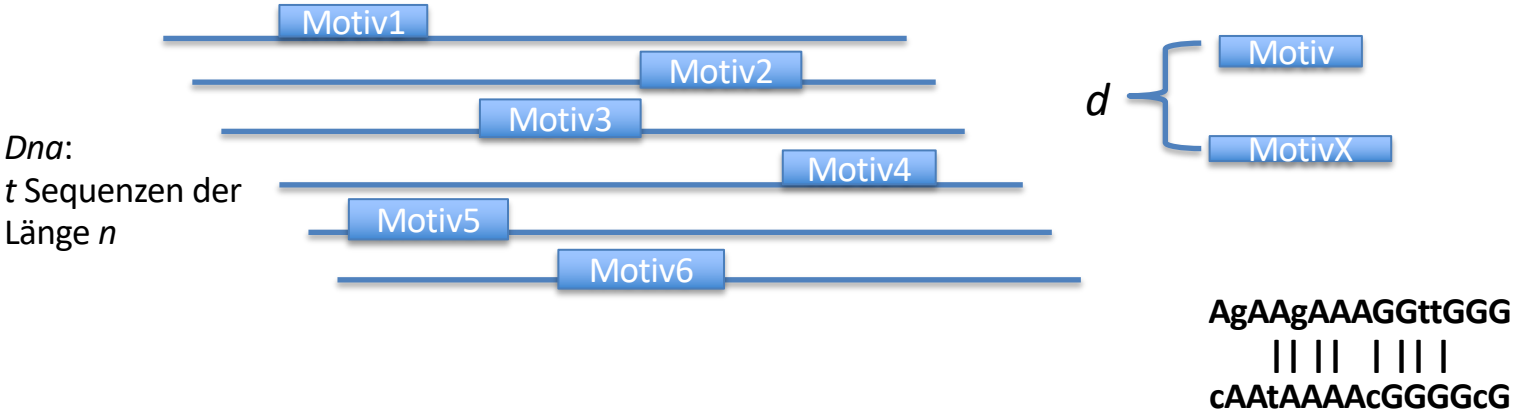
Algorithmen der Sequenzanalyse: Motivanalysen

AlgSeq – 04/11/2024

Prof. M. Sammeth

Implantierte Motive

(*k,d*)-Motiv := Sei *Dna* eine Menge von Strings und *d* ein Integer, dann ist ein *k*-mer ein **(*k,d*)-Motiv**, wenn es in jedem String aus *Dna* mit höchstens *d* Mismatches vorkommt.



Implantiertes Motiv Problem:
Finde alle (k,d)-Motive in einer Menge von Zeichenketten.

Input: Eine Menge von Zeichenketten Dna, Integer *k* und *d*.

Output: Alle (*k,d*)-Motive in *Dna*.

Implantiertes Motiv mit roher Gewalt lösen

ein “brute force” Ansatz für das ***Implantiertes Motiv*** Problem:

Algorithm: MOTIVEAUFZAEHLEN(Dna,k, d)

```
Patterns ← ∅
for jedes k-mer Muster in Dna do
    for jedes k-mer Muster', das sich von Muster mit höchstens d
     Unterschieden ("Mismatches") unterscheidet do
        if Muster' in jeder Zeichenkette von Dna mit höchstens d
         Mismatches vorkommt then
            Patterns ← Patterns ∪ {Muster'}
entferne alle Duplikate aus Patterns
return Patterns
```

paarweise Vergleiche zwischen Sequenzen helfen kaum (s.u.) → **zwei** Sprünge im Vergleich: *Muster* → *Muster'* → ZÄHLEN$_2$*(Muster')*

d-Nachbarschaften

Für *d*= 1 kann die Nachbarschaft (1-Nachbarschaft) mit einer Doppelschleife erzeugt werden:

Algorithm: UNMITTELBARENACHBARN(*Muster*)

```
Nachbarschaft ← {Muster}
for i = 1 to |Muster'| do
    Symbol ← i-tes Nukleotid von Muster
    for jedes Nukleotid x unterschiedlich von Symbol do
        Nachbar ← Muster mit dem i-ten Nukleotid substituiert durch x
        Nachbarschaft ← Nachbarschaft ∪ {Nachbar}
return Nachbarschaft
```

Wie kann uns dieser Algorithmus helfen,
um *d*-Nachbarschaften für *d*>1 zu generieren?

