CS 68: BIOINFORMATICS

Prof. Sara Mathieson
Swarthmore College
Spring 2018

Outline: Apr 4

- Lab 5 Examples
- HMM example in population genetics
- Recap Viterbi Algorithm
- Forward-Backward Algorithm
- Posterior Decoding
- In lab tomorrow: working in log-space

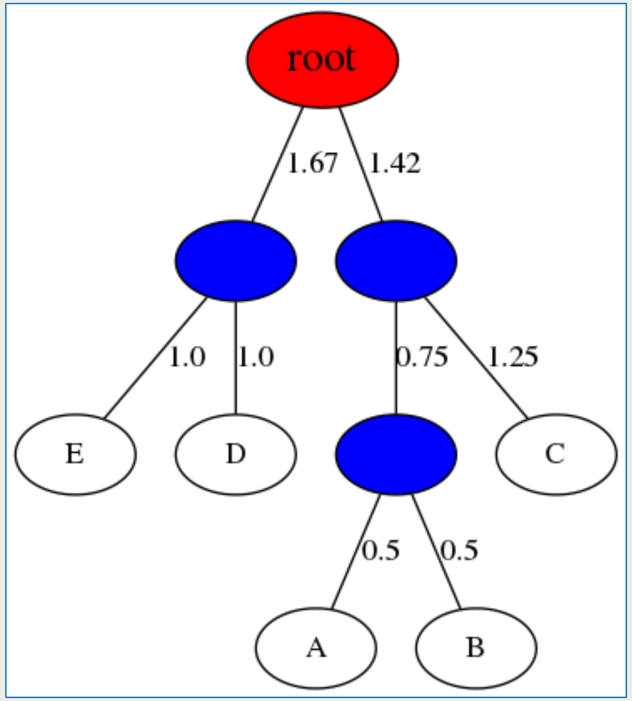
Notes:

- Office hours TODAY 1-3pm
- Lab 7 due tonight
- Lab 8: 1.5 week lab (last graded lab)

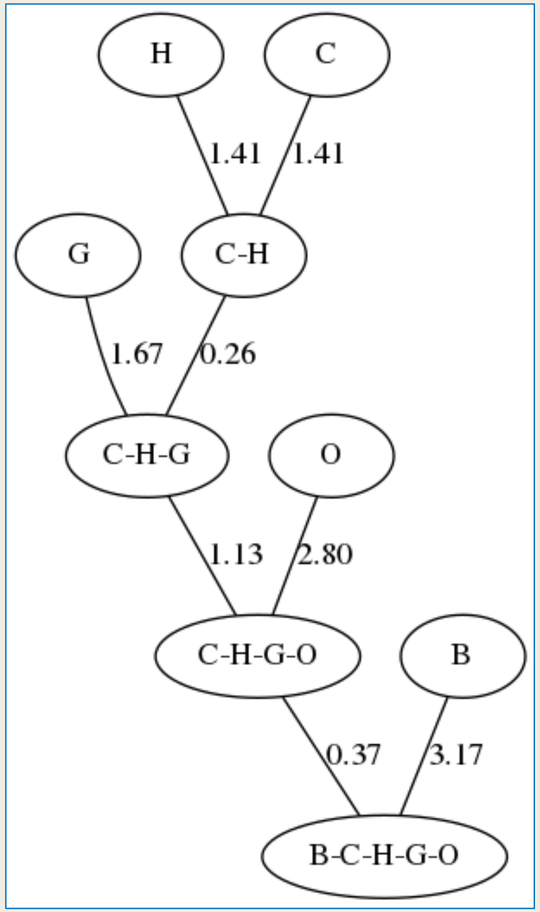
Lab 5 Examples

Lab 5: UPGMA visualizations

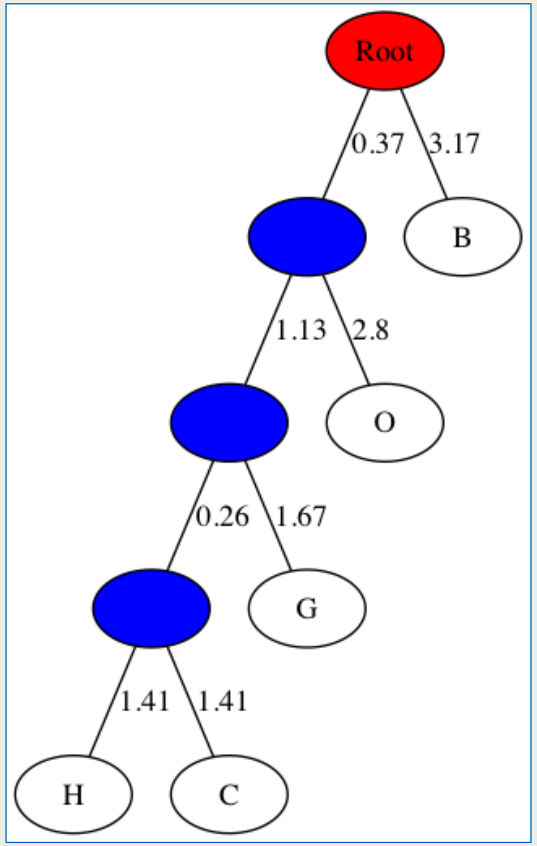
Sam & Hunter



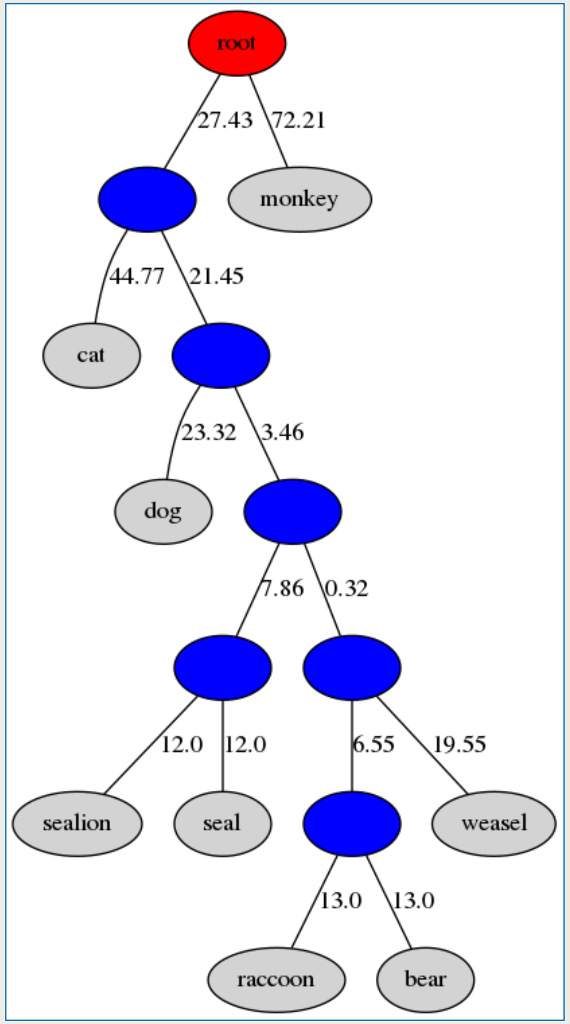
Ellen & Douglas



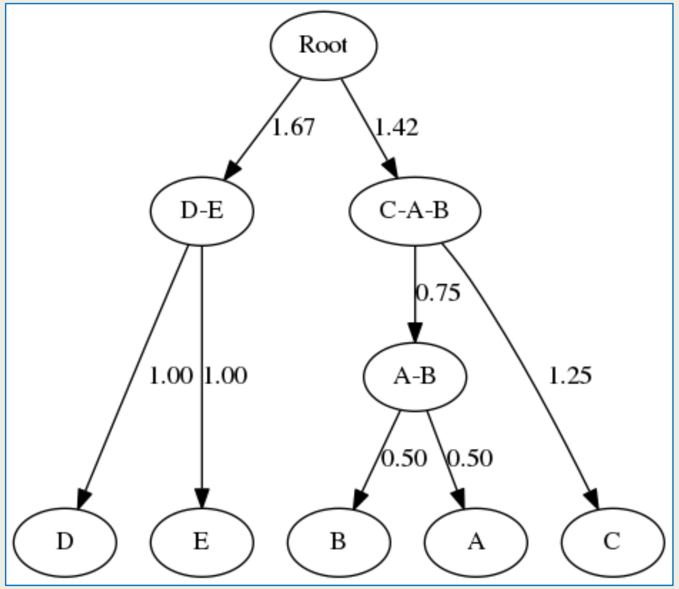
Tyler & Nathan



Neeraj & Tim

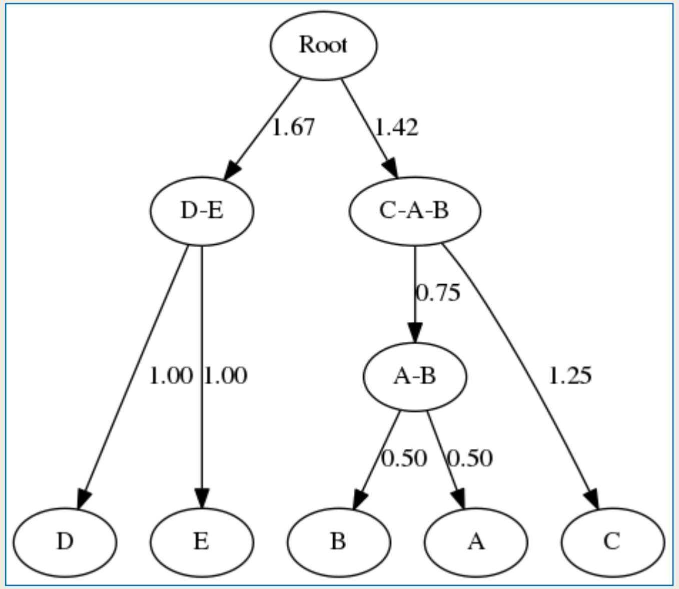


Lab 5: UPGMA visualizations (including branch lengths)



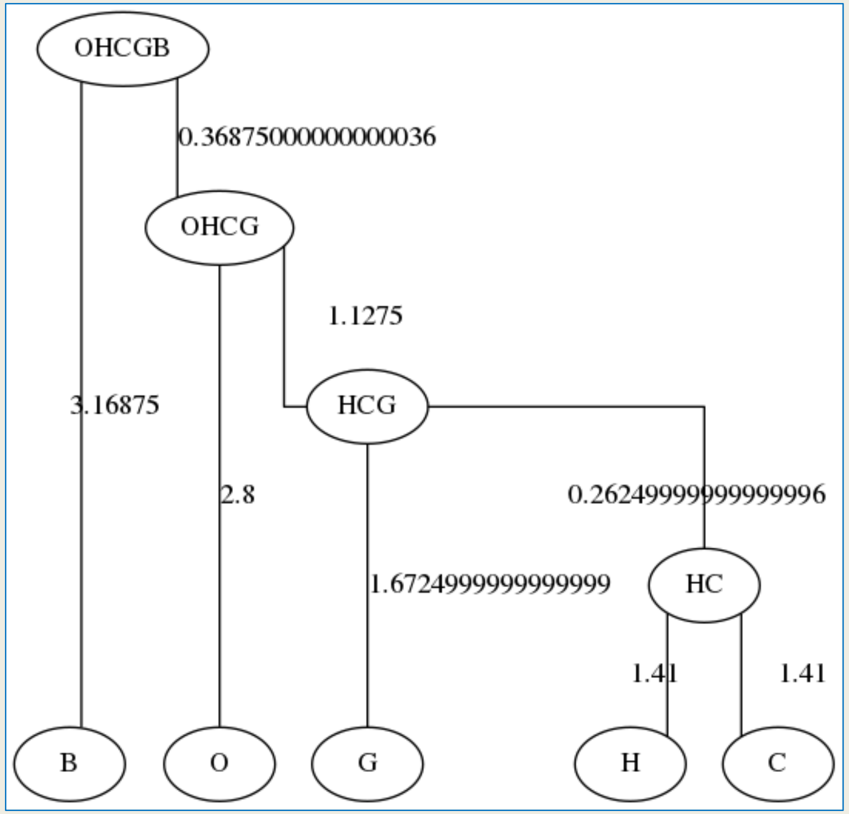
Sarah & Tommy

Lab 5: UPGMA visualizations (including branch lengths)

