

Fragestellung

ggaattccaaaaaaaaaatacgaactacacctgctccggagcccgcggcggtacctgcagcggaggagctctgtcttcccccttcatctcacgcgagcccggcgtcccgc
ccgtgcgccccggcgcagcccgccagtccgcccggagcccgccagtcgcccgcgctgcacgcccggggtgaaccctctgccctcgtgggacagagggccccgcag
ccgtcatgctttccgccatctacacagtctggcgggactgctgttctgcccgtcctgggtaacctctgctgccatacttctccaggacataggctacttctgaaggtggcc
gccgtgggcccggagggtgcgcagctacgggcagcggcggccggcgcaccatcctgcgggcgttctggagaaagcgcgccagacgccacacaagcctttctgct
cttccgcgacgagactctcacctacgcgcaggtggaccgg ...

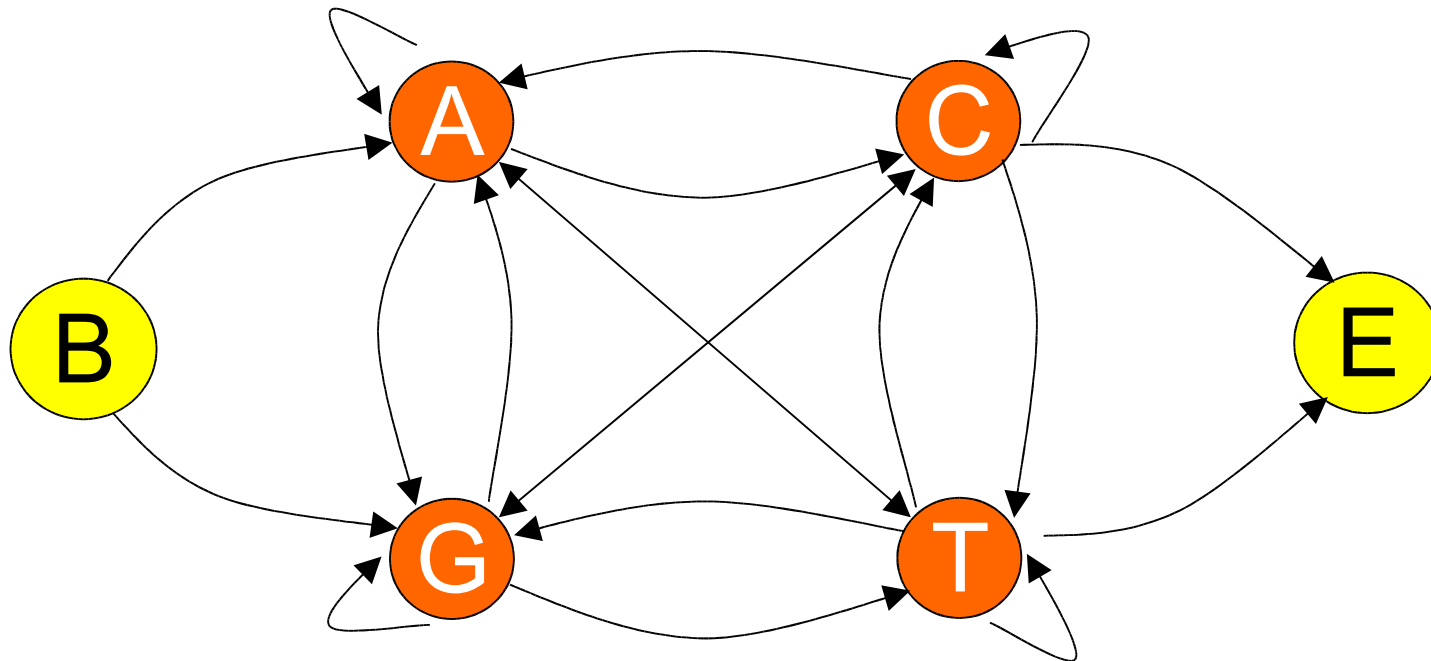
ataggagaaggctctgttggcagaagacctaaagacgggaggaggcgctggcgtcagcgggtcgttctcagccgtgaccacgagcggccagcaggtgtcagtgcg
tctcgtttgccctggcgtcggctcctggaccctccccacctcggctaaagggggcgcggtccagacccccgggtcttgcgtttcccgcagacagcatcgcggctggga
ggacacgtagccgcttctcc ctaagcggagggtctgaactccggcttgtaccacttccccgtgggtgctgtccattccgcc ccaggggttc

Frage 1: Stammt dieses Stück DNA aus einer CpG Insel?

Frage 2: Oder ist in dieser Sequenz eine CpG Insel versteckt?

Modell: DNA als Markov-Kette erster Ordnung

Dass eine bestimmte Base an der Position X_i vorkommt, hängt nur von der vorherigen Base an Position X_{i-1} ab (**Markov-Eigenschaft**).



Übergangswahrscheinlichkeit:

$$\{x\} = \{AGT\mathbf{CG}TATCGT\dots\}$$

$$\alpha_{x_{i-1} x_i} = P(x_i = t \mid x_{i-1} = s)$$

$$\alpha_{\mathbf{CG}} = P(x_i = G \mid x_{i-1} = C)$$

Übergangswahrscheinlichkeiten:

1. Man nehme:
- viele Sequenzen
4. Zähle die Übergänge zwischen den jeweiligen Basenpaaren.
Bsp.:
AAGTCTGACGTGCGCTAGATTA`GGCCA
3. Berechne die **Übergangswahrscheinlichkeiten**

$$\alpha_{kl} = \frac{\text{Anzahl der Übergänge von k nach l}}{\text{Summe aller k Übergänge}}$$

Anzahl der Übergänge:

	A	C	G	T
A				
C				
G				
T				

Übergangswahrscheinlichkeiten:

	A	C	G	T	Σ
A	0,16	0,16	0,5	0,16	1
C	0,16	0,16	0,33	0,33	1
G	0,25	3,375	0,125	0,25	1
T	0,33	0,16	0,33	0,16	1

Wahrscheinlichkeit für bestimmte Sequenz:

Gegebene Sequenz: $\{x\} = \{AGTCGTATCGT\dots\}$

Wahrscheinlichkeit, dass genau diese Sequenz erzeugt wird, kann als Produkt der einzelnen Übergangswahrscheinlichkeiten dargestellt werden:

$$P(x) = P(x_i | x_{i-1}) * P(x_{i-1} | x_{i-2}) * \dots * P(x_2 | x_1) * P(x_1)$$

Allgemeine Form:
$$P(x) = P(x_1) \prod_{i=2}^L a_{x_{i-1}x_i}$$

Die Summe der Wahrscheinlichkeiten aller möglichen Sequenzen ergibt 1:

$$\sum P(x) = 1$$

CpG+ / CpG- Region ?