Iterative Lösung

Algorithm: NachbarnIterativ(*Muster*, *Muster*,*d*)

Nachbarschaft $\leftarrow \{$*Muster*$\}$
for $j \leftarrow 1$ bis d **do**
    for jede Zeichenkette *Muster′* in *Nachbarschaft* **do**
        Nachbarschaft $\leftarrow$
        Nachbarschaft $\cup$ UnmittelbareNachbarn(*Muster'*)
return *Nachbarschaft*

Algorithm: UnmittelbareNachbarn(*Muster*)

Nachbarschaft $\leftarrow \{$*Muster*$\}$
for $i = 1$ to $|$*Muster'*$|$ **do**
    Symbol $\leftarrow$ i-tes Nukleotid von *Muster*
    for jedes Nukleotid x unterschiedlich von *Symbol* **do**
        Nachbar $\leftarrow$ *Muster* mit dem i-ten Nukleotid substituiert durch x
        Nachbarschaft $\leftarrow$ *Nachbarschaft* $\cup \{$*Nachbar*$\}$
return *Nachbarschaft*

Rekursiver Ansatz

Algorithm: NACHBARN(*Muster*, *Muster*,*d*)

```
if d = 0 then
    return Muster
if |Muster| = 1 then
    return {A,C,G,T}
Nachbarschaft ← ∅
SuffixNachbarschaft ← NACHBARN(Suffix(Muster), d)
for jede Zeichenkette Text aus SuffixNachbarschaft do
    if HAMMINGDISTANZ(SUFFIX(Muster, d)) < d then
        for jedes Nukleotid x do
            Nachbarschaft ← Nachbarschaft ∪ x · Text
        else
            Nachbarschaft ←
              Nachbarschaft ∪ ERSTESSYMBOL(Muster) · Text
        Nachbarschaft ←
          Nachbarschaft ∪ UNMITTELBARENACHBARN(Muster')
return Nachbarschaft
```

Subtiles Motiv Problem

hier betrachtete Instanz des Problems (NF-$\varkappa$B Bindemotive): **Subtiles Motiv Problem**

Das 15-mer Motiv **AAAAAAAAGGGGGGG** *wurde mit vier zufälligen Mutationen implantiert in zehn 600nt (=typische Länge von upstream regulatorischen Regionen) Sequenzen:*

1 atgaccgggatactgatAgAAgAAAGGttGGGggcgtacacattagataaacgtatgaagtacgttagactcggcgccgccg
2 acccctatttttgagcagatttagtgacctggaaaaaaaatttgagtacaaaacttttccgaatacAAtAAAAcGGcGGGa
3 tgagtatccctgggatgacttAAAAtAAtGGaGtGGtgctctcccgatttttgaatatgtaggatcattcgccagggtccga
4 gctgagaattggatgcAAAAAAAGGGattGtccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggaga
5 tccctttgcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaatAtAAtAAAGGaaGGGcttatag
6 gtcaatcatgttcttgtgaatggatttAAcAAtAAGGGctGGgaccgcttggcgcacccaaattcagtgtgggcgagcgcaa
7 cggttttggcccttgttagaggcccccgtAtAAAcAAGGaGGGccaattatgagagagctaatctatcgcgtgcgtgttcat
8 aacttgagttAAAAAAtAGGGaGccctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgta
9 ttggcccattggctaaaagcccaacttgacaaatggaagatagaatccttgcatActAAAAAGGaGcGGaccgaaagggaag
10 ctggtgagcaacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagcttActAAAAAGGaGcGGa

Implantiertes Motiv mit roher Gewalt lösen

ein “brute force” Ansatz für das ***Implantiertes Motiv*** Problem:

Algorithm: MOTIVEAUFZAEHLEN(Dna,k, d)

```
Patterns ← ∅
for jedes k-mer Muster in Dna do
    for jedes k-mer Muster', das sich von Muster mit höchstens d
     Unterschieden ("Mismatches") unterscheidet do
        if Muster' in jeder Zeichenkette von Dna mit höchstens d
         Mismatches vorkommt then
            Patterns ← Patterns ∪ {Muster'}
entferne alle Duplikate aus Patterns
return Patterns
```

➡ paarweise Vergleiche zwischen Sequenzen helfen kaum (s.u.) → **zwei** Sprünge im Vergleich: *Muster* → *Muster'* → ZÄHLEN$_2$*(Muster')*

➡ *sehr langsam* für große Were von *k* und/oder *d* !

Motive bewerten

Motive:= eine (*t x k*) Matrix von Symbolen der *t* als Motive betrachteten *k*-mere.

Motive
(hier Bindestellen NF-$\varkappa$B)

```
T C G G G G g T T T t t
c C G G t G A c T T a C
a C G G G G A T T T t C
T t G G G G A c T T t t
a a G G G G A c T T C C
T t G G G G A c T T C C
T C G G G G A T T c a t
T C G G G G A T T c C t
T a G G G G A a c T a C
T C G G G t A T a a C C
```

SCORE*(Motive)*:= Anzahl der nicht-konservierten Symbole einer Spalte, d.h. Anzahl der Symbole, die nicht dem häufigsten Symbol der Spalte entsprechen.

