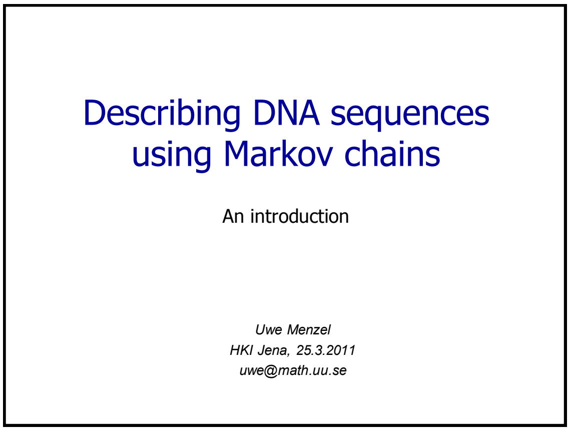
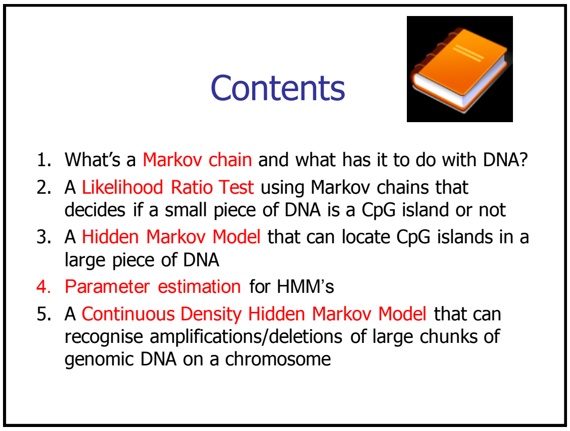


Describing DNA sequences using Markov chains

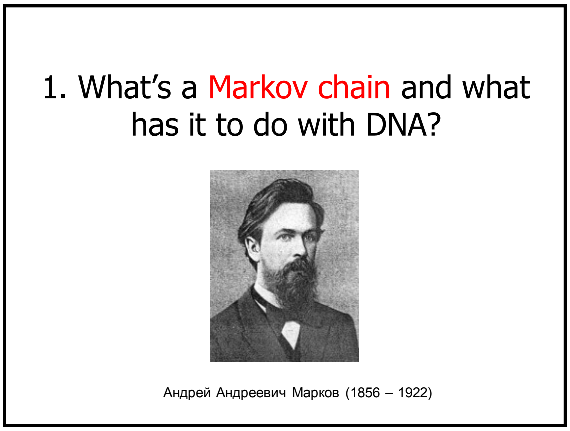
An introduction

Uwe Menzel
HKI Jena, 25.3.2011
uwe@math.uu.se



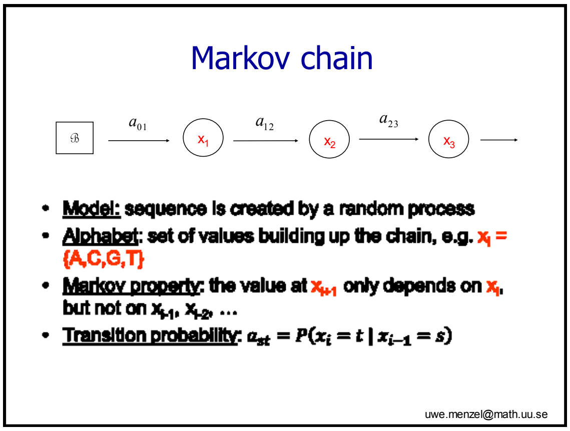
Contents

1. What’s a Markov chain and what has it to do with DNA?
2. A Likelihood Ratio Test using Markov chains that decides if a small piece of DNA is a CpG island or not
3. A Hidden Markov Model that can locate CpG islands in a large piece of DNA
4. Parameter estimation for HMM’s
5. A Continuous Density Hidden Markov Model that can recognise amplifications/deletions of large chunks of genomic DNA on a chromosome



1. What’s a Markov chain and what has it to do with DNA?

Андрей Андреевич Марков (1856 – 1922)

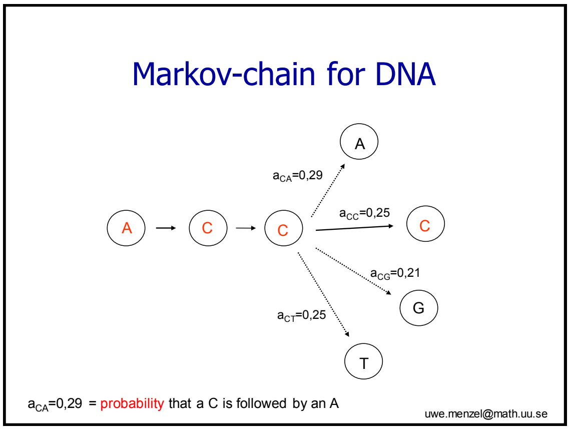


Markov chain

$\mathcal{B}$
a_{01}
x_1
a_{12}
x_2
a_{23}
x_3

- Model: sequence is created by a random process
- Alphabet: set of values building up the chain, e.g. x_i = {A,C,G,T}
- Markov property: the value at x_{i+1} only depends on x_i, but not on x_{i-1}, x_{i-2}, …
- Transition probability: $a_{st} = P(x_i = t \mid x_{i-1} = s)$

uwe.menzel@math.uu.se

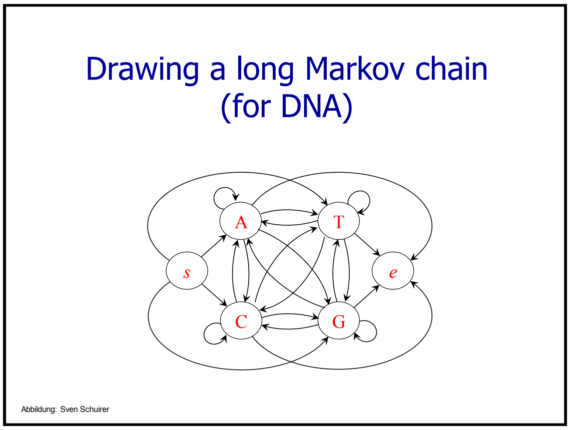


Markov-chain for DNA

A
C
C
A
a_{CA}=0,29
a_{CC}=0,25
C
a_{CG}=0,21
G
a_{CT}=0,25
T

a_{CA}=0,29 = probability that a C is followed by an A

uwe.menzel@math.uu.se



Drawing a long Markov chain (for DNA)