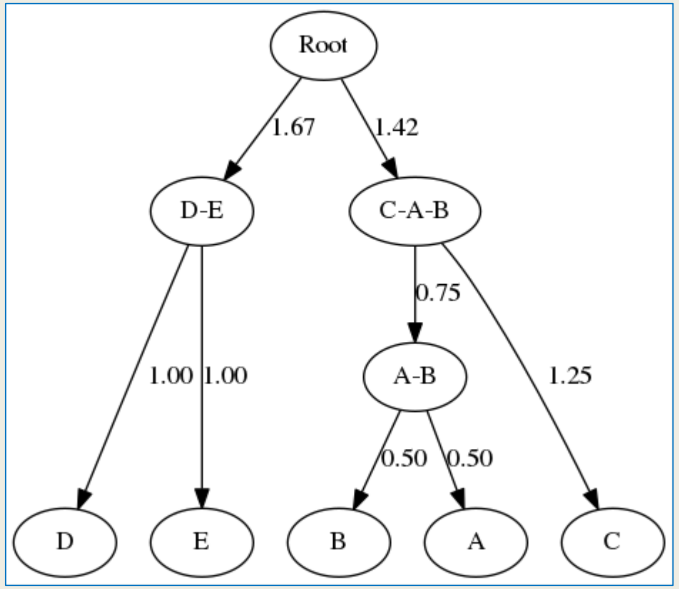


Sarah & Tommy

Genji & Eugene

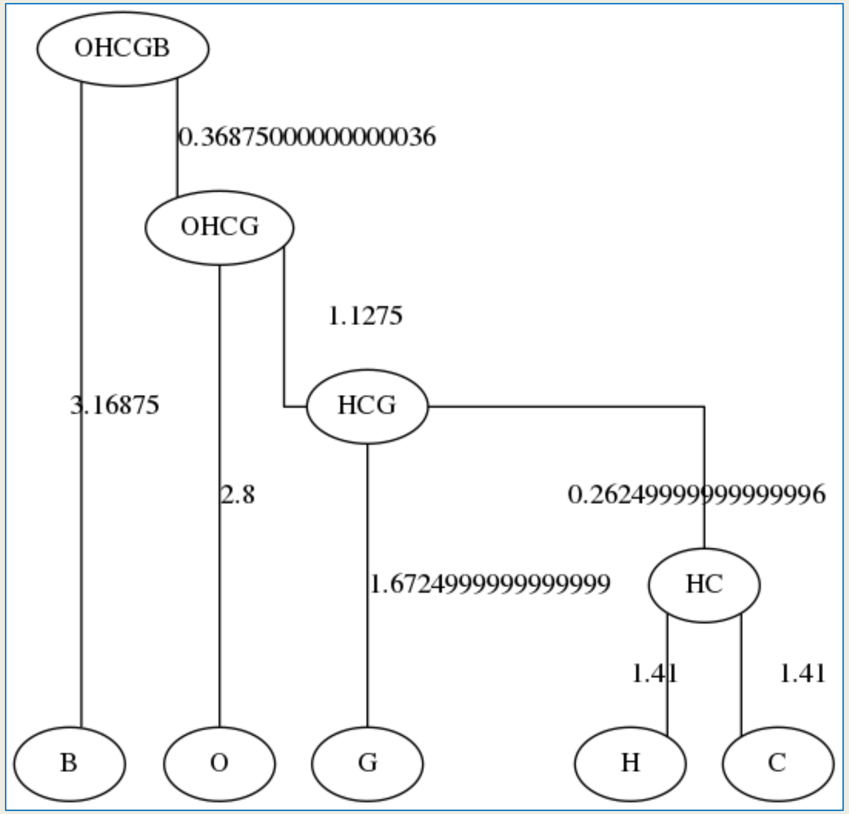


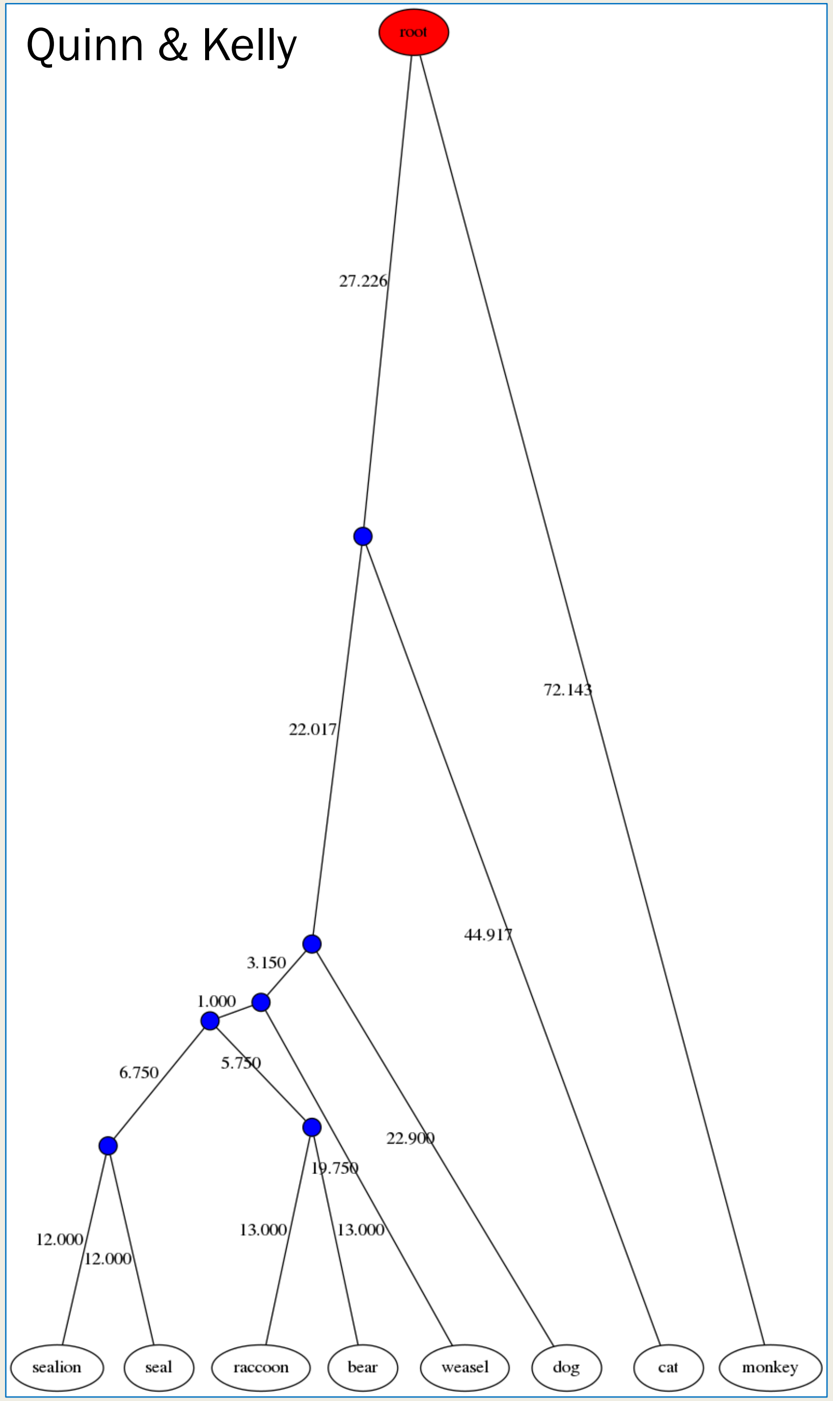
Lab 5: UPGMA visualizations (including branch lengths)



