CS 364 Computational Biology

Sara Mathieson

Haverford College

Slides modified from Iain Mathieson

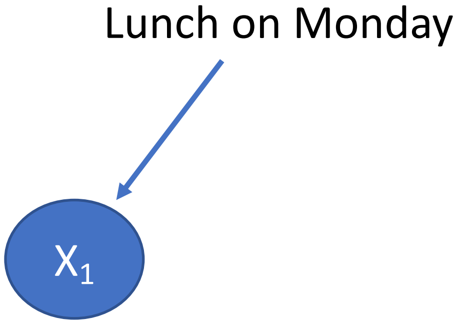
Outline

- HMM example in population genetics
- Recap Viterbi Algorithm
- Forward-Backward Algorithm
- Posterior Decoding

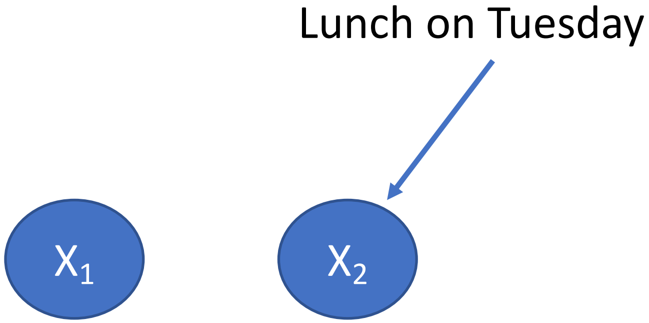
Notes:

- Today and Tuesday: HMMs
- Next Thursday: review
- Please fill out exam/generative AI poll!

Hidden Markov models



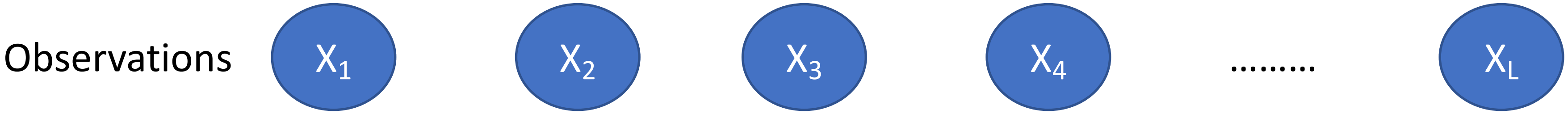
Lunch on Monday



Lunch on Tuesday

X_1

X_2



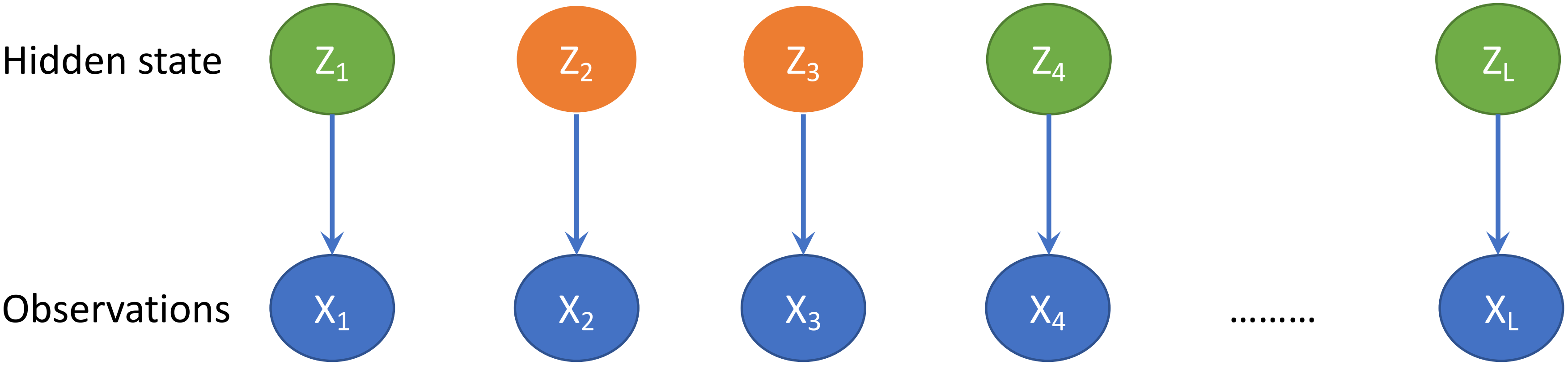
Observations

X_1 X_2 X_3 X_4 ……… X_L

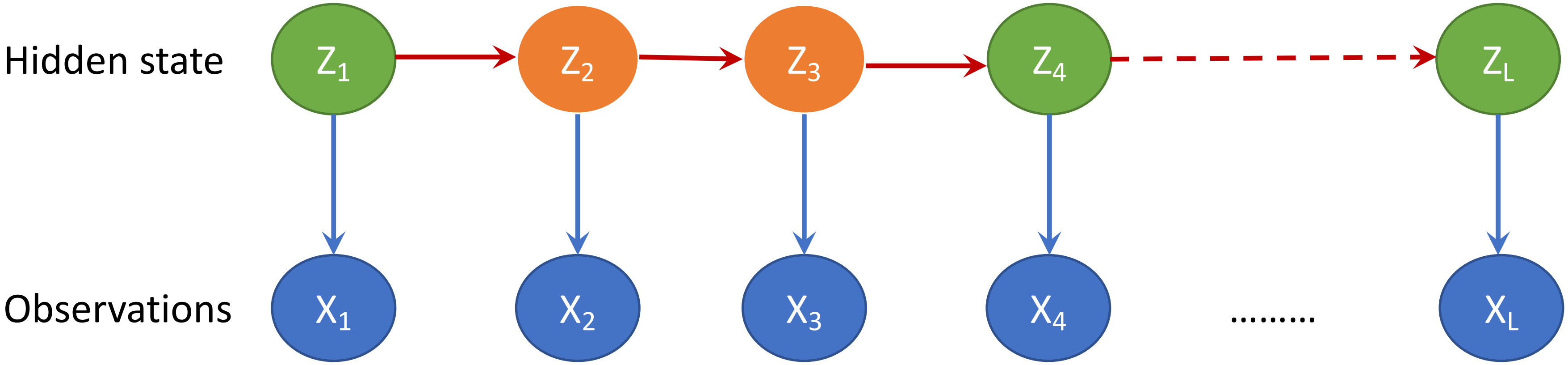
Mood: “Busy” or “Relaxed”

Hidden state Z_1

Observations X_1 X_2 X_3 X_4 ……… X_L


Hidden state
Z_1
Z_2
Z_3
Z_4
Z_L
Observations
X_1
X_2
X_3
X_4
....
X_L

Markov property: hidden state i depends on state i-1 but *only* on state i-1



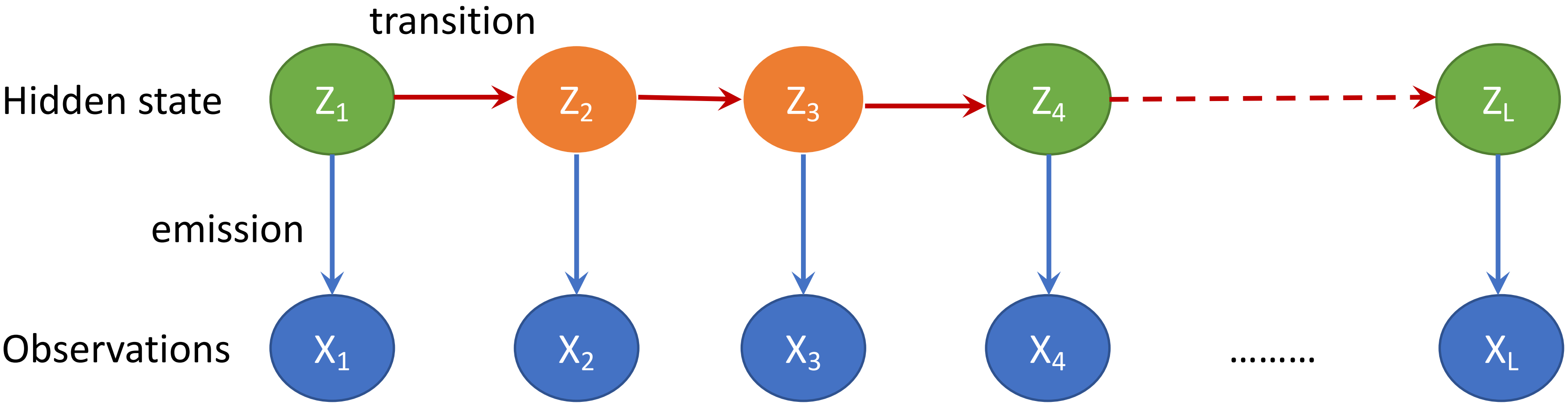
Hidden state

Z_1 Z_2 Z_3 Z_4 Z_L

Observations

X_1 X_2 X_3 X_4 ……… X_L

Markov property: hidden state i depends on state i-1 but *only* on state i-1



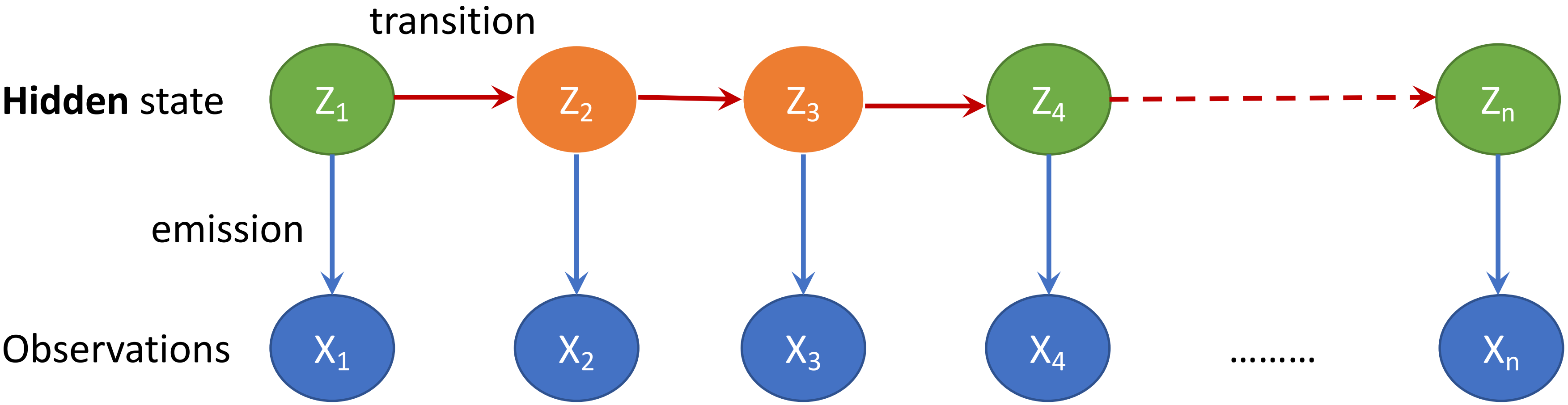
Hidden Markov model:

We have a series of **observations** that depend on some underlying **hidden state**.

