

# Discussion Section 9

- HW 8
- Expectation-maximization
- Markov Chain Monte Carlo (MCMC)

# HW 8: Evolutionarily conserved segments

- ENCODE region 004 (chromosome 22)
- Multiple alignment of human, dog, and mouse
- 2 states: neutral (fast-evolving) and conserved (slow evolving)
- Emitted symbols are multiple alignment columns (e.g. 'AAT')
- Viterbi parse (no iteration)

# HW 8: Input

# HW 8: Input

- Original maf format
  - Sequences broken into alignment blocks based on the species included
  - <http://genome.ucsc.edu/FAQ/FAQformat.html#format5>

# HW 8: Input

- Original maf format
  - Sequences broken into alignment blocks based on the species included
  - <http://genome.ucsc.edu/FAQ/FAQformat.html#format5>
- Homework file format
  - Only 3 species
  - Gaps in human sequence and ambiguous bases replaced with 'A' for simplicity

# HW 8: Setting parameters

# HW 8: Setting parameters

- Emission probabilities
  - Neutral state: observed frequencies in neutral data set
  - Conserved state: observed frequencies in functional data set

# HW 8: Setting parameters

- Emission probabilities
  - Neutral state: observed frequencies in neutral data set
  - Conserved state: observed frequencies in functional data set
- Transition probabilities
  - Given in assignment, more likely to go from conserved to neutral



# HW 8: Setting parameters

- Emission probabilities
  - Neutral state: observed frequencies in neutral data set
  - Conserved state: observed frequencies in functional data set
- Transition probabilities
  - Given in assignment, more likely to go from conserved to neutral
- Initial probabilities
  - Given in assignment, more likely to start in neutral state

# HW 8: Output

# HW 8: Output

- Parameter values
  - Include the emission probabilities you calculated from the neutral and conserved data sets

# HW 8: Output

- Parameter values
  - Include the emission probabilities you calculated from the neutral and conserved data sets
- State and segment histograms

# HW 8: Output

- Parameter values
  - Include the emission probabilities you calculated from the neutral and conserved data sets
- State and segment histograms
- Coordinates of 10 longest conserved segments

# HW 8: Output

- Parameter values
  - Include the emission probabilities you calculated from the neutral and conserved data sets
- State and segment histograms
- Coordinates of 10 longest conserved segments
- Brief annotations for the 5 longest conserved segments (look at UCSC genome browser)

# Expectation-maximization (EM) algorithm

# Expectation-maximization (EM) algorithm

- General algorithm for maximum likelihood estimation with “missing data”



# Expectation-maximization (EM) algorithm

- General algorithm for maximum likelihood estimation with “missing data”
  - Clustering
  - Machine learning
  - Computer vision
  - Natural language processing

# Expectation-maximization (EM) algorithm

- Goal is to find parameters that maximize the log likelihood of a particular dataset

# Expectation-maximization (EM) algorithm

- Goal is to find parameters that maximize the log likelihood of a particular dataset
  - $x$  = observed data (sequence)

# Expectation-maximization (EM) algorithm

- Goal is to find parameters that maximize the log likelihood of a particular dataset
  - $x$  = observed data (sequence)
  - $y$  = latent/missing data (states)

# Expectation-maximization (EM) algorithm

- Goal is to find parameters that maximize the log likelihood of a particular dataset
  - $x$  = observed data (sequence)
  - $y$  = latent/missing data (states)
  - $\theta$  = parameters (initiation, transition, emission probabilities)