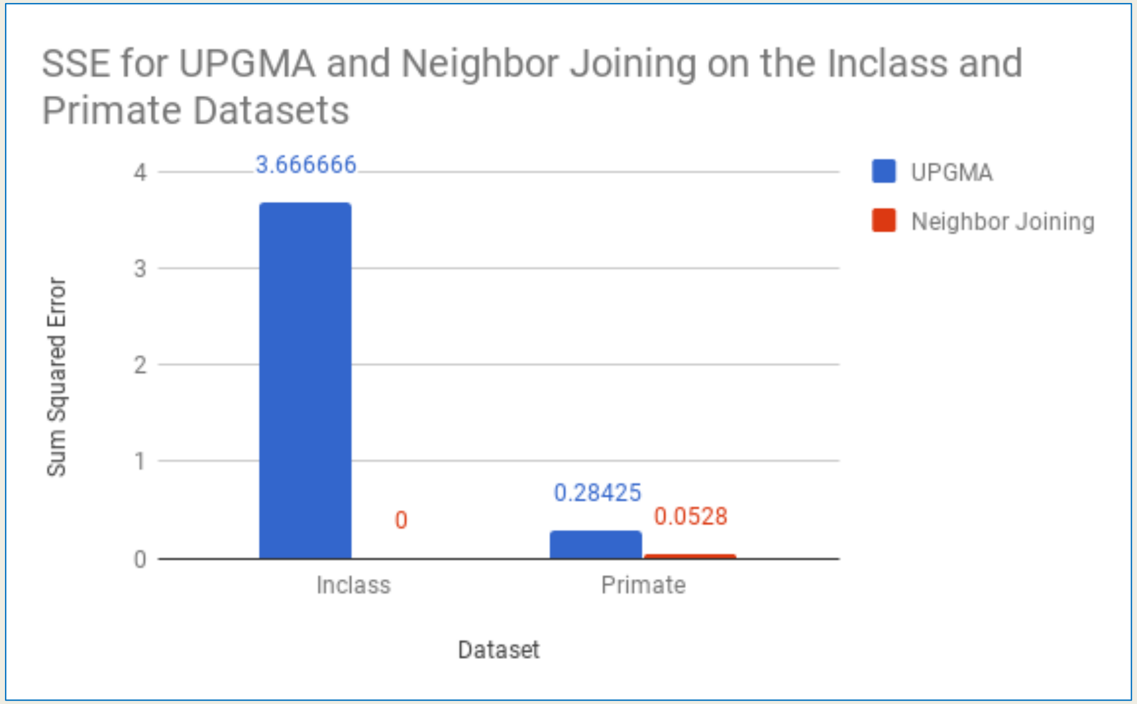
Sarah & Tommy

Genji & Eugene

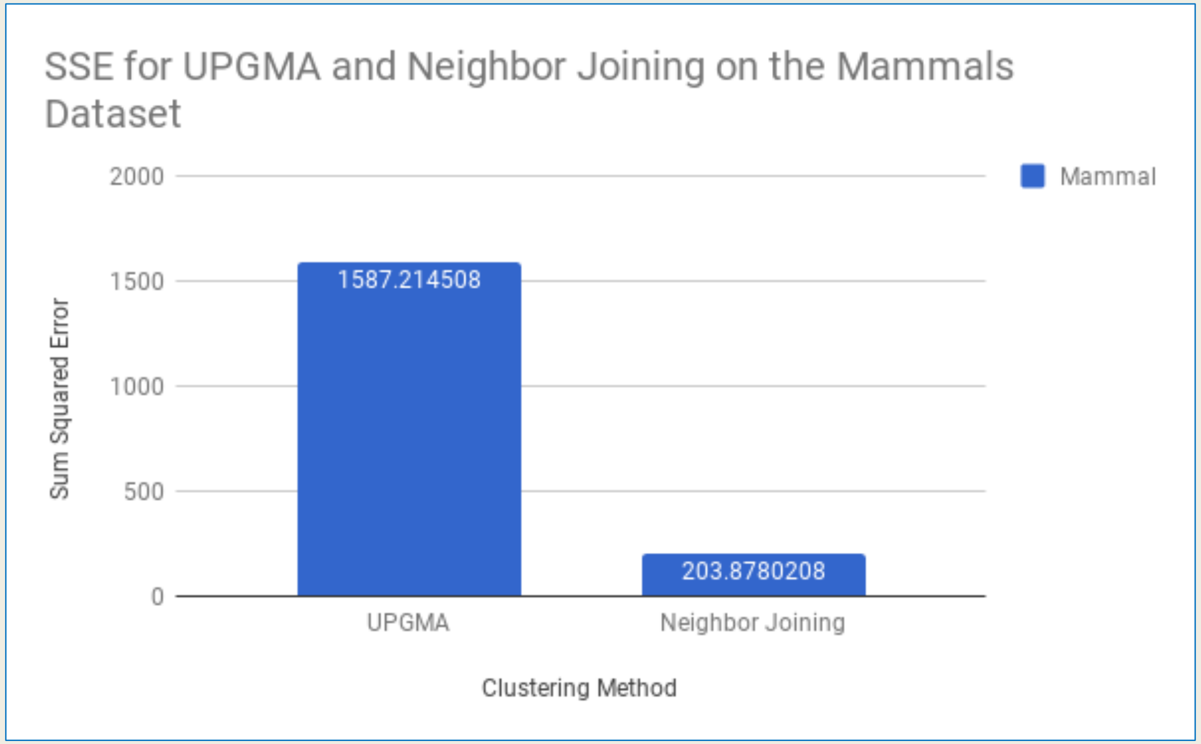



Quinn & Kelly

Lab 5: dissimilarity map comparisons



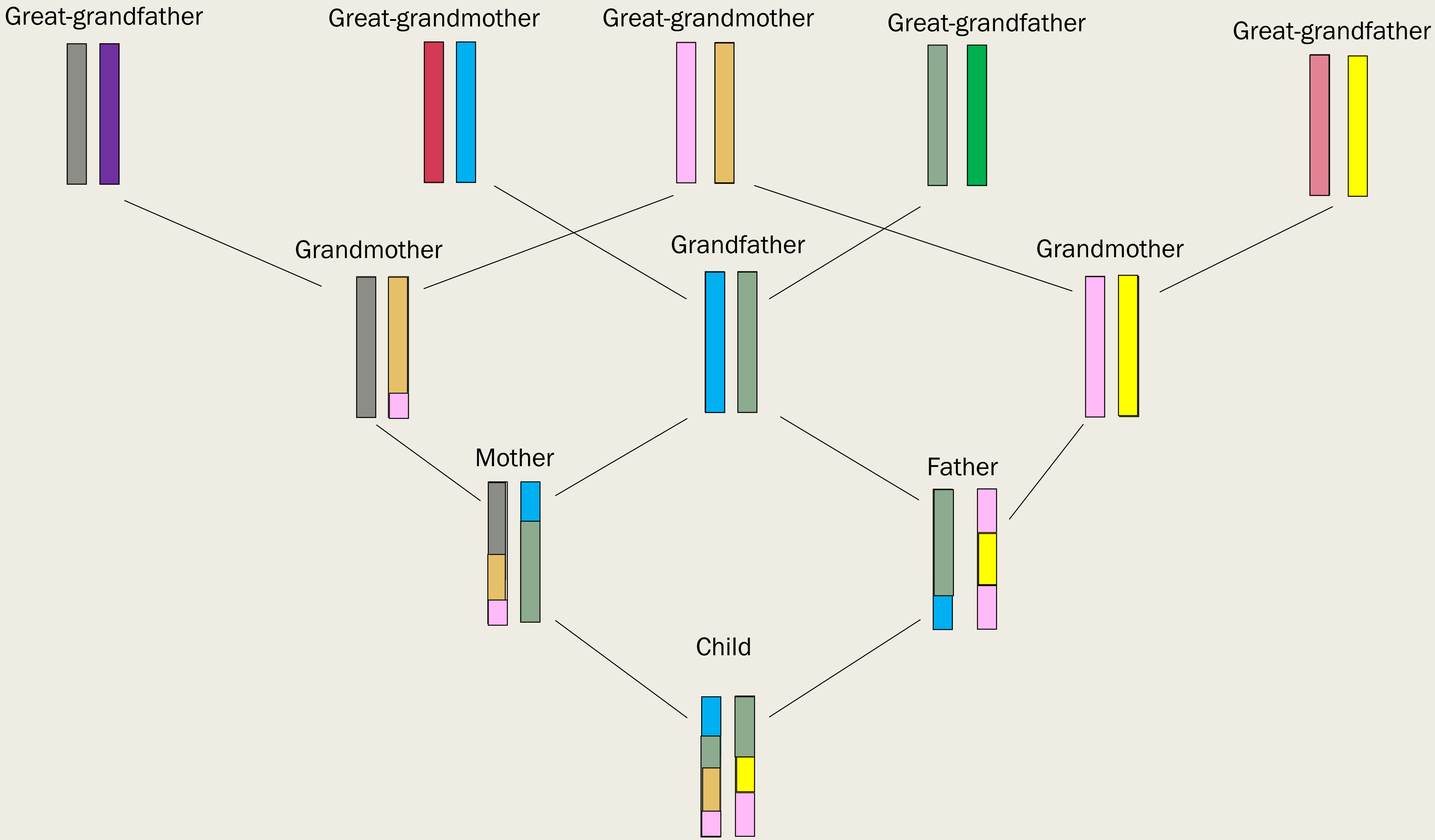
William



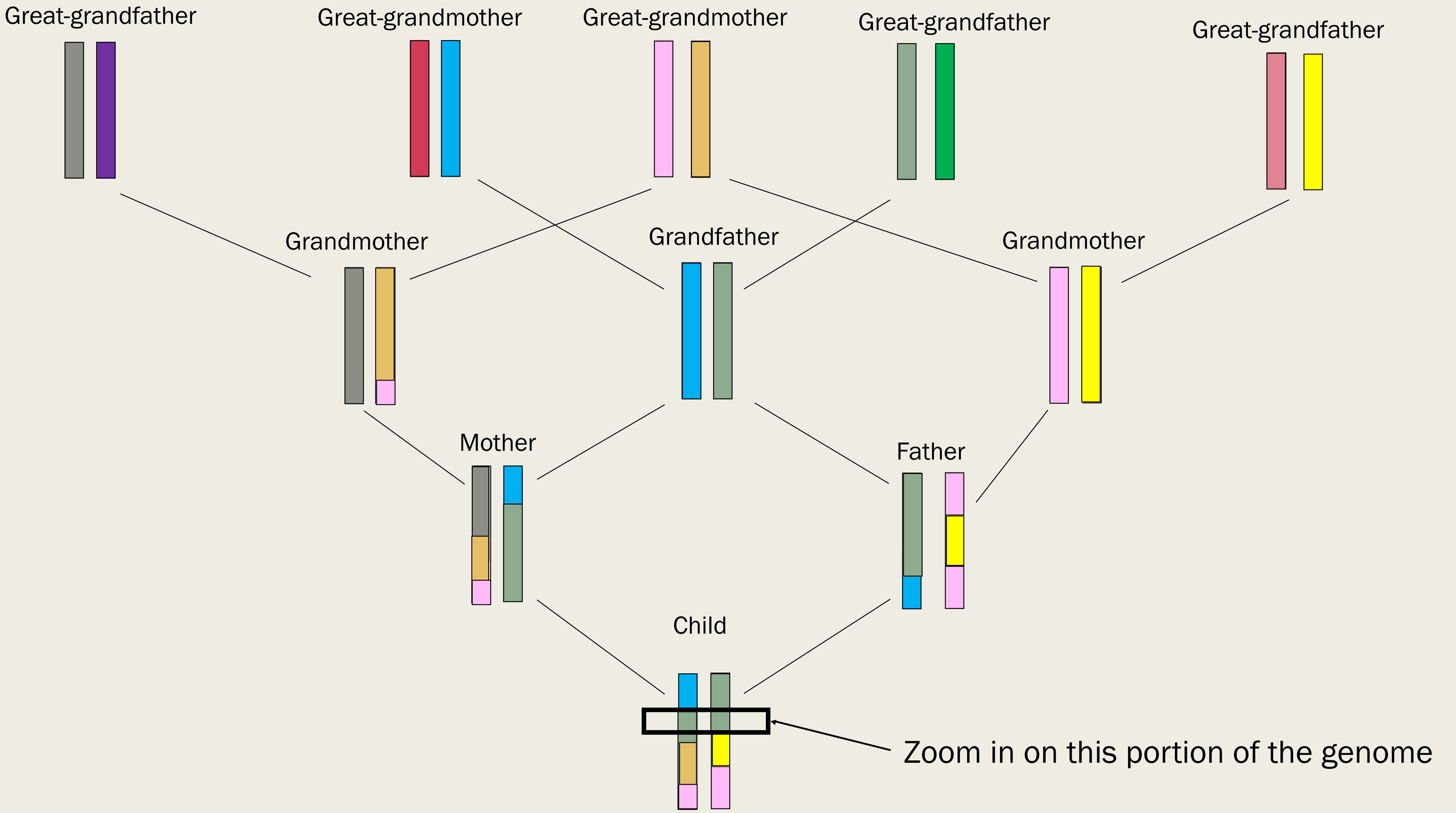
HMM example from population genetics

Back to recombination….

