Algorithmen der Sequenzanalyse: Motivanalysen

AlgSeq – 11/11/2024 (Helau)

Prof. M. Sammeth

Definitionen: Motive bewerten

(i) **Distanz eines *Musters* zu einer *Zeichenkette*:**

$d(Muster, Text) := \min_{\text{über alle } k\text{-mere } Muster' \text{ in Text}} \text{HammingDistanz}(Muster, Muster')$

z.B.:
$d($**GATTCATCA**, gcaaa**GACGCTGA**ccaa$) = 3$

(ii) **Distanz eines *Musters* zu einer Menge von *Zeichenketten*:**

$d(Muster, Dna) := \sum_{i=1}^{t} d(Muster, Dna_i)$

z.B.:
$d($**AAA**, $Dna) =$ **1** + **1** + **2** + **0** + **1** = **5**

Dna=

ttaccttAAC	1
gATAtctgtc	1
ACGgcgttcg	2
ccctAAAgag	0
cgtcAGAggt	1

Definitionen: Motive bewerten

(iii) **Ähnlichstes Muster in einer Zeichenkette**

MOTIV(*Muster*, *Text*) := $\underset{\text{über alle } k\text{-mere } Muster' \text{ in Text}}{\text{argmin}}$ HAMMINGDISTANZ(*Muster*, *Muster'*)

z.B.:
MOTIV(**GATTCTCA**, gacaaa**GACGCTGA**ccaa) = **GACGCTGA**
(kann mehrdeutig sein!)

(iv) **Änlichste Muster in einer Menge von Zeichenketten**

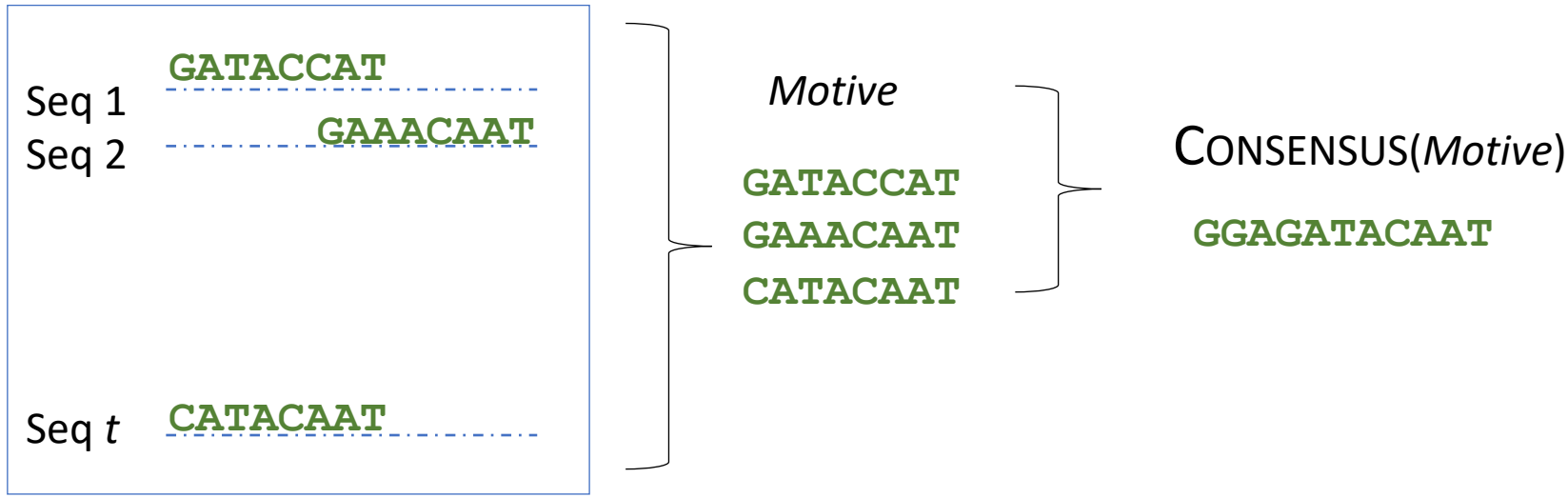
MOTIVE(*Muster*, *Dna*) := $\bigcup_{i=1}^{t}$ MOTIV(*Muster*, Dna_i)

