

Entwurf eines DNA- Algorithmus zur Lösung eines nichttrivialen informatischen Problems

Am Beispiel des Hitting-Set-Problems

Bearbeitet von Philipp Kahn und Ricardo Nußbaum

Gliederung

1. Aufgabenstellung und Herangehensweise
2. Hitting-Set
 - 2.1 Formale Definition
 - 2.2 Algorithmus und Beispiele
3. DNA-Algorithmus
 1. Vorbedingungen
 2. Algorithmus
 3. Beispiel
4. Analyse des DNA-Algorithmus
 1. Umsetzungs idee
5. Fazit
6. Quellen

1. Aufgabenstellung und Herangehensweise

- ▶ Entwurf eines DNA-Algorithmus zur Lösung eines nichttrivialen informatischen Problems
- ▶ Wahl fiel auf Hitting-Set-Problem
- ▶ Verarbeitung gut möglich durch mengenbasierte Beschreibung
- ▶ Aus Lösungsansätzen des Hitting-Set-Problems DNA-Algorithmus entwickeln

2. Hitting-Set

2.1 Formale Definition

- ▶ NP-Vollständig
- ▶ Gegeben:
 - ▶ $\{S_1, \dots, S_n\}$, wobei jedes S_i Teilmenge von T
 - ▶ Positive, ganze Zahl $k \leq |T|$
- ▶ Gesucht:
 - ▶ Teilmenge H von T , sodass $|H| \leq k$ und $H \cap S_i \neq \emptyset$, für jedes $i \in \{1, \dots, n\}$

2.2 Algorithmus und Beispiele

Durch Hinzufügen

1. H ist anfangs leer
2. Hinzufügen eines zufälligen Elements aus S_1 in H
3. Alle S mit $H \cap S_i \neq \emptyset$ zurücklegen
4. Falls $|H| < k$ und noch S vorhanden
 - ▶ Aus den restlichen S eins zufällig auswählen und aus diesem ein zufälliges Element wählen und zu H hinzufügen
 - ▶ Gehe zu 3.
5. Falls alle S zurückgelegt
 - ▶ Gebe H aus
6. Sonst Error

Durch Löschen

1. H enthält anfangs alle Elemente, welche in S vorkommen
2. Permutiere H
3. $i = 1$
4. Falls $|H| > k$ und $i \leq |H|$
 - ▶ Lösche aus H das i-te Element e
 - ▶ Ist H noch immer ein Hitting-Set für die Menge aller S?
 - ▶ Falls Ja: gehe zu 4.
 - ▶ Falls Nein: $i++$; füge e zu H an i-ter Stelle hinzu
5. Falls $|H| = k$
 - ▶ Gebe H aus
6. Sonst Error

Beispiel

Philipp Kahn, Ricardo Nußbaum

Friedrich-Schiller-Universität Jena

3. DNA-Algorithmus

Nung-Yue Shi; Chih-Ping Chu
Intelligent Systems Design and Applications, 2008. ISDA '08.
Eighth International Conference on, Issue Date: 26-28 Nov. 2008

3.1 Vorbedingungen

- ▶ Tubes T als Speichermedien
- ▶ Zahlendarstellung
 - ▶ n-Bit Nummer für jedes mögliche Hitting-Set
n = Anzahl aller Elemente der Grundmenge
 - ▶ i-tes Element ist im Subset vorhanden?:
 - ▶ Stelle i der n-Bit Nummer ist 1
 - ▶ Sonst 0
- ▶ Grundmenge $S = \{1,2,3,4\}$
liefert folgende mögliche Subsets

Subset	Encoding
\emptyset	0000
{1}	0001
{2}	0010
{3}	0100
{4}	1000
{1,2}	0011
{1,3}	0101
{1,4}	1001
{2,3}	0110
{2,4}	1010
{3,4}	1100
{1,2,3}	0111
{1,2,4}	1011
{1,3,4}	1101
{2,3,4}	1110
{1,2,3,4}	1111

3.2 Algorithmus

(1) Ausgangs-Tube T_0 (alle möglichen Subsets der Grundmenge)

(2) For each C_a :

For each b in C_a :

If b von C_a ist i -tes Element in S :

Subset mit i -tem Bit == 1 in T_b geben

Subset mit i -tem Bit == 0 in T_0 geben (T_0 belassen)

Lösche T_0

Erstelle neues T_0 durch Vereinigung aller T_b

(3) If T_0 nicht leer:

(1) T_0 ist Hitting-Set

► Grundmenge/Universum $S = \{1, 2, 3, 4, \dots, n\}$
Menge von Teilmengen $C = \{ \{C_1\}, \dots, \{C_m\} \}$

► Beispiel:

► Grundmenge/Universum $S = \{1, 2, 3, 4\}$
Menge von Teilmengen $C = \{ \{1, 2, 3\}, \{4\} \}$

3.3 Algorithmus Beispiel

- ▶ Grundmenge/Universum $S = \{1,2,3,4\}$ Menge von Teilmengen $C = \{\{1,2,3\}, \{4\}\}$
- ▶ T_0 ist Potenzmenge von S 
- ▶ Setze $a = 1$, $b = 1$
 - ▶ $T_1 = 0001\ 0011\ 0101\ 1001\ 0111\ 1011\ 1101\ 1111$
 - ▶ $T_0 = 0000\ 0010\ 0100\ 0110\ 1010\ 1100\ 1110\ 1000$
- ▶ Setze $a = 1$, $b = 2$
 - ▶ $T_2 = 0010\ 0110\ 1010\ 1110$
 - ▶ $T_0 = 0000\ 1100\ 1000\ 0100$
- ▶ Setze $a = 1$, $b = 3$
 - ▶ $T_3 = 1100\ 0100$
 - ▶ $T_0 = 0000\ 1000$
- ▶ Lösche T_0
- ▶ Neues $T_0 =$ Vereinigung aller T_b 
- ▶ Setze $a = 2$, $b = 1$
 - ▶ $T_1 = 1001\ 1010\ 1100\ 1011\ 1101\ 1110\ 1111$
 - ▶ $T_0 =$ irrelevant, da keine weiteren Elemente
- ▶ T_1 ist Lösung = $1001\ 1010\ 1100\ 1011\ 1101\ 1110\ 1111$

Subset	Encoding
{1}	0001
{2}	0010
{3}	0100
{1,2}	0011
{1,3}	0101
{1,4}	1001
{2,3}	0110
{2,4}	1010
{3,4}	1100
{1,2,3}	0111
{1,2,4}	1011
{1,3,4}	1101
{2,3,4}	1110
{1,2,3,4}	1111

4. Analyse des DNA-Algorithmus

4.1 Umsetzungsidee

- ▶ Codierung der n-Bit Zahlen durch Nucleotide
- ▶ Beispiel:
 - ▶ Adenin = 1
 - ▶ Guanin = 1
 - ▶ Cytosin = 0
 - ▶ Thymin = 0
- ▶ Einzelstränge betrachtet, oder bei Doppelstrang spiegelverkehrte Sequenzierung notwendig.

Subset	Encoding	Nucleotide
∅	0000	CCCC
{1}	0001	CCCA
{2}	0010	CCAC
{3}	0100	CACC
{4}	1000	ACCC
{1,2}	0011	CCAA
{1,3}	0101	CACA
{1,4}	1001	ACCA
{2,3}	0110	CAAC
{2,4}	1010	ACAC
{3,4}	1100	AACC
{1,2,3}	0111	CAAA
{1,2,4}	1011	ACAA
{1,3,4}	1101	AACA
{2,3,4}	1110	AAAC
{1,2,3,4}	1111	AAAA

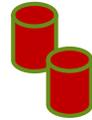
4.1 Umsetzungsidee



Ausgangs T_0 , Teilmengen C_i : Synthesis, Dilution



Redundante T_0 vorliegen



Teilmengen-Tubes mit Elementen codiert als n-Bit-Zahl codiert als Nucleotid-Sequenz

Sequencing → Bestimmung der Inhalte der C_i

4.1 Umsetzungsidee

► Abfolge einer For each b in C_a Schleife:



Cut von T_0

Agarosegel - Isolation

mit allen Kombinationen an Stelle $x_i = 1$

Längen 4 raus nehmen in neues T_0



Cut von T_0

Agarosegel - Isolation

mit allen Kombinationen an Stelle $x_i = 0$

Längen 4 raus nehmen in neues T_i

... ..

... Für alle Elemente b

Union

in neues T_0

Sequencing des Ergebnis-Tubes, Abstimmung auf K durch $K = \text{Anzahl der 1-en}$

5. Fazit

- Theoretisch „Praktische“ Umsetzung Schwierig
- Wenig verfügbare Forschungsliteratur
- Praktischer Nutzen durch noch zu lange Dauer der Teilschritte begrenzt

- Möglichkeiten !

6. Quellenangabe

- ▶ Vorlesung Molekulare Algorithmen
- ▶ <https://de.wikipedia.org/wiki/Hitting-Set-Problem>
- ▶ <https://people.cs.umass.edu/~barring/cs311/disc/10.html>
- ▶ Nung-Yue Shi; Chih-Ping Chu
Intelligent Systems Design and Applications, 2008. ISDA '08.
Eighth International Conference on, Issue Date: 26-28 Nov. 2008

Vielen Dank für Ihre
Aufmerksamkeit