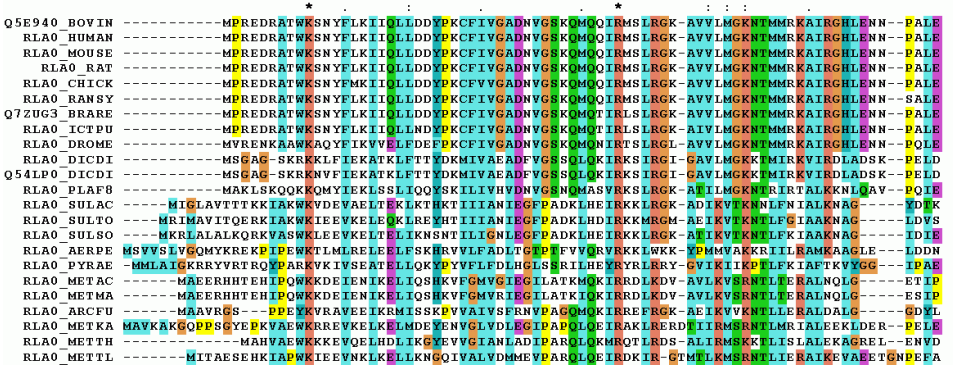
11. Multiple Sequence Alignment

Outline

Multiple sequence alignment

- multiple sequence alignment (MSA) is sequence alignment of three or more biological sequences such as DNA, RNA, or protein
- an example protein MSA



Multiple sequence alignment

- each row of the MSA corresponds to the sequence of a specific protein
- each column of the MSA corresponds to a position in the sequence
- dash symbol means the sequence does not have an amino acid aligned at that position
- protein sequences are in the same MSA are evolutionarily related: they are homologous
- homologous sequences are derived from a common ancestor, so they are similar in sequence, structure, and function

Multiple sequence alignment

- MSA of a protein contains more information than a single sequence
- can be used to identify conserved regions in the protein
- conserved regions are often important for the protein's function
- used to infer the evolutionary relationships between the sequences
- used to search for homologous sequences in a database
- used to predict the structure and function of a protein

Multiple sequence alignment

- multiple algorithms exist for constructing MSAs
- most algorithms require a query sequence and a database of sequences
- they iteratively search for homologous sequences in the database and align them
- example algorithms: Clustal Omega, MUSCLE

Protein family

- a protein family is a group of proteins that share a common evolutionary origin
- members of a protein family are homologous and have similar sequences, structures, and functions
- sequences of a protein family are aligned to create a multiple sequence alignment
- the Pfam database is a collection of protein families

Outline

Probabilistic models in general

- data: $\{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}$, where $x^{(i)}$ is a data sample and could be a scale or a vector
- a probabilistic model of the data defines a probability distribution $P(x;\theta)$
- θ is a set of parameters that define the model
- assumes that the observed data are generated by the model, i.e., the data are samples from the distribution $P(x;\theta^*)$
- θ^* is the true parameter value of the model
- learning the model means estimating the parameters θ from the data

A simple example of probabilistic model

- observed data: $0.43, 2.49, -1.91, 0.29, -2.1, 0.44$
- model:

$$p(x;\theta) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- $\theta = (\mu, \sigma^2)$ is the set of parameters
- how to estimate θ from the data?

Maximum likelihood estimation

- a general approach to estimate the parameters of a probabilistic model based on the observed data
- estimates the parameters θ by maximizing the likelihood function

$$L(\theta) = P(x^{(1)}, x^{(2)}, \ldots, x^{(N)}; \theta) = \prod_{i=1}^{N} P(x^{(i)}; \theta)$$

- the estimate $\hat{\theta} = \arg\max_\theta L(\theta)$
- it is often easier to maximize the log-likelihood function

$$\ell(\theta) = \log L(\theta) = \sum_{i=1}^{N} \log P(x^{(i)}; \theta)$$

The probability distribution $P(x;\theta)$

- an assumption about the data and an approximation of the true distribution
- several factors influence the choice of the distribution
  - the nature of the data
  - the complexity of the model
  - the computational cost of estimating the parameters
  - the interpretability of the model
  - the need of sampling from the distribution or computing the likelihood
- by choosing a distribution with inherent structures, we could infer the structures from data

Example $P(x, \theta)$ with varying complexity and structrues

- a Gaussian distribution
- a mixture of Gaussians
- a Gaussian process
- a hidden Markov model
- the Boltzmann machine
- large language models
- autoregressive probabilistic models
- variational autoencoders
- restricted Boltzmann machines
- the Ising model
- the Potts model
- large language models of proteins