z.B.: MOTIVE(**AAA**, *Dna*)= {**AAC**, **ATA**, **ACG**, **AAA**, **AGA**}

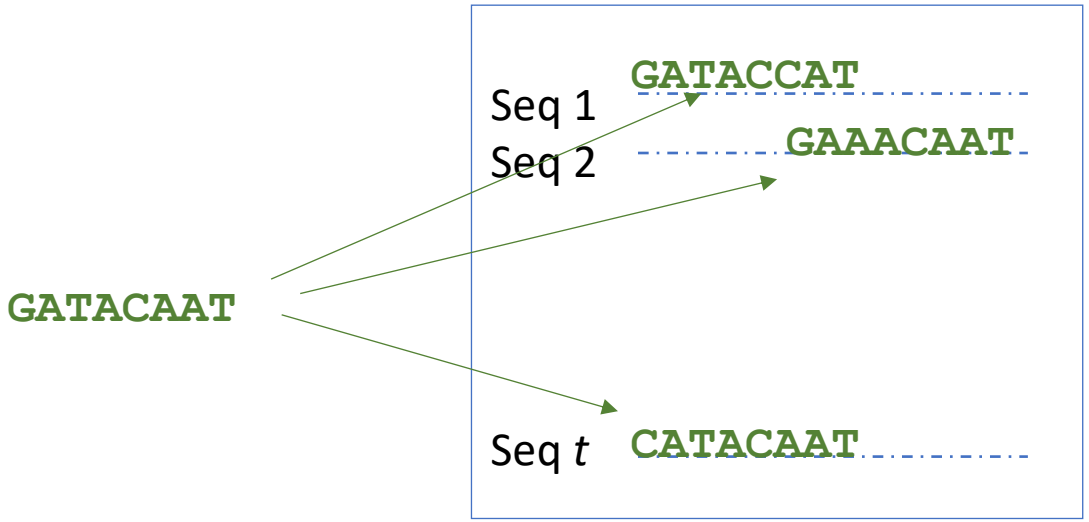
Dna=
ttaccttAAC
gATAtctgtc
ACGgcgttcg
ccctAAAgag
cgtcAGAggt

Eine neue Perspektive

BRUTEFORCEMOTIVSUCHE: ∀ *Motive* → suche CONSENSUS(*Motive*)



Alternative Strategie: ∀ CONSENSUS(*Motive*) → suche *Motive*



Re-Definition des Problemes

Motiv-Findungs Problem:
Gegeben eine Menge von Zeichenketten, finde eine Menge von k-meren,
mit je einem k-mer von jeder Zeichenkette, welche das Score des intrinsischen Motives 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.

Äquivalentes Motiv-Findungs Problem:
Gegeben eine Menge von Zeichenketten, finde ein Muster und eine Menge von k-meren,
(mit je einem k-mer von jeder Zeichenkette), welches die Distanz zwischen allen möglichen Mustern über alle möglichen Mengen von k-meren minimiert.

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

Ausgabe: Ein *k*-mer *Muster* und eine Menge *Motive* mit *k*-meren
(eines von jeder Zeichenkette in Dna), welche *d*(*Muster*, *Motive*)
für alle möglichen *Muster* und *Motive* minimiert.

Summen von SCORE(Motive) sind kommutativ

$$\text{SCORE(Motive)} = \sum_{Spalten\ j} \sum_{Zeilen\ i} I_{i,j} = \sum_{Zeilen\ i} \sum_{Spalten\ j} I_{i,j}$$, mit $I_{i,j}$ = { 0 *iff* $Dna_i(j, 1)$ = CONSENSUS(*j*, 1) ; 1 anderfalls

Motive	**T**	**C**	**G**	**G**	**G**	**G**	**g**	**T**	**T**	**T**	**t**	**t**	**3**
	c	**C**	**G**	**G**	**t**	**G**	**A**	**c**	**T**	**T**	**a**	**C**	**+ 4**
	a	**C**	**G**	**G**	**G**	**G**	**A**	**T**	**T**	**T**	**t**	**C**	**+ 2**
	T	**t**	**G**	**G**	**G**	**G**	**A**	**c**	**T**	**T**	**t**	**t**	**+ 4**
	a	**a**	**G**	**G**	**G**	**G**	**A**	**c**	**T**	**T**	**C**	**C**	**+ 3**
	T	**t**	**G**	**G**	**G**	**G**	**A**	**c**	**T**	**T**	**C**	**C**	**+ 2**
	T	**C**	**G**	**G**	**G**	**G**	**A**	**T**	**T**	**c**	**a**	**t**	**+ 3**
	T	**C**	**G**	**G**	**G**	**G**	**A**	**T**	**T**	**c**	**C**	**t**	**+ 2**
	T	**a**	**G**	**G**	**G**	**G**	**A**	**a**	**c**	**T**	**a**	**C**	**+ 4**
	T	**C**	**G**	**G**	**G**	**t**	**A**	**T**	**a**	**a**	**C**	**C**	**+ 3**