Parameters: for each value of the hidden state, what's the probability of each possible observations [i.e. what is $P(X_i|Z_i)$?]. This is the **emission probability** ⟶

If we are in hidden state Z_i, what is the probability that we are in each of the possible hidden states in Z_{i+1}. [i.e. what is $P(Z_{i+1}|Z_i)$?]. This is the **transition probability** ⟶

Markov property: hidden state i depends on state i-1 but *only* on state i-1

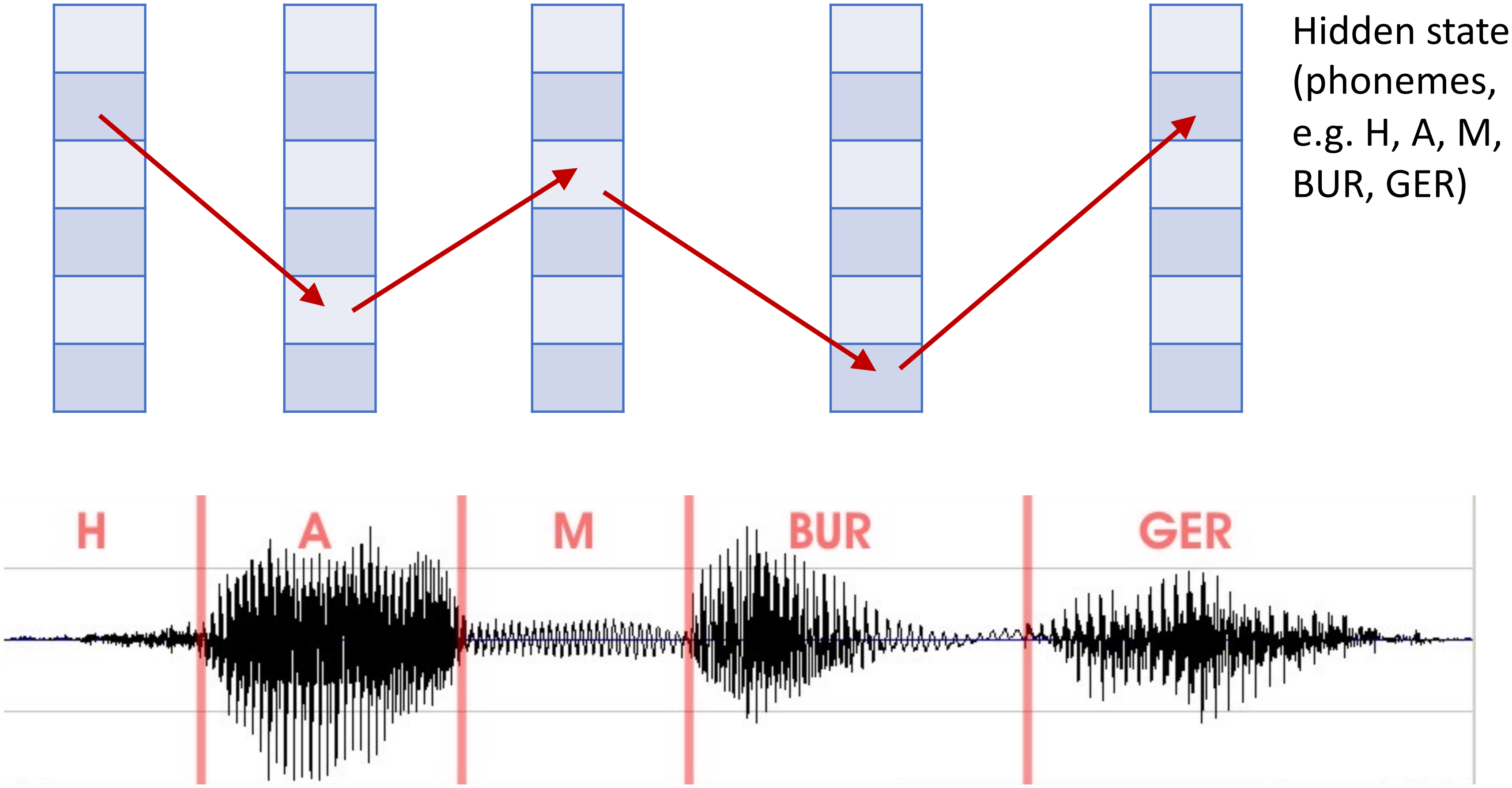


What can we learn

1) Given the observations, what were the hidden states? i.e. given the record of what I ate for lunch over a week, what was my mood on each day?

2) What are the parameters. e.g. if I am busy one day, what is the probability that I am relaxed the next day (transition)? If I am relaxed today, what is the probability that I eat out (emission)?

Example: Speech recognition



Hidden states (phonemes, e.g. H, A, M, BUR, GER)

Observations (waveforms)

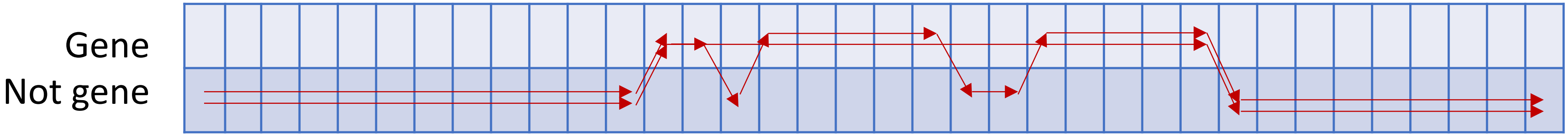
https://www.mm.cs.stonybrook.edu/hci/speech/speech.html

Example: sequences (gene finding)

Where is the gene?

Coding sequences have higher CG content than noncoding regions

ACATCAGCTACGATCGATGCGGGCATGAGTCCCATATATATTTAG



Transition probabilities

	Gene	Not
Gene	0.9	0.1
Not	0.05	0.95

(e.g. average gene length is 10 and 1/20 of the genome is genic)

Emission probabilities

	A	C	G	T
Gene	0.25	0.25	0.25	0.25
Not	0.35	0.15	0.15	0.35

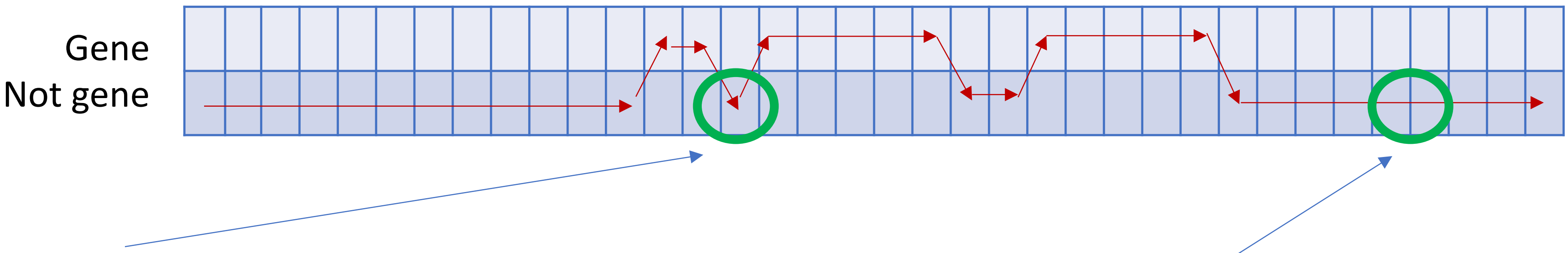
(e.g. GC content is 50% in genes and 30% outside)

Example: sequences (gene finding)

Where is the gene?

Coding sequences have higher CG content than noncoding regions

ACATCAGCTACGATCGATGCGGGCATGAGTCCCATATATTTAG



The most likely path says that this nucleotide is not in a gene

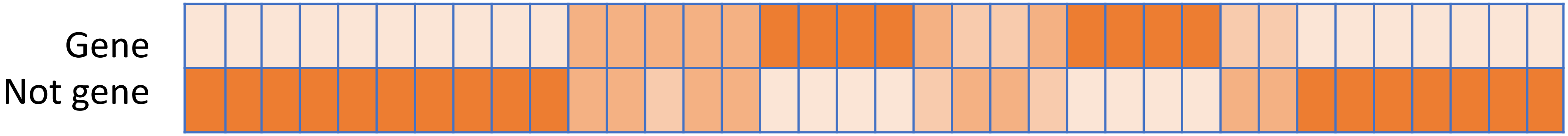
But surely it is more likely to be in a gene than this one?

Example: sequences (gene finding)

Where is the gene?

Coding sequences have higher CG content than noncoding regions

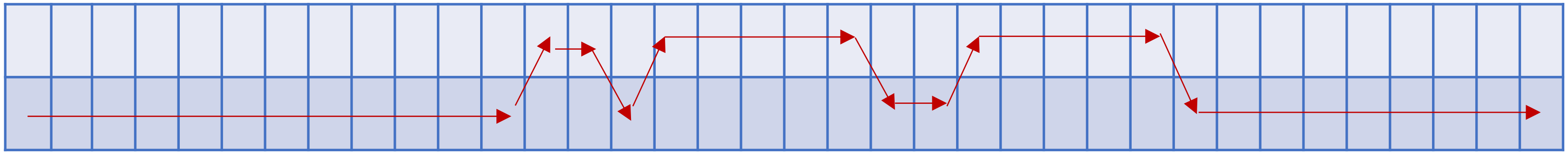
ACATCAGCTACGATCGATGCGGGCATGAGTCCCATATATATTTAG



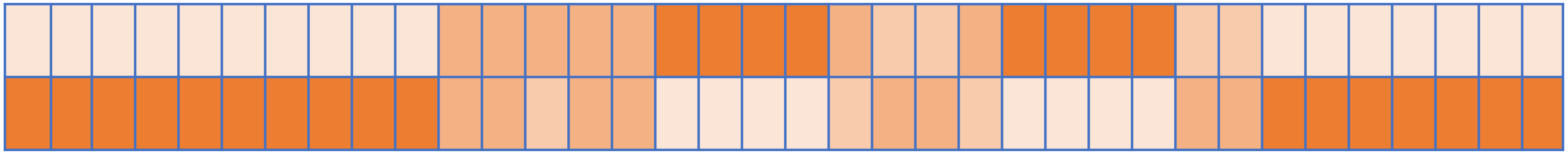
Find probability of each hidden state, rather than most likely path

Two ways of learning the hidden states

a) Find the most likely path: Viterbi algorithm



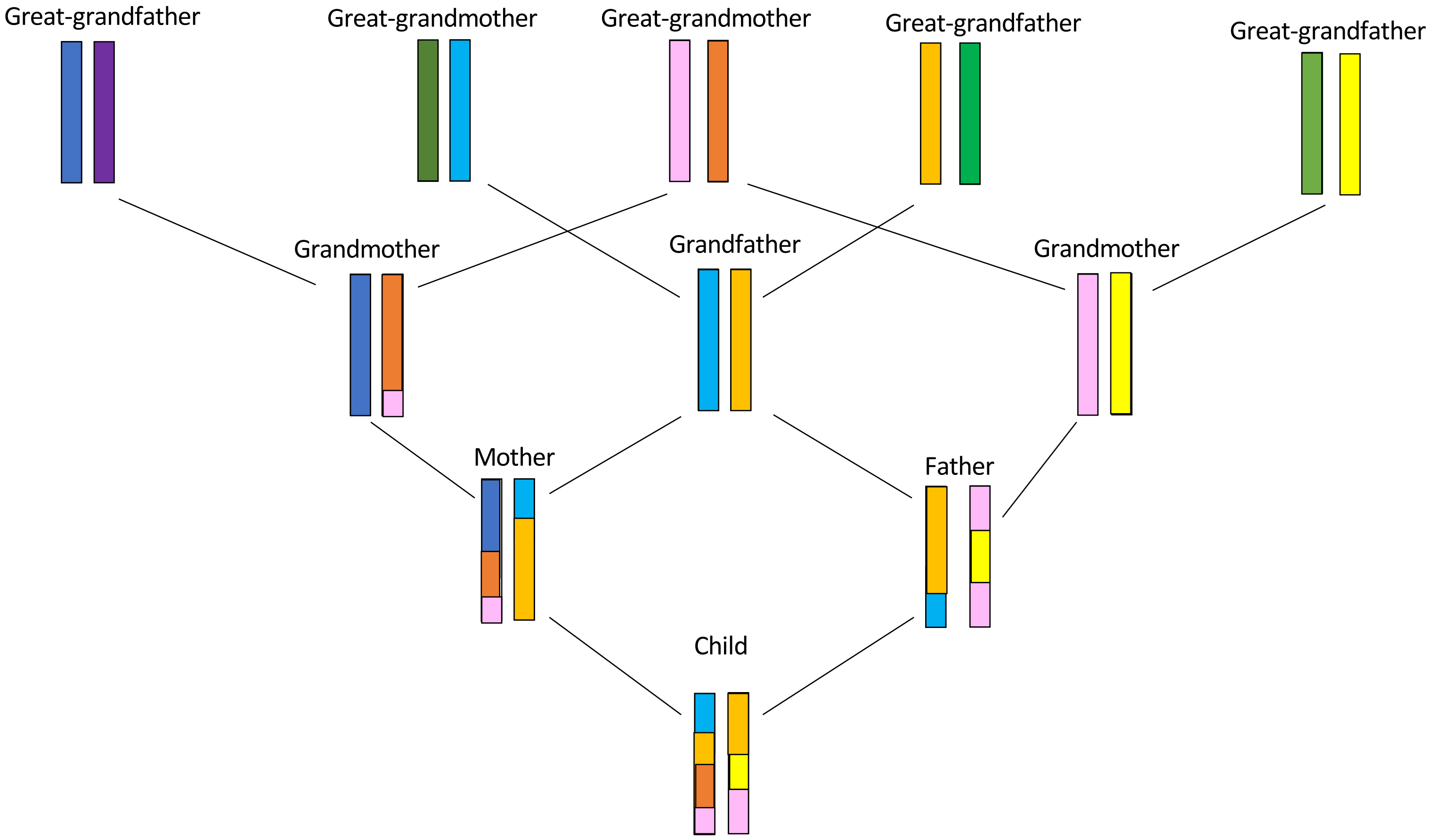
b) Find posterior probability of each state: Forward-backward algorithm



