Study guide

November 4, 2019

This study guide is intended to help you to review for exams. This is not an exhaustive list of the topics covered in the class and there is no guarantee that these questions are representative of the questions on the exam. You should also review the notes you took in class, the notes and readings on the syllabus, and your homework assignments.

Amino acid substitution models and matrices

- Log-odds formalization.
  - Know how to score an alignment given probability of a match/mismatch.
  - What does the likelihood ratio of an alignment mean? What does the log likelihood ratio of an alignment mean?
  - What does it mean if the likelihood ratio is less than/greater than 1?
  - What does it mean if the log-likelihood ratio is less than/greater than 0?
- Deriving amino acid substitution matrices: overview
  - Substitution models should reflect biophysical properties. Pairs of residues with similar properties represent conservative replacements and should have higher similarity scores than pairs of residues with different properties, which represent non-conservative replacements.
  - Substitution matrices should be parameterized by evolutionary divergence.
  - Given the greater number and variety of amino acids, compared with nucleotides, amino acid substitution models rely more heavily on learning parameters from data than nucleotide models.
  - Two families of amino acid substitution matrices: the PAM matrices and the BLOSUM matrices. Both families were derived according to the following general approach, although the details of each step differ between the two methods.

1. Use a set of "trusted" multiple sequence alignments (ungapped) to infer model parameters.
2. Count observed amino acid pairs in the trusted alignments, correcting for various types of sample bias.
3. Estimate substitution frequencies from amino acid pair counts.
4. Construct a log odds scoring matrix from substitution frequencies.

- The PAM model: The Dayhoff Markov model of amino acid replacement.
  - The unit of divergence used is the PAM or "percent accepted mutation". How is the PAM defined?
  - Dayhoff's PAM matrices are derived from a Markov model of amino acid replacement. What is the basic structure of this model?
  - What are the properties of the data that Dayhoff used to obtain amino acid pair counts for her model? How are those properties related to the underlying assumptions of the Markov chain strategy that she used?
  - How did Dayhoff derive counts from that data set?
  - How did Dayhoff account for potential sample bias in her data?
  - How did Dayhoff use the amino acid counts to derive the PAM transition matrix? How does this derivation account for differences in amino acid frequency and amino acid mutability?
  - How did Dayhoff ensure that her basic model corresponds to exactly 1 PAM of divergence?
  - How is the PAM-N model derived from the PAM-1 model?
  - How are multiple substitutions accounted for in the PAM framework?
  - How are the PAM log odds substitution matrices derived from the Dayhoff Markov model transition matrices?
  - The transition matrices are not symmetric. The substitution matrices are symmetric. What is the biological intuition associated with this observation?
- BLOSUM matrices
  - What are the properties of the data that the Henikoffs used to obtain amino acid pair counts for the BLOSUM matrices?
  - Partitioning sequences into clusters based on percent identity is a key aspect of the BLOSUM method.
    - How are the clusters used in the process of counting amino acid pairs?
    - How does the use of clusters account for sample bias?
    - How does the use of clusters lead to a family of matrices parameterized by divergence?

- Log odds substitution matrices: Both the PAM and BLOSUM substitution matrices are log-odds matrices. You should understand and be able to work with the log odds substitution matrix framework.
  - When a log odds substitution matrix is used to score an alignment, the alignment score corresponds to a log likelihood ratio; what does this mean?
  - How should a positive element in a substitution matrix be interpreted in this context?
  - How should a negative element in a substitution matrix be interpreted in this context?
  - When comparing the main diagonal elements of matrices representing different amounts of divergence, what trends would you expect to see?
  - When comparing the off-diagonal elements of matrices representing different amounts of divergence, what trends would you expect to see?
- What are the similarities and differences between the PAM and BLOSUM models/matrices?
  - What are the major differences between the data used for the BLOSUM matrices and the data used for the PAM matrices?
  - What are the major differences in how sequence divergence is represented in the BLOSUM matrices compared to the PAM matrices?
  - Be able to rank levels of sequence divergence in the two models.