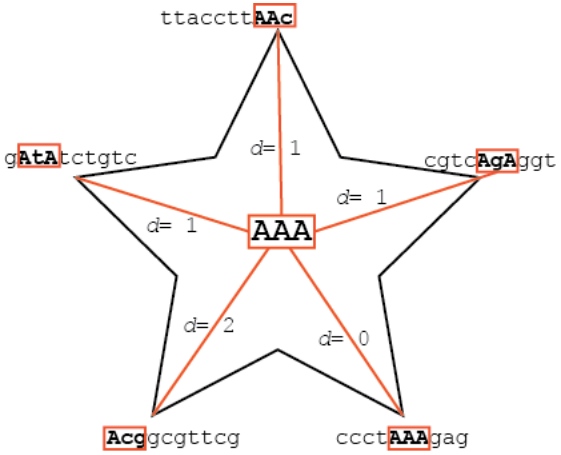
SCORE(*Motive*)	**3+**	**4+**	**0+**	**0+**	**1+**	**1+**	**1+**	**5+**	**2+**	**3+**	**6+**	**4**	**= 30**
CONSENSUS(*Motive*)	**T**	**C**	**G**	**G**	**G**	**G**	**A**	**T**	**T**	**T**	**C**	**C**	

Die Median-Zeichenkette

Median-Zeichenketten Problem:
Finde die Median Zeichenkette.

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

Ausgabe: Ein *k*-mer *Muster*, das *d*(*Muster*, *Dna*) über alle *k*-mere Muster minimiert.



Algorithm: MEDIANZEICHENKETTE(Dna,k)

$Distanz \leftarrow \infty$
for jedes k-mer von `AA...AA` bis `TT...TT` **do**
  if $Distanz > d(Muster, Dna)$ **then**
    $Distanz \leftarrow d(Muster, Dna)$
    $Median \leftarrow Muster$
return $Median$

Laufzeit-Vergleich

$\mathcal{O}(\text{BRUTEFORCEMOTIVSUCHE})$
$\in \mathcal{O}(n^t \cdot k \cdot t)$

vs.

$\mathcal{O}(\text{MEDIANZEICHENKETTE})$
$\in \mathcal{O}(4^k \cdot k \cdot n \cdot t)$

mit $k \leq 20$ (Motivlänge) und
$t \sim 10^3$ (tausende Sequenzen)

2

Mr. BRUTEFORCE

1

Mr. MEDIANZEICHENKETTE

Praktische Laufzeit

1 atgaccgggatactgat**AgAAAgAAGGttGGG**ggcgtacacattagataaacgtatgaagtacgttagactcggcgccgccg
2 acccctatttttgagcagatttagtgacctggaaaaaaaatttgagtacaaaacttttccgaata**cAAtAAAAcGGcGGG**a
3 tgagtatccctgggatgactt**AAAAtAAtGGaGtGG**tgctctcccgattttgaatatgtaggatcattcgccagggtccga
4 gctgagaattggatg**AAAAAAAAGGGattG**tccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggaga
5 tccctttgcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaat**AtAAtAAAGGaaGGG**cttatag
6 gtcaatcatgttcttgtgaatggattt**AAcAAtAAGGGctGG**gaccgcttggcgcacccaaattcagtgtgggcgagcgcaa
7 cggttttggcccttgttagaggcccccgt**AtAAAcAAGGaGGGc**caattatgagagagctaatctatcgcgtgcgtgttcat
8 aacttgagtt**AAAAAAtAGGGaGcc**ctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgta
9 ttggcccattggctaaaagcccaacttgacaaatggaagatagaatccttgcat**ActAAAAAGGaGcGG**accgaaagggaag
10 ctggtgagcaacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagctt**ActAAAAAGGaGcGG**a

Subtiles Motif Problem (upstream regionen):
15-mer Motiv **AAAAAAAAGGGGGGG** ist implantiert in zehn 600nt Sequenzen.

