

CS-E5865 Computational genomics

Autumn 2019, Lecture 5: HMM algorithms, Pair HMM & Profile HMM Lecturer: Pekka Marttinen Assistants: Alejandro Ponce de León, Zeinab Yousefi, Onur Poyraz

HMM problems and algorithms

- Find the most likely hidden state sequence => Viterbi algorithm
- Estimate the emission and transition probabilities of the HMM => Viterbi training ✓
- Determine the probability of a sequence s given the HMM model => Forward algorithm
- Determine the probability of being in state k at position i
 => Posterior decoding

Forward, Backward & Posterior decoding



The Forward Algorithm

- Task: calculate the probability P(s) of sequence s, given by our HMM
- Sum over all possible hidden state paths (set Π) that could have been used to generate s:

$$P(s) = \sum_{\pi \in \Pi} P(s,\pi) = \sum_{\pi \in \Pi} P(s|\pi)P(\pi)$$

- Exponential sum, cannot enumerate over all state paths!
- Again, we will define a dynamic programming problem, and fill a table of *forward probabilities*

$$F_{k}(i) = P(s_{1}...s_{i}, \pi_{i} = k)$$

- Probability of emitting the prefix $\boldsymbol{s}_1,...,\boldsymbol{s}_i$ and ending up in state k

The Forward Algorithm – derivation

$$F_k(i) = P(s_1...s_i, \pi_i = k)$$

$$= \Sigma_{l} P(s_{1}...s_{i}, \pi_{i-1} = l, \pi_{i} = k)$$

$$= \sum_{l} P(s_1...s_{i-1}, \pi_{i-1} = l) P(\pi_i = k | \pi_{i-1} = l) P(s_i | \pi_i = k)$$

$$= \Sigma_{l} \mathbf{P}(\mathbf{s}_{1} \dots \mathbf{s}_{i-1}, \boldsymbol{\pi}_{i-1} = \mathbf{I}) \mathbf{T}_{lk} \mathbf{E}_{k}(\mathbf{s}_{i})$$

$$= E_k(s_i) \Sigma_l \frac{F_l(i-1)}{T_{lk}}$$

• Sum over all possibilities of emitting $s_1, ..., s_{i-1}$ ending up in state 1, and then making a transition from 1 to k, and emitting s_i



The Forward Algorithm

- $F_k(i) = E_k(s_i) \Sigma_l F_l(i-1) T_{lk}$
- Dynamic programming formulation:
 - table F of size m X n where:
 - m=num of hidden states
 - n=length of the observed sequence

Initialization: (first column) $F(k,1) = 1/m E_k(s_1)$, for all k > 0**Iteration**

- $$\begin{split} F(k,i) &= E_k(s_i) \ \Sigma_l \ F(l,i-1) \ T_{lk,} \ \text{for} \\ \text{all } k, \ \text{and for all } i=2,...,n \end{split}$$
- **Termination:** (sum all the values in the last column) $P(s) = \Sigma_k F(k,n)$
- Difference to Viterbi: replace max with sum



Forward at the Casino

```
forward <- function(s, T, E) {</pre>
  n.states <- ncol(E)
  F <- matrix(rep(0, n.states * length(s)), nrow = n.states)</pre>
  F[,1] <-1 / n.states * E[s[1],]
  for (i in 2:length(s)) {
    for (l in 1:n.states) {
      F[1,i] <- sum(F[,i-1] * T[,1])
      F[1,i] <- F[1,i] * E[s[i],1]</pre>
    3
  }
  prob <- sum(F[,length(s)])</pre>
  res <- list()
  res$fprob <- prob</pre>
  res$F <- F
  return(res)
```

0.7 1 2 3 4 5 6	: 1/6 : 1/6 : 1/6 : 1/6 : 1/6 : 1/6 : 1/6 : 1/6	1: 1/ 2: 1/ 3: 1/ 4: 1/ 5: 1/ 6: 1/	0.7 10 10 10 10 10 10 10 10 10 10
> T [1,] [2,] > E [1,] [2,] [3,] [4,] [5,] [6,]	[,1] [0.7 0.3 [0.1666 0.1666 0.1666 0.1666 0.1666	[,2] 0.3 0.7 [,1] [6667 6667 6667 6667 6667	(,2] 0.1 0.1 0.1 0.1 0.1 0.1 0.5