- Übergangswahrscheinlichkeiten getrennt berechnet für CpG + / CpG - Regionen:
(Grundlage: ca. 60 000 Nucleotides aus Human DNA-Sequenzen,
davon 48 CpG Inseln,
daraus Übergangswahrscheinlichkeiten für Inseln und
nicht Inseln berechnet (Durbin et al., 1998))

CpG+	A	C	G	T
A	0.180	0.274	0.426	0.120
C	0.171	0.368	0.274	0.188
G	0.161	0.339	0.375	0.125
T	0.079	0.355	0.384	0.182

CpG-	A	C	G	T
A	0.300	0.205	0.285	0.210
C	0.322	0.298	0.078	0.302
G	0.248	0.246	0.298	0.208
T	0.177	0.239	0.292	0.292

Unterscheidung mit Markov-Kette

- Berechnung des log-odds Scores. $S(x)$

$$S(x) = \log_2 \frac{P(x | \text{model } +)}{P(x | \text{model } -)}$$

$$i \sum_{i=1}^L \log_2 \frac{a_{x_{i-1}x_i}^+}{a_{x_{i-1}x_i}^-}$$

$$= \sum_{i=1}^L \log_2 \beta_{x_{i-1}x_i}$$

β	A	C	G	T
A	-0.740	0.419	0.580	-0.803
C	-0.913	0.302	1.812	-0.685
G	-0.624	0.461	0.331	-0.730
T	-1.169	0.573	0.393	-0.679

$$S(x) = \log_2 \frac{P(x | \text{model } +)}{P(x | \text{model } -)} = \sum_{i=1}^L \log_2 \frac{a_{x_{i-1}x_i}^+}{a_{x_{i-1}x_i}^-} = -1,49$$

Frage: Ist Sequenz eine CpG-Insel?

- Vergleiche log-odd-Scores

$$S(x) = \log_2 \frac{P(x|\text{model}+)}{P(x|\text{?}-\text{Region})} = \sum_{i=1}^L \log_2 \frac{a_{x_{i-1}x_i}^+}{a_{x_{i-1}x_i}^{\text{?}-\text{Region}}}$$

- Wenn gleiche Region (also CpG+), dann:
 - $S(x) \sim 0$
- Wenn verschiedene Regionen (also aus CpG-), dann:
 - $S(x) < 0$.

Frage: Ist in der Sequenz eine CpG-Insel?

1. „Fensteralgorithmus“
2. Hidden Markov Modell

„Fensteralgorithmus“

1. 100 Nucleotide 100 Nucleotide
 links von x rechts von x

ATTATGT...AGGCTC **G** TACGTGC...TAGCGCTAGTCCGATAGCTGATCGTC

2. Berechne
Übergangswahr-
scheinlichkeits-
matrix für **linke**
Seite Berechne
Übergangswahr-
scheinlichkeits-
matrix für **rechte**
Seite

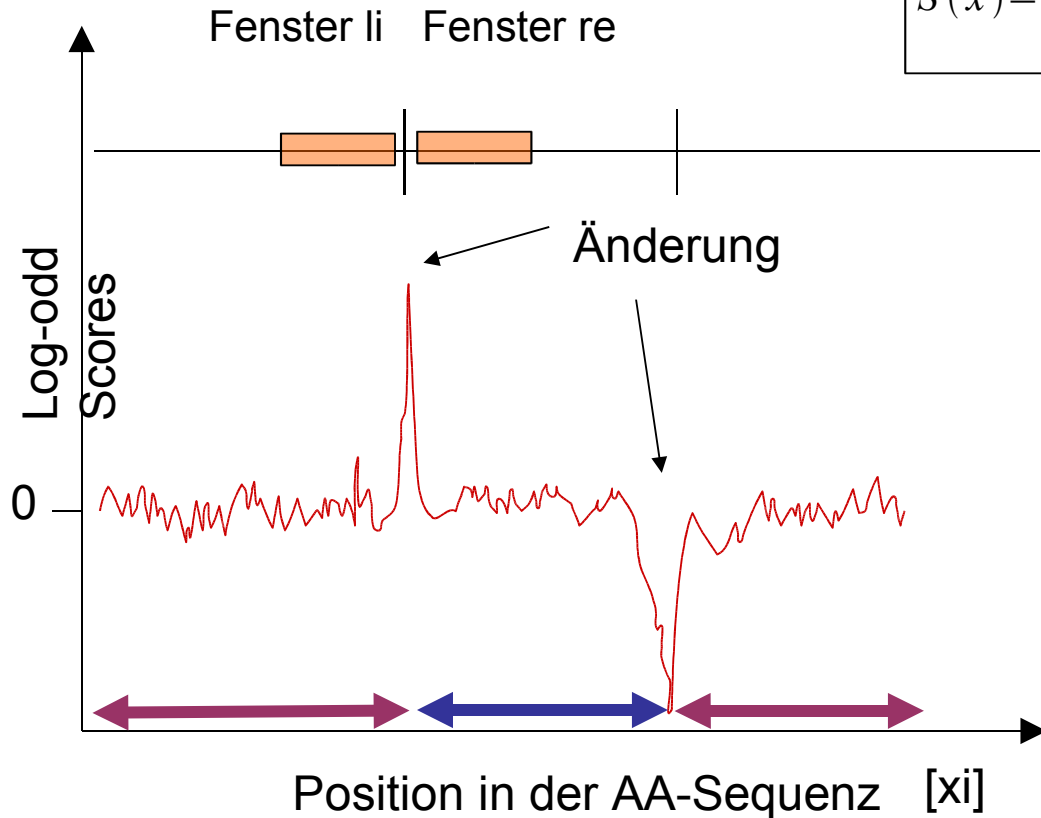
↑
x

3.
$$S(x) = \log_2 \frac{P(x|links)}{P(x|rechts)} = \sum_{i=1}^L \log_2 \frac{a_{x_{i-1}x_i}^{links}}{a_{x_{i-1}x_i}^{rechts}}$$

