = \left\lfloor \frac{\ell-1}{4} \right\rfloor$, $d = \left\lfloor \frac{\ell}{4} \right\rfloor + 1$

Srovnání s ostatními metodami: w v závislosti na m a metodě

m	16	32	48	64	80	96	200	300	400	500
asympt. optim. metoda	1	3	10	12	21	30	88	150	210	284
perf. pravítka	7	14	23	31	41	50	109	166	224	283
las. metoda + hlad. alg.	7	16	26	35	46	57	128	197	266	337

$(m, 2)$-problém: hladový algoritmus

Věta (Kucherov, Noé, Roytberg, 2005)

Nechť je k fixované a $w(m)$ označuje maximální váhu seedu řešícího (m, k)-problém. Pak

$$m - w(m) \in \Theta\left(m^{\frac{k}{k+1}}\right)$$

$(m, 2)$-problém: hladový algoritmus

Věta (Kucherov, Noé, Roytberg, 2005)

Nechť je k fixované a $w(m)$ označuje maximální váhu seedu řešícího (m, k)-problém. Pak

$$m - w(m) \in \Theta\left(m^{\frac{k}{k+1}}\right)$$

- Pro každé k tedy musí platit:

$$\lim_{m \to +\infty} \frac{w(m)}{m} = 1.$$

$(m, 2)$-problém: hladový algoritmus

Věta (Kucherov, Noé, Roytberg, 2005)

Nechť je k fixované a $w(m)$ označuje maximální váhu seedu řešícího (m, k)-problém. Pak

$$m - w(m) \in \Theta\left(m^{\frac{k}{k+1}}\right)$$

- Pro každé k tedy musí platit:

$$\lim_{m \to +\infty} \frac{w(m)}{m} = 1.$$

- Pro hladový algoritmus však dostaneme odhad:

$$\frac{w(m, s)}{m} \leq \frac{3}{16}\left(4 - \frac{1}{m}\right).$$

$(m, 2)$-problém: hladový algoritmus

Věta (Kucherov, Noé, Roytberg, 2005)

Nechť je k fixované a $w(m)$ označuje maximální váhu seedu řešícího (m, k)-problém. Pak

$$m - w(m) \in \Theta\left(m^{\frac{k}{k+1}}\right)$$

- Pro každé k tedy musí platit:

$$\lim_{m \to +\infty} \frac{w(m)}{m} = 1.$$

- Pro hladový algoritmus však dostaneme odhad:

$$\frac{w(m, s)}{m} \leq \frac{3}{16}\left(4 - \frac{1}{m}\right).$$

$\Longrightarrow$ **Hladový algoritmus tedy obecně neposkytuje optimální ani asymptoticky optimální seedy.**

$(m, 2)$-problém: cyklická pravítka

Definice

$P \in \{\#, -\}^p$ nazveme (úplné) **cyklické pravítko**, pokud pro každé přirozené d existuje přirozené i tak, že

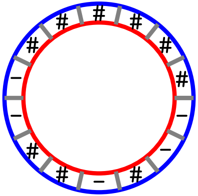
$$(P^\omega)_i = (P^\omega)_{i+d} = -$$

$(m,2)$-problém: cyklická pravítka

Definice

$P \in \{\#, -\}^p$ nazveme (úplné) **cyklické pravítko**, pokud pro každé přirozené d existuje přirozené i tak, že

$$(P^\omega)_i = (P^\omega)_{i+d} = -$$

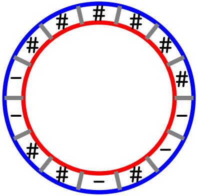


$(m, 2)$-problém: cyklická pravítka

Definice

$P \in \{\#, -\}^p$ nazveme (úplné) **cyklické pravítko**, pokud pro každé přirozené d existuje přirozené i tak, že

$$(P^\omega)_i = (P^\omega)_{i+d} = -$$



Věta

Uvažujme konkrétní $(m, 2)$-problém a délku seedů s. Označme $\ell := m - s$. Pak pro každé cyklické pravítko P délky $p \leq \ell + 1$ platí:

prefix P^ω délky s

řeší náš $(m, 2)$-problém.

Hlavní hypotéza

Hypotéza

Nechť seed Q řeší $(m, 2)$-problém, označme $\ell = m - s$. Pak existuje cyklické pravítko P délky $p \leq \ell + 1$ takové, že

- $(m, 2)$-problém řeší také seed

$$\tilde{Q} = \text{prefix } P^{\omega} \text{ délky } s,$$

- $w(Q) \leq w(\tilde{Q})$, tedy seed $\tilde{Q}$ je alespoň stejně tak dobrý jako Q.