$$\mathcal{O}(\text{MedianZeichenkette}) \in \mathcal{O}(4^k \cdot k \cdot n \cdot t)$$

“Kleines” Subtiles Problem

1 atgaccgggatactgat**AgAAAgAAGGttGGG**ggcgtacacattagataaacgtatgaagtacgttagactcggcgccgccg
2 acccctatttttgagcagatttagtgacctggaaaaaaaatttgagtacaaaacttttccgaata**cAAtAAAAcGGcGGG**a
3 tgagtatccctgggatgactt**AAAAtAAtGGaGtGG**tgctctcccgattttgaatatgtaggatcattcgccagggtccga
4 gctgagaattggatg**AAAAAAAAGGGattG**tccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggaga
5 tccctttgcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaat**AtAAtAAAGGaaGGG**cttatag
6 gtcaatcatgttcttgtgaatggattt**AAcAAtAAGGGctGG**gaccgcttggcgcacccaaattcagtgtgggcgagcgcaa
7 cggttttggcccttgttagaggcccccgt**AtAAAcAAGGaGGGc**caattatgagagagctaatctatcgcgtgcgtgttcat
8 aacttgagtt**AAAAAAtAGGGaGcc**ctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgta
9 ttggcccattggctaaaagcccaacttgacaaatggaagatagaatccttgcat**ActAAAAAGGaGcGG**accgaaagggaag
10 ctggtgagcaacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagctt**ActAAAAAGGaGcGG**a

➡ MEDIANZEICHENKETTE() für *k*= 13 laufen lassen, und “hoffen”, dass es uns einen Teil (d.h. Subsequenz) der Lösung gibt

4^{13} = 67.108.864 (~ 10^7)

➡

>½ Tag

➡

gefundenes Motiv:
AAAAAt**AG**a**GGGG**

Greedy Algorithmen

Brute Force
Algorithmus :=
enumeriert exhaustiv alle möglichen Ergebnisse und wählt davon dann die beste Lösung.

VS.

***Greedy* Algorithmus** :=
iteratives Vorgehen, bei dem in jedem Durchlauf die “attraktivste” Möglichkeit gewählt wird.

exhaustive Suchraum Analyse:

- findet optimale Lösung(en), per Definition
- aber die Berechnung kann sehr lange dauern

inexakte Heuristik:

- kann desaströses Ergebnis produzieren
- aber für viele (biologische) Probleme nützlich

Mit Profilen Würfeln

Motive

```
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
```

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
```

Pr(*Sequenz* | PROFIL(*Motive*)) := Wahrscheinlichkeit, dass PROFIL das *k*-mer *Sequenz* generiert.

$$= \prod_{i=1}^{k} \text{PROFIL}(Motive)_{Sequenz(i),i}$$

Das “Profil-wahrscheinlichste” *k*-mer

Pr(*Sequenz* | PROFIL(*Motive*)) erlaubt Sequenzen nach PROFIL zu bewerten,
→ zu vergleichen.

Profil-wahrscheinlichstes *k*-mer Problem:
Finde das PROFIL-wahrscheinlichste k-mer einer Zeichenkette.

Eingabe: Eine Zeichenkette *Text*, eine ganze Zahl *k*, eine 4 x *k* Matrix PROFIL.

Ausgabe: Ein *k*-mer aus Text, das über alle *k*-mere in *Text* am Wahrscheinlichsten aus PROFIL generiert wird.

wie kann dieses Optimierungskriterium als
Zielfunktion einer *greedy* Suche verwendet werden?

Motiv-Suche "*Greedy*"

Algorithm: MotivSucheGreedy(Dna,k,t)

```
BesteMotive ← Matrix von Motiven aus den ersten k-meren der
 Zeichenketten in Dna
for jedes k-mer Motiv in der ersten Zeichenkette von Dna do
    Motiv_1 ← Motiv
    for i ← 2 bis t do
        bilde Profile aus den Motiven Motiv_1, ..., Motiv_{i-1}
        Motiv_i ← Profil-wahrscheinlichstes k-mer im i-ten Zeichenkette
         von Dna
    Motive ← (Motiv_1, ..., Motiv_t)
    if Score(Motive) < Score(BesteMotive) then
        BesteMotive ← Motive
return BesteMotive
```

Das Problem von MotivSucheGreedy

Minimalbsp: finde das (4,1)-Motiv **ACGT** implantiert in folgenden Strings *Dna*:

Dna=