# Expectation-maximization (EM) algorithm

- Goal is to find parameters that maximize the log likelihood of a particular dataset
  - $x$  = observed data (sequence)
  - $y$  = latent/missing data (states)
  - $\theta$  = parameters (initiation, transition, emission probabilities)

$$\log P(x|\theta) = \log \sum_y P(x, y|\theta)$$

# Expectation-maximization (EM) algorithm

- Goal is to find parameters that maximize the log likelihood of a particular dataset
  - $x$  = observed data (sequence)
  - $y$  = latent/missing data (states)
  - $\theta$  = parameters (initiation, transition, emission probabilities)

$$\log P(x|\theta) = \log \sum_y P(x, y|\theta)$$

- Iteratively determine better set of parameters  $\theta$

# Expectation-maximization (EM) algorithm

- Goal is to find parameters that maximize the log likelihood of a particular dataset
  - $x$  = observed data (sequence)
  - $y$  = latent/missing data (states)
  - $\theta$  = parameters (initiation, transition, emission probabilities)

$$\log P(x|\theta) = \log \sum_y P(x, y|\theta)$$

- Iteratively determine better set of parameters  $\theta$

$$\theta^t \rightarrow \theta^{t+1}$$



# Expectation-maximization (EM) algorithm

- Goal is to find parameters that maximize the log likelihood of a particular dataset
  - It's difficult to directly maximize

$$\log P(x|\theta) = \log \sum_y P(x, y|\theta)$$

# Expectation-maximization (EM) algorithm

- Goal is to find parameters that maximize the log likelihood of a particular dataset
  - It's difficult to directly maximize

$$\log P(x|\theta) = \log \sum_y P(x, y|\theta)$$

- Instead, we want a simpler function based on current estimates for  $\theta$  to find a better  $\theta$  estimate

# Expectation-maximization (EM) algorithm

- Goal is to find parameters that maximize the log likelihood of a particular dataset

$$\log P(x|\theta) = \log \sum_y P(x, y|\theta)$$

# Expectation-maximization (EM) algorithm

- Goal is to find parameters that maximize the log likelihood of a particular dataset

$$\log P(x|\theta) = \log \sum_y P(x, y|\theta)$$

- With  $P(x, y|\theta) = P(y|x, \theta)P(x|\theta)$  we can rewrite the log likelihood as

# Expectation-maximization (EM) algorithm

- Goal is to find parameters that maximize the log likelihood of a particular dataset

$$\log P(x|\theta) = \log \sum_y P(x, y|\theta)$$

- With  $P(x, y|\theta) = P(y|x, \theta)P(x|\theta)$  we can rewrite the log likelihood as

$$\log P(x|\theta) = \log P(x, y|\theta) - \log P(y|x, \theta)$$

# Expectation-maximization (EM) algorithm

- Goal is to find parameters that maximize the log likelihood of a particular dataset

$$\log P(x|\theta) = \log \sum_y P(x, y|\theta)$$

- With  $P(x, y|\theta) = P(y|x, \theta)P(x|\theta)$  we can rewrite the log likelihood as

$$\log P(x|\theta) = \log P(x, y|\theta) - \log P(y|x, \theta)$$

- Multiply both sides by  $P(y|x, \theta^t)$  and integrate over  $y$  (basically multiplying by 1 to both sides)

# Expectation-maximization (EM) algorithm

- Goal is to find parameters that maximize the log likelihood of a particular dataset

$$\log P(x|\theta) = \log \sum_y P(x, y|\theta)$$

- With  $P(x, y|\theta) = P(y|x, \theta)P(x|\theta)$  we can rewrite the log likelihood as

$$\log P(x|\theta) = \log P(x, y|\theta) - \log P(y|x, \theta)$$

- Multiply both sides by  $P(y|x, \theta^t)$  and integrate over  $y$  (basically multiplying by 1 to both sides)

$$\log P(x|\theta) = \sum_y P(y|x, \theta^t) \log P(x, y|\theta) - \sum_y P(y|x, \theta^t) \log P(y|x, \theta)$$

# Expectation-maximization (EM) algorithm

$$\log P(x|\theta) = \sum_y P(y|x, \theta^t) \log P(x, y|\theta) - \sum_y P(y|x, \theta^t) \log P(y|x, \theta)$$

t



# Expectation-maximization (EM) algorithm

$$\log P(x|\theta) = \sum_y P(y|x, \theta^t) \log P(x, y|\theta) - \sum_y P(y|x, \theta^t) \log P(y|x, \theta)$$