Hlavní hypotéza

Hypotéza

Nechť seed Q řeší $(m, 2)$-problém, označme $\ell = m - s$. Pak existuje cyklické pravítko P délky $p \leq \ell + 1$ takové, že

- $(m, 2)$-problém řeší také seed
$$\tilde{Q} = \text{prefix } P^\omega \text{ délky } s,$$
- $w(Q) \leq w(\tilde{Q})$, tedy seed $\tilde{Q}$ je alespoň stejně tak dobrý jako Q.

Hlavní přínos: Dokázalo by se, že v optimálních seedech je struktura, a při hledání těchto seedů by se významně snížila složitost algoritmů.

Hlavní hypotéza

Hypotéza

Nechť seed Q řeší $(m, 2)$-problém, označme $\ell = m - s$. Pak existuje cyklické pravítko P délky $p \leq \ell + 1$ takové, že

- $(m, 2)$-problém řeší také seed

$$\tilde{Q} = \text{prefix } P^{\omega} \text{ délky } s,$$

- $w(Q) \leq w(\tilde{Q})$, tedy seed $\tilde{Q}$ je alespoň stejně tak dobrý jako Q.

Hlavní přínos: Dokázalo by se, že v optimálních seedech je struktura, a při hledání těchto seedů by se významně snížila složitost algoritmů.

Příklad ($(100, 2)$-problém, $s = 61$)

```
49   ###################-#########----------###################-##
50   ###################-#######-####--------###################-#
51   ###################-######-######-------###################-#
52   ###################-####-####-#####-----###################-#
53   ################-#####-###-###-####-#--################-#####
```

Hlavní hypotéza

Hypotéza

Nechť seed Q řeší $(m,2)$-problém, označme $\ell = m - s$. Pak existuje cyklické pravítko P délky $p \leq \ell + 1$ takové, že

- $(m,2)$-problém řeší také seed

$$\tilde{Q} = \text{prefix } P^{\omega} \text{ délky } s,$$

- $w(Q) \leq w(\tilde{Q})$, tedy seed $\tilde{Q}$ je alespoň stejně tak dobrý jako Q.

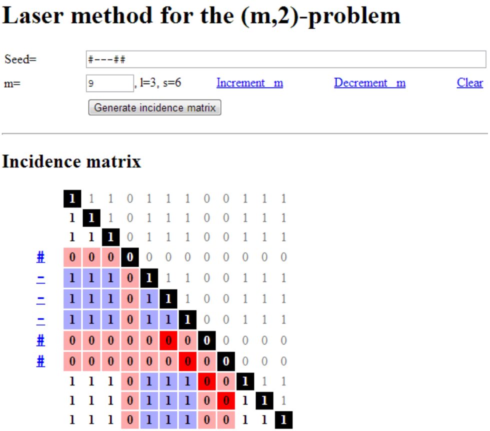
Hlavní přínos: Dokázalo by se, že v optimálních seedech je struktura, a při hledání těchto seedů by se významně snížila složitost algoritmů.

Příklad ($(100,2)$-problém, $s = 61$)

```
49   ###################-#########----------###################-##
50   ###################-#######-####--------###################-#
51   ###################-######-######-------###################-#
52   ###################-####-####-#####-----###################-#
53   ################-#####-###-###-####-#--################-#####
```

Přiložené programy

1. GUI pro laserovou metodu (HTML+JS)



2. Framework pro seedy, branch&bound (C++)

3. Několik skriptů pro hledání opt. seedů, hladový algoritmus, atd. (Python)

Závěr

Shrnutí:

i) $(m, 1)$-problém je nyní plně vyřešen.

ii) $(m, 2)$-problém stále plně vyřešen není. V rámci této práce bylo:

- vysloveno nové kritérium pro řešitelnost,
- vyšetřen hladový algoritmus,
- vylepšeny algoritmy pro hledání seedů,
- vyslovena hypotéza o struktuře optimálních seedů.

iii) (m, k)-problém, kde $k > 2$.

- Část výsledků z $(m, 2)$-problému je nejspíše možné zobecnit.

Závěr

Co dál:

i) Dokázat/vyvrátit hypotézu o tom, že nejlepší seedy lze získat pomocí cyklických pravítek.

ii) Důkladně popsat vlastnosti cyklických pravítek.

1. Zjistit, zda byly již v ekvivalentní formulaci studované v některé oblasti (algebra, teorie kódování, apod.).
2. Zjistit, zda je úloha hledání optimálních cyklických pravítek NP-úplná.
3. Popsat rychlé algoritmy pro jejich konstrukci a pro konstrukci jejich aproximací.

Děkuji za pozornost.