1. Wiederhole Schritt 1-3 für x+1 bis (Sequenzlänge – Fenstergröße)
2. Drucke S(x) Werte für jede Position xi
3. Graphische Darstellung liefert Übergänge

Ergebnis „Fensteralgorithmus“

$$S(x) = \log_2 \frac{P(x|links)}{P(x|rechts)} = \sum_{i=1}^L \log_2 \frac{a_{x_{i-1}x_i}^{links}}{a_{x_{i-1}x_i}^{rechts}}$$



Ergebnis:

Wenn beide Fenster

aus gleicher Region:

$P(x|links)/P(x|rechts)=1$

$\log 1 = 0$

Wenn beide Fenster aus

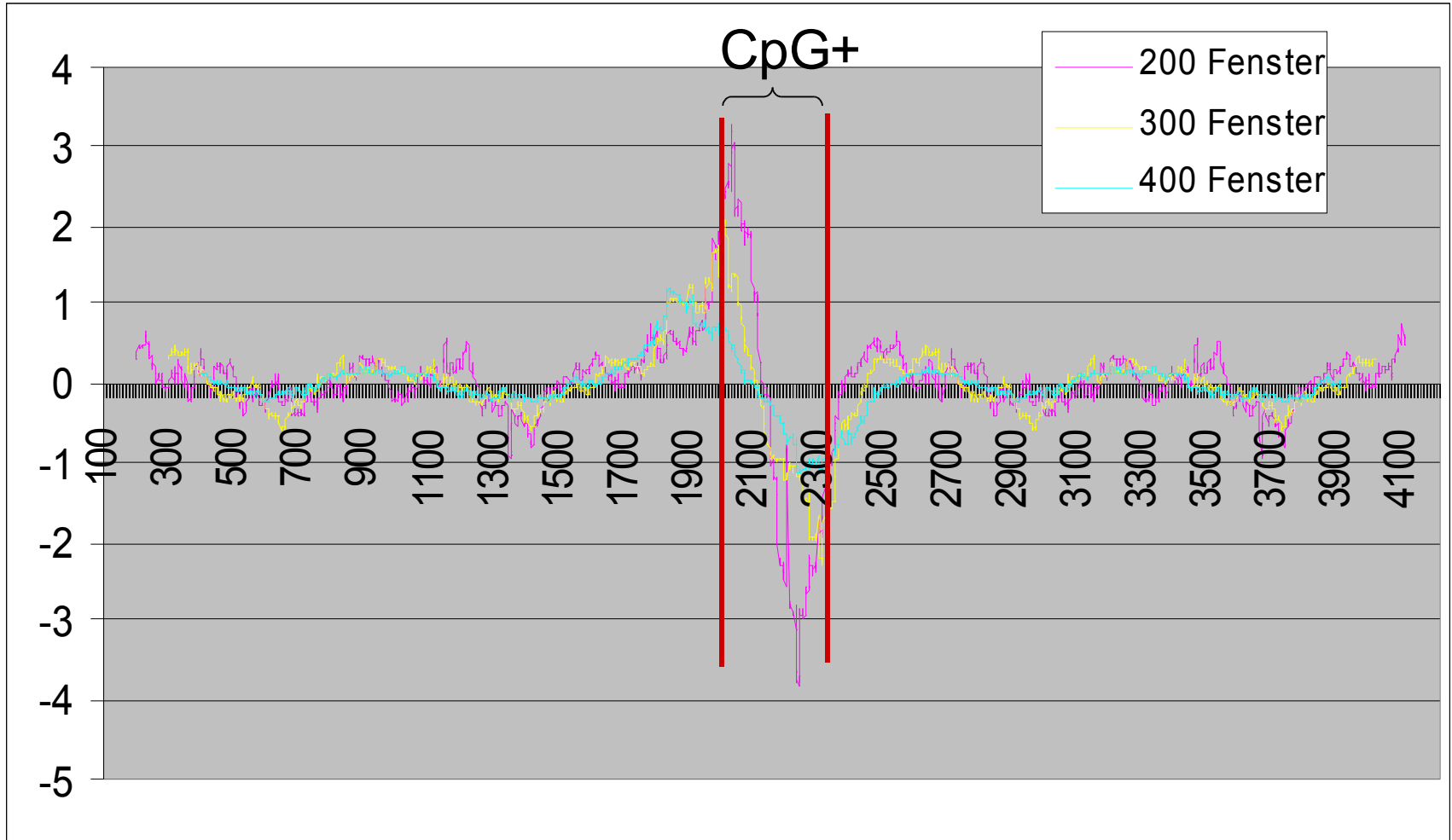
verschiedenen Regionen:

$P(x|links)/P(x|rechts) \neq 1$

$\log (\neq 1) \neq 0$

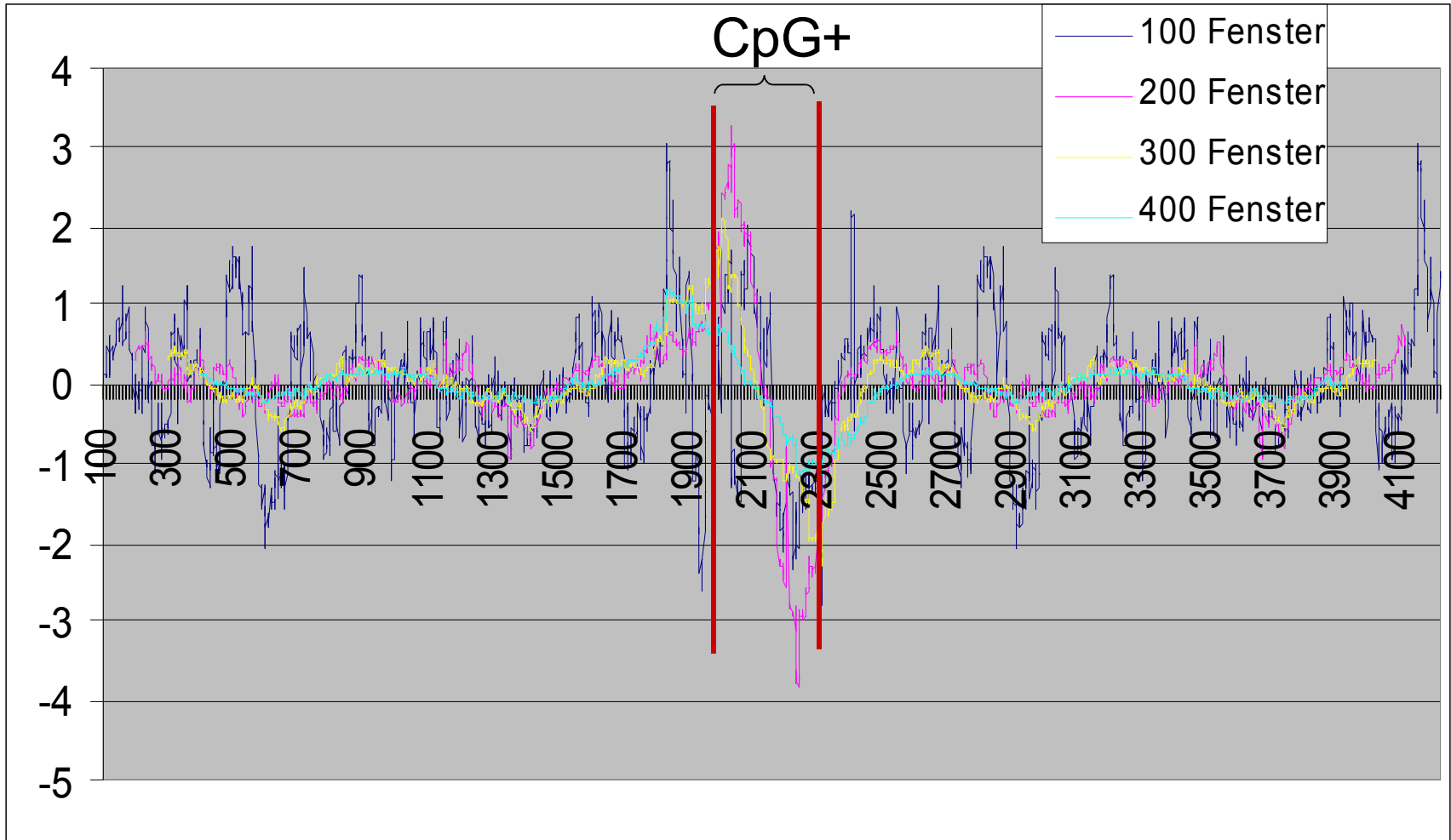
Ergebnis „Fensteralgorithmus“

Berechnet aus selbst erstellten Sequenzen



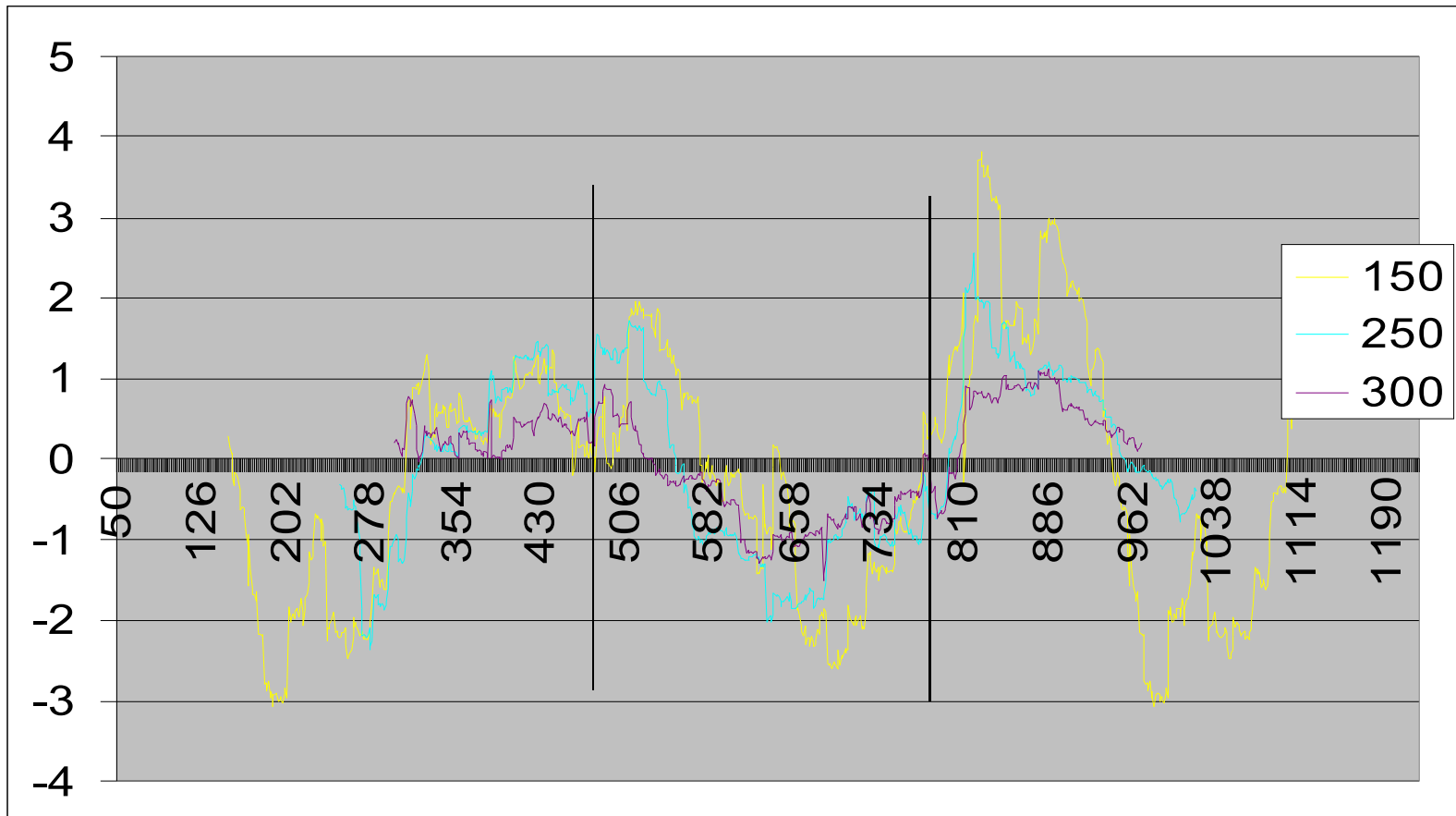
Ergebnis „Fensteralgorithmus“

Berechnet aus selbst erstellten Sequenzen



Ergebnis „Fensteralgorithmus“

Berechnet aus realen Sequenzen



Ergebnis des “Fensteralgorithmus” nicht immer eindeutig,
deshalb:

CpG-Insel finden mit Hilfe von
Hidden Markov Modell (HMM)

Beschreibung:

2. Ein Hidden-Markov Modell ist ein Markov Modell
 - bei dem nur die Sequenz der Ausgabe-Symbole beobachtbar ist,
 - bei dem die Sequenz der Zustände verborgen bleibt