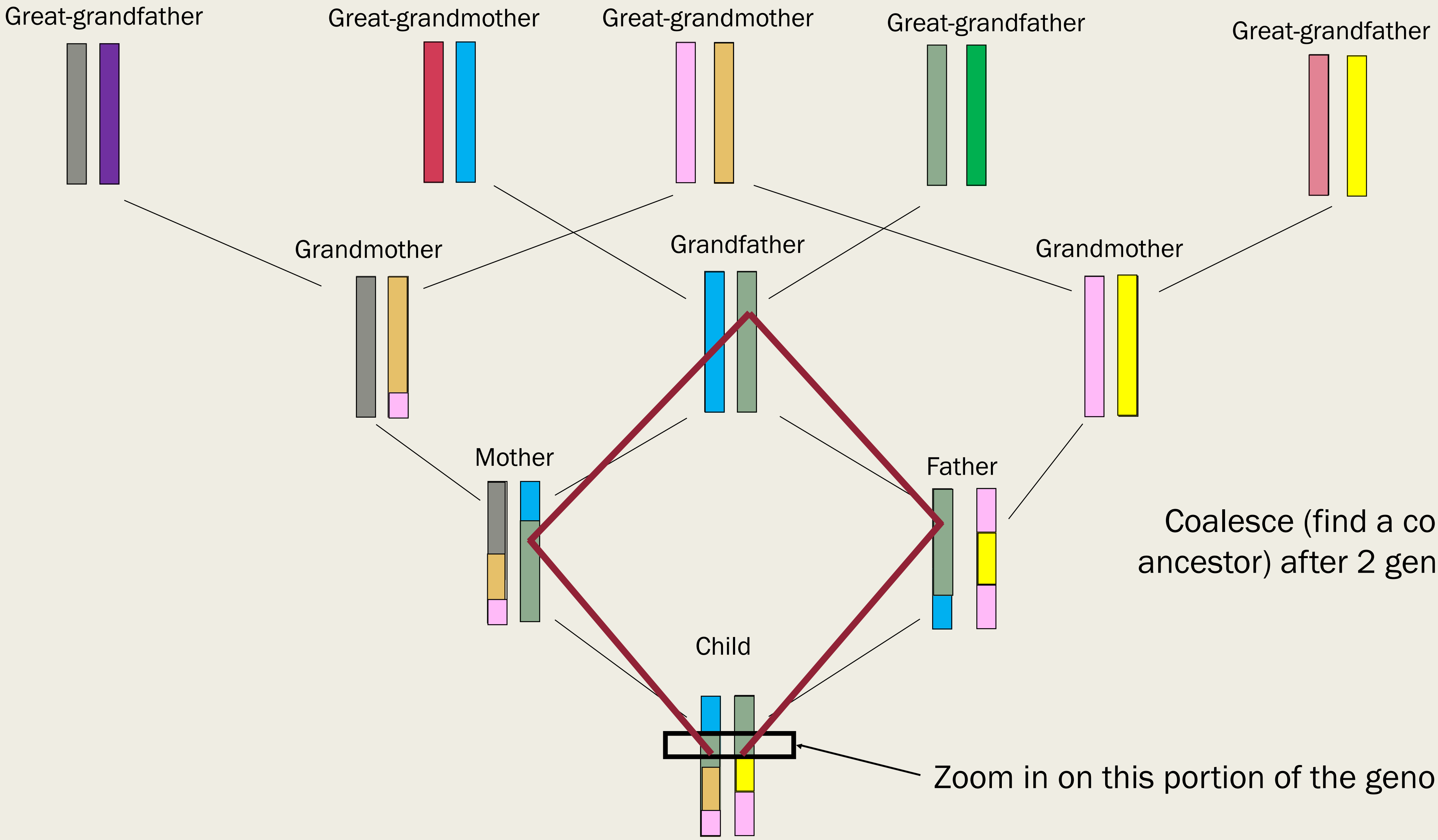
Recombination over time



Recombination over time



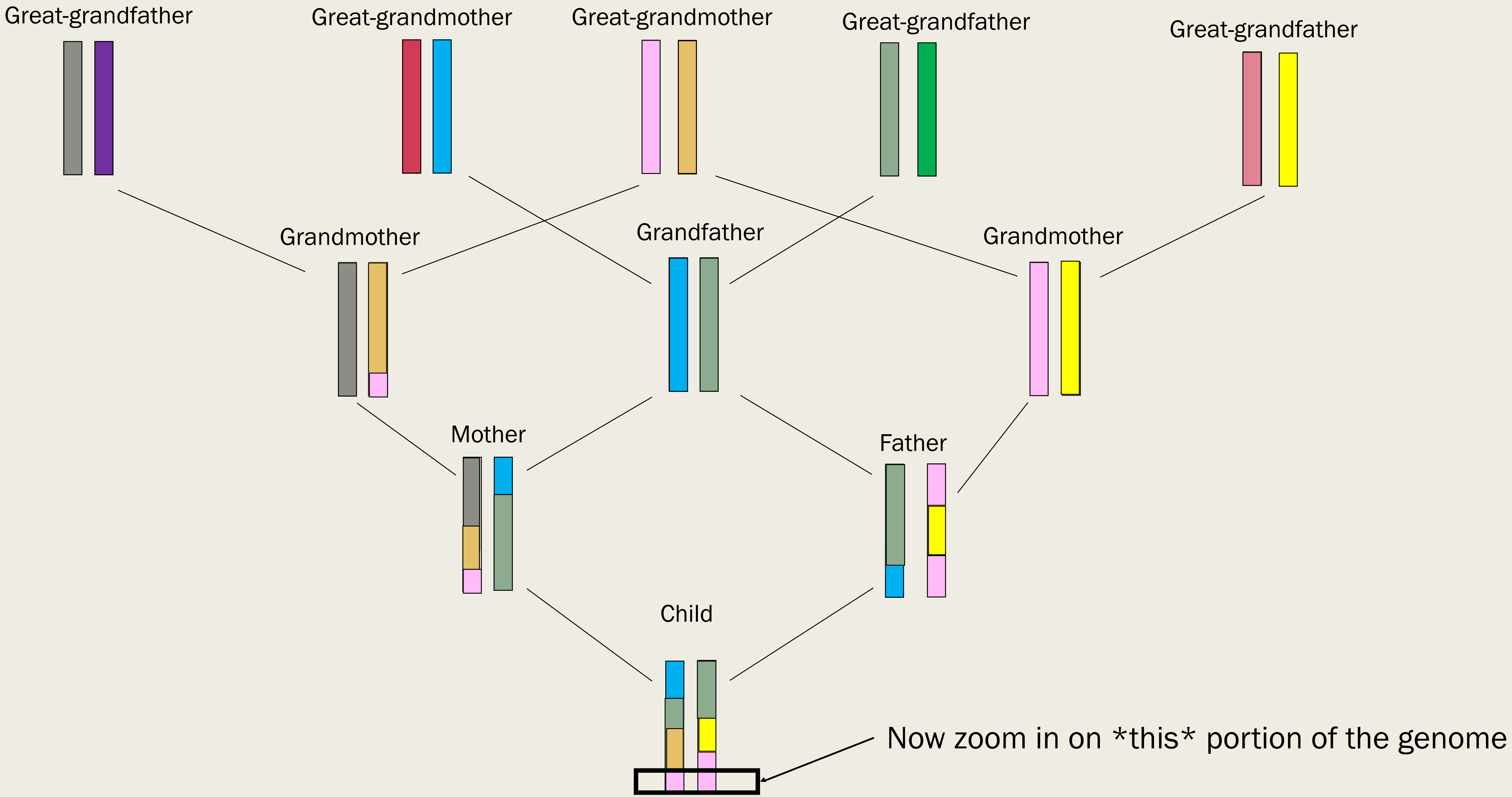
Recombination over time



Coalesce (find a common ancestor) after 2 generations

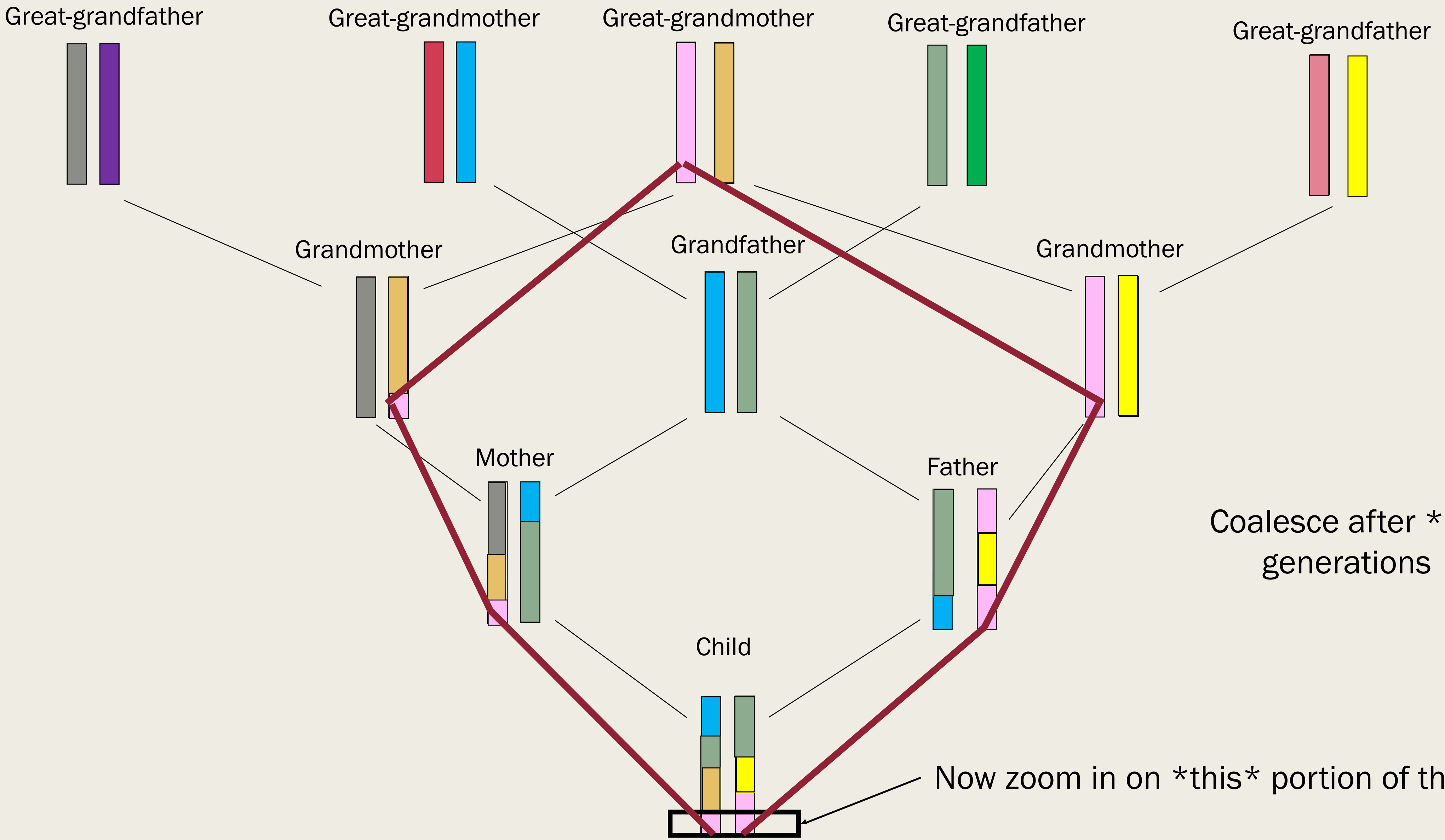
Zoom in on this portion of the genome

Recombination over time



Now zoom in on *this* portion of the genome

Recombination over time



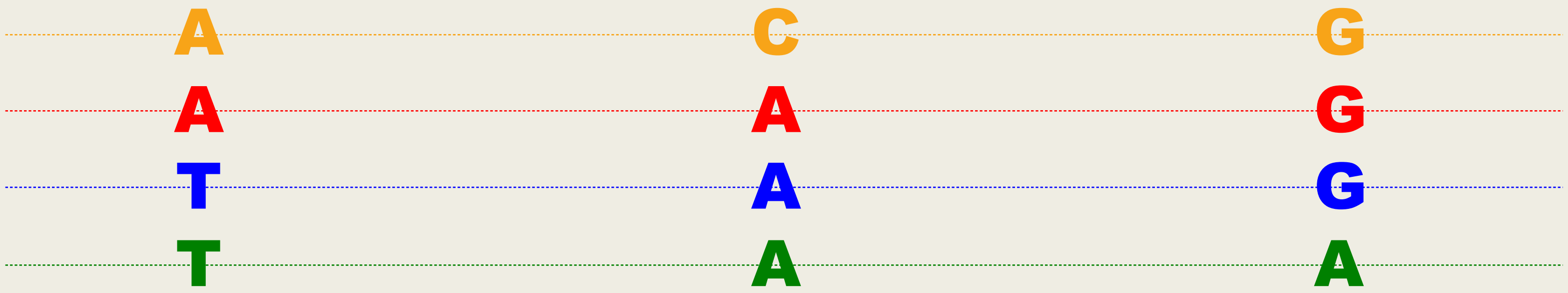
Coalesce after *3* generations

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



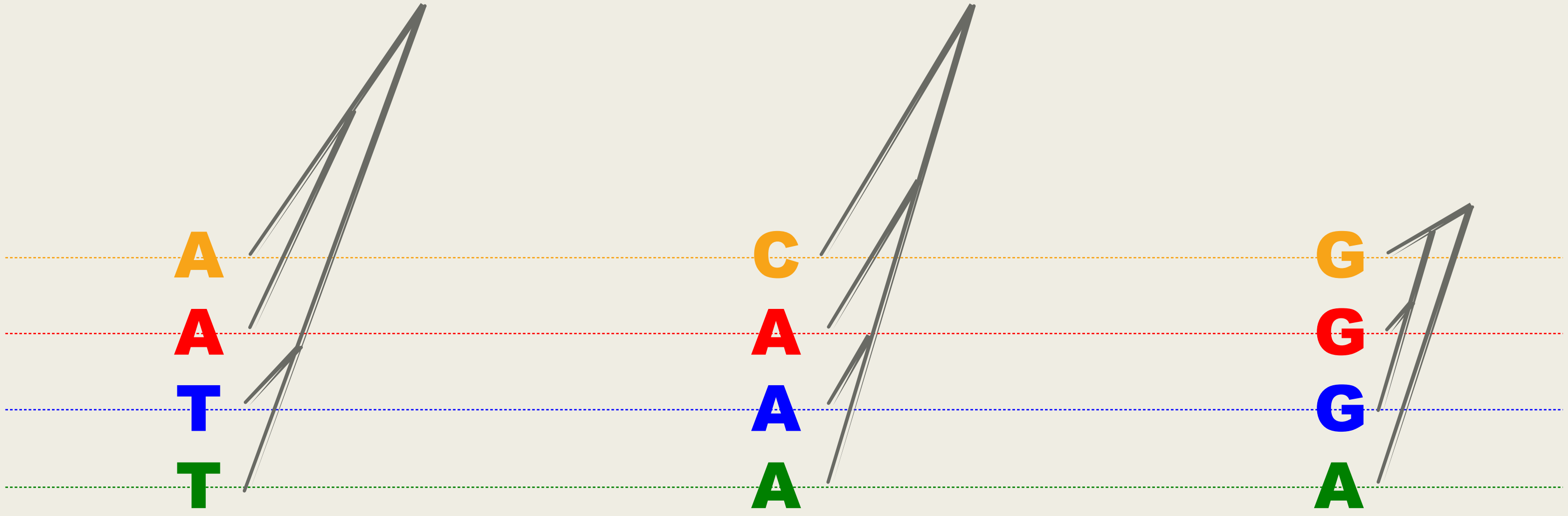
A C G

A A G

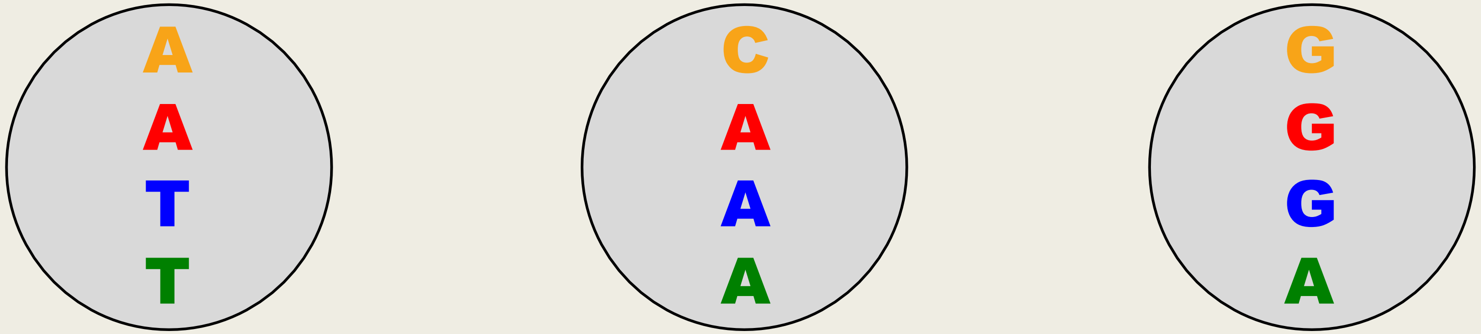
T A G

T A A

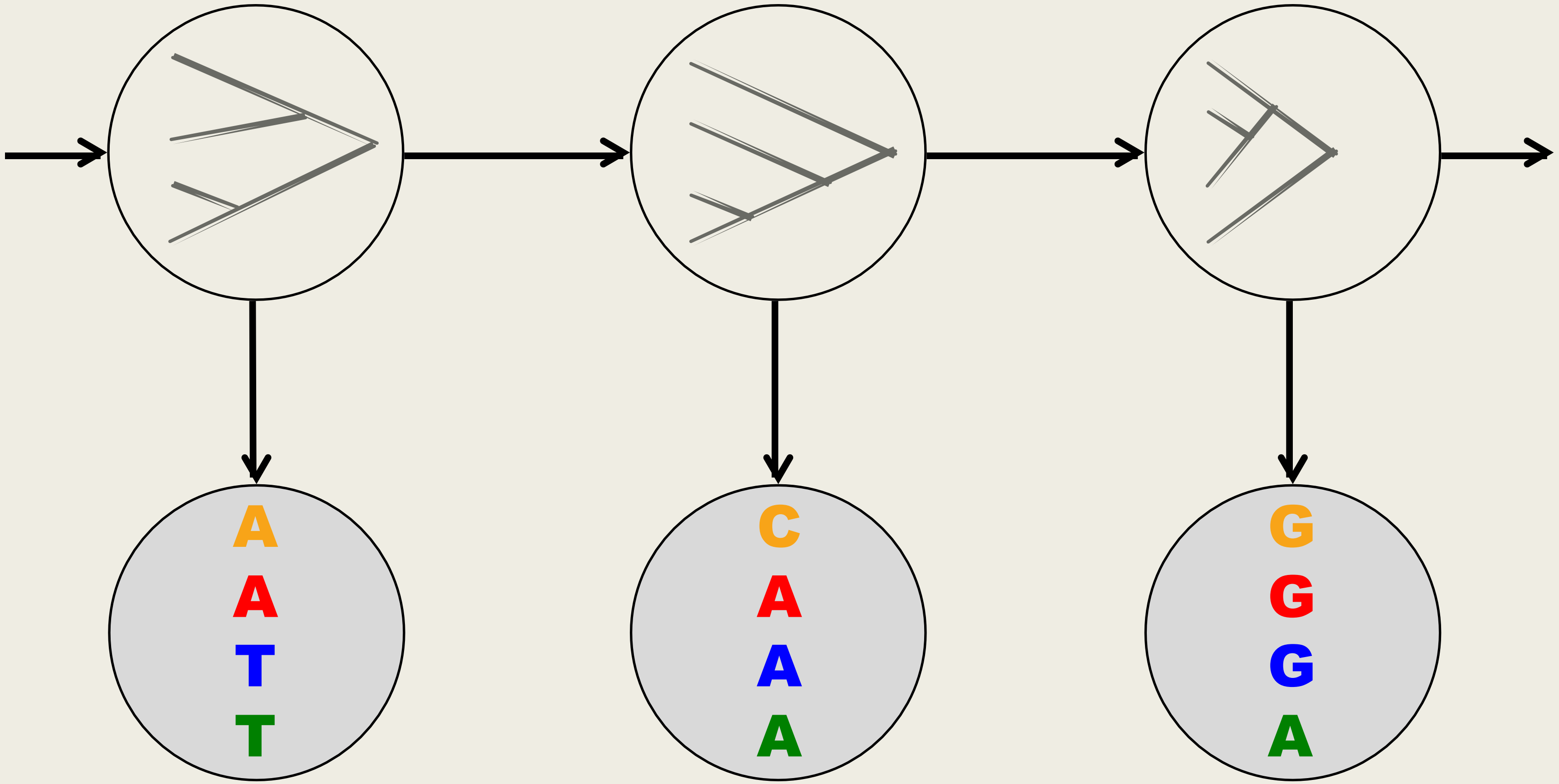