- Define the function Q

t

# Expectation-maximization (EM) algorithm

$$\log P(x|\theta) = \sum_y P(y|x, \theta^t) \log P(x, y|\theta) - \sum_y P(y|x, \theta^t) \log P(y|x, \theta)$$

- Define the function Q

$$Q(\theta|\theta^t) = \sum_y P(y|x, \theta^t) \log P(x, y|\theta)$$

t

# Expectation-maximization (EM) algorithm

$$\log P(x|\theta) = \sum_y P(y|x, \theta^t) \log P(x, y|\theta) - \sum_y P(y|x, \theta^t) \log P(y|x, \theta)$$

- Define the function Q

$$Q(\theta|\theta^t) = \sum_y P(y|x, \theta^t) \log P(x, y|\theta)$$

t

# Expectation-maximization (EM) algorithm

$$\log P(x|\theta) = \sum_y P(y|x, \theta^t) \log P(x, y|\theta) - \sum_y P(y|x, \theta^t) \log P(y|x, \theta)$$

- Define the function  $Q$

$$Q(\theta|\theta^t) = \sum_y P(y|x, \theta^t) \log P(x, y|\theta)$$

- We can then subtract two log likelihoods (given  $\theta$  and  $\theta^t$ ) to get

$$\log P(x|\theta) - \log P(x|\theta^t) = Q(\theta|\theta^t) - Q(\theta^t|\theta^t) + \sum_y P(y|x, \theta^t) \log \frac{P(y|x, \theta^t)}{P(y|x, \theta)}$$

# Expectation-maximization (EM) algorithm

$$\log P(x|\theta) - \log P(x|\theta^t) = Q(\theta|\theta^t) - Q(\theta^t|\theta^t) + \sum_y P(y|x, \theta^t) \log \frac{P(y|x, \theta^t)}{P(y|x, \theta)}$$

# Expectation-maximization (EM) algorithm

$$\log P(x|\theta) - \log P(x|\theta^t) = Q(\theta|\theta^t) - Q(\theta^t|\theta^t) + \sum_y P(y|x, \theta^t) \log \frac{P(y|x, \theta^t)}{P(y|x, \theta)}$$

- Gibb's inequality tells us that the rightmost term is positive, so we can drop it to find

$$\log P(x|\theta) - \log P(x|\theta^t) \geq Q(\theta|\theta^t) - Q(\theta^t|\theta^t)$$

# Expectation-maximization (EM) algorithm

$$\log P(x|\theta) - \log P(x|\theta^t) = Q(\theta|\theta^t) - Q(\theta^t|\theta^t) + \sum_y P(y|x, \theta^t) \log \frac{P(y|x, \theta^t)}{P(y|x, \theta)}$$

- Gibb's inequality tells us that the rightmost term is positive, so we can drop it to find

$$\log P(x|\theta) - \log P(x|\theta^t) \geq Q(\theta|\theta^t) - Q(\theta^t|\theta^t)$$

- Thus, by maximizing the value of  $Q(\theta|\theta^t)$ , we can maximize the increase in log likelihood

# Expectation-maximization (EM) algorithm

$$\log P(x|\theta) - \log P(x|\theta^t) = Q(\theta|\theta^t) - Q(\theta^t|\theta^t) + \sum_y P(y|x, \theta^t) \log \frac{P(y|x, \theta^t)}{P(y|x, \theta)}$$

- Gibb's inequality tells us that the rightmost term is positive, so we can drop it to find

$$\log P(x|\theta) - \log P(x|\theta^t) \geq Q(\theta|\theta^t) - Q(\theta^t|\theta^t)$$