5. Es kann mehrere Zustandssequenzen geben, die dasselbe Ausgabe-Symbol erzeugen

Beispiel „Dozent“

Beispiel: Gemütszustand des Dozenten bedingt Umfang der Übungsaufgaben

Situation:

- Dozent vergibt 3 Arten von Ü-Aufgaben:
 - A: Arbeitsumfang 5 Minuten
 - B: Arbeitsumfang 1 Stunde
 - C: Arbeitsumfang 3 Stunden
- Dozent zeigt Euch nicht täglich seine Gemütsverfassung, aber Ihr hab inzwischen herausgefunden, dass er entweder in guter, neutraler, oder schlechter Stimmung sein kann (hält den ganzen Tag über an und wechselt über Nacht)

Frage:

- In welcher Gemütsverfassung ist der Dozent wenn er die Ü-Aufgaben vergibt?

Beispiel „Dozent“

Situation:

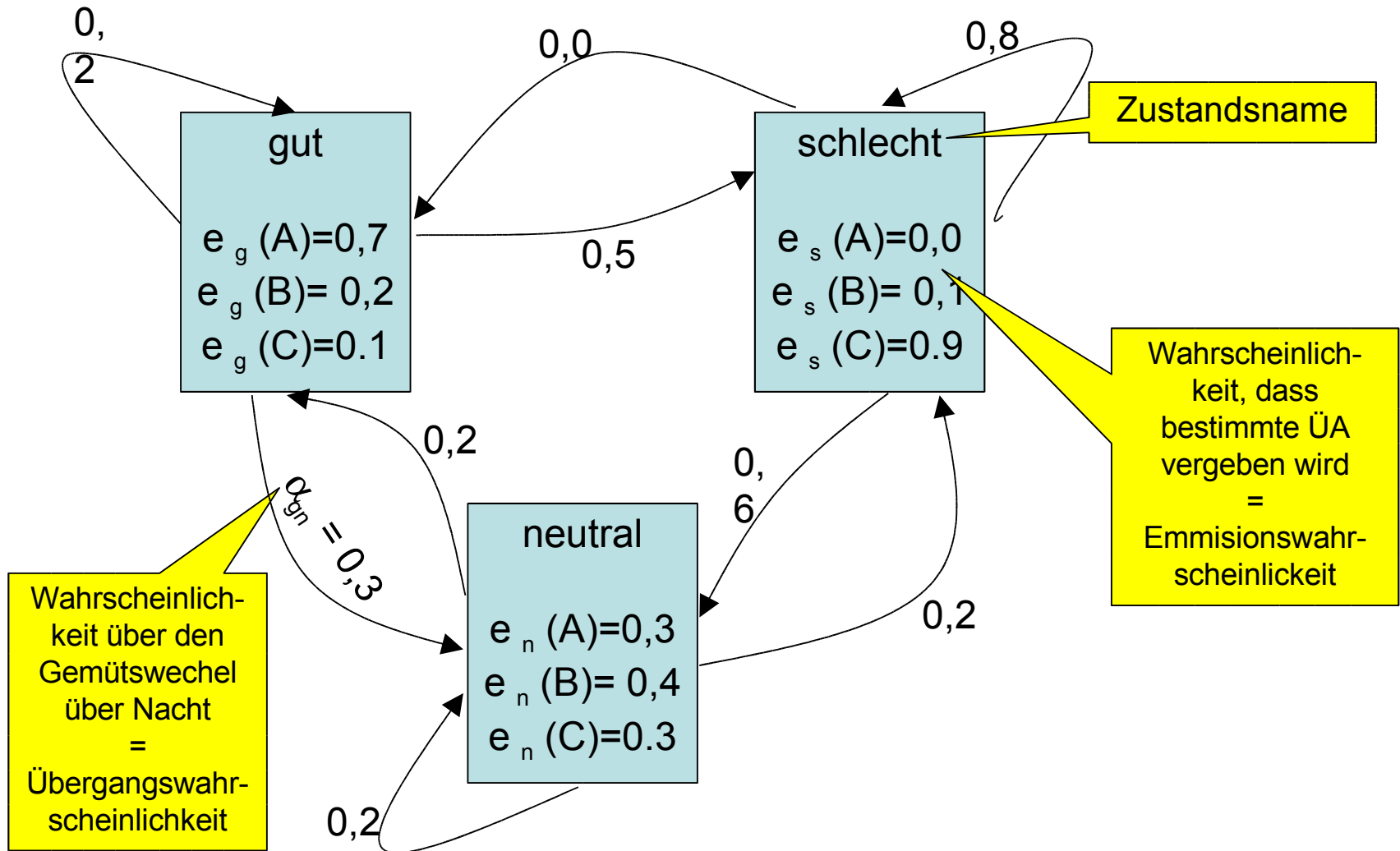
- In der letzten Woche gab es folgende ÜAs:

Montag:	A
Dienstag:	C
Mittwoch:	B
Donnerstag:	A
Freitag:	C

Frage:

- Welches ist die wahrscheinlichste Gemütsverfassungskurve diese Woche? (Decodierung Viterbi-Algorithmus)
- Mit welcher Wahrscheinlichkeit wurde diese Abfolge von ÜA gestellt. (Vorwärts-Algorithmus)

Hidden-Markov-Modell „Dozent“



Elemente des Hidden-Markov-Modells

Zustände	$S = \{k_1, k_2, \dots, k_m\}$	Launen: {gut, neutral, schlecht}
Emitierte Symbole	$E = \{b_1, b_2, \dots, b_n\}$	Übungsaufgaben: {A,B,C}
Übergangswahrscheinlichkeits Matrix	$\alpha_{k-1 k} =$ Wahrscheinlichkeit, dass Zustand k auf Zustand k-1 folgt	$\alpha_{g s} = 0,5$
Emissionswahrscheinlichkeit	$e_k(b) =$ Wahrscheinlichkeit, dass Zustand k das Symbol b erzeugt; $b \in Q$	Für Zustand gut gelaunt: $e_g(A) = 0,7$ $e_g(B) = 0,2$ $e_g(C) = 0,1$

Beispiel „Dozent“

Situation:

In der letzten Woche gab es folgende ÜAs:

Montag: A

Dienstag: C

Mittwoch: B

Donnerstag: A

Freitag: C

Frage:

- Welches ist die wahrscheinlichste Gemütsverfassungskurve diese Woche? Antwort Viterbi-Algorithmus (Decodierung)

Decodierung: Viterbi Algorithmus:

Input:

- Hidden-Markov-Modell (alle Übergangswahrscheinlichkeiten und alle Emmissionswahrscheinlichkeiten)
- Symbolsequenz E (beobachtete, ausgegebene Symbole aus dem Hidden-Markov-Modell)

Output:

- Zustandssequenz S , welche den wahrscheinlichsten Pfad durch das HMM angibt (Wahrscheinlichster Pfad für eine bestimmte emmitierte Symbolsequenz)

Viterbi Algorithmus: Allgemein

Input:

- Hidden-Markov-Modell:
 $\alpha_{k,l}$ = Übergangswahrscheinlichkeit von Zustand k nach l
 $\varepsilon_k(b)$ = Wahrscheinlichkeit, daß Zustand k das Symbol b emittiert
- $E = x_1, \dots, x_i, \dots, X_n$: Sichtbare Symbolsequenz

Prinzip:

- Mit dynamischer Programmierung
- Matrix $v_k(i)$, wobei $v_k(i)$ = Wahrscheinlichkeit vom wahrscheinlichsten Pfad der Symbolsequenz x_1, \dots, x_i , wobei das x_i te Symbol (x_i) durch Zustand k emittiert wurde

$$v_k(i) = \varepsilon_i(x_i) * \max_l (v_l(i-1) * \alpha_{l,k}) \text{ für alle Zustände } k$$

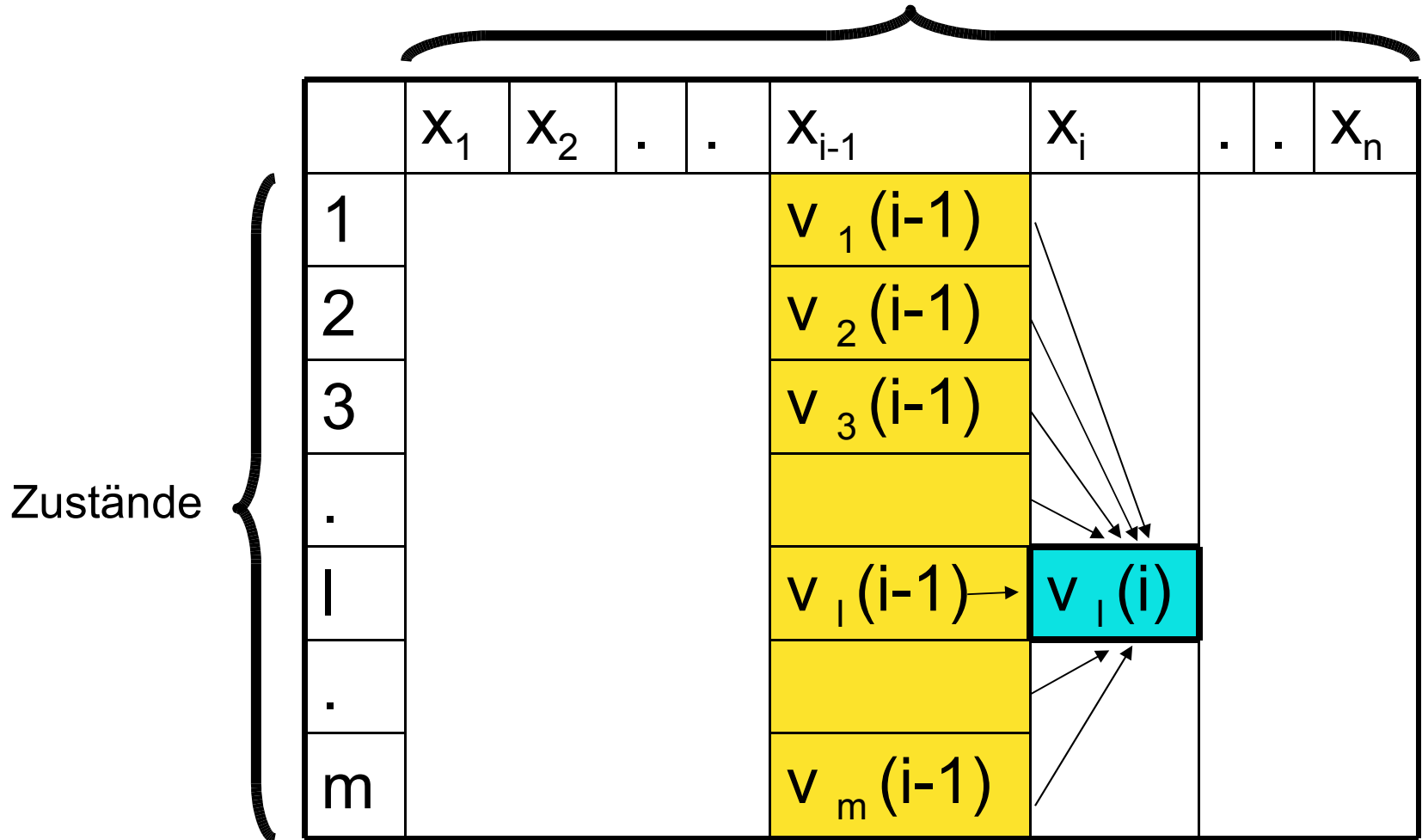
$$\text{Initialisierung: } v_k(1) = \varepsilon_k(x_1) / \text{Anzahl der Zustände}$$