$F(k,i) = E_k(s_i) \Sigma_l F(l,i-1) T_{lk}$

Forward at the Casino • Initialization $F(1,1) = E_{Fair}(s_1) 0.5 = 0.1667 * 0.5 = 0.08335$ $F(2,1) = E_{Loaded}(s_1) 0.5 = 0.1 * 0.5 = 0.0500$	0.7 1: 1/6 2: 1/6 3: 1/6 4: 1/6 5: 1/6 6: 1/6 Fair 0.3 0.3 1: 1/10 2: 1/10 3: 1/10 4: 1/10 5: 1/10 0.3 5: 1/10 6: 1/2 Loaded
> 2	<pre>> T [,1] [,2] [1,] 0.7 0.3 [2,] 0.3 0.7 > E [,1] [,2] [1,] 0.1666667 0.1 [2,] 0.1666667 0.1 [3,] 0.1666667 0.1 [4,] 0.1666667 0.1 [4,] 0.1666667 0.1 [5,] 0.1666667 0.1</pre>
<pre>[1] 3 2 2 1 2 3 6 6 6 6 > forward.res\$F [,1] [,2] [,3] [,4] [,5] [,6] [,7] [1,1 0.0833333 0.01222222 0.0017259259 0.0002406914 3.342288e-05 4.634329e-06 6.422555e-07 [2,1 0.05000000 0.00600000 0.0007866667 0.0001068444 1.469985e-05 2.031676e-06 1.406236e-06 > forward.res\$fprob [1] 1.03473e-07</pre>	[6, 0.1666667 0.5] [,8] [,9] [,10] 7 1.452416e-07 4.637090e-08 1.679837e-08 5 5.885209e-07 2.277686e-07 8.667463e-08

 $F(k,i) = E_k(s_i) \Sigma_l F(l,i-1) T_{lk}$



Forward at the Casino 0.7 1/6 0.3 2:1/6 Compute F(1,2)3:1/6 3: 1/10 For clarity, we'll use informal notation F(fair, 2) for F(1, 2)4: 1/6 4: 1/10 0.3 5:1/6 5: 1/10 $F(fair,2) = P(s_1,s_2, \pi_2 = fair)$ 6: 1/6 6: 1/2 $= E_{fair}(s_2) \Sigma_l F(l,1) T_{l fair}$ Fair Loaded =0.1667 [F(fair,1) $T_{fair,fair}$ + F(loaded,1) $T_{loaded,fair}$] т [.1] [.2][1,] 0.3 0.7 =0.1667[0,0833 * 0,7 + 0,05 * 0,3] = 0.0122[2,] 0.3 0.7 > E [.2] [.1] [1,] 0.1666667 0.1 [2.] 0.1666667 0.1 And so on, until the table is filled... [3.] 0.1666667 0.1 0.1666667 0.1 0.1666667 0.1 [5.] [1] 3 2 2 1 2 3 6 6 6 6 [6.] 0.1666667 0.5 > forward.res\$F [,4] [,1]F.21 [,3] [,5] [,6] [,7] [,8] [,9] [.10] [1,] 0.08333333300.0122222200.0017259259 0.0002406914 3.342288e-05 4.634329e-06 6.422555e-07 1.452416e-07 4.637090e-08 1.679837e-08 [2,] 0.05000000 0.00600000 0.0007866667 0.0001068444 1.469985e-05 2.031676e-06 1.406236e-06 5.885209e-07 2.277686e-07 8.667463e-08 > forward.res\$fprob [1] 1.03473e-07

 $F(k,i) = E_k(s_i) \Sigma_l F(l,i-1) T_{lk}$



Forward at the Casino 0.7 1: 1/10 1: 1/6 2:1/6 The probability of the full sequence s: 3:1/6 3: 1/10 4: 1/6 P(s) = F(fair, 10) + F(loaded, 10)4: 1/100.3 5: 1/6 5: 1/10 Note: working with logarithms is not as ۲ 6: 1/6 6: 1/2 straightforward as with Viterbi (logarithm of a sum Fair Loaded does not simplify). For an algorithm that deals with this issue, see ۲ [.1] [.2] 0.3 e.g., Bishop: Pattern Recognition and Machine Γ2.1 0.3 > E Learning, Ch. 13.2 (not required on this course). [.2] [1,] 0.1666667 0.1 0.1666667 0.1 0.1666667 0.1 0.1 0.1666667 0.1666667 0.1 0.5 [6.] 0.1666667 [1] 3 2 2 1 2 3 6 6 6 6 forward.res\$F [.1] [,2] [,3] [,4] [,5] [.6] [.7] [.8] F.91 [1,] 0.083333333 0.01222222 0.0017259259 0.0002406914 3.342288e-05 4.634329e-06 6.422555e-07 1.452416e-07 4.637090e-08 1.6<u>79837e-0</u>8 [2,] 0.05000000 0.00600000 0.0007866667 0.0001068444 1.469985e-05 2.031676e-06 1.406236e-06 5.885209e-07 2.277686e-07<u>8.667463e-08</u> forward.res\$fprob 1.03473e-07

