

Needleman / Wunsch Algorithmus

ASE Übung

17.05.2019

Alignments

```
AGCAGCAGTCGTAGGAC-GCTG
  ||||| | ||| | | ||
CATAGCAGGC-TAGTGCTGATG
```

Alignment Scores

```
ACGAT
  || |
-CGCT
```

match_score = 1

mismatch_penalty = -1

gap_penalty = -2

$\Rightarrow -2 + 1 + 1 + -1 + 1 = 0$

Score Matrix

		-	C	G	C	T
-	+	-	-	-	-	-
-						
A						
C						
G						
A						
T						

Vorausgehende Indels

		-	C	G	C	T
	---	+	-----	-----	-----	-----
-		0	-2	-4	-6	-8
A		-2				
C		-4				
G		-6				
A		-8				
T		-10				

Berechnung der Einträge

		-	C	G	C	T
	---	+	-----	-----	-----	-----
-		0	-2	-4	-6	-8
A		-2	-1			
C		-4				
G		-6				
A		-8				
T		-10				

Berechnung der Einträge (cont.)

		-	C	G	C	T
	---	+	-----	-----	-----	-----
-		0	-2	-4	-6	-8
A		-2	-1	-3		
C		-4				
G		-6				
A		-8				
T		-10				

Berechnung der Einträge (cont.)

		-	C	G	C	T
	---	+	-----	-----	-----	-----
-		0	-2	-4	-6	-8
A		-2	-1	-3	-5	
C		-4				
G		-6				
A		-8				
T		-10				

Berechnung der Einträge (cont.)

		-	C	G	C	T
	---	+	-----	-----	-----	-----
-		0	-2	-4	-6	-8
A		-2	-1	-3	-5	-7
C		-4				
G		-6				
A		-8				
T		-10				

Berechnung der Einträge (cont.)

		-	C	G	C	T
	---	+	-----	-----	-----	-----
-		0	-2	-4	-6	-8
A		-2	-1	-3	-5	-7
C		-4	-1			
G		-6				
A		-8				
T		-10				

Berechnung der Einträge (cont.)

		-	C	G	C	T
	---	+	-----	-----	-----	-----
-		0	-2	-4	-6	-8
A		-2	-1	-3	-5	-7
C		-4	-1	-2		
G		-6				
A		-8				
T		-10				

Vollständige Matrix

		-	C	G	C	T
---	+	---	---	---	---	---
-		0	-2	-4	-6	-8
A		-2	-1	-3	-5	-7
C		-4	-1	-2	-2	-4
G		-6	-3	0	-2	-3
A		-8	-5	-2	-1	-3
T		-10	-7	-4	-3	0

Traceback Matrix

		-	G	C	G	T
-	+	-	-	-	-	-
-						
A						
C						
G						
A						
T						

		-	G	C	G	T
-	+	-	-	-	-	-
-						
A						
C						
G						
A						
T						

Traceback Matrix (cont.)

		-	G	C	G	T
	---	+	-----	-----	-----	-----
-		0	-2	-4	-6	-8
A		-2				
C		-4				
G		-6				
A		-8				
T		-10				

		-	G	C	G	T
	---	+	-----	-----	-----	-----
-		i	i	i	i	i
A		d				
C		d				
G		d				
A		d				
T		d				

Traceback Matrix (cont.)

		-	G	C	G	T
	---	+	-----	-----	-----	-----
-		0	-2	-4	-6	-8
A		-2	-1			
C		-4				
G		-6				
A		-8				
T		-10				

		-	G	C	G	T
	---	+	-----	-----	-----	-----
-		i	i	i	i	i
A		d	s			
C		d				
G		d				
A		d				
T		d				

Traceback Matrix (cont.)

		-	G	C	G	T
	---	+	---	---	---	---
-		0	-2	-4	-6	-8
A		-2	-1	-3	-5	-7
C		-4	-3	0	-2	-4
G		-6	-3	-2	1	-1
A		-8	-5	-4	-1	0
T		-10	-7	-6	-3	0

		-	G	C	G	T
	---	+	---	---	---	---
-		i	i	i	i	i
A		d	s	s	s	s
C		d	s	s	i	i
G		d	s	d	s	i
A		d	d	s	d	s
T		d	d	s	d	s

Traceback

		-	G	C	G	T
	---	+	---	---	---	---
-		i	i	i	i	i
A		d	s	s	s	s
C		d	s	s	i	i
G		d	s	d	s	i
A		d	d	s	d	s
T		d	d	s	d	s

T
T

Traceback (cont.)