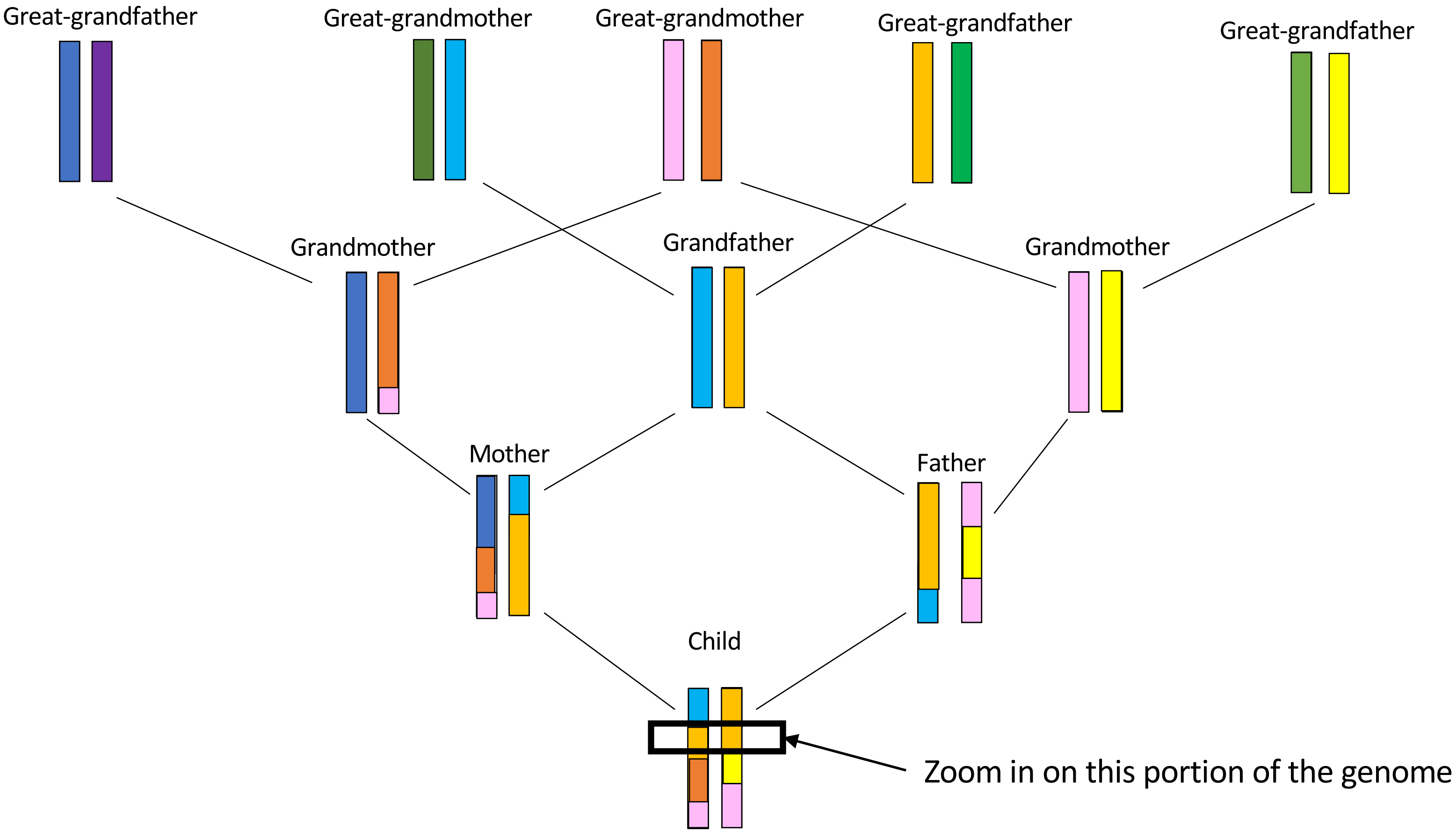
Note that the Viterbi path is NOT the same as taking the maximum cells from the Forward-Backward algorithm (“posterior decoding”). The Viterbi path must be a path that is actually possible, but the maximum posterior might not be (e.g. because it implies impossible transitions).