$$F(k,i) = E_k(s_i) \Sigma_l F(l,i-1) T_{lk}$$



Backward Algorithm - motivation

- Posterior decoding problem: We want to compute the probability of state k for position i given sequence s: P(π_i = k | s)
 - e.g. "During i'th roll Casino was using the loaded dice", "Nucleotide $s_{\rm i}$ belongs to an ORF"
 - This is different from computing the most likely path $\pi_{1} \dots \pi_{n}$ by Viterbi
- We compute the result by splitting the sequence into two parts and computing the probabilities of prefixes and suffixes of s, such that the hidden state at position i is k: $P(\pi_i = k, s) = P(s_1...s_i, \pi_i = k, s_{i+1}...s_n)$

$$= P(s_1...s_i, \pi_i = k) P(s_{i+1}...s_n | s_1...s_i, \pi_i = k)$$

= $P(s_1...s_i, \pi_i = k) P(s_{i+1}...s_n | \pi_i = k)$

Forward, $F_k(i)$ Backward, $B_k(i)$

• Then,
$$P(\pi_i = k | s) = P(\pi_i = k, s) / P(s)$$

The Backward Algorithm – derivation

Define the backward probability:

$$B_{k}(i) = P(s_{i+1}...s_{n} | \pi_{i} = k)$$

$$= \Sigma_{1}P(s_{i+1},s_{i+2}, ..., s_{n}, \pi_{i+1} = l | \pi_{i} = k)$$

$$= \Sigma_{1}P(s_{i+1},s_{i+2}, ..., s_{n} | \pi_{i+1} = l) P(\pi_{i+1} = l | \pi_{i} = k)$$

$$= \Sigma_{1}P(s_{i+2}, ..., s_{n} | \pi_{i+1} = l) P(s_{i+1} | \pi_{i+1} = l) P(\pi_{i+1} = l | \pi_{i} = k)$$

$$= \Sigma_{1}E_{l}(s_{i+1})T_{kl}B_{l}(i+1)$$



The Backward Algorithm

We can compute $B_k(i)$ for all k, i, using dynamic programming

- Fill in a table B of size m X n where:
 - m=nr of hidden states
 - n=length of the observed sequence

Initialization:

B(k,n) = 1, for all k

Iteration: (backward from position n to 1) $B(k,i) = \sum_{l} E_{l}(s_{i+1}) T_{kl} B(l,i+1)$



Backward at the casino

```
backward <- function(s, T, E) {</pre>
  n.states <- ncol(E)
  B \ll matrix(rep(0, n.states * length(s)), nrow = n.states)
  B[, length(s)] <- 1
  for (i in seq(length(s)-1,1)) {
    for (k in 1:n.states) {
      B[k,i] <- sum(E[s[i+1],] * B[,i+1] * T[k,])</pre>
    }
  }
  res <- list()</pre>
  res$B <- B
```





Backward at the casino

- Initialization
- B(1,n)=F(fair,n)=1
- B(2,n)=F(loaded,n)=1



> s
[1] 3 2 2 1 2 3 6 6 6 6
> backward.res\$B
 [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,] 8.623071e-07 6.208959e-06 4.450478e-05 0.0003147900 0.002138403 0.01265679 0.03362963 0.09111111 0.2666667 [2,] 6.322813e-07 4.597620e-06 3.389115e-05 0.0002593093 0.002176988 0.02205926 0.05822222 0.15333333 0.4000000 1



$B(k,i) = \sum_{l} E_{l}(s_{i+1}) T_{kl} B(l,i+1)$

Backward at the casino

• Recursion, for example:

B(fair,7) = P($s_8,...,s_{10}$ | π_7 = fair)

 $=T_{fair,loaded}E_{loaded}(s_8)B(loaded,8)+T_{fair,fair}E_{fair}(s_8)B(fair,8)$ =T_{fair,loaded}E_{loaded}(6)B(loaded,8)+T_{fair,fair}E_{fair}(6)B(fair,8)