```
ttACCTtaac
gATGTctgtc
acgGCGTtag
ccctaACGAg
cgtcagAGGT
```

für $Motiv_1$ = ACCT (das implantierte Motiv) ergibt sich Pr(**ATGT**) = 0

ZAEHLEN(*Motive*)=

```
A: 1 0 0 0
C: 0 1 1 0
G: 0 0 0 0
T: 0 0 0 1
```

Matrix ist *sparse*, alle Wahrscheinlichkeiten von **abweichenden Motiven** sind gleich und **gleich 0** !

Historische Zitate zu Wahrscheinlichkeit 0 und 1

Bsp: Wahrscheinlichkeit, dass die Sonne morgen nicht aufgeht.
5000 Jahre lang wurde überliefert, dass die Sonne aufgeht,
p= 0 dass sie morgen nicht aufgeht

Cromwell's Regel

*Abgesehen von logischen Zuständen, die **wahr** oder falsch sein können, sollten bei empirischen Abschätzungen **0** und **1** nicht als Wahrscheinlichkeiten verwendet werden.*

Oliver Cromwell (vor Angriff auf die Schotten) 1650

Laplace's Nachfolger Regel (Pseudocounts):

geht davon aus, dass bei der nächsten Beobachtung das bisher uneingetretene Ereignis vorkommt.

(p= 1/ 1826251, dass die Sonne morgen nicht aufgeht)

Pierre-Simon Laplace
1749-1827
French Physicist and mathematician

"I had no need of that hypothesis." A famous answer to the question from Napoleon about why he didn't mention the name of God in his work

MotivSucheGreedy mit Laplace's Nachfolger Regel

Dna=

```
ttACCTtaac
gATGTctgtc
acgGCGTtag
ccctaACGAg
cgtcagAGGT
```

Bsp: implantiertes (4,1)-Motiv **ACGT**

erste Sequenz `tt`**ACCT**`taac`

Motive= **ACCT**

ZAEHLEN(*Motive*)=

```
A: 1+1 0+1 0+1 0+1
C: 0+1 1+1 1+1 0+1
G: 0+1 0+1 0+1 0+1
T: 0+1 0+1 0+1 1+1
```

Motiv-Matrix mit **ACCT** aus der ersten Sequenz:

jedes Mögliche Ereignis wird einmal mehr als beobachtet gezählt (*Pseudocounts*)

PROFIL(*Motive*)=

```
A: 2/5 1/5 1/5 1/5
C: 1/5 2/5 2/5 1/5
G: 1/5 1/5 1/5 1/5
T: 1/5 1/5 1/5 2/5
```

Summe der Beobachtungen vergrößert sich entsprechend auch

Motive= **ACCT**

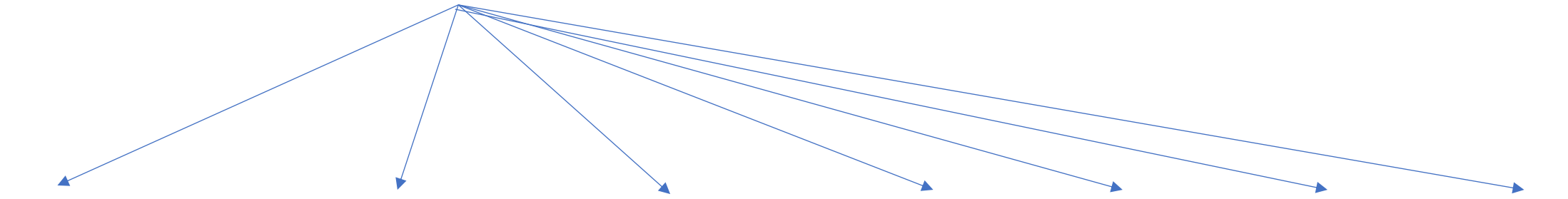
ZAEHLEN(*Motive*)=