Probabilistic models of MSA

- a MSA is a collection of sequences: $\{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}$, where $x^{(i)}$ is a sequence of amino acids
- a probabilistic model of MSA defines a probability distribution $P(x; \theta)$
- $\{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}$ are assumed to be samples from the distribution $P(x; \theta^*)$
- two examples of probabilistic models of MSA
  - MSA profile (position independent model)
  - Potts model (directed coupling analysis)

MSA profile

- assumes that amino acids at each position are independent
- the probability of a sequence is the product of the probabilities of each amino acid at each position

$$P(x;\theta) = \prod_{k=1}^{L} P(x_k;\theta_k)$$

- L is the length of the sequence and θ_k is the set of parameters for the k-th position
- $P(x_k;\theta_k)$ is the probability distribution of amino acid types at the k-th position

MSA profile

- assume there are no gaps in the MSA, then x_k has 20 possible values (20 amino acids)
- the probability distribution $P(x_k; \theta_k)$ is a multinomial distribution

$$P(x_k = i; \theta_k) = \theta_{i,k}$$

- $\theta_{i,k}$ is the probability of the k-th position being the i-th amino acid and $\sum_{i=1}^{20} \theta_{i,k} = 1$
- estimate $\theta_{i,k}$ with MLE and is equal to the frequency of each amino acid at each position

$$\hat{\theta}_{i,k} = \frac{N_{i,k}}{N}$$

- $N_{i,k}$ is the number of times the i-th amino acid appears at the k-th position in the MSA

MSA profile

- is a matrix of size $20 \times L$

$$\begin{pmatrix} \theta_{1,1} & \theta_{1,2} & \dots & \theta_{1,L} \\ \theta_{2,1} & \theta_{2,2} & \dots & \theta_{2,L} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{20,1} & \theta_{20,2} & \dots & \theta_{20,L} \end{pmatrix}$$

- used by many ML methods as input features
- captures more information about a protein family than a single sequence
- easy to sample sequences from the distribution and compute the likelihood of a sequence
- ignores dependency between positions

Computing the likelihood of a sequence and sampling from MSA profile

- the likelihood of a sequence x is given by

$$P(x;\theta) = \prod_{k=1}^{L} P(x_k;\theta_k) = \prod_{k=1}^{L} \theta_{x_k,k}$$

- to sample a sequence from the MSA profile, we can sample each position independently
- for each position k, sample an amino acid type x_k from the multinomial distribution $P(x_k;\theta_k)$
- the sampled sequence is $x = (x_1, x_2, \ldots, x_L)$

Potts model

- a more complex model that captures the dependency between positions
- assumes that the probability of a sequence is given by a Boltzmann distribution

$$P(x;\theta) = \frac{1}{Z(\theta)} e^{-E(x;\theta)}$$

- $E(x;\theta)$ is the "energy" of the sequence and $Z(\theta)$ is the partition function

$$Z(\theta) = \sum_{x} e^{-E(x;\theta)}$$

- the sum in the partition function is over all possible sequences
- how many possible sequences are there for a given length L?

Potts model

- the energy function is given by

$$E(x;\theta) = \sum_{k=1}^{L} h_k(x_k) + \sum_{k=1}^{L-1} \sum_{l=k+1}^{L} J_{kl}(x_k, x_l)$$

- $h_k(x_k)$ is the "field" at position k
- $h_k(x_k)$ captures preferences of the amino acid types at position k
- $J_{kl}(x_k, x_l)$ is the "coupling" between positions k and l
- $J_{kl}(x_k, x_l)$ captures the dependency between the amino acid types at positions k and l

The field term

- assume there are no gaps in the MSA, then x_k has 20 possible values (20 amino acids)
- to specify the field term, we need to define $h_k(x_k)$ for each amino acid type
- let $h_k(x_k = i) = h_{i,k}$, the field term at the k-th position is given by

$$h_k(x_k) = \sum_{i=1}^{20} h_{i,k} \cdot \mathbb{1}\{x_k = i\}$$

- the total field term is given by

$$\sum_{k=1}^{L} h_k(x_k) = \sum_{k=1}^{L} \sum_{i=1}^{20} h_{i,k} \cdot \mathbb{1}\{x_k = i\}$$

