Universität Hamburg
Fakultät für Mathematik,
Informatik und Naturwissenschaften
Department Informatik

Bachelorarbeit

Speichereffiziente Methoden zur Repräsentation von paarweisen Sequenz-Alignments

Thorben Wiese

3wiese@informatik.uni-hamburg.de
Studiengang B.Sc. Informatik
Matr.-Nr. 6537204
Fachsemester 6

Erstgutachter Universität Hamburg:	Prof. Dr. Stefan Kurtz
Zweitgutachter Universität Hamburg:	Dr. Giorgio Gonnella

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

IT .....

WWW .....

1 Einleitung

Problem darstellen

2 CIGAR-Strings

2.0.1 Komplexität

2.0.2 Speicherverbrauch

2.0.3 Grafiken

3 TracePoint Konzept

3.0.1 Komplexität

3.0.2 Speicherverbrauch

3.0.3 Grafiken

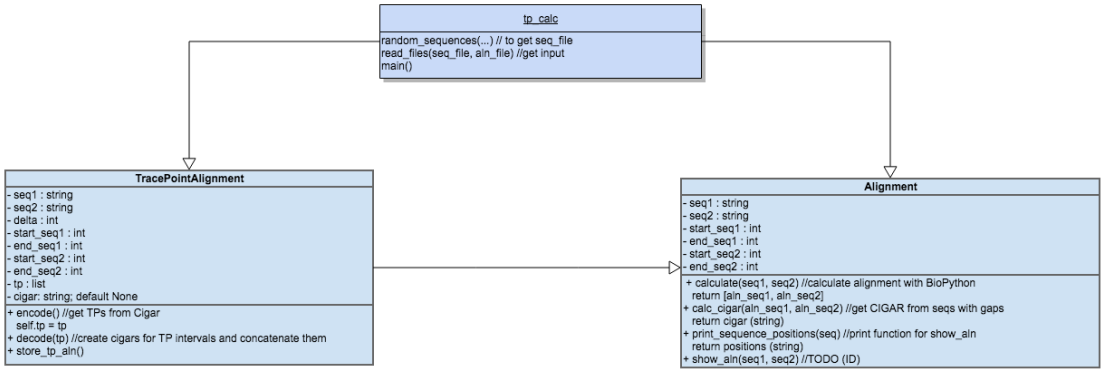
4 Optimierung

4.0.1 Delta-Kodierung

4.0.2 Grafiken Vergleich

5 Programm

5.0.1 Aufbau



5.0.2 Funktionalität

5.0.3 Grafiken Geschwindigkeit, Speicher

Algorithm 1 Computation of Trace Points from a given CIGAR-String

Input: $seq1, seq2, start\_seq1, end\_seq1, start\_seq2, \Delta, cigar$ mit
$|seq1|, |seq2|, |cigar| > 0$;
$start\_seq1, start\_seq2 \geq 0$;
$start\_seq1 < end\_seq1$ und
$\Delta > 0$

Output: Array TP of Trace Points

function *encode*($seq1, seq2, start\_seq1, end\_seq1, start\_seq2, \Delta, cigar$)
  $itv\_size \leftarrow MAX(1, \lceil start\_seq1/\Delta \rceil)$
  $itv\_count \leftarrow MIN(\lceil |seq1|/\Delta \rceil, \lceil |seq2|/\Delta \rceil)$
  for $i \leftarrow 0$ **upto** $|itv\_count|$ **do**

$$itv[i] \leftarrow \begin{cases} start\_seq1, itv\_size \cdot \Delta - 1 & \text{if } i = 0 \\ (itv\_size + i - 1) \cdot \Delta, (itv\_size + i) \cdot \Delta - 1 & \text{if } 0 < i < |itv\_count| \\ (itv\_size + i - 1) \cdot \Delta, end\_seq1 - 1 & \text{else.} \end{cases}$$

  end for
  $count1, count2, count3 \leftarrow 0$
  $TP \leftarrow$ Array for Trace Points
  for each (cig_count, cig_symbol) in cigar **do**
    for $i \leftarrow 0$ **upto** cig_count **do**
      if cig_symbol = 'I' **then**
        increment $count1$
      else if cig_symbol = 'D' **then**
        increment $count2$
      else
        increment $count1, count2$
      end if
      if count1 = intervals[count3][1] + 1 **and** count1 $\neq |seq1|$ **then**
        append (count2 - 1 + start_seq2) to TP
      end if
      if count $\neq |itv| - 1$ **then**
        increment $count3$
      end if
    end for
  end for
  return TP
end function

Algorithm 2 Computation of a CIGAR-String from a given Trace Point Array

Input: $seq1, seq2, \Delta, TP$ mit
$|seq1|, |seq2|, \Delta, |TP| > 0$

Output: CIGAR-String

1: **function** *decode*($seq1, seq2, \Delta, TP$)
2: $cig \leftarrow$ empty String
3: **for** $i \leftarrow 0$ **upto** $|TP|$ **do**
4: append to cig:

$$\begin{cases} \textbf{cigar}(seq1[0...\Delta], seq2[0...TP[i]+1]) & \text{if } i = 0 \\ \textbf{cigar}(seq1[i \cdot \Delta ... |seq1|], seq2[TP[i-1]+1 ... |seq2|]) & \text{if } i = |TP|-1 \\ \textbf{cigar}(seq1[i \cdot \Delta ... (i+1) \cdot \Delta], seq2[TP[i-1]+1]...TP[i]+1) & \text{else.} \end{cases}$$

5: **end for**
6: $cig \leftarrow$ **combine**(cig)
7: **return** cig
8: **end function**
9:
10: **function** *combine*($cigar$)
11: $cig \leftarrow$ empty String
12: $tmp \leftarrow 0$
13: **for** each (cig_count, cig_symbol) in cigar **do**
14: $tmp \leftarrow tmp + previous\_cig\_count$
15: **if** cig_symbol = previous_cig_symbol **then**
16: **if** not last element in cigar **then**
17: $tmp \leftarrow 0$
18: **end if**
19: **end if**
20: **if** last element in cigar **then**
21: append $(tmp + cig\_count, cig\_symbol)$ to cig
22: **end if**
23: **end for**
24: **return** cig
25: **end function**

6 Fazit

Problem durch Programm gelöst

Literaturverzeichnis

Eidesstattliche Erklärung

Ich versichere, dass ich die vorstehende Arbeit selbstständig und ohne fremde Hilfe angefertigt und mich anderer als der im beigefügten Verzeichnis angegebenen Hilfsmittel nicht bedient habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht.

Hamburg, den _____ Unterschrift: _____