Tree changes along the genome!



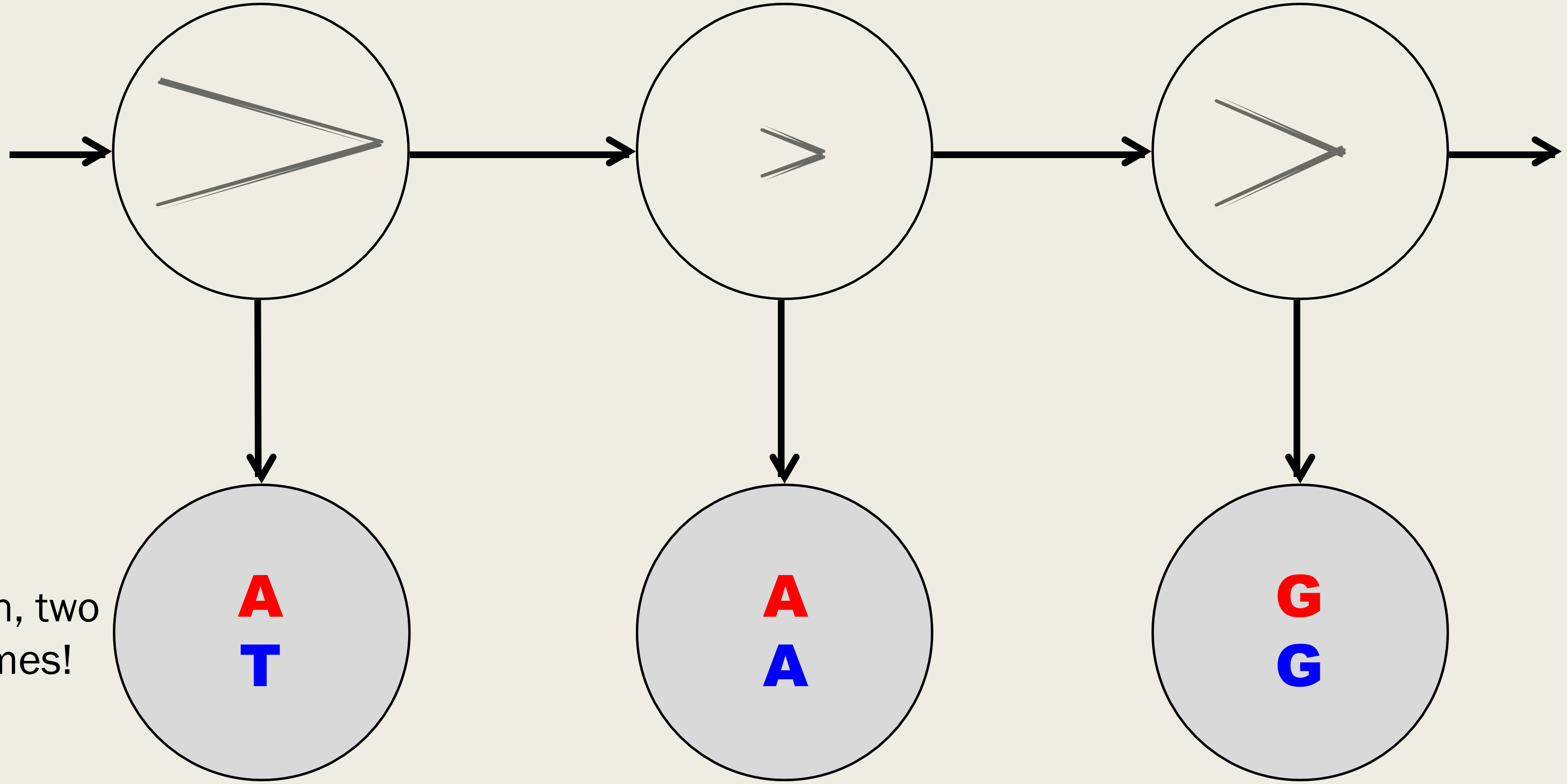
HMM observations: sequence data



HMM hidden states: the tree



Number of possible trees grows exponentially… just look at n=2



One person, two chromosomes!

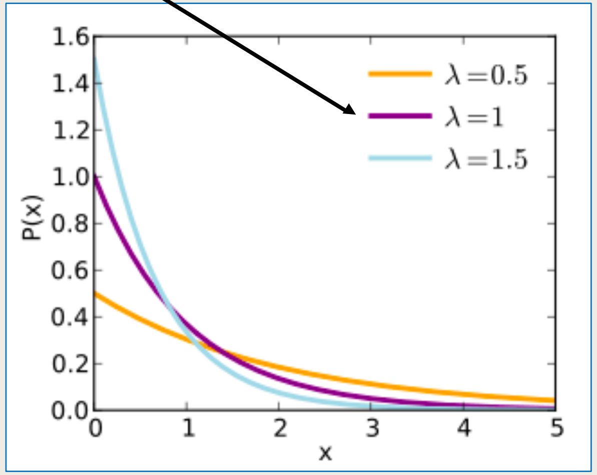
Now the hidden state becomes
the *time* of coalescence

PSMC: pairwise sequentially Markovian coalescent

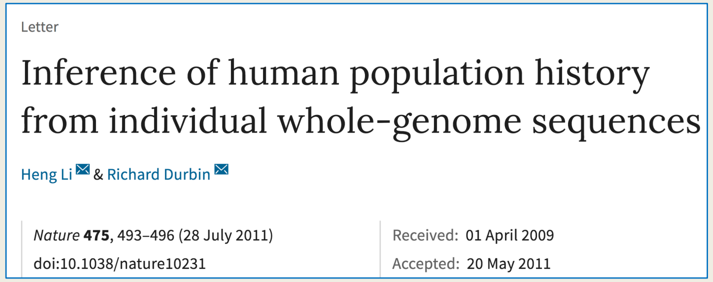
- The distribution of pairwise coalescence times should be exponential with parameter 1

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

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



- We can use this to reconstruct the population size change history over time!