- Thus, by maximizing the value of  $Q(\theta|\theta^t)$ , we can maximize the increase in log likelihood

$$\theta^{t+1} = \operatorname{argmax}_{\theta} Q(\theta|\theta^t)$$



# Expectation-maximization (EM) algorithm

- Basic EM algorithm

# Expectation-maximization (EM) algorithm

- Basic EM algorithm
  - Expectation step: calculate Q function

$$Q(\theta|\theta^t) = \sum_y P(y|x, \theta^t) \log P(x, y|\theta)$$

# Expectation-maximization (EM) algorithm

- Basic EM algorithm
  - Expectation step: calculate Q function

$$Q(\theta|\theta^t) = \sum_y P(y|x, \theta^t) \log P(x, y|\theta)$$

- Maximization step: Choose new parameters to maximize Q function

$$\theta^{t+1} = \operatorname{argmax}_{\theta} Q(\theta|\theta^t)$$

# Baum-Welch is an EM algorithm

- Latent data are states at each position

# Baum-Welch is an EM algorithm

- Latent data are states at each position
  - Maximize over all possible sequences of states

# Baum-Welch is an EM algorithm

- Latent data are states at each position
  - Maximize over all possible sequences of states

$$\log P(x|\theta) = \sum_{\pi} \log P(x, \pi|\theta)$$

# Baum-Welch is an EM algorithm

- Latent data are states at each position
  - Maximize over all possible sequences of states

$$\log P(x|\theta) = \sum_{\pi} \log P(x, \pi|\theta)$$

$$Q(\theta|\theta^t) = \sum_{\pi} P(\pi|x, \theta^t) \log P(x, \pi|\theta)$$

# Baum-Welch is an EM algorithm

- Latent data are states at each position
  - Maximize over all possible sequences of states

$$\log P(x|\theta) = \sum_{\pi} \log P(x, \pi|\theta)$$

$$Q(\theta|\theta^t) = \sum_{\pi} P(\pi|x, \theta^t) \log P(x, \pi|\theta)$$

- E step: calculate expectations given emission and transition probabilities



# Baum-Welch is an EM algorithm

- Latent data are states at each position
  - Maximize over all possible sequences of states

$$\log P(x|\theta) = \sum_{\pi} \log P(x, \pi|\theta)$$

$$Q(\theta|\theta^t) = \sum_{\pi} P(\pi|x, \theta^t) \log P(x, \pi|\theta)$$

- E step: calculate expectations given emission and transition probabilities
- M step: reestimate emission and transition probabilities

# Markov Chain Monte Carlo (MCMC) methods

# Markov Chain Monte Carlo (MCMC) methods

- Combination of

# Markov Chain Monte Carlo (MCMC) methods

- Combination of
  - Markov chains and...

# Markov Chain Monte Carlo (MCMC) methods

- Combination of
  - Markov chains and...
  - Monte Carlo methods

# Markov chain



Andrey Markov

# Markov chain

- Similar to Hidden Markov Models, but all information is observed



Andrey Markov

# Markov chain

- Similar to Hidden Markov Models, but all information is observed
- Markov property – current state only depends on previous state