HMM example from population genetics

Recombination over time

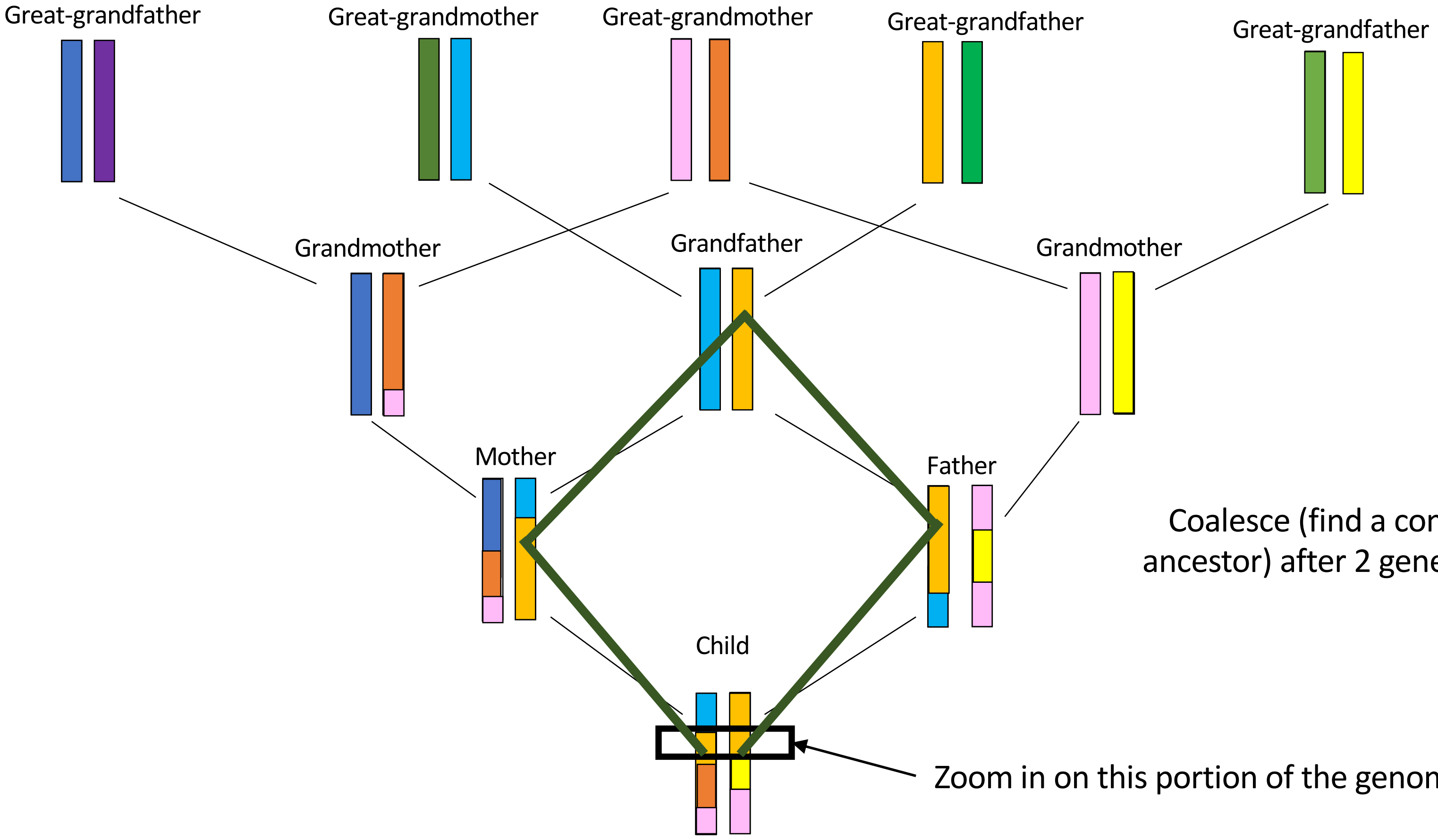


Recombination over time



Zoom in on this portion of the genome

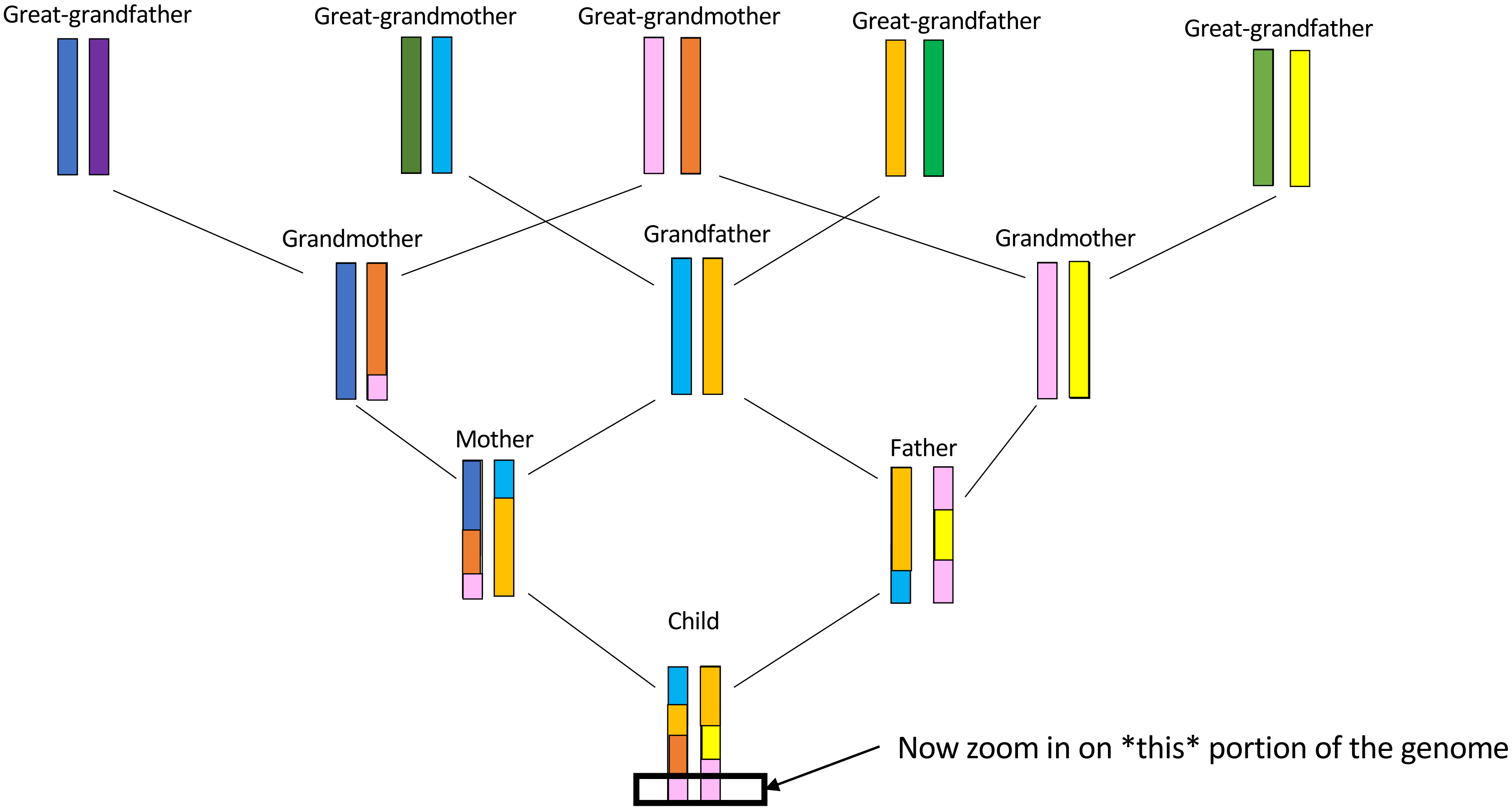
Recombination over time



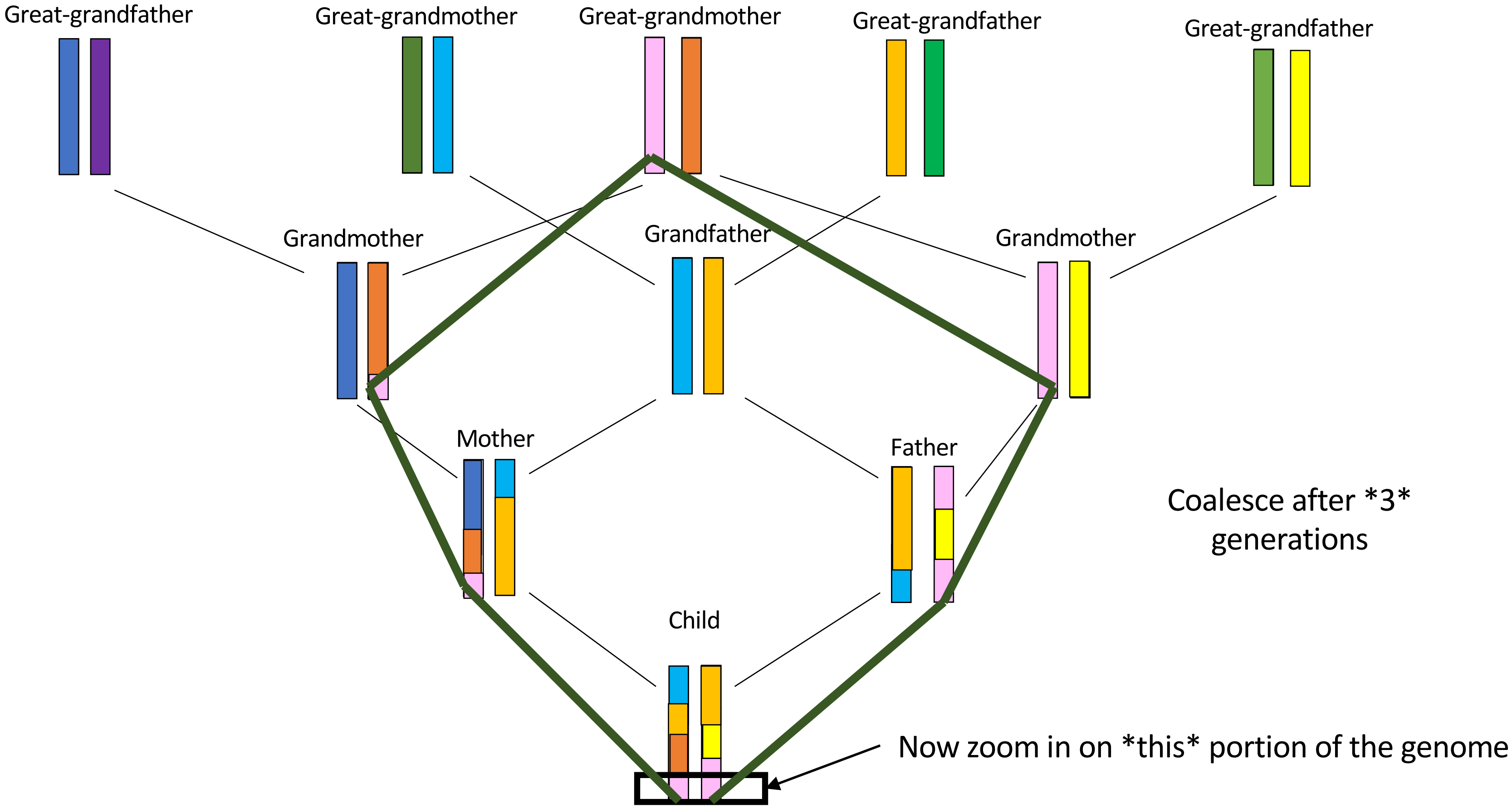
Coalesce (find a common ancestor) after 2 generations

Zoom in on this portion of the genome

Recombination over time



Recombination over time



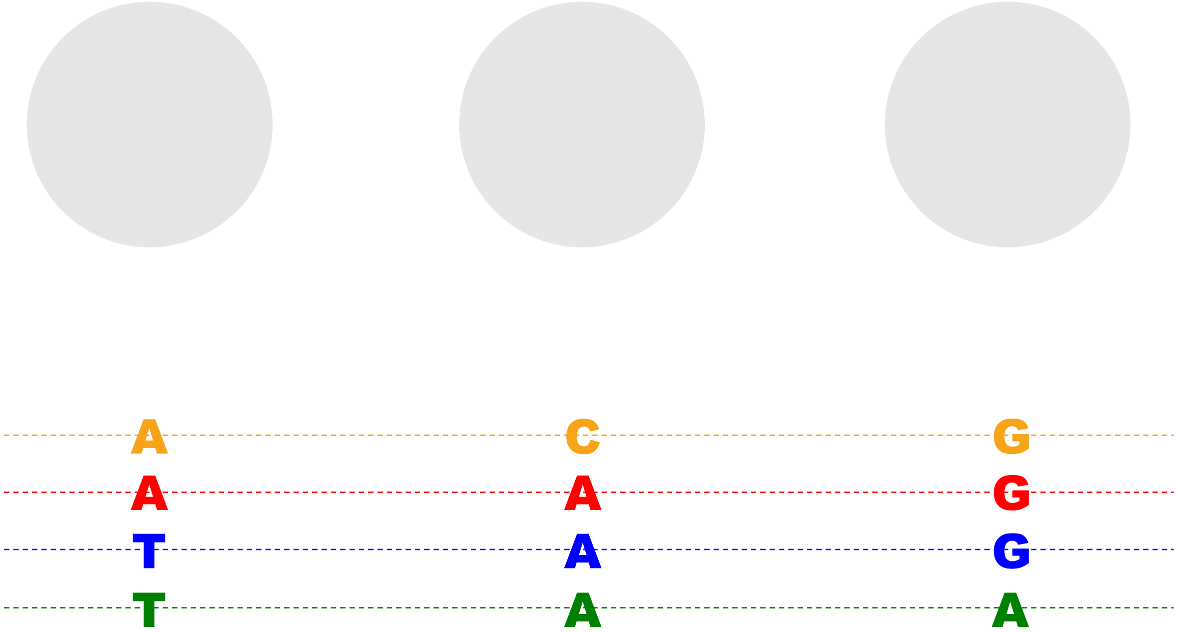
Coalesce after *3* generations

Now zoom in on *this* portion of the genome

How could we encode this as an HMM?

- Take-home message: the tree changes across the genome! Both topology (for n > 2) and branch lengths

Sequence data at many sites



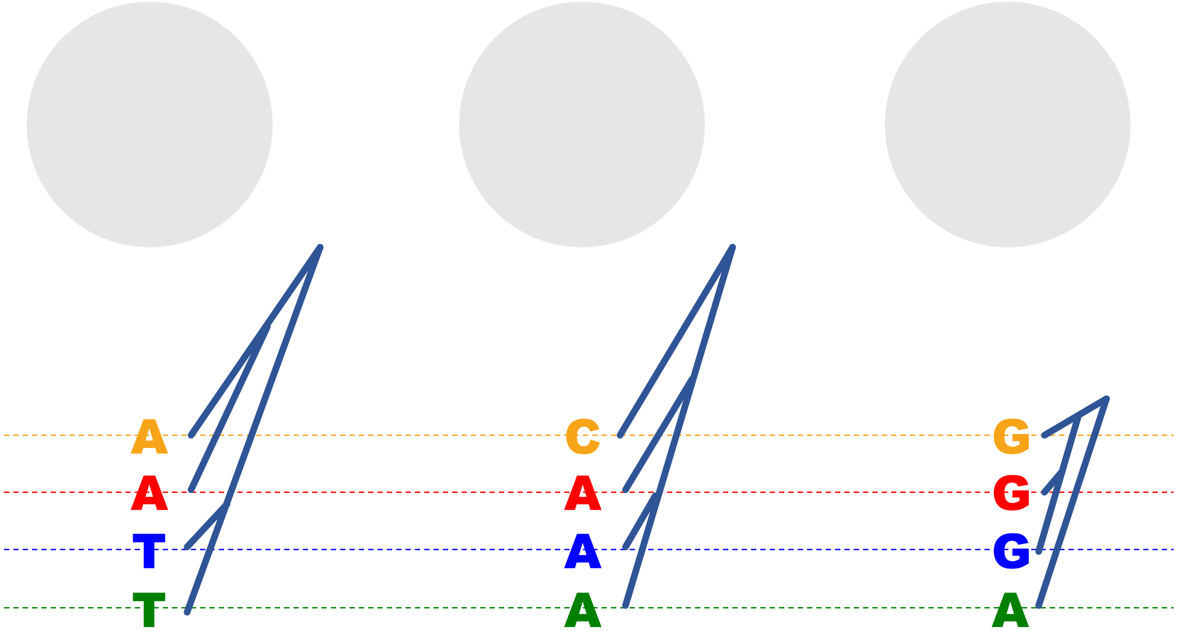
A C G

A A G

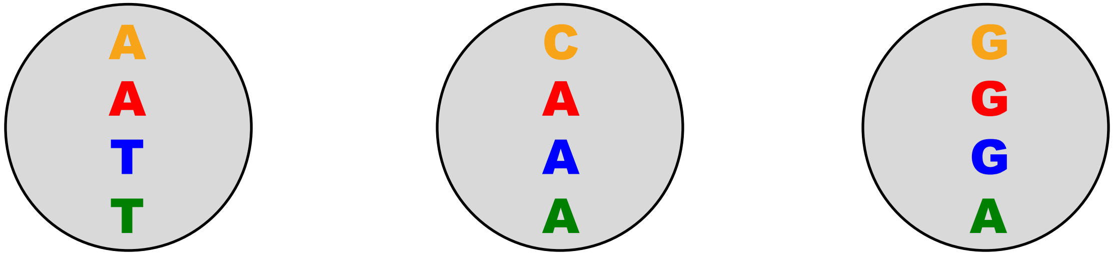
T A G

T A A

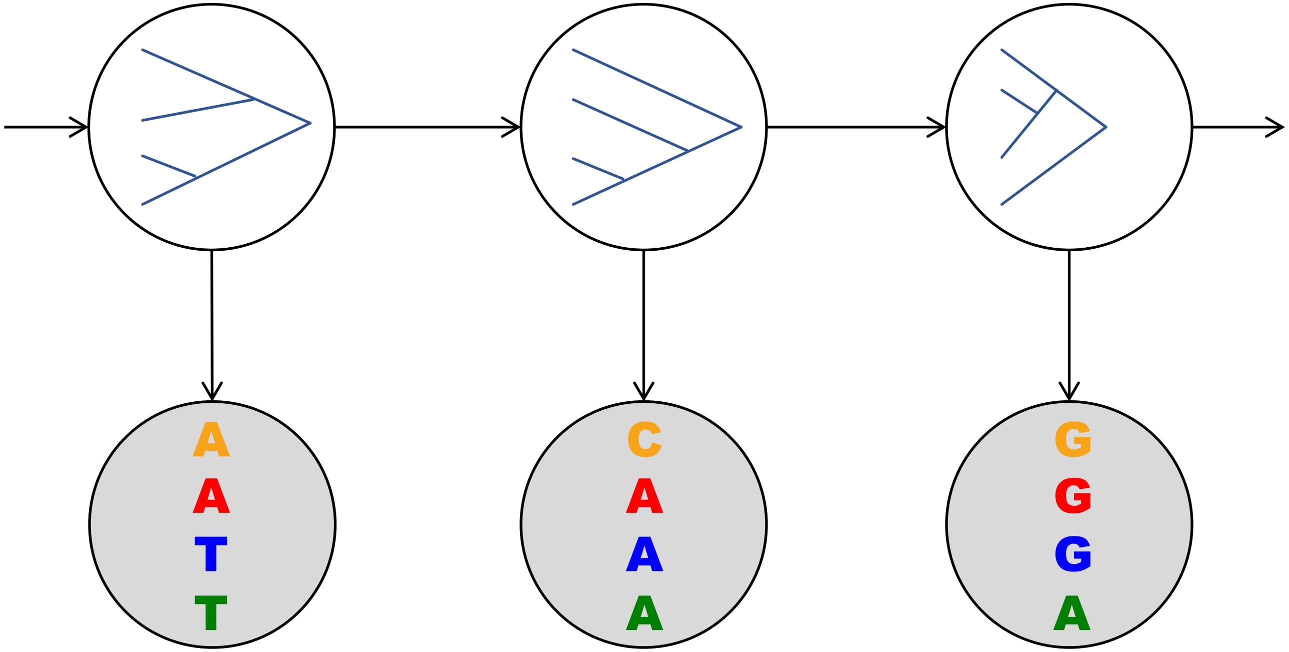
Tree changes along the genome!