Abbildung: Sven Schuirer

Notation for the joint probability of a chain

$P(\vec{x}) = P(X_1 = x_1, \quad X_2 = x_2, \quad X_3 = x_3, \quad \dots, \quad X_N = x_N)$

$= P(x_1, x_2, x_3, \dots, x_N)$

$P(\vec{x}) = P(X_1 = A, \quad X_2 = C, \quad X_3 = C, \quad X_4 = G, \quad X_5 = T)$

$= P(A, C, C, G, T)$

Probability of a particular chain

$P(\vec{x}) = P(x_1, x_2, x_3, \dots, x_{N-1}, x_N)$

Use multiple times : $P(x,y) = P(x \mid y) \cdot P(y)$

$P(\vec{x}) = P(x_1) \cdot P(x_2 \mid x_1) \cdot \underline{P(x_3 \mid x_2, x_1)} \cdot P(x_4 \mid x_3, x_2, x_1) \cdot \dots$

With the Markov property ,this becomes much easier :

$P(\vec{x}) = P(x_1) \cdot P(x_2 \mid x_1) \cdot \underline{P(x_3 \mid x_2)} \cdot P(x_4 \mid x_3) \cdot \dots \cdot P(x_N \mid x_{N-1})$

Probability of the Markov chain[1]

$P(\vec{x}) = P(x_1) \cdot P(x_2 \mid x_1) \cdot P(x_3 \mid x_2) \cdot P(x_4 \mid x_3) \cdot \dots \cdot P(x_N \mid x_{N-1})$

let $a_{x_{i-1}x_i} = P(x_i \mid x_{i-1})$

$P(\vec{x}) = P(x_1) \cdot a_{x_1x_2} \cdot a_{x_2x_3} \cdot \quad \dots \quad \cdot a_{x_{N-2}x_{N-1}} \cdot a_{x_{N-1}x_N}$

$P(\vec{x}) = P(x_1) \cdot \prod_{i=2}^{N} a_{x_{i-1}x_i} \quad$ *with* $\quad P(x_1) = a_{x_0x_1}$

$P(\vec{x}) = \prod_{i=1}^{N} a_{x_{i-1}x_i}$

[1]considering homogeneous Markov-chains only

ML estimators for the transition probabilities in DNA

- count the frequency of dinucleotides in genomic data c_{st}
- Normalization: (a = probabilities; c = "counts"):

$$a_{st} = \frac{c_{st}}{\sum_i c_{si}} \qquad s,t \in \{A,C,G,T\}$$

ML estimators for the transition probabilities in DNA

$c_{CG} = 100 \qquad c_{CA} = 150 \qquad c_{CT} = 50 \qquad c_{CC} = 100$

$\boxed{a_{CG} = \frac{c_{CG}}{c_{CG} + c_{CA} + c_{CT} + c_{CC}}} = \frac{100}{100 + 150 + 50 + 100} = 0{,}25$

$a_{CA} = \frac{150}{400} = 0{,}375$

$a_{CT} = \frac{50}{400} = 0{,}125$

$a_{CC} = \frac{100}{400} = 0{,}25$

$\boxed{a_{CG} + a_{CA} + a_{CT} + a_{CC} = 1 \quad \textit{line total}}$

Matrix of transition probabilities

	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

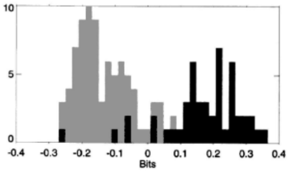
Stochastisc matrix

$P(C, A, A, G) = a_{0C} \cdot a_{CA} \cdot a_{AA} \cdot a_{AG}$

$= 0{,}25 \cdot 0{,}322 \cdot 0{,}300 \cdot 0{,}285 = 0{,}00688$

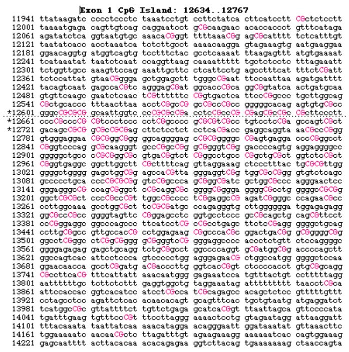
Language recognition (Markov)

2. A Likelihood Ratio Test using Markov chains that decides if a small piece of DNA is a CpG island or not



What is a CpG island ?

