Gibbsovo vzorkování – analýza a modifikace

Otakar Trunda

Gibbsovo vzorkování - připomenutí

- Vstup: *n* sekvencí DNA $\{s_1, \ldots, s_n\}$, délka hledaného motivu *l*
- Algoritmus:
  - 1. Z každé sekvence s_i vyber vzorek a_i náhodně
  - 2. Vyber náhodně sekvenci s_i
  - 3. Vytvoř profil *X* ze vzorků $a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n$
  - 4. Spočítej pravděpodobnostní model pozadí *Q* ze sekvencí $s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n$
  - 5. Pro každý podřetězec *a* délky *l* ze sekvence s_i spočti $w(a) = P(a|X) / P(a|Q)$
  - 6. Vyber podřetězec *a* podle $w(a)$ ruletově
  - 7. Vzorek a_i nahraď řetězcem *a* a jdi na krok 2

Gibbsovo vzorkování - připomenutí

- Vstup: *n* sekvencí DNA $\{s_1, \ldots, s_n\}$, délka hledaného motivu *l*
- Algoritmus:
  - 1. Z každé sekvence s_i vyber vzorek a_i náhodně
  - 2. Vyber náhodně sekvenci s_i
  - 3. Vytvoř profil *X* ze vzorků $a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n$
  - 4. Spočítej pravděpodobnostní model pozadí *Q* ze sekvencí $s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n$
  - 5. Pro každý podřetězec *a* délky *l* ze sekvence s_i spočti $w(a) = P(a|X) / P(a|Q)$
  - 6. Vyber podřetězec *a* podle $w(a)$ ruletově
  - 7. Vzorek a_i nahraď řetězcem *a* a jdi na krok 2

Pravděpodobnostní model pozadí

- K čemu slouží?
  - Vyrovnávání šancí
  - Explorace x exploitace
- Jednoduchý přístup: pouze relativní četnosti jednotlivých znaků
  - Předpokládá, že znaky se vyskytují nezávisle na sobě
  - Tento předpoklad však není oprávněný
  - Přesnější model by měl přinést lepší výsledky (rychlejší konvergenci)

Pravděpodobnostní model pozadí

- Jak postihnout závislosti mezi výskyty znaků
- Při výpočtu vezmeme v úvahu také okolí znaků
  - Okolí: levé, pravé, oboustranné, jak velké
- Budeme počítat četnost dvojic, trojic, obecně *n*-tic znaků
  - Tedy použijeme tzv. Markovův řetězec řádu *n*
- Zvýšená časová a zejména paměťová složitost
- Zkusíme zhodnotit, zda přínos tohoto postupu převýší jeho cenu