= 0.3 * 0.5 * 0.1533 + 0.7 * 0.1667 * 0.0911

=0.0336



> S										
[1]	3221236	5666								
> bac	kward.res\$B									
	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[.7]	[,8]	[,9]	[,10]
[1,] 8	8.623071e-07	6.208959e-06	4.450478e-05	0.0003147900	0.002138403	0.0126567	0.03362963	0.09111111	0.2666667	1
[2,] (6.322813e-07	4.597620e-06	3.389115e-05	0.0002593093	0.002176988	0.02205926	0.05822222	0.15333333	0.400000	1

Posterior Decoding

We can now calculate

 $F_{k}(i) B_{k}(i)$ $P(\pi_{i} = k \mid s) = -----$ P(s)

$$\begin{array}{l} \mathsf{P}(\pi_{i} = \mathsf{k} \mid \mathsf{s}) = \\ \mathsf{P}(\pi_{i} = \mathsf{k} \,, \, \mathsf{s})/\mathsf{P}(\mathsf{s}) = \\ \mathsf{P}(\mathsf{s}_{1}, \, \dots, \, \mathsf{s}_{i}, \, \pi_{i} = \mathsf{k}, \, \mathsf{s}_{i+1}, \, \dots \, \mathsf{s}_{n}) \; / \; \mathsf{P}(\mathsf{s}) = \\ \mathsf{P}(\mathsf{s}_{1}, \, \dots, \, \mathsf{s}_{i}, \, \pi_{i} = \mathsf{k}) \; \mathsf{P}(\mathsf{s}_{i+1}, \, \dots \, \mathsf{s}_{n} \mid \pi_{i} = \mathsf{k}) \; / \; \mathsf{P}(\mathsf{s}) = \\ \mathsf{F}_{\mathsf{k}}(\mathsf{i}) \; \mathsf{B}_{\mathsf{k}}(\mathsf{i}) \; / \; \mathsf{P}(\mathsf{s}) \end{array}$$

Posterior Decoding now gives the most likely state at position *i* of sequence:

$$\pi^*_{i} = \operatorname{argmax}_k P(\pi_i = k \mid s)$$



Decoding problem

- We have now 2 methods for decoding:
 - Posterior decoding
 - Viterbi algorithm
- Which is most appropriate?



Posterior Decoding

For each state

- Posterior Decoding gives us a probability distribution for the state at each position
- This is sometimes more informative than Viterbi path π^*
 - Posterior decoding takes into account all possible paths when determining the most likely state
 - Viterbi method only takes into account one path, which may end up representing a minimal fraction of the total probability



Posterior Decoding



•
$$P(\pi_i = k \mid s) = \Sigma_{\pi} P(\pi \mid s) \mathbf{1}(\pi_i = k)$$

= $\Sigma_{\pi:\pi[i] = k} P(\pi \mid s)$

 $\mathbf{1}(\psi) = 1$, if ψ is true 0, otherwise



HMMs for sequence alignment



Hidden Markov Models for Sequence Alignment

- So far, we have used HMMs to detect certain regions from a single sequence
- HMMs can also be used for sequence alignment tasks
 - Pair-HMM can be used to find high-scoring alignments between two sequences, allowing gaps
 - Profile-HMM can be used to model a multiple alignment of a set of sequences



Pair HMM

- Given 2 sequences X and Y, we want to identify their alignment
- Pair HMM consists of
 - Begin and End state which do not emit symbols
 - Three normal states
 - M (match)
 - X (gap in Y)
 - Y (gap in X)



- X TAG-CTATCAC--GACCGC-GGTCGATTTGCCCGACC
- Y -AGGCTATCACCTGACCTCCAGGCCGA--TGCCC---

XMMYMMMMMMYYMMMMMMMMXXMMMMMXXX

Pair HMM - Transitions

- Transition from M to X (resp. Y) opens a gap in Y (resp. X),
- Transition back to M closes the gap
 - δ ~ open gap probability
 - $-\epsilon$ ~ extend gap probability



- X TAG-CTATCAC--GACCGC-GGTCGATTTGCCCGACC
- Y -AGGCTATCACCTGACCTCCAGGCCGA--TGCCC---XMMYMMMMMMYYMMMMMMMXXMMMMMXXX

Pair HMM - Emissions

- State M: emit (b,b') with probability E_M(b,b')
- State X: emit (b,-) with probability E_x(b,-)
- State Y: emit (-,b') with probability E_Y(-,b')