Position Specific Scoring Matrices and the Gibbs sampler

- Position specific scoring matrices (PSSMs)
  - A formalism for modeling ungapped multiple alignments
  - You should be familiar with each step in the calculation of a PSSM from an alignment:
    1. Frequency matrix
    2. Propensity matrix
    3. Log odds scoring matrix
  - Pseudocounts
    - What are they?
    - What is the rationale for using pseudocounts?
    - Understand how to construct a PSSM using pseudocounts.
  - Recognition with PSSMs: You should know how to use a PSSM to score each position in an unlabeled sequence to find new instances of the motif.
  - The score of a window is analogous to a log likelihood ratio. You should understand why this is true. What are the alternate and null hypotheses represented by this likelihood ratio?
  - How are PSSMs similar to amino acid substitution matrices? How do they differ from amino acid substitution matrices?
- The Gibbs sampler
  - In the context of biomolecular sequence analysis, the Gibbs sampler is a motif discovery method based on the PSSM formalism.
  - The Gibbs sampler simulates the stationary distribution of a Markov chain.
    - You should have a basic understanding of this Markov chain
    - What are the states?
    - How are states connected?
  - You should understand the basic structure of the Gibbs sampler algorithm.
  - The Gibbs sampler is guaranteed to find a globally optimal solution. What feature of the algorithm keeps it from getting trapped in local optima?
  - Even though the Gibbs sampler algorithm is guaranteed to converge to a global optimal, running the algorithm several times with different starting configurations is recommended. What is the rationale for this?
  - What is a probability density function (pdf)? What is a cumulative density function (cdf)? You should be able to calculate a cdf from a pdf.
  - You should know how to generate random numbers according to an arbitrary probability distribution, given the cdf of that distribution.

  - What are the underlying assumptions of the Gibbs sampler for biomolecular motif discovery? In what ways are they unrealistic?
  - What implementation decisions must the user make in order to apply the Gibbs sampler to a particular motif discovery problem?

- Limitations of PSSMs
  - PSSMs are designed to model fixed length conserved motifs, such as transcription factor binding sites. You should understand the following limitations of PSSMs and be able to explain how these limitations result from the way in which PSSMs are defined.
    * PSSMs cannot model positional dependencies,
    * PSSMs are not well suited to modeling variable length patterns,
    * PSSMs cannot recognize pattern instances containing insertions or deletions,
    * Boundary detection: PSSMs are not well suited to determining the precise location of transitions between distinct biological regions. Examples of such boundaries include the first membrane-bound amino acid in a transmembrane region, the first nucleotide in a binding site, the beginning of a gene, etc.

Hidden Markov models

- Definitions and terminology
  - The formal definition of an HMM has the following components:
    1. N states $E_1 \dots E_N$
    2. An alphabet, $\Sigma = \{\sigma_1, \sigma_2 \dots \sigma_M\}$
    3. Parameters, λ:
       (a) Initial distribution vector $\pi = (\pi_i)$
       (b) Transition probability matrix a_{ij}
       (c) Emission probabilities: $e_i(\sigma)$ is the probability that state E_i emits $\sigma \in \sum$
  - An HMM is a generative model that emits a sequence $O = O_1, O_2, \dots O_T$ while passing through a sequence of states $Q = q_1, q_2, \dots q_T$. We refer to the sequence of states that emitted O as the "state path".
  - If multiple sequences are under consideration we use superscripts to distinguish them: $O^1, O^2, \dots O^k$, where $O^d = O_1^d, O_2^d, \dots O_{T_d}^d$. Similarly, multiple state paths are denoted $Q^1, Q^2, \dots$, where $Q^d = q_1^d, q_2^d, \dots q_{T_d}^d$.
  - Given a sequence $O = O_1, O_2, \dots O_T$ and a state path $Q = q_1, q_2, \dots q_T$, the joint probability of visiting the states in Q and emitting O is

$$P(O, Q|\lambda) = \pi_{q_1} \cdot e_{q_1}(O_1) \cdot a_{q_1 q_2} e_{q_2}(O_2) \cdot a_{q_1 q_2} \cdot e_{q_3}(O_3) \dots a_{q_{T-1} q_T} e_{q_T}(O_T).$$

  - The total probability that O was emitted by a given HMM, with parameters λ, is

$$P(O) = \sum_b P(O|Q^b, \lambda) \cdot P(Q^b|\lambda) = \sum_b P(O, Q^b|\lambda).$$

  - The sum of $P(O, Q|\lambda)$, over all sequences in Σ^* and all state paths is one:

$$\sum_d \sum_b P(O^d, Q^b) = 1.$$

  - What is meant by the "parameters" of an HMM?
  - What does λ usually refer to in HMM terminology?
  - What is "hidden" in a Hidden Markov model?
  - What is "decoding" and where does this term come from?
- Hidden Markov models (HMMs) are an extension of Markov chains.
  - What properties do HMMs have in common with Markov chains?
  - What features are unique to HMMs?
  - What are the advantages of using an HMM, compared to a Markov chain?