```
A: 1+1 0+1 0+1 0+1
C: 0+1 1+1 1+1 0+1
G: 0+1 0+1 0+1 0+1
T: 0+1 0+1 0+1 1+1
```

PROFIL(**ACCT**)=

```
A: 2/5 1/5 1/5 1/5
C: 1/5 2/5 2/5 1/5
G: 1/5 1/5 1/5 1/5
T: 1/5 1/5 1/5 2/5
```

zweite Sequenz gATGTctgtc



gATG ACCT	ATGT ACCT	TGTc ACCT	GTct ACCT	Tctg ACCT	ctgt ACCT	tgtc ACCT
$1/5^4$	**$4/5^4$**	$1/5^4$	**$4/5^4$**	$2/5^4$	$2/5^4$	$1/5^4$

Zwei profil-wahrscheinlichste Motive (!)

Motive= **ACCT**
ATGT

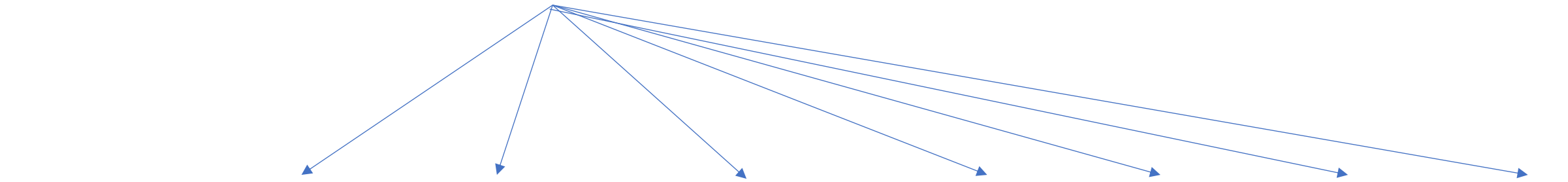
ZAEHLEN(*Motive*)=

A:	**2**+1	0+1	0+1	0+1
C:	0+1	**1**+1	**1**+1	0+1
G:	0+1	0+1	**1**+1	0+1
T:	0+1	**1**+1	0+1	**2**+1

PROFIL(**ACCT**)=

A:	3/6	1/6	1/6	1/6
C:	1/6	2/6	2/6	1/6
G:	1/6	1/6	2/6	1/6
T:	1/6	2/6	1/6	3/6

dritte Sequenz acg**GCGT**tag



Es kann nur einen geben...

Motive= **ACCT**
ATGT
acg**G**

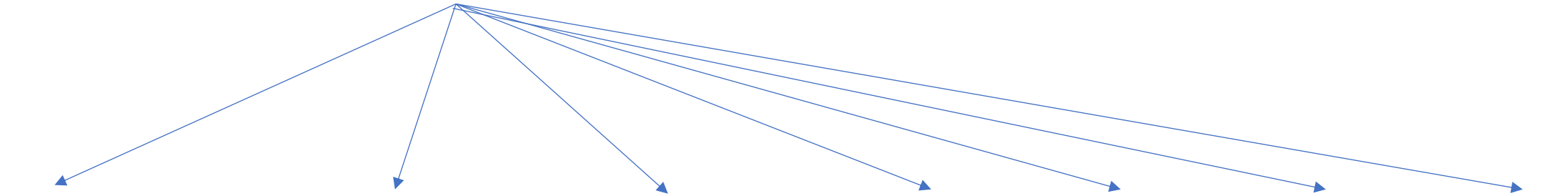
ZAEHLEN(*Motive*)=