X TAG-CTATCAC--GACCGC-GGTCGATTTGCCCGACC

Y -AGGCTATCACCTGACCTCCAGGCCGA--TGCCC---XMMYMMMMMMYYMMMMMMMXXMMMMMXXX

Pair HMMs – Finding Optimal Alignment

- A state sequence π from begin to end state that emits x and y gives an alignment for them
 - Transition and emission probabilities give the probability of the alignment
- The best alignment of two sequences corresponds to the most probable state sequence

 $\pi^* = \operatorname{argmax}_{\pi} P(x, y, \pi)$

- Can be computed by the Viterbi algorithm
 - X TAG-CTATCAC--GACCGC-GGTCGATTTGCCCGACC
 - Y -AGGCTATCACCTGACCTCCAGGCCGA--TGCCC---XMMYMMMMMMYYMMMMMMMXXMMMMMXXX

Viterbi for pair-HMMs



X TAG-CTATCAC--GACCGC-GGTCGATTTGCCCGACC Y -AGGCTATCACCTGACCTCCAGGCCGA--TGCCC---XMMYMMMMMMYYMMMMMMMMXXMMMMMXXX

Full model

• The complete model should also contain the transitions between the begin, end and normal states





Pair-HMM vs. Needleman-Wunsch



- Similarities:
 - HMM transition to a match state ~ NW diagonal move
 - HMM tr. to Y state ~ NW horizontal move
 - HMM tr. to X state ~ NW vertical move
 - HMM Emissions ~ NW substitutions
- Important difference:
 - HMM transition and emission probabilities can be trained
 - NW substitution scores fixed

Profile Hidden Markov Models



From Sequence to Structure to Function

- In functional genomics the goal is to annotate the genes by their function (e.g. catalysis of a biochemical reaction)
- In principle, possible functions of proteins are determined by their 3D structure
- 3D structure is in principle determined by the amino acid sequence
- Consequently, the amino sequence should determine the function



From Sequence to Structure to Function

- However, predicting the 3D structure of a protein (aka Protein folding problem) from the amino acid sequence is extremely difficult
 - Not fully solved yet
- Also, predicting the function from the 3D structure is not easy
 - Require molecular simulations run on supercomputers
- A shortcut is offered by Hidden Markov Models



Profile Hidden Markov Models

- Protein families:
 - Sets of related sequences and structures
 - Diverged from each other in their primary sequence during evolution
 - Some regions are more conserved than others
- Profile HMM is tailored to the family
 - by defining the HMM structure to match the family
 - by training the parameters with the sequences of that family