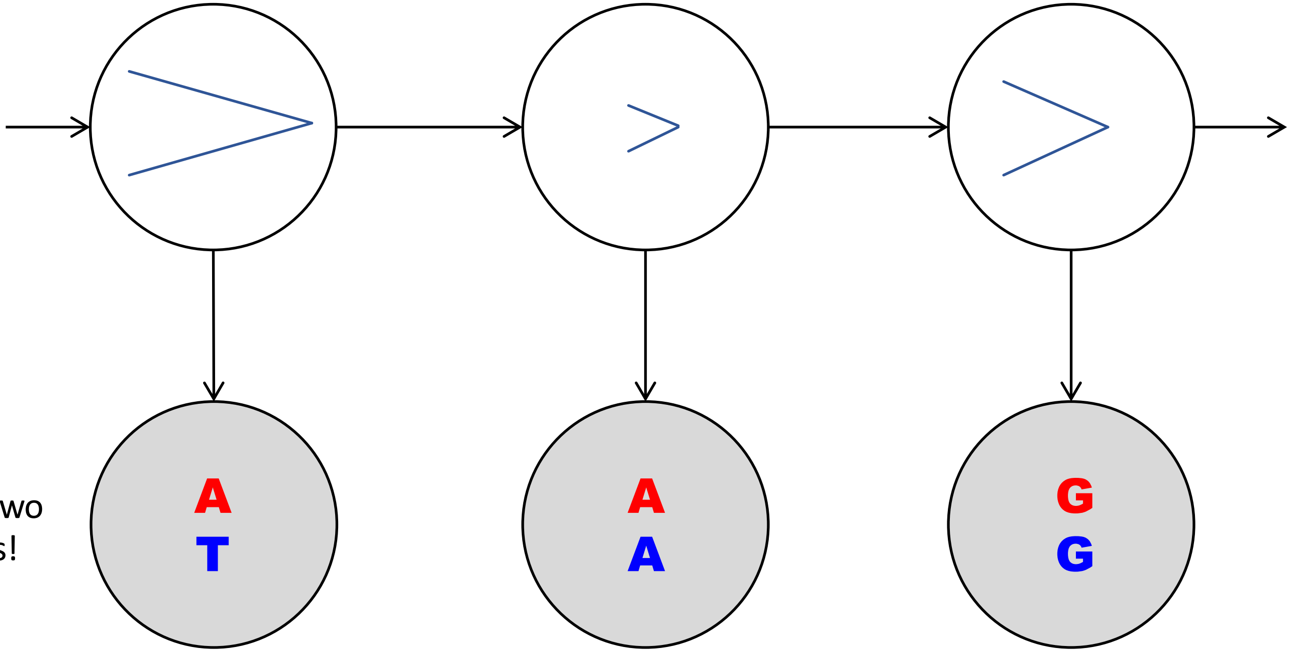
HMM observations: sequence data



HMM hidden states: the tree



Number of possible trees grows exponentially…



One person, two chromosomes!

Now the hidden state becomes the *time* of coalescence

PSMC: pairwise sequentially Markovian coalescent

- The distribution of pairwise coalescence times should be **exponential** with parameter 1

- If this differs from the exponential distribution, there were probably **population size changes**

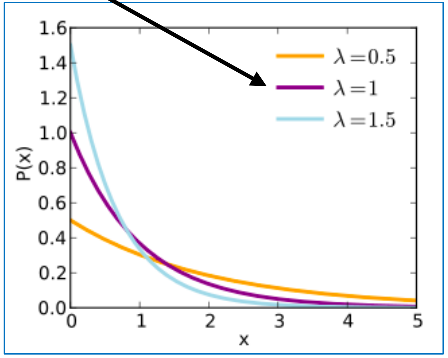
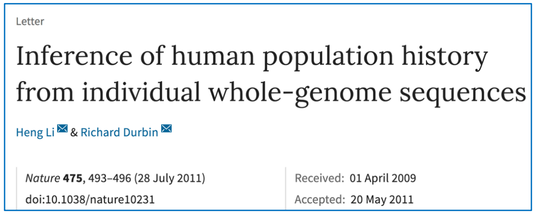


Image: wikipedia

- If all coalescence times are very recent, small population size

- If all coalescence times are very ancient, large population size



Letter

Inference of human population history from individual whole-genome sequences

Heng Li & Richard Durbin

Nature **475**, 493–496 (28 July 2011)

doi:10.1038/nature10231

Received: 01 April 2009

Accepted: 20 May 2011

PSMC: an HMM for two sequences

Letter

Inference of human population history from individual whole-genome sequences

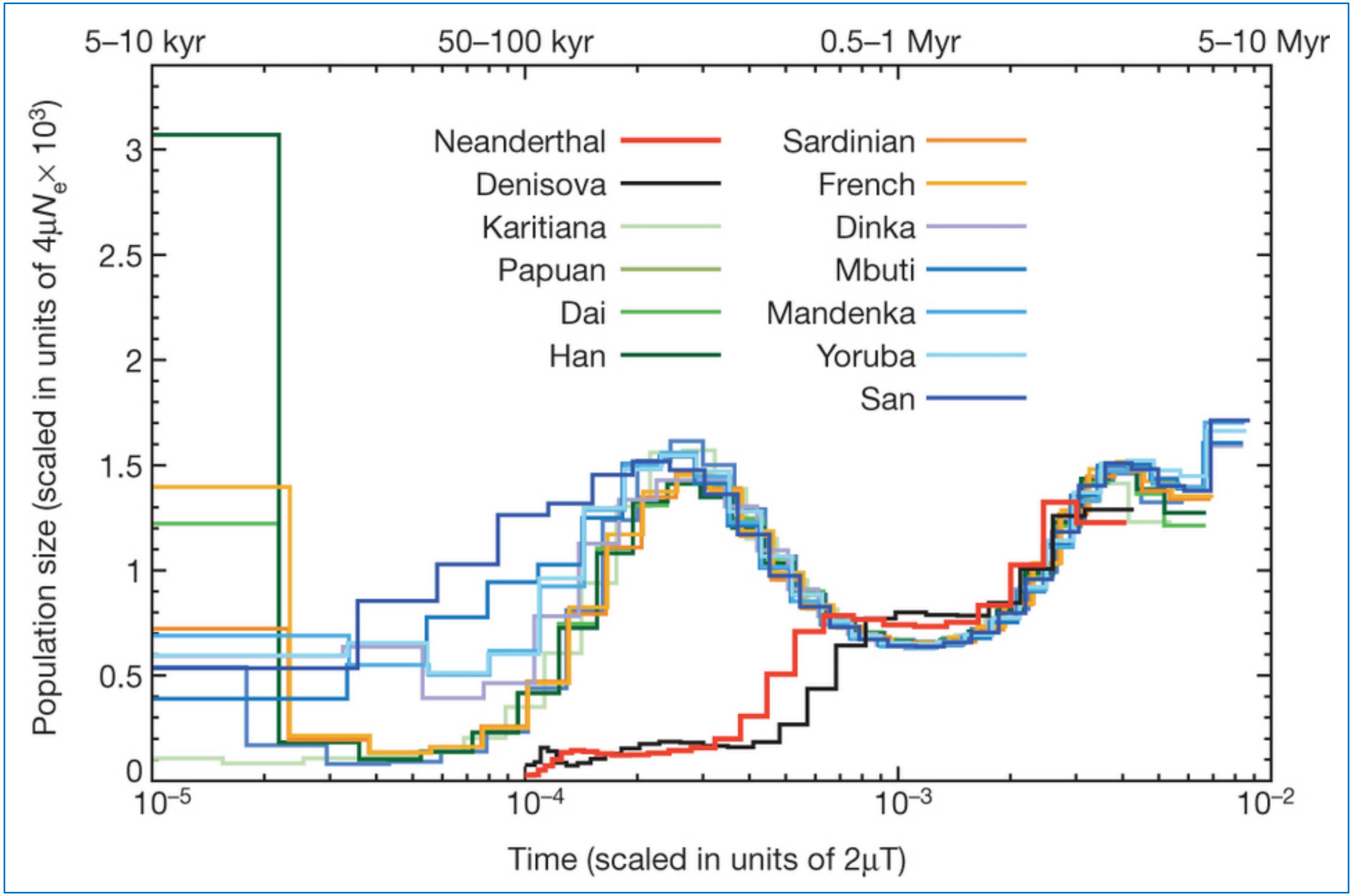
Heng Li & Richard Durbin

Nature **475**, 493–496 (28 July 2011)

doi:10.1038/nature10231

Received: 01 April 2009

Accepted: 20 May 2011



"The complete genome sequence of a Neanderthal from the Altai Mountains", Prufer et al (2014)

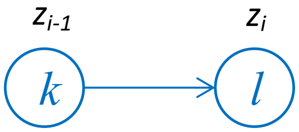
HMM definition

K = num hidden states, *B* = num emitted states

- Transition probabilities:

(*K* x *K* matrix)

$$\boxed{a_{kl} = P(z_i = l | z_{i-1} = k)}$$



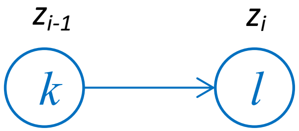
HMM definition

K = num hidden states, *B* = num emitted states

- Transition probabilities:

(*K* x *K* matrix)

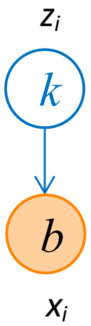
$$\boxed{a_{kl} = P(z_i = l | z_{i-1} = k)}$$



- Emission probabilities:

(*K* x *B* matrix)

$$\boxed{e_k(b) = P(x_i = b | z_i = k)}$$



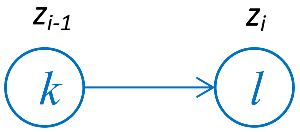
HMM definition

K = num hidden states, *B* = num emitted states

- Transition probabilities:

(*K* x *K* matrix)

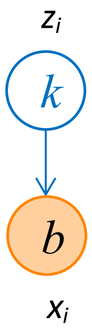
$$a_{kl} = P(z_i = l | z_{i-1} = k)$$



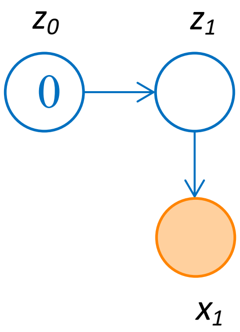
- Emission probabilities:

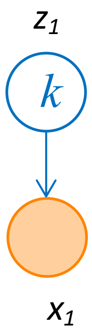
(*K* x *B* matrix)

$$e_k(b) = P(x_i = b | z_i = k)$$



- A way to deal with the first state:





$\pi_k = p(z_1 = k)$

Z_0 = Special start state with no emission

π_k = probability of starting in state k

(*K* x *1* vector)

Do this way for lab!

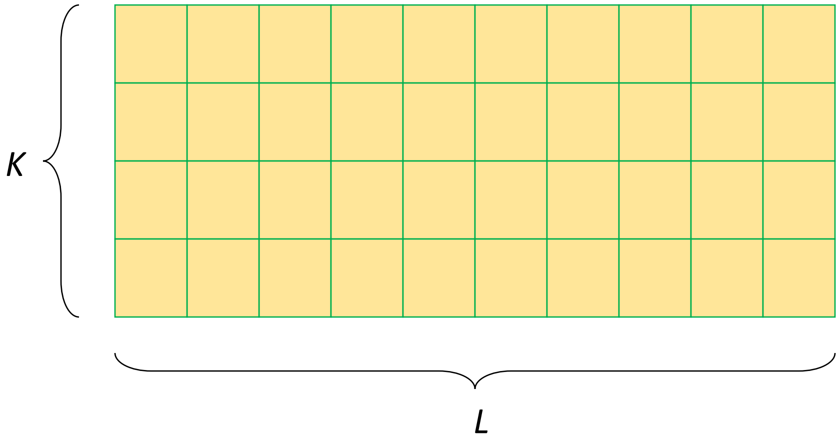
Viterbi Algorithm

- **Input:** observed sequence ($x_1, x_2, \ldots, x_L$) and transition/emission probabilities (***a*** and ***e*** matrices)
- **Output:** most probable (i.e. most likely) hidden state sequence $\boldsymbol{z}^*$

Viterbi Algorithm

- **Input:** observed sequence (x_1,x_2,…,x_L) and transition/emission probabilities (***a*** and ***e*** matrices)
- **Output:** most probable (i.e. most likely) hidden state sequence $\boldsymbol{z}^*$

- **Initialization:** create a *K* x *L* matrix, this will be our dynamic programming (DP) table V



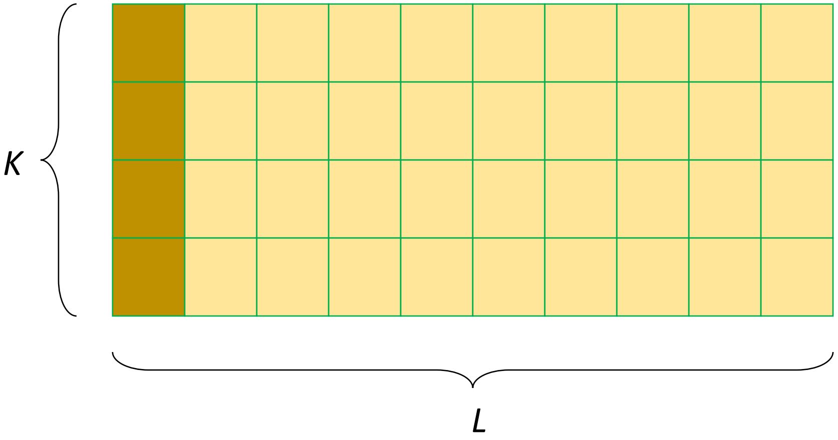
Viterbi Algorithm

- **Input:** observed sequence (x_1,x_2,…,x_L) and transition/emission probabilities (***a*** and ***e*** matrices)
- **Output:** most probable (i.e. most likely) hidden state sequence $\boldsymbol{z}^*$

- **Initialization:** create a *K* x *L* matrix, this will be our dynamic programming (DP) table

$$\boxed{V_k(1) = \pi_k e_k(x_1)}$$

(Note: there are lots of ways to initialize, this avoids a special start state.)