Andrey Markov

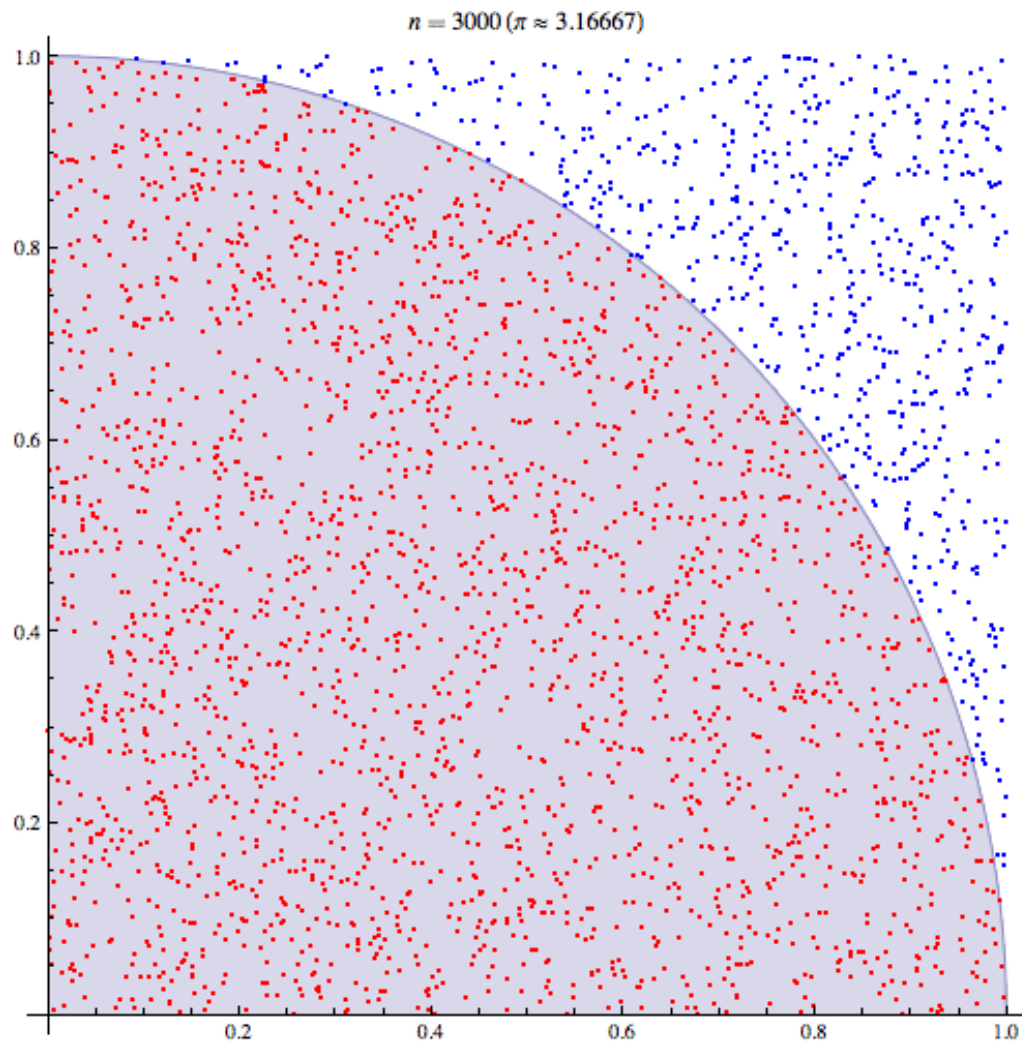


# Monte Carlo methods

- Use random sampling to obtain numerical results

# Monte Carlo methods

- Use random sampling to obtain numerical results



# Markov Chain Monte Carlo

- Basic idea:
  - Random sampling of a complex probability distribution using a Markov chain

# Markov Chain Monte Carlo

- Basic idea:
  - Random sampling of a complex probability distribution using a Markov chain
    - Next sample is related to previous sample (Markov property)

# Markov Chain Monte Carlo

- Basic idea:
  - Random sampling of a complex probability distribution using a Markov chain
    - Next sample is related to previous sample (Markov property)
- Can be used to compute integrals and expected values

# Metropolis-Hastings algorithm (MCMC)

# Metropolis-Hastings algorithm (MCMC)

- At each step, pick a candidate for the next sample value based on the current sample value

# Metropolis-Hastings algorithm (MCMC)

- At each step, pick a candidate for the next sample value based on the current sample value
- With some probability, accept the candidate and use it in the next iteration



# Metropolis-Hastings algorithm (MCMC)

- At each step, pick a candidate for the next sample value based on the current sample value
- With some probability, accept the candidate and use it in the next iteration
- How do you know when to accept?

# Metropolis-Hastings algorithm (MCMC)

- At each step, pick a candidate for the next sample value based on the current sample value
- With some probability, accept the candidate and use it in the next iteration
- How do you know when to accept?
  - If higher probability, accept

# Metropolis-Hastings algorithm (MCMC)

- At each step, pick a candidate for the next sample value based on the current sample value
- With some probability, accept the candidate and use it in the next iteration
- How do you know when to accept?
  - If higher probability, accept
  - If lower probability, accept based on ratio of probabilities