	1	[:		
LAC11_ARATH/31-147	100.0%	DVQVKNI	SRICNA	P I VTVHGI	TPGP VIA	REGURVII	NVTHEV	QXN-	MSIHWHG	LKQYRNGWAD
2 LAC2_ORYSJ/34-150	76.1%	DIVMSNV	RLCHE	AM <mark>VIVN</mark> GS	SY <mark>PGP<mark>H</mark>IYA</mark>	REGURVIV	NVTHEV	KEN-	MITINWHG	LKORRHGWAD
09AUIO_PINTA/32-148	70.9%	DIKENV	TRLCHT	EPIVIANG	FPGPIIHA	REGDIVITY	KVINHV	T <mark>YN</mark> -	-VS IHWHG	IROLRIGWAD
4 Q9ZQW3_POPTR/30-146	65.8%	RVVLTNT	TKL <mark>C</mark> SS	SIPAING	(FPGP <mark>H</mark> IYA	REGUNVNI	RLINQV	QXN-	-VTVHWHG	VSSCET <mark>GWAD</mark>
5 LAC10_ARATH/30-146	70.1%	NVTXQV	T <mark>RIC</mark> ST	O IVIVIG)	FPGPTIYA	NEDDTILV	NVVNNV	K <mark>YN</mark> -	-VS I HWHG	IRULRTGWAD
5 LAC4_ARATH/32-148	70.9%	N <mark>V</mark> VM <mark>KN</mark> V	T <mark>RLC</mark> SS	PTVTVNG J	RYPGP II IYA	REDUTILI	KVVNHV	K <mark>YN</mark> -	-V <mark>S</mark> IHWHG	WROWRIGWAD
7 LAC4_ORYSJ/36-152	71.8%	N <mark>VO</mark> MANA	TRLCNT	SMVIVNC	(C <mark>PGP</mark> ELV <mark>A</mark>	REGURVVI	RVINNV	AK <mark>N</mark> -	ISLHWHC	WROWRIGWAD
08H6A0_LOLPR/37-153	69.2%	N <mark>VO</mark> MB <mark>N</mark> V	T <mark>RLC</mark> AT	SIVIVNGI	IY <mark>PGP</mark> ALV <mark>A</mark>	RKCHRVLV	RVINHV	AK <mark>N</mark> -	MTLHWHG	IROLRSGWAD
09AUI4_PINTA/36-152	66.7%	HIRLNNI	T <mark>RLC</mark> HT	PLITYNG	Y <mark>PGP</mark> AII <mark>A</mark>	REGURVVI	NVTNHV	KD <mark>N</mark> -	-VT IHWHG	IR <mark>QIR</mark> SAWAD
0 09AUI1_PINTA/41-156	65.0%	NVRL	TRLCHS	PLVIVNGJ	RY <mark>PGP</mark> IF <mark>A</mark>	R-GDRVII	KLVNHV	KD <mark>N</mark> -	-VT IHWHG	WROLRSGWAD
Q9AUI6_PINTA/39-155	66.7%	H <mark>W</mark> RM <mark>KN</mark> V	T <mark>RLC</mark> HT	PLITVNC)	(S <mark>PGP</mark> KIVV	REDURVII	KVHNHV	KD <mark>N</mark> -	-V <mark>S</mark> I HWHC	IROLRSGWAD
2 LAC2_ARATH/35-151	68.4%	DIQLENI	T <mark>RLC</mark> KT	TIVTVNG	FPGPRVTA	REGINLOI	k vv nev	SN <mark>N</mark> -	- I <mark>S</mark> I HWHG	IROLRSGWAD
0920W2_POPTR/41-157	60.7%	N I TH <mark>KN</mark> F	T <mark>RLC</mark> HT	RSL <mark>VI</mark> VNGO	FPGPRLVA	RECDOVLV	KVVNHV	AIN-	ITIHWHG	VROLTTGWAD
1 LAC17_ARATH/30-146	62.4%	E IKMQNV	T <mark>RLC</mark> HT	SLVSVNG	FPGPKLIA	REGDOVLI	K V V NQ V	PN <mark>N</mark> -	ISLHWHG	IROLRSGWAD
5 024042_LIRTU/42-158	65.0%	DIRLONV	TRLCHT	SIVIVNGO)FPGPKIV <mark>A</mark>	RECORVER	KVVNHV	<u>0</u> 48-	- ITLHWHG	VROLRSGWAD
0.0 40 41 T TRUTT 100 1 40	10.00	COLUMN TAXAB	A Dr. Caller	No. or contract	THE R. LAW ST.	No. 1 No. 2 No.			A DESCRIPTION OF A DESC	an Stor of Louis a



Profile Hidden Markov Models – the approach

- We construct a HMM of a set of proteins that share a function or structural regions (called *domains*)
- This model can be used to give a *probability* for each new protein sequence to share that same function or domain
- The same sequence can be tested against a large set of HMM models
 - high probability by a HMM indicates that our new sequence may share the domain or function modeled by that HMM