Viterbi Algorithm

- **Input:** observed sequence (x_1,x_2,…,x_L) and transition/emission probabilities (***a*** and ***e*** matrices)
- **Output:** most probable (i.e. most likely) hidden state sequence $\boldsymbol{z}^*$

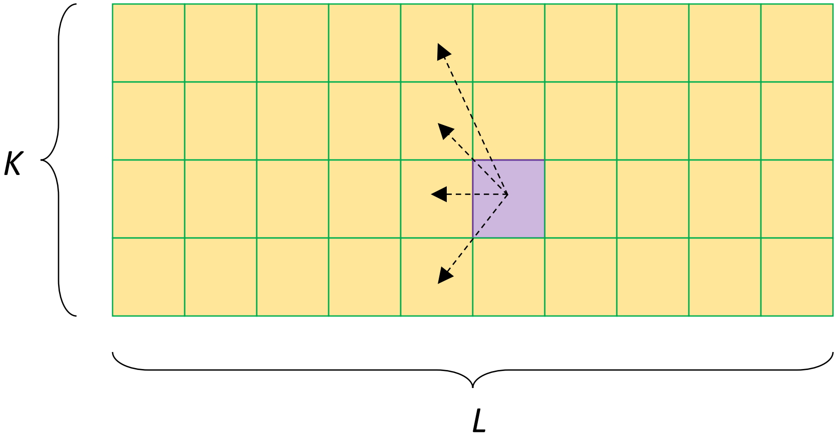
- **Initialization:** create a *K* x *L* matrix, this will be our dynamic programming (DP) table

$$\boxed{V_k(1) = \pi_k e_k(x_1)}$$

(Note: there are lots of ways to initialize, this avoids a special start state.)

- **Recursion:**

$$\boxed{V_k(i) = e_k(x_i) \cdot \max_l \left\{ V_l(i-1) \cdot a_{lk} \right\}}$$



Viterbi Algorithm

- **Input:** observed sequence (x_1,x_2,…,x_L) and transition/emission probabilities (***a*** and ***e*** matrices)
- **Output:** most probable (i.e. most likely) hidden state sequence $\boldsymbol{z}^*$

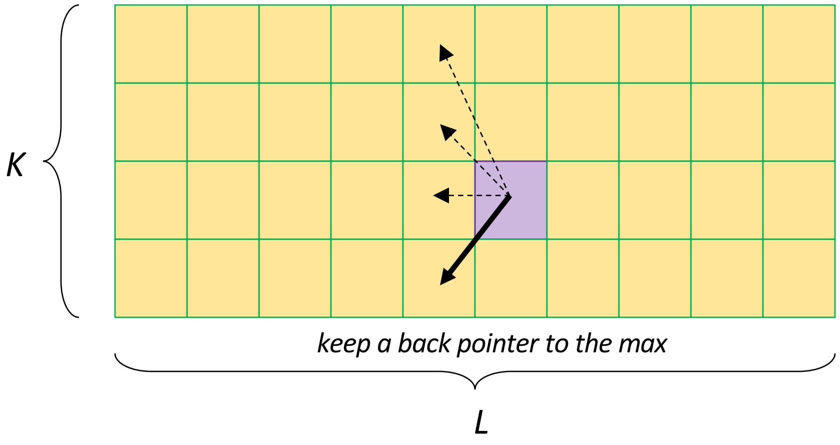
- **Initialization:** create a *K* x *L* matrix, this will be our dynamic programming (DP) table

$$\boxed{V_k(1) = \pi_k e_k(x_1)}$$

(Note: there are lots of ways to initialize, this avoids a special start state.)

- **Recursion:**

$$\boxed{V_k(i) = e_k(x_i) \cdot \max_l \left\{ V_l(i-1) \cdot a_{lk} \right\}}$$



keep a back pointer to the max

Viterbi Algorithm

- **Input:** observed sequence ($x_1, x_2, \ldots, x_L$) and transition/emission probabilities (***a*** and ***e*** matrices)
- **Output:** most probable (i.e. most likely) hidden state sequence $\boldsymbol{z}^*$

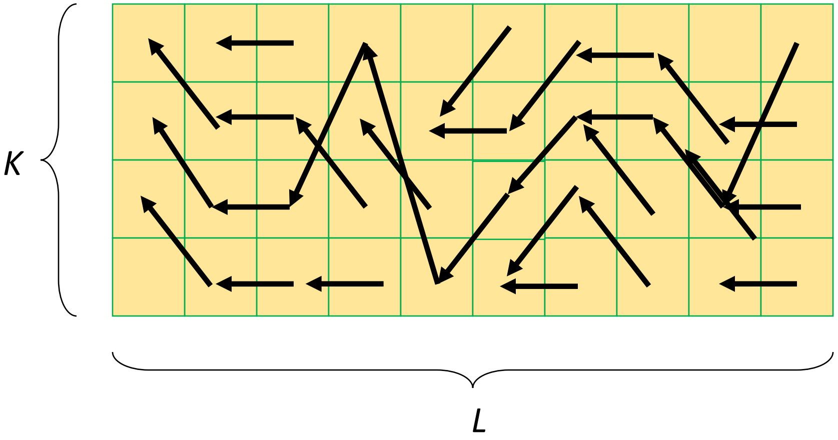
- **Initialization:** create a *K* x *L* matrix, this will be our dynamic programming (DP) table

$$V_k(1) = \pi_k e_k(x_1)$$

(Note: there are lots of ways to initialize, this avoids a special start state.)

- **Recursion:**

$$V_k(i) = e_k(x_i) \cdot \max_l \left\{ V_l(i-1) \cdot a_{lk} \right\}$$



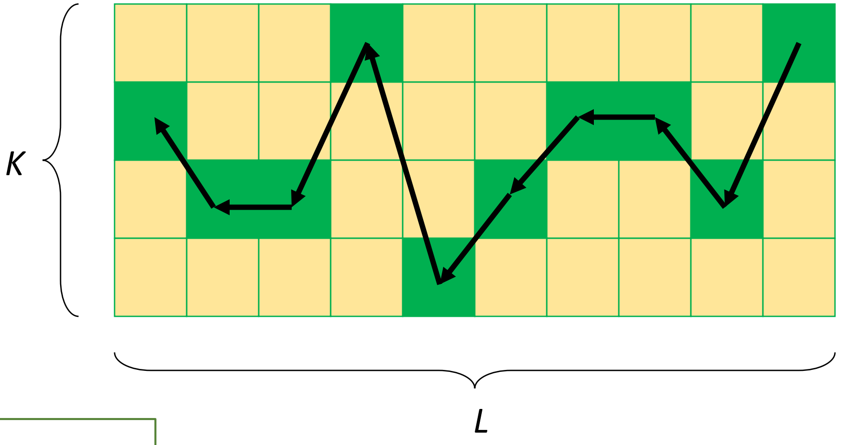
Viterbi Algorithm

- **Input:** observed sequence (x_1,x_2,…,x_L) and transition/emission probabilities (***a*** and ***e*** matrices)
- **Output:** most probable (i.e. most likely) hidden state sequence $\boldsymbol{z}^*$

- **Initialization:** create a *K* x *L* matrix, this will be our dynamic programming (DP) table

$$\boxed{V_k(1) = \pi_k e_k(x_1)}$$

(Note: there are lots of ways to initialize, this avoids a special start state.)



- **Recursion:**

$$\boxed{V_k(i) = e_k(x_i) \cdot \max_l \left\{ V_l(i-1) \cdot a_{lk} \right\}}$$

- **Termination and traceback:** $$\boxed{P(\vec{x}, \vec{z}^*) = \max_k \left\{ V_k(L) \right\}}$$

Viterbi Algorithm

- **Input:** observed sequence (x_1,x_2,…,x_L) and transition/emission probabilities (***a*** and ***e*** matrices)
- **Output:** most probable (i.e. most likely) hidden state sequence $\boldsymbol{z}^*$

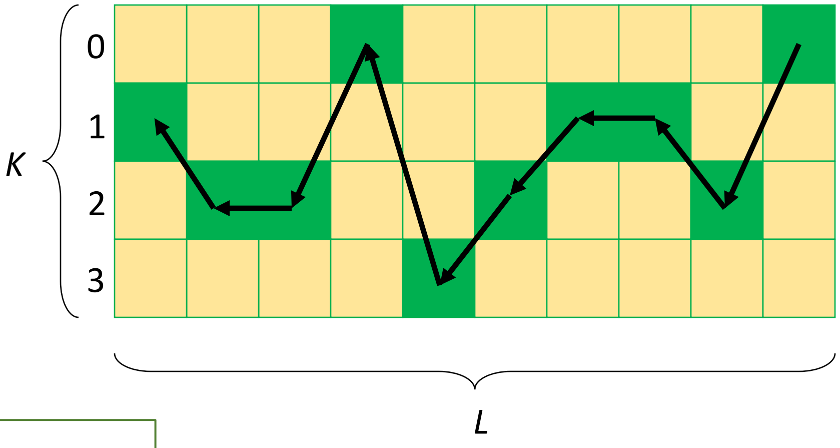
- **Initialization:** create a *K* x *L* matrix, this will be our dynamic programming (DP) table

$$V_k(1) = \pi_k e_k(x_1)$$

(Note: there are lots of ways to initialize, this avoids a special start state.)

- **Recursion:**

$$V_k(i) = e_k(x_i) \cdot \max_l \left\{ V_l(i-1) \cdot a_{lk} \right\}$$



- **Termination and traceback:**

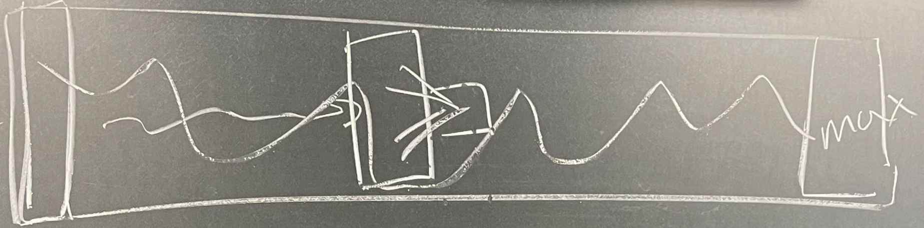
$$P(\vec{x}, \vec{z}^*) = \max_k \left\{ V_k(L) \right\}$$

$\boldsymbol{z}^*$ = (1,2,2,0,3,2,1,1,2,0)

K states, L observations: what is the complexity of the Viterbi algorithm?

Viterbi algorithm worksheet

$K = 4$



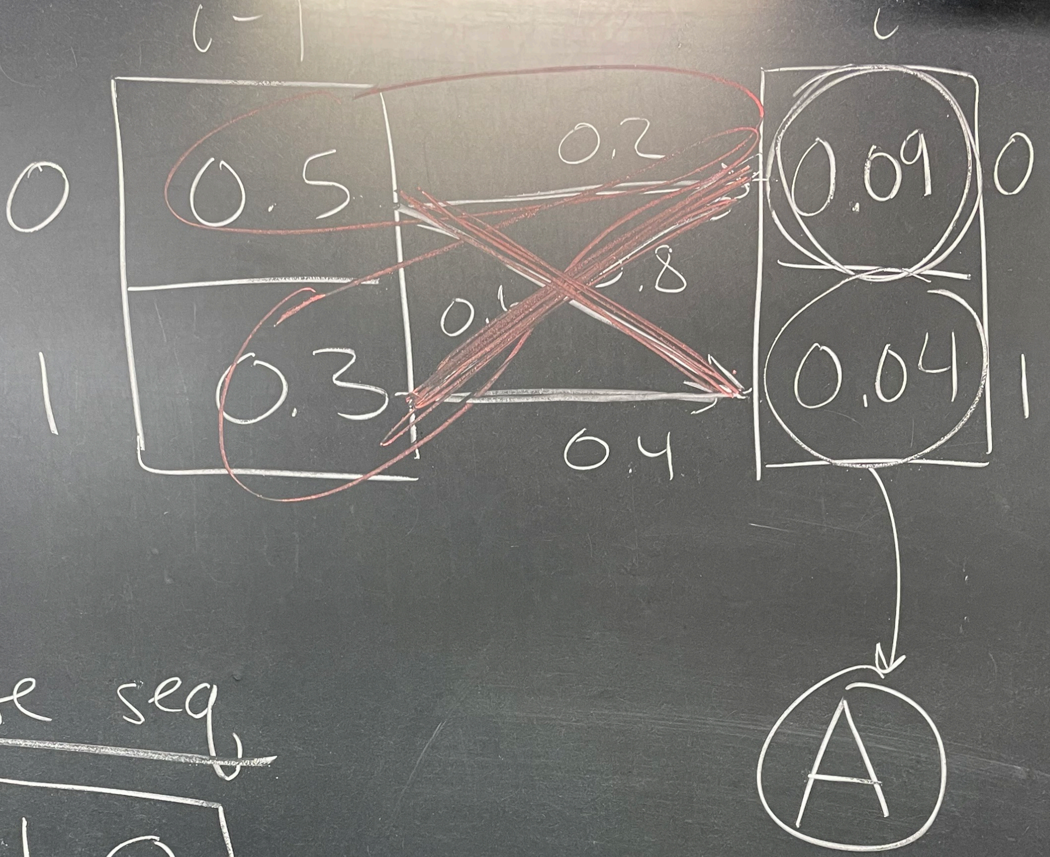
$L = 100{,}000$

① $\underset{i=1}{K} \cdot \underset{i=2}{K} \cdot \underset{i=3}{K} \cdots \underset{i=L}{K} = K^L$