- Motif recognition using HMMs
  - HMMs can be used to answer various questions about patterns in biomolecular sequences. Given a pattern recognition problem in a new biological context, you should be able to determine which of the methods that you have learned in class can be applied to answer the question. In many cases, there may be more than one approach to answering the question. The correct approach may depend on how the HMM is designed.
  - Examples of recognition questions:
    * What is the probability that a given sequence, O, was generated by the HMM? *Example:* Is the sequence a transmembrane protein?
    * What is the state path that emitted a given sequence O? Otherwise stated, the goal is to assign a state to every symbol in an unlabeled sequence, O. *Example:* Identify the cytosolic, transmembrane, and extracellular regions in the sequence. In this case, we wish to assign the labels E, M, or C to each amino acid residue in the sequence.
    * What is the probability of being in state S_i when O_t is emitted? *Example:* Is a given residue localized to the membrane?
  - Calculating the total probability of a sequence, O.
    * The Forward algorithm is a dynamic program that recursively calculates $\alpha(t, i) = P(O_1, O_2, O_3, ...O_t, q_t = E_i)$.
      · What are the initiation, recursion and termination steps of this algorithm?
      · What is the complexity of the Forward algorithm in terms of the the number of states and length of O?
      · Given an HMM and a sequence, O, you should know how to apply the algorithm to calculate $P(O|\lambda)$.
    * The Backward algorithm is a dynamic program that recursively calculates $\beta(t + 1, i) = P(O_{t+1}, O_{t+2}, ...O_T|q_t = E_i)$.
      · What are the initiation, recursion and termination steps of this algorithm?
      · What is the complexity of the Backward algorithm in terms of the the number of states and length of O?
      · Given an HMM and a sequence, O, you should know how to apply the algorithm to calculate $P(O|\lambda)$.
      · Since the Forward algorithm can be used to calculate $P(O|\lambda)$, why is the Backward algorithm needed?

* A common use of the Forward algorithm is to classify a sequence by calculating the probability that it was emitted by a particular model. Typically, we compare the likelihood of the sequence under two competing hypotheses using a log-likelihood ratio:

$$\log \frac{P(O|H_1)}{P(O|H_2)}.$$

  - Often, H_2 is a null hypothesis.
  - Why is it useful to consider the ratio of two likelihoods instead of merely calculating $P(O|H_1)$?
  - What is the benefit of using a log likelihood ratio, instead of just a likelihood ratio?

– Decoding

* Given an unlabeled sequence, the goal of decoding is to classify (i.e., label) each symbol in the sequence with its associated state. In the HMM formalism, we do this by inferring the state path that generated the sequence.
* Viterbi decoding
  - Viterbi decoding assumes that the *most probable path*, $Q^* = \text{argmax}_Q\, P(Q|O, \lambda)$ is the best estimate of the state path that emitted the sequence.
  - The Viterbi algorithm actually calculates $\text{argmax}_Q\, P(Q, O|\lambda)$, rather than $\text{argmax}_Q\, P(Q|O, \lambda)$. What is the meaning of this distinction? Why is this acceptable?
  - The Viterbi algorithm is a dynamic program that recursively calculates $\delta(t, i)$, the probability of emitting $O_1 \ldots O_t$ via the most probable path that ends in E_i.
  - What are the initiation, recursion and termination steps of this algorithm?
  - How does the traceback work?
  - What is the complexity of the Viterbi algorithm in terms of the the number of states and length of O?
  - Given an HMM and a sequence, O, you should know how to apply the algorithm to obtain Q^*.
* Posterior decoding
  - Posterior decoding assumes that the sequence of *most probable states*, $\hat{Q} = \hat{q}_1 \ldots \hat{q}_T$ is the best estimate of the state path that emitted the sequence.
  - The most probable state at time t is the state that has the highest probability of emitting O_t when all possible state paths are considered:

$$\begin{aligned} \hat{q}_t &= \operatorname*{argmax}_i P(q_t = E_i, O_t) \\ &= \operatorname*{argmax}_i \alpha(t, i) \cdot \beta(t + 1, i). \end{aligned}$$

  - The most probable state, $\hat{q}$, can be estimated by using the Forward algorithm to calculate $\alpha(t, i)$ and the Backward algorithm to calculate $\beta(t + 1, i)$.

- The sequence of most probable states may not be a valid state path; that is, it is possible that $P(O, \hat{Q}|\lambda) = 0$. How can that be?

* Comparing Viterbi and Posterior decoding
  - Under what circumstances might posterior decoding provide a better estimate than Viterbi decoding?
  - Under what circumstances might Viterbi and posterior decoding provide the same estimate?