Letter

Inference of human population history from individual whole-genome sequences

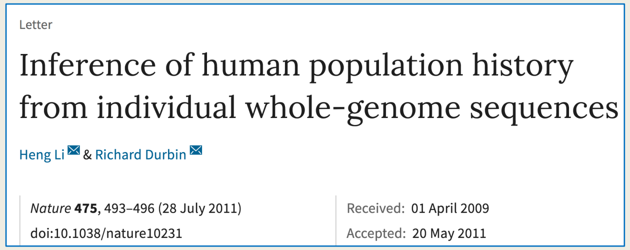
Heng Li & Richard Durbin

Nature **475**, 493–496 (28 July 2011)
doi:10.1038/nature10231

Received: 01 April 2009
Accepted: 20 May 2011

Image: wikipedia

PSMC: an HMM for two sequences



Letter

Inference of human population history from individual whole-genome sequences

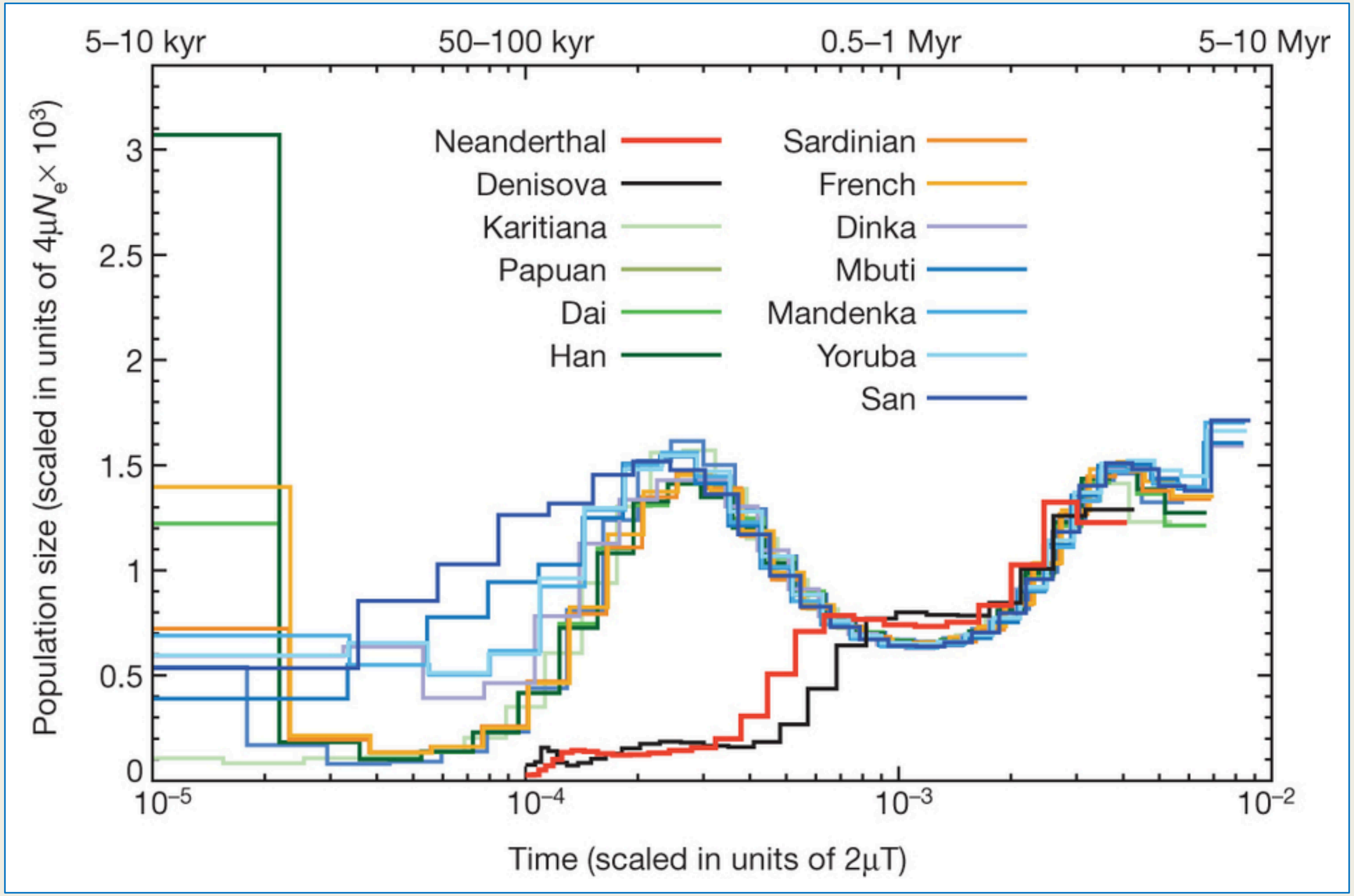
Heng Li & Richard Durbin

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)

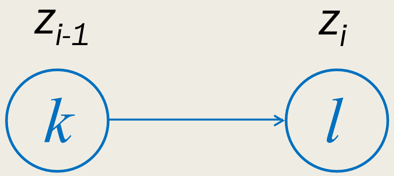
Recap Viterbi Algorithm

K = num hidden states, B = num emitted states

HMM definition

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

  (K x K matrix)



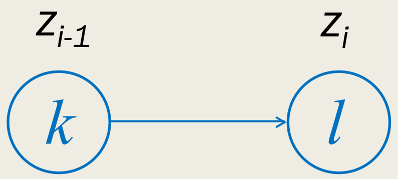
HMM definition

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

- Transition probabilities:

  (*K* x *K* matrix)

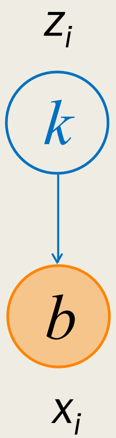
$$a_{kl} = P(z_i = l | z_{i-1} = k)$$



- Emission probabilities:

  (*K* x *B* matrix)

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



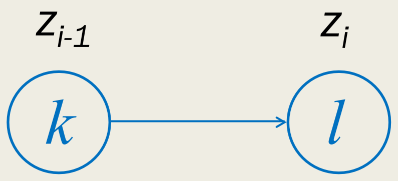
HMM definition

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

- Transition probabilities:

  (*K* x *K* matrix)

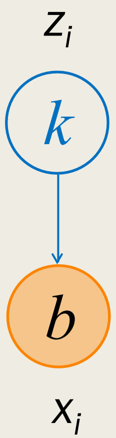
$$a_{kl} = P(z_i = l | z_{i-1} = k)$$



- Emission probabilities:

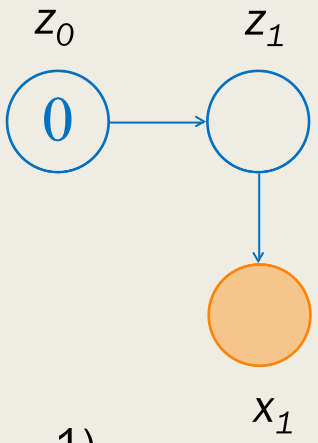
  (*K* x *B* matrix)

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



- A way to deal with the initial state

1) Special start state with no emission



1)

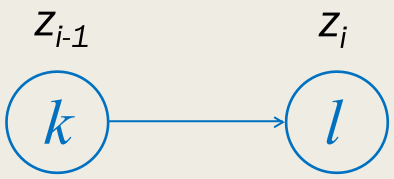
HMM definition

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

- Transition probabilities:

  (*K* x *K* matrix)

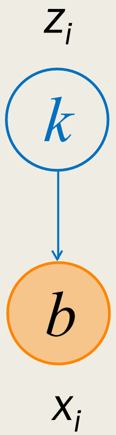
$$a_{kl} = P(z_i = l | z_{i-1} = k)$$



- Emission probabilities:

  (*K* x *B* matrix)

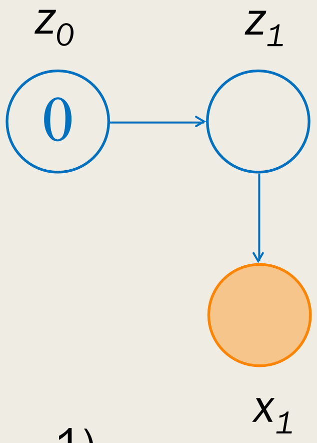
$$e_k(b) = P(x_i = b | z_i = k)$$



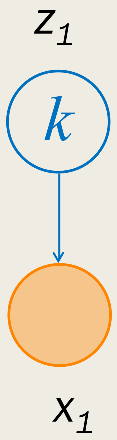
- A way to deal with the initial state

1) Special start state with no emission
2) Probability distribution over initial states

$$\pi_k = \text{probability of starting in state } k$$



1)



2)