The coupling term

- to specify it, we need to define $J_{kl}(x_k, x_l)$ for each pair of amino acid types
- let $J_{kl}(x_k = i, x_l = j) = J_{i,j}^{k,l}$, the coupling term at positions k and l is given by

$$J_{kl}(x_k, x_l) = \sum_{i=1}^{20} \sum_{j=1}^{20} J_{i,j}^{k,l} \cdot \mathbb{1}\{x_k = i\} \cdot \mathbb{1}\{x_l = j\}$$

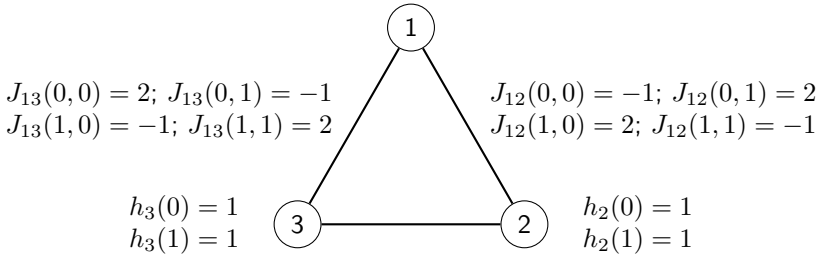
- the total coupling term is given by

$$\sum_{k=1}^{L-1} \sum_{l=k+11}^{L} J_{kl}(x_k, x_l) = \sum_{k=1}^{L-1} \sum_{l=k+1}^{L} \sum_{i=1}^{20} \sum_{j=1}^{20} J_{i,j}^{k,l} \cdot \mathbb{1}\{x_k = i\} \cdot \mathbb{1}\{x_l = j\}$$

An example Potts model

- a fully connected undirected probabilistic graphical model

$$h_1(0) = 2$$
$$h_1(1) = 1$$



$$J_{23}(0,0) = 0;\ J_{23}(0,1) = 0$$
$$J_{23}(1,0) = 0;\ J_{23}(1,1) = 0$$

An example Potts model

state (x_1, x_2, x_3)	energy	probability
(0, 0, 0)	5	0.012
(0, 0, 1)	2	0.244
(0, 1, 0)	8	0.001
(0, 1, 1)	5	0.012
(1, 0, 0)	4	0.033
(1, 0, 1)	7	0.002
(1, 1, 0)	1	0.663
(1, 1, 1)	4	0.033

- compute the energy of each state

$$
\begin{aligned}
E((0,0,0)) &= h_1(0) + h_2(0) + h_3(0) \\
&\quad + J_{12}(0,0) + J_{13}(0,0) + J_{23}(0,0) \\
&= 2 + 1 + 1 - 1 + 2 + 0 = 5 \\
E((1,1,0)) &= h_1(1) + h_2(1) + h_3(1) \\
&\quad + J_{12}(1,1) + J_{13}(1,1) + J_{23}(1,1) \\
&= 1 + 1 + 1 - 1 - 1 + 0 = 1 \\
&\cdots
\end{aligned}
$$

Correlation caused by indirect coupling

- the marginal probability of x_2, x_3 is

	$x_2 = 0$	$x_2 = 1$
$x_3 = 0$	0.045	0.664
$x_3 = 1$	0.246	0.045

- when $x_2 = 0$, x_3 is more likely to be 1; when $x_2 = 1$, x_3 is more likely to be 0
- x_2 and x_3 are correlated
- but the coupling term $J_{23}(x_2, x_3)$ is zero
- the correlation is caused by the indirect coupling between x_2 and x_3 through x_1

Learning the Potts model of a MSA

- because the Potts model is a probabilistic model, we can use MLE to estimate the parameters
- given a MSA $\{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}$, the average log-likelihood function is

$$\ell(\theta) = \frac{1}{N}\sum_{i=1}^{N} \log P(x^{(i)};\theta) = \frac{1}{N}\sum_{i=1}^{N}\Big(-E(x^{(i)};\theta) - \log Z(\theta)\Big)$$

- the gradient of the log-likelihood function is

$$\nabla_\theta \ell(\theta) = \underbrace{-\frac{1}{N}\sum_{n=1}^{N}\nabla_\theta E(x^{(n)};\theta)}_{\text{the first term}} + \underbrace{\nabla_\theta \log Z(\theta)}_{\text{the second term}}$$

Computing the gradients of the first term

- the energy function $E(x^{(n)};\theta)$ of the sequence $x^{(n)}$ is

$$E(x^{(n)};\theta) = \sum_{k=1}^{L}\sum_{i=1}^{20} h_{i,k}\cdot \mathbb{1}\{x_k^{(n)}=i\} + \sum_{k=1}^{L-1}\sum_{l=k+1}^{L}\sum_{i=1}^{20}\sum_{j=1}^{20} J_{i,j}^{k,l}\cdot \mathbb{1}\{x_k^{(n)}=i\}\cdot \mathbb{1}\{x_l^{(n)}=j\}$$