② at least $\cancel{O(KL)}$

$O(LK^2)$

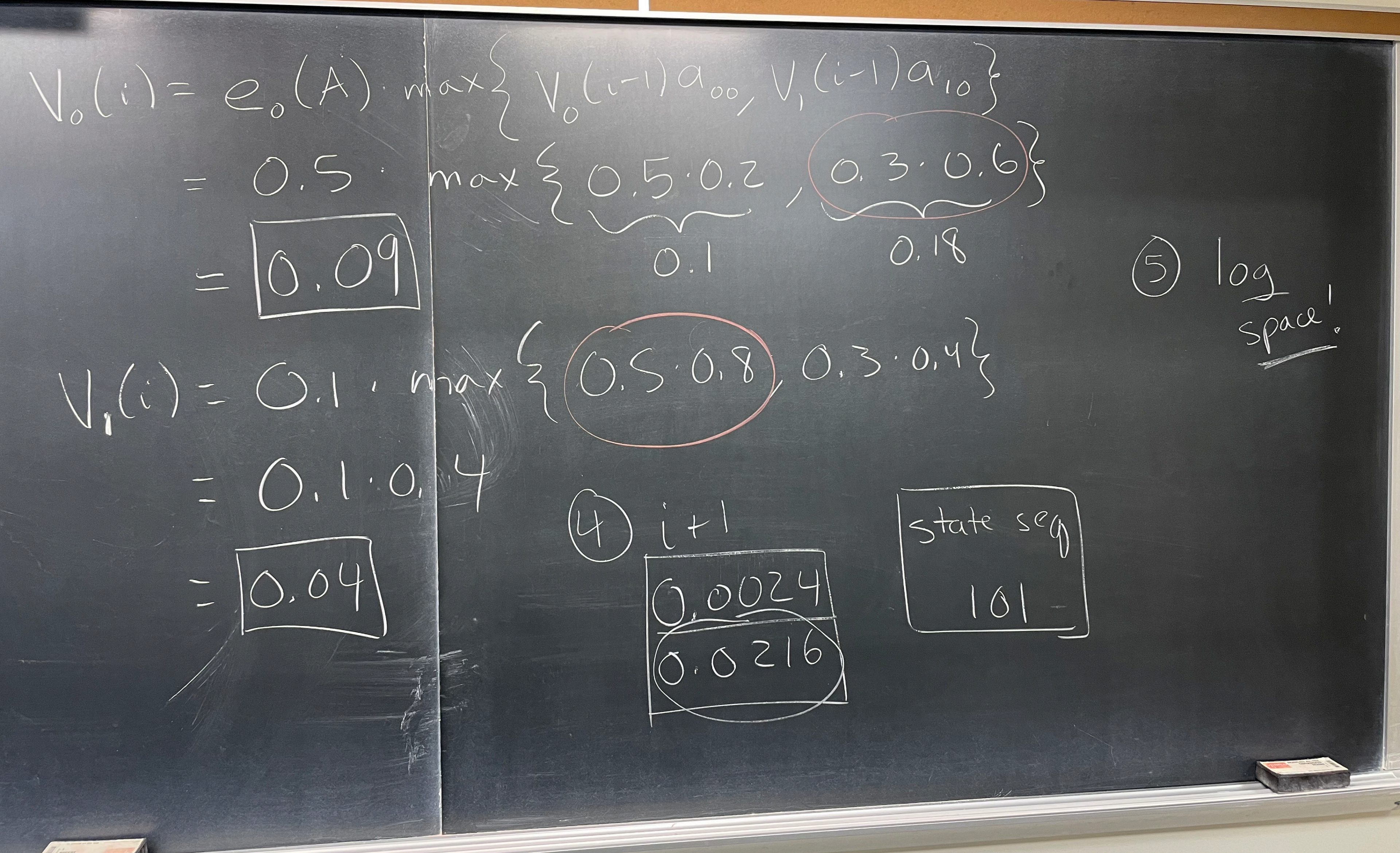
3



State seq.

$$\boxed{1\ 0}$$

$$[z_1, z_2] = \bar{\bar{z}}^*$$



$$V_0(i) = e_0(A) \cdot \max\{V_0(i-1)a_{00}, V_1(i-1)a_{10}\}$$

$$= 0.5 \cdot \max\{\underbrace{0.5 \cdot 0.2}_{0.1}, \underbrace{0.3 \cdot 0.6}_{0.18}\}$$

$$= \boxed{0.09}$$

$$V_1(i) = 0.1 \cdot \max\{0.5 \cdot 0.8, 0.3 \cdot 0.4\}$$

$$= 0.1 \cdot 0.4$$

$$= \boxed{0.04}$$

④ i+1

0.0024
0.0216

state seq

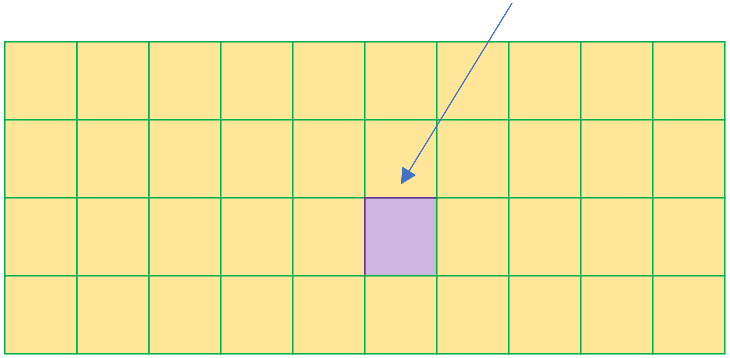
101

⑤ log space!

Forward-backward Algorithm

- **Input:** observed sequence x=(x_1,x_2,…,x_L) and transition/emission probabilities (***a*** and ***e*** matrices)
- **Output:** posterior probability of being in each hidden state at each time point $P(Z_i=k|x)$

What is the probability of being in this state, conditional on the observations?

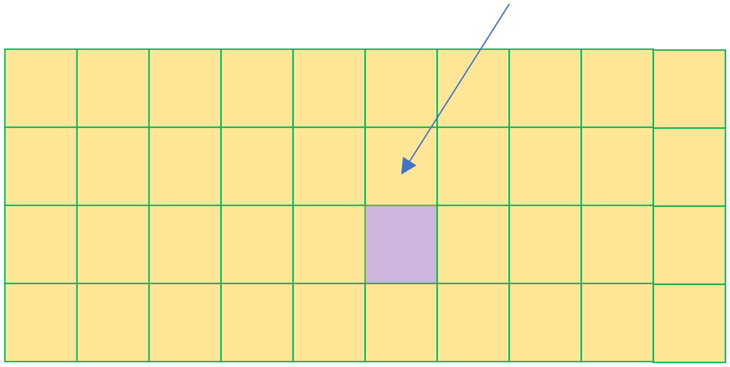


Want to compute $P(Z_i=k|x_1,x_2 \ldots x_L) = P(Z_i=k, x_1,x_2 \ldots x_L) / P(x_1,x_2 \ldots x_L)$

Forward-backward Algorithm

Forward matrix $f_K(i)$

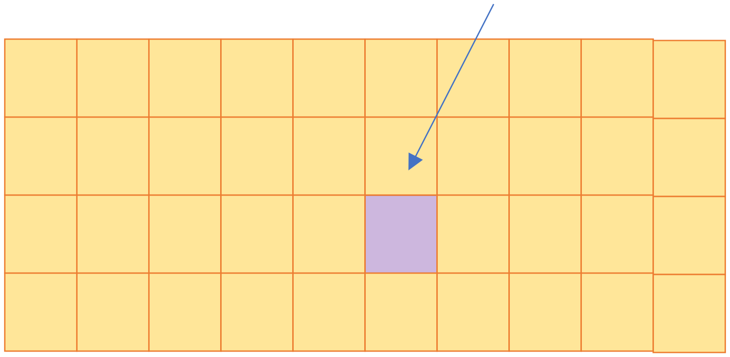
What is the probability of being in this state, conditional on all the earlier observations



×

Backward matrix $b_k(i)$

What is the probability of all the later observations, conditional on being in this state



Want to compute $P(Z_i\!=\!k \mid x_1,x_2 \ldots x_L) = P(Z_i\!=\!k,\ x_1,x_2 \ldots x_L\ ) \,/\, P(x_1,x_2 \ldots x_L)$

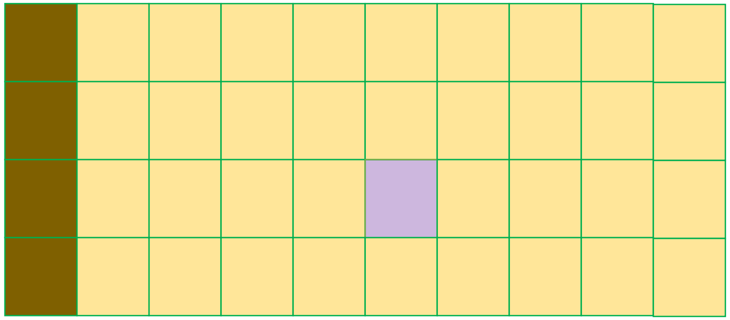
But we can break this up: $P(Z_i\!=\!k,\ x_1,x_2 \ldots x_L\ ) = \boxed{P(Z_i\!=\!k,\ x_1,x_2 \ldots x_i\ )}\boxed{P(x_{i+1},x_2 \ldots x_L \mid Z_i\!=\!k)}$

Probability of being in state k at time i, given the sequence up to time i: Forward algorithm

Probability of the sequence after time i, given that you are in state k at time i: Backward algorithm

Forward Algorithm

1. $f_k(1) = \pi_k e_k(x_1)$



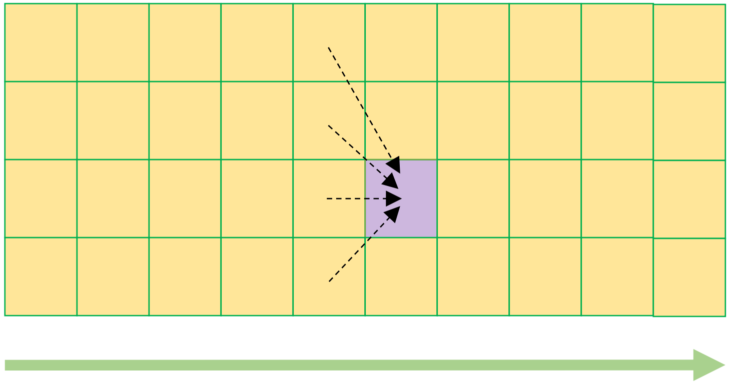
$P(Z_i=k, x_1,x_2 \ldots x_i)$

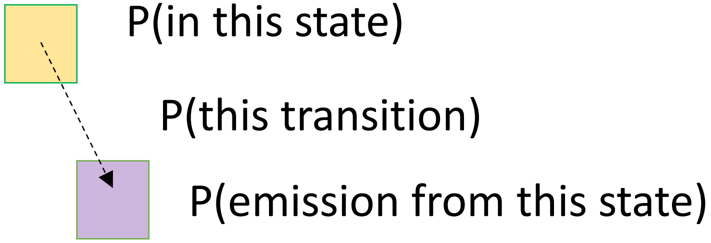
Probability of being in state k at time i, given the sequence up to time i: Forward algorithm

Forward Algorithm

1. $f_k(1) = \pi_k e_k(x_1)$

2 $f_k(i) = \Sigma_{j=1}^{K} f_j(i-1) a_{jk} e_k(x_i)$





P(in this state)

P(this transition)

P(emission from this state)

P(Z$_i$=k, x$_1$,x$_2$…x$_i$)

Probability of being in state k at time i, given the sequence up to time i: Forward algorithm