- What CpG frequency do we expect ?
  - $P_{CG} \approx$ ¼ · ¼ = 1/16 ; more precisely 0,21 · 0,21 ≈ 4,4%
- actual frequency is only 0,8 % (mammalia)
- cytosine (C) in a CpG is chemically unstable:
  - methylation, deamination: CG → $C^{meth}G$ → TG
- CpG-islands have a higher CpG percentage, compared to the rest of the genome



```
              Exon 1 CpG Island: 12634..12767
 11941 ttataagatc cccctccctc taaatcctgt ccttctatca cttcatcctt CGctctcctt
 12001 taaaatgaga cagttgtcag caggaatcct gCGcaagaac acaccacccct gtttcataga
 12061 agatatctca ggtaatgtgc aaacaCGggt ttttaaaCGg agCGcatttt tctcatttgt
 12121 taatatcacc acctaaatca tctcttgcct aaaacaagga gtagaaagtg aatgaaggaa
 12181 ggaacaggtg atggtcagtg tcctttctac gcctcaaaat ttaagagttt atgtgaaaat
 12241 tcataaatat taatctcaat ccaggttaag caaaattttt tgctctcctc tttagaaatt
 12301 tctggttgcc aaagttccag aaattgcttc ctcattcctg agcctttcat tttctCGatt
 12361 tctccattat gtaaCGggga gctggagctt tgggcCGaat ttccaattaa agatgatttt
 12421 tacagtcaat gagccaCGtc agggagCGat ggcaccCGca ggCGgtatca actgatgcaa
 12481 gtgttcaagc gaatctcaac tCGttttttc CGgtgactca ttccCGgccc tgcttggcag
 12541 CGctgcaccc tttaacttaa acctCGgcCG gcCGccCGcc gggggcacag agtgtgCGcc
*12601 gggcCGCGCG gcaattggtc ccCGCGcCGa cctcCGccCG CGagCGcCGc CGcttccctt
*12661 cccCGcccCG CGtccctccc cctCGgcccc gCGCGtCGcc tgtcctcCGa gccagtCGct
*12721 gacagcCGCG gCGcCGCGag cttctcctct cctcaCGacc gaggcaggta aaCGccCGgg
 12781 gtgggaggaa CGCGggCGgg ggcaggggag cCGCGggggc CGagtgagga cccCGggcct
 12841 CGggtcccag gCGcaagggt gccCGgcCGg gCGgggtCGg gaccccagtg aggaggggcc
 12901 gggggctgcc cCGCGggCGc gtgaCGgtct CGggcctgcc CGgctgCGct ggtctcCGct
 12961 CGggtgaggc ggcttggctt CGcttttcag gttaggaaag ctccctttac tgCGCGttgg
 13021 ggggctgggg gagctggCGg agccaCGtta gggaggtCGg tggCGcCGgg gtgtctcagc
 13081 gccccctgca cccCGCGCGg gtcCGgccca gCGggCGatc gctggCGccc agggaactcc
 13141 gggagggcCG ccagCGggct cCGcaggCGc ggggCGggga ggggCGcctg ggggcCGCGg
 13201 ggctCGCGct cccCGccCGt tggcCGcccc tCGgaggcCG agatCGgggc ccagaaCGcc
 13261 ccttggcaaa gcctggCGct tcCGCGatgc ccagagggtg cttgggggga tggagagagg
 13321 ggCGccCGcc ggggtagttc CGggagcctc ggtgcctccc gcCGcagctg cagCGttcct
 13381 ccCGggaggc ggcccagccc ttcatcctCG cCGcctgagc ttctcCGagg ggggctgcag
 13441 ccttgCGgcc gttgccacCG cctggagaag CGgcccaCGc ggactgaCGg gCGggggCGg
 13501 ggcctCGggc ctCGgCGggg gCGgggtcCG gggaggcccc accctctgtt ctccaggggc
 13561 ggggagagag gagctgcagg tctgCGgcct ggccccaggt gCGatggCGg accccagctt
 13621 ggccagtcac attcctccca gtccccctgg agggagaaCG ctggccatgg ggggctccaa
 13681 ggaacaacca gcctCGgatg aCGacccttg ggtcacCGgt ctccccacct gtgCGgcagg
 13741 CGccttcaCG tttcattatt aaacaatggg gagaaatcca tgtttactgt cctttttagg
 13801 aatttttttgc tcttctcttt gaggtggctg taggaaatag attttttttt taacctCGca
 13861 attccaccac ggtcacatcc atcctCGcca tCGcagagcc acagctctcc gtttttgttt
 13921 cctagcctcc agattctcac acaacacagt gcagtttcac tgctgtaatg atgaggatct
 13981 tcatggcCGc gttattttct tgttctgaga gcatcaCGgt ttaattagca gttccccata
 14041 tgatttgaag tgtttccCGt ttccttaggg aaaactcctg gtagaatagg attaaggatt
 14101 tttacaaata taattatcaa aaacatagga acagggaatt ggataaatat gttaaacttc
 14161 tggaaaaatc aacaaCGctc ttagatttgt agaagaaagg aaaaatcac cagtggaaag
 14221 gagcaatttt acttacacaa acacagagaa ggtcttacag tgaaaaaaag ctaaccagta
```

"Training": Finding transition probabilities for both CpG islands and non-islands

CpG-Islands

	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

mod+

Non-Islands

	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

mod-

$\boxed{a^+_{CG} = 0{,}274}$ $a^-_{CG} = 0{,}078$

How to discriminate between CpG islands and non-islands

X = (ATCGCGCGGC)

$$\underline{P(X \mid \text{mod}+)} = \prod_i a^+_{x_{i-1}x_i} = a^+_{0A} \cdot a^+_{AT} \cdot a^+_{TC} \cdot a^+_{CG} \cdot a^+_{GC} \cdot a^+_{CG} \cdot a^+_{GC} \cdot a^+_{CG} \cdot a^+_{GG} \cdot a^+_{GC}$$

$$= 0{,}25 \cdot 0{,}120 \cdot 0{,}355 \cdot 0{,}274 \cdot 0{,}339 \cdot 0{,}274 \cdot 0{,}339 \cdot 0{,}274 \cdot 0{,}375 \cdot 0{,}339 = \underline{3{,}125 \cdot 10^{-6}}$$

$$\underline{P(X \mid \text{mod}-)} = \prod_i a^-_{x_{i-1}x_i} = a^-_{0A} \cdot a^-_{AT} \cdot a^-_{TC} \cdot a^-_{CG} \cdot a^-_{GC} \cdot a^-_{CG} \cdot a^-_{GC} \cdot a^-_{CG} \cdot a^-_{GG} \cdot a^-_{GC}$$

$$= 0{,}25 \cdot 0{,}210 \cdot 0{,}239 \cdot 0{,}078 \cdot 0{,}246 \cdot 0{,}078 \cdot 0{,}246 \cdot 0{,}078 \cdot 0{,}298 \cdot 0{,}246 = \underline{2{,}65 \cdot 10^{-8}}$$