```
A: 3+1 0+1 0+1 0+1
C: 0+1 2+1 1+1 0+1
G: 0+1 0+1 2+1 1+1
T: 0+1 1+1 0+1 2+1
```

PROFIL(**ACCT**)=

```
A: 4/7 1/7 1/7 1/7
C: 1/7 3/7 2/7 1/7
G: 1/7 1/7 3/7 2/7
T: 1/7 2/7 1/7 3/7
```

vierte Sequenz cccta**ACGA**g



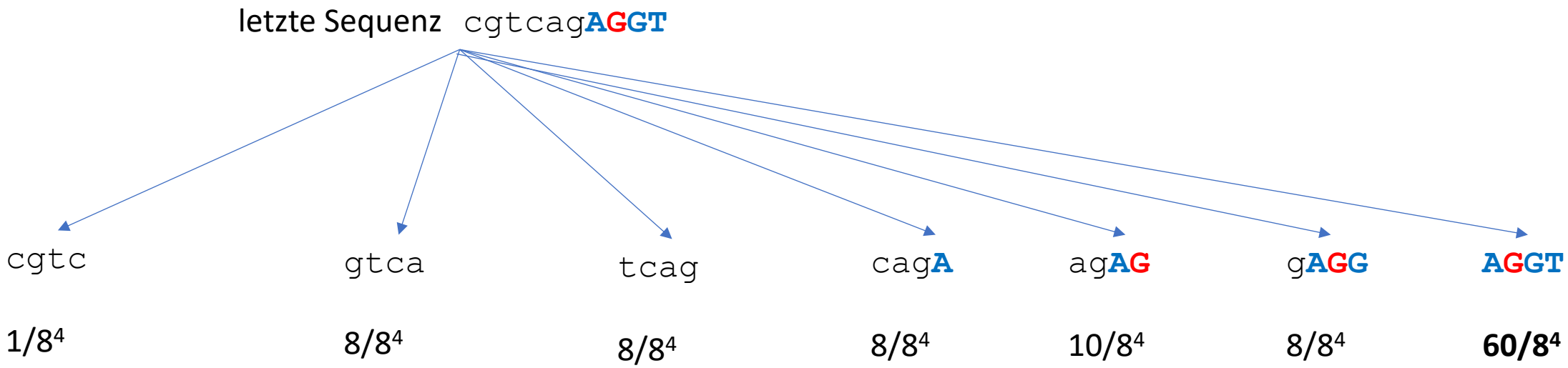
$18/7^4$ $3/7^4$ $2/7^4$ $1/7^4$ $16/7^4$ **$36/7^4$** $2/7^4$

Korrektes Motiv

Motive= **ACCT**
ATGT
acg**G**
ACGA

ZAEHLEN(*Motive*)= A: **4**+1 0+1 0+1 **1**+1
C: 0+1 **3**+1 **1**+1 0+1
G: 0+1 0+1 **3**+1 **1**+1
T: 0+1 **1**+1 0+1 **2**+1

PROFIL(**ACCT**)= A: 5/8 1/8 1/8 2/8
C: 1/8 4/8 2/8 1/8
G: 1/8 1/8 4/8 2/8
T: 1/8 2/8 1/8 3/8



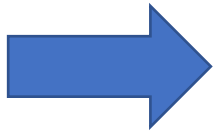
Korrektes Motiv

Greedy Motiv-Suche

implantiertes (4,1)-Motiv **ACGT**

Dna=

ttACCTtaac
gATGTctgtc
acgGCGTtag
ccctaACGAg
cgtcagAGGT



Motive=

ACCT
ATGT
acgG
ACGA
AGGT

Consensus= **ACGT**

Zufalls-basierte Algorithmen

Randomisierte Algorithmen
benutzen zufällige Elemente, um Lösungen zu finden

Monte Carlo Algorithmen (hier)

- können sub-optimale Lösungen finden
  (normalerweise mit einer kleine Fehlerwahrscheinlichkeit)
- sind sehr schnell, können mehrfach laufen

Las Vegas Algorithmen:

- Garantie der besten/exakten Lösung
- Laufzeit oft nicht abschätzbar

Die Profil-wahrscheinlichsten Motive von Sequenzen

PROFIL(*Motive*) := (4 x k) Matrix berechnet über *Motive*.

Profil := eine beliebige (4 x k) Matrix mit Wahrscheinlichkeitsverteilungen.

Pr(*k-mer* | *Profil*) := Wahrscheinlichkeit, dass ein *k*-mer von einem Profil generiert wird

- Profil-wahrscheinlichtses Motiv in einem *Text*:

MOTIV(*Profil, Text*) := $\underset{\text{Sequenz in Text}}{\text{argmin}}$ *Pr*(*Sequenz* | *Profil*)

- Profil-wahrscheinlichtse Motive aus jeder Sequenz einer Menge *Dna*:

MOTIVE(*Profil, Dna*) := $\bigcup_{i=1}^{t}$ MOTIV(*Profil*, Dna_i)

Beispiel, gegeben:

Dna= ttaccttaac
gatgtctgtc
acggcgttag
ccctaacgag
cgtcagaggt

PROFIL= A: 4/5 0 0 1/5
C: 0 3/5 1/5 0
G: 1/5 1/5 4/5 0
T: 0 1/5 0 4/5

Profil-wahrscheinlichte 4-mere:

MOTIVE*(Profil, Dna)* = tt**acct**taac
g**atgt**ctgtc
acg**gcgt**tag
ccct**acga**g
cgtcag**aggt**

PROFIL(*Motive*)= A: 4/5 0 0 1/5
C: 0 3/5 1/5 0
G: 1/5 1/5 4/5 0
T: 0 1/5 0 4/5

usw.

PROFIL(MOTIVE(*Profil, Dna*)) → MOTIVE(PROFIL*(*MOTIVE*(Profil, Dna*))) →

PROFIL(MOTIVE(PROFIL*(*MOTIVE*(Profil, Dna*))))) → ...

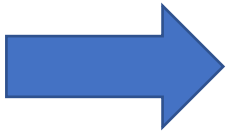
Randomisierte Motiv-Suche

Algorithm: RandomisierteMotivSuche(Dna, k,t)

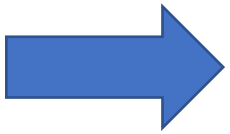
wähle zufällige k-mere $Motive = (Motiv_1, \ldots, Motiv_t)$ von je einer Sequenz in Dna
$BesteMotive \leftarrow Motive$
while für immer **do**
    $Profile \leftarrow$ Profile($Motive$)
    $Motive \leftarrow$ Motive($Profile$, Dna)
    if Score($Motive$) < Score($BesteMotive$) **then**
        $BesteMotive \leftarrow Motive$
    'else
        return $BesteMotive$

Beispiel, ein implantiertes (4,1)-Motiv **ACGT**:

Dna=
ttACCT**taac**
gAT**GTct**gtc
ccgGCGTtag
c**acta**ACGAg
cgtcag**AGGT**

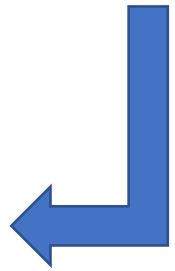


Motive=
taac
GTct
ccgG
acta
AGGT



PROFIL(*Motive*)=