Algorithmus:

- Zelle für Zelle Matrix $v_k(i)$ berechnen
- Pointer einführen (zeigt auf Max)
- Rekonstruiere Pfad entlang der Pointer „trace back“, beginne bei höchstem Wert in letzter Spalte

Viterbi Algorithmus




Symbol Sequenz



$$v_l(i) = \varepsilon_i(x_i) * \max_k (v_k(i-1) * \alpha_{kl}) \text{ für alle Zustände } k$$

Berechnung des Viterbi Algorithmus „Dozent“

1. Initialisierung: $(v_k(1) = \epsilon_k(x_1)) / \text{Anzahl der Zustände}$

	A	C	B	A	C
Gut 	$\epsilon_g(A) = 0,7$ $v_g(1) = 0,23$				
Neutral 	$\epsilon_n(A) = 0,3$ $v_n(1) = 0,1$				
Schlecht 	$\epsilon_s(A) = 0,0$ $v_s(1) = 0,0$				

emitierte
Symbole

nicht
sichtbare
Zustände




Berechnung des Viterbi Algorithmus: „Dozent“

1. $v_i(i) = \varepsilon_i(x_i) * \max_k (v_k(i-1) * \alpha_{ki})$ für alle Zustände k
2. Pointer zeigt auf Maximum

	A	C
Gut	$\varepsilon_g(A) = 0,7$ $v_g(1) = 0,23$	1 $\alpha_{gg} = 0.2 : 0,23 * 0,2 = 0,046 \leq \max_k (v_k(i-1) * \alpha_{ki})$ 2. $\alpha_{ng} = 0.3 : 0,1 * 0,3 = 0,03$ 3. $\alpha_{sg} = 0.5 : 0,0 * 0,5 = 0,0$ $\varepsilon_g(C) = 0,1$ $v_g(2) = 0,1 * 0,046 = 0,0046$
Neutral	$\varepsilon_n(A) = 0,3$ $v_n(1) = 0,1$	1 $\alpha_{gn} = 0.3 : 0,23 * 0,3 = 0,069 \leq \max_k (v_k(i-1) * \alpha_{ki})$ 2. $\alpha_{nn} = 0.2 : 0,1 * 0,2 = 0,02$ 3. $\alpha_{sn} = 0.2 : 0,0 * 0,2 = 0,0$ $\varepsilon_g(C) = 0,3$
Schlecht	$\varepsilon_s(A) = 0,0$ $v_s(1) = 0,0$	$v_g(2) = 0,3 * 0,069 = 0,0207$

Berechnung des Viterbi Alogarismus „Dozent“






Continue.... „trace Back“

	A	C	B	A	C
Gut 	0,23	0,0046	0,000828	0,0011592	0,0000231
Neutral 	0,1	0,0207	0,00828	0,0006624	0,0001043
Schlecht 	0,0	0,1035	0,00828	0	0,0005216

The table illustrates the Viterbi algorithm's trace back process. The columns represent states A, C, B, A, C. The rows represent sentiment classes: Gut (smiley), Neutral (neutral), and Schlecht (sad). The values in the cells represent the maximum probability for each state and class. Red text indicates the values that are part of the optimal path. Arrows show the backtracking from the final state (C) to the previous state (A), then to the previous state (C), and finally to the previous state (A).

Ergebnis Viterbi Algorithmus: Beispiel „Dozent“

- Welches ist die wahrscheinlichste Gemütsverfassungs-Kurve diese Woche?

Tage	Mo	Di	Mi	Do	Fr
ÜAs	A	C	B	A	C
Laune					

Andere wichtige Algorithmen in Verbindung mit HMM

- **Rückwärts – Algorithmus:**
Manchmal will man nur wissen, wie wahrscheinlich es ist, ob das Symbol x vom Zustand k emittiert wurde. Wie groß ist die bedingte Wahrscheinlichkeit?
- **Baum-Welch Algorithmus:**
Lernen der Parameter α_j und $e_k(b)$; dafür müssen viele Emissionssequenzen und die dazugehörigen Zustandsequenzen vorliegen.