SCORE*(Motive)* **3+4+0+0+1+1+1+5+2+3+6+4=30**

Motive bewerten (2)

ZÄHLEN*(Motive)* := (4 x *k*) Matrix der absoluten Häufigkeiten jedes DNA-Symboles.

ZÄHLEN*(Motive)*

```
A: 2 2  0  0 0 0 9 1 1 1 3 0
C: 1 6  0  0 0 0 0 4 1 2 4 6
G: 0 0 10 10 9 9 1 0 0 0 0 0
T: 7 2  0  0 1 1 0 5 8 7 3 4
```

PROFIL*(Motive)*: (4 x *k*) Matrix der relativen Häufigkeiten jedes DNA-Symboles.

PROFIL*(Motive)*

```
A: .2 .2 0 0 0  0  .9 .1 .1 .1 .3 0
C: .1 .6 0 0 0  0  0  .4 .1 .2 .4 .6
G: 0  0  1 1 .9 .9 .1 0  0  0  0  0
T: .7 .2 0 0 .1 .1 0  .5 .8 .7 .3 .4
```

CONSENSUSSTRING*(Motive)* := Konkatenation des häufigsten Symboles jeder Spalte
(zufällige Entscheidung falls mehrere zusammen am häufigsten)

CONSENSUSSTRING*(Motive)* **T C G G G G A T T T C C**

Das Problem mit dem Konsensus

NF-$\varkappa$B (Transkriptionsfaktor Bindestelle in *D.melanogaster*):

```
1 2 3 4 5 6 7  8   9 10 11  12
T C G G G G A T/C  T  T  C C/T
```

CSRE (Transkriptionsfaktor Bindestelle in *S.cerevisiae*):

```
1   2    3    4    5    6    7   8  9  10   11  12 13  14   15   16
C  G/C  G/T  T/A  C/T  G/C  C/G  A  T  G/T  C/G  A  T  C/T  C/T  G/T
```

Konsensus (Zeichenkette) kann Mehrdeutigkeiten nicht korrekt abbilden.

Entropie

Die Entropie *H*:
ist ein kondensiertes Maß der Unsicherheit über eine Wahrscheinlichkeitsverteilung

$H(p_1, \ldots, p_k) = -\sum_{i=1}^{k} p_i \, log_2(p_i)$, mit $\log_2(0) := 0$

z.B. in PROFIL(Motive) von NF-ϰB:

```
A:   .2   .2    0    0    0    0   .9   .1   .1   .1   .3    0
C:   .1   .6    0    0    0    0    0   .4   .1   .2   .4   .6
G:    0    0    1    1   .9   .9   .1    0    0    0    0    0
T:   .7   .2    0    0   .1   .1    0   .5   .8   .7   .3   .4
```

zweite Spalte: *H*(0.2, 0.6, 0.0, 0.2) ~ 1.371

konserviertere letzte Spalte: *H*(0.0, 0.6, 0.0, 0.4) ~ 0.971

5. Spalte – stark conserviert: *H*(0.0, 0.0, 0.9, 0.1) ~ 0.467

je *größer* die Konservierung, desto *kleiner* die Entropie

Motiv Logos

Informationsgehalt (IC *information content*) := 2 - $H(p_1, ..., p_N)$

je *größer* die Konservierung, desto *größer* der Informationsgehalt!

Das Motiv Findungs Problem

Motiv-Findungs Problem:
Gegeben eine Menge von Zeichenketten, finde *Motive,* eine Menge von k-meren mit je einem k-mer von jeder Zeichenkette, welche das SCORE(*Motive*) minimiert.

Eingabe: Eine Menge von Zeichenketten *Dna* und eine ganze Zahl *k*.

Ausgabe: Eine Menge *Motive* mit k-meren, eines von jeder Zeichenkette in Dna, welche SCORE(*Motive*) für alle möglichen *k*-mere minimiert.

```
 1 atgaccgggatactgatAgAAAgAAGGttGGGggcgtacacattagataaacgtatgaagtacgttagactcggcgccgccg
 2 acccctatttttgagcagatttagtgacctggaaaaaaaatttgagtacaaaacttttccgaatacAAtAAAAcGGcGGGa
 3 tgagtatccctgggatgacttAAAAtAAtGGaGtGGtgctctcccgatttttgaatatgtaggatcattcgccagggtccga
 4 gctgagaattggatgAAAAAAAAGGGattGtccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggaga
 5 tccctttgcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaatAtAAtAAAGGaaGGGcttatag
 6 gtcaatcatgttcttgtgaatggatttAAcAAtAAGGGctGGgaccgcttggcgcacccaaattcagtgtgggcgagcgcaa
 7 cggttttggcccttgttagaggcccccgtAtAAAcAAGGaGGGccaattatgagagagctaatctatcgcgtgcgtgttcat
 8 aacttgagttAAAAAAtAGGGaGccctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgta
 9 ttggcccattggctaaaagcccaacttgacaaatggaagatagaatccttgcatActAAAAAGGaGcGGaccgaaagggaag
10 ctggtgagcaacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagcttActAAAAAGGaGcGGa
```

(*k,d*)-Motiv := Sei *Dna* eine Menge von Strings und *d* ein Integer, dann ist ein *k*-mer ein **(*k,d*)-Motiv**, wenn es in jedem String aus *Dna* mit höchstens *d* Mismatches vorkommt.