1 IAC11 ARATH/31-147	100.0%	
2 LAC2 0RYSJ/34-150	76.1%	TEXTS NO STREATE AND A DESCRIPTION OF THE PROPERTY OF A DESCRIPTION OF THE PROPERTY OF THE PRO
3 09AUIO PINTA/32-148	70.9%	TKE SHOTSLCHT
4 09Z0W3 POPTR/30-146	65.8%	PWVLTHINKLCSS
5 LAC10 ARATH/30-146	70.1%	NWVTXOVTRICSTXOIVTVNGKTPGPTIVANEDDPILVNVVNVVKYN-VSIHWHGIRDIRTGWADG
6 LAC4 ARATH/32-148	70.9%	NWVMXNVTRICSSXPTVTVNCRYPCPTIYAREDDTLLIXYVHEVKYN-VSIEWHEVROVROVROVROVRO
7 LAC4_ORYSJ/36-152	71.8%	NYQYANATRICNTXSYYTYNGQCPGPELVAREGDRYYIRYYAKN-ISILWHGVRQVRDWRDG
8 Q8H6A0 LOLPR/37-153	69.2%	NVOMANVTRLCATXSIVIVNCEYPCPALVARXCHOVLVRVINHVAHN-MTLWHGIROLRSGWADC
9 09AUI4_PINTA/36-152	66.7%	HIRLANITRLCHTXELITVNGEYPGPAIIARECDRYVINKYKDN-VTHWHGIRQIRSAWADC
10 Q9AUI1_PINTA/41-156	65.0%	NWRL <mark>XNVTRLC</mark> HS <mark>XRLVTVNCRYPCP</mark> TIF AR-CDRVII KLVNHVKDN-VTIHWHCVRULRSGWADD
11 Q9AUI6_PINTA/39-155	66.7%	HWRMKNVTRLCHTXPLITVNCKSPGPKIVVRZDDRVIIKWKHKVKDN-VSIHWHGIRDLRSGWADC
12 LAC2_ARATH/35-151	68.4%	DIĞLÄNITALCKTÄTIVTVNCKIPGPARTALCONLOIKVVNKVSNN-ISIHWHGIRĞLASSWADI
13 Q9ZQW2_POPTR/41-157	60.7%	NITK <mark>XNFTRLC</mark> HTRSL <mark>VTVNG</mark> QF PGP RLV <mark>AREGD</mark> QVLVK V VHKVAE <mark>N</mark> -IT THWHG VRQLTTSWADC
14 LAC17_ARATH/30-146	62.4%	IIKMQNVTQLCHTXIXSVNCQTPCPKLIANICDQVLIKVVNQVPNN-ISLIWICIRQLZSCWADC
15 024042_LIRTU/42-158	65.0%	DIRLONVIRLCHTXSIVIVIGOFDEPKIVARCEDEV/V/KVVHVONN-IITLHMHGVROLRSEWADC
16 024041 110701/32_148	AR 98	מהבעצולה דולים 1985–1997 ב- 1997 ב- 1997 אליליאל יראלאליה פרעינים אינטיפאליאנים דוואלי 1977 מליא אליאראלי ד
		$\begin{array}{c} T(d_2)d_1 \\ (d_2) \\ (d_3) \\ (d_4) \\ (d_$

 $m_2 T_{(m_3 m_2)}$

Figure 1. The model.

 m_3

Tilma

 m_5

m₄

1 [



 m_0

m₁

Example: zinc finger domain

- Typically it functions as interaction module that binds DNA, RNA, proteins, or other small, useful molecules
- Several variants exist, one of which is depicted above right
- Below right a protein with three zinc finger domains embedded







Example: zinc finger domain

- Part of the multiple alignment of • proteins containing the zinc finger domain is depicted below
- The full alignment has 194 proteins ۲
- A profile HMM can be trained to • recognize new members of the family
 - Does not require 3D structure
- PFAM database contains a large number of profile HMMs for different structural and functional domains or motifs



. ...

			1	L		•			•		• 1	32
1	SRYC_DROME/358-380	100.0%		YQC	D1	CG	QK	FVQ <mark>k</mark>	INLT	HHAR	I	
2	INSM1_HUMAN/441-464	33.3%		HLC	PV	CG	-ES	ASK	GAQE	RHLR	LL	
3	XFIN_XENLA/1276-1298	26.1%		YG <mark>C</mark>	NC	CD	-RS	STH	SASV	RHQR	MC	
4	XFIN_XENLA/1044-1066	43.5%		Y K <mark>C</mark>	GL	CE	-RS	PVB <mark>k</mark>	SALS	RHQR	VB	
5	ZNF76_HUMAN/285-309	40.0%		YTC	PE-PH	CG	RG	TSA	TNYK	NHVR	IB	
6	CF2_DROME/401-423	47.8%		YTC	SY	CG	KS	FTQSI	NTLK	QHTR	I	
7	IKZF1_MOUSE/144-166	47.8%		FQ <mark>C</mark>	NQ	CG	AS	FTQK	GNLL	RHIK	L	
8	EVI1_HUMAN/131-154	37.5%		YB <mark>C</mark>	EN	I <mark>C</mark> A	KV	TDP	SNLQ	RHIR	SQ <mark>E</mark>	
9	TRA1_CAEEL/337-362	23.1%		¥S <mark>C</mark>	QI-PQ	CT	-KS	YTDP	SS <mark>L</mark> R	KHIK	AV	
10	SUHW_DROAN/349-373	32.0%		YA <mark>C</mark>	KI	CG	KD	TRS	Y H <mark>L</mark> K	RHQK	YS-SC	
11	EGR1_HUMAN/396-418	43.5%		FA <mark>C</mark>	DI	CG	R <mark>K</mark>	ARSI	DERK	RHTK	I	
12	ADR1_YEAST/104-126	30.4%		FVC	EV	CT	RA	ARQI	eh <mark>l</mark> k	RHYR	SB	
13	SDC1_CAEEL/268-290	47.8%		YF <mark>c</mark>	HI	CG	TV	FIEQI	DNLF	KHWR	LB	
14	SDC1_CAEEL/145-168	33.3%		Y M C	QV	CL	-TL	GHT	YNLF	MHWR	TSC	
15	KRUH_DROME/299-321	43.5%		FE <mark>C</mark>	EF	CH	KL	r sv k i	BNLQ	VHRR	I	
16	TTKB DROME/538-561	41.7%		YPC	PF	CF	KE	TRK	DNMT	AHVK	II	



Structure of a Profile HMM



Figure 1. The model.