- its gradient with respect to the $h_{i,k}$ is

$$\nabla_{h_{i,k}} E(x^{(n)};\theta) = \mathbb{1}\{x_k^{(n)}=i\}$$

- its gradient with respect to the $J_{i,j}^{k,l}$ is

$$\nabla_{J_{i,j}^{k,l}} E(x^{(n)};\theta) = \mathbb{1}\{x_k^{(n)}=i\}\cdot \mathbb{1}\{x_l^{(n)}=j\}$$

Computing the gradients of the first term

- the gradient of first term of the mean log-likelihood function is

$$\nabla_{h_{i,k}} \left[-\frac{1}{N} \sum_{n=1}^{N} E(x^{(n)}; \theta) \right] = -\frac{1}{N} \sum_{n=1}^{N} \mathbb{1}\{x_k^{(n)} = i\} = -\left\langle \mathbb{1}\{x_k^{(n)} = i\} \right\rangle_{\text{data}}$$

$$\begin{aligned} \nabla_{J_{i,j}^{k,l}} \left[-\frac{1}{N} \sum_{n=1}^{N} E(x^{(n)}; \theta) \right] &= -\frac{1}{N} \sum_{n=1}^{N} \mathbb{1}\{x_k^{(n)} = i\} \cdot \mathbb{1}\{x_l^{(n)} = j\} \\ &= -\left\langle \mathbb{1}\{x_k^{(n)} = i\} \cdot \mathbb{1}\{x_l^{(n)} = j\} \right\rangle_{\text{data}} \end{aligned}$$

- $\langle \cdot \rangle_{\text{data}}$ is the average over the data
- the above gradients can be easily evaluated from the MSA

Computing the gradients of the second term

- the partition function $Z(\theta)$ is given by $Z(\theta) = \sum_x e^{-E(x;\theta)}$
- the gradient of the log-partition function with respect to the $h_{i,k}$ is

$$\begin{aligned}
\nabla_{h_{i,k}} \log Z(\theta) &= \frac{1}{Z(\theta)} \nabla_{h_{i,k}} Z(\theta) = \frac{1}{Z(\theta)} \sum_x e^{-E(x;\theta)} \nabla_{h_{i,k}} E(x;\theta) \\
&= \frac{1}{Z(\theta)} \sum_x e^{-E(x;\theta)} \mathbb{1}\{x_k = i\} \\
&= \sum_x P(x;\theta) \mathbb{1}\{x_k = i\} \\
&= \langle \mathbb{1}\{x_k = i\} \rangle_{\text{model}}
\end{aligned}$$

- $\langle \cdot \rangle_{\text{model}}$ is the average over the model

Computing the gradients of the second term

- the gradient of the log-partition function with respect to the $J_{i,j}^{k,l}$ is

$$\begin{aligned}
\nabla_{J_{i,j}^{k,l}} \log Z(\theta) &= \frac{1}{Z(\theta)} \nabla_{J_{i,j}^{k,l}} Z(\theta) = \frac{1}{Z(\theta)} \sum_x e^{-E(x;\theta)} \nabla_{J_{i,j}^{k,l}} E(x;\theta) \\
&= \frac{1}{Z(\theta)} \sum_x e^{-E(x;\theta)} \mathbb{1}\{x_k = i\} \cdot \mathbb{1}\{x_l = j\} \\
&= \sum_x P(x;\theta) \mathbb{1}\{x_k = i\} \cdot \mathbb{1}\{x_l = j\} \\
&= \langle \mathbb{1}\{x_k = i\} \cdot \mathbb{1}\{x_l = j\} \rangle_{\text{model}}
\end{aligned}$$

The gradients of the mean log-likelihood function

- the gradient of the mean log-likelihood function is

$$\nabla_{h_{i,k}} \ell(\theta) = -\left\langle \mathbb{1}\{x_k^{(n)} = i\} \right\rangle_{\text{data}} + \langle \mathbb{1}\{x_k = i\} \rangle_{\text{model}}$$

$$\nabla_{J_{i,j}^{k,l}} \ell(\theta) = -\left\langle \mathbb{1}\{x_k^{(n)} = i\} \cdot \mathbb{1}\{x_l^{(n)} = j\} \right\rangle_{\text{data}} + \langle \mathbb{1}\{x_k = i\} \cdot \mathbb{1}\{x_l = j\} \rangle_{\text{model}}$$