```
A: 0.4 0.2 0.2 0.2
C: 0.2 0.4 0.2 0.2
G: 0.2 0.2 0.4 0.2
T: 0.2 0.2 0.2 0.4
```



Wahrscheinlichkeiten *Pr*(*kmer*, PROFIL(*Motive*)):

ttAC	tACC	ACCT	CCT**t**	CT**ta**	T**taa**	**taac**
.0016	.0016	**.0128**	.0064	.0016	.0016	.0016
gAT**G**	AT**GT**	T**GTc**	**GTct**	**Tctg**	**ctgt**	**tgtc**
.0016	**.0128**	.0016	.0032	.0032	.0032	.0016
ccgG	**cgG**C	**gG**CG	**G**CGT	CGTt	GTta	Ttag
.0064	.0032	.0016	**.0128**	.0032	.0016	.0016
cact	**acta**	**cta**A	**ta**AC	**a**ACG	ACGA	CGAg
.0032	.0064	.0016	.0016	.0032	**.0128**	.0016
cgtc	gtca	tcag	cag**A**	ag**AG**	g**AGG**	**AGGT**
.0016	.0016	.0016	.0032	.0032	.0032	**.0128**

MOTIVE(PROFIL(*Motive*))
ACCT
ATGT
GCGT
ACGA
AGGT

Übungen 06

1. Implementieren Sie MEDIANZEICHENKETTE und vergleichen Sie Ihre Lösung mit dem im Seminar besprochenen, "reduzierten" Subtilen Motiv Problem für *k*= 13.

Eingabe: *Dna*= Algo04_subtiles_motiv.txt, *k*= 13
erwartete Ausgabe: eine Matrix *Motive* mit SCORE(*Motive*)=29 und CONSENSUS(*Motive*)= **AAAAA**t**AG**a**GGGG**.

2. Implementieren Sie GREEDYMOTIVSUCHE ohne sowie mit Pseudocounts, und bewerten Sie die Lösung, die jeder dieser beiden Algorithmen für das "Subtile Motif Problem" findet, indem Sie entsprechend SCORE(Motive) berechnen.

Eingabe: *Dna*= Algo04_subtiles_motiv.txt, *k*= 15
erwartete Ausgabe: das Motive mit dem Consensus des implantierten (15,4)-Motiv AAAAAAAAGGGGGGG, bzw. Motive deren SCORE(Motive) diesem implantierten Muster gleichkommt

3. Implementieren Sie ebenfalls den Algorithmus RANDOMISIERTEMOTIVSUCHE und wenden Sie diesen dann auf das Subtile Motiv Problem an. Wie groß ist das beste, und wie groß der Median von SCORE(Moitve) nach 20, 200 bzw. nach 2000 Durchläufen?

Eingabe/erwartete Ausgabe: s. Aufgabe 2