Einsatzgebiete von HMM

- **Spracherkennung**
- **Mustererkennung**
- **Handschriftenerkennung**
- **Bioinformatik:**
 - **Paarweises oder multiples Alignment**
 - **Profil-HMM für Sequenzfamilien:** läßt sich eine gefundene Sequenz einer bekannten Proteinfamilie zuordnen?
 - **Genvohersagen:** Programme wie GLIMMER oder GENIE beruhen auf HMMs höherer Ordnung

Zusammenfassung

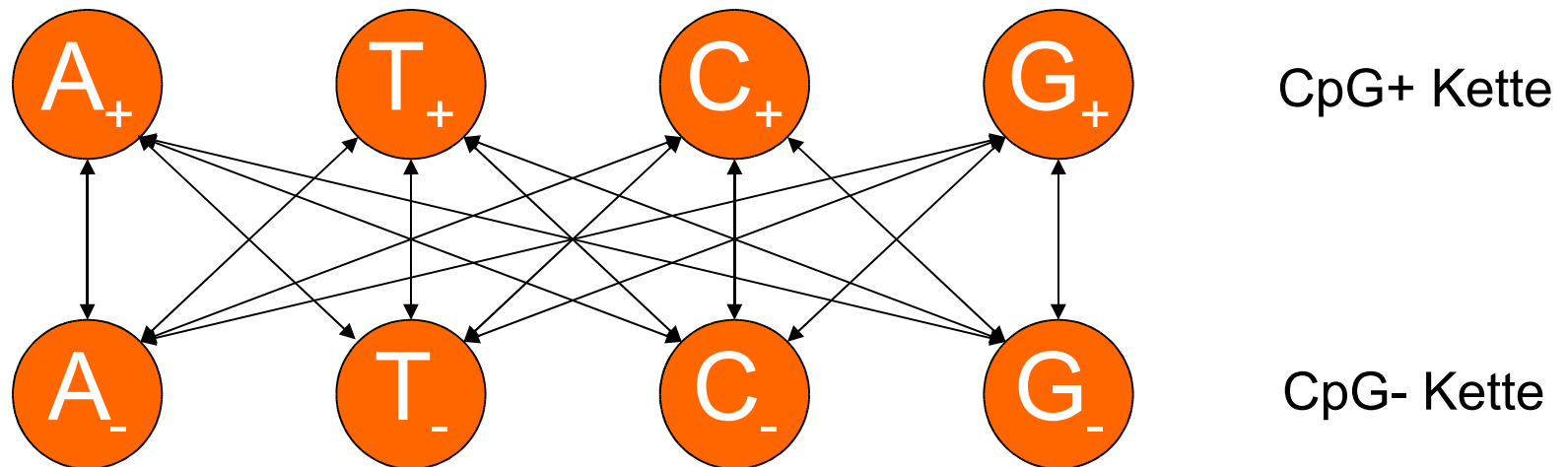
- Markov-Eigenschaft:
 - Gedächtnislos, d.h. jeder Zustand hängt nur von seinem Vorgängerzustand ab.
- HMM sind Markov-Ketten-Modelle,
 - deren Zustände nicht direkt beobachtet werden können, es können nur deren Signale erfasst und ausgewertet werden
 - Das jeweilige Signal kann von verschiedenen Zuständen erzeugt werden.
- Wichtige Algorithmen in diesem Zusammenhang:
 - Viterbi-Algorithmus
 - Vorwärts/Rückwärts-Algorithmus
 - Baum-Welch Algorithmus

Literatur

- Durbin, E., Krogh, M.: biological sequence analysis, Cambridge University Press 1998.
- Rabiner , L.R.: A Tutorial on Hidden Markov Models and selected Applications in Speech Recognition.
- Bioinformatik Skript von Volker Heun:
www.bio.informatik.uni-muenchen.de/personen/heun/lecturenotes
(liegt auf /home/public/HMM-Vortrag)
- Wahrscheinlichkeitsskript Schaum´s Outline Kapitel 7
- Hidden Markov Models: v. Nikolas Dörfler 21.11.2003 (liegt auf /home/public/HMM-Vortrag)

Was ist „Hidden“ im „CpG-Modell“

- Beide CpG-Ketten-Modelle vereinigen
- Einführen einer kleinen Übergangswahrscheinlichkeiten zwischen beiden Ketten
- Nun muss entschieden werden ob z.B. C aus dem C+ oder dem C- Modell stammt
- Wenn wir das Symbol C sehen, ist nicht klar of es aus + oder – Region stammt. Diese Information ist versteckt (**hidden**)



(Schema gezeichnet ohne Übergäng zwischen den AA innerhalb einer Kette)

Elemente des Hidden-Markov-Modells

Zustände	$S = \{ k1, k2, \dots, km \}$	$\{A+,C+,G+,T+,A-,C-,G-,T-\}$
Emitierte Symbole	$E = \{b1, b2, \dots, bn\}$	$\{A,C,G,T\}$
Übergangswahrscheinlichkeits Matrix	$\alpha_{k-1 k} =$ Wahrscheinlichkeit, dass Zustand k auf Zustand k-1 folgt	
Emissionswahrscheinlichkeit	$e_k(b) =$ Wahrscheinlichkeit, dass Zustand k das Symbol b erzeugt; $b \in Q$	Im Falle der CpG Inseln =1, A+ und A- emittieren nur A $e_{A+}(A) = 1,$ $e_{A-}(A) = 1,$

Viterbi Algorithmus: Notation

Running time:

- Es sei:
m: Anzahl der Zustände
n: Sequenzlänge
- Um einen Eintrag in der v Matrix zu berechnen, muss der Algorithmus m andere Einträge der Matrix v , m Übergangswahrscheinlichkeiten und 1 Emmissionswahrscheinlichkeit betrachten
=> $O(m)$ Schritte für jeden Eintrag von v
- Anzahl der Einträge in Matrix: $n*m$
=> running time: $O(nm^2)$