- $\langle \cdot \rangle_{\text{data}}$ is the average over the data, which can be easily evaluated from the MSA
- $\langle \cdot \rangle_{\text{model}}$ is the average over the model
- computing $\langle \cdot \rangle_{\text{model}}$ exactly is intractable
- $\langle \cdot \rangle_{\text{model}}$ can be approximated using Monte Carlo sampling

Monte Carlo sampling from the Potts model

- the Potts model is often sampled using the Gibbs sampling algorithm
- Gibbs sampling is a special case of Markov Chain Monte Carlo (MCMC) sampling
- the idea is to sample from the joint distribution $P(x;\theta)$ by sampling from the conditional distribution of each variable given the others
- the conditional distribution of x_k in a sequence x is given by

$$P(x_k = i|x_{-k};\theta) = \frac{e^{-E((x_k=i,x_{-k});\theta)}}{\sum_{j=1}^{20} e^{-E((x_k=j,x_{-k});\theta)}}$$

- x_{-k} is the sequence x with the k-th position removed
- the conditional distribution of x_k is a multinomial distribution

Sampling from the Potts model using Gibbs sampling

- initialize the sequence x with a random sequence
- for each position k in the sequence, sample x_k from the conditional distribution $P(x_k|x_{-k};\theta)$
- repeat the above step for a number of iterations
- the sampled sequence is $x = (x_1, x_2, \ldots, x_L)$
- the sampled sequence is one sample from the Potts model

Learning a Potts model from a MSA by MLE

- the gradients of the mean log-likelihood function is

$$\nabla_\theta \ell(\theta) = \underbrace{-\frac{1}{N}\sum_{n=1}^{N} \nabla_\theta E(x^{(n)};\theta)}_{\text{the first term}} + \underbrace{\nabla_\theta \log Z(\theta)}_{\text{the second term}}$$

- the first term can be computed exactly from the MSA and the second term can be approximated using Gibbs sampling
- to maximize $\ell(\theta)$, we can use a stochastic gradient ascent algorithm
- in every iteration of the optimization, Gibbs sampling from the Potts model is needed, which makes the optimization expensive

Alternative apporaches to learn a Potts model

- learning a Potts model from a MSA using MLE is expensive
- alternative approaches exist that do not require Gibbs sampling in every iteration and are faster than MLE
  1. mean-field approximation
  2. maximum pseudo-likelihood estimation
  3. approximate MLE using variational inference
  4. noise contrastive estimation
  5. contrastive divergence
- these approaches are based on different approximations of MLE

Maximum pseudo-likelihood estimation

- the likelihood function given a sequence is given by

$$L(\theta) = P(x;\theta) = \frac{1}{Z(\theta)} e^{-E(x;\theta)}$$

- the pseudo-likelihood function given a sequence is given by

$$L(\theta) = \prod_{k=1}^{L} P(x_k | x_{-k}; \theta)$$

- the pseudo-likelihood function is an approximation of the likelihood function
- for a Potts model, it is expensive to evaluate the likelihood function and its gradient whereas the pseudo-likelihood function is easier to evaluate

Maximum pseudo-likelihood estimation of Potts model from MSA

- the mean log-pseudo-likelihood function is given by

$$\ell(\theta) = \frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{L} \log P(x_k^{(n)} | x_{-k}^{(n)}; \theta)$$

- the mean log-pseudo-likelihood function and its gradient can be directly computed from the MSA without Gibbs sampling
- we can maximize the mean log-pseudo-likelihood function using any gradient ascent algorithm

Coupling in Potts model predicts contacts in protein structure

- the Potts model captures the dependency between positions in a MSA of a protein family
- the dependency is represented by the coupling terms $J_{kl}(x_k, x_l)$
- one cause of the dependency is the physical contact between residues in protein structures
- the coupling terms $J_{kl}(x_k, x_l)$ can be used to infer the contacts between residues in a protein structure
- the larger scale of the coupling term, the more likely the two residues are in contact
- for each pair of residues, the coupling term $J_{kl}(x_k, x_l)$ is a matrix of size 20×20
- the matrix norm of the coupling term can be used to measure the strength of the coupling between the two residues

Coupling in Potts model predicts contacts in protein structure

- MSA is from the protein family PF00041
- maximum pseudo-likelihood estimation is used to learn the Potts model
- red triangles are top ranked contacts predicted based on the coupling terms
- blue circles are the actual contacts in the protein structure