Implantiertes Motiv Problem:
Finde alle (k,d)-Motive in einer Menge von Zeichenketten.

Input: Eine Menge von Zeichenketten Dna, Integer *k* und *d*.

Output: Alle (*k,d*)-Motive in *Dna*.

Motiv-Findungs Problem:
Gegeben eine Menge von Zeichenketten, finde *Motive,* eine Menge von k-meren mit je einem k-mer von jeder Zeichenkette, welche das SCORE(*Motive*) minimiert.

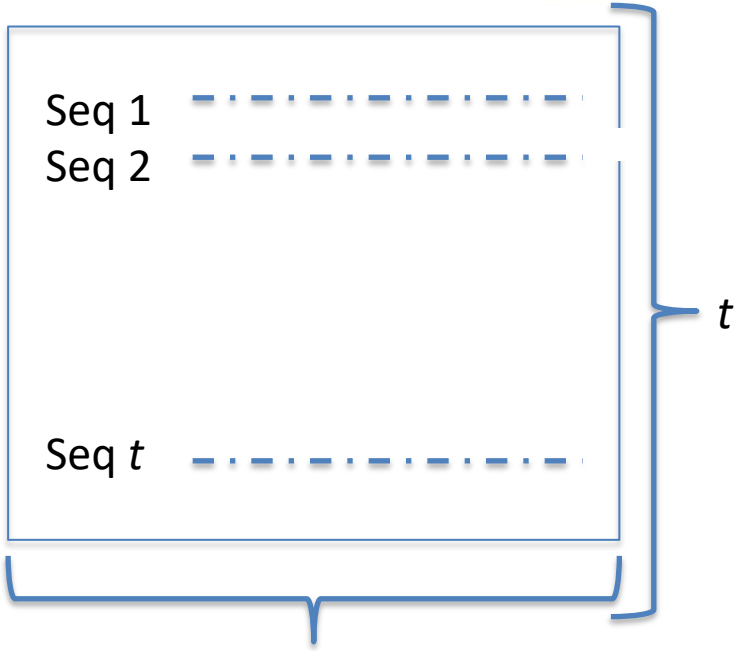
Eingabe: Eine Menge von Zeichenketten *Dna* und eine ganze Zahl *k*.

Ausgabe: Eine Menge *Motive* mit k-meren, eines von jeder Zeichenkette in Dna, welche SCORE(*Motive*) für alle möglichen *k*-mere minimiert.

Motiv-Suche “Brute Force”

“Brute Force” Methode:
BRUTEFORCEMOTIVSUCHE:

- für alle möglichen *k*-mer Kompositionen *Motive* in *Dna*
- berechne SCORE(*Motive*)
- bestimme das Minimum aller SCORE(*Motive*)



t Sequenzen in Dna
jede Sequenz mit *n* Nukleotiden
→ $(n - k + 1)$ *k*-mere in einer Sequenz
→ $(n - k + 1)^t$ Motiv-Matrizen
→ $k \cdot t$ Schritte für SCORE(*Motive*)

→ insgesamt:
$\mathcal{O}$(BRUTEFORCEMOTIVSUCHE) $\in \mathcal{O}(n^t \cdot k \cdot t)$

Übungen

(1) Beschreiben Sie für eine *t* x *k* Matrix *Motive* den minimalen und den maximalen Wert von SCORE*(Motive).*

(2) Alternativ bzw. analog zur Berechnung von SCORE*(Motive)* können auch die Entropie-Werte der einzelnen Spalten von *Motive* zu einer Gesamt-Entropie der Matrix aufaddiert werden. Bestimmen Sie nach dieser Regel die Entropie der Matrix des NF-$\varkappa$B Motives aus dem Seminar.

(3) Was ist der *maximale* und was der *minimale* Wert für den *Informationsgehalt*, und *bei welchen Eingaben* tritt jeder dieser Extremwerte auf?

(4) Implementieren Sie BRUTEFORCEMOTIVSUCHE(*Dna*, *k*) und versuchen Sie damit, die in den Folien verwendete "Miniatur" des subtilen Motiv Problemes mit *n*= 82 zu lösen.

Eingabe: Datei `Algo05_subtiles_motiv_mini.txt`

erwartete Ausgabe: das 15-mer AAAAAAAAGGGGGGG