$\pi_k = p(z_1 = k)$

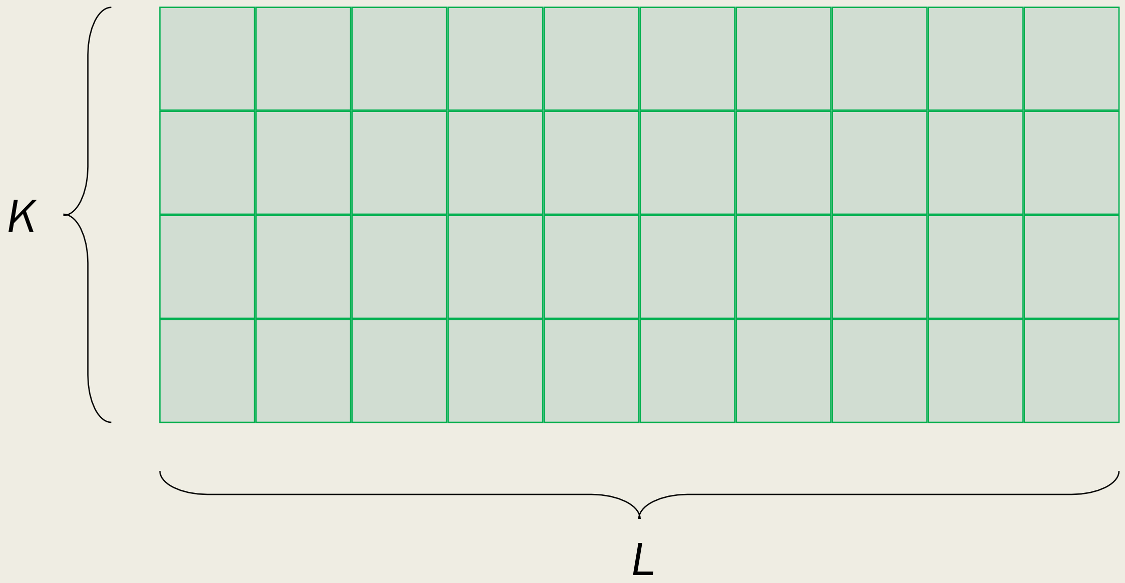
Viterbi Algorithm

- **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 $\boldsymbol{z}^*$

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



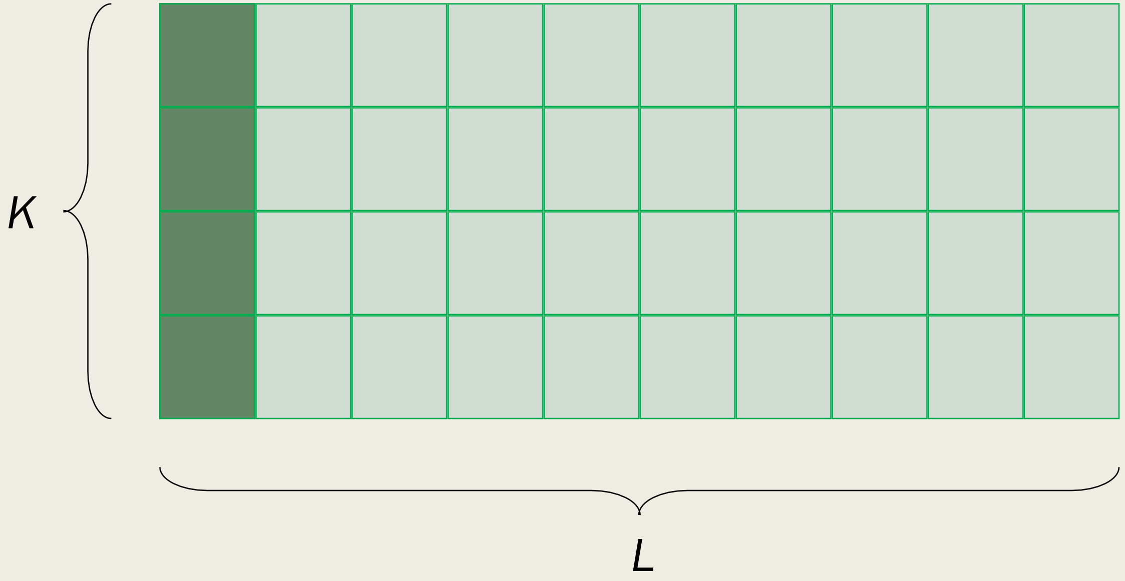
Viterbi Algorithm

- **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

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

(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 $\boldsymbol{z}^*$

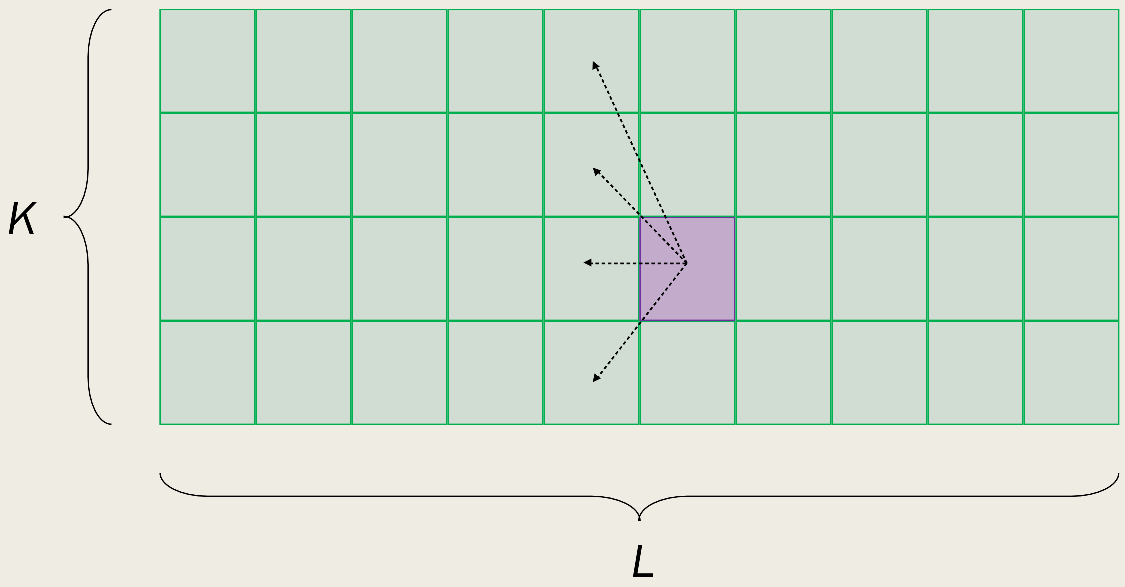
- **Initialization:** create a *K* x *L* matrix, this will be our dynamic programming (DP) table

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

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

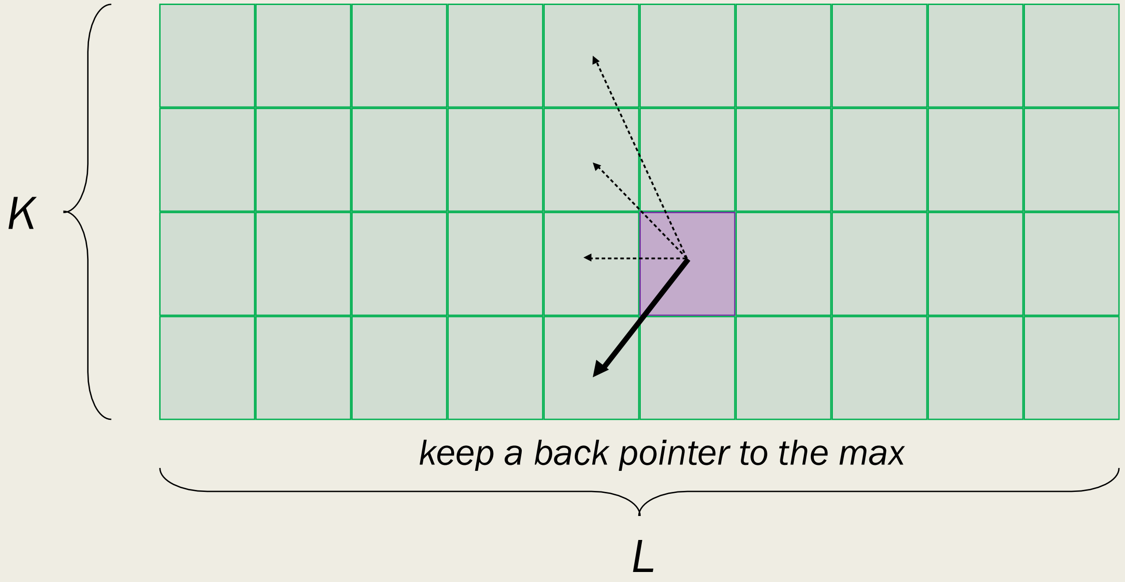
- **Initialization:** create a *K* x *L* matrix, this will be our dynamic programming (DP) table

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

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

- **Recursion:**

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



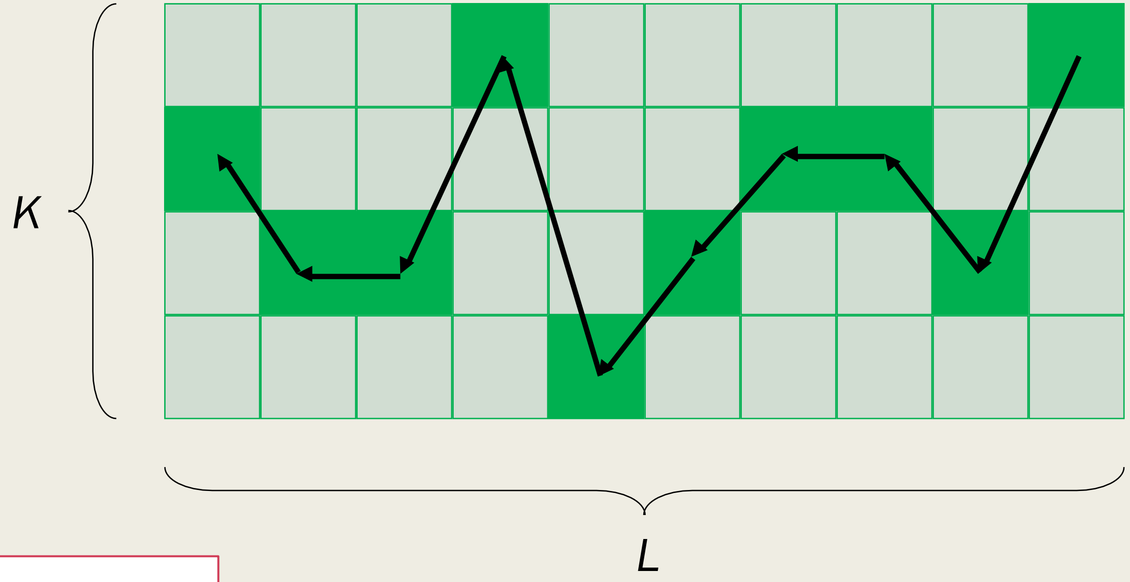
Viterbi Algorithm

- **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 ***z****

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

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

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

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 $\boldsymbol{z}^*$

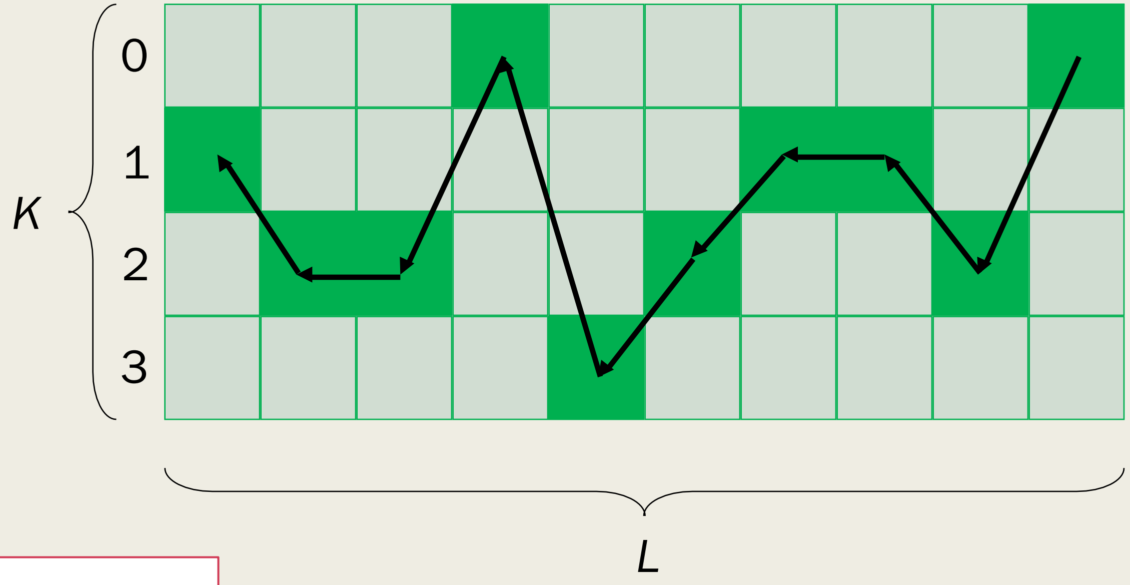
- **Initialization:** create a *K* x *L* matrix, this will be our dynamic programming (DP) table

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

(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:**

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

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

What is wrong with Viterbi?

What is wrong with Viterbi?

- Only one path! We can’t compute the probability of being in state *k* at step *i*

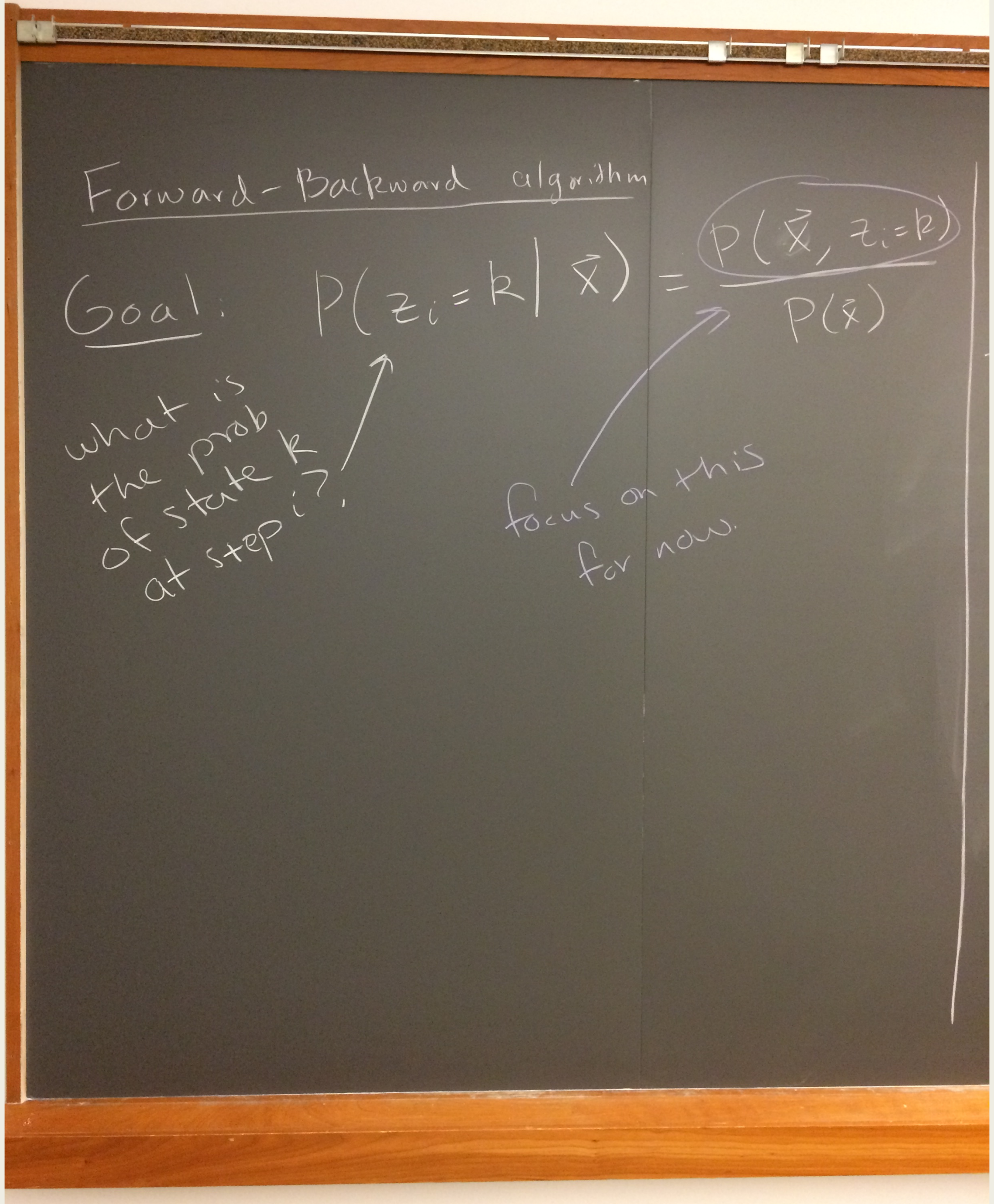
What is wrong with Viterbi?