Result: *It is more likely that X is a CpG-Island*

Likelihood Ratio Test for Discrimination of CpG islands and non-islands

- According to the model, a sequence X is a CpG-island if:

$$P(X \mid \text{mod}+) > P(X \mid \text{mod}-)$$

$$\frac{P(X \mid \text{mod}+)}{P(X \mid \text{mod}-)} > 1$$

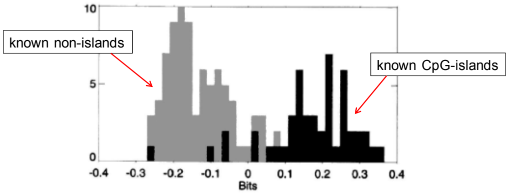
log likelihood ratios

$$\boxed{S = \log\left[\frac{P(X \mid \text{mod}+)}{P(X \mid \text{mod}-)}\right]} = \sum_{i=1}^{L} \log \frac{a^{+}_{x_{i-1}x_i}}{a^{-}_{x_{i-1}x_i}} = \sum_{i=1}^{L} \beta_{x_{i-1}x_i} > 0$$

log odds score (S)

Does the method really work?

- Go back and calculate the scores for both training sets:



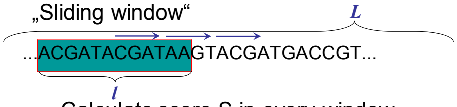
Errors caused by: incorrect labels in the training sets, hard to determine borders between CpG-islands and non-islands

Picture from: Durbin et al. (Ed): Biological Sequence Analysis, Cambridge University Press, 1998

Pros and Cons of the scoring model

- Given a short piece of DNA, you can decide if it is a CpG-island or not
- You cannot identify a potential CpG-island in a long sequence (⟹ HMM)

Long sequence: Finding CpG-islands with a sliding window



– Calculate score S in every window

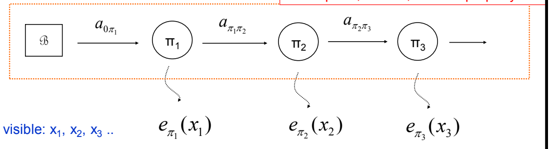
– Disadvantages:
  - Runtime (?)
  - unknown size of the island (?)

Bildquelle: Sven Schuirer

3. A Hidden Markov Model that can locate CpG islands in a large piece of DNA

Hidden Markov Model

state ”path”, hidden, Markov property



visible: x_1, x_2, x_3 ..

Probability of a particular sequence of states and symbols:

$$P(x,\pi) = a_{0\pi_1} \cdot \quad e_{\pi_1}(x_1) \cdot a_{\pi_1\pi_2} \cdot \quad e_{\pi_2}(x_2) \cdot a_{\pi_2\pi_3} \cdot \quad \ldots$$

$$P(x,\pi) = a_{0\pi_1} \cdot \prod_{i=1}^{L} e_{\pi_i}(x_i) \cdot a_{\pi_i\pi_{i+1}}$$

Probability of a particular sequence of states and symbols

:

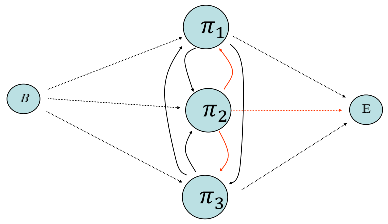
$$P(x,\pi) = a_{0\pi_1} \cdot \quad e_{\pi_1}(x_1) \cdot a_{\pi_1\pi_2} \cdot \quad e_{\pi_2}(x_2) \cdot a_{\pi_2\pi_3} \cdot \quad \ldots$$

$$P(x,\pi) = a_{0\pi_1} \cdot \prod_{i=1}^{L} e_{\pi_i}(x_i) \cdot a_{\pi_i\pi_{i+1}}$$

$a_{kl} = \mathrm{P}(\pi_i = l \mid \pi_{i-1} = k)$ transition probabilities (within state path)

$e_k(b) = P(x_i = b \mid \pi_i = k)$ emission probabilities

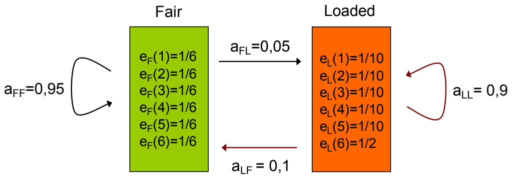
Drawing a HMM for an arbitrary long sequence



$$a_{\pi_2\pi_1} + a_{\pi_2\pi_3} + a_{\pi_2E} = 1$$

uwe@math.uu.se

HMM: Casino with a fair and a loaded die



Fair	Loaded
$e_F(1)$=1/6	$e_L(1)$=1/10
$e_F(2)$=1/6	$e_L(2)$=1/10
$e_F(3)$=1/6	$e_L(3)$=1/10
$e_F(4)$=1/6	$e_L(4)$=1/10
$e_F(5)$=1/6	$e_L(5)$=1/10
$e_F(6)$=1/6	$e_L(6)$=1/2

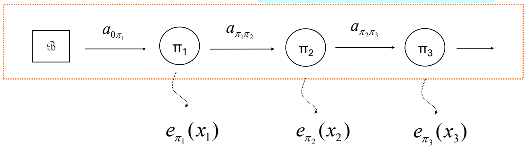
Observer sees emissions only): 3 4 2 4 6 4 6 3 4 6 6 3 6 6 3 4 6 6

State is hidden for the observer: F F F F F F F F F L L L L L L L L L L L

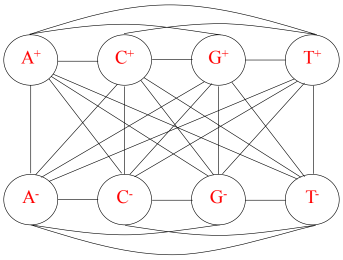
HMM for the recognition of CpG islands embedded in genomic DNA