Experimentální úloha

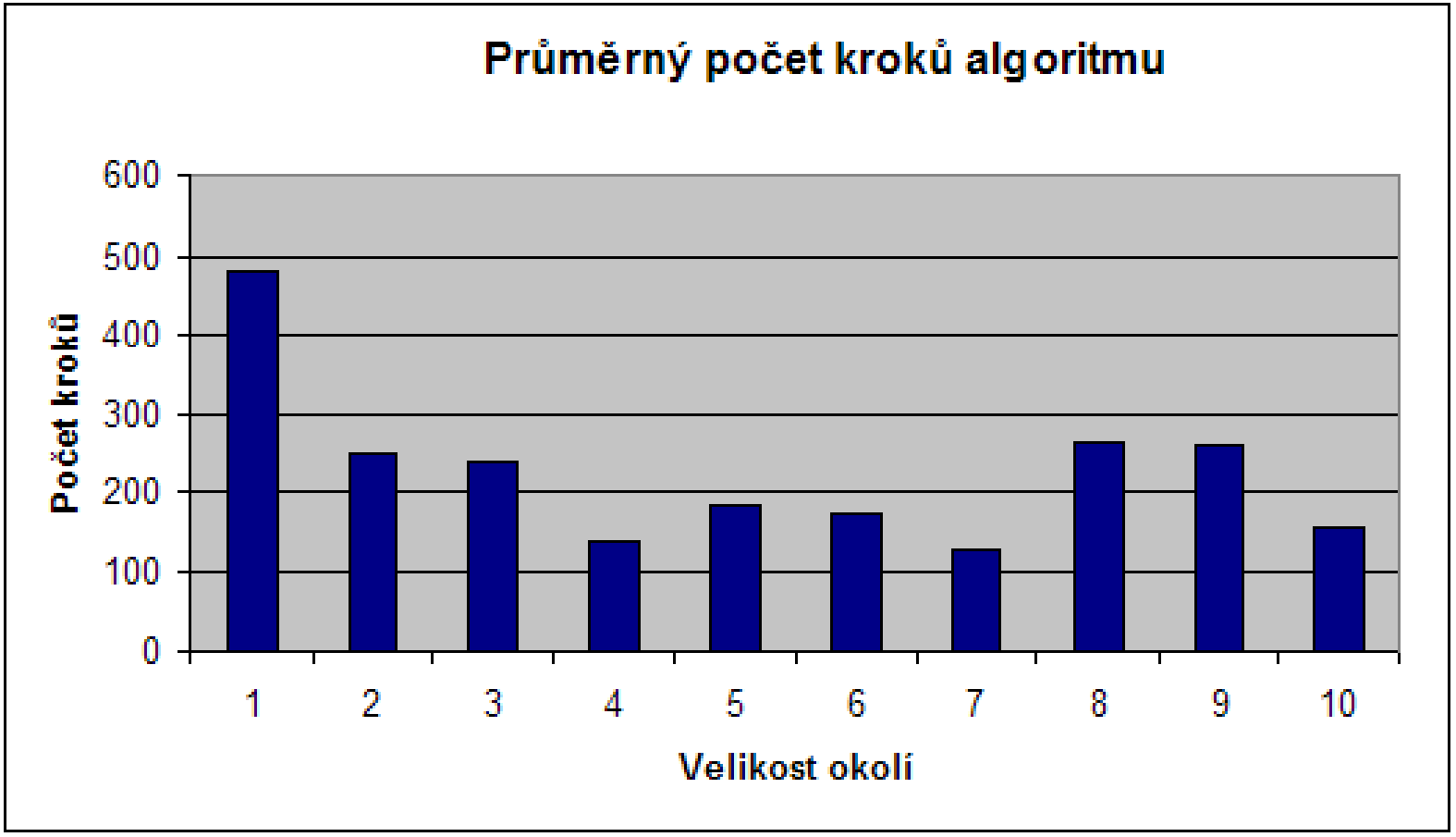
- Hledáme motiv délky 19 s nejvýše čtyřmi mutacemi
- 15 sekvencí délky 700
- Sledujeme rychlost konvergence algoritmu (počet průchodů hlavním cyklem)
- Algoritmus je randomizovaný – provádíme 10 pokusů, měříme střední hodnotu
- Předpokládaný výsledek: Při použití většího okolí bude konvergence rychlejší

Výsledky

- Výsledky všech deseti pokusů:

Velikost okolí	1	2	3	4	5	6	7	8	9	10
Výsledky pokusů	322	577	301	158	206	83	67	227	200	77
	455	106	438	78	94	103	491	262	412	157
	1526	161	538	70	61	234	54	113	220	133
	927	53	58	217	65	496	123	361	336	135
	482	112	69	205	363	377	108	265	386	141
	597	181	266	79	143	165	115	329	162	400
	504	762	127	389	140	90	211	221	175	89
	146	149	103	122	75	209	249	505	215	659
	115	523	613	98	365	78	86	122	173	100
	410	176	71	145	416	104	53	325	370	53
Průměr	548.4	280	258.4	156.1	192.8	193.9	155.7	273	264.9	194.4
Průměr 2	480.375	248.125	239.125	137.75	181.375	170.625	126.625	264	259.375	154

Výsledky


Průměrný počet kroků algoritmu
Počet kroků
600
500
400
300
200
100
0
1
2
3
4
5
6
7
8
9
10
Velikost okolí

Zhodnocení výsledků

- Využití okolí skutečně zrychlilo konvergenci algoritmu
- Další zvětšování okolí už nepřineslo podstatné zlepšení výkonu
- Časová složitost jen mírně horší (předzpracování)
- Paměťová složitost značně vzroste
- Závěr: Využití rozumně velkého okolí (v řádu jednotek) se vyplatí

Závěrečné poznámky k algoritmu

- Jak nastavit ukončovací podmínku?
- Posouvání (shifting) je nezbytné pro dosažení žádoucího výkonu
- Nápady na další modifikace:
  - Využití okolí i při počítání $P(a|X)$
  - Občasné použití „hladových“ kroků
  - Dynamicky měnit poměr explorace a exploitace
  - …