- Only one path! We can't compute the probability of being in state *k* at step *i*
- We don't know if there are many possible paths, all with very similar probabilities

What is wrong with Viterbi?

- Only one path! We can’t compute the probability of being in state *k* at step *i*
- We don’t know if there are many possible paths, all with very similar probabilities
- And a note for later: we may not know the transition and emission probabilities

Forward-Backward algorithm

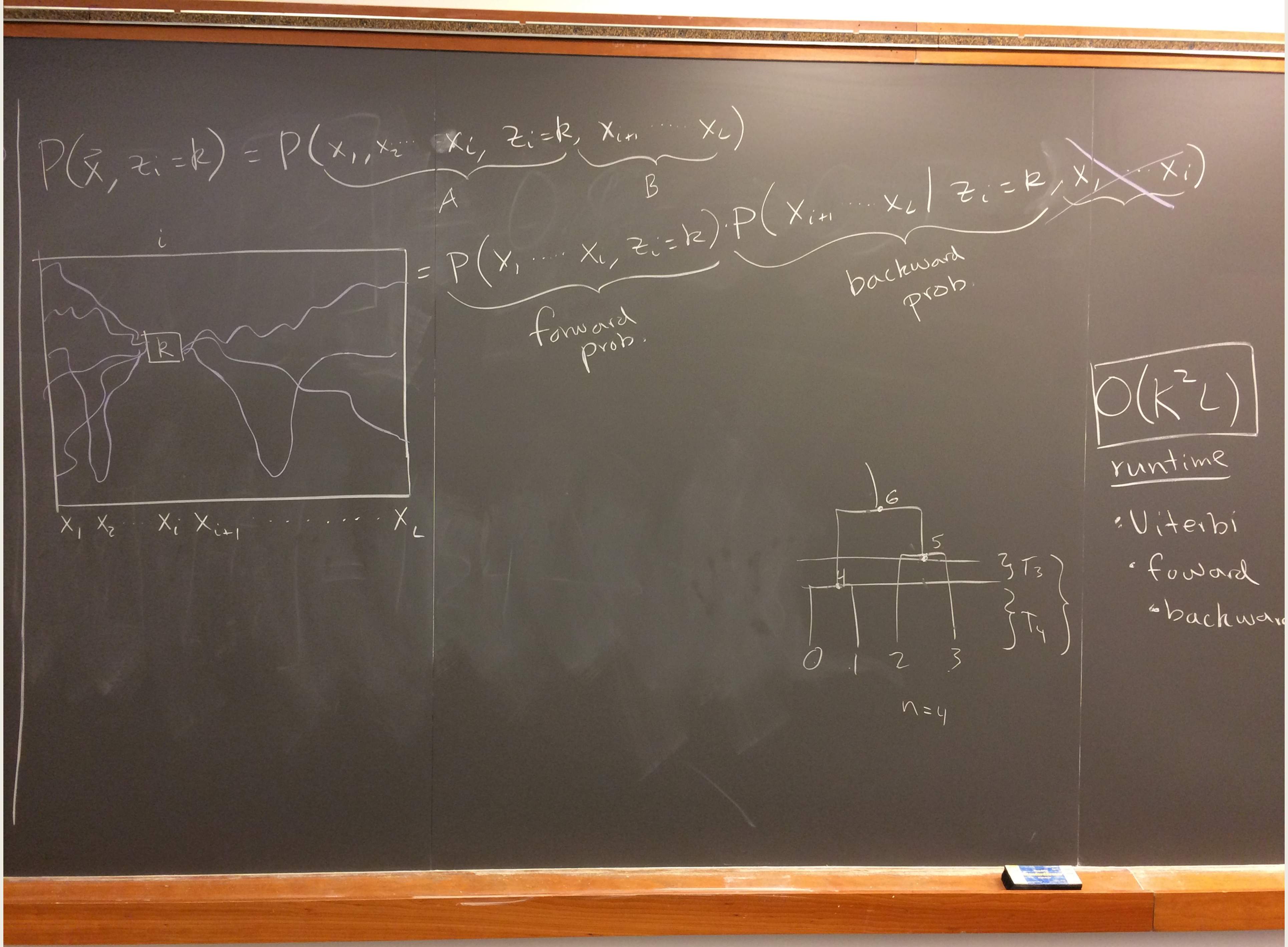


Forward-Backward algorithm

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

what is the prob of state k at step i?

focus on this for now.



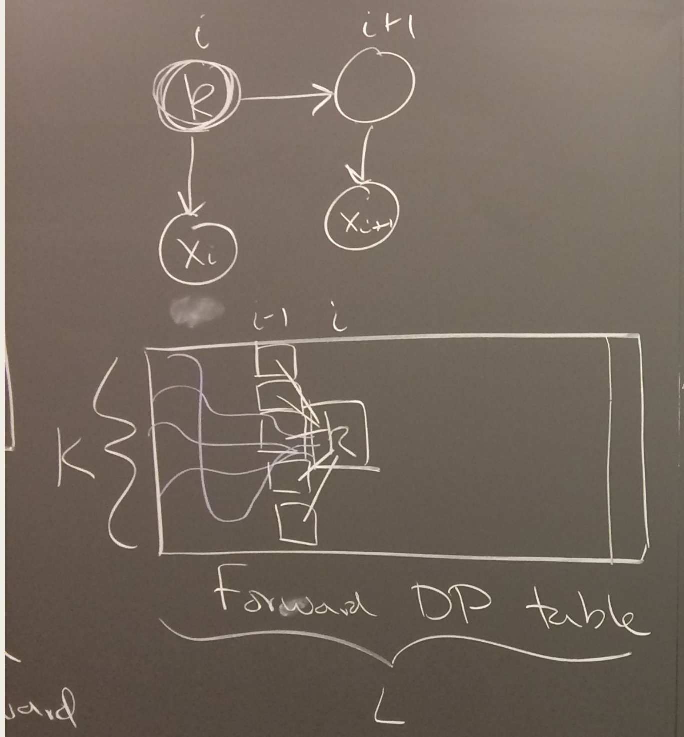
$$P(\vec{x}, z_i=k) = P(\underbrace{x_1, x_2 \cdots x_i, z_i=k}_{A}, \underbrace{x_{i+1} \cdots x_L}_{B})$$

$$= \underbrace{P(x_1 \cdots x_i, z_i=k)}_{\text{forward prob.}} \cdot \underbrace{P(x_{i+1} \cdots x_L \mid z_i=k}_{\text{backward prob.}}, \cancel{x_1 \cdots x_i})$$

$$\boxed{O(K^2L)}$$

runtime

- Viterbi
- forward
- backwa



Forward Algorithm

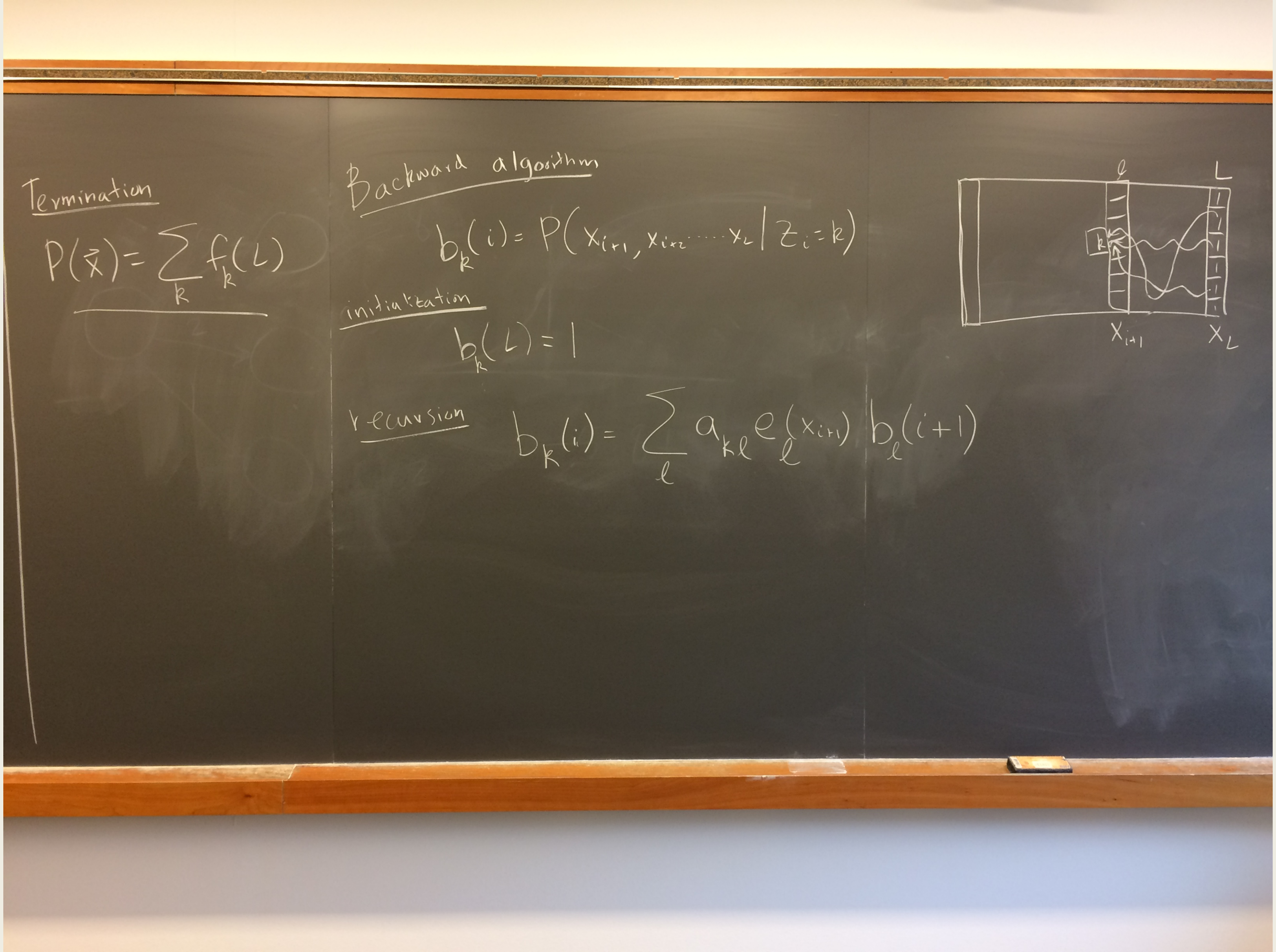
$f_k(i)$ = prob of observing $x_1 \cdots x_i$ & being in state k at step i

$f_k(i) = P(x_1 \cdots x_i, z_i = k)$

Recursion: $f_k(i) = \underbrace{e_k(x_i)}_{\text{emit } x_i}\sum_l \underbrace{f_l(i-1)}_{\text{prev step in state } l}\underbrace{a_{lk}}_{\text{transition from } l \to k}$

Initialization

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



Termination

$$P(\vec{x}) = \sum_{k} f_k(L)$$

Backward algorithm

$$b_k(i) = P(x_{i+1}, x_{i+2} \cdots x_L \mid z_i = k)$$

initialization

$$b_k(L) = 1$$

recursion

$$b_k(i) = \sum_{\ell} a_{k\ell} e_{\ell}(x_{i+1}) b_{\ell}(i+1)$$