- States: A^+, C^+, G^+, T^+, A^-, C^-, G^-, T^-
- Symbols: A, C, G, T

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



HMM for CpG islands in DNA-sequence



Bildquelle: Sven Schuirer

Bildquelle: Sven Schuirer

Transition probabilities

π_i/π_{i+1}	A^+	C^+	G^+	T^+	A^-	C^-	G^-	T^-
A^+	0.180p	0.274p	0.426p	0.120p	$\frac{1-p}{4}$	$\frac{1-p}{4}$	$\frac{1-p}{4}$	$\frac{1-p}{4}$
C^+	0.171p	0.368p	0.274p	0.188p	$\frac{1-p}{4}$	$\frac{1-p}{4}$	$\frac{1-p}{4}$	$\frac{1-p}{4}$
G^+	0.161p	0.339p	0.375p	0.125p	$\frac{1-p}{4}$	$\frac{1-p}{4}$	$\frac{1-p}{4}$	$\frac{1-p}{4}$
T^+	0.079p	0.355p	0.384p	0.182p	$\frac{1-p}{4}$	$\frac{1-p}{4}$	$\frac{1-p}{4}$	$\frac{1-p}{4}$
A^-	$\frac{1-q}{4}$	$\frac{1-q}{4}$	$\frac{1-q}{4}$	$\frac{1-q}{4}$	0.300q	0.205q	0.285q	0.210q
C^-	$\frac{1-q}{4}$	$\frac{1-q}{4}$	$\frac{1-q}{4}$	$\frac{1-q}{4}$	0.322q	0.298q	**0.078q**	0.302q
G^-	$\frac{1-q}{4}$	$\frac{1-q}{4}$	$\frac{1-q}{4}$	$\frac{1-q}{4}$	0.248q	0.246q	0.298q	0.208q
T^-	$\frac{1-q}{4}$	$\frac{1-q}{4}$	$\frac{1-q}{4}$	$\frac{1-q}{4}$	0.177q	0.239q	0.292q	0.292q

p = P (*remains in CpG-islands*) ≈*0.95* , *q = P (remains in non-island)* ≈*0.99*

Emission probabilities

$$e_{C^+}(C) = 1$$

emitting state

symbol that is emitted

$$e_{C^+}(C)=1; \quad e_{C^-}(C)=1 \qquad e_{\pi_i}(C)=0 \ \ otherwise$$
$$e_{A^+}(A)=1; \quad e_{A^-}(A)=1 \qquad e_{\pi_i}(A)=0 \ \ otherwise$$
$$e_{G^+}(G)=1; \quad e_{G^-}(G)=1 \qquad e_{\pi_i}(G)=0 \ \ otherwise$$
$$e_{T^+}(T)=1; \quad e_{T^-}(T)=1 \qquad e_{\pi_i}(T)=0 \ \ otherwise$$

Decoding: finding hidden states from observations

- Observed sequence (”*emissions*”):
  - C G C G (Sorry!, still a short sequence !!)
- might have been generated by the ”*state*”-sequences:
  - $C^+ G^+ C^+ G^+$
  - $C^- G^- C^- G^-$
  - $C^+ G^- C^+ G^-$
  - …
- How to find the ”best” sequence of states ?
  - practical not possible to calculate all potential pathes …
  - → Viterbi – algorithm (”dynamic programming”)

Transition probabilities (see table above)

$$p = 0{,}95 \ (stays\ in\ +) \qquad q = 0{,}99 \ (stays\ in\ -)$$
$$a_{C^+G^+} = 0{,}274 \cdot 0{,}95 = 0{,}26$$
$$a_{G^+C^+} = 0{,}339 \cdot 0{,}95 = 0{,}322$$
$$a_{C^-G^-} = 0{,}078 \cdot 0{,}99 = 0{,}0772$$
$$a_{G^-C^-} = 0{,}246 \cdot 0{,}99 = 0{,}2435$$
$$a_{C^+G^-} = (1-0{,}95)/4 = 0{,}0125 \quad small$$
$$a_{G^-C^+} = (1-0{,}99)/4 = 0{,}0025 \quad small$$

HMM:

$$P(x,\pi) = a_{0\pi_1} \cdot \prod_{i=1}^{L} e_{\pi_i}(x_i) \cdot a_{\pi_i \pi_{i+1}}$$

$$\boxed{\begin{aligned}&P(X=C,G,C,G;\quad \pi = C^+,G^+,C^+,G^+) =\\ &= a_{0C^+} \cdot e_{C^+}(C) \cdot \quad a_{C^+G^+} \cdot e_{G^+}(G) \cdot \quad a_{G^+C^+} \cdot \quad e_{C^+}(C) \cdot \quad a_{C^+G^+} \cdot \quad e_{G^+}(G) \cdot a_{G^+0}\\ &= 0{,}13 \cdot \quad 1 \cdot \quad 0{,}26 \cdot \quad 1 \cdot \quad 0{,}322 \cdot \quad 1 \cdot \quad 0{,}26 \cdot \quad 1 \cdot \quad 1 \quad = 0{,}00283\end{aligned}}$$

$$\boxed{\begin{aligned}&P(X=C,G,C,G;\quad \pi = C^-,G^-,C^-,G^-) =\\ &= a_{0C^-} \cdot e_{C^-}(C) \cdot \quad a_{C^-G^-} \cdot e_{G^-}(G) \cdot \quad a_{G^-C^-} \cdot \quad e_{C^-}(C) \cdot \quad a_{C^-G^-} \cdot \quad e_{G^-}(G) \cdot a_{G^-0}\\ &= 0{,}13 \cdot \quad 1 \cdot \quad 0{,}0772 \cdot \quad 1 \cdot \quad 0{,}2435 \cdot \quad 1 \cdot \quad 0{,}0772 \cdot \quad 1 \cdot \quad 1 \quad = 0{,}000189\end{aligned}}$$

$$\boxed{\begin{aligned}&P(X=C,G,C,G;\quad \pi = C^+,G^-,C^+,G^-) =\\ &= a_{0C^+} \cdot e_{C^+}(C) \cdot \quad a_{C^+G^-} \cdot \quad e_{G^-}(G) \cdot \quad a_{G^-C^+} \cdot \quad e_{C^+}(C) \cdot \quad a_{C^+G^-} \cdot \quad e_{G^-}(G) \cdot \quad a_{G^-0}\\ &= 0{,}13 \cdot \quad 1 \cdot \quad 0{,}0125 \cdot \quad 1 \cdot \quad 0{,}0025 \cdot \quad 1 \cdot \quad 0{,}0125 \cdot \quad 1 \cdot \quad 1 \quad = 5 \cdot 10^{-8}\end{aligned}}$$