Forward Backward Algorithm

Goal

posterior

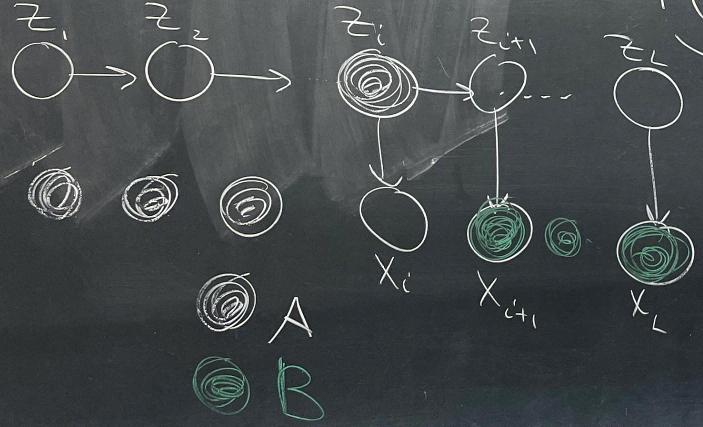
$$P(z_i=k \mid \vec{x}) = \frac{P(\vec{x}, z_i=k)}{P(\vec{x})}$$

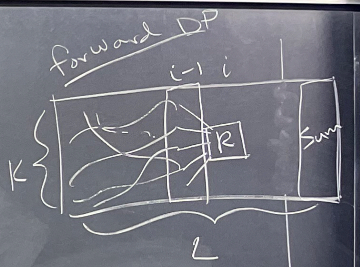
^for now

$$p(\vec{x}, z_i=k) = P(\overbrace{x_1, x_2 \cdots x_i, z_i=k}^{\text{up to } i \;\; A}, \overbrace{x_{i+1}, x_{i+2} \cdots x_L}^{\text{after } i \;\; B})$$

BAYES

$$= P(\underbrace{\overbrace{x_1, x_2 \cdots x_i, z_i=k}^{A}}_{\text{forward probabilities}}) \cdot P(\underbrace{\overbrace{x_{i+1} \cdots x_L}^{B} \mid z_i=k, \overbrace{\cancel{x_1, x_2 \cdots} x_i}^{A}}_{\text{backward probabilites.}})$$





Forward Algorithm

$f_k(i)$ = prob of observing $x_1 \dots x_i$ & ending in state k

$= P(x_1 \dots x_i, z_i = k)$

all possible prev states

recursion: $f_k(i) = e_k(x_i) \underbrace{\sum_l}_{\text{integrating over all possibilities}} f_l(i-1)\, a_{lk}$

initialization (same as viterbi)

$$f_k(1) = \pi_k e_k(x_1)$$

termination $P(\vec{x}) = \sum_k f_k(L)$

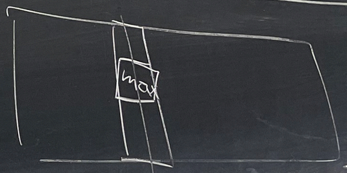
together

$$p(\vec{x}, z_i = k) = f_k(i)\, b_k(i)$$

$$\boxed{p(z_i = k \mid \vec{x}) = \frac{f_k(i)\, b_k(i)}{P(\vec{x})}}$$

posterior prob

posterior decoding $\boxed{\hat{z}_i = \underset{k}{\operatorname{argmax}}\, p(z_i = k \mid \vec{x})}$



Backward Algorithm

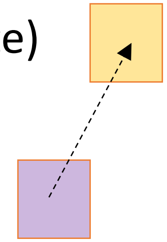
1 $b_k(K) = 1$

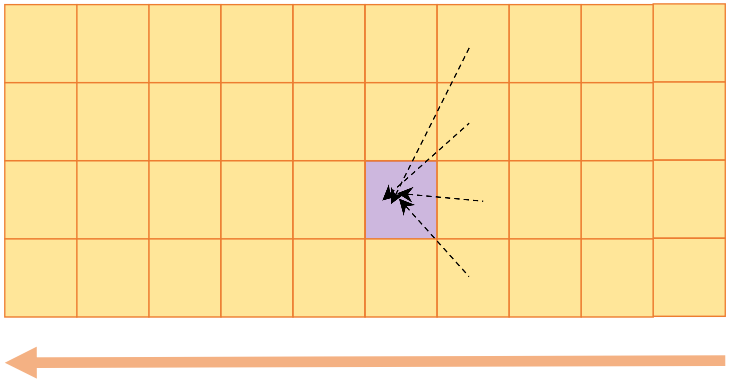
2 $b_k(i) = \Sigma_{j=1}^{K} b_j(i+1) a_{ij} e_j(x_{i+1})$

P(emission)

P(this state)

P(transition)





P(x$_{i+1}$,x$_2$…x$_L$|Z$_i$=k)

Probability of the sequence after time i, given that you are in state k at time i: Backward algorithm

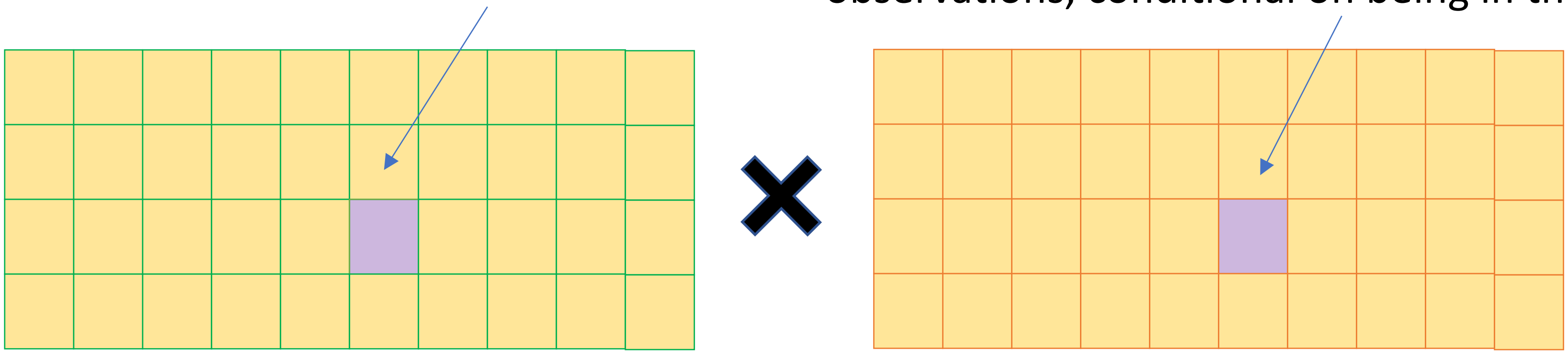
Forward-backward Algorithm

Forward matrix $f_K(i)$

Backward matrix $b_k(i)$

What is the probability of being in this state, conditional on all the earlier observations

What is the probability of all the later observations, conditional on being in this state



Want to compute $P(Z_i=k|x_1,x_2\ldots x_L) = P(Z_i=k,\ x_1,x_2\ldots x_L\ ) \,/\, P(x_1,x_2\ldots x_L)$

But we can break this up: $P(Z_i=k,\ x_1,x_2\ldots x_L\ ) = \boxed{P(Z_i=k,\ x_1,x_2\ldots x_i\ )}\boxed{P(x_{i+1},x_2\ldots x_L|Z_i=k)}$

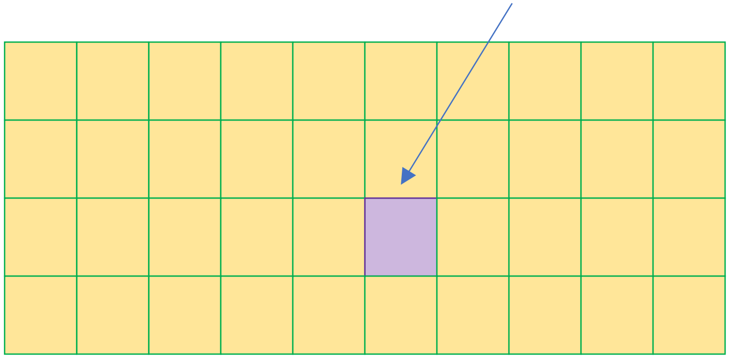
Probability of being in state k at time i, given the sequence up to time i: Forward algorithm

Probability of the sequence after time i, given that you are in state k at time i: Backward algorithm

Forward-backward Algorithm

- **Input:** observed sequence $x=(x_1,x_2,\ldots,x_L)$ and transition/emission probabilities (***a*** and ***e*** matrices)
- **Output:** posterior probability of being in each hidden state at each time point $P(Z_i=k|x)$

What is the probability of being in this state, conditional on the observations?



Want to compute $P(Z_i=k|x_1,x_2\ldots x_L) = P(Z_i=k, x_1,x_2\ldots x_L) / P(x_1,x_2\ldots x_L)$

What is the runtime of the forward-backward algorithm?

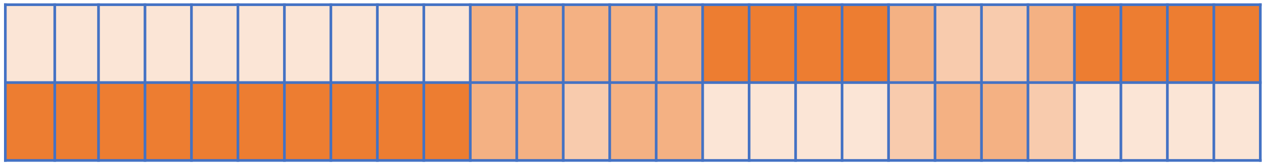
Parameter estimation

So far we assumed that we know the emission and transition probabilities, and the initial state distribution. If we don’t know these, can we estimate them?

Yes – using the Baum-Welch algorithm; a special case of the Expectation-Maximization (EM) algorithm:

0. Guess some parameters and run the F-B algorithm

1. Use estimated parameters to run the F-B algorithm

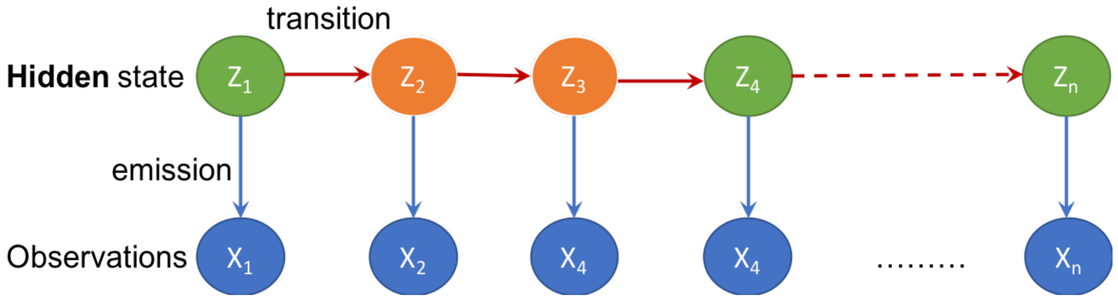


2. Use the output of the F-B algorithm to estimate the parameters

(estimate the parameters by using the empirical probabilities from the FB matrix)

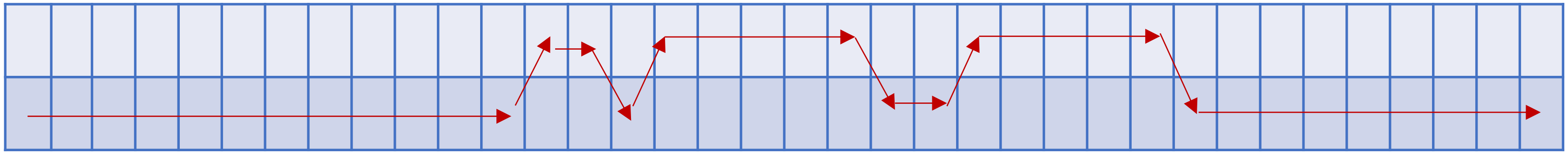
Summary of HMMs

Structure of a Hidden Markov Model: Observations, hidden states, emissions, transitions

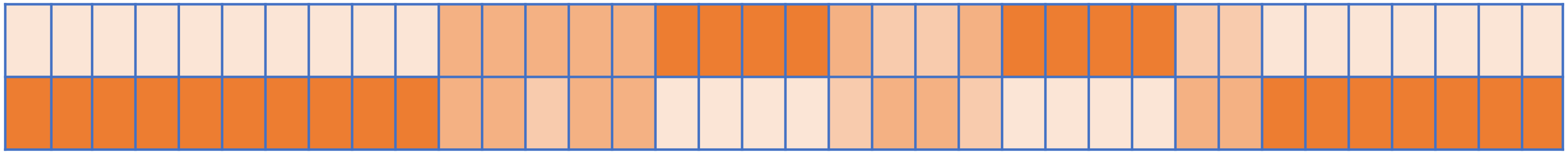


Three inference problems:

1) What is the most likely sequence of hidden states?: Viterbi algorithm



2) What is the most likely hidden state at each observation? Forward-Backward algorithm



3) What are the parameters? Baum-Welch algorithm