Übungen zur Vorlesung Sequenzanalyse I

Universität Bielefeld, WiSe 2009/2010
Prof. Dr. Jens Stoye · Dipl.-Inform. Nils Hoffmann

http://wiki.techfak.uni-bielefeld.de/gi/GILectures/2009winter/SequenzAnalyse

Blatt 4 vom 5.11.2009
Abgabe in einer Woche vor Beginn der Vorlesung.

Aufgabe 1 Edit-Matrix (4 Punkte)
Die Edit-Distanz kann effizient mit Hilfe der Edit-Matrix berechnet werden.

1. Berechne die Edit-Matrix $D(x, y)$ für die Sequenzen $x =$ `HALALI` und $y =$ `HALO` und gib die Matrix, sowie die Edit-Distanz $d(x, y)$ der beiden Strings an.
   (Schreibe x vertikal links neben die Matrix und y oben horizontal an die Matrix.)
   Erstelle während der Berechnung von $D(x, y)$ auch die Backtracing-Matrix $E(x, y)$ und gib diese ebenfalls an.
2. Verwende $E(x, y)$, um *alle* optimalen Edit-Sequenzen zu finden und gib sie an.

Aufgabe 2 Edit-Distanzen (3 Punkte)
Die Rekurrenz zur Berechnung der Standard-Edit-Distanz mit Einheitskosten lautet
für $1 \leq i \leq |x|, 1 \leq j \leq |y|$:

$$D(i,j) = \min \begin{cases} D(i-1,\, j-1) + \mathbb{1}_{\{x[i] \neq y[i]\}} \\ D(i-1,\, j) + 1 \\ D(i,\, j-1) + 1 \end{cases}$$

Die Rekursionsbasis ist gegeben durch:

$$D(0,\, j) = j \text{ für } 0 \leq j \leq |x| \text{ und } D(i,\, 0) = i \text{ für } 0 \leq i \leq |y|$$

1. Wie sehen die Rekurrenz und die Basisfälle für die *Hamming-Distanz* aus?
2. Wie sehen die Rekurrenz und die Basisfälle für die *LCS-Distanz* aus?
3. Ist es möglich, die *Edit+Flip-Distanz* rekursiv zu berechnen? Welche Schwierigkeiten treten hierbei auf?

Aufgabe 3 Edit-Distanz, Backtracing (3 Punkte)
Die JAVA-Klasse `EditDistance` vom letzten Zettel wird im Folgenden um einige Methoden erweitert.

1. Schreibe in `EditDistance` eine Methode

   ```
   public int[][] calculateEditDistanceMatrix(String a, String b)
   ```

   welche die von Dir bisher in `getDistance` implementierte Berechnung der Edit-Distanz kapselt. Diese Methode wird dann entsprechend innerhalb von `getDistance` aufgerufen und die Rückgabe in einer lokalen Variablen `int[][] editDistanceMatrix` gespeichert.
2. Implementiere in der JAVA-Klasse `EditDistance` eine Methode

   ```
   public void printMatrix(int[][] editDistanceMatrix, String a, String b)
   ```

   die die Strings und die Einträge der Edit-Distanzmatrix nach deren Berechnung wie in Aufgabe 1.1 formatiert und auf der Konsole ausgibt.
3. Erweitere `EditDistance` um die Methode

   ```
   public String traceback(int[][] editDistanceMatrix)
   ```

und implementiere darin das Backtracing, um eine optimale Edit-Sequenz mit Hilfe der Edit-Distanzmatrix `editDistanceMatrix` zu finden und gib die vereinfachte Edit-Sequenz als String zurück. Eine solche vereinfachte Edit-Sequenz ist z.B. $\mathcal{CCS}$ für die Strings `aab` und `aac`. Erlaubt sind die Edit-Operationen copy ($\mathcal{C}$), substitute ($\mathcal{S}$), insert ($\mathcal{I}$) und delete ($\mathcal{D}$), wobei der Buchstabe c (s.h. Kapitel 3.5, Seite 16 im Skript) bei $\mathcal{S}$ und $\mathcal{I}$ aus den Strings rekonstruiert werden muss.

4. Implementiere abschließend eine Methode

   ```
   public void printAlignment(String a, String b, String editSequence)
   ```

   zur formatierten Ausgabe der alignierten Strings (als Beispiel, s.h. Kapitel 5.1, S.39 im Skript) auf der Konsole.

5. Rufe alle drei implementierten Methoden nach der Berechnung der Edit-Distanzmatrix innerhalb der Methode `getDistance` auf, um die Matrix, die Edit-Sequenz, sowie das Alignment auf der Konsole auszugeben.