Finding the most probable path: the Viterbi algorithm

$$P(x,\pi) = a_{0\pi_1} \cdot \prod_{i=1}^{L} e_{\pi_i}(x_i) \cdot a_{\pi_i \pi_{i+1}}$$

Solution: $\pi^* = argmax_\pi \ P(x,\pi) = argmax_\pi \ P(\pi \mid x)$

Recursion:

$v_k(i)$: probability of the most probable path ending in state k with observation i

$$\boxed{v_l(i+1) = e_l(x_{i+1}) \ max_k\{v_k(i) \cdot a_{kl}\}}$$

Viterby

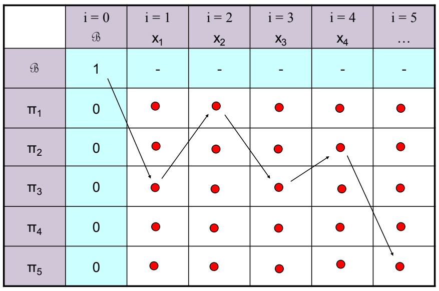
Durbin et al., Biological sequence analysis, p. 56

$$v_l(i+1) = e_l(x_{i+1}) \ max_k \ (v_k(i) \cdot a_{kl})$$

v	v(0)	C	G	C	G
B	1	0	0	0	0
A_+	0	0	0	0	0
C_+	0	0.13	0	0.012	0
G_+	0	0	0.034	0	0.0032
T_+	0	0	0	0	0
A_-	0	0	0	0	0
C_-	0	0.13	0	0.0026	0
G_-	0	0	0.010	0	0.00021
T_-	0	0	0	0	0

We've found a CpG island, thus!

Trellis-Diagramm



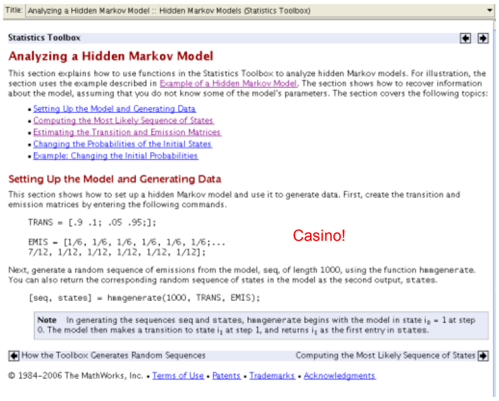
	i = 0 ℬ	i = 1 x_1	i = 2 x_2	i = 3 x_3	i = 4 x_4	i = 5 …
ℬ	1	-	-	-	-	-
π_1	0	●	●	●	●	●
π_2	0	●	●	●	●	●
π_3	0	●	●	●	●	●
π_4	0	●	●	●	●	●
π_5	0	●	●	●	●	●

Remarks

- Result: we've found that the whole sequence CGCG is a CpG island (sequence was very short!)
- Method works for arbitrary long sequence and might then switch between long stretches of + and – states, respectively … as it should
- Most probable path (Viterby) is not the only solution: posterior decoding; $\hat{\pi}_i = argmax_k P(\pi_i = k \mid x\,)$ calculated by the Forward- and Backward algorithm

Software

- R – Scripte:
  - CRAN: Bioconductor
  - http://www.stat.uni-muenchen.de/~semwiso/stochastische-prozesse/
    (Ludwig Fahrmeir / Christiane Belitz)
- MATLAB
  - stats package



Title: Analyzing a Hidden Markov Model :: Hidden Markov Models (Statistics Toolbox)

Statistics Toolbox

Analyzing a Hidden Markov Model

This section explains how to use functions in the Statistics Toolbox to analyze hidden Markov models. For illustration, the section uses the example described in Example of a Hidden Markov Model. The section shows how to recover information about the model, assuming that you do not know some of the model's parameters. The section covers the following topics:

- Setting Up the Model and Generating Data
- Computing the Most Likely Sequence of States
- Estimating the Transition and Emission Matrices
- Changing the Probabilities of the Initial States
- Example: Changing the Initial Probabilities

Setting Up the Model and Generating Data

This section shows how to set up a hidden Markov model and use it to generate data. First, create the transition and emission matrices by entering the following commands.

```
TRANS = [.9 .1; .05 .95;];

EMIS = [1/6, 1/6, 1/6, 1/6, 1/6, 1/6;...
7/12, 1/12, 1/12, 1/12, 1/12, 1/12];
```

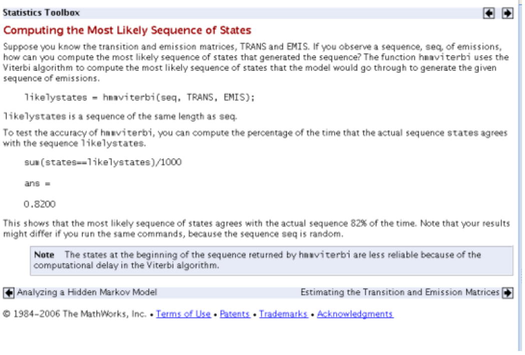
Casino!

Next, generate a random sequence of emissions from the model, seq, of length 1000, using the function hmmgenerate. You can also return the corresponding random sequence of states in the model as the second output, states.

```
[seq, states] = hmmgenerate(1000, TRANS, EMIS);
```

Note In generating the sequences seq and states, hmmgenerate begins with the model in state i_0 = 1 at step 0. The model then makes a transition to state i_1 at step 1, and returns i_1 as the first entry in states.