		-	G	C	G	T
	---+	-----	-----	-----	-----	-----
-		i	i	i	i	i
A		d	s	s	s	s
C		d	s	s	i	i
G		d	s	d	s	i
A		d	d	s	d	s
T		d	d	s	d	s

AT
-T

Traceback (cont.)

		-	G	C	G	T
	---	+	---	---	---	---
-		i	i	i	i	i
A		d	s	s	s	s
C		d	s	s	i	i
G		d	s	d	s	i
A		d	d	s	d	s
T		d	d	s	d	s

GAT

G-T

Traceback (cont.)

		-	G	C	G	T
	---	+	---	---	---	---
-		i	i	i	i	i
A		d	s	s	s	s
C		d	s	s	i	i
G		d	s	d	s	i
A		d	d	s	d	s
T		d	d	s	d	s

CGAT

CG-T

Traceback (cont.)

		-	G	C	G	T
	---	+	---	---	---	---
-		i	i	i	i	i
A		d	s	s	s	s
C		d	s	s	i	i
G		d	s	d	s	i
A		d	d	s	d	s
T		d	d	s	d	s

ACGAT

GCG-T

Listen

```
>>> an_empty_list = []
>>> a_list_of_integers = [4, 3, 2, 1, 0]
>>> a_list_of_integers[1]
3
>>> a_list_of_integers[4] = 10
>>> a_list_of_integers
[4, 3, 2, 1, 10]
```

Geschachtelte Listen

```
>>> two_by_two_identity_matrix = [[1, 0], [0, 1]]
>>> two_by_two_identity_matrix[0][0]
1
>>> two_by_two_identity_matrix[1][0]
0
```

For-Schleife

```
>>> some_string = "abc"
>>> for c in some_string:
...     print(c)
...
a
b
c
```


For-Schleife + range() Funktion

```
>>> for i in range(10, 13):  
...     print(i)  
...  
10  
11  
12
```

max() Funktion

```
>>> max(0, 12, 5, 7)  
12
```

Dateien

```
cp -r /home/olzhabaev/Public/ase_uebung_05 .  
cd ase_uebung_05
```

Implementation

1. Erste Spalte der Score- und Traceback-Matrizen korrekt befüllen
2. Erste Zeile der Score- und Traceback-Matrizen korrekt befüllen
3. Geschachtelten Schleife: Inertion- und Deletion Scores berechnen
4. Substitution-Score berechnen
5. Max-Score ermitteln und in Score-Matrix schreiben
6. Operation ermitteln und in Traceback-Matrix schreiben

Frage 1

Wie wächst die Laufzeit und der Speicherbedarf der vorgestellten Version des Needleman-Wunsch Algorithmus in Abhängigkeit von den Längen der Sequenzen?

Konstante Operationen

```
for i = 0 to m:                # performed m + 1 times
    M[i][0] = gap_penalty * i # constant operation
    T[i][0] = 'd'             # constant operation
```

Score / Traceback Matrix

		-	G	C	G	T
---	+	-----	-----	-----	-----	-----
-		0	-2	-4	-6	-8
A		-2	-1	-3	-5	-7
C		-4	-3	0	-2	-4
G		-6	-3	-2	1	-1
A		-8	-5	-4	-1	0
T		-10	-7	-6	-3	0

		-	G	C	G	T
---	+	-----	-----	-----	-----	-----
-		i	i	i	i	i
A		d	s	s	s	s
C		d	s	s	i	i
G		d	s	d	s	i
A		d	d	s	d	s
T		d	d	s	d	s

Frage 2

In der Vorlesung wurde eine kompliziertere Version des Algorithmus vorgestellt, welche zwischen den Kosten für das Erzeugen und dem Erweitern von Lücken unterscheidet.

Wie verhält sich die Laufzeit und der Speicherbedarf dort?

Affine Gap Kosten

$$gap_penalty = gap_start + (n - 1) \times gap_extend$$

		-	G	C	G	T
-		0	-2	-4	-6	-8
A		-2	-1	-3	-5	-7
C		-4	-3	0	-2	-4
G		-6	-3	-2	1	-1
A		-8	-5	-4	?	
T						

Frage 3

Überlegen Sie, wie das Verfahren erweitert werden könnte, um ein optimales Alignment zwischen beliebig vielen Sequenzen zu berechnen.

Wie würde die Laufzeit und der Speicherbedarf dabei aussehen?

Polynomielles / Exponentielles Wachstum