Match states

- Correspond to the columns of the multiple alignment
- Number of match states picked using expert knowledge e.g.
 - average length of sequences in the alignment
 - number of columns that contain at least 50% non-gap symbols
- Initial emission probabilities can be computed from the multiple alignment:
 - For each amino acid, count the times it appears in each column

			1 [
1	LAC11_ARATH/31-147	100.0%	DVOVKNISTICNAXPIVIVNEMEPEPIVYAREEDDRVIINVINEWQYN-MSIEWEELKQYDNEW	A D G
2	LAC2_ORY3J/34-150	76.1%	DIVYSNVSRICHEXAMVIVNCSYPCPTIYARESDRVIVNVINHVKHN-MTIHWHGIKORRAGW	A D G
з	Q9AUIO_PINTA/32-148	70.9%	DIKE KNYTYLCHTEPIVTANGKEPGPTIHAREGDTVTYKVTNHYTYN-VSIHWHGIRDTVAW	A D G
4	Q9ZQW3_P0PTR/30-146	65.8%	RVUTHTIKLCSSXSIPTINGKIPGPTIYAREGDNVNIRITNOVOYN-VTVHUKGVSSCITGW	A D G
5	LAC10_ARATH/30-146	70.1%	NYVTXQVTXICSTXQIVXVKKTPGPTIYANEDDTIYYYYNYKYN-VSIHWHGIRDDTGW	A D G
6	LAC4_ARATH/32-148	70.9%	HWWWKNYTYLCSSYPTYTYNGRYPGPTIYAREDDULLIKYYNEYKYN-VSTEWEGVRDYRDW	A D G
7	LAC4_0RY3J/36-152	71.8%	NVOYANATRICNTXSMVIVACOCPOPELVARIOTRVIRVINVAHN-ISIIAWGVROVRIVW	ADC.
8	Q8H6A0_LOLPR/37-153	69.2%	NVOMANYTYLCATXSIVIVNGEYPGPALVARKGERVLVRVINEVAEH-MILEWEGIRODRSOW	A D G
9	Q9AUI4_PINTA/36-152	66.7%	HIRLM NITRLCHTXPLINVNCIYPOPAIIARECDRYVINVNHVKDN-VTIHWHGIRQIR	ADC.
0	Q9AUI1_PINTA/41-156	65.0%	NVRLXNVTRLCHSXPLVTVNCRYPGPTIFAP-GDRVIIKLVNHVKDN-VTIHWHGVRQLRSGW	ADC.
1	Q9AUI6_PINTA/39-155	66.7%	HVRM <mark>KNYTRLC</mark> HT <mark>XPLIZVNC</mark> KSPGPKIVVREDDRVIIKVHHHVKDN-VSIHWHCIRDDRSOW	A D G
2	LAC2_ARATH/35-151	68.4%	DIQLXHITRLCKTXTIVLVNGKTPGPRVIARECDNLQIKVVNHVSNN-ISIHWHGIRQLZSGW	ADC.
з	Q9ZQW2_P0PTR/41-157	60.7%	NITRXNFTRLCHTRSLVTVNCQFPGPRLVAREGQVLVKVVNHVAEN-ITTHWHGVRQLTTGW	ADC.
4	LAC17_ARATH/30-146	62.4%	EIKMQNYTQLCHTXSLVSVNGQEPGPKLIAREGDQVLIKVVNQVPHN-ISLHWHGIRQUQSOW	A D G
5	024042_LIRTU/42-158	65.0%	DIRLONVINCHT ISIVIVICOPERKIVAREDIAV/XXVIIIV 041-ITLAWICVRODASE//	A D G
4	024041 T TERT/22-148	62.08		and.



Insertion and deletion states

- Insertion states allow the profile HMM to model symbols in the sequences that do not match the model
 - aligning a symbol in sequence to a gap in the model
- Deletion states allow the profile HMM to model symbols deleted from the sequence
 - aligning a gap in a sequence to a symbol in the model



Summary: Building Profile HMM topology

(a) Sequence Alignment





(C)