How the Toolbox Generates Random Sequences | Computing the Most Likely Sequence of States



Statistics Toolbox

Computing the Most Likely Sequence of States

Suppose you know the transition and emission matrices, TRANS and EMIS. If you observe a sequence, seq, of emissions, how can you compute the most likely sequence of states that generated the sequence? The function hmmviterbi uses the Viterbi algorithm to compute the most likely sequence of states that the model would go through to generate the given sequence of emissions.

```
likelystates = hmmviterbi(seq, TRANS, EMIS);
```

likelystates is a sequence of the same length as seq.

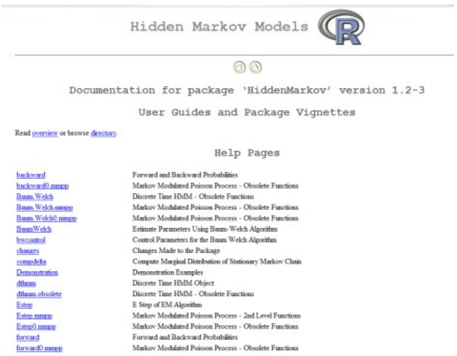
To test the accuracy of hmmviterbi, you can compute the percentage of the time that the actual sequence states agrees with the sequence likelystates.

```
sum(states==likelystates)/1000

ans =

0.8200
```

This shows that the most likely sequence of states agrees with the actual sequence 82% of the time. Note that your results might differ if you run the same commands, because the sequence seq is random.

Note The states at the beginning of the sequence returned by hmmviterbi are less reliable because of the computational delay in the Viterbi algorithm.

Analyzing a Hidden Markov Model | Estimating the Transition and Emission Matrices



Hidden Markov Models

Documentation for package 'HiddenMarkov' version 1.2-3

User Guides and Package Vignettes

Read overview or browse directory.

Help Pages

backward	Forward and Backward Probabilities
backward0.mmpp	Markov Modulated Poisson Process - Obsolete Functions
Baum.Welch	Discrete Time HMM - Obsolete Functions
Baum.Welch.mmpp	Markov Modulated Poisson Process - Obsolete Functions
Baum.Welch0.mmpp	Markov Modulated Poisson Process - Obsolete Functions
BaumWelch	Estimate Parameters Using Baum-Welch Algorithm
bwcontrol	Control Parameters for the Baum Welch Algorithm
changes	Changes Made to the Package
compdelta	Compute Marginal Distribution of Stationary Markov Chain
Demonstration	Demonstration Examples
dthmm	Discrete Time HMM Object
dthmm.obsolete	Discrete Time HMM - Obsolete Functions
Estep	E Step of EM Algorithm
Estep.mmpp	Markov Modulated Poisson Process - 2nd Level Functions
Estep0.mmpp	Markov Modulated Poisson Process - Obsolete Functions
forward	Forward and Backward Probabilities
forward0.mmpp	Markov Modulated Poisson Process - Obsolete Functions

4. Parameter estimation for HMM's

? $a_{\pi_i \pi_{i+1}}$ $e_{\pi_i}(x_i)$?

4. 1. Parameter estimation if we have a training set where the states are known

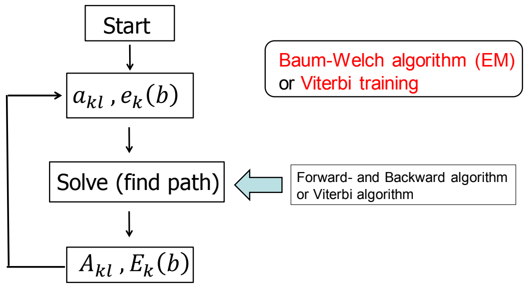
$$a_{kl} = \frac{A_{kl}}{\sum_{l} A_{kl}} \qquad k,l \in \{A^+,C^+,G^+,T^+,A^-,C^-,G^-,T^-\}$$

$$e_k(b) = \frac{E_k(b)}{\sum_{b} E_k(b)} \qquad k \in \{A^+,C^+,G^+,T^+,A^-,C^-,G^-,T^-\}$$

$$b \in \{A,C,G,T\}$$

- A, E: counts, from the training set with known path
- a and e are ML estimators, as before
- problem with overfitting

4. 2. Parameter estimation if we have no training set with known states



Start

$a_{kl}, e_k(b)$

Solve (find path)

$A_{kl}, E_k(b)$

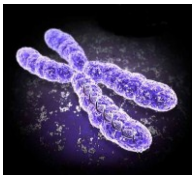
Baum-Welch algorithm (EM) or Viterbi training

Forward- and Backward algorithm or Viterbi algorithm

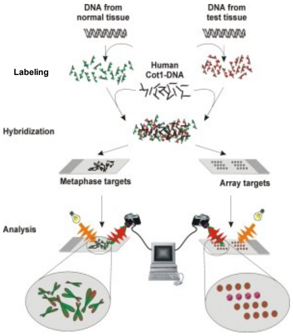
Sources

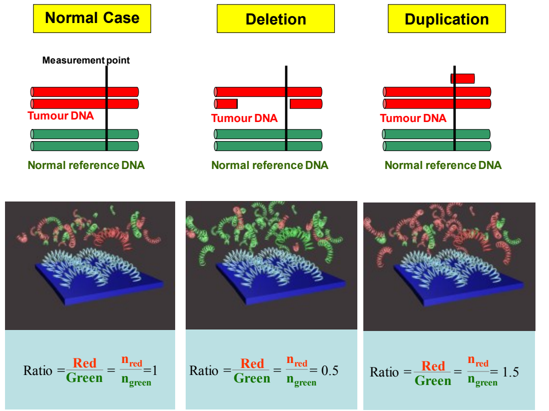
- Durbin et al (Ed.)., *Biological Sequence Analysis*, Cambridge University Press 1998
- Rabiner, L. R., *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, Proceedings of the IEEE, Vol. 77, No. 2, 1989
- http://www.stat.uni-muenchen.de/~semwiso/stochastische-prozesse/
- http://www.itu.dk/~sestoft/bsa.html

5. A Continuous Density Hidden Markov Model that can recognise amplifications and deletions of large chunks of genomic DNA on a chromosome



Metaphase-CGH and Microarray-CGH

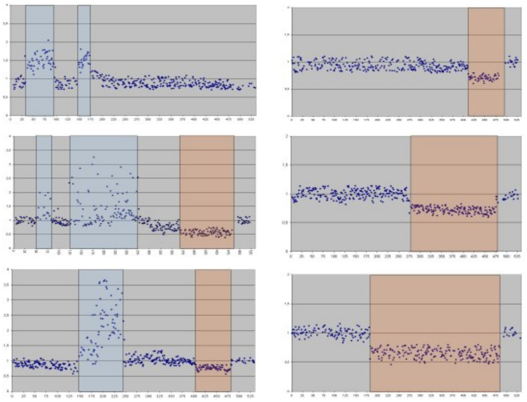


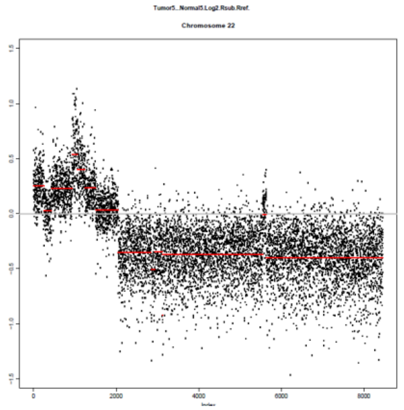


$\text{Ratio} = \frac{\text{Red}}{\text{Green}} = \frac{n_{red}}{n_{green}} = 1$

$\text{Ratio} = \frac{\text{Red}}{\text{Green}} = \frac{n_{red}}{n_{green}} = 0.5$

$\text{Ratio} = \frac{\text{Red}}{\text{Green}} = \frac{n_{red}}{n_{green}} = 1.5$





Continuous Density Hidden Markov Model

- Hidden states: Copy number (0, 1, 2, 3, 4, >4)
- Emission probabilities: Gaussian spectrum

$$\boxed{e_{\pi_i}(x_i) = P(x_i \mid \pi_i, \mu_i, \sigma_i) \sim \mathrm{N}(\mu_i, \sigma_i)}$$

SMAP

- Segmental Maximum A Posteriori
- maximize the joint posterior probability of the states (π) and the parameters (μ)

$$\boxed{\hat{\vartheta} = argmax_{\vartheta}\ max_{\pi}\ P(\vartheta, \pi \mid x)}$$

$\vartheta = (a, \mu, \sigma)$: transition prob., means and variances of the Gaussians

Recursion: maximize with respect to π , then with respect to ϑ

Segmental MAP

$p(\theta, z|x) = \frac{p(z,\theta,x)}{p(x)} = \frac{p(z,x|\theta)\cdot p(\theta)}{p(x)}$

Find a θ that maximizes $p(\theta, z|x)$:

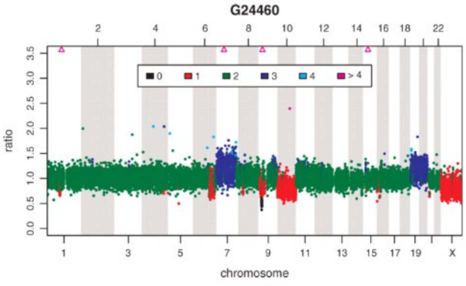
$$\theta = \operatorname*{argmax}_{\theta} \max_{z} p(\theta, z|x) = \operatorname*{argmax}_{\theta} \max_{z} p(x, z|\theta) \cdot p(\theta)$$

Alternate maximization over z and θ yields a sequence of non-decreasing $p(\theta, z|x)$:

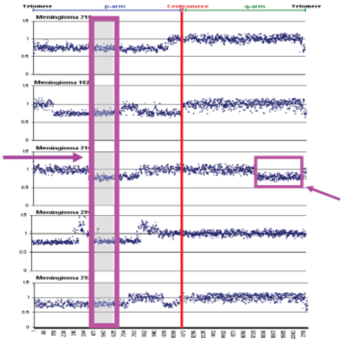
$$z_{t+1} = \operatorname*{argmax}_{z} p(x, z|\theta_t) \qquad \text{Viterbi}$$

$$\theta_{t+1} = \operatorname*{argmax}_{\theta} p(x, z_{t+1}|\theta) \cdot p(\theta)$$

SMAP - Result



Six meningiomas analyzed on chr. 1 array



Analysis of 1p will allow to define a small overlapping region of deletions

Deletions on 1q have not been described so far in meningioma

Thank you for your attention!

Entropy of a DNA-sequence

Let x_i *be an alphabet, e.g.* $x_i = \{A, C, G, T\}$

$$H(X) = -\sum_i p(x_i) \cdot \log(p(x_i)) = -\sum_i p_i \cdot \log(p_i)$$

$$p(A) = p(C) = p(G) = p(T) = \frac{1}{4}$$

$$H = -\sum_{i=1}^{4} \frac{1}{4} \cdot \log_2\left(\frac{1}{4}\right) = 2 \quad 2\ bit; \quad 2\ Yes/No-questions$$

Durbin et al, Chapter 11.2

Rett-Syndrom

- 1:15,000 (nur Mädchen)
- Im Alter von ca. 1 Jahr verlieren sie das Interesse an anderen Menschen und entwickeln stereotypische Verhaltensweisen (z.B. Händeringen)
- Ursache: Mutation in **MeCP2** (methyl CpG-binding protein 2, X-Chr.)
- Bindet an methylierte **CpG-Inseln** in Promotoren

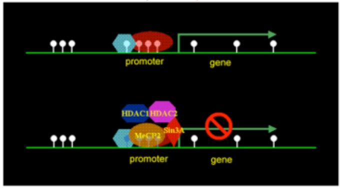


Bild: Martin